virtana

WorkloadWisdom 6.8.4

User Guide

January 10, 2021

Revision History

Revision #	Date	Software Version
1	02/28/2018	WorkloadWisdom 6.0
2	11/21/2018	WorkloadWisdom 6.2
3	07/19/2019	WorkloadWisdom 6.4
4	12/02/2019	WorkloadWisdom 6.6
5	06/30/2020	WorkloadWisdom 6.8
6	01/10/2021	WorkloadWisdom 6.8.4

Table of Contents

Chapter 1

Introducing Storage Testing and WorkloadWisdom

This section explains the need for storage infrastructure testing, and introduces the concept of WorkloadWisdom.

Storage Testing

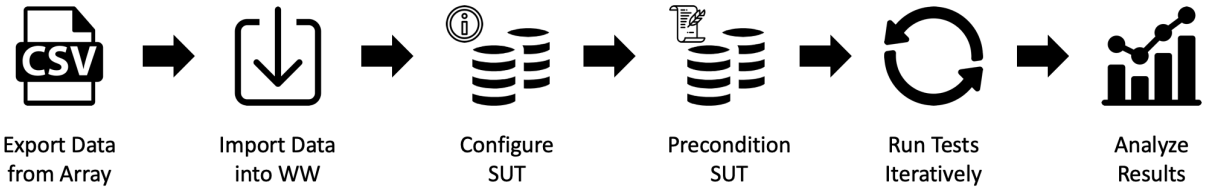
Change in a production storage environment is inevitable. Change can be planned, such as technology change, vendor replacement, firmware upgrade, or it can be unexpected, for example, application change, user behavior change, or outages.

When a change occurs, there is a potential for performance degradation, as the performance of a storage infrastructure is highly dependent on the characteristics of the storage workloads running on it. The most reliable way to de-risk these changes is to measure the performance impact of these changes against your own, unique, workloads. Different organizations have different workload characteristics, especially in a shared SAN/NAS environment where multiple applications coexist.

To perform Storage Testing correctly for your own environment, we recommend the *Best Practices for Evaluating Storage Using Workload Testing and Validation* located

athttp://www.virtualinstruments.com/cnt/uploads/2017/06/WP_LDX_HowTest.pdf.

The methodology can be briefly summarized into the following workflow:

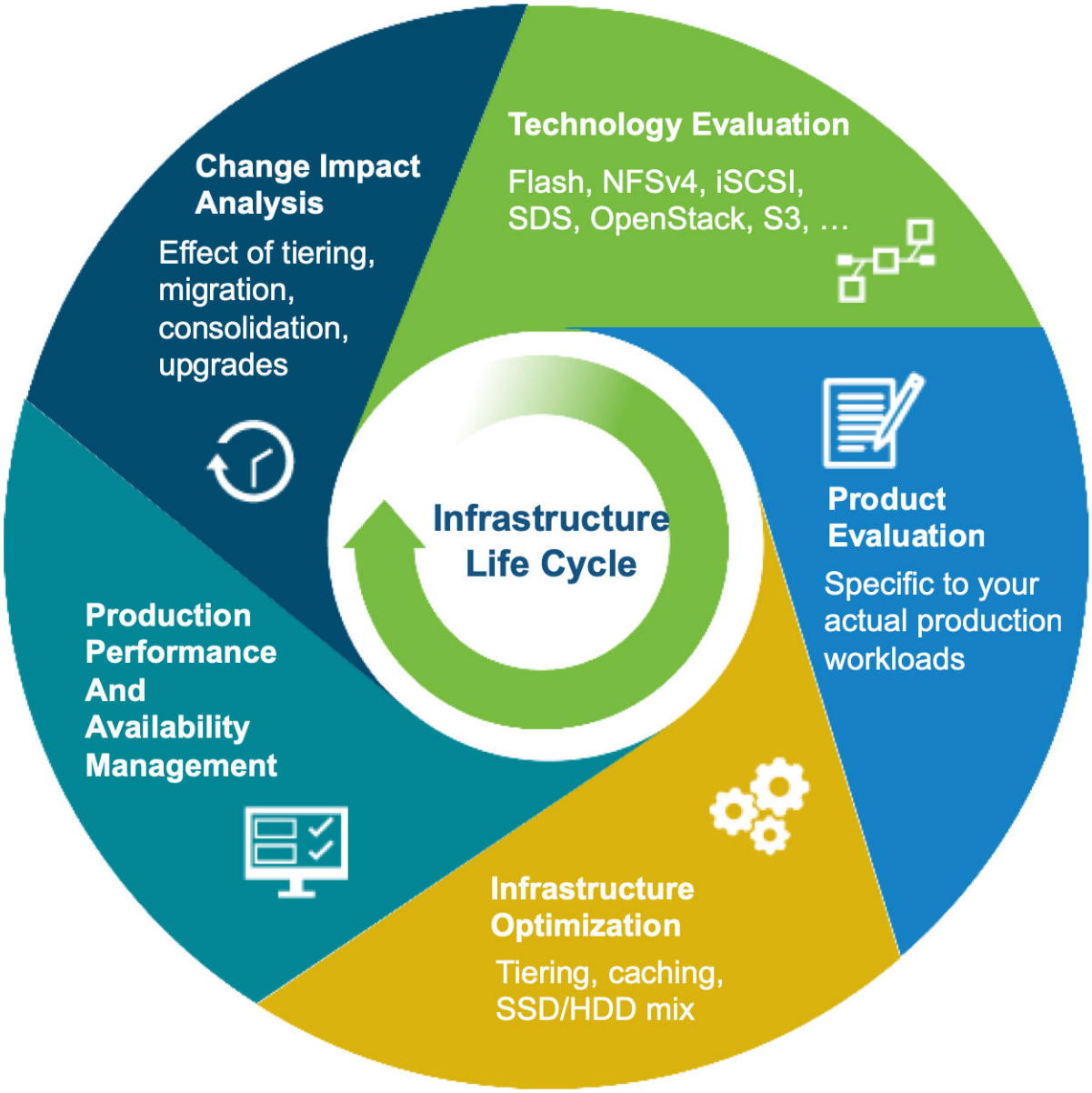


1. *Acquire your own production workload data.* When possible, use your own production workload data as the starting point. No other organization's storage workloads are exactly identical to yours. To acquire your own production workload data, you can either export the data that is recorded by the storage array or 3rd party monitoring tools into a .csv file, or use VirtualWisdom® data that is collected from VirtualWisdom Probes.
2. *Create a realistic workload test.* Use your production workload data, import it into WorkloadWisdom using the Workload Data Importer[1]. The Workload Data Importer analyzes and creates a realistic baseline workload test based on your very own production workload data. Once you have the baseline workload test created, you can scale it up for future proofing, or modify other aspects of the workload test such as data content and many others.
3. *Configure the target storage.* Configure and optimize the storage array according to the vendor's best practices. In addition to configuring the storage array itself, the network between the simulated initiators/clients and the storage also needs to be considered a part of the storage test environment, as it is a mandatory part of every production storage infrastructure.
4. *Pre-condition the array.* Pre-condition the SAN storage arrays prior to running workload Tests, especially for all flash arrays. Flash arrays need to be "broken in" to bring them into the state that they would be in production. Otherwise, the performance results are not realistic.
5. *Run workload tests.* Run the baseline workload test to make sure the test results are as expected. Once the baseline results are confirmed, perform subsequent workload tests by scaling up the load or modifying other parameters, such as block sizes and deduplication ratio. Carefully track the configuration and test results from each workload Test Run.

[1]While WorkloadWisdom makes the task of analyzing your own production workloads as easy as possible, it still requires certain domain expertise and knowledge. If it is your first time performing this task, it is strongly recommended that you consult Virtual Instruments Services team.

6. *Analyze workload test results.* With all workload Test Runs completed and their results identified, create a Report that compares the KPIs from each Test Run and use the data as one of input for change confirmation.

Storage Life Cycle



You can use WorkloadWisdom to obtain storage performance analytics across the storage life cycle:

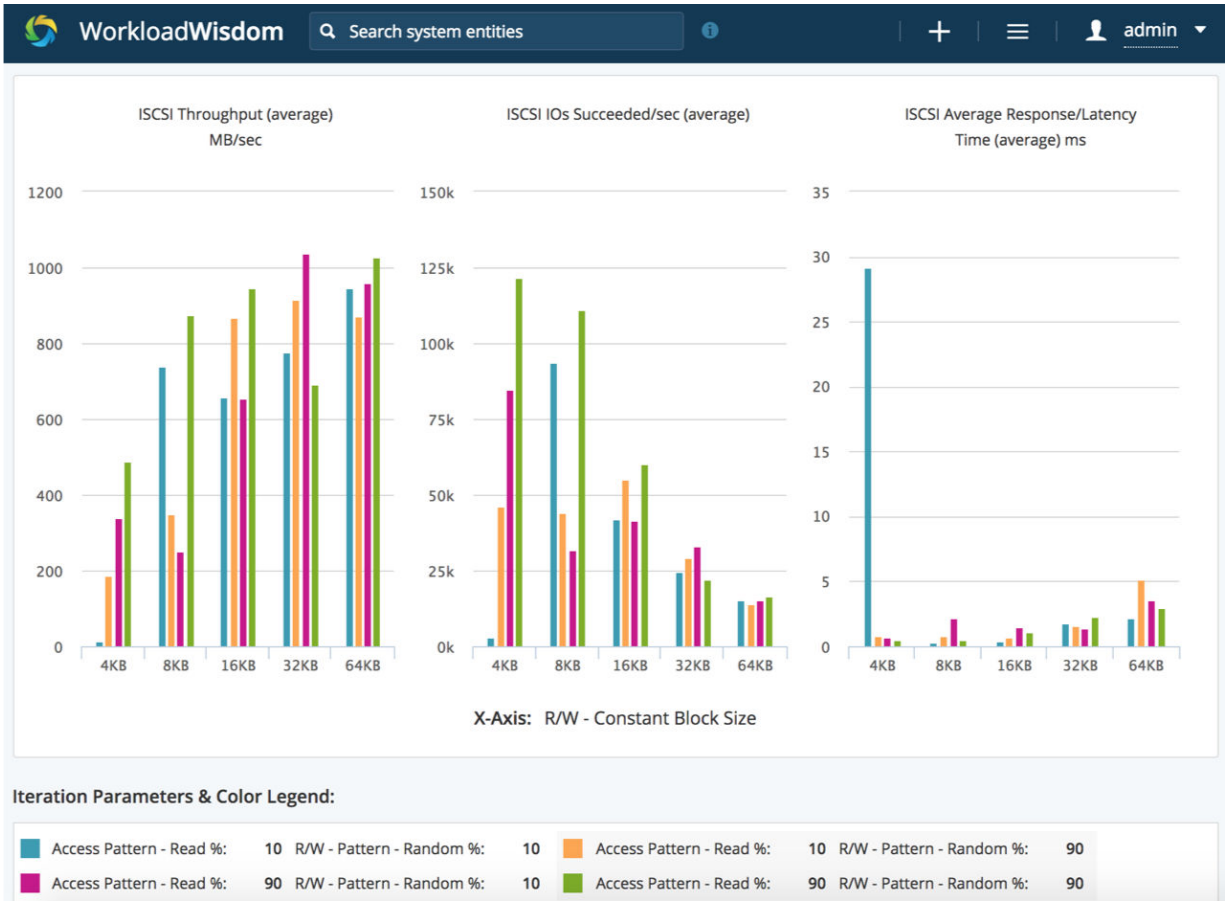
- *Storage Configuration Optimization.* Determine what is the optimal storage price/performance configuration by varying dozens of settings like load, block/file size, compression/de-duplication, tiering, queue depths, and caching.
- *Problem resolution*. Analyze production workloads performance for transient issues or other systemic issues, and quickly reproduce the workload behaviors in the lab for faster resolution.
- *Change Validation*. Verify and load test final storage configurations in a pre-production environment and observer performance in production.
- *Change Management Validation.* Validate the effect on scalability, latency, throughput, and IOPS of application upgrades and firmware updates on storage infrastructure before cutting over to production.
- *Technology Evaluation.* Determine which storage architecture, technology or protocol offers the highest performance or is the most cost-effective when running workloads that reflect your specific applications.
- *Storage Product Evaluation.* Determine which vendor and storage system offers the highest performance or is the most cost-effective when running workloads that reflect your specific applications.

WorkloadWisdom

WorkloadWisdom®, formerly Load DynamiX® Enterprise (LDX-E), is the only Storage Performance Analytics and Performance Validation Solution in the market today that is designed to model your own production workloads. You can use WorkloadWisdom to acquire workload profiles directly from your live production storage environments. You can also recreate those workloads in the lab for planning, evaluation, optimization and problem resolution or troubleshooting.

Understanding real application IO workloads has traditionally been one of the most challenging barriers to storage performance validation. The ability to understand real application I/O workloads marks a major step forward in enabling organizations to perform storage validation across their critical applications.

Whether you work at a storage or network technology vendor or at an enterprise IT organization, WorkloadWisdom simplifies your storage testing and evaluation process and empowers you with an unprecedented level of insight into your application workloads.



The WorkloadWisdom Appliance is a web-based analytics and performance validation management appliance. It is easy to deploy and available as either physical or virtual appliance that provides you with:

- Production workload acquisition and characterization
- Easy to use workload characterization and modeling
- Automated Storage array performance profiling
- Advanced Storage and workload analytics and reporting
- An enterprise class test and lab management

Chapter 2

Deploying WorkloadWisdom

Design Limits and Support

You can use WorkloadWisdom to retrieve, calculate, store, and display thousands of statistics every second per test port, while running a workload test that lasts for days.

However, you can overload WorkloadWisdom if there are too many concurrent workload tests or concurrent test ports running. You can also overload WorkloadWisdom if you simultaneously connect too many concurrent web browsers, as each browser connection consumes resources.

If different users have multiple web browsers and resolution sizes, the screen display experience can vary from one to another, and only a finite number of configurations can be supported and verified completely.

To achieve the best user experience possible, it is strongly recommended that you operate within the following system design limits:

- 32 concurrent test ports, for example:
  - 1 concurrent 32-port test
  - 16 concurrent 2-port tests
  - 2 concurrent 16-port tests
- 7 days maximum test duration

For test runs longer than 3 days, it is highly recommended to leave the Results Dashboard page open in your web browser, in the background. This allows WorkloadWisdom to perform summarization of data points in small increments, and minimizes the need for WorkloadWisdom to summarize days' worth of data points at once, which can cause system performance degradation.

- 4 concurrent users

Recommend 1 open tab per user, as each concurrently opened tab generates load on WorkloadWisdom.

- Fibre Channel lab environment with up to 320,000 ITLs
- Imported TDE workload file maximum size: 100MB
- Supported browsers: Chrome and Safari

You can use WorkloadWisdom with any browser as WorkloadWisdom does not reject connections from unsupported browsers. However, you might experience UI issues with non-recommended browsers. Virtana might not be able to resolve UI issues with unsupported browsers.

- Browser minimum screen resolution: 1024 x 768

If your normal use case exceeds these specifications, contact Virtana Support.

Get Started with WorkloadWisdom

Logging In

Usernames in WorkloadWisdom are always email addresses.

Login to WorkloadWisdom using the default Username: admin@example.com and default Password: welcome.

NOTE

Unless you are required by policy to create unique accounts for each individual user, it is recommended that the users share the Administrator account instead of creating unique User accounts for different users. User type accounts cannot perform many System Management tasks, and do not have visibility into other Users' Resources and Data if their privacy settings are not set to Public.

The first time that you log in, the End User License Agreement (EULA) displays. Scroll to the bottom of the EULA and click **Accept**. Once you accept the EULA, the WorkloadWisdom *Home* screen displays.

If you are a WorkloadWisdom administrator, the first thing that you should do is change your user ID. See Changing the Administrator's User ID and Password [14]. Make sure you keep the new user ID and password in a safe and retrievable location, as calling Support is required if you forget the user ID or password.

If you are not a WorkloadWisdom administrator, go to Using the Home Page [15].

Changing the Administrator's User ID and Password

Administrators can change their user ID (email address) and password as follows:

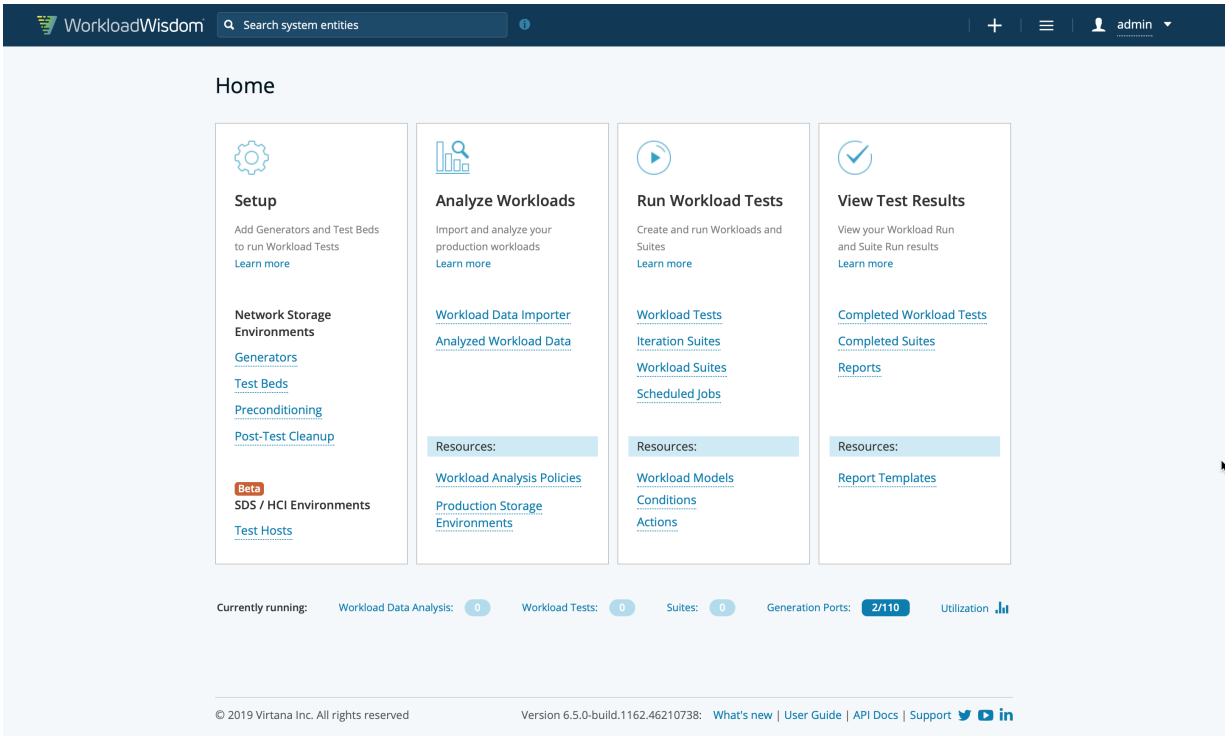
1. From the *admin* drop-down menu, select *My Profile*.
2. Enter the following information on the *My Profile* screen:
   - Your first and last name
   - Your email address
   - Your new password
   - Confirmation of your new password
   - Your current password
3. Click **Update**.

NOTE

If you forget your password, you cannot recover it. Contact Virtual Instruments Support for assistance.

Using the Home Page

When you login to WorkloadWisdom the first screen displayed is the *Home* screen.



The main part of the *Home* screen is structured into four vertical sections. Each section lists tasks that are associated with it. If there are resources associated with the section, they are listed below the *Resources* label.

- *Setup*. Contains links to manage test environment Resources such as Workload Generators, Test Beds, and Preconditioning workloads for storage arrays.
- *Analyze Workloads*. Contains links to acquire and analyze workloads from your live production storage or lab environments.
- *Run Workload Tests*. Contains links to test running Resources such as Workload Tests and Suites.
- *View Test Results*. Contains links to view the Data from Workload Test Runs and Suite Runs.

You cannot use the *back* function associated with your browser to navigate in WorkloadWisdom.

Using the Title Bar and Accessing Documentation

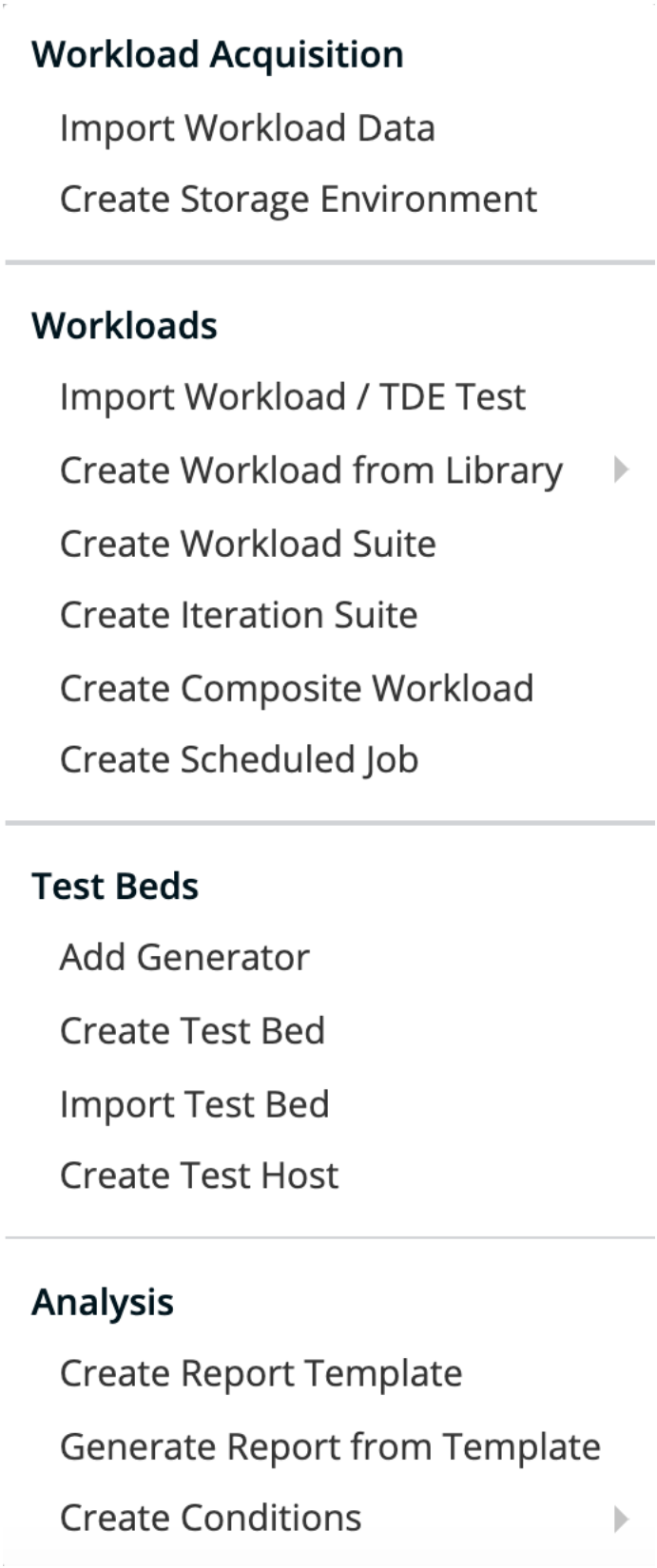
Every WorkloadWisdom screen, including the *Home* screen, has a title bar at the top. The title bar gives you quick and easy access to some WorkloadWisdom capabilities.

Search Box

Use the search box, populated with the text “Search system entities,” to search for resources by name or tag.

New Item Menu

Click  , also known as the *New Item* menu to access the following drop-down menu from which you can quickly access various WorkloadWisdom tasks.

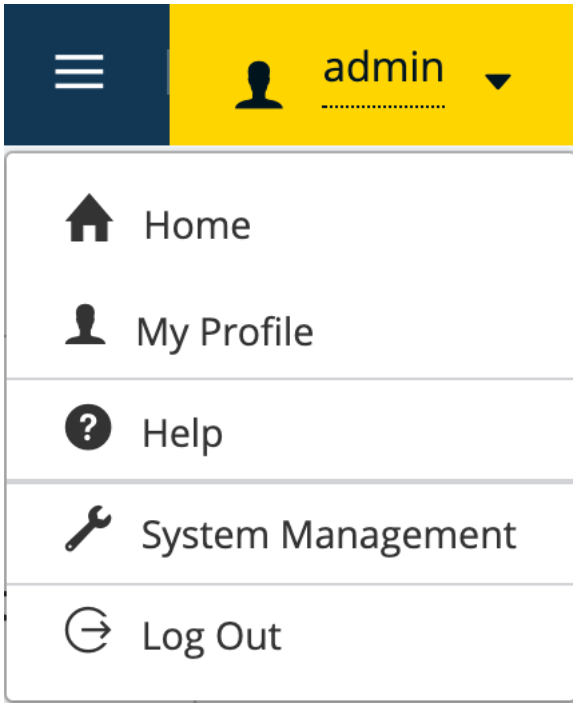

Workload Acquisition
Import Workload Data
Create Storage Environment
Workloads
Import Workload / TDE Test
Create Workload from Library
Create Workload Suite
Create Iteration Suite
Create Composite Workload
Create Scheduled Job
Test Beds
Add Generator
Create Test Bed
Import Test Bed
Create Test Host
Analysis
Create Report Template
Generate Report from Template
Create Conditions

Task Shortcut Menu

Click , also known as the *Task Shortcut* menu to access a shortcut that brings the four vertical sections of WorkloadWisdom tasks from the *Home* page into view. You can access any of the tasks from this shortcut view.

User Menu

The drop-down *User* menu displays as either *admin*, if you are an administrator, or <your name> if you are not an administrator. Use this drop-down menu to access the *Home* screen, your user profile (*My Profile*), the *WorkloadWisdom User Guide* (*Help*) or to logout (*Log Out*). Only administrators have assess to the *System Management* option, which also only displays for administrators.



WorkloadWisdom Documentation

In the lower righthand corner of the *Home* screen, and all screens, there are links to WorkloadWisdom documentation. Click the title of the documentation that you would like to access to view it.

Chapter 3

Setting Up WorkloadWisdom

Generators

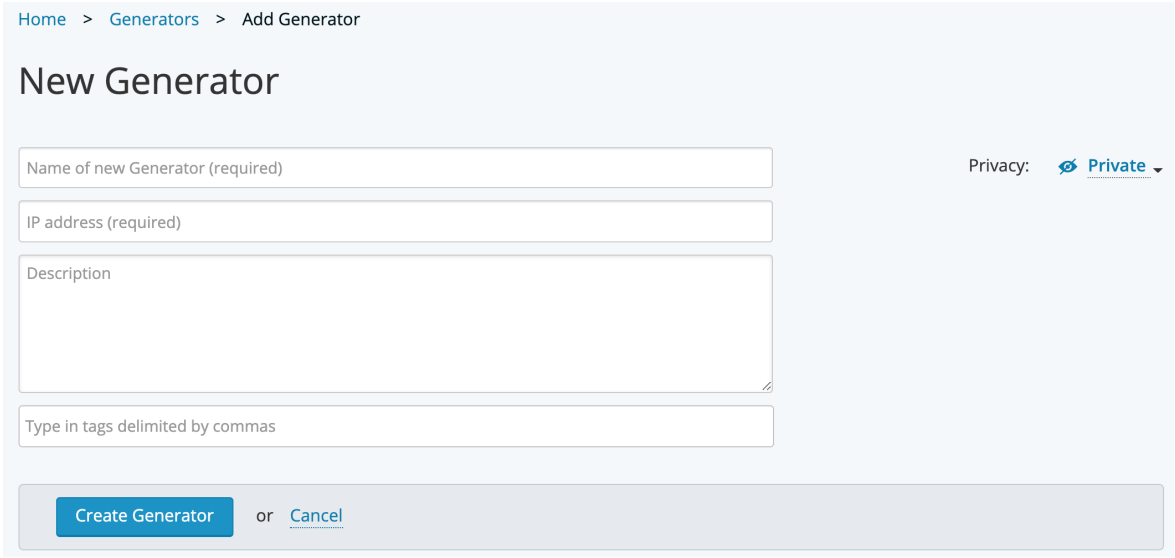
Workload Generators, formerly known as LoadGen Appliances or Load Generation Appliances, generate IOs on the wire over Ethernet or Fibre Channel interfaces that the System Under Test will receive, as if it was receiving them from real storage protocol clients. A Workload Generator emulates one or potentially millions of user or client applications accessing the storage infrastructure using supported file, object, or block protocols.

WorkloadWisdom provides the ability to share, analyze results, and centrally manage Workload Generators.

Adding a Workload Generator

Follow these steps to add a Workload Generator to WorkloadWisdom:

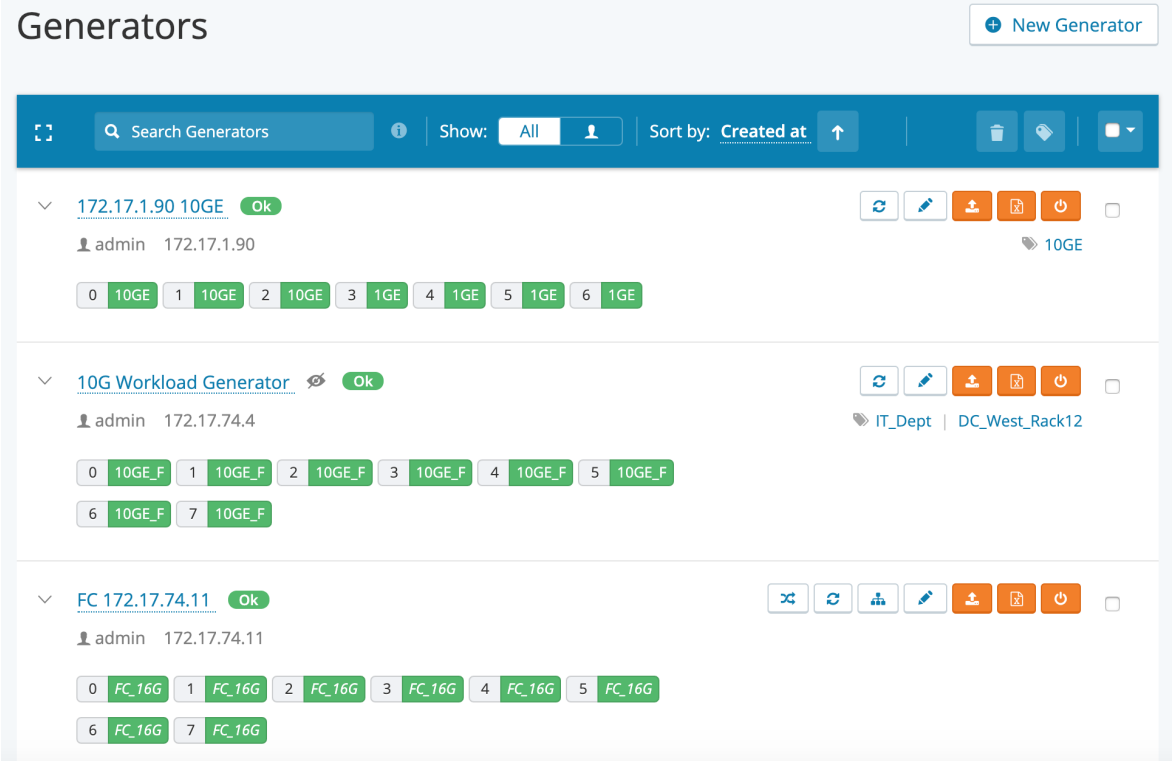
1. Click *Add Generator* from the *New Item Menu* drop-down menu.



2. Enter the following information for the Workload Generator:
   - *Name* for the Workload Generator.
   - *IP Address* of the Workload Generator.
   - Optionally, a *Description* of the Workload Generator.
   - Optionally, a *Tag List* to facilitate searching.
   - Choose the *Privacy* drop-down to show or hide the Workload Generator from the public generator list.

> **CAUTION**
> If you set the *Privacy* to Private, then other User accounts will not be able to see or use this Workload Generator.

3. Click **Create Generator** to add the Workload Generator.
4. Confirm that the Workload Generator appears in the list of generators. Select Generators from the bread crumbs list at the top-left of the screen. The Workload Generator that you added should display in the list.



NOTE

Each of the Workload Generator Test Ports is listed along with their speed and current state. A red port state indicates that the Test Port is currently in use.

Managing Workload Generators

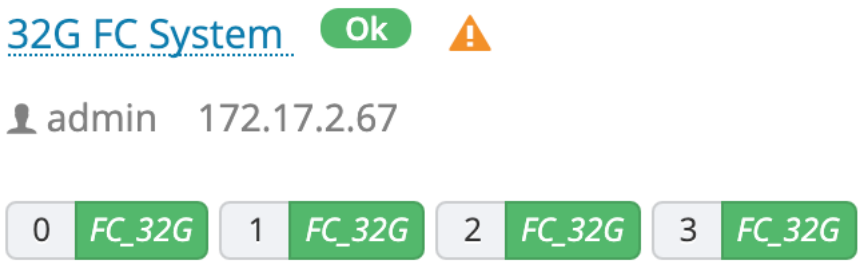
The *Generators* screen displays a summary list of Workload Generators to which you have access. The information on this screen is refreshed every ten seconds.

You can perform a variety of tasks associated with Workload Generators by using the white and orange buttons to the right of each Workload Generator in the list. Hovering over each of the orange buttons displays its function.

> **NOTE**
> Throughout this User Guide, the terms FC Workload Generator or Fibre Channel Workload Generator without the SB or SANBlaze prefix apply to Virtana FC-SCSI Workload Generators only. All references to SANBlaze Workload Generators have the SB or SANBlaze prefix, such as “SB FC-NVMe Port”.

1. If you are working with a Fibre Channel or a SANBlaze Fibre Channel Workload Generator:
   a. For Fibre Channel Workload Generators
      i. Click **Rediscover Targets** to rediscover all Fibre Channel targets that are connected to the Workload Generator.
      ii. Click **Reload FC port information**to scan all the accessible LUNs.

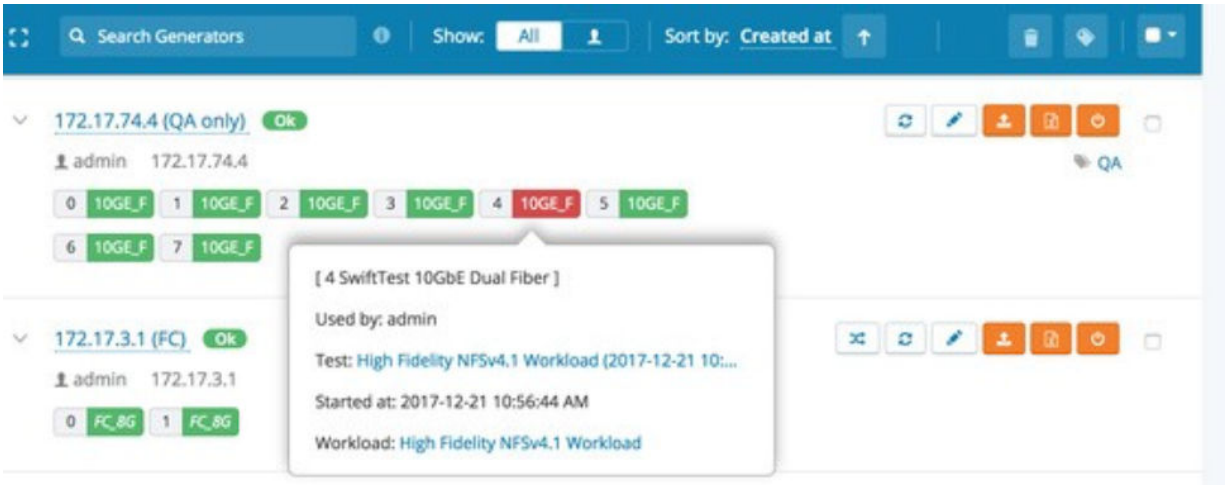
      > **NOTE**
      > If the ⚠ icon displays next to the Workload Generator name as shown below, that means a detectable change was identified for the list of accessible LUNs. Click **Reload FC port information**.

      iii. Proceed to Step 2.
   b. For SANBlaze Fibre Channel Workload Generators
      i. If the port is not in the correct mode, click on **Switch to NVMe** or **Switch to SCSI** to correct the port mode.
      ii. Click **Rediscover Targets** to rediscover all Fibre Channel targets and rescan LUNs that are connected to the Workload Generator.
      iii. Proceed to Step 2.
2. Click **Refresh** to retrieve the latest information about the Workload Generator, such as version number and port status.

3. Click **Edit** to edit information about the Workload Generator, such as name and IP address.
4. Click **Update Firmware** to update the Workload Generator's firmware.
5. Click **Licensing** to view or modify the Workload Generator's license.
6. Click **Reboot Generator or Port** to restart the Workload Generator software or reboot a Workload Generator port.

If a port is currently idle and available to run workload tests, it displays with a green light. If a port is currently busy, it displays with a red light. You can click a busy port and view additional information to determine what the port is doing, and when the port is available again.



When you click a busy port, the following information is provided:

- *Used by*: Username of the WorkloadWisdom user that is currently accessing the port.

If the port is busy from outside of WorkloadWisdom, for example, directly from the TDE, the generic name of `TDE/API/Other` displays.

- *Test*: Name of the active workload run that is running on the port.

Test and *Workload* are hyperlinked. You can click *Test* to access the results of the test run, or click *Workload* to access information about the workload model.

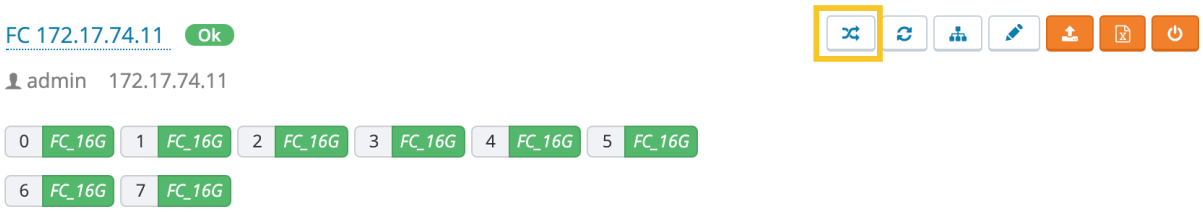
- *Started at*: Timestamp or the time that the workload run started.
- *Workload*: Workload Template from which the test was spawned.

Test and *Workload* are hyperlinked. You can click *Test* to access the results of the test run, or click *Workload* to access information about the workload model.

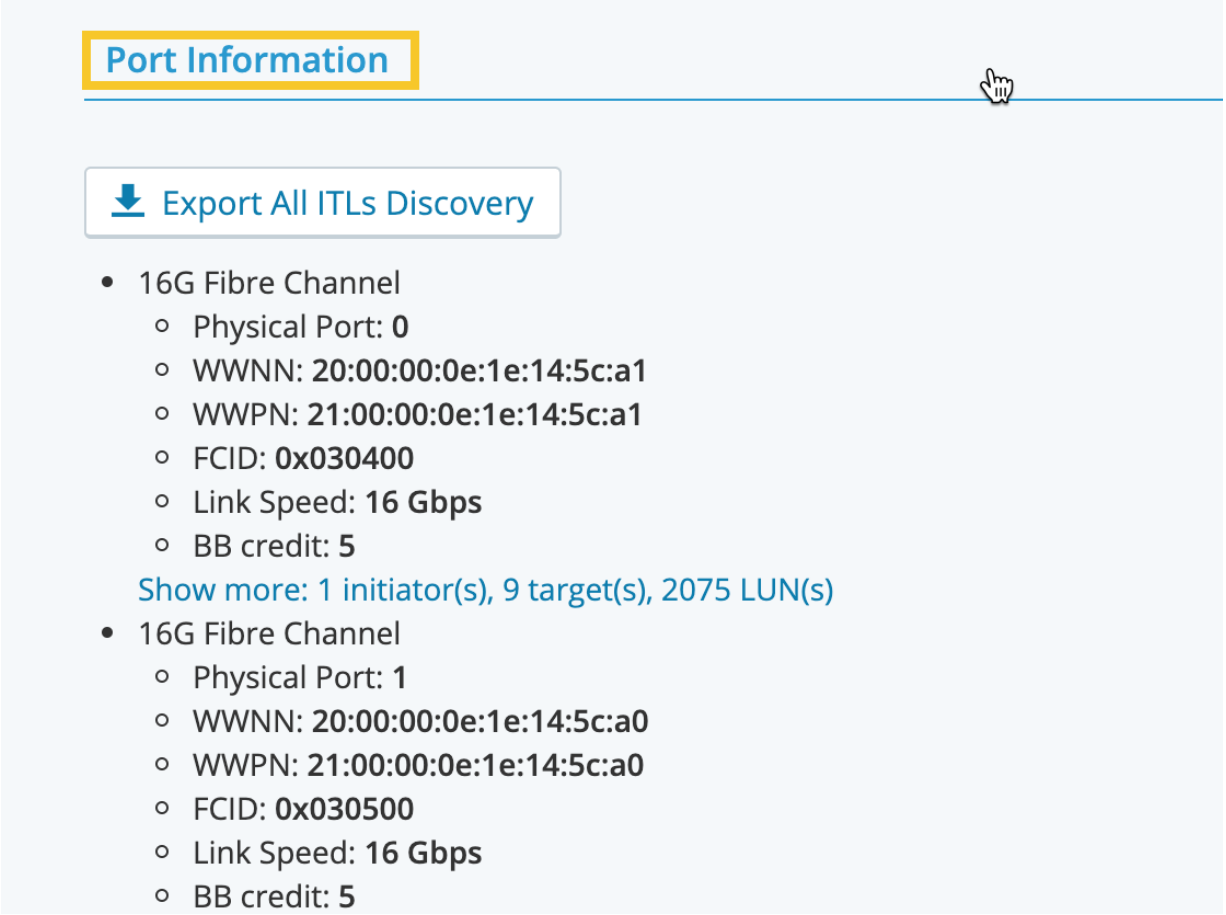
Fibre Channel Workload Generators: Targets and LUNs

Due to the nature of Fibre Channel in which devices automatically log in and perform discovery upon link up, the FC Workload Generators automatically log in and perform discovery of Targets and LUNs upon initialization. The discovery of Targets / LUNs and the extraction of LUNs details are two separate functions that are triggered separately.

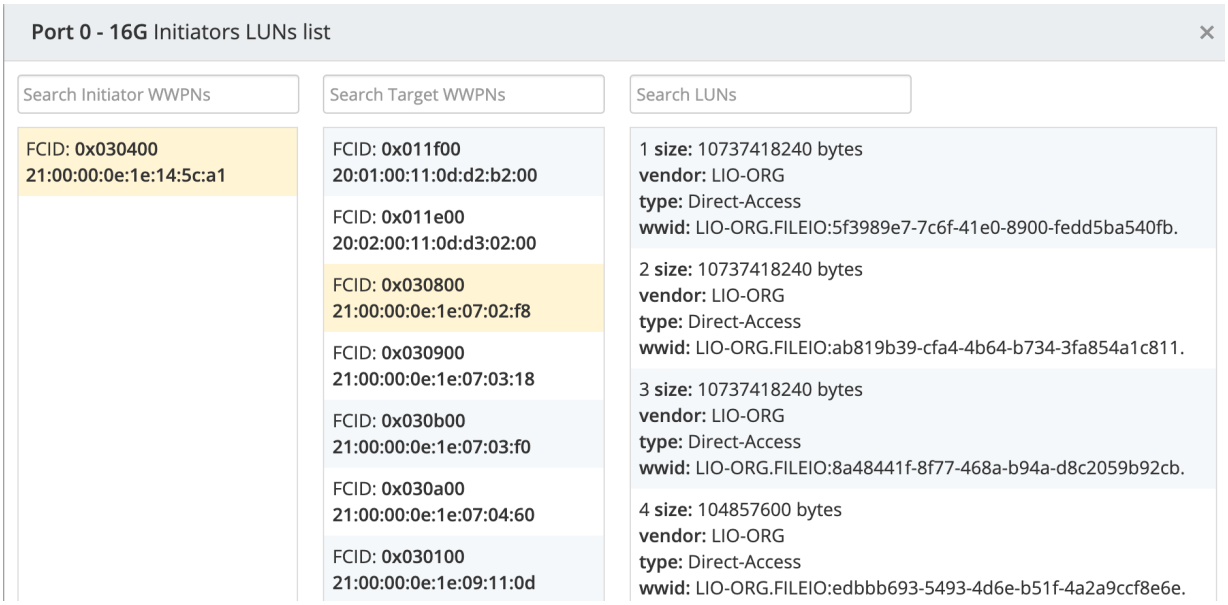
To initiate a discovery of the latest Targets and LUNs at any time, click on the *Rediscover Targets* button.



Depending on the size of the fabric, the discovery process can take seconds to minutes. To view the results of the Target discovery, click on *Port Information* to expand the section.



To extract the details of the Targets and LUNs, such as Target WWPN and LUN ID, both of which are required information to have when creating an FC Test Bed, click on the *Show more* link to bring up the **Port Initiators LUNs list** window.



This window shows exactly which Targets and LUNs are accessible to the selected Initiator. This is the information that is used when you create an FC Test Bed. Therefore, if you have a lab environment in which the Targets and LUNs change frequently, it is best practice to go to the *Generators* page to initiate Target discovery and extract Target/LUN information before creating or editing an FC Test Bed.

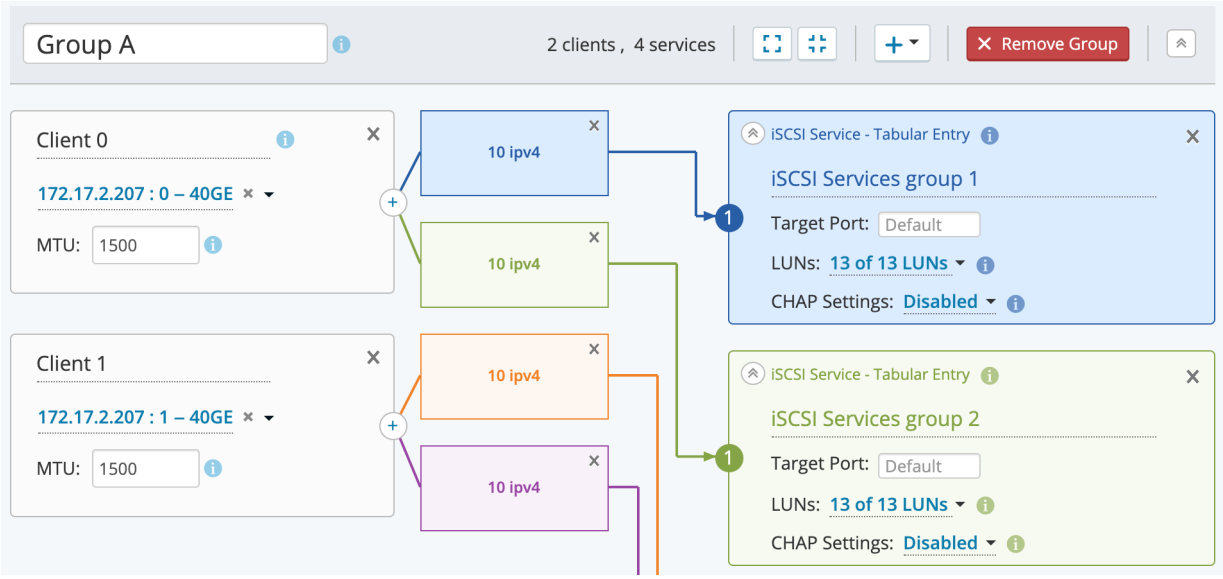
> **CAUTION**
> If there are a lot of Initiators/Targets/LUNs, it may take up to several minutes for the entire dataset to be completely loaded, and during this time, the window cannot be closed. For example, if you have an environment with approximately 200,000 ITLs, it could take approximately two minutes for the entire dataset to be loaded.

Test Beds

A Test Bed is a test environment in which a Workload Test runs. You can separate the test definition and environment by using a Test Bed. This makes it possible to easily and flexibly run a Workload Test on different test environments, by simply selecting a different Test Bed when you run a Workload Test.

At a high level, a Test Bed defines the following:

- One or more Workload Generator client Test Ports
- One or more client and/or initiator addresses
- One or more Services, often the System Under Test, where each Service is defined by:
  - A storage protocol
  - A server IP address or an FC Target WWPN
  - Additional mandatory information for the selected storage protocol such as Shares for SMB or AWS Region for Amazon S3
- One or more Links that connect a Workload Generator client Test Port to a Service



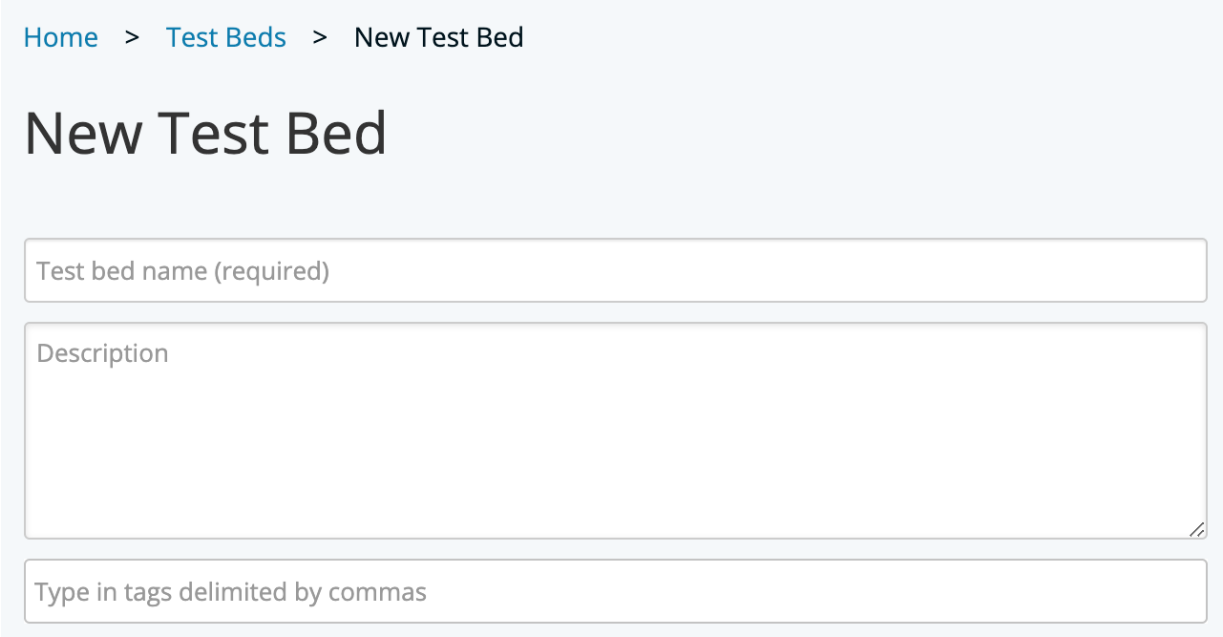
Using New Test Beds

There are two options to create a new test bed: *Create New Test Bed* and *Import Test Bed*.

Both options are accessible from the *Test Beds* screen using the *New Test Bed* drop-down menu, or by selecting these options from the *New Items* drop-down menu.

- *Create New Test Bed*. Create a new test bed from a blank template. This is the most commonly used option.
- *Import Test Bed*. Import an existing test bed that has been previously exported.

Creating a New Test Bed



Assign a *Test Bed name* so you can easily identify your Test Bed. You can optionally add a *Description* to the new Test Bed.

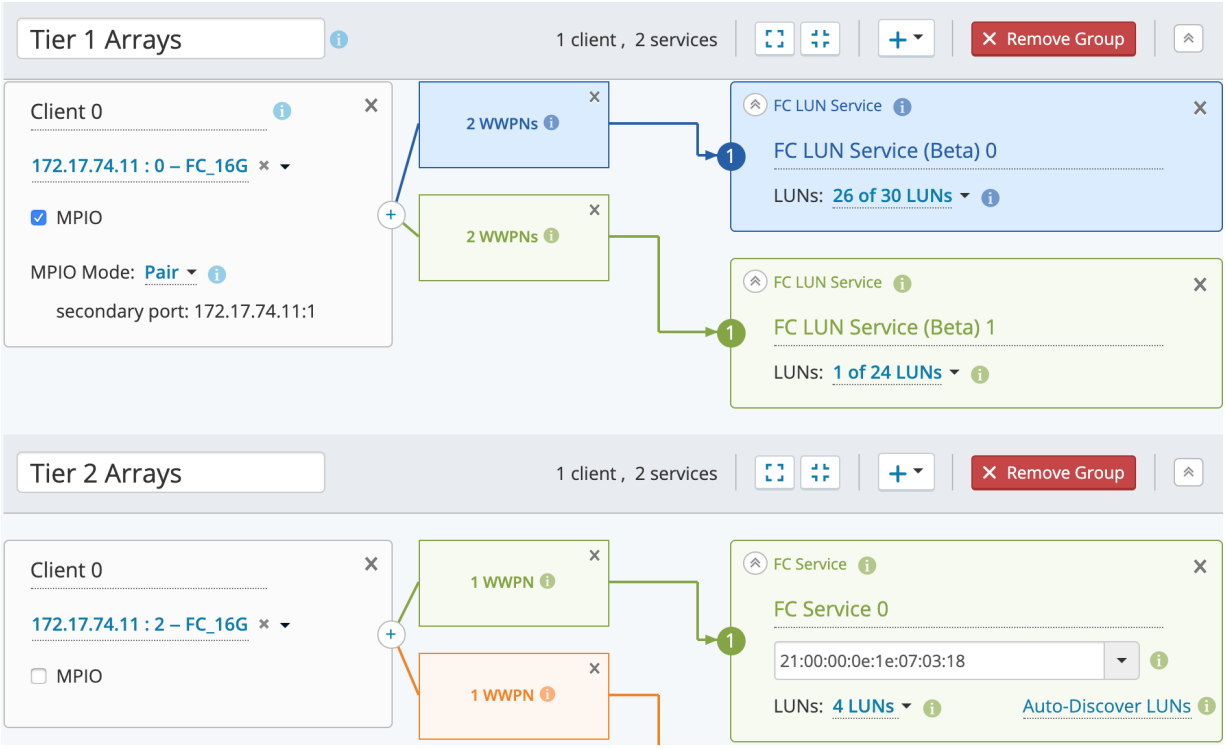
Optionally, add one or more *Tags* to describe the Test Bed. Test bed tags and names are searchable in WorkloadWisdom.

Using Test Bed Groups

A Test Bed Group is a logical grouping of one or more Workload Generator client Test Ports to SUT Services.

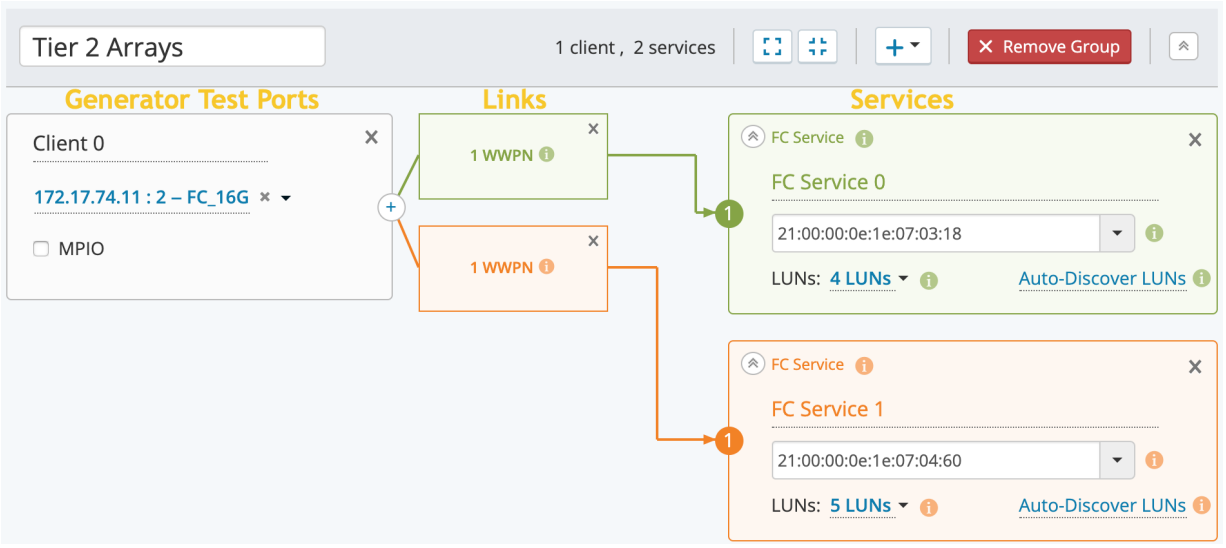
While the Test Bed Group itself does not provide any tangible functionality, in a Test Bed with a large number of Links, or in a Test Bed with multiple storage arrays allocated to different workloads, Test Bed Groups are useful in logically separating the different parts of the test environment.

In the example below, two test bed groups are used to separate the SUT services that are allocated to two different workloads.



Workload Generator Port to SUT Service Links

At a minimum, a functional test bed contains one Workload Generator client port, one SUT service, and one link to connect the Workload Generator client port to the SUT service.



- *Generator Client Test Port*. Test Port from a Workload Generator.

- *SUT Service*. Storage Service that is provided by the SUT, along with additional access information that is required for the selected storage protocol.
- *Link*. Connection between one Workload Generator Client Port and one SUT Service. The exact content of the Link varies depending on the Test Port and the SUT service that it is connecting. In most cases, the Link defines the addresses used by the Test Port for the selected protocol.

Workload Generator Client Test Port

The following Workload Generator Client Test Ports are available:

- *FC Port*: FC-SCSI Test Port on a Workload Generator that supports 4, 8, 16, and 32GFC
- *FCoE 10GE Port*: FCoE Test Port on a Workload Generator that supports 10GbE
- *10GE Port*: 10 GbE Test Port on a Workload Generator
- *25GE Port*: 25GbE Test Port on a Workload Generator
- *40GE Port*: 40 GbE Test Port on a Workload Generator
- *1GE Port*: 1 GbE Test Port on a Workload Generator
- *Virtual Port*: Virtual Test Port on a Virtual Workload Generator

The following SANBlaze Workload Generator Client Test Ports are available:

- *SB FC-NVMe Port*: FC-NVMe Test Port on a SANBlaze Workload Generator that supports 16 and 32GFC

Ethernet Ports

MTU can be set for all Ethernet Ports. The default is 1,500 bytes, and the maximum is 16,128 bytes for Workload Generators with 10GbE or higher speed test ports. A common “jumbo frame” used in the industry, in particular with iSCSI, is 9,000 bytes.

FC Ports

When MPIO is enabled on the FC Port, certain logics are implemented to provide better usability by reducing the amount of manual configuration required. When MPIO is enabled, the following MPIO Group options are available:

- *Single*: One Workload Generator port is used to generate I/Os to the destination target/LUNs over multiple Paths
- *Pair*: Two Workload Generator ports are hardcoded into a single MPIO Group used to generate I/Os to the destination target/LUNs over multiple paths, in which the following pair options are supported:
  - MPIO Group 0: FC port 0 and FC port 1
  - MPIO Group 1: FC port 2 and FC port 3

  - MPIO Group 2: FC port 4 and FC port 5
  - MPIO Group 3: FC port 6 and FC port 7
- *Quad*: This option appears only for Workload Generators with more than four ports. Ports are hardcoded into one MPIO group used to generate I/Os to the destination target/LUNs over multiple paths.
- *All Ports*: All ports on a Workload Generator are hardcoded into one MPIO group used to generate I/Os to the destination target/LUNs over multiple paths.
- *Port Pairs*: Number of sets of pair mode.
  - *Two pairs*: Two sets of pair mode
  - *Four pairs*: Four sets of pair mode

Each MPIO group has a primary port and a secondary port. The notion of primary port and secondary port becomes important in MPIO failover/failback scenarios. When MPIO is enabled, the port you select as the Workload Generator client port assumes the primary port role. The other port in the same MPIO group automatically assumes the secondary port role.

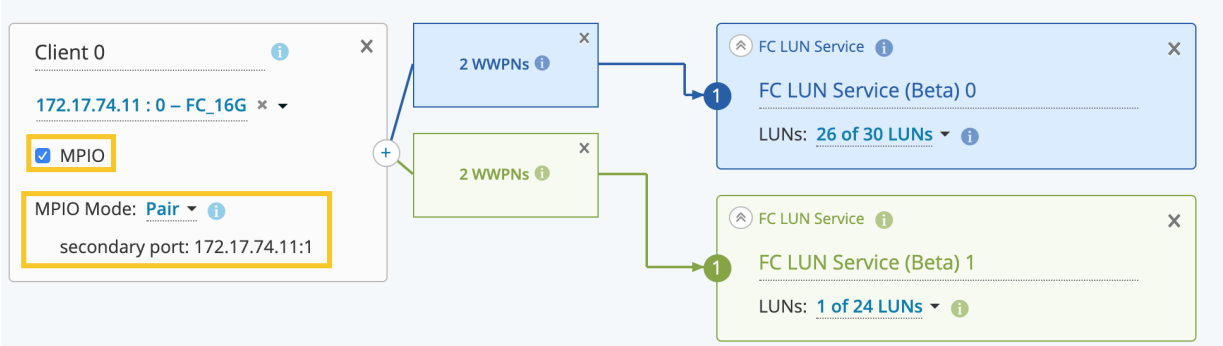
NOTE

All Test Ports inside an MPIO Group share the Group's performance resources. Each MPIO Group has a performance limit that is less than the total Generator limit. As the number of Test Ports inside the same MPIO Group increases, the maximum output does not increase linearly. Using multiple *Port Pairs* enables you to maximize the load potential of the Generator, as there are multiple MPIO Groups being used. For example, if you have an 8-port Generator, you would be able to generate more load by using *Four Pairs* (a *Port Pairs* option) than using *All Ports*. The disadvantage is that in *Port Pairs* mode, each MPIO Group has only two Test Ports, which means there is only one backup port in case one port fails. Whereas in *All Ports* mode, you have seven backup ports.

TIP

If you are primarily running a load balancing test in which you want to maximize the generated load, and if you want to use more than two Test Ports, use *Port Pairs*. If you are running a failover test in which you will intentionally tear down the primary link during the test run, and if you want traffic to run only on the primary link before the tear down, use *Quad* or *All Ports*.

In the following example, for FC Port 0 on the FC Workload Generator, 172.17.3.1 is the primary port in an MPIO Group, and FC Port 1 is automatically designated as the secondary port in the same MPIO Group.



System Under Test (SUT) Service

The following SUT Services are available:

- *FC LUN Service*: Enhanced FC Service or FC MPIO Service designed for multipathing. This service allows all possible paths to one or more LUNs sharing the same WWID to be automatically configured, without requiring the user to specify the exact path between each Initiator and Target.
- *SB FC-NVMe Service*: SUT that supports FC-NVMe Service designed for multipathing that allows all possible paths to one or more namespaces to be automatically configured.
- *iSCSI Service-Tabular Entry*: Enhanced iSCSI Service that supports auto-discovery of iSCSI Targets and LUNs. Also supports entering iSCSI Targets and LUNs in tabular format or by importing a CSV file.
- *NFS Service*: SUT that supports NFS. One IP address with at least one valid share must be properly configured and accessible by the Workload Generator's NFS Clients. Optional configurations include Authentication methods.

- *SMB Service*: SUT that supports SMB. One IP address with at least one valid share and at least one valid username and password, and optional domain, must be properly configured and accessible by the Workload Generator's SMB clients. Optional configurations include Authentication methods and Packet Signing.
- *Amazon™ S3 Service*: SUT that supports S3 API over HTTP or HTTPS. One IP address, authentication method, authentication keys, and AWS Region must be properly configured and accessible by the Workload Generator's S3 Clients. Optional configurations include custom HTTP Request Headers. If HTTPS is used, optional configurations include Client certificate, and Server certificate.
- *Op-ST Swift Service*: SUT that supports OpenStack-Swift API. One IP address, authentication method, authentication keys and base request URI must be properly configured and accessible by the Workload Generator's OpenStack Swift Clients. Optional configurations include custom HTTP Request Headers.
- *More*: less commonly used or obsolete Test Beds. Obsolete Test Beds are maintained for at least one additional release, but will be removed eventually.
  - *FC Service*: SUT that supports Fibre Channel. One target WWPN with at least one valid LUN must be properly configured and accessible by the Workload Generator's FC initiators.

    NOTICE
    While supported for backward compatibility reasons, this functionality is largely made obsolete by the FC LUN Service. It is recommended to use the FC LUN Service by default, unless you require the ability to specify the exact path between an Initiator and a Target.

  - *FC MPIO Service*: SUT that supports Fibre Channel with MPIO enabled. Two target WWPNs with at least one valid LUN per Target WWPN must be properly configured and accessible by the Workload Generator's FC initiators. When ALUA is also enabled in the Fibre Channel Workload, the LUNs must be configured to hold the same WWID.

    NOTICE
    While supported for backward compatibility reasons, this functionality is largely made obsolete by the FC LUN Service. It is recommended to use the FC LUN Service by default, unless you require the ability to specify the exact path between an Initiator and a Target.

  - *iSCSI Service*: SUT that supports iSCSI. One Target IQN with an IP address and at least one valid LUN must be properly configured and accessible by the Workload Generator's iSCSI initiators.

> **NOTICE**
> While supported for backward compatibility reasons, this functionality is largely made obsolete by the iSCSI Service-Tabular Entry. It is recommended to use the iSCSI Service-Tabular Entry by default, unless you need to create an iSCSI test environment in which the Initiator IP addresses are not contiguous or masked to different subnets.

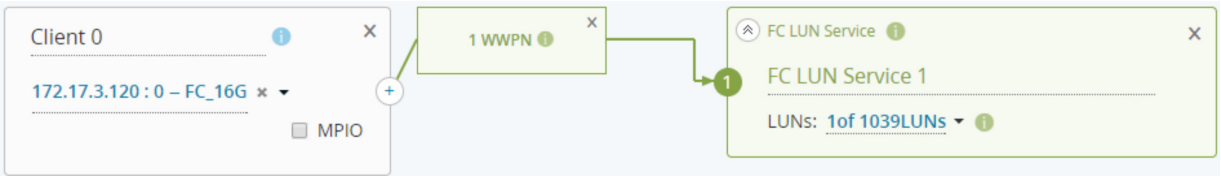
- *HTTP Service*: SUT that supports HTTP or HTTPS. One IP address must be properly configured and accessible by the Workload Generator's HTTP Clients.
- *Amazon™ S3 Service (obsolete)*: SUT that supports S3 API. One IP address, authentication method, authentication keys, and AWS Region must be properly configured and accessible by the Workload Generator's S3 Clients. Optional configurations include custom HTTP Request Headers.

Block Storage Services

Configure Test Beds that provide block storage services: FC-SCSI, FC-NVMe, and iSCSI

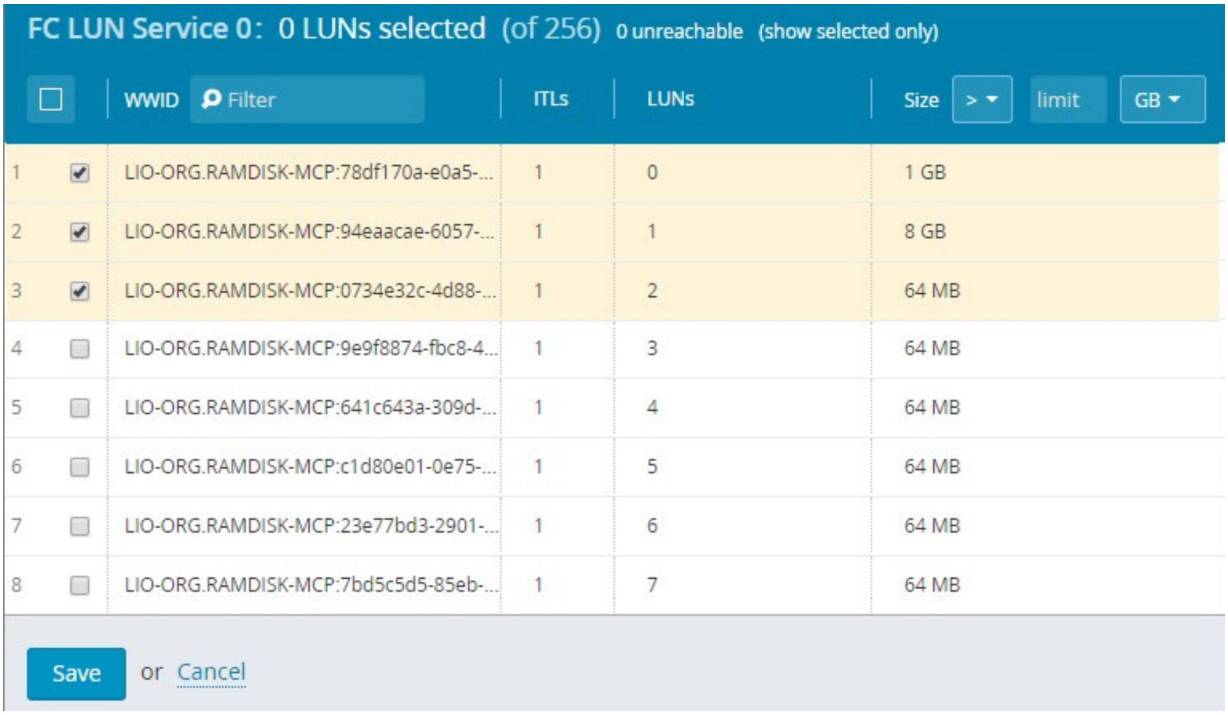
FC LUN Service

Use the FC LUN service to automatically configure and use all available paths from a WorkloadWisdom initiator to one or more LUNs.



Add an FC LUN service, connect a link from the Workload Generator FC Port to the FC LUN Service. Click on the *LUNs: 0 of n* link to open the drop-down menu with LUNs that are accessible to the selected Workload Generator FC Port.

Click the *LUNs* drop-down menu to see a list of discovered WWIDs, that is, a collection of LUNs with the same WWID, the number of ITLs available to the WWID, and the reported WWID associated with the LUNs. Select a WWID to use in the test bed



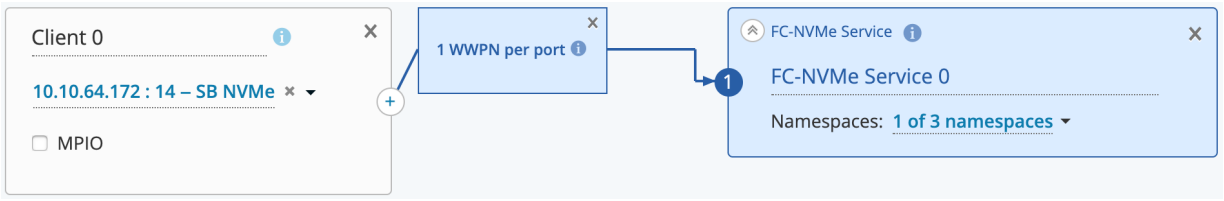
You are not able to select the exact path or exact target WWPN to use to reach a LUN. In the FC LUN Service, it is by design that all available paths to the selected LUN may be used. Use the FC Service or the FC MPIO Service if you want to select the exact path and / or exact target WWPN.

NOTE

When using the MPIO Two Pairs and MPIO Four Pairs mode, which are available for the FC LUN Service Test Bed, the FC Workload's Load Properties values you specified are divided by two and four for each Pair, respectively. For example, if you specify 100 concurrent workers in the FC Workload, and apply it on an FC LUN Service Test Bed with the MPIO Four Pairs mode set, 25 concurrent workers are assigned to each pair.

SB FC-NVMe Service

Use the SB FC-NVMe Service to automatically configure and use all available paths from a SANBlaze FC-NVMe Host to one or more FC-NVMe Namespaces.



Add an FC-NVMe service, connect a link from the SB FC-NVMe Port to the FC-NVMe Service. Click on the *Namespaces: 0 of n* link to open the drop-down menu with Namespaces that are accessible to the selected SB FC-NVMe Port.

NOTE

All physical SB FC ports set to NVMe mode are represented under one logical SB FC-NVMe port in Test Bed. So the one logical SB FC-NVMe port may contain one or more physical SB FC ports that are set to NVMe mode, even though they are not separately identified in the test bed.

Click the *Namespaces* drop-down menu to see a list of discovered namespace GUIDs, that is, a collection of namespaces with the same GUID, the number of paths available to the GUID, and the reported ID associated with the namespace. Select a Namespace GUID to use in the test bed.

FC-NVMe Service 0: 1 namespace selected (of 3)

		GUID (Filter)	Paths	ID	Size (> limit GB)
1	☐	600110d005b6db0106060001ff77e309	4	1	1 GB
2	☐	600110d005b6db00060600002cf21f0a	4	1	256 MB
3	☑	600110d005b6db00060600002cf21f0b	4	2	512 MB

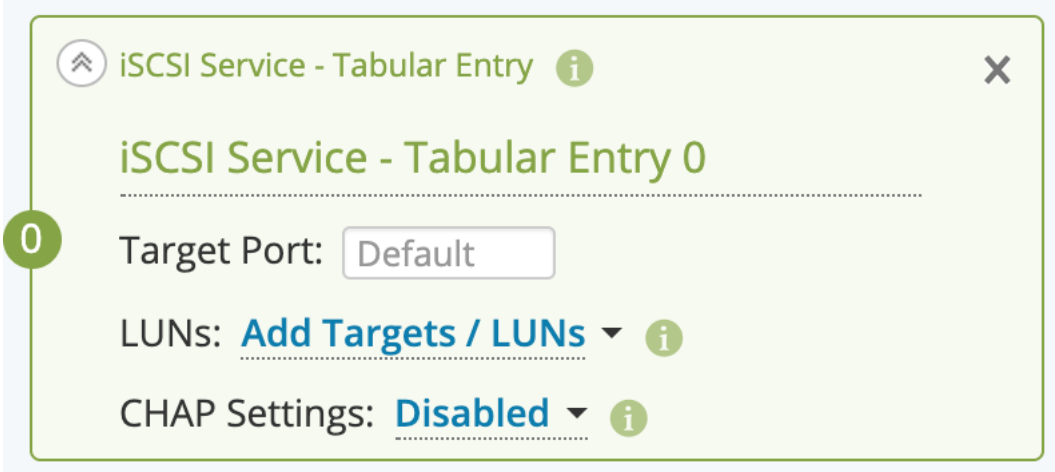
Optionally enable MPIO to enable MPIO failover and load balancing functionalities. The specific MPIO policy is specified when configuring the FC-NVMe Workload Test.

iSCSI Service-Tabular Entry

You can create an iSCSI service automatically using iSCSI Auto-Discovery, or you can add initiators and targets using iSCSI Service-Tabular Entry either individually or by importing a properly formatted CSV file.

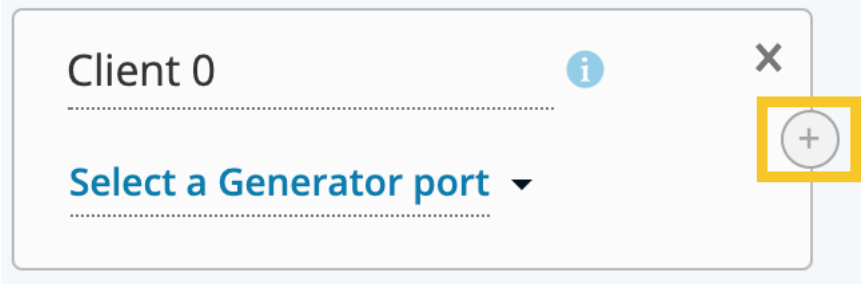
Using iSCSI Service-Tabular Entry by Auto-Discovery

1. Select Create Test Bed from the quick access + menu at the top of the screen.
2. Enter the following information on the *New Test Bed* screen:
   - *Test bed name*, *Description*, and *Tags* for your test bed.
   - *Group Name* for your test bed.
3. For Client 0, click **Select a Generator Port** and choose a port.
4. Optional: If you need additional clients, click the + drop-down menu and click **Client**, then select a generator port for the client.
5. Click the + drop-down menu and select **iSCSI Service-Tabular Entry**.

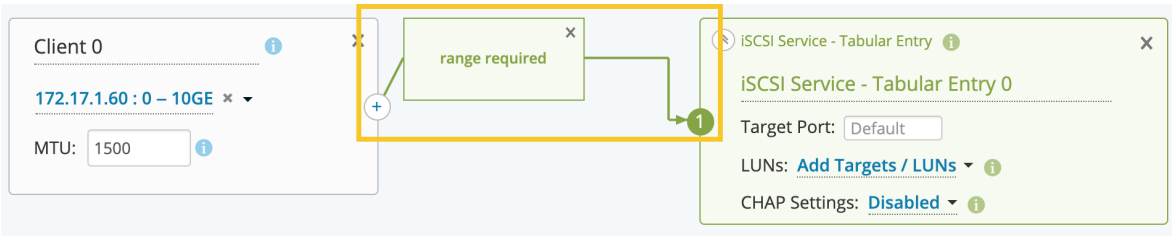


You can add one or multiple Services from the drop-down menu.

6. Click the '+' button on the right edge of each *Client* box and select the service or services to link to.



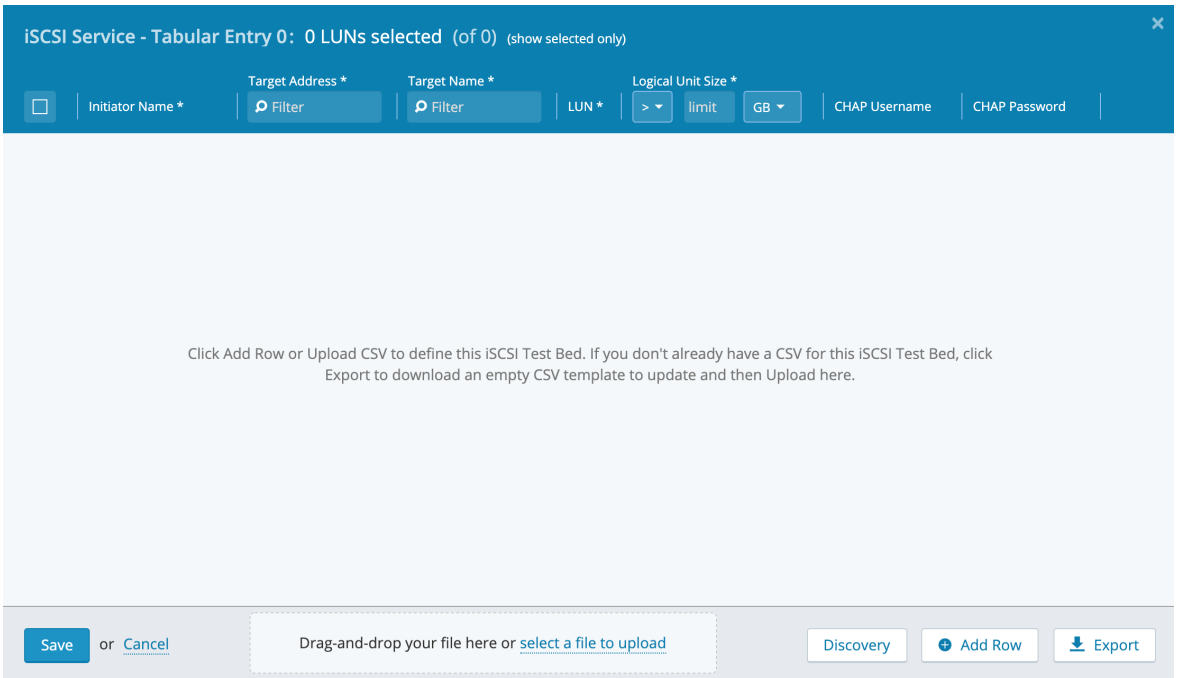
7. An arrow between the two boxes appears. You can create a single or multiple links to one or more services.



8. Optional: Set the *Target Port* in the *iSCSI-Service-Tabular Entry* box, if necessary.

> **NOTE**
> The default port is the default iSCSI port, 3260. Thesetting is applied to all targets configured in the service. Change the default port setting if your lab configuration requires it.

9. In the *iSCSI-Service-Tabular Entry* box, click **Add Targets/LUNs** and then click **Discovery** on the LUN selection page.



10. In the iSCSI Discovery window, enter the following information that is required to initiate the discovery process:
    a. Portal Info
        i. Address: IP address of a destination iSCSI Target Portal that can return information about the iSCSI Targets that are accessible to the iSCSI Initiator that is doing the discovery using standard iSCSI requests and responses. The iSCSI Target Portal may or may not be an accessible iSCSI Target itself.

ii. Port: TCP port that the iSCSI Target Portal is listening on for iSCSI requests and responses.
iii. Timeout: The maximum amount of time the iSCSI Discovery process will last.
iv. Authentication: Enable the use of one or more supported AuthMethods defined by the iSCSI RFC such as CHAP.

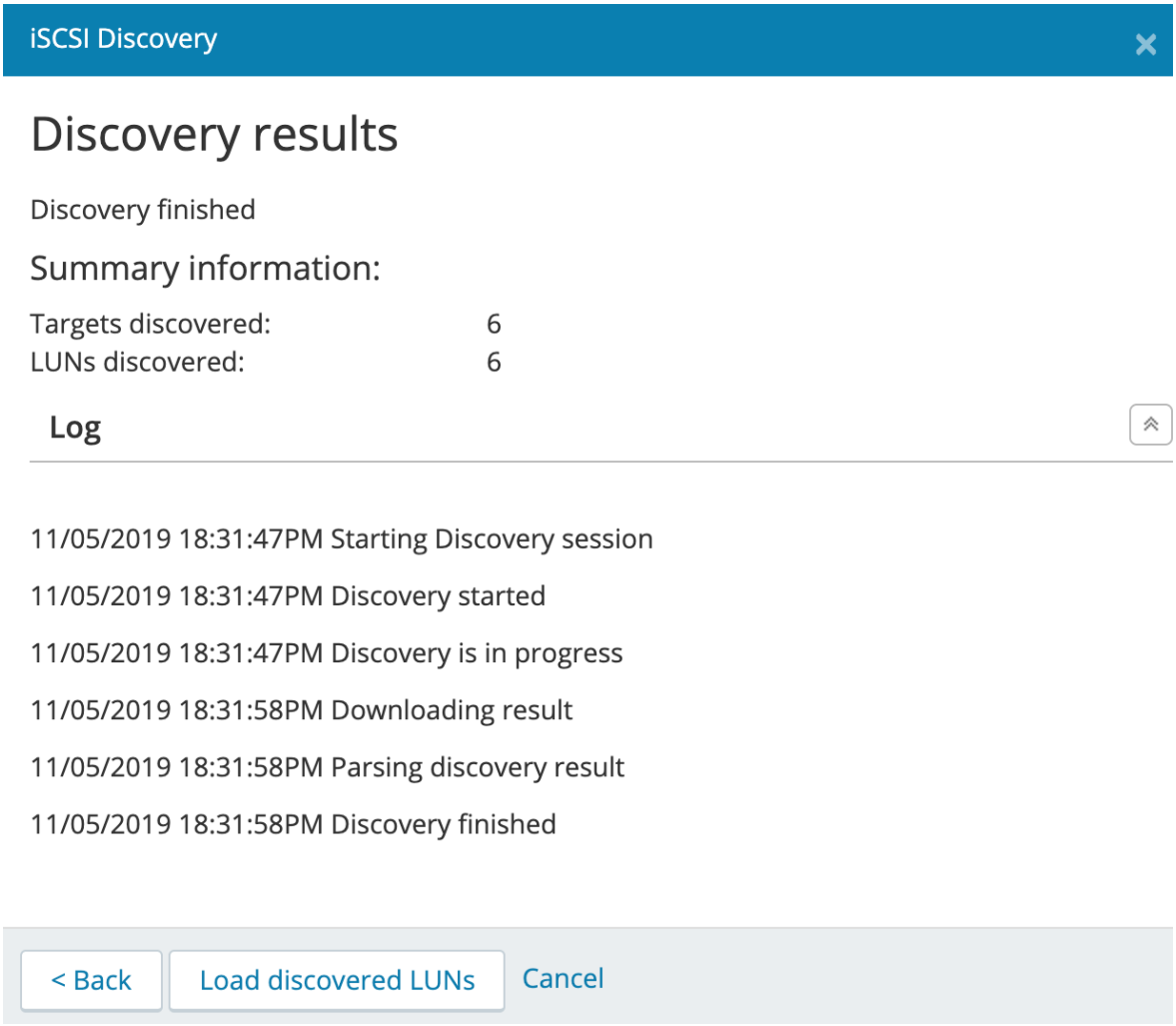
b. Initiator Info
   i. Name: Enter an iSCSI Initiator Name (loosely referred to as the IQN) that you want to use to perform the iSCSI Discovery. You should use a Name that you intend to use for the iSCSI Workload Test itself, not only for the Discovery, as the accessible Targets may vary based on the iSCSI Initiator Name depending on your lab configuration.

c. Network Info
   i. Enter the common IP address parameters (address, netmask, gateway) for the iSCSI Initiator that is doing the discovery. If you already have these configured in the Test Bed Link, simply click *Load from link* to apply those settings automatically.

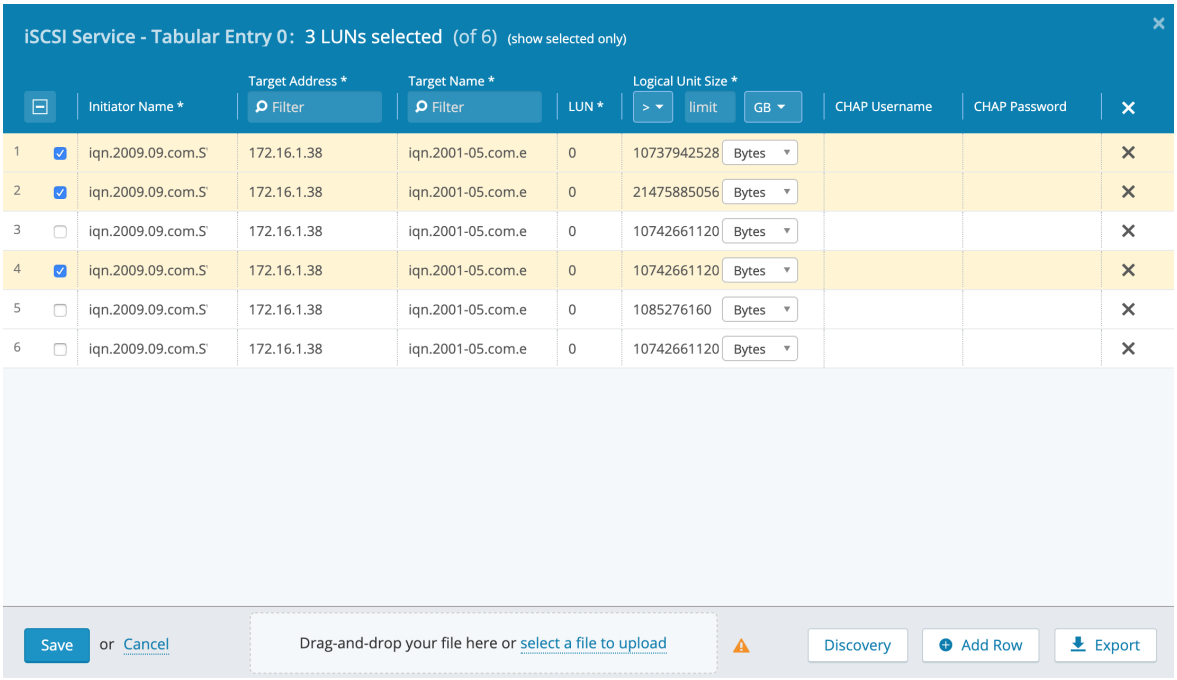
d. Enable Trace
   i. Optionally download a capture of the iSCSI Discovery process to be viewed using PCAP readers such as WireShark.

e. Previous Results
   i. If you have performed iSCSI Discovery before, you can click on the entry to view a summary of the most recent iSCSI Discovery results, and optionally apply these Targets and LUNs by clicking the **Load discovered LUNs** button.

11. Click **Start** to initiate the iSCSI Discovery process. The *Discovery Results* page loads.



12. When the iSCSI Discovery completes, a numerical summary of the discovered Targets and LUNs is shown.
    If you selected Enable Trace, you can now download the .pcap capture by clicking *Download PCAP*.
    A log is also available that tracks the timestamps of various key stages of the iSCSI Discovery process.
    If iSCSI errors are encountered during the discovery process, these errors are shown in the Errors section.
13. Click **Previous Results** and then **Load discovered LUNs** to apply the previously discovered Targets and LUNs that are accessible to the iSCSI Initiator used to perform the previous discovery.



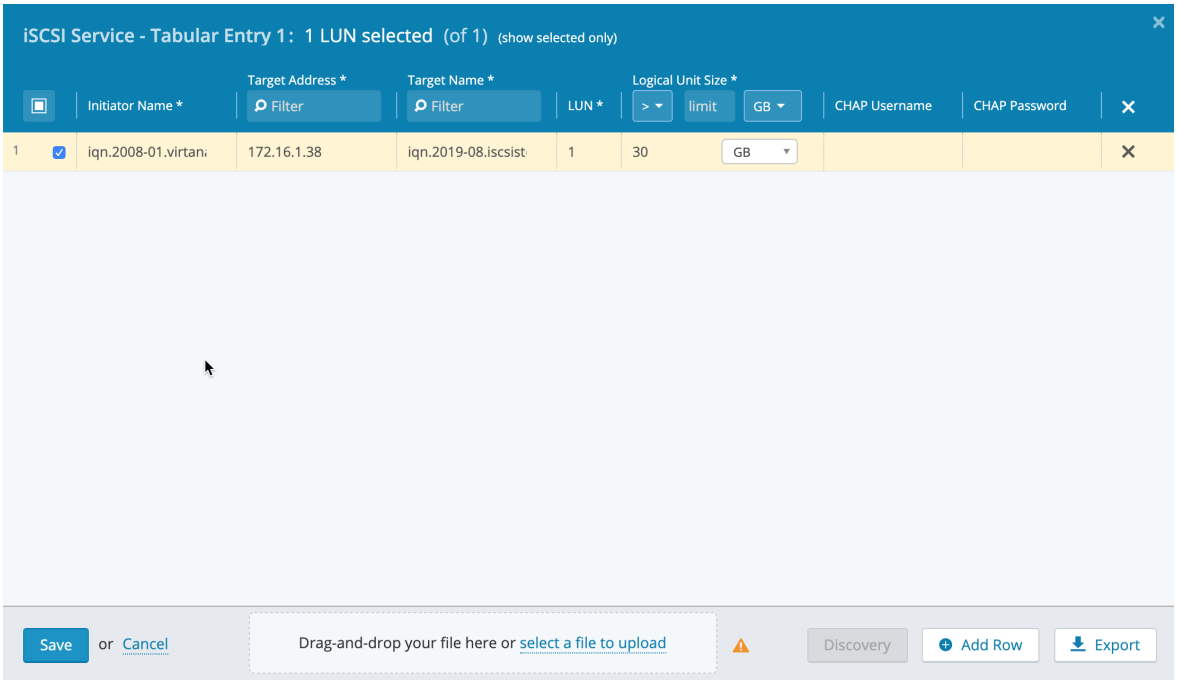
The discovered information is applied to the iSCSI Service-Tabular Entry table. By default all discovered Targets and LUNs are selected. You can select or deselect one or more Targets to specify which Targets and LUNs are used in the Test Bed. If an entry is not selected, then that Target/LUN is not used in the Test Bed.

You can also customize the discovered information or export the table to a CSV file for additional modification, and then import it back into WorkloadWisdom.

14. Click **Save** to save the Targets/LUNs selections and settings.
15. Perform any other optional task needed, such as enabling/disabling CHAP, or adding more Client Ports or Services or Links.
16. When ready to save the Test Bed, click **Create Test Bed**.

Using iSCSI Service-Tabular Entry by Adding Individual Entries in Tabular Format

1. Follow the steps described in the section iSCSI Service-Tabular Entry [37] until you reach the *Discovery* step.
2. Click **+ Add Row** to add one entry at a time to the test bed. You can add multiple rows by clicking **+Add Row** for as many entries as you need. Click the **X** at the end of each row to delete the corresponding entry.



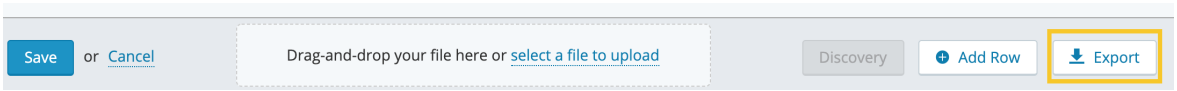
When you add an entry, the entry is populated with its *Initiator Name* and *Target Name*. By default, when an entry is added, the checkbox in the first column is checked. The checkbox indicates that it is selected as part of the test bed.

3. For each entry, complete the information associated by the other columns:
   a. *Target Address*. IP address or hostname of the location of the iSCSI target on the network.
   b. *LUN*. LUN being tested
   c. *Logical Unit Size*. Size of the iSCSI target. You can specify this in GB, bytes, KB, or MB. GB is the default.
   d. *CHAP Username*. Username for iSCSI access. This is an optional field that is disabled by default.
   e. *CHAP Password*. Password for iSCSI access. This is an optional field that is disabled by default.
4. Use the checkbox in the 

column to select or deselect the entries to include in the Test Bed
5. Use the search boxes associated with *Target Address*, *Target Name*, and *Logical Unit Size* columns to filter entries to include in the test.
6. Click **Save** to save the Targets/LUNs selections and settings.
7. Perform any other optional task needed, such as enabling/disabling CHAP, or adding more Client Ports or Services or Links.
8. When ready to save the Test Bed, click **Create Test Bed**.

Using iSCSI Service-Tabular Entry by Uploading a File

1. Follow the steps described in the section iSCSI Service-Tabular Entry [37] until you reach the *Discovery* step.
2. Click **Export** to download a WorkloadWisdom-formatted CSV file.



3. Click select a file to upload or drag and drop a CSV file into the box to upload a CSV file.
4. Use the checkbox in the 

column to select or deselect the entries to include in the Test Bed
5. Use the search boxes associated with *Target Address*, *Target Name*, and *Logical Unit Size* columns to filter entries to include in the test.
6. Click **Save** to save the Targets/LUNs selections and settings.
7. Perform any other optional task needed, such as enabling/disabling CHAP, or adding more Client Ports or Services or Links.
8. When ready to save the Test Bed, click **Create Test Bed**.

File and Object Storage Services

Configure Test Beds that provide File storage services (NFS, SMB) and Object storage services (Amazon S3, OpenStack Swift).

Common File and Object Storage Parameters

These settings apply to all File and Object Storage services.

- *Service Address:* IPv4 address or hostname of the server and the TCP Port associated with the service. If you enter a hostname, you need to enter DNS information in the Link settings.
- *Port:* TCP Port associated with the service. Unless you have specific requirements (for example, Port 8080), you can leave the *Port* field blank, and WorkloadWisdom automatically selects the standard-defined TCP Port.

NFS and SMB Service Parameters

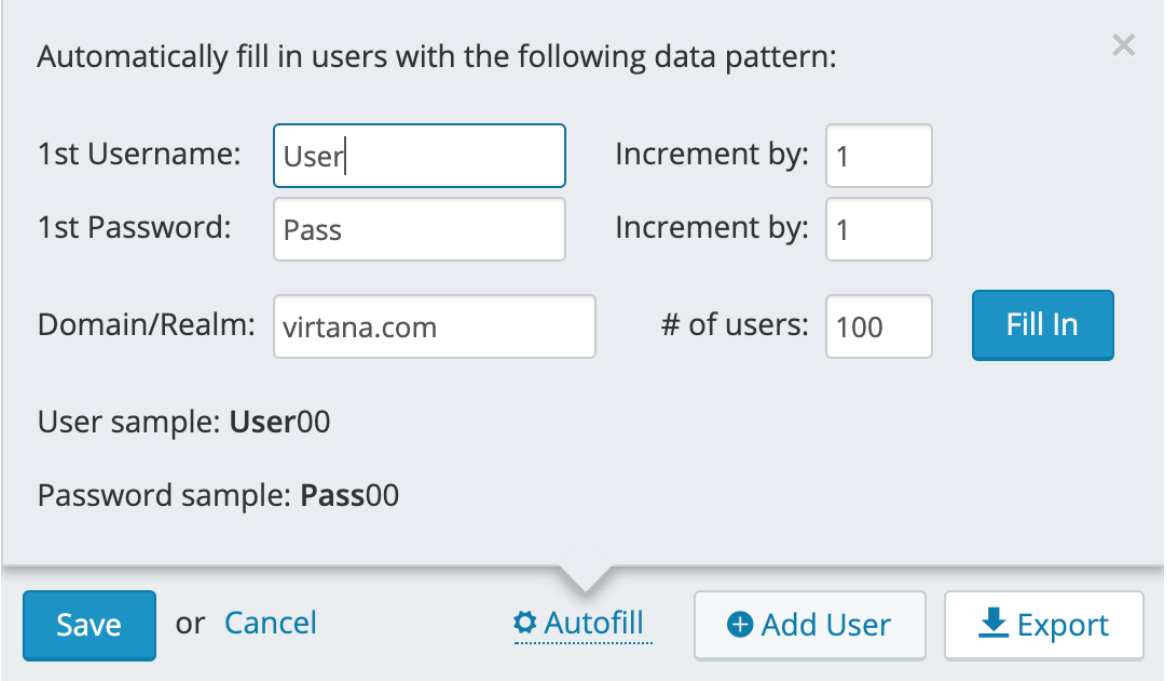
- *Shares:* Path and name of the shares or mount points on the server where the workload will create files and folders. You do not need to include a slash at the end.
  - Syntax example for NFS: `/rpool/fs13`
  - Syntax example for SMB: `vol_cifs`
  - To specify a large number of Shares, you can either use the Autofill function to create linearly incrementing Share names, or you can export a blank .csv template and modify it outside of WorkloadWisdom and then import it back in.

- A maximum number of 2,000 Shares is allowed per Service. You can create multiple Services per Test Bed, up to a maximum of 10,000 per Test Bed.

- *Kerberos:* Enables the use of a Kerberos service for user authentication. Available for NFS and SMB Services.
  - NFS
    - *Mode:* Select one of the following three standard Kerberos authentication options:
      - *krb5:* Uses Kerberos v5 to authenticate users
      - *krb5i:* Uses Kerberos v5 with integrity checksum
      - *krb5p:* uses Kerberos v5 with integrity checksum and encryption
    - *Kerberos Address:* Address of the Kerberos service, and optionally a non-standard port. If not specified, the realm specified in the NFS User menu is used as the Kerberos address.
  - SMB
    - *Kerberos Address*: Address of the Kerberos service, and optionally a non-standard port. If not specified, the domain name specified in the SMB User menu is used as the Kerberos address.
- *Users:* List of usernames, passwords, and optionally domains or realms.

User	Password	Domain / Realm	✕
User00	Pass00	virtana.com	✕
User01	Pass01	virtana.com	✕
User02	Pass02	virtana.com	✕
User03	Pass03	virtana.com	✕
User04	Pass04	virtana.com	✕
User05	Pass05	virtana.com	✕
User06	Pass06	virtana.com	✕
User07	Pass07	virtana.com	✕

  - To specify a large number of Users, either use the Autofill function to create linearly incrementing User names, Passwords, and Domains or Realms, or you can export a blank .csv template and modify it outside of WorkloadWisdom and then import it back in.

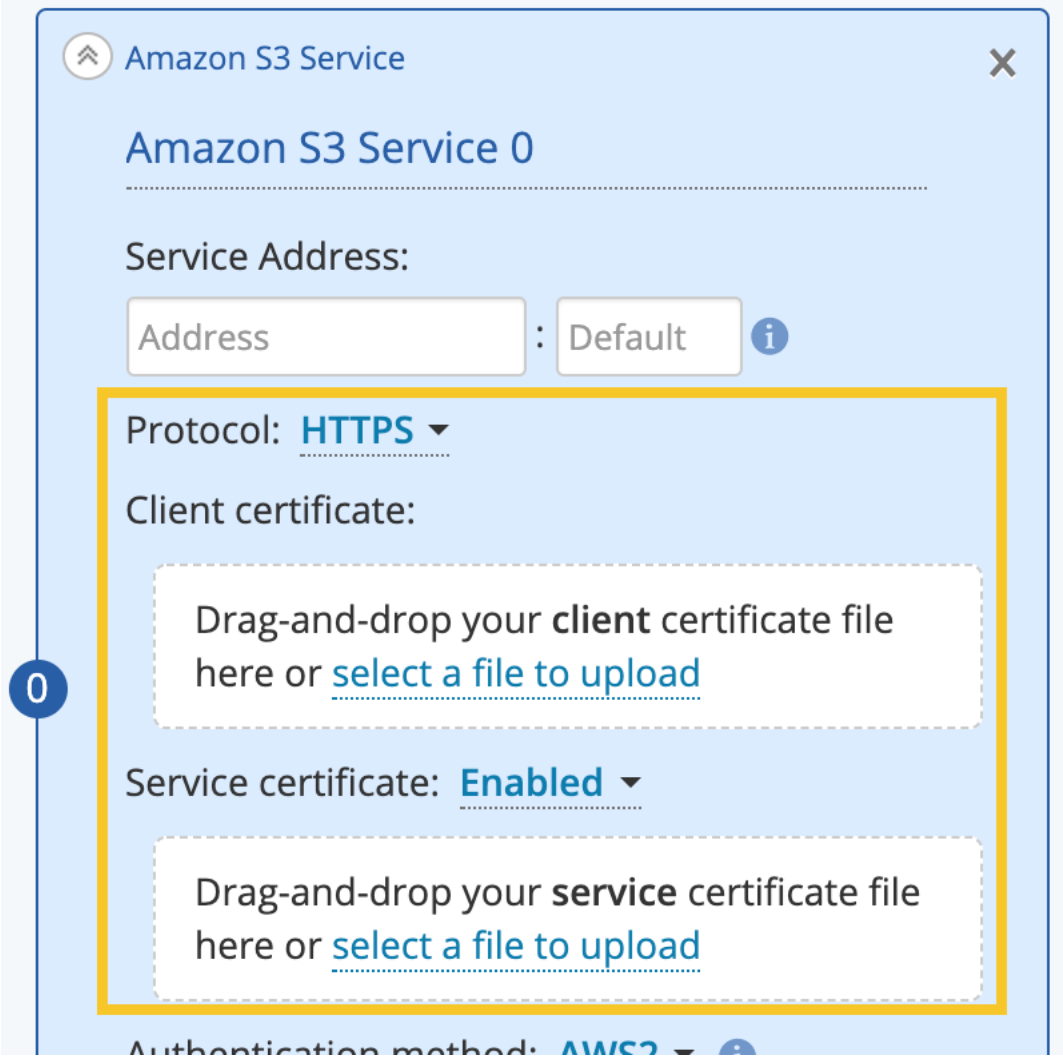


- For NFS, the Users menu is only available when you enable Kerberos.

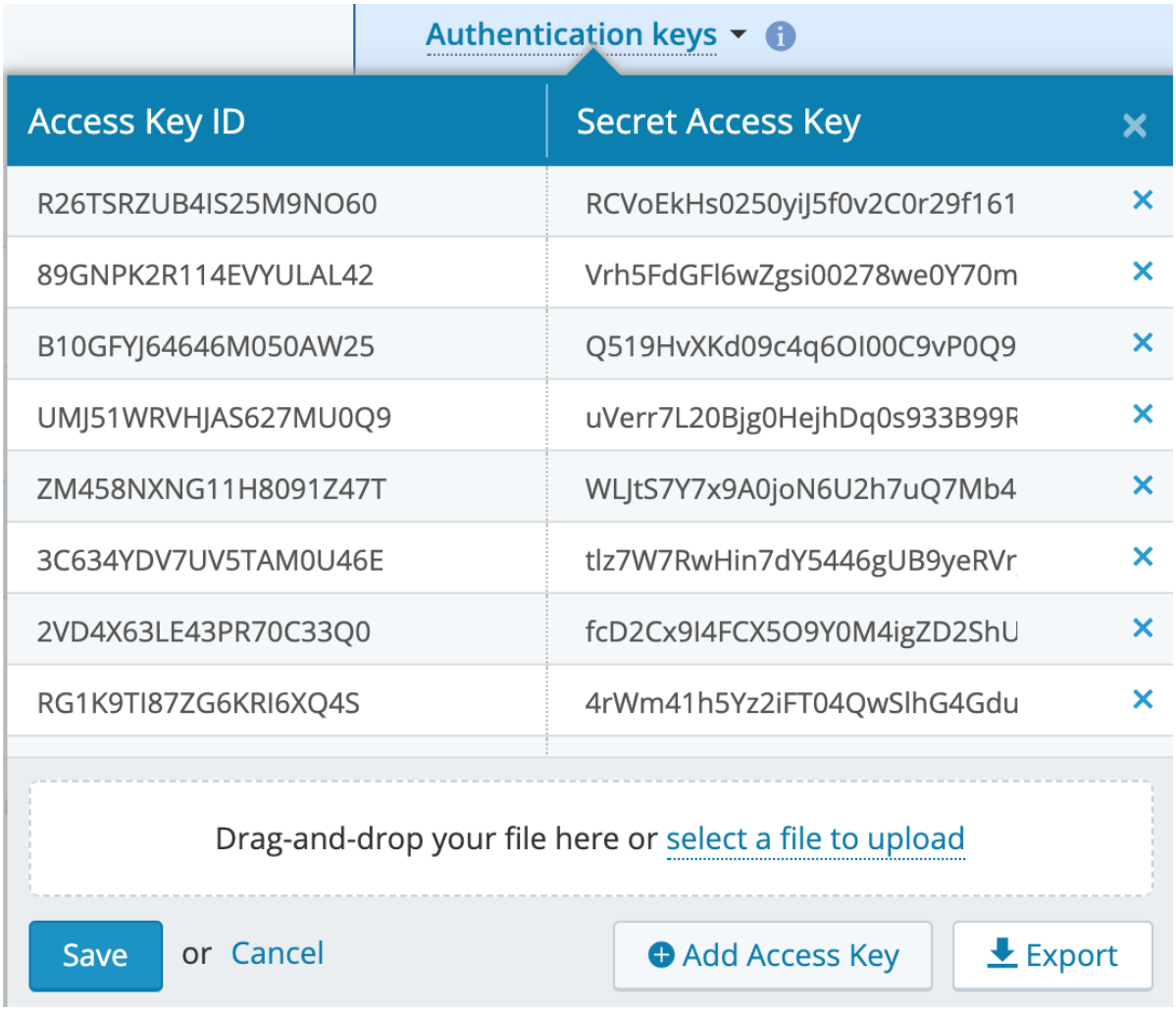
Object Service Parameters

Amazon S3

- *Protocol: HTTP* or *HTTPS*
  - *HTTPS*: Optionally configure Client and Server certificates. If *Enabled*, upload a valid Server certificate.
- *Authentication method: Disabled*, *AWS2*, or *AWS4*
  - It is rare that no authentication method is used, except in a lab environment. AWS4 is the common default authentication method for Amazon S3.



- *AWS Region:* AWS Region, selected from the list.
- *Authentication Keys:* If you use selected AWS2 or AWS4 for the *Authentication Method*, you need to enter the *Access Key ID* and the *Secret Access Key*. To specify a large number of S3 clients, you can export a blank .csv template and modify it outside of WorkloadWisdom and then import it back in.



- *Add request headers:* Many request header key value pairs exist, and different applications tend to include different key value pairs in the request header. Optionally use this field to define additional HTTP request headers to include in every HTTP request sent by WorkloadWisdom HTTP clients. Only use this field if you are familiar with the key value pairs required.

> **CAUTION**
> The request headers you define here will be included in all requests.

Openstack Swift

- *Protocol: HTTP or HTTPS*

- *Authentication method: TempAuth* or *Keystone* (OpenStack).
- *Auth Address:* Available only if you select *Keystone*. Enter the IPv4 address (or hostname) and port of the keystone authentication server.
- *Base Request URI:* Base URI used for OpenStack-Swift authentication. If you are not sure, you might be able to leave it blank and use the default of */auth/v1.0*, which is automatically used if the field is blank.
- *Authentication:* If you selected *TempAuth*, enter the X-Auth-User value and X-Auth-Key value. If you selected *Keystone*, enter the username, password, and tenant.
- *Add request headers:* Many request header key value pairs exist, and different applications tend to include different key value pairs in the request header. Optionally use this field to define additional HTTP request headers to include in every HTTP request sent by WorkloadWisdom HTTP clients. Only use this field if you are familiar with the key value pairs required.

> **CAUTION**
> The request headers you define here will be included in all requests.

Less Common or Obsolete Services: Additional Information

> **CAUTION**
> The Obsolete Services documented in this section are still supported. However, there will not be new functionalities developed for these Services. In a future release, one or more of these Services will either be rolled in to other Services and / or be removed.

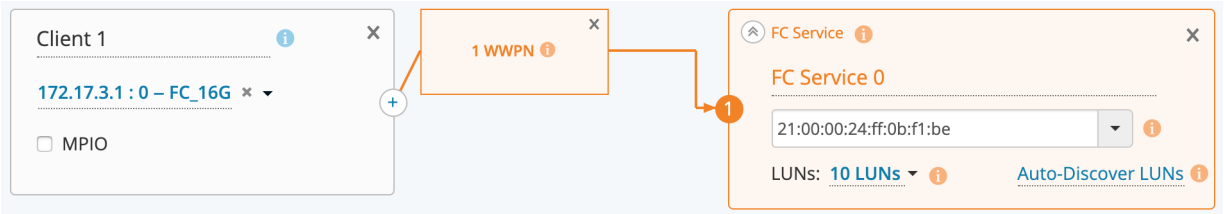
FC Service

If you want to create an FC test bed and precisely define the ITL connection, it is recommended that you use the *FC Service* or *FC MPIO Service*.

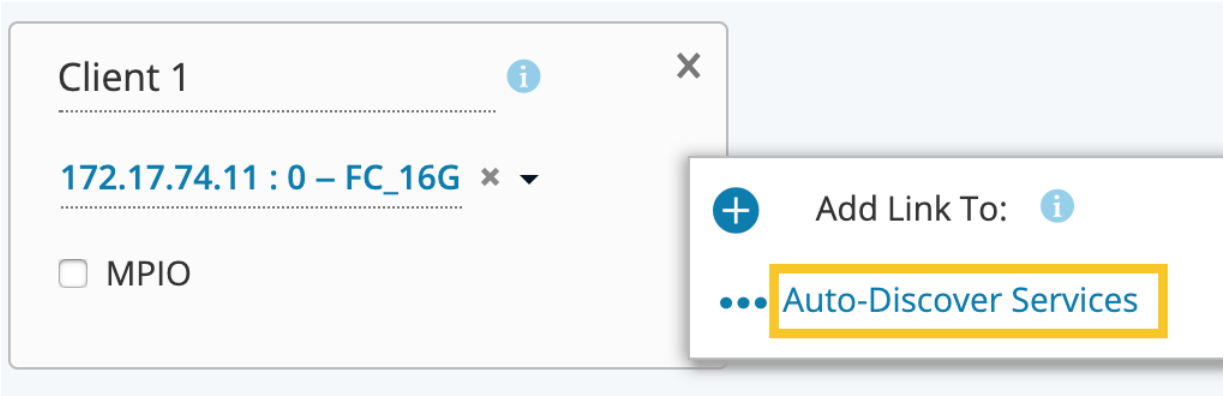
If you want to create an FC test bed where all available paths to a set of LUNs (grouped by WWID) are used and automatically configured, it is recommended that you use the *FC LUN Service*.

> **CAUTION**
> When using Auto-Discover Services, a list of discovered targets is not generated on-demand. The list is generated when you click *Rediscover Targets* on the FC Workload Generator. To do this, go to *Generators*, select the FC Workload Generator, and click **Rediscover Targets**.

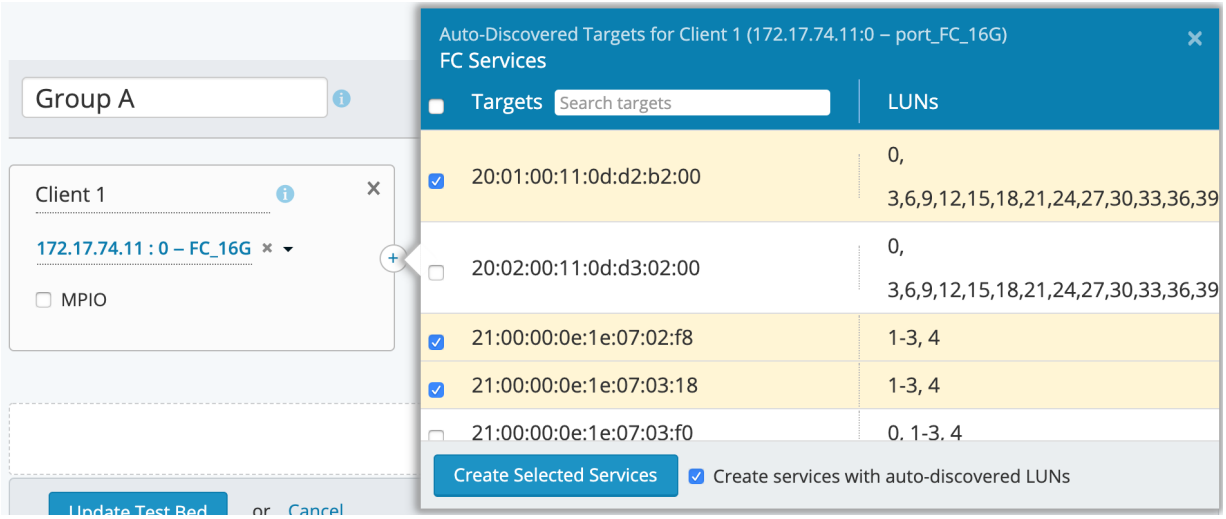
You can use the FC Service to define a one-to-one ITL connection between the Workload Generator's FC initiator, the SUT's FC target WWPN, and the SUT target's LUNs.



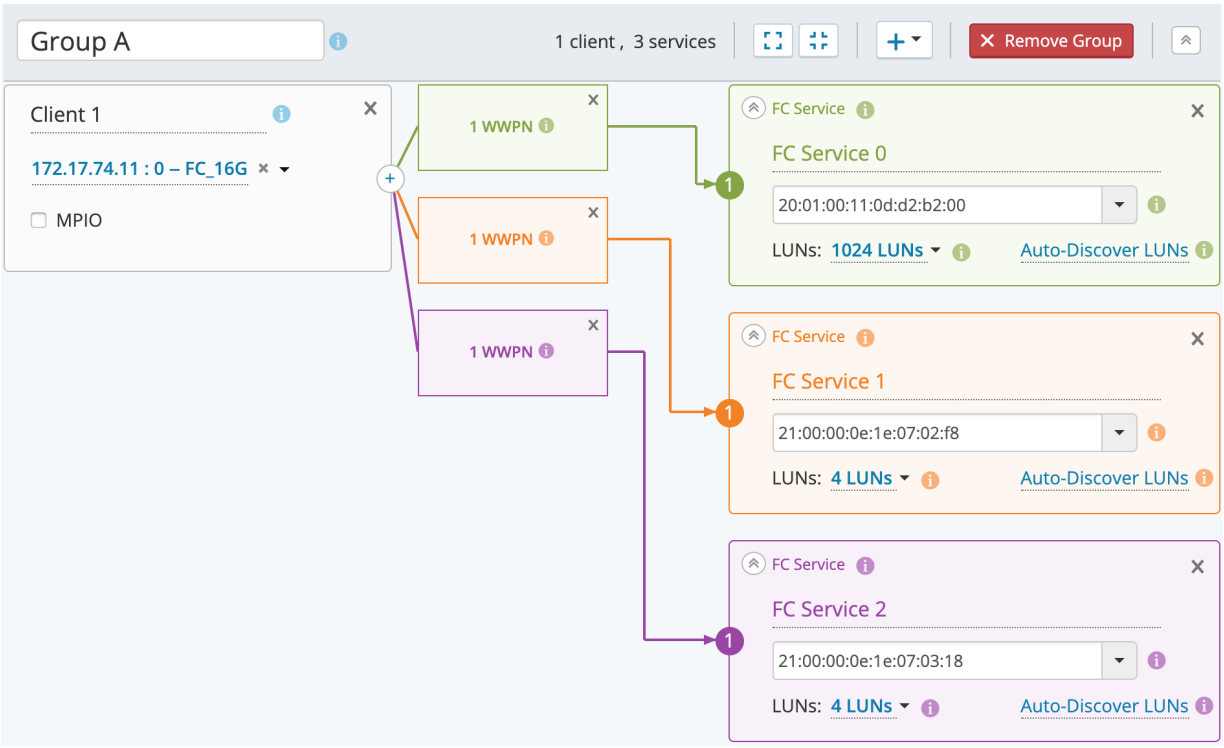
The number of available targets and LUNs can be high, and WWPN values are not easily readable or memorized. It is strongly recommended that you use *Auto-Discover Services* to automatically discover available targets.



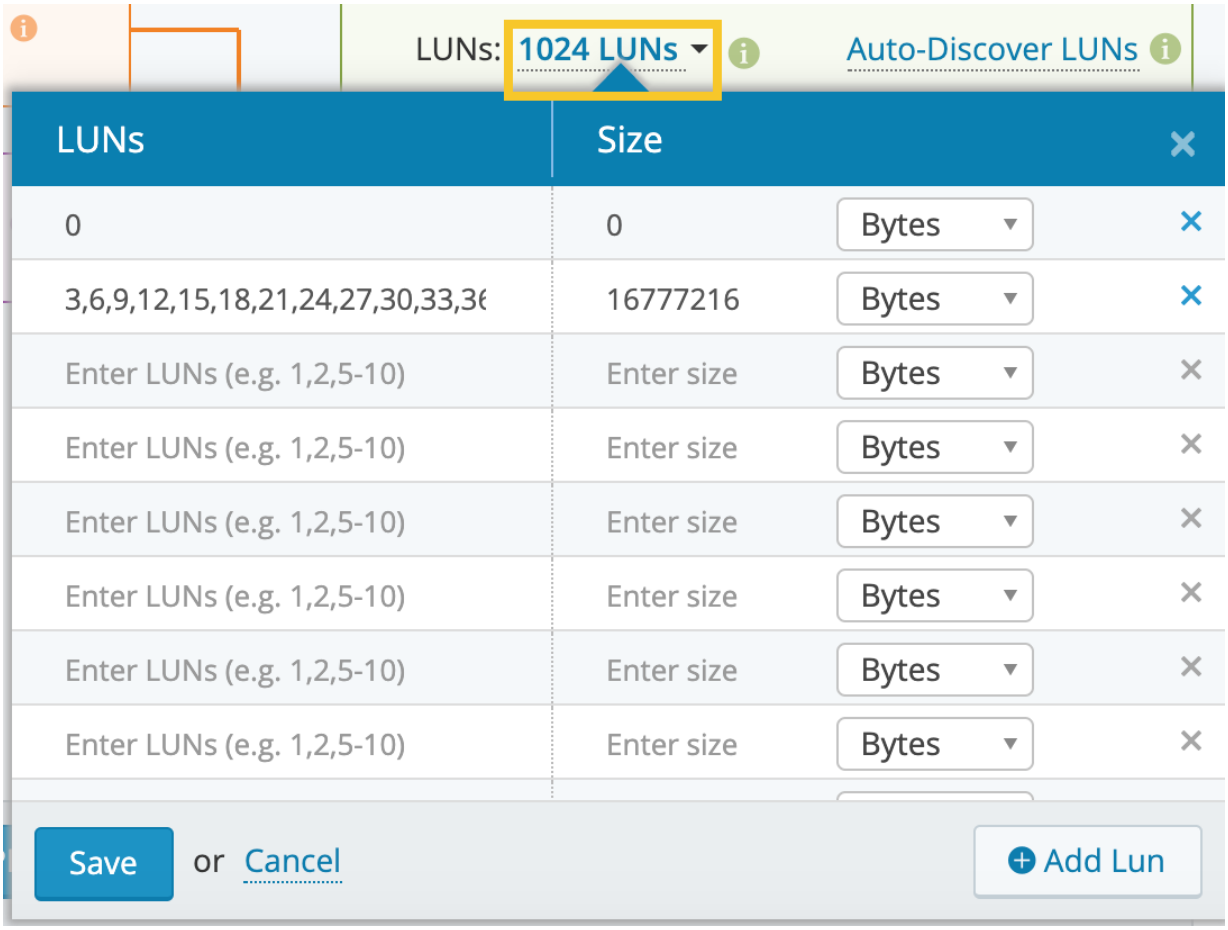
After you select *Auto-Discover Services*, a list of discovered targets displays. You can manually select a target WWPN from the list.



After you click **Create Selected Services**, a functional FC test bed is created.



You can continue to expand this test bed by adding more links to more targets, and/or modifying the LUNs list by selecting from the *LUNs* drop-down menu.



From the list of LUNs, you can remove LUNs that you do not want to generate IOs, when running a workload test. You need at least one LUN to have a functional FC test bed.

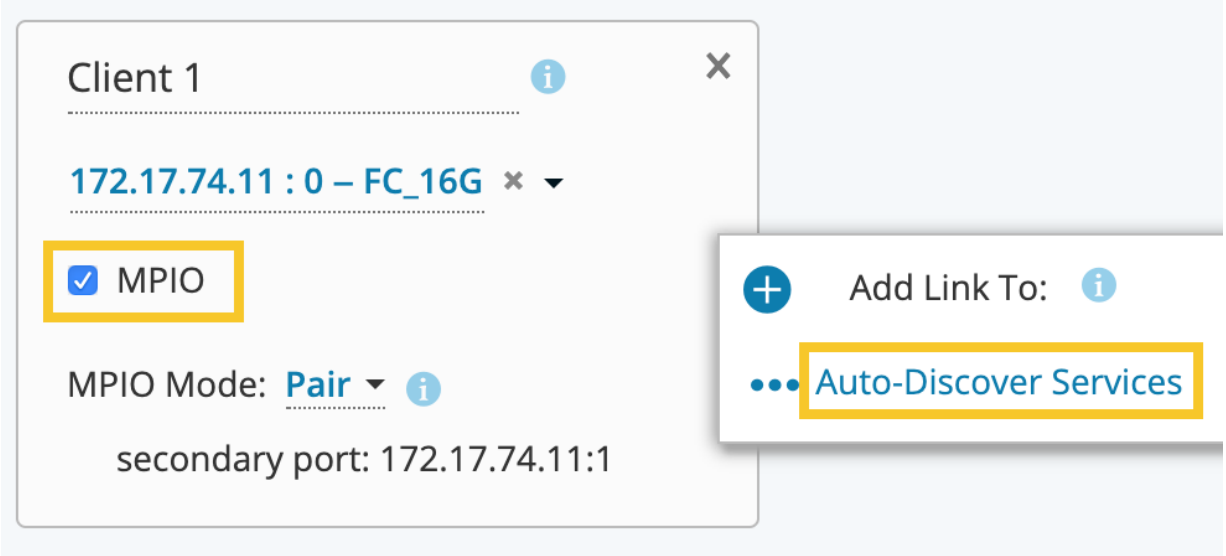
If you do not want to use auto-discover to create the FC Service, you can enter the Target WWPN and LUNs manually. Be sure that the manually configured WWPNs and LUNs are accessible by the Workload Generator's FC Initiator.

FC MPIO Service

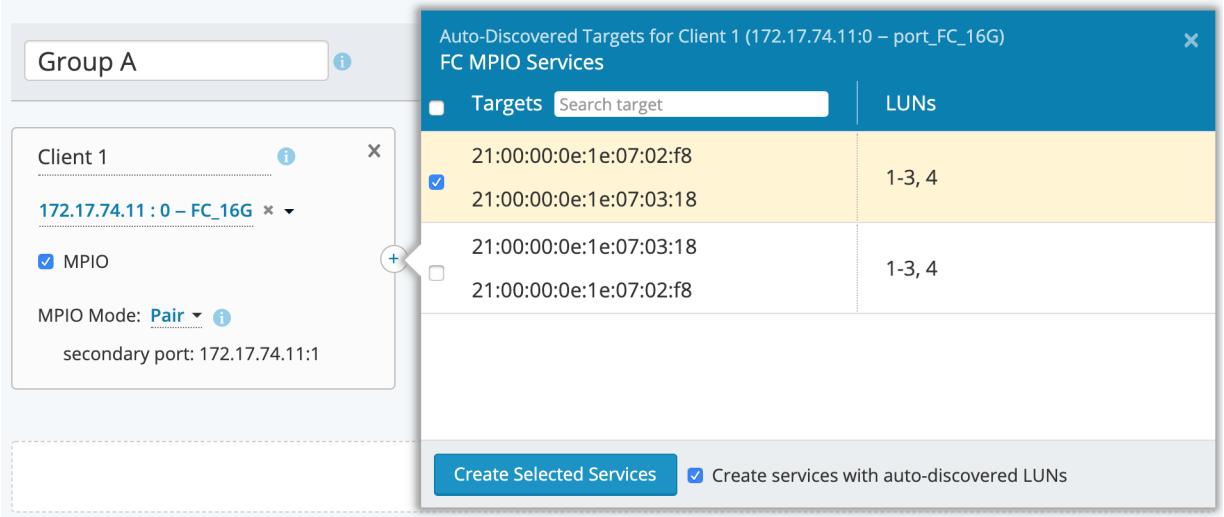
Use the FC MPIO Service to configure specific M:M multipath connections for WorkloadWisdom's MPIO-enabled initiators to MPIO-enabled targets with access to a common set of LUNs.

In the FC MPIO Service, there is a one-to-one relationship between WorkloadWisdom initiators to FC Targets. If you want to define four specific paths, that is, four unique Target WWPNs with access to the same LUNs, you must use four Workload Generator FC Ports (each Port is one Initiator) by enabling MPIO and selecting the *Quad* option under *MPIO Mode*.

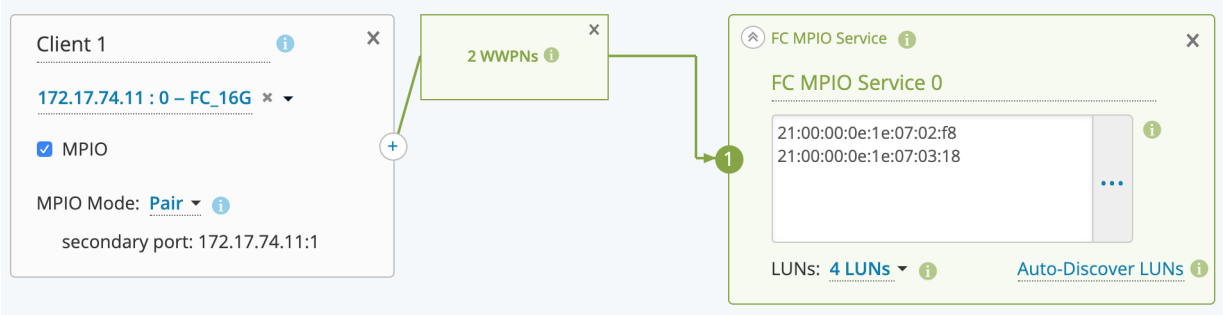
Similar to the FC Service, it is recommended that you use *Auto-Discover Services* to automatically retrieve a list of available MPIO-enabled targets and LUNs.



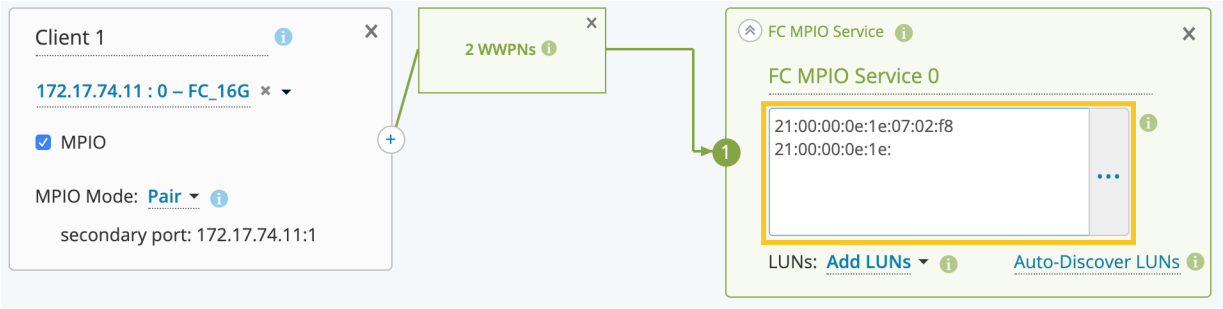
A list is displayed of discovered MPIO-enabled Targets, and the Target WWPNs/LUNs that are accessible to all selected WorkloadWisdom MPIO-enabled Initiators are grouped together.



Select one or more groups and click **Create Selected Services**. A functional M:M MPIO test bed is created.



Similar to the FC Service, you can opt out of creating the FC MPIO test bed using *Auto-Discover*, and manually define exactly how each initiator connects to each target WWPN.



iSCSI Service (Obsolete)

The iSCSI Service is a single Target service for iSCSI that was introduced before the **iSCSI Service – Tabular Entry** service, and offers limited functionality in comparison. It may be preferred if you are working with a very small iSCSI test environment with only one iSCSI Target. See the iSCSI Service-Tabular Entry [37] section for information on the available parameters.

Amazon S3 Service (Obsolete)

The **Amazon S3** Service found under this menu is made obsolete by the new **Amazon S3** Service described earlier. The functionalities provided by this obsolete Service is a subset of the new Service. See the Amazon S3 [45] section for information on the available parameters.

Links

The following Link parameters are available.

FC Services

- *NPIV Enabled.* Enables or disables the creation of NPIV interfaces. This option is not currently available for the FC MPIO Client. The time it takes to create and/or clear NPIVs is usually several seconds for each NPIV, and also depends on how many Targets and LUNs visible to each NPIV in the fabric. If you are applying hundreds of NPIVs, it can take over an hour. An NPIV progress bar is displayed when NPIV configuration is applied.
- *Range.* Specifies the starting and ending NPIV WWPN. The ending NPIV WWPN value must be numerically greater than the starting NPIV WWPN.
- *Reconnect.* The following two reconnect parameters are available:
  - *Reconnection Attempts.* Number of times the initiators attempt to reconnect to the FC Service before terminating the connection. When a connection is terminated, no further actions are performed for the terminated connection.
  - *Polling Interval.* Time interval between two consecutive Reconnection Attempts.

If NPIVs are enabled, after defining the WWPN range for the NPIVs, you must click *Create Test Bed & Create WWPNs* to apply the NPIV configuration. A real-time NPIV Configuration Progress pop-up dialog display the current status of NPIV configuration.

SB FC-NVMe Service

- *NPIV Enabled.* Enables or disables the creation of NPIV interfaces.
- *NPIVs per port***.** Specifies the total number of NPIV interfaces to create per physical port.

IP Storage Services

The following Link parameters are available to all Services that run over IP.

- *IP Range.* Starting IP address and ending IP address for the WorkloadWisdom clients. The ending IP address value must be numerically greater than the starting IP address.
- *IP Configuration.* (both IPv4 & IPv6)
  - *Gateway Address.* IP address the protocol client uses to reach the destination when the route to the protocol Service IP address is unknown.
  - *Allowed Port Numbers.* TCP source port range used by the protocol clients.
  - *Gratuitous ARP.* If enabled, the protocol client broadcasts a gratuitous ARP message that identifies the new binding of a source IP address and a source MAC address.
- DNS
  - *DNS Server IP.* IP address of the DNS Server to which clients send DNS queries to resolve the IP addresses for the hostname entered in the test bed service**.**
  - *DNS Resolution Timeout (seconds).* Amount of time the clients wait for a response to the query before concluding that the DNS query has timed out**.**

  - *DNS Query Retry Interval.* Amount of time the clients wait before retrying a query if it is set to issue a retry.
  - *Maximum Time to Live.* Amount of time a positive DNS response is cached by the client for a query**.**
  - *Negative Response Expiration Interval.* Amount of time a negative DNS response are cached by the client for a query**.**
- Advanced DNS settings
  - *DNS Cache Locality*. Specifies whether the information provided by the DNS server is applicable on a:
    - *Per Link*basis. All clients associated with the same test bed link can use the hostname – IP address information provided in the DNS server's response addressed to any client sharing the same test bed link. Clients on a different test bed link in which no client from that link has received a DNS server response needs to issue a query to resolve the IP address to a hostname, even if another client from another test bed link has received the response.
    - *Per Port* basis. All clients associated with the same Workload Generator port can use the hostname. IP address information provided in the DNS server's response addressed to any client sharing the same Workload Generator port, whether or not that response was received by a client from the same test bed link.
    - *Per Source* basis. Each unique client issues its own query to resolve the IP address to a hostname, even if other clients on the same test bed link or Workload Generator port have received responses to the same hostname IP address query.
  - *Resource Record Set Ordering.* Specifies how the Client uses the information provided by the DNS Server when multiple IP addresses are available for the same hostname**.**
  - *DNS Transport Protocol*. Uses either TCP or UDP to transmit DNS messages.
- VLAN
  - *Generate VLANs***.** Number of unique VLAN IDs available for use by the protocol clients.
  - *Starting from VLAN ID.* Lowest numerical VLAN ID available for use and the first VLAN ID that is used.
  - *Increment by*. Incremental value that is added to the previous value.
- MAC
  - *Generate MAC addresses.* Number of unique MAC addresses available for use by the protocol Clients.
  - *Start from address***.** Lowest numerical MAC address available for use and the first MAC address that is used.
  - *Increment by*. Incremental value that is added to the previous value.

The following Link parameters are only available to the obsolete iSCSI Service (Obsolete) [53], and do not apply to the iSCSI Service-Tabular Entry [37].

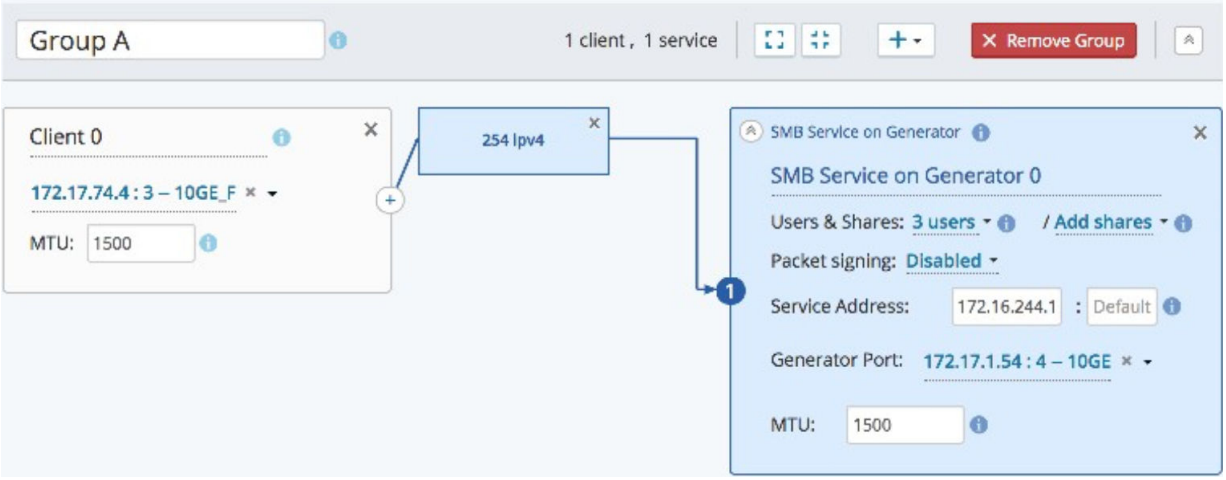
- *Initiator***.** Define the Initiator IQN.

Per IETF RFC3720, the format of the IQN should be `iqn.{yyyy-mm}.{reversed domain name}:{string}`. For example: `iqn.2015-09.com.loaddynamix:initiator.i001.port0.appl0` is a valid IQN that conforms to the specified structure

- Number of IQNs:
  - *One IQN.* Use one IQN for all Initiator IP addresses
  - *One IQN per IP.* Use one unique IQN for each Initiator IP address
  - *Specify number of IQNs.* Use the specified number of IQNs across the Initiator IP addresses
- *Custom Number of IQNs.* Specifies the number of IQNs when *Number of IQNs* is set to *Specify number of IQNs*.
- *IQNs name.* Read-only preview of the range of Initiator IQNs that can be used.

Simulated Storage Services on Test Ports

In addition to defining an SUT Service, the Workload Generator Test Port can also provide protocol server emulation, thereby providing a Workload Generator port service. This is useful in testing devices between the client and server. For example, a WAN optimization appliance, which optimizes the bandwidth usage between the protocol client and the protocol server. In the example below, a Workload Generator port service is used instead of an SUT Service, as the SUT is the device in between (not identified in the test bed).



The following types of Workload Generator port services are available:

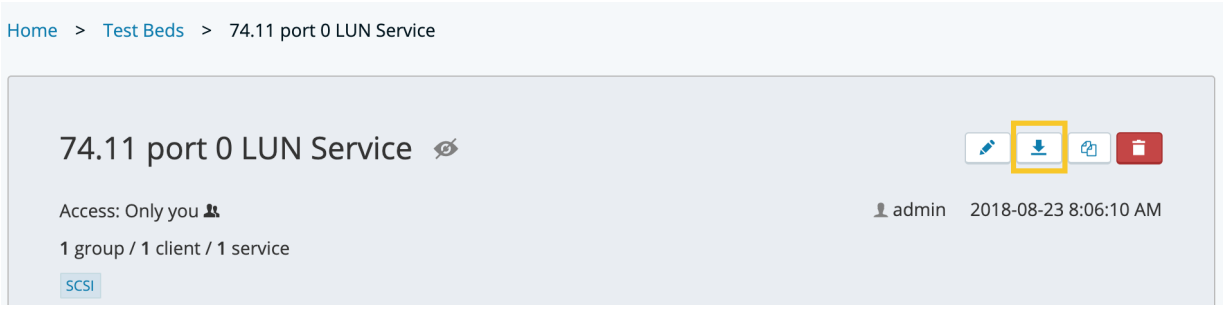
- *NFS Service on Workload Generator.* NFS Server emulation running on a Workload Generator port. One IP address with at least one valid share must be properly configured and accessible by the Workload Generator's NFS clients. In addition, one Workload Generator server port must be specified. A Workload Generator port cannot be used as both a client port and a server port in the same test bed.

- *SMB Service on Workload Generator.* SMB Server emulation running on a Workload Generator port. One IP address with at least one valid share and at least one valid username and password, and optional domain, must be properly configured and accessible by the Workload Generator's SMB clients. In addition, one Workload Generator server port must be specified. A Workload Generator port cannot be used as both a client port and a server port in the same test bed.
- *HTTP Service on Workload Generator.* HTTP Server emulation running on a Workload Generator port. One IP address must be properly configured and accessible by the Workload Generator's HTTP clients. In addition, one Workload Generator Server Port must be specified. A Workload Generator port cannot be used as both a client port and a server port in the same test bed.

Importing Test Beds

When you select *Import Test Bed*, you are prompted to select a test bed file (`.zip`) to be imported. Once the file is imported, you have the option to modify the contents of the test bed.

To export a Test Bed to be imported elsewhere, select a Test Bed either from the Test Bed list or from a selected Test Bed, then click the export button.



Preconditioning

With the introduction of Workload Data Importer, the common use case for block performance testing broadened. To better align the product workflow with the methodology of testing block storage arrays, a new *Preconditioning Workload* is introduced, separate from the workload test itself.

The previous product workflow is as follows. For backwards compatibility, this previous workflow is still available with WorkloadWisdom.

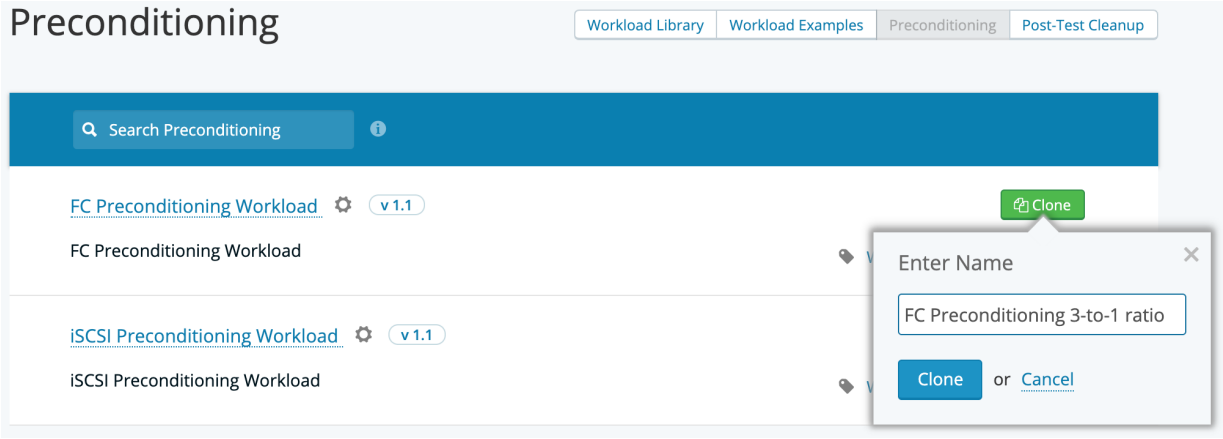
1. Open *FC/iSCSI Workload Test 1*.
2. Select *Run Pre-Test Only* or *Run Pre-Test before the Test* to precondition the system under test.
3. Start *FC/iSCSI Workload Test 1* (if not done in Step 2).
4. Open *FC/iSCSI Workload Test 2*.
5. Select *Do not run Pre-Test*.
6. Start *FC/iSCSI Workload Test 2*.
7. Open *FC/iSCSI Workload Test 3*.
8. Select *Do not run Pre-Test*.
9. Start *FC/iSCSI Workload Test 3*.
10. And so forth.

The new product workflow is:

1. Open the new *FC/iSCSI Preconditioning Workload* and start.
2. Open *FC/iSCSI Workload Test 1* and start.
3. Open *FC/iSCSI Workload Test 2* and start.
4. Open *FC/iSCSI Workload Test 3* and start.
5. And so forth.

To access the preconditioning workloads, select *Preconditioning* from the home page.

There are two preconditioning workloads; FC and iSCSI. Clone the desired preconditioning workload and name it accordingly.



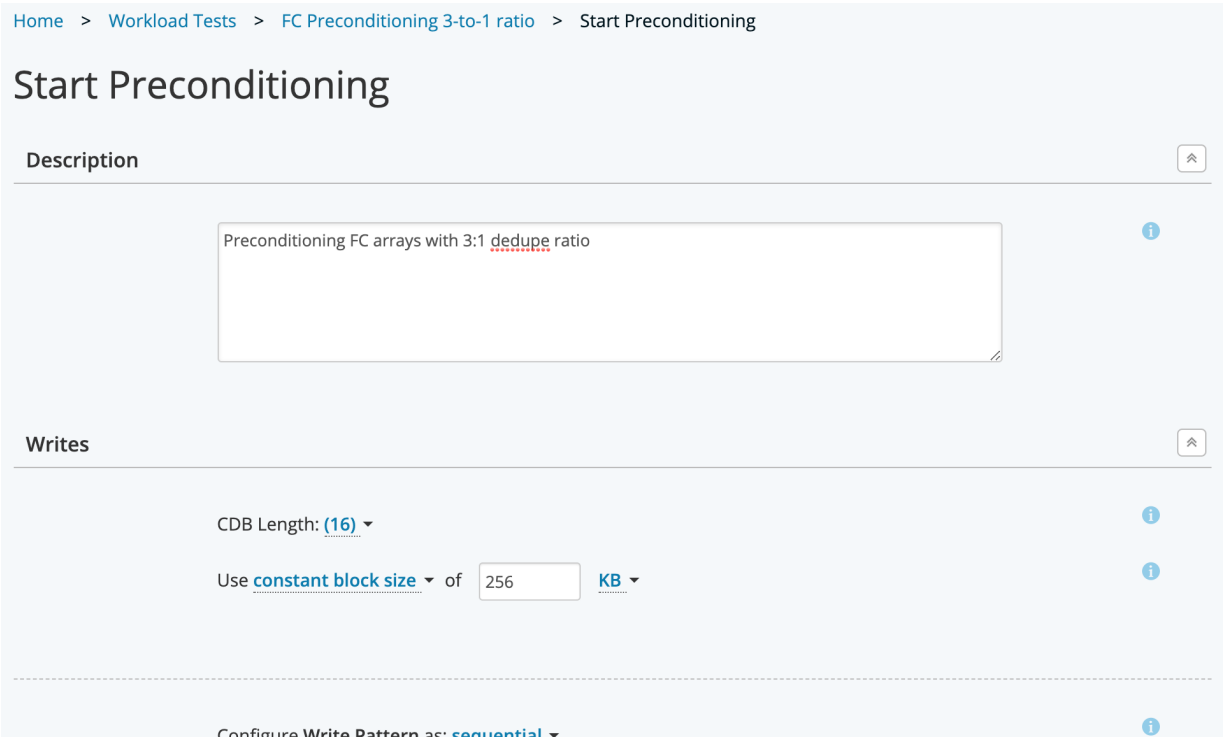
Before you can run the preconditioning workload, you need to apply it to a Test Bed that contains the storage array that you want to precondition. You can do that by clicking either the *Start* button or the *Edit* button.

- When using *Start*, your information is not saved as the default settings for the next time you open the same workload.

- When using *Edit*, your information is saved as the default settings, that are displayed the next time you open the same workload.

After the Test Bed information is saved into the default settings of the workload, the *Run Now* button is available as long as the Test Bed is currently available for use.

The preconditioning workload supports a subset of the Common Workload Concepts and Settings [93], and more specifically, a subset of the Block Protocol SCSI Workloads [150] settings. The major difference is that the user interface is simplified to show only what is needed to precondition an array rather than to create a production like workload.



The CDB Length (size of the SCSI header) is configurable. Typically, this setting should be 10 unless the LUN is larger than 2TB or large request sizes is used. For larger LUNs or request sizes, 16 is most the most common. You can use this for cases where the best

option is unknown. It adds very minor overhead and should significantly change the time needed to precondition.

You can specify request sizes like other workloads. The data can either be written sequentially or randomly to the I/O region specified. In either case, each logical block address of the LUN is only written one time.

Data parameters can be set like other block workloads except the choices are limited to random data content and data reduction options to avoid patterns that do not adequately precondition the array.

The workload supports *load properties*, *tracing parameters*, *runtime parameters*, and *start parameters*.

Post-Test Cleanup

You can use post-test cleanup to remove partial[2] or all content in a location specified by a test bed.

> **WARNING**
> Cleanup Workloads can erase all content in the location you specified, whether the content was created by WorkloadWisdom or not.

A pre-test sets up the testing environment and normally includes preparing the content in a file or object system so the workload can be run against it. These contents are created as part of the pre-test. See Pre-Test [102]

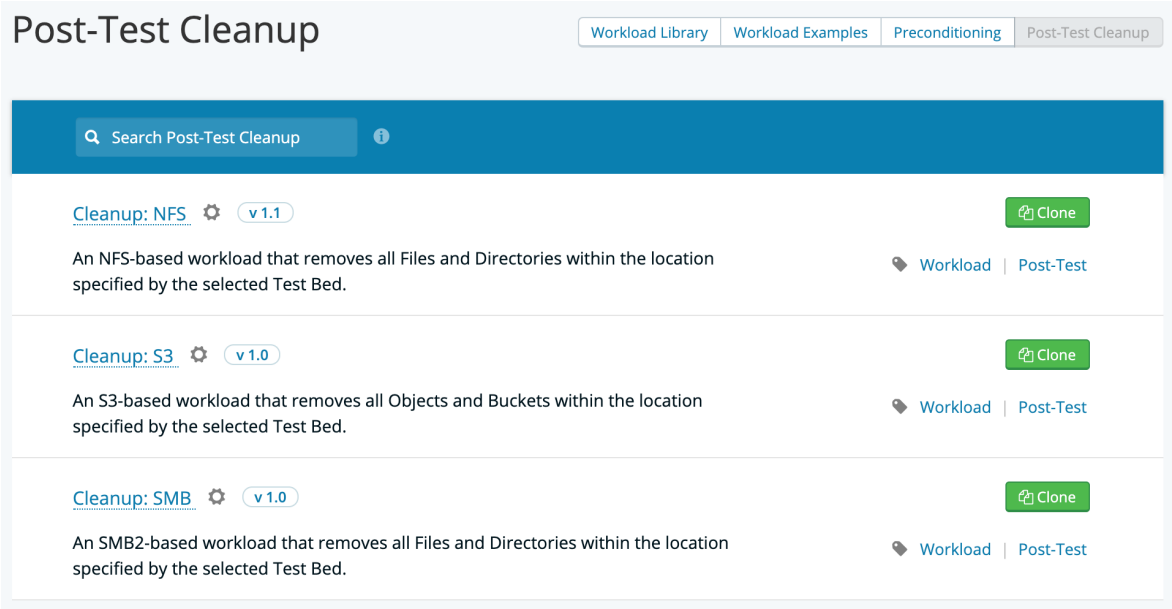
When the workload test finishes, content created as part of the pre-test remain. Sometimes keeping the content is a desired result, because it gives a capability to run tests again on the same test environment.

Post-test cleanup is a specific type of workload test that removes all files and directories in a location specified by a test bed, whether the files and directories are created by a pre-test, workload test, or other means. As with all workload tests, you can export this test from WorkloadWisdom, import this test to WorkloadWisdom, or reconfigure it.

[2]For Cleanup Workloads that support partial content filtering.

Using Post-Test Cleanup

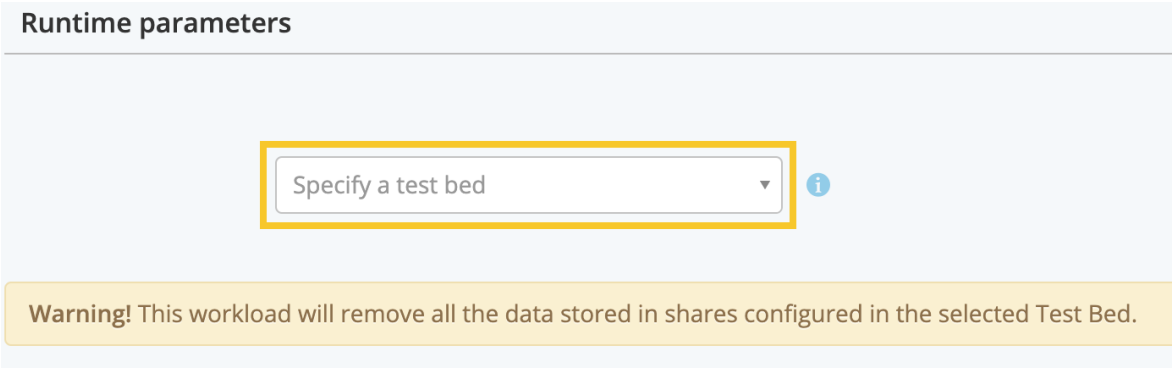
1. Click *Post-Test Cleanup* from the *Setup* section, located on the left vertical section of the *Home* page.
   The *Post-test Cleanup* screen displays.



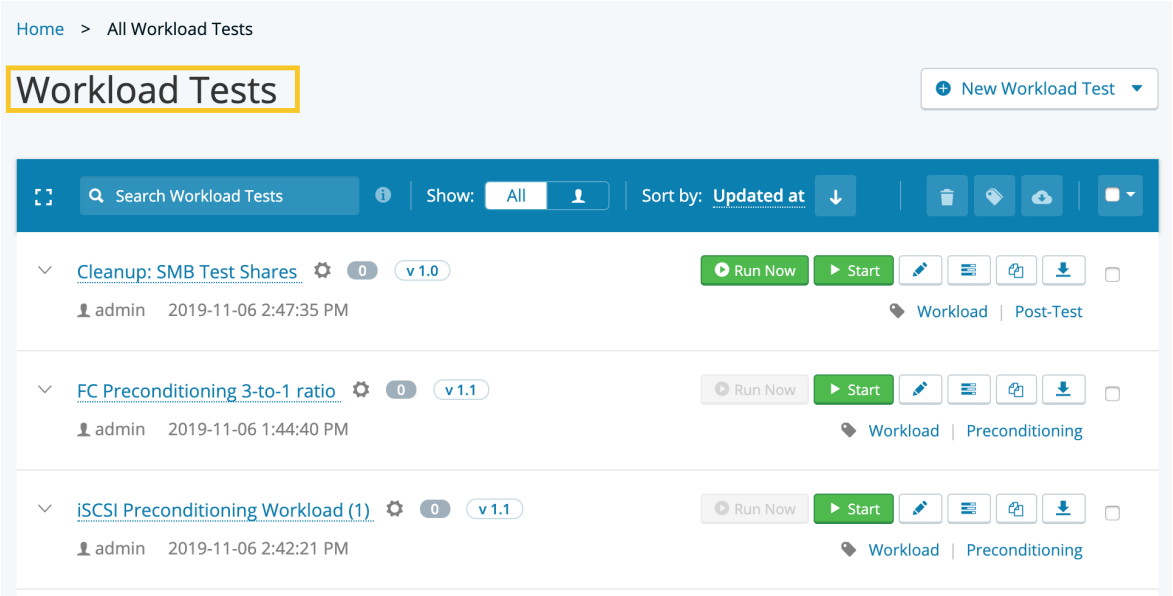
2. To use *Post-Test Cleanup* workload test, clone it from the library by clicking **Clone**.
3. Click **Start** or **Edit** to configure the Cleanup workload.

> **TIP**
> While both options allow you to define, customize, and run the Cleanup workload each time you run it, use **Edit** if you plan to configure a Cleanup workload with certain default settings that you always want to start with and use it repeatedly, and use **Start** if you plan to run the Cleanup workload ad-hoc right away.

4. Once cloned, configure the workload and select a Test Bed.



5. Depending on whether you selected **Start** or **Edit** earlier, click **Start** to run the workload right away, or **Update Workload Test** to apply the changes first and then open the workload to run it.
6. You can access the cloned Cleanup workload from the *Workload Tests* page.



7. Once the Cleanup workload is running, a default max duration of 7 days is set. This is the maximum time duration for post-test cleanup, not the actual expected run time. Post-test cleanup finishes when all the data is removed, an error is encountered, or after it reaches the maximum of seven days.

Additional SMB Options

The SMB Cleanup Workload has the following additional option:

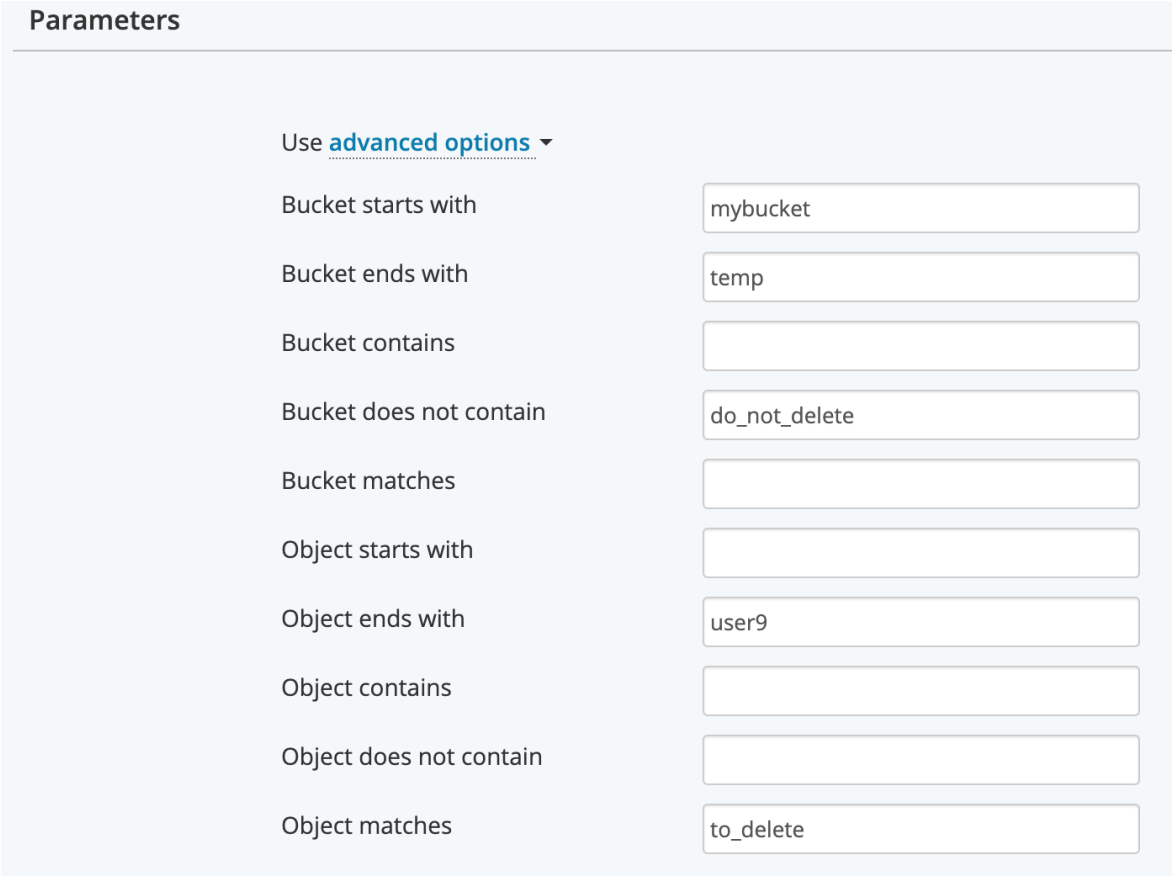
Remove read-only directories and files: If selected, directories and files that have the read-only attribute are also deleted during the cleanup workload run, in addition to directories and files that do not have the read-only attribute.

Additional S3 Options

The S3 Cleanup Workload has the following additional option.

The S3 Cleanup workload provides additional options to filter out a specific set of Buckets and / or Objects to delete from the Object Storage System.

1. Go to the **Parameters** section of the **Start** or **Edit** page for the cloned S3 Cleanup workload.
2. Use the **simple options** to simply specify one text based wildcard for Buckets and one for Objects, or use the **advanced options** to apply multiple text based wildcards for Buckets and multiple for Objects. With the **advanced options**, you can optionally use regex in the **matches** fields.



All Bucket fields are connected together with a logical AND, and all Object fields are connected together with a logical AND. If a field is blank, then that field is ignored.

In the example above:

- The following Buckets would be included in the Object delete list: mybucket_test01_temp, mybucket_test02_temp
- The following Buckets would be excluded from the Object delete list: mybucket_do_not_delete_test01_temp, mybucket_all
- Within the list of included Buckets:
  - The following Objects would be deleted: myobject_101_to_delete_user9, myobject_102_to_delete_user9.
  - The following Objects would not be deleted: myobject_101_save_user9, myobject_to_delete_user10.
- If a Bucket that is included in the list of Buckets is empty, then it will be deleted.

Test Hosts (Beta)

Starting with WorkloadWisdom 6.6.1, a new software based Workload Generator called StackPlay Engine is available in Beta, designed for testing hyperconverged infrastructures, software defined infrastructures, or any Windows/Linux instance whether it is a VM or a baremetal server. The new software based Workload Generator generates compute load, network load, and disk load simultaneously, simulating an application's impact on the infrastructure in its deployed environment.

Full documentation will be provided when StackPlay becomes a fully GA product.

To get started on this Beta product, please contact your Account team. Support will be provided by the Account team, Product Management team, and Engineering team until it becomes GA.

Chapter 4

Analyzing Workloads

Understanding real application IO workloads has traditionally been one of the most challenging barriers to storage performance validation, yet it is the most important because everyone's workload is different as even the same application can behave differently when deployed in different environments. The workload acquisition and analysis capabilities in WorkloadWisdom make it possible to understand and simulate your mission critical applications behavior across your entire infrastructure or your customer's entire infrastructure.

Workload IO data can be obtained from the data collected by storage or compute devices, and from Infrastructure Performance Analytics Solutions like Virtual Instrument's VirtualWisdom. The obtained data, called "production workload data," can be imported after it is processed into a supported CSV (Comma Separated Values) text file using the Workload Data Importer for workload analysis.

Workload Data Importer

The Workload Data Importer is designed to import and analyze workload data and then create a temporal workload model out of the imported workload data. The workload data can be exported from storage vendor system's IO stats or storage monitoring tools.

The data you import into WorkloadWisdom is in .csv format. The data is analyzed to understand the workload characteristics in terms of IOPs, throughput, read/write ratio, command distribution, and so forth.

Different storage systems collect different stats at different granularity, and the data that are collected are stored in different structures. WorkloadWisdom builds a dedicated analyzer and parser, called *Analysis Policy*, for each unique source of workload data. The Supported Analysis Policies section provides a full list of workload data sources that are supported.

For better manageability over time, imported workload data files are associated with *Production Storage Environments*. A *Production Storage Environment (PSE)* is a logical repository for one or more imported workload data files, and you should name it such that at a later date, you can easily find workload data for a specific storage environment that you care about. You can have, and are encouraged to have, multiple PSEs defined in WorkloadWisdom.

For example, if you are an IT Storage Engineer for your company's IT infrastructure, then you may have the following PSEs defined:

- Tier 1 Applications
- US West data center 2
- NAS Infrastructure
- Vendor X arrays

And if you are a QA engineer for your company's storage product team, then you may have the following PSEs defined:

- Customer X Mission Critical data
- Customer samples for regression resting
- Support Escalation

When you import a workload data file, you choose the PSE that you want to associate the file with. At a later time, if you want to go back to some past workload data from *US West data center 2* then it is easier to find it by browsing the PSEs you have created.

To get started, the following workflow is recommended, once you have the workload data file ready for import into WorkloadWisdom.

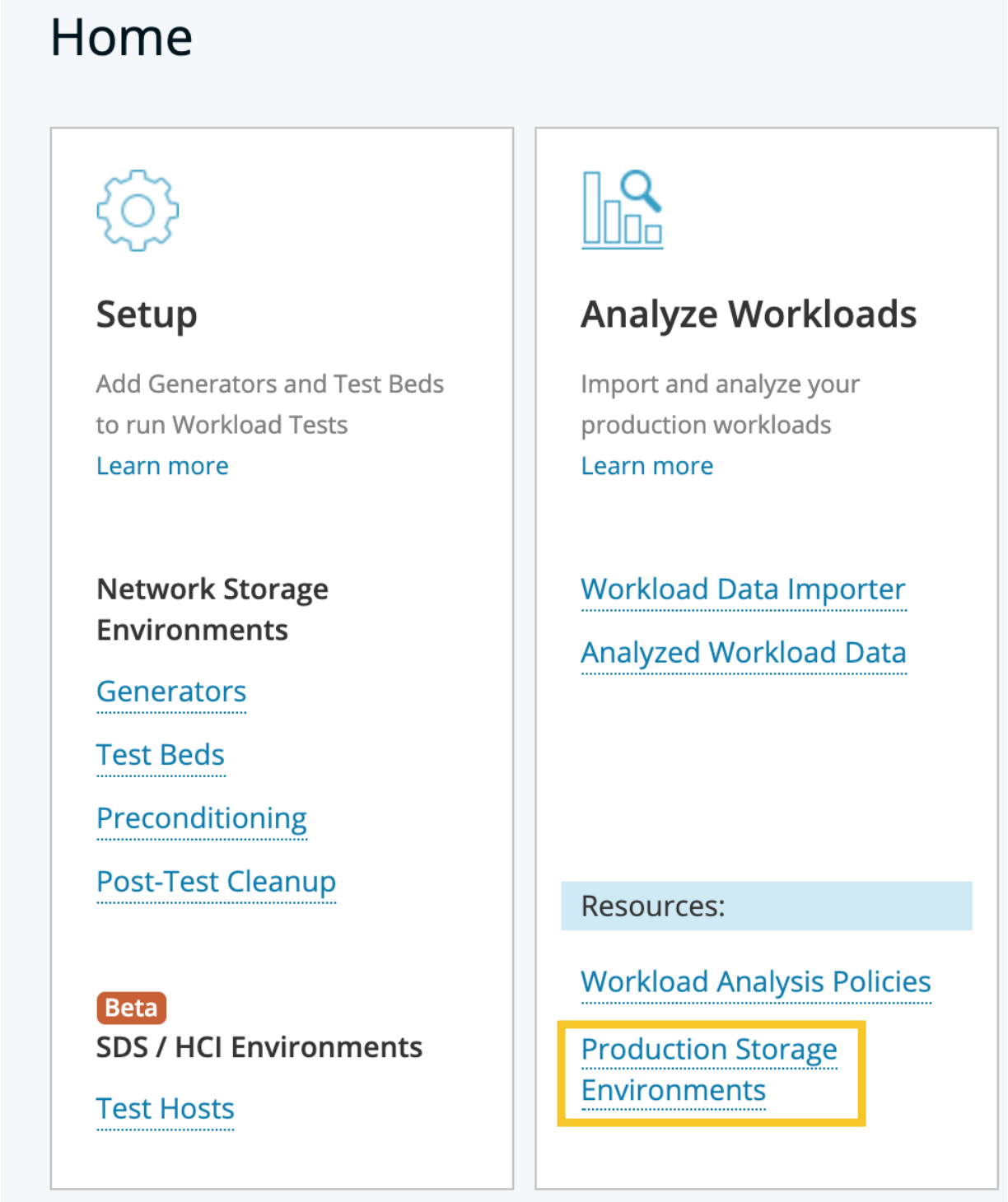
1. Create *Production Storage Environment*
2. Use *Workload Data Importer* to import the workload data file
3. Wait for the analysis to complete, and review the analyzed workload data
4. Create a workload model out of the analyzed workload data
5. Run tests using the workload model

NOTICE

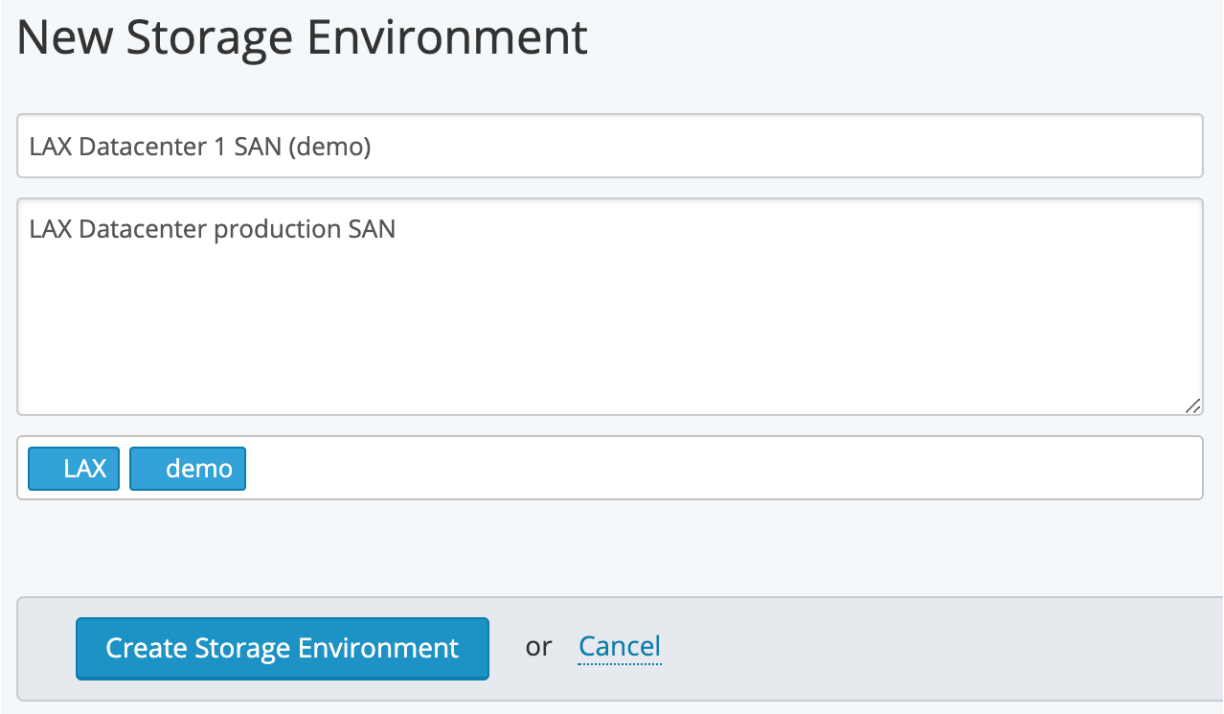
While the process of importing a supported workload data file into WorkloadWisdom is very simple and straightforward, much of the time may be spent collecting and extracting that data from a storage array in the first place. If this is your first time doing something like this, it is **strongly recommended** that you contact Virtana for help. Our Support team can help you with the general workflow and basics of extracting workload data, but it is beyond the scope of our standard product Support. For more hands-on and specific assistance, our Professional Services team can help train you and / or extract the workload data for you, so that you can become more familiar with the process of collecting workload data from your storage arrays.

Create Production Storage Environment

From the *Home* page, go to *Production Storage Environments*.


Home
Setup
Add Generators and Test Beds to run Workload Tests
Learn more
Network Storage Environments
Generators
Test Beds
Preconditioning
Post-Test Cleanup
Beta
SDS / HCI Environments
Test Hosts
Analyze Workloads
Import and analyze your production workloads
Learn more
Workload Data Importer
Analyzed Workload Data
Resources:
Workload Analysis Policies
Production Storage Environments

Click the button **New Production Storage Environment**, and then give it a descriptive name that you will likely be able to find at a later date. Also, optionally include a few words or sentences to describe the PSE. These content are searchable.



Click **Create Storage Environment**, and proceed to importing workload data.

Importing a Workload Data File

Once you have a supported workload data file in hand, you can start importing it using *Workload Data Importer (WDI)*. In most cases, you will import one single CSV file, and it is recommended that you import one single CSV file. However, you can import up to five CSV files at a time either directly or compressed with a ZIP file. When importing multiple files, they are processed and analyzed together – as if it is a single data set. A single Analysis Policy is used for each instance of the Workload Data Importer run, so the type and order of columns, as well as the formatting of individual values, must be consistent across all CSV files in the same instance.

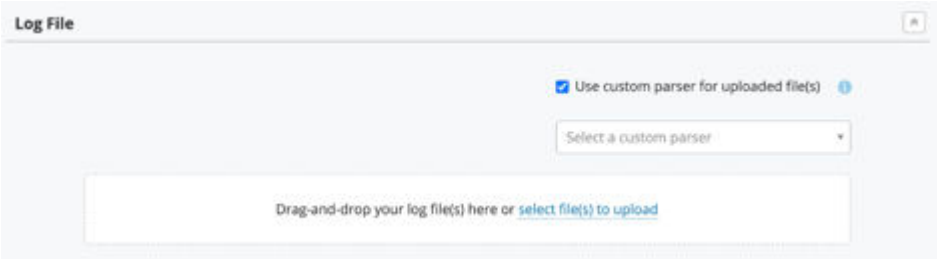
For example, attempting to import two CSV files at the same time, one in `EMC® VMAX®` format and one in `Hitachi® Tuning Manager` format is *not* allowed, as there is not a single analysis policy that can be applied to both formats. In this scenario, you need to do two separate imports. Also, importing two different CSV files from the same storage array, for example, one CSV file for command statistics and one CSV file for file system layout, is also not allowed, as these are still two different CSV file structures.

Additionally, when importing multiple CSV files, make sure that the data in the files does *not* overlap in time. Explained another way – you cannot have the same timestamp present in multiple files.You can select the files to import in two ways. You can drag up to five files into the *Drag and drop* section on the *Log File* screen. You can also click the *select a file to upload link*, which takes you into a standard file system dialog box, and then select the files there.

Custom Parser

If imported data cannot be recognized there is an option to use the custom parser. Custom parser uses a custom script produced by Virtana Professional Services team. The script prepares input data and converts it to the format that can be recognized by WorkloadWisdom.

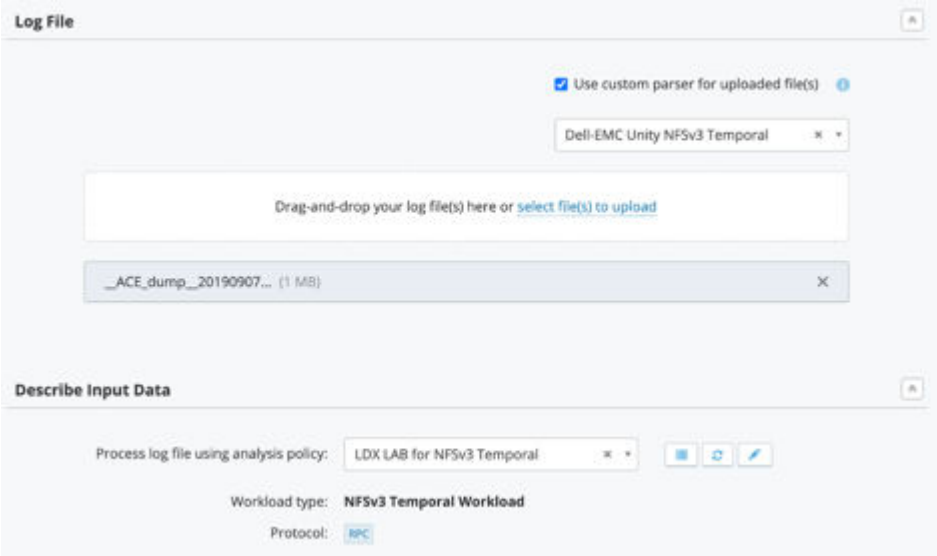
The list of available custom parsers expands as more custom scripts are produced. Currently, Dell EMC Unity data is supported.



Imported data must be in the form of a zip archive. The top level of this archive should contain files that are used as input data for the custom parser. As soon as the zip file is uploaded the parsing process begins.

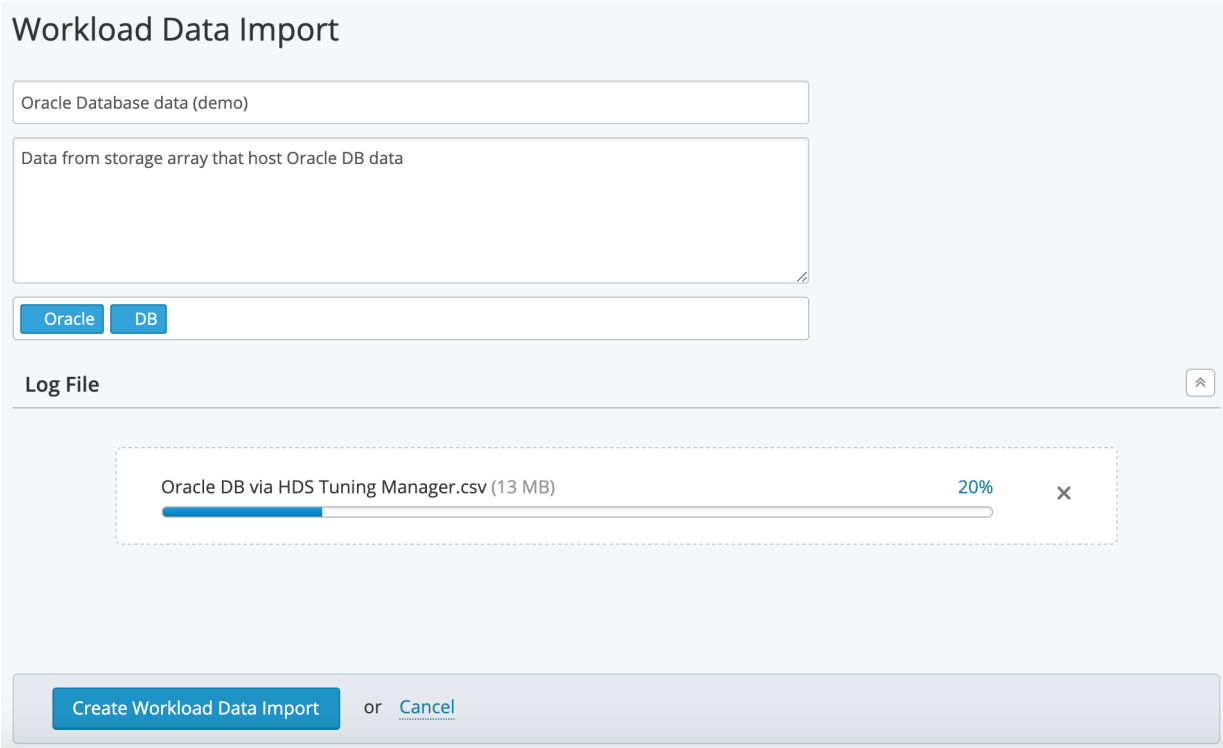
Parsing time depends on the size of input data. If any errors are produced by the parsing script, the errors are shown on the screen and the selected archive cannot be used.

When the custom parsing stage finishes successfully the output of the parser is used as imported data on the Workload Data Import page. The next steps are identical to the import process without using the custom parser.

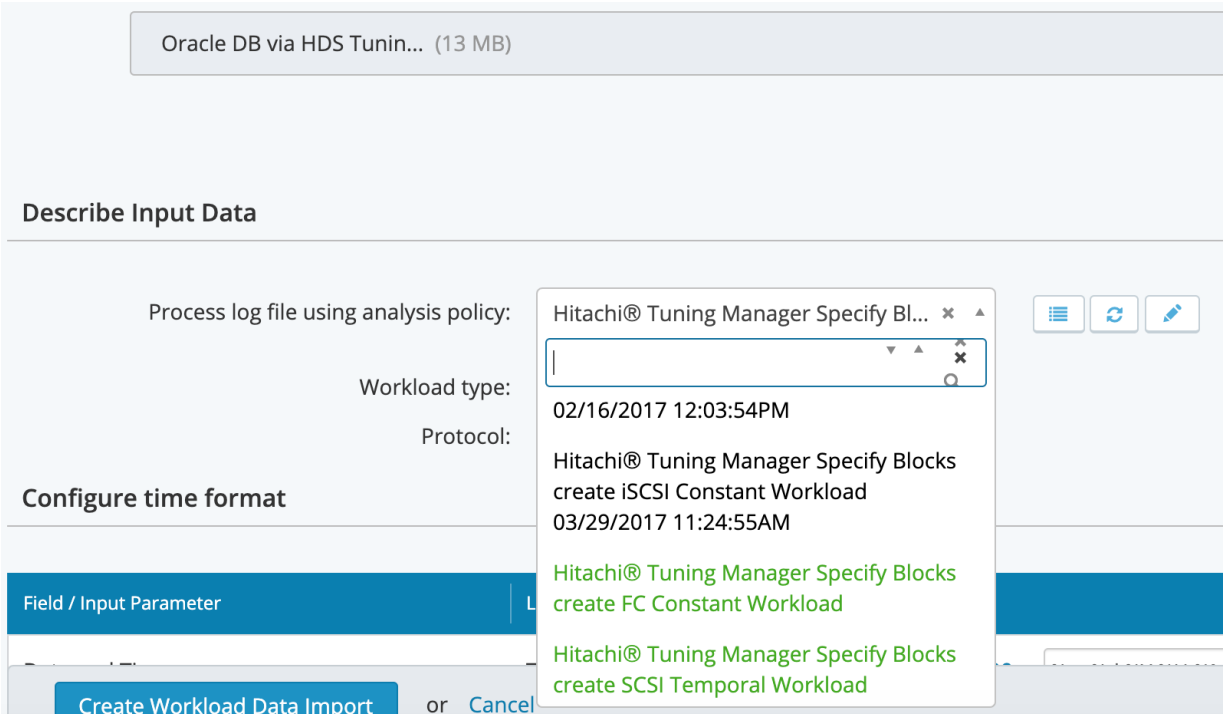


Import Process

The importer uploads the file and then filters the list of analysis policies to those that match the file. It then automatically shows a list of analysis policies that can be applied to the imported CSV file.



The default analysis policies that ship with WorkloadWisdom appear in green, and if you have any modified and customized versions of the analysis policy that is applicable to the imported CSV file, then they will appear in black.



Analysis policies parse and analyze the production workload data, and allow you to select the workload type you want to create from the workload data. Typically, there is a set of policies for each supported workload data source, where each policy in the set supports creating a different type of workload. Often, for each supported CSV file structure, there can be multiple analysis policies for it. For example, you can have an analysis policy for HDS data that creates constant workloads, and you can have an analysis policy for the same HDS data that creates temporal workloads. The analysis policy is labeled accordingly so you know 1) what data source it can analyze, and 2) what type of workload it creates.

Workloads are labeled *Constant* or *Temporal*. A temporal workload varies the IOPs over time, according to the patterns observed in the data, and can only be created through workload acquisition. A constant workload uses a set level of IOPs, based on observed values of IOPs, latency, or throughput. Constant or temporal workloads can be averaged into one summary workload, or clustered into multiple workloads that run together and make up a composite workload.

Select the analysis policy that creates the type of workloads in which you are interested. If you click the button next to the drop-down list it opens a new window or tab and takes you to the Workload Analysis Policy Library.

You can edit certain aspects of the policy that you selected for this workload by clicking on the button.

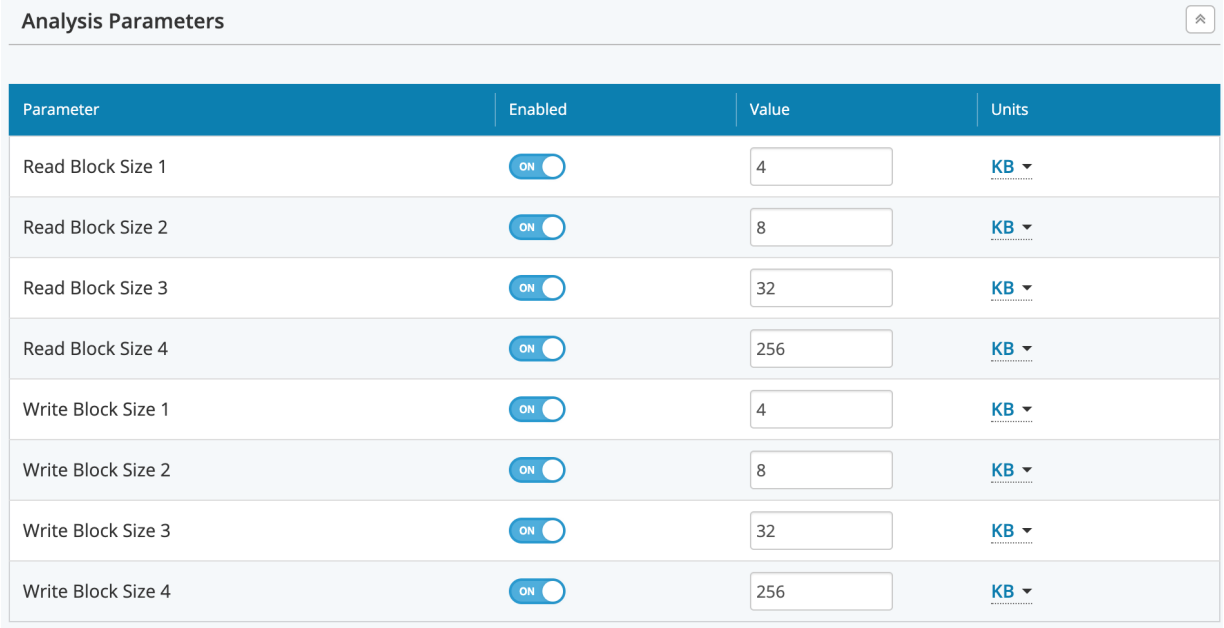
Items that are not visible in the edit page, such as what workload is created or how the data is analyzed, require Virtana Support to change them. If there is not an Analysis Policy that meets your needs, contact Virtana Support.

Editing Analysis Parameters

Analysis Parameters allow you to specify values to be used in creating workloads that are not available through acquisition. For example, actual block sizes are not tracked by most arrays. An average block size can typically be calculated by dividing throughput by read/write operations.

In reality, applications use a variety of block sizes so Analysis Parameters allow you to specify the block sizes to use. The analysis calculates the appropriate % of each block size that arrives at the average block size. For VirtualWisdom data, the top *ten* read block sizes and the top *ten* write block sizes are used during this analysis. For other logs, the top *four* read and write block sizes are used for analysis.

If you are working with workload data and an analysis policy that does not give the block sizes information at all, whether directly or indirectly, then you can still manually specify the distribution of block sizes. In order for a valid workload model to be created, at least one read and one write block size must be specified.



Other types of parameters can be configured in a similar fashion. The types of parameters that are user configurable are defined in the Calculation Rules Analysis Policy, which specifies what user input is required to support calculated values.

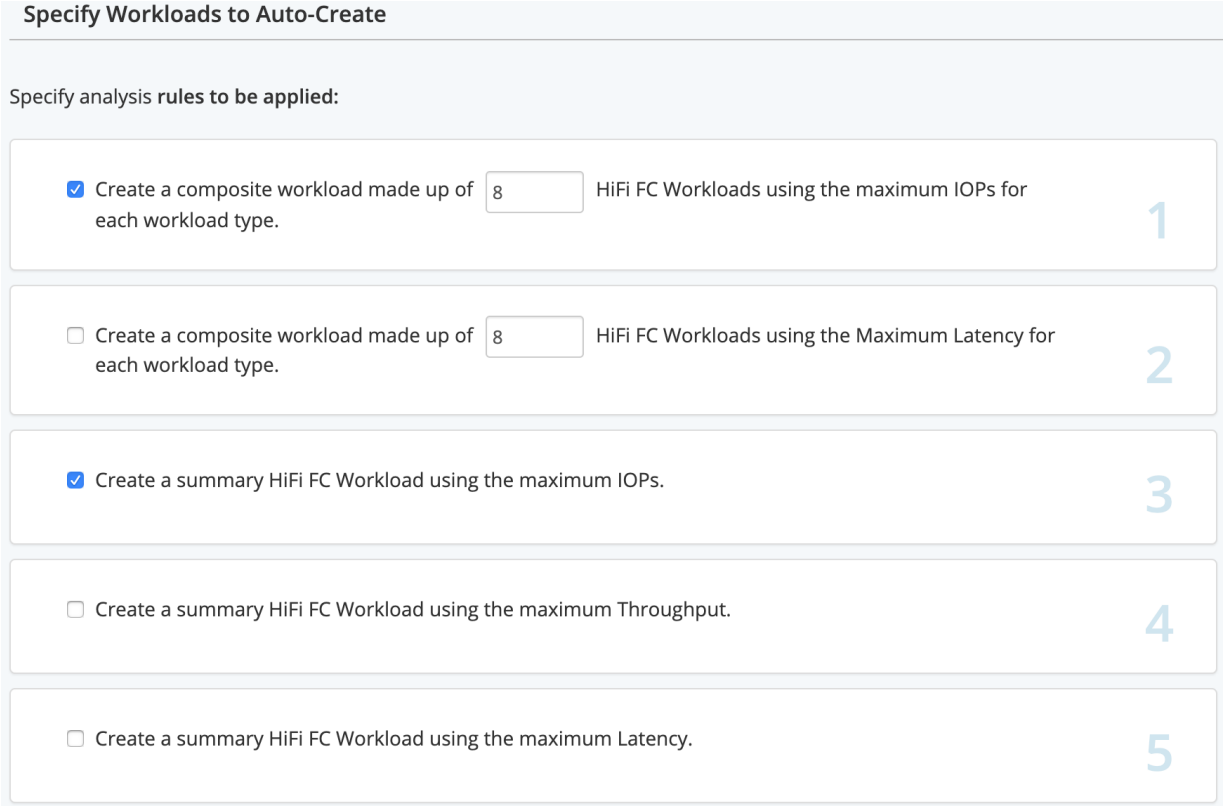
Specifying Workloads to Create

The Analysis Policy specifies the type of analysis to be done and the type of workload to be created. There are usually several workload model options, depending on the selected Analysis Policy. To review the difference between the different types of workloads, see the Running Workload Tests [89] section. At a high level, there are two decisions to make:

- Composite versus Summary
  - Summary: a single workload model that aggregates some of the workload characteristics. Simpler to manage but loses some fidelity.
  - Composite: a group of workload models that are joined together into a "super" workload, where each individual workload model has its own workload characteristics. More complex to manage but provides the highest fidelity.
- Temporal versus Constant
  - Temporal: a workload model in which the load varies over time as observed in the workload data. Recommended if your main use case is to size the system under test for day to day workloads.
  - Constant (maximum): a workload model that uses the maximum observed load all the time. Recommended if your main use case is to find the peak handling capabilities of the system under test.

You can select more than one workload to create. When you do, WorkloadWisdom will simply create the selected workload types at once, and you will have workload models for each of the selected options.

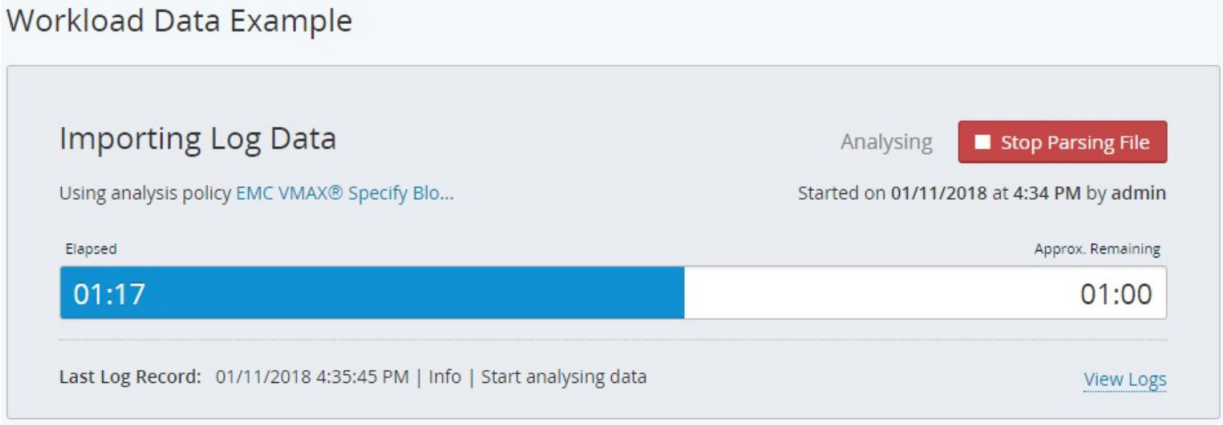
Once the parameters are set, click **Create Workload Data Import** to take you to the *Specify Workloads to Auto-Create* screen.



Visualizing the Production Workload Data Acquired from the Workload Data Importer

The *Workload Data Example* screen displays the workload's IO Profile, including IOPs, throughput, latency, read/write ratio, command mix, block size, and so forth.

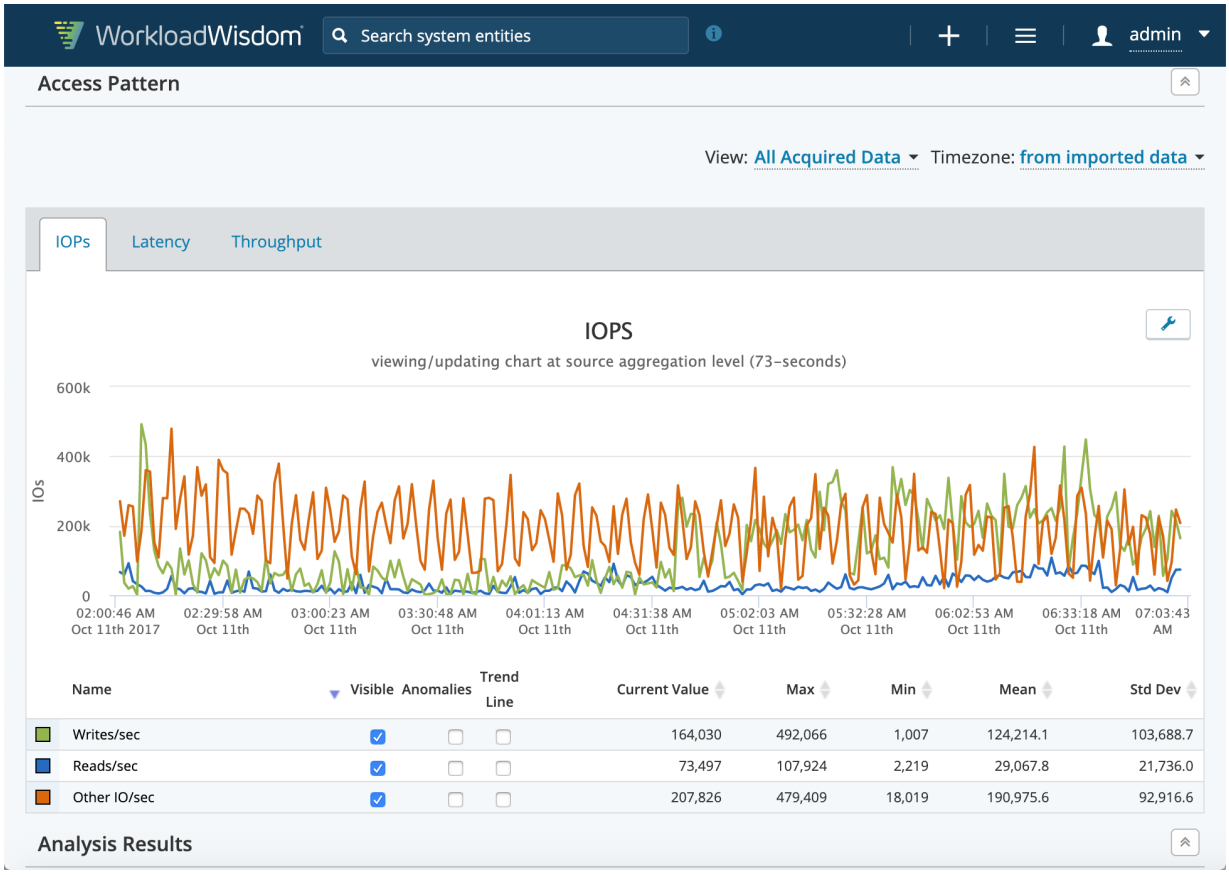
During a workload data import the data is displayed in real-time on the charts as it is parsed. The first section of the screen provides the status of data import and time remaining to complete the parsing and analysis. Click **Stop Parsing File** to stop parsing the file.

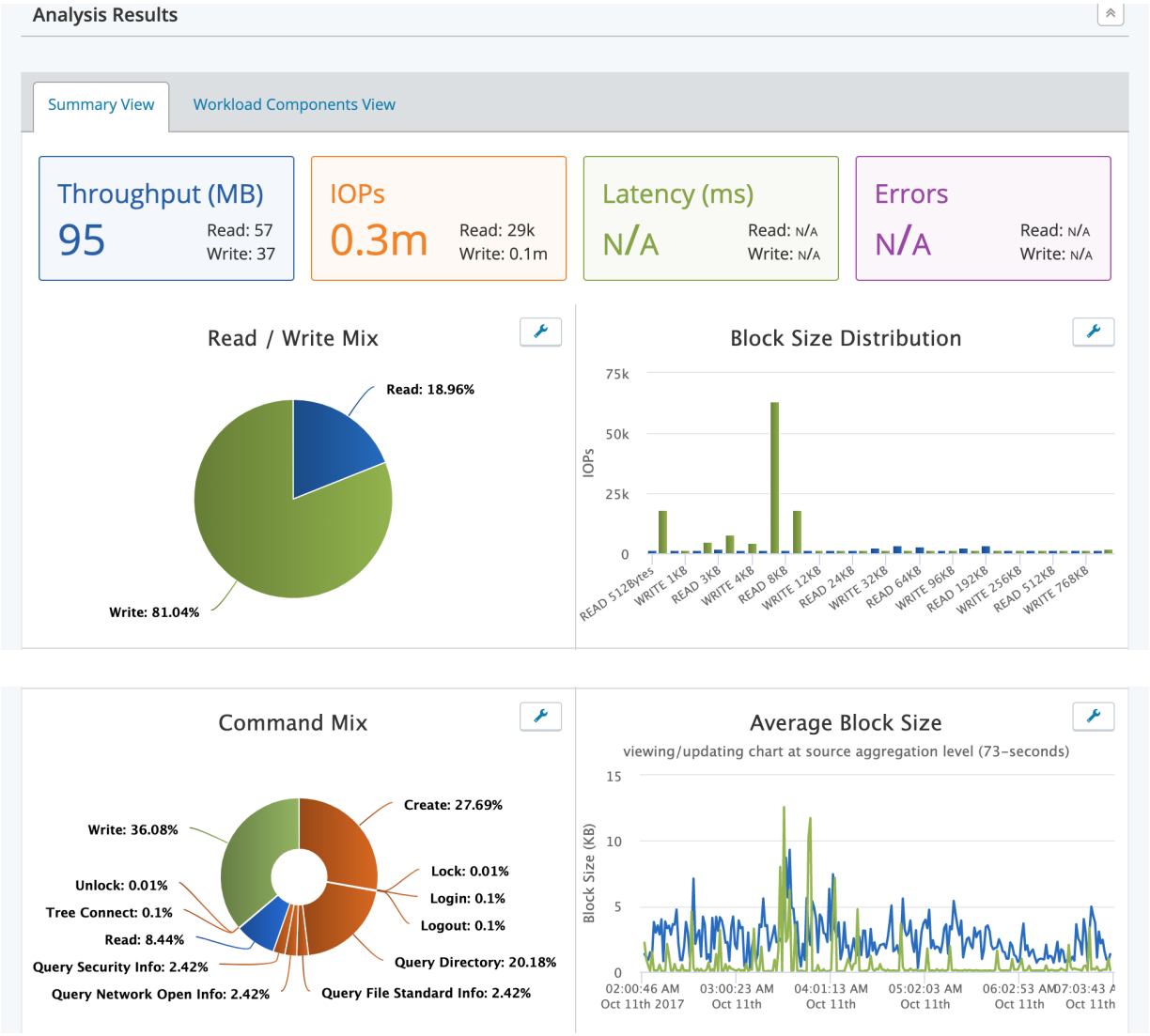


The information in the charts, how to use them, how to view the other sections of the production workload data, and how to create workloads for a specific time range, see Analyzed Workload Data [76].

Analyzed Workload Data

Analyzed workload data is the result of importing data using the Workload Data Importer for analysis and workload creation.

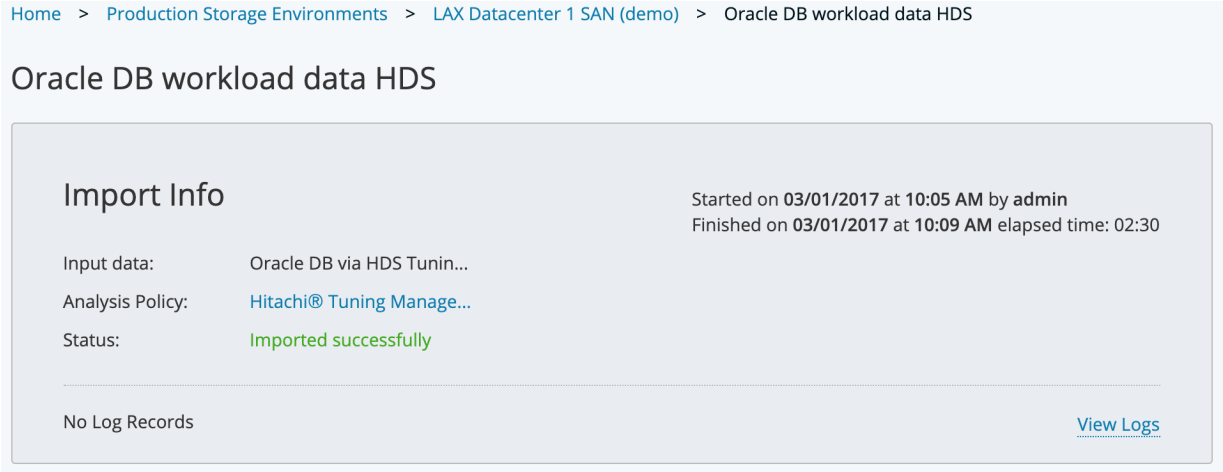

WorkloadWisdom
Search system entities
admin
Access Pattern
View: All Acquired Data
Timezone: from imported data
IOPs
Latency
Throughput
IOPS
viewing/updating chart at source aggregation level (73-seconds)
600k
400k
200k
0
IOs
02:00:46 AM Oct 11th 2017
02:29:58 AM Oct 11th
03:00:23 AM Oct 11th
03:30:48 AM Oct 11th
04:01:13 AM Oct 11th
04:31:38 AM Oct 11th
05:02:03 AM Oct 11th
05:32:28 AM Oct 11th
06:02:53 AM Oct 11th
06:33:18 AM Oct 11th
07:03:43 AM
Name
Visible
Anomalies
Trend Line
Current Value
Max
Min
Mean
Std Dev
Writes/sec 164,030 492,066 1,007 124,214.1 103,688.7
Reads/sec 73,497 107,924 2,219 29,067.8 21,736.0
Other IO/sec 207,826 479,409 18,019 190,975.6 92,916.6
Analysis Results



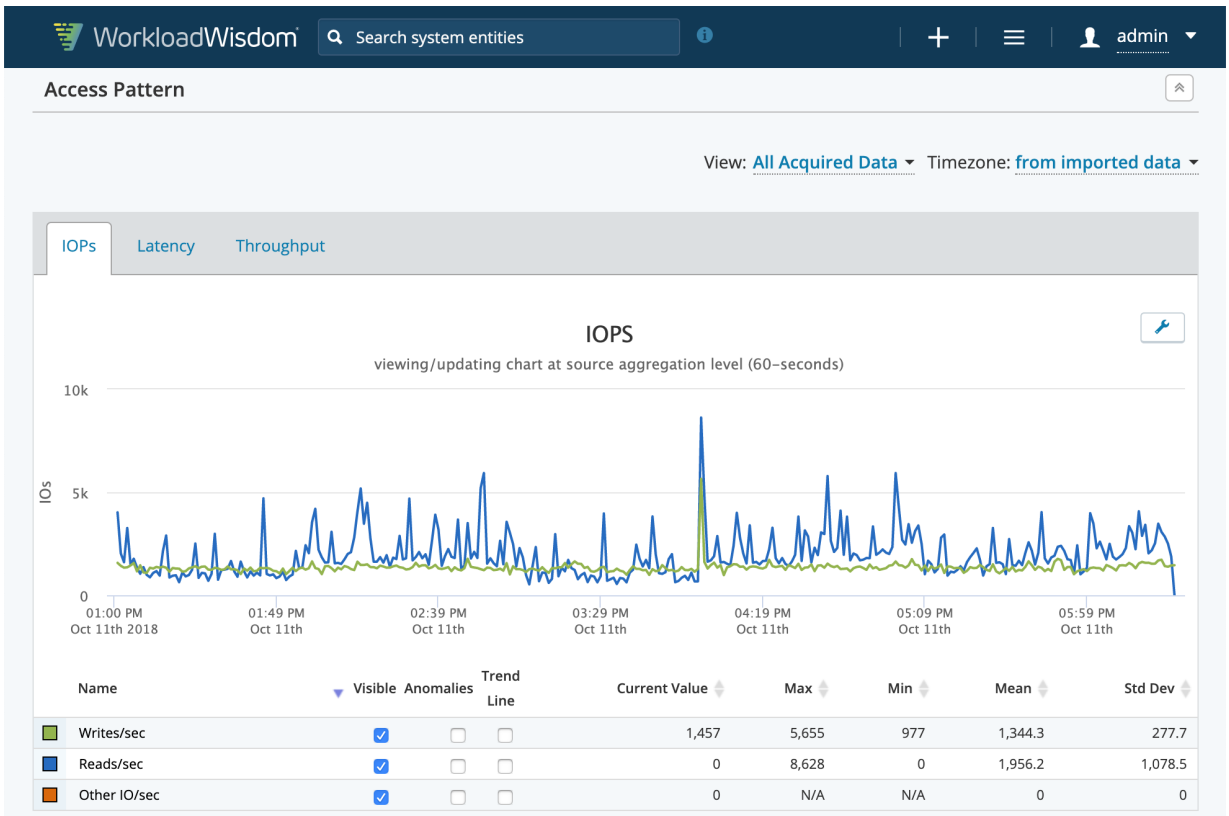
Locating Production Workload Data

If you followed the Workload Data Importer workflow, then you will automatically be brought to the analysis screen. You can also view analyzed production workload data that was previously done by clicking on it from the list of *Running Analysis* or *Completed Analysis*. You can also view it by selecting it from the list while viewing a production storage environment. You can also view production workload data automatically after starting an acquisition profile or while processing Workload Data Importer imports.

Visualizing the Production Workload Data



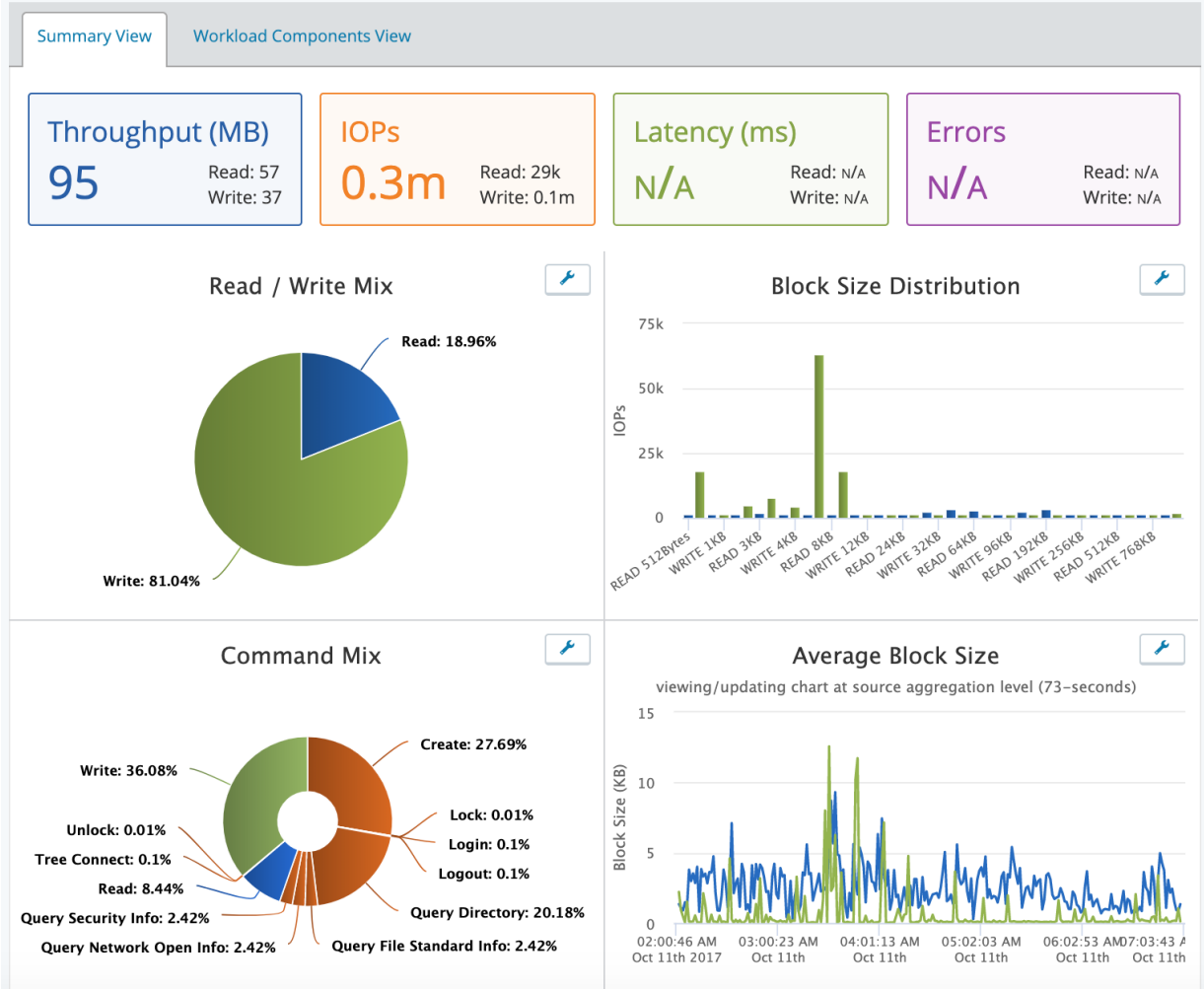
The top section displays information about the imported production workload data file, the analysis policy used, and status. If the data is currently being gathered, it displays a status for how long data has been collected or analyzed and approximately how much time remains. It displays if there are any issues gathering the data and has a link to a log.



The *Access Pattern* section displays *IOPs*, *Latency* (if available) and *Throughput*. Reads are displayed in blue, and writes in green.

Summary View

Summary View displays your analysis results and provides an overview of the production workload data. *Summary View* also provides an overview of the type of workload that would be created if you selected only a single summary workload. When you select a time window on IOPs, Latency, or Throughput chart in the *Access Pattern* section, the numbers on *IOPs*, *Latency*, *Throughput*, and *Errors* are changed on the corresponded values.



Read/Write Mix: Displays the read/write ratio observed in the workload data. This value is used to determine read/write ratio in created workloads.

Block Size Distribution: Displays the distribution of observed request sizes and their average IOPs by size. This information provides you insight into the block sizes that are configured in the servers and applications communicating with the storage infrastructure.

Command Mix: Displays the ratio of all observed commands. When troubleshooting, it is useful to determine if there are any unexpected commands or if the command ratio is higher than expected. Sometimes it is helpful just to see that a particular command is present, as some commands can be quite disruptive or are not seen in production environments.

Average Block Size: Gives a good indication of the changing relationship between IOPs and Throughput. Batch activities, transactions such as video streaming, or applications configured with large block sizes typically have higher average request sizes.

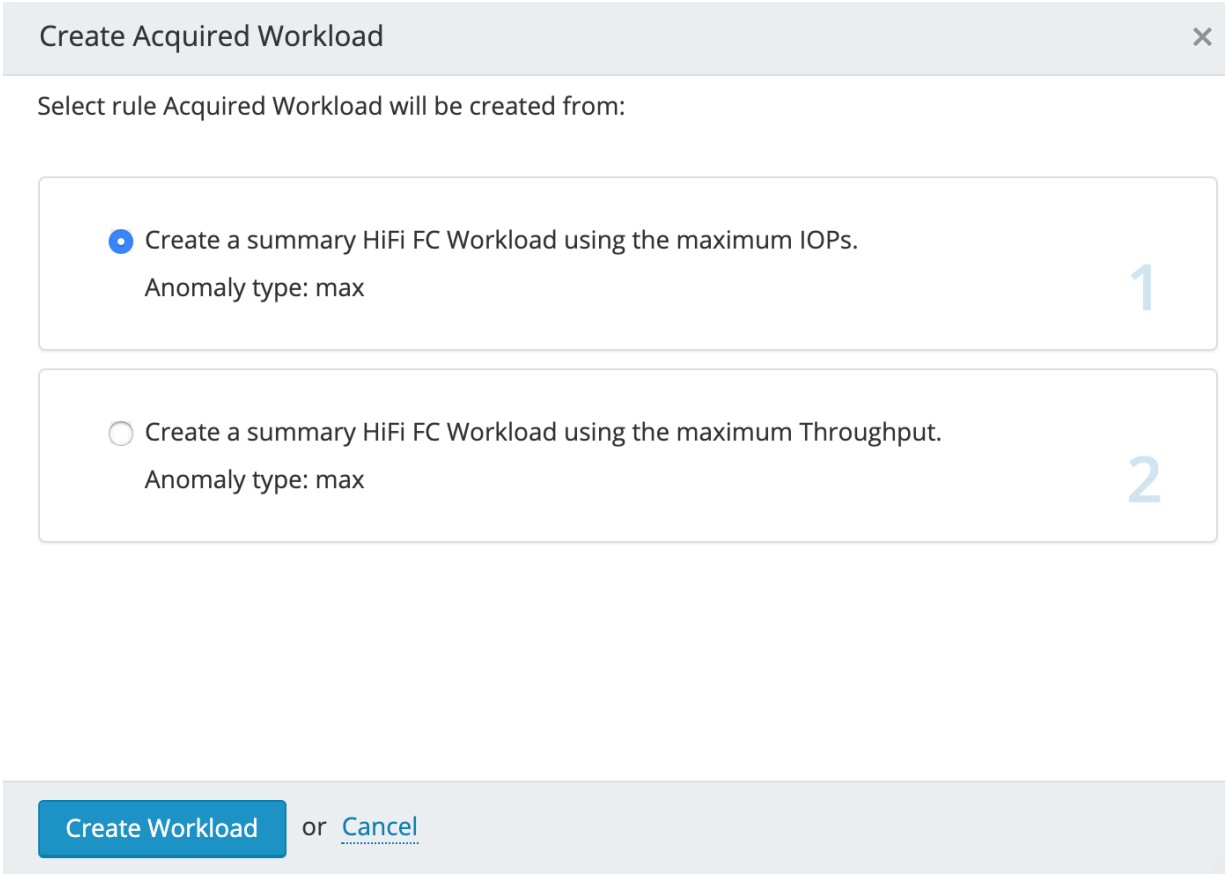
Latency vs. IOPs: Displays if latency is limiting IOPs or if IOPs are causing latency. The information is displayed as a scatter plot with a trend line. If the trend line ramps up to the right, the amount of IOPs is impacting the latency. The steeper the slope the greater the impact. Often there might be more than one apparent trend as different request sizes take longer. Intervals with fewer IO of large size tend to have higher latency. If the line is trending up this is generally an indication that the latency is caused by the IOPs, but is significantly limiting it. If the line is trending down or flat, this indicates that the array is having no trouble keeping up with the requested workload.

Latency vs. Request Size: Displays how much the request sizes are impacting performance. All things being equal, a block size that is twice as big as another should take twice as long. You might see that is not the case and it can be normal. For example, you might see that 4KB reads take the same amount of time as 8KB reads. This is often an indication that the array is configured with an 8KB block size, and is accessing 8KB, when only 4KB is requested. It might be a good idea to consider aligning the server block size settings with the arrays by changing one or the other.

Creating Summary Workloads from Production Workload Data

Click **Create Workload** to generate a summary workload from production workload data.

The following options are provided as displayed below, if during the Workload Data Importer process you selected multiple workloads to create and if the analysis policy you chose supports the workload types. So it is not a defect if you do not see this pop-up window at all.

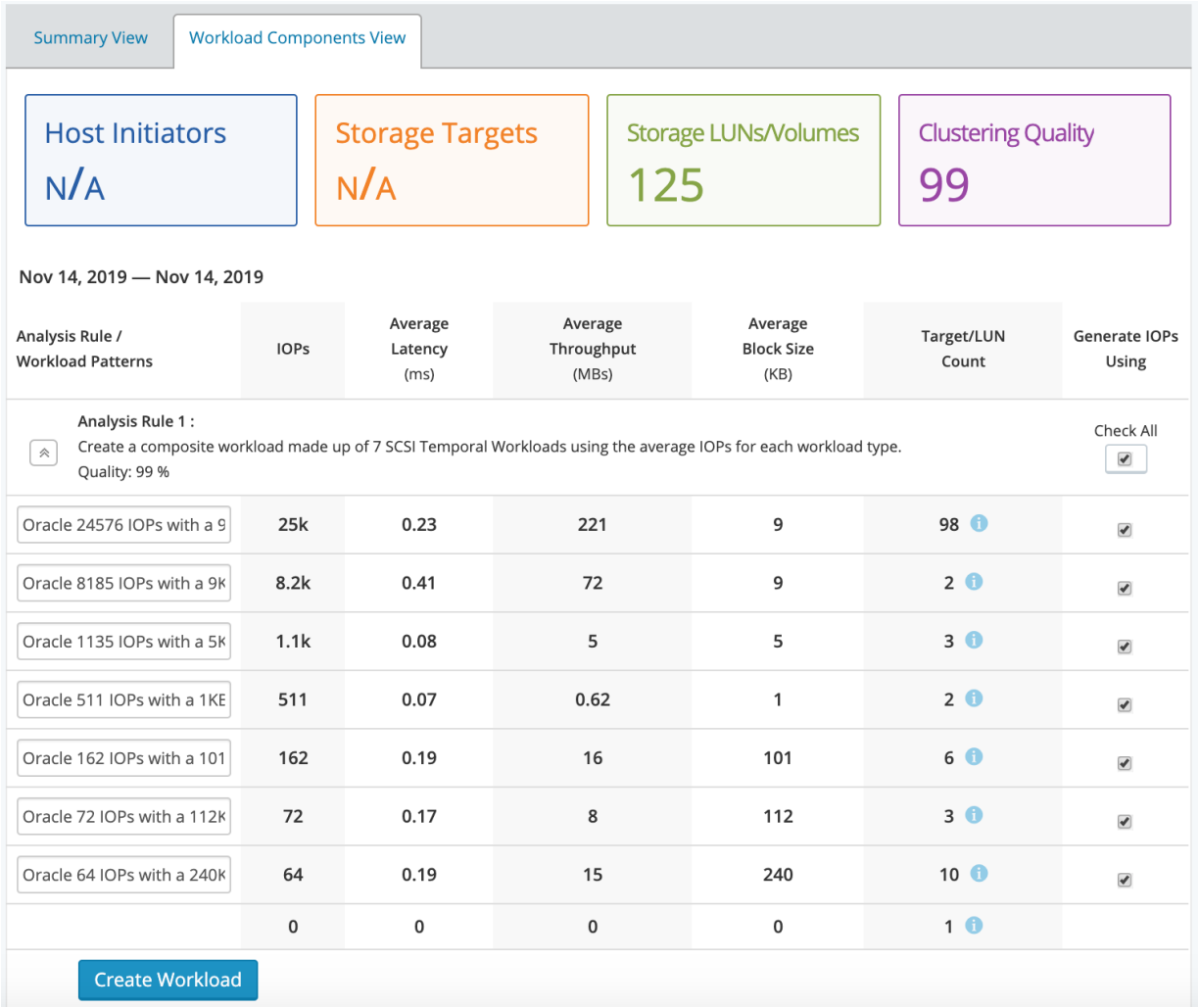


These options allow you to determine the amount of load to generate when running the modeled workload in a test environment. The option you select is reflected in the Load Properties section of the workload configuration on the next page. Regardless of which option you select, you can customize any of the parameters that are exposed in the workload configuration page before starting a test. See Creating a New Workload Test [92]. The option you select modifies a set of defaults in the workload configuration page so that you do not have to manually enter the information.

Workload Components View

If during the Workload Data Importer process you selected the Composite Workload option and if the analysis policy supports it, then the *Workloads Components View* tab is available. *Workloads Components View* displays the discrete workloads identified by WorkloadWisdom's workload analytics. The key characteristics for each identified workload are displayed, including how many Target/LUNs or SDFs (Source Destination Filesystem ID). *Quality %* indicates the relative accuracy of workload identification determined by the analytics.

You can rename workloads inline based on their observed behavior, number of LUNs or shares, or on the suspected type of workload present, like writes to Redo LUNs of a database or shares.



Each component workload is automatically named with the first word from the production workload data name along with the IOPs, average request size, and number of LUNs or SDFs that the workload represents and should be run against. The workloads are automatically sorted by the descending value of IOPs.

NOTE

You can use special characters such as underscore (_) or hyphen (-) in the first “word” of the production workload for a more detailed name for each component workload. Any workload with less than one IOP is displayed, but it cannot be used and a workload is not created.

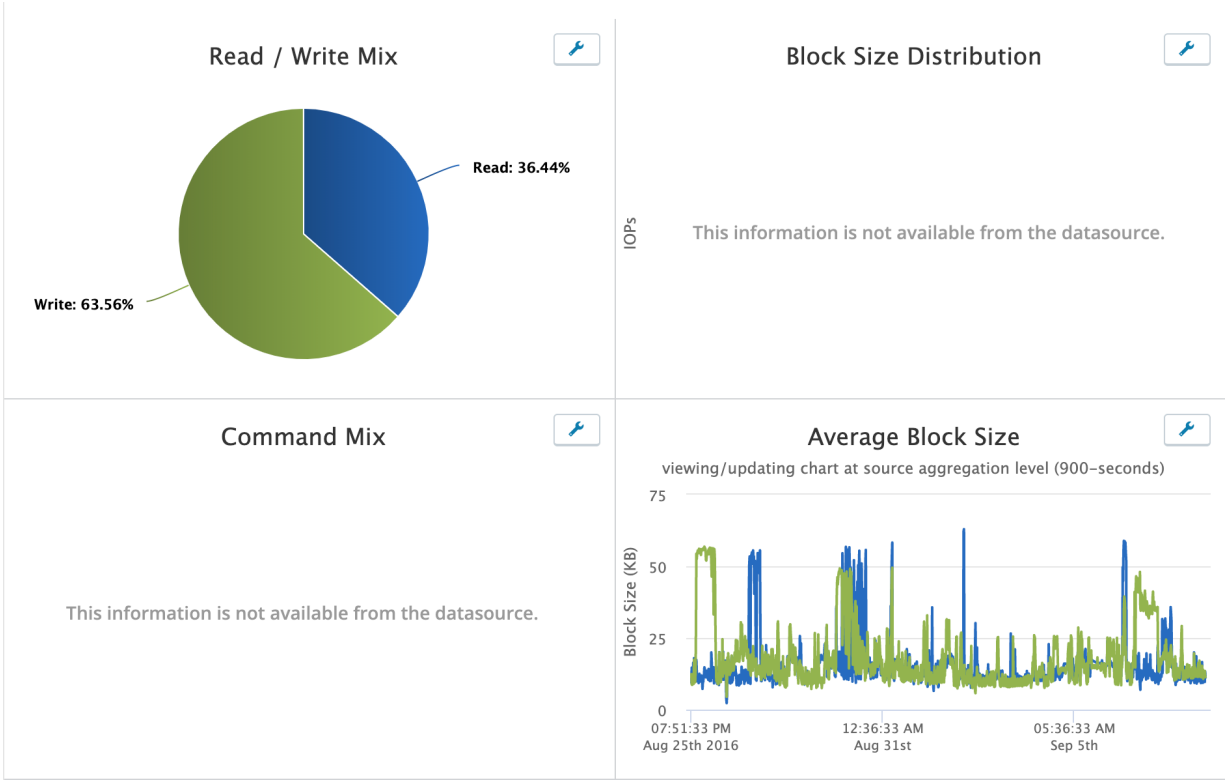
Creating Composite Workloads from Production Workload Data

From the *Workloads Components View* tab, click **Create Workload** to generate a composite workload from production workload data. A workload is created for each of the selected components. The workload is named according to what was entered in the box in the first column. A composite workload is also created that includes all of the selected workloads. Once the workloads have been created, you are redirected to the newly created composite workload. Each component workloads is mapped to a test bed connection with the number of LUNs or SDFs clustered from the production workload.

Analysis Results for Workload Data Importer

Depending on the source of the data that is imported not all of the information might be available on the *Analysis Results* page.

For example, the following screen shot displays a case where the storage array does not collect enough information from its production workload data to be analyzed more thoroughly.



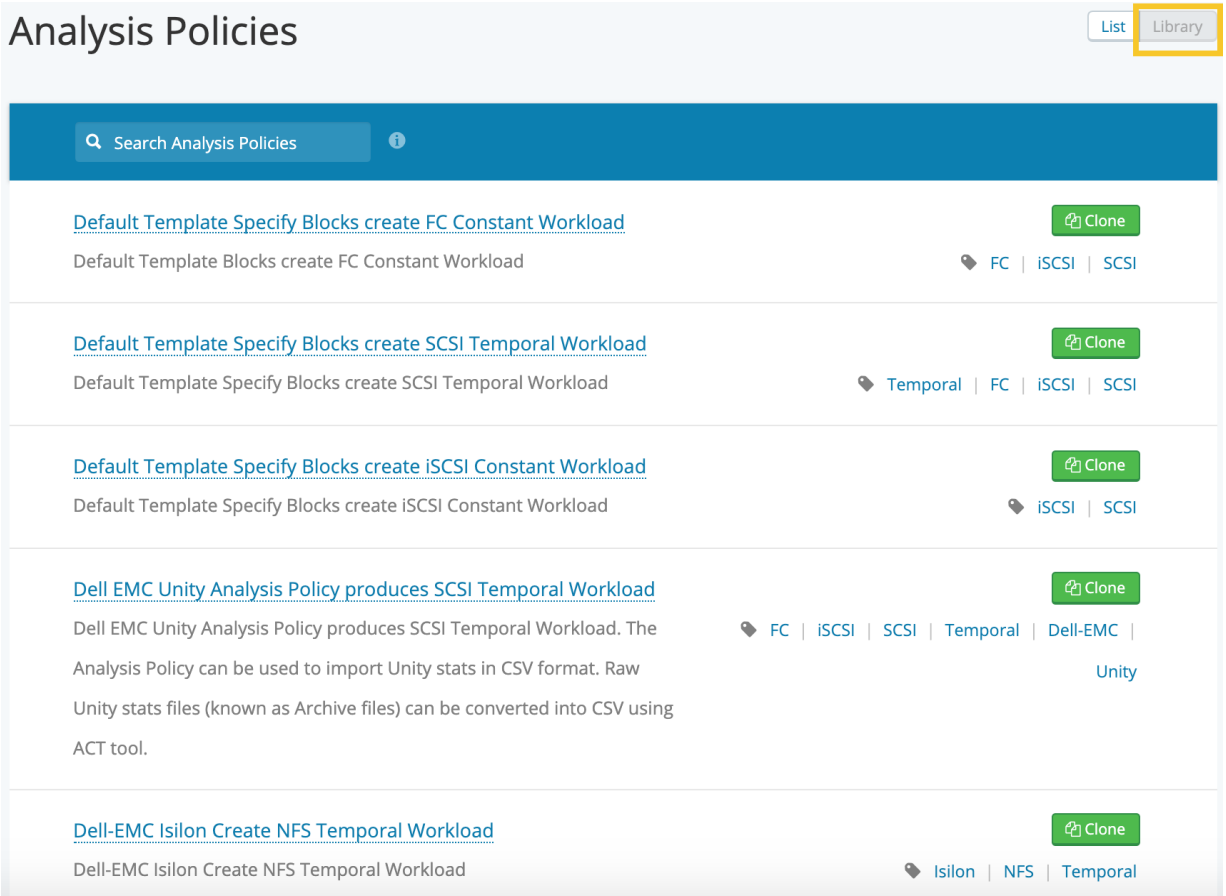
Resources for *Analyze Workloads*

Using Workload Analysis Policies

Workload analysis policies are used to analyze the imported workload data.

An analysis policy defines how a production workload is characterized. An analysis policy maps data from different sources so that it can be visualized, analyzed, and used to create one or more workloads. Therefore, a dedicated analysis policy is required for each unique data source structure.

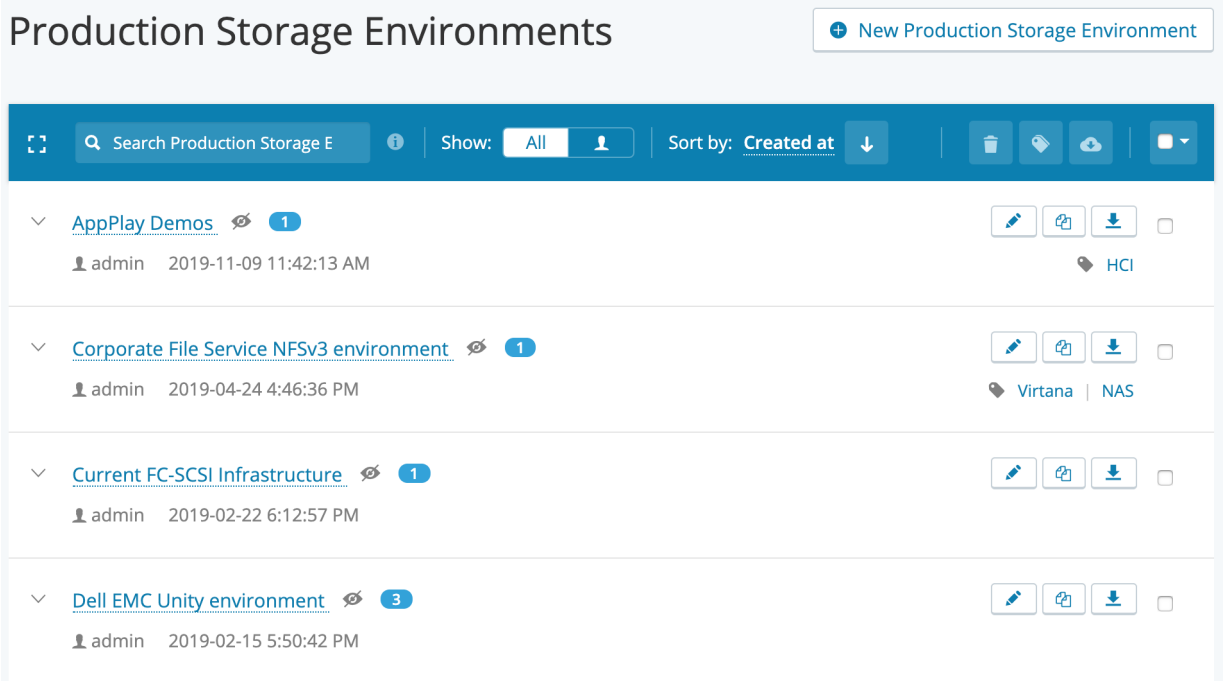
A set of out-of-the-box analysis policies are provided for supported storage arrays that are generally well known. In addition, a set of default (generic) analysis policies templates are also available for you to customize. If you are not familiar with the concept of analysis policies, work with Virtana Support to make changes to them.



For a list of supported analysis policies that are shipped with the WorkloadWisdom, refer to Supported Analysis Policies."

Using Production Storage Environments

Production Storage Environments (PSE) describe the infrastructure where production workload data comes from. This enables all data from the same source to be easily accessed.



Adding/Editing a Production Storage Environment

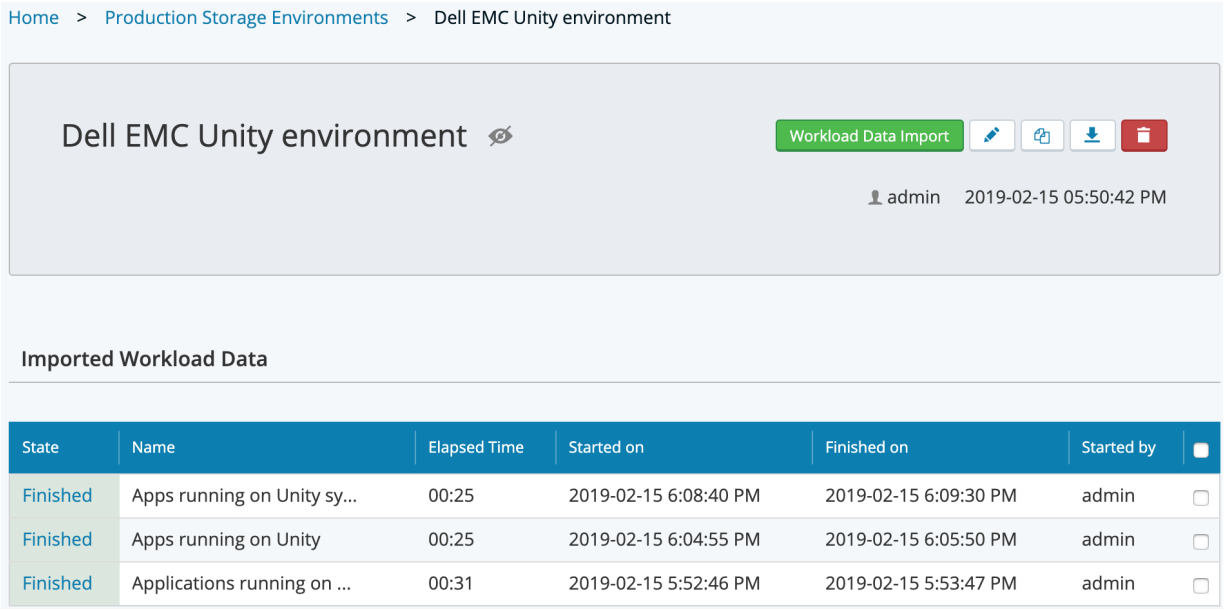
Click **New Production Storage Environment** to add new Production Storage Environments. You can edit existing PSEs by clicking the edit button next to the PSE that you want to edit.

The following information about the PSE can be added or modified.

- *Name*. Name of the PSE. It is recommended that the name follows a convention that allows for the identification of network segments being monitored.
- *Description*. Short description of the PSE. It is recommended that you use this to describe the environment that you are monitoring. The description might be an application or a portion of the storage environment such as a single storage array.
- *Tag List*. Tags make it easy to search for a specific PSE if you have a large deployment. Examples of good tags include network segments, applications, or arrays that being monitored.

Viewing Workload Production Storage Environment

When you click a Production Storage Environment (PSE) you can see and manage all of the production workload data acquired for the PSE.


Home > Production Storage Environments > Dell EMC Unity environment
Dell EMC Unity environment
Workload Data Import
admin 2019-02-15 05:50:42 PM
Imported Workload Data
State Name Elapsed Time Started on Finished on Started by
Finished Apps running on Unity sy... 00:25 2019-02-15 6:08:40 PM 2019-02-15 6:09:30 PM admin
Finished Apps running on Unity 00:25 2019-02-15 6:04:55 PM 2019-02-15 6:05:50 PM admin
Finished Applications running on ... 00:31 2019-02-15 5:52:46 PM 2019-02-15 5:53:47 PM admin

Chapter 5

Running Workload Tests

- **How to Read This Chapter**
- **Workload Tests Concepts**
- **Creating a New Workload Test**
- **File Protocol Workloads**
- **Object Protocol Workloads**
- **Block Protocol SCSI Workloads**
- **Block Protocol FC-NVMe Workloads**
- **VDI Workloads**
- **Composite Workloads**
- **Pre-tests**
- **Workload Suites**
- **Iteration Suites**
- **Resources for** *Run Workload Tests*
- **Conditions**

Use the *Run Workload Tests* section, located on the third-from-the-left vertical section of the *Home* page, to create and run workload tests and suites.

How to Read This Chapter

Many workload models are available in WorkloadWisdom, and they share many similar structures and concepts, such as Read/Write percentage and Load Properties.

This section starts by describing general workload concepts that are common to most workload models, and then subsequent subsections describe protocol specific differences where applicable.

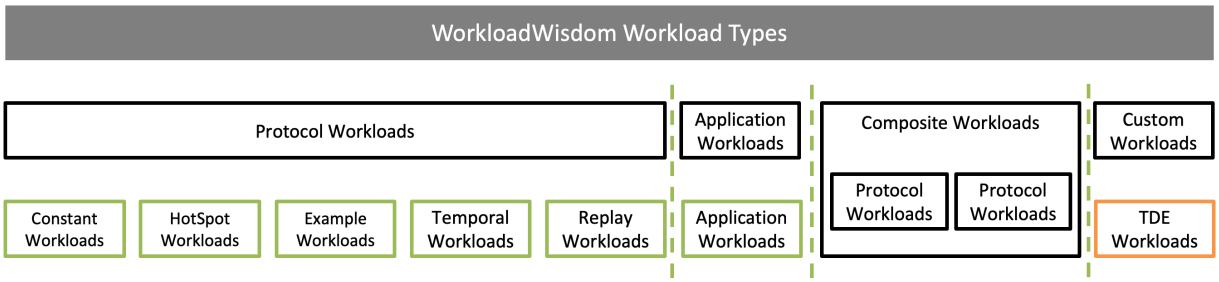
For example, Data Content is a common workload concept that applies to all protocol workloads, but block workloads have data deduplication and design that vary from file workloads. Those protocol-specific implementations are described in the Block Protocol Workloads subsection and the File Protocol Workloads subsection, respectively. Similarly, Object Protocol Workloads have unique concepts that are different from block and file workloads.

Workload Tests Concepts

For storage protocols, a workload is comprised of the storage I/O requests made by applications or host systems against the storage infrastructure. When these requests are analyzed in terms of IOPs, read or write percentages, sequential or random behavior, and block size distribution, their characteristics can be used as the basis for modeling that workload. The resulting model can be used to simulate the workload to support numerous activities across the storage lifecycle development, pre-production or production uses like testing, technology or design evaluation, capacity planning, performance analysis and optimization.

A workload test is a configured instance of a workload that is ready to run to produce test results. Consequently, an executing or executed instance of a workload test is a workload test run.

WorkloadWisdom supports different types of workload tests to support different use cases. The classification of the workload tests is illustrated and described below. Understanding the category of a workload model helps understand the similarities and differences between different workload models' design functionalities, and when to use what.



- *Protocol Workloads*. Out-of-the-box protocol workload models for key file, block and object protocols that make it easy to create and model storage workloads specific to your environment or applications. A protocol workload is not based on any specific application, but a template of protocol parameters and I/O characterization settings. This enables you to simulate an application workload as it would be observed on the wire. In a protocol workload model, key workload parameters can be varied such as % of read vs % of writes, % random vs % sequential, distribution of file or block sizes, and so forth. Several types of protocol workloads are available:
  - *High Fidelity Workloads*. Constant workloads that do not vary the amount of generated Load over time. “Load” in this context can be specified in terms of Throughput or IOPS. They are recommended for use cases where changes are made to the environment (such as introducing a failure or turning on backups). Because the workload is constant it is easier to measure the impact of the changes in the environment. They are also good tests to run for long periods of time at a high workload to ascertain if performance degrades over time.
  - *HotSpot Workloads*. Constant workloads that provide the ability to specify the locality and drift of IOs throughout the workload test run.

    NOTICE
    HotSpot Workloads will be not be further enhanced going forward, and will likely be deprecated in upcoming releases.

  - *Example Workloads*. Constant workloads with pre-defined protocol parameters and I/O characterization, based on a specific instance of a specific application in a specific deployment. The source of the pre-defined protocol parameters and I/O characterization is provided for each example workload. Example workloads are recommended if you do not have data about the production workloads that you want to model and you want some examples to start with.
  - *Temporal Workloads*. Protocol workloads that change their behavior over time. They represent the most realistic behavior and can be useful for assessing impacts of bursts or changes in workload behavior impact to performance.
  - *Replay Workloads*. Protocol workloads that replays the exact command sequence as recorded in the supported source files that provide per command sequences.

- *Application Workloads.* WorkloadWisdom also provides several highly intuitive application workloads out of the box, implemented based on specific applications analyzed by Virtana. Application workloads differ from protocol workloads in that application workloads provide parameters that are specific to the application, such as Number of VMs and Linked Clone Size for VDI application workloads, instead of protocol parameters such as % of read vs % of writes in the case of protocol workloads. In addition, Application Workloads may also vary certain aspects of load over time based on user inputs on application behavior over time, but not directly control exactly how that load varies over time. Application workloads are recommended for those who are familiar with the specific application they want to model, and who want to do so using parameters specific to that application.
- *Composite Workloads.* Special category of workload test that enables multiple existing protocol workload tests to be combined into one, such that each individual Protocol workload test carries its own unique IO profiles as opposed to aggregating or averaging the IO profiles into one. Composite workloads are very useful in realistically simulating modern applications in which they tend to have multiple concurrent processes, and each process can have unique IO patterns.
- Custom Workloads.
  - *TDE Workloads.* Specific protocol workload models or application workloads are not provided, although Workload Generators support the protocol. An existing test, authored in the Virtual Instruments Test Development Environment (TDE) can be imported and run in WorkloadWisdom. TDE workloads are recommended for those who are storage protocol experts, and who have a strong need to create completely customized workloads that are not provided by WorkloadWisdom.

Creating a New Workload Test

1. From the *Home* screen, choose *Workload Models* from *Run Workload Tests* section or *Create Workload from Library* from the new item menu.
2. Select the workload model from the list of out-of-the-box models listed. To create a new workload based on that model, click **Clone**.
3. A dialog box displays
   a. Enter a name for your workload
   b. Click **Clone** to complete the creation of your workload.
4. A new workload is created from the library and added to your workloads list.
5. You can hover over the buttons to identify their function. The following buttons are available:
   - Use **Run Now**, if activated, to run the workload immediately without further changes or reviews.

> **NOTE**
> **Run Now** is activated when it contains all required settings. If this is not activated for a workload, most likely it is because the workload does not have a Test Bed selected.

- Use **Start** to load the Start page for the workload with the current settings, and optionally set runtime parameters for this Workload Test Run.
- Use **Edit** to edit the workload settings and customize it to your application values.
- Use **Generate Test Bed** to generate a test bed from the workload. See Test Beds [26].

> **CAUTION**
> **Generate Test Bed** is *not* designed to create a fully complete test bed in all cases, especially when working with TDE workloads. The function is designed to serve as a template tool, to help you get started with creating a test bed from an existing workload. You need to complete defining the test bed.

- Use **Clone** to create a copy of the selected workload.
- Use **Export** to export the selected workload to a zip file that can be imported into any WorkloadWisdom instance running the same version.
- Use **Delete** to delete the selected workload.

Common Workload Concepts and Settings

Each protocol has specific setting available to describe the I/O access, read/write behavior, metadata, file or object system, pre-test, data, error handling and other relevant parameters that control workload-specific behavior. You can edit and save these settings with the workload. You can also modify these settings when you start the workload. WorkloadWisdom uses a variety of techniques, including probabilities, loops, threads, and others, to ensure the user-configured percentages are generated as accurately as possible. It is impractical to describe all possible permutations of configurations in this guide.

For advanced protocol users only: If it is absolutely necessary to view the exact command sequence of a workload configuration, it is recommended that you 1) run a workload test, 2) export the specific workload test run's configuration file from its test run results dashboard, and 3) import it into TDE (Test Development Environment). You can

contact Support to download Virtual Instruments' Test Development Environment (TDE) software, a Windows-based program that enables custom workload creation and modification.

The sections common to all workloads are the *Name*, *Description*, *Tags*, *Privacy*, *Data Parameters*, *Load Profile* and *Tracing Parameters*. Protocol specific parameters are discussed in subsequent sections.

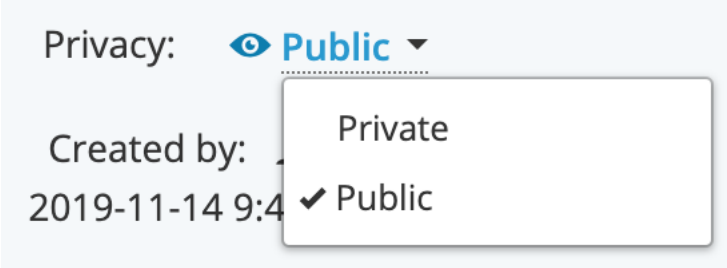
Name, Description, and Tags

Click in the associated text box to edit a workload's name, description, and tags.

Tags are automatically created from one or more words entered in the tags box when you press **Return**. To remove tags, click the x associated with a tag. Virtual Instruments recommends that team members use the same tagging standards.

Privacy and Access

You can set a workload for public or private access. Public access means that all users can see and use workload. You can edit a workload if you are the owner. If you are not the owner, you can save the workload by clicking *Save a Copy*. Only the user who created a private workload can access it. To change a workload's privacy setting, click on the down arrow and select *Public* or *Private*.



Access Patterns (Constant Workloads Only)

Access Patterns help you understand the frequently and means by which your storage is accessed. It is important to consider several access pattern use cases, such as average or peak usage, business cycles, or special events such as marketing campaigns.

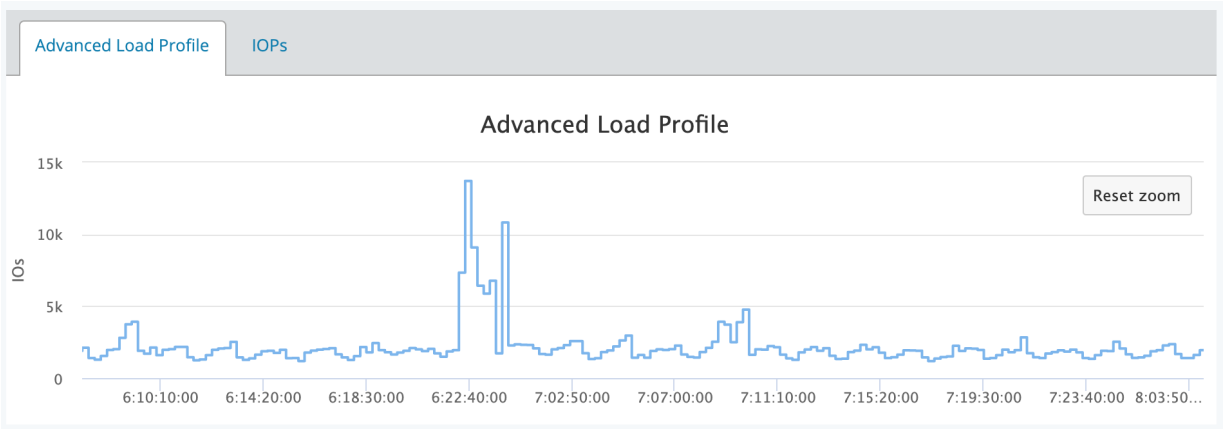
Access Patterns are different for file, object, and block.

For constant workloads, you can specify the percentage distribution of the supported commands.

For temporal workloads, Access Patterns are not available for configuration because their values are preset based on the imported production workload data (via Workload Data Importer).

Advanced Load Profile (Temporal Workloads Only)

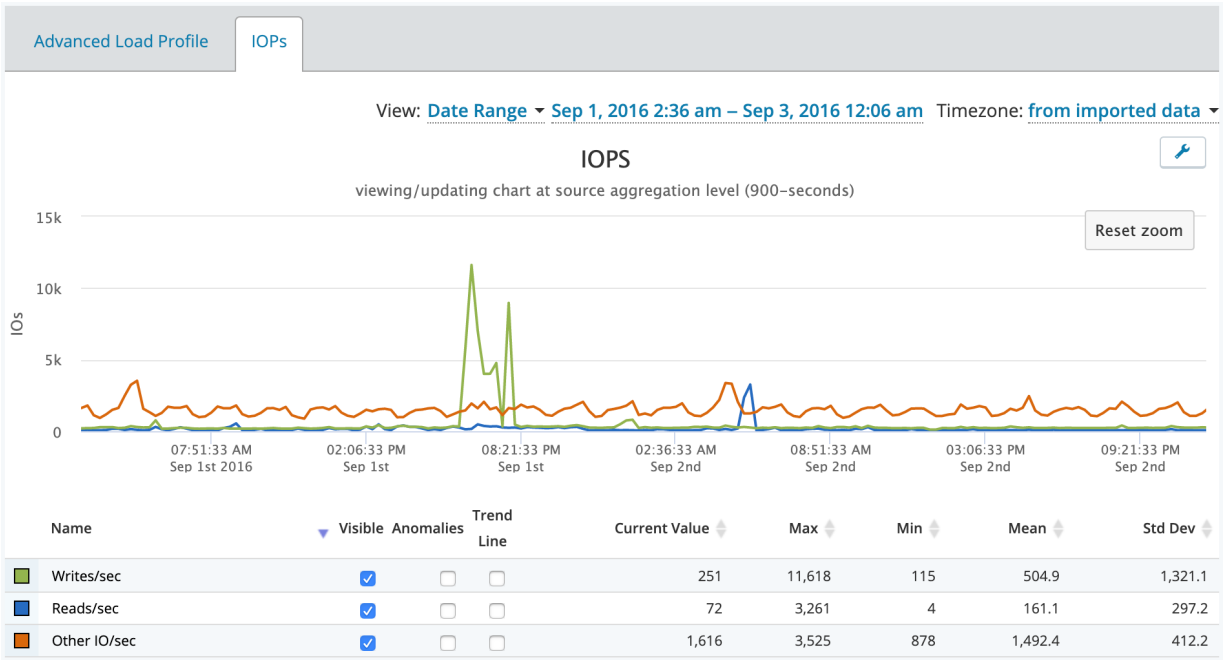
Temporal Workloads automatically determine the distribution and load of the commands. Use the *Advanced Load Profile* to display the level of total IOPs that are performed during each period of the workload.



While the IOPs section gives a more commonly view depiction of IO in which the points are connected, that load profile displays the level that is maintained for a period of time before moving to a new level.

IOPs

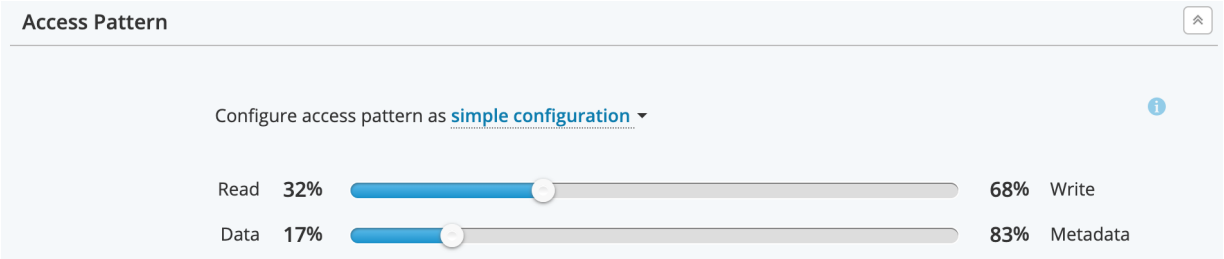
The workload automatically adjusts the number of IOPs for reads and writes attempted according to the chart displayed. If there are significant numbers of other IOs, consider adding those using TDE. The commands issued can be determined by looking at the original production workload that was used to make the workload.



The workload can be limited to a certain time period by selecting the time period of interest before starting the workload. If the duration of the test is longer than the selected time period, the pattern repeats until the duration is reached.

Writes and Reads

Use the *Writes and Reads* section to configure the block sizes, the percentage of random versus sequential behaviors and the number asynchronous I/Os. While there is a separate section for writes and reads, the configuration options for writes and reads are the same. To choose whether a single block size or a distribution is used, select the down arrow next to the current value.



Use the write/read pattern to select the desired percentage of commands that are followed by a command to a subsequent location to the previous command verses from a new random location. Dragging it to the left decreases the random and increases the sequential percentage. The current settings indicate that one in three writes are random

and two in three writes are sequential. This setting is only available for block and file protocols because object protocols do not permit access to a portion of an object. In the case of a file workload that is configured with small file sizes, it might not be possible to achieve significant sequential access. This depends on the ratio of files sizes to block sizes.

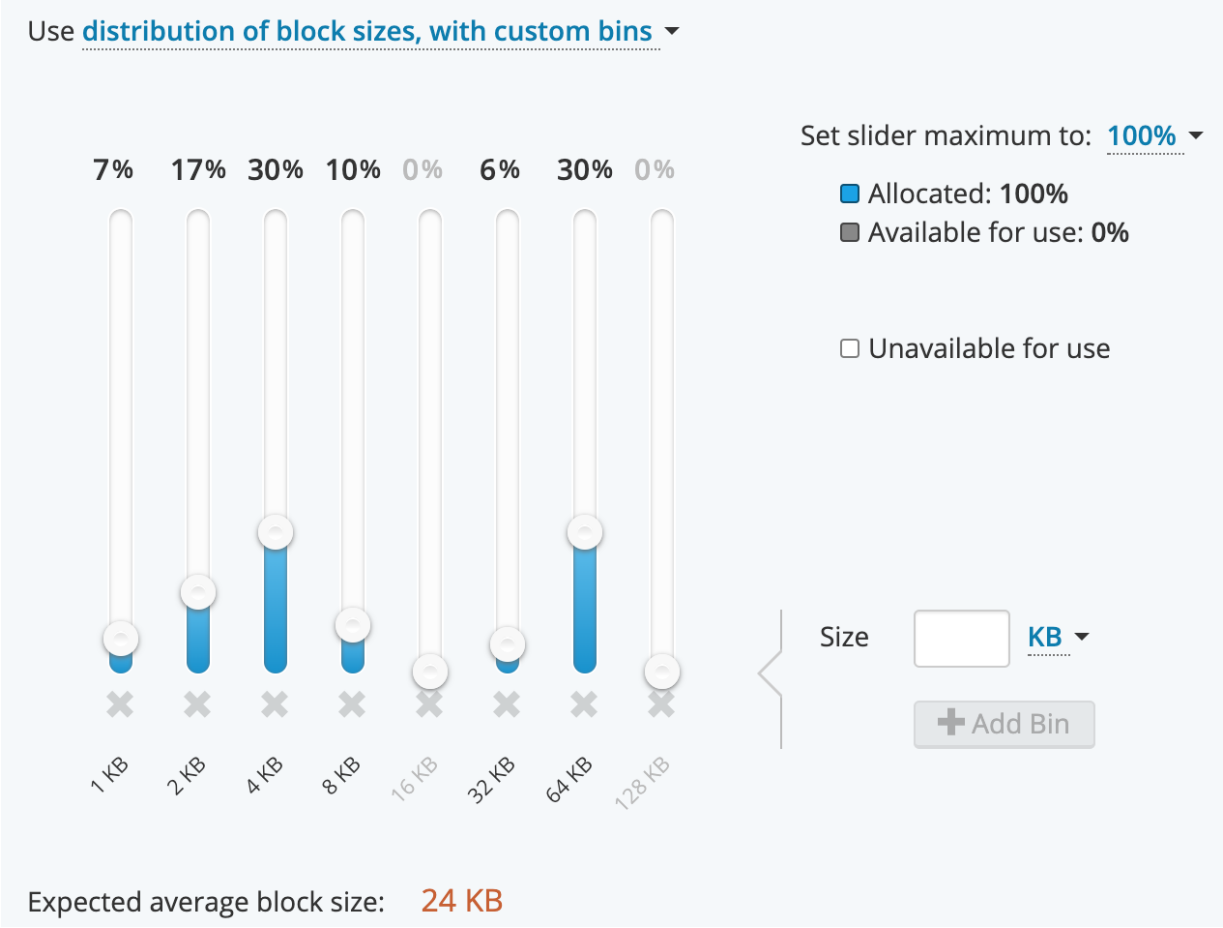
If you want to use the same parameters for reads as for writes, check the check-box in the *Reads* section.

Request/Block Sizes

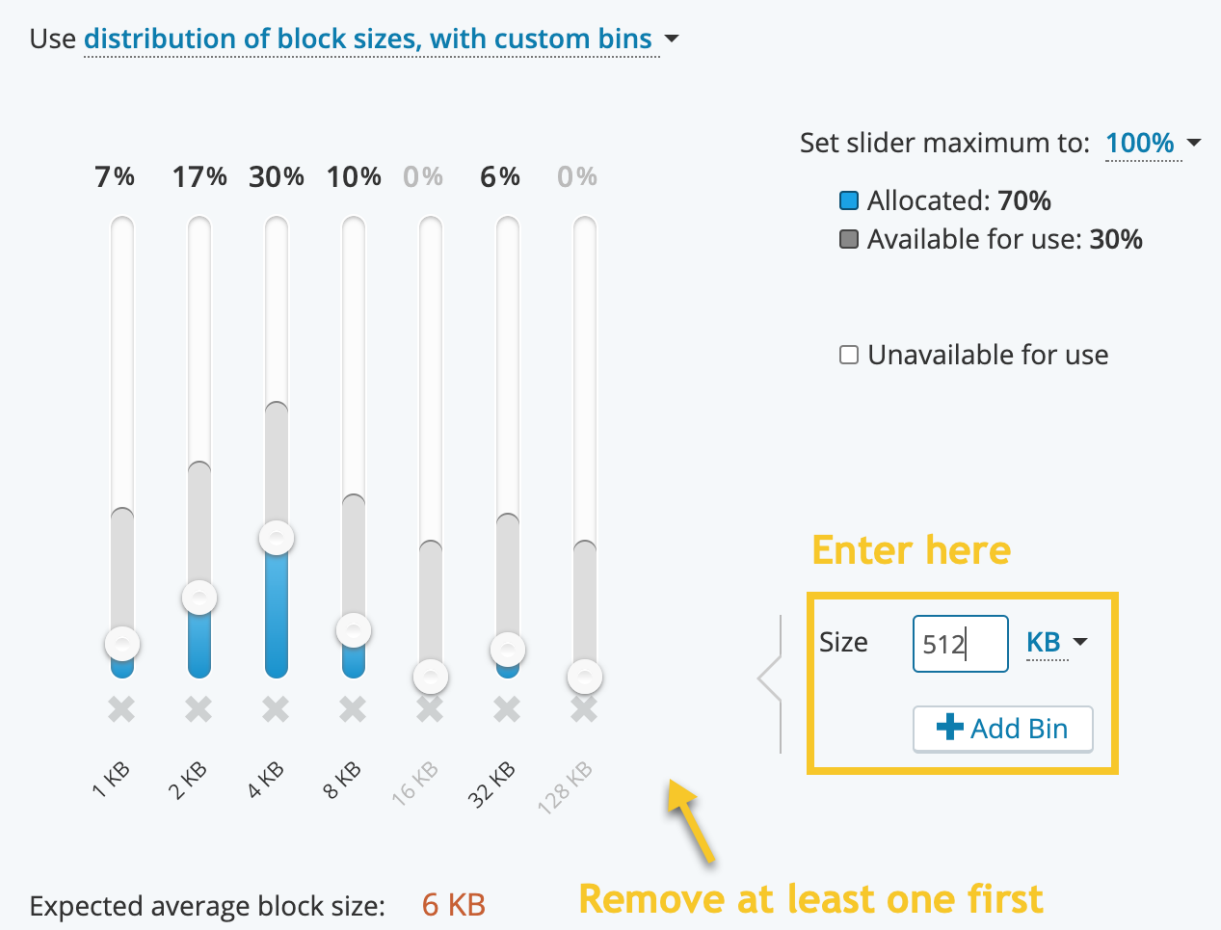
Most protocols break up actual application requests in to chunks of different sizes. Request sizes are sometimes referred to as block sizes. Referring to them as block sizes is misleading as there is a block size (minimum accessible size) associated with the actual parsing of the data on an array. The request sizes used depend on the application, OS, and protocol. Most real world applications use a variety of request sizes though there are some that only use a single request size.

To configure a workload to use a single size, use constant block size. The default value for constant block size is 512 bytes. 1 MB is supported for NFS and 2 MB is supported for SMB.

In addition to constant block size, you can use the bin distribution of block sizes and custom bins. You can use the default or create your own custom bins. A maximum of eight bins are supported.



You can remove the default bins and replace them with your own custom bins by clicking **+Add Bin**. You can specify the block size in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB).

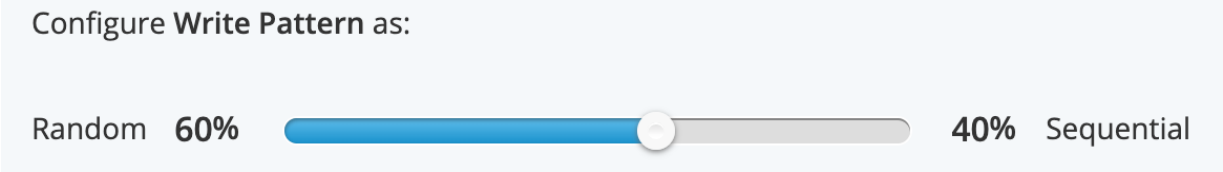


You can change the ratio of size distributions by moving the sliders up to increase the value, or down to decrease the value. You can set the slider maximum by selecting the maximum percentage value the sliders can have in the top righthand corner of the bin distribution section. An average file block size is displayed as often only the average block size in an environment is known or can be calculated.

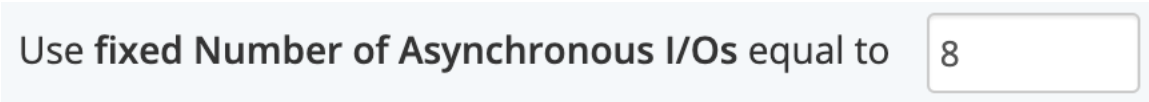
It is import actual typical block sizes are used to achieve an odd average block size rather than using the odd size itself. It is very unusual and not desired for systems to be configured with odd block sizes. Arrays are typically optimized to handle requests that are aligned to the array's block size.

For file protocols the request sizes are typically smaller than for block protocols so different defaults are used depending on the protocol though any custom setting can be configured if the protocol supports it.

In addition to block sizes, you can also specify how much of the read / write operations are performed on sequentially incrementing offsets or random offsets.



Another behavior you can control is the number of Asynchronous I/Os, which at a high level specifies how many read / write requests can be outstanding (i.e., without responses) simultaneously. However, the exact implementation of this setting differs a little bit depending on the protocol, and they are described elsewhere, in topics that explain each workload in more detail. The setting for the number of Asynchronous I/Os is applied to both the sequential and the random requests.

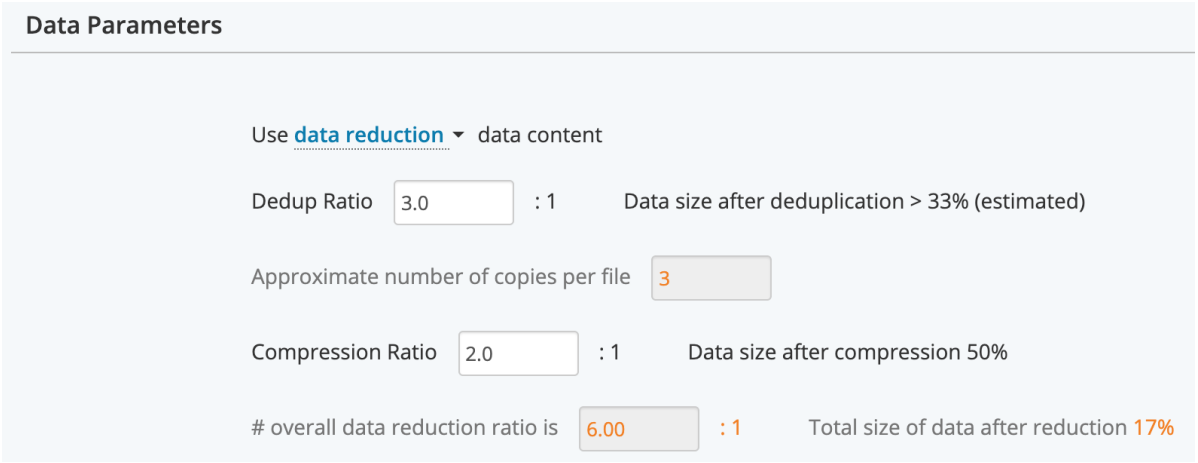


Data Parameters

Data parameters specify the nature of the test data being created and whether it is compressible or reducible. To change the method, select the down arrow next to the current value.

The following types of data are available in the data parameter section:

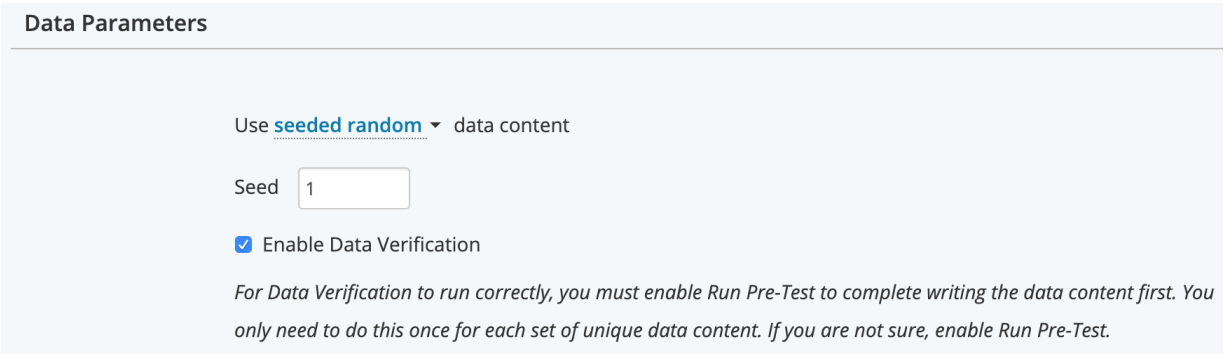
- *Constant* generates the same data pattern repeatedly.
- *Sequential* generates a pattern of sequentially incrementing data.
- *Random* generates a randomly created data pattern.
- *Seeded Random* generates a randomly generated pattern that can be recreated. You can specify the seed to be used.
- *Data Reduction* enables you to define two types of data reduction:
  - *Compression* generates a sequence of data pattern in which a certain percentage of the data pattern is non-compressible (random numbers) and the remaining percentage of the data pattern is compressible (all zeros).
  - *Deduplication* generates a sequence of data pattern that should yield a certain value of deduplication ratio. Refer to the explanation provided for each workload type that supports Deduplication, as the implementation varies.



Not all workloads support the deduplication setting of data reduction. For those that support it, the implementation varies due to the nature of different deduplication methods across different technologies. For example, data deduplication in block is often performed at the block level, and deduplication of files is often performed at the file level. The protocol specific parts of this section provide more information about the protocol's deduplication design.

Data Verification

Certain data content types support Data Verification, which instructs WorkloadWisdom to check for data integrity when reading the data from the system under test. In order for Data Verification to run, you must seed the system under test with a repeatable data pattern first, and then when you run the workload, you must have at least 1% Read operations, and must specify the same data pattern to be expected when reading. Any change to the written data on the system under test between seeding the data patterns and executing data verification may produce invalid results, including change in file system structure.



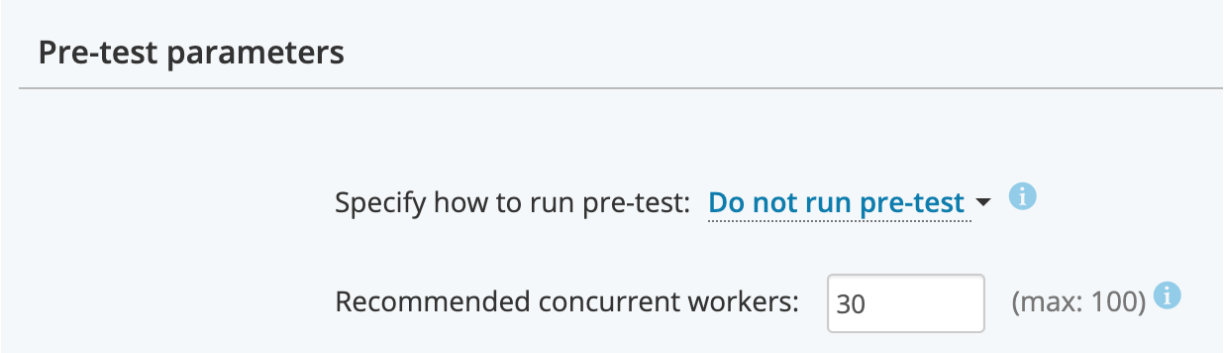
The following data content types support Data Verification:

- Sequential
- Seeded Random

If you are not sure whether the previously written data pattern on the system under test has changed, simply enable Pre-Test before you run the workload. This will ensure that the file system and the data patterns will be recreated before data verification is performed.

Pre-Test

A pre-test sets up the testing environment and normally includes preparing the file or object system so the workload can be run against it. The pre-test section specifies when the pre-test runs. Different protocols also have some different options available. For block workloads, it is recommended to set *Do not run pre-test*, and use the preconditioning workload instead. See Preconditioning [57].



NOTE
Pre-Test is not available for FC-NVMe workloads.

Load Properties

Constant Protocol Workloads

The load properties section defines the type of load, number of concurrent workers, rampup, and rampdown durations.

- Load: Amount of IOs sent to the System Under Test. Three types of load are available:

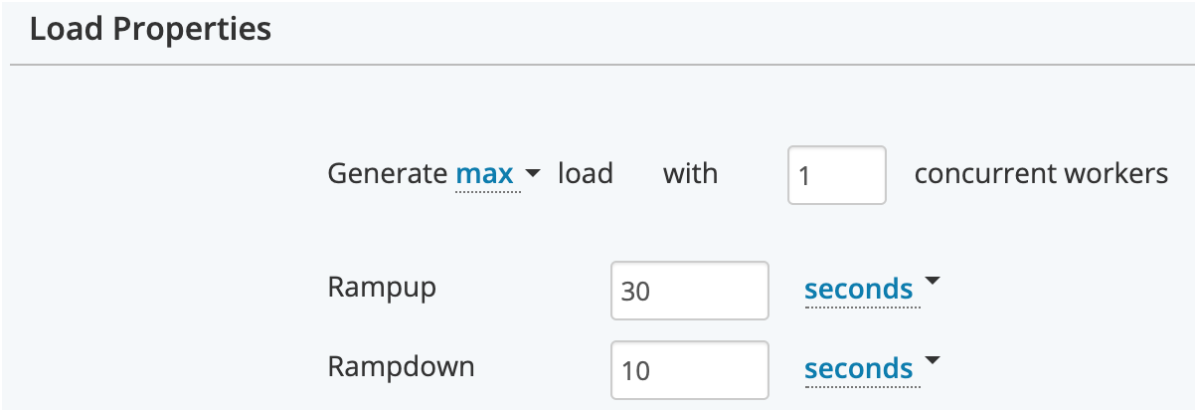
  - Max: For the configured workload, generate the maximum load possible with the number of concurrent workers. The “maximum load possible” can vary even for the same exact workload and same exact load settings, depending on Test Bed selected and the state of the System Under Test (including lab network).
  - Throughput: For the configured workload, generate up to the specified amount of throughput with the number of concurrent workers.
  - Actions per Second (IOPS): For the configured workload, generate up to the specified amount of Actions per Second (generally known as and generally equivalent to IOPS) with the number of concurrent workers.
- Worker: An entity that executes the configured workload from start to finish. Depending on how the workload is configured, a worker can represent an application, or an application’s process, or a VM, or a host, or a collection of hosts behind an edge device, or more. Essentially, whatever you are looking to simulate with the configured workload, one worker will run one instance of that workload from start to finish. N concurrent workers means N instances of that workload will run concurrently. When a worker finishes executing the configured workload from start to finish, and if the workload run duration has not run out, then a new worker will run the configured workload from start to finish. Therefore, it is possible to see by the end of a test run that multiple workers have ran the workload, even though you have specified set concurrent workers to 1. It is important that enough concurrent workers are specified to access the number of LUNs/Volumes with the number of desired users and IP addresses specified in the test bed. Each worker only accesses one LUN or volume and only uses one user or IP address.
  - For IP-based workloads, “start” is usually observable on the wire as it is usually marked by a new TCP connection attempt, and “close” is usually marked with closing the new TCP connection.

    NOTE

    NFS can have multiple new TCP connections back to back at the start of a workload from each worker. For example, an NFSv3 worker will first open and close new TCP connections for Portmapper and Mount, before opening a new TCP connection for NFSv3.

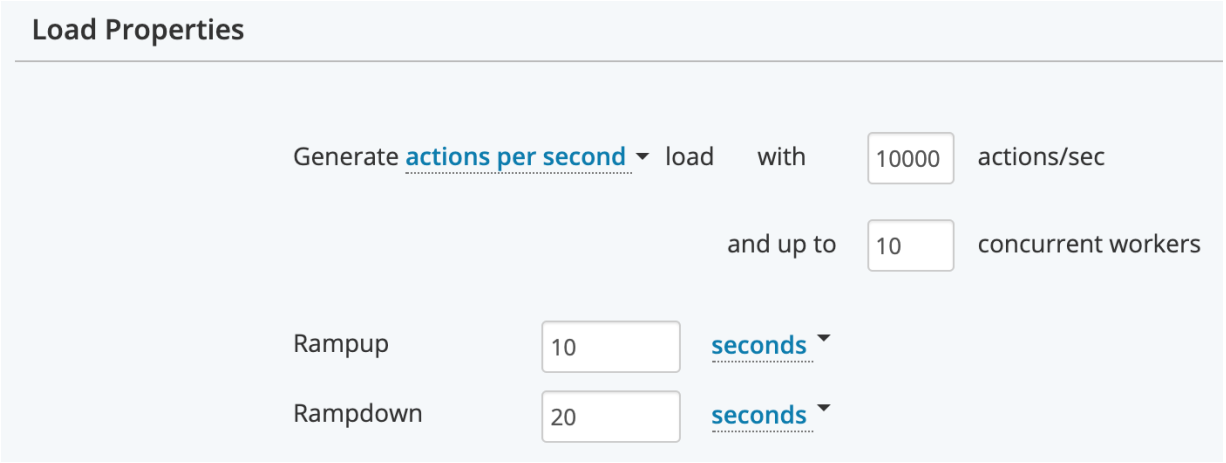
  - For FC-based workloads, “start” is usually observable on the wire as it is usually marked by a Test Unit Ready command, but “close” is not usually easily observable on the wire since there is no “SCSI layer connection”.
- Rampup and Rampdown: *Rampup* enables you to specify the time taken to reach the load parameters specified and *Rampdown* specifies the time taken to reduce the specified load parameters to zero. Editing the text boxes can change the *Rampup* and *Rampdown* values. To change the time unit from the default of seconds to minutes, hours or days, click the down arrow and select a new value.



Rampup and Rampdown: *Rampup* enables you to specify the time taken to reach the load parameters specified and *Rampdown* specifies the time taken to reduce the specified load parameters to zero. Editing the text boxes can change the *Rampup* and *Rampdown* values. To change the time unit from the default of seconds to minutes, hours or days, click the down arrow and select a new value.

This example generates the maximum load possible for the configured workload with a single concurrent worker. Rampup and Rampdown do not have effect in this example because the load type is “max” and concurrent workers = 1.

To use a different load type, select the down arrow next to the current value and choose a different load measurement method.

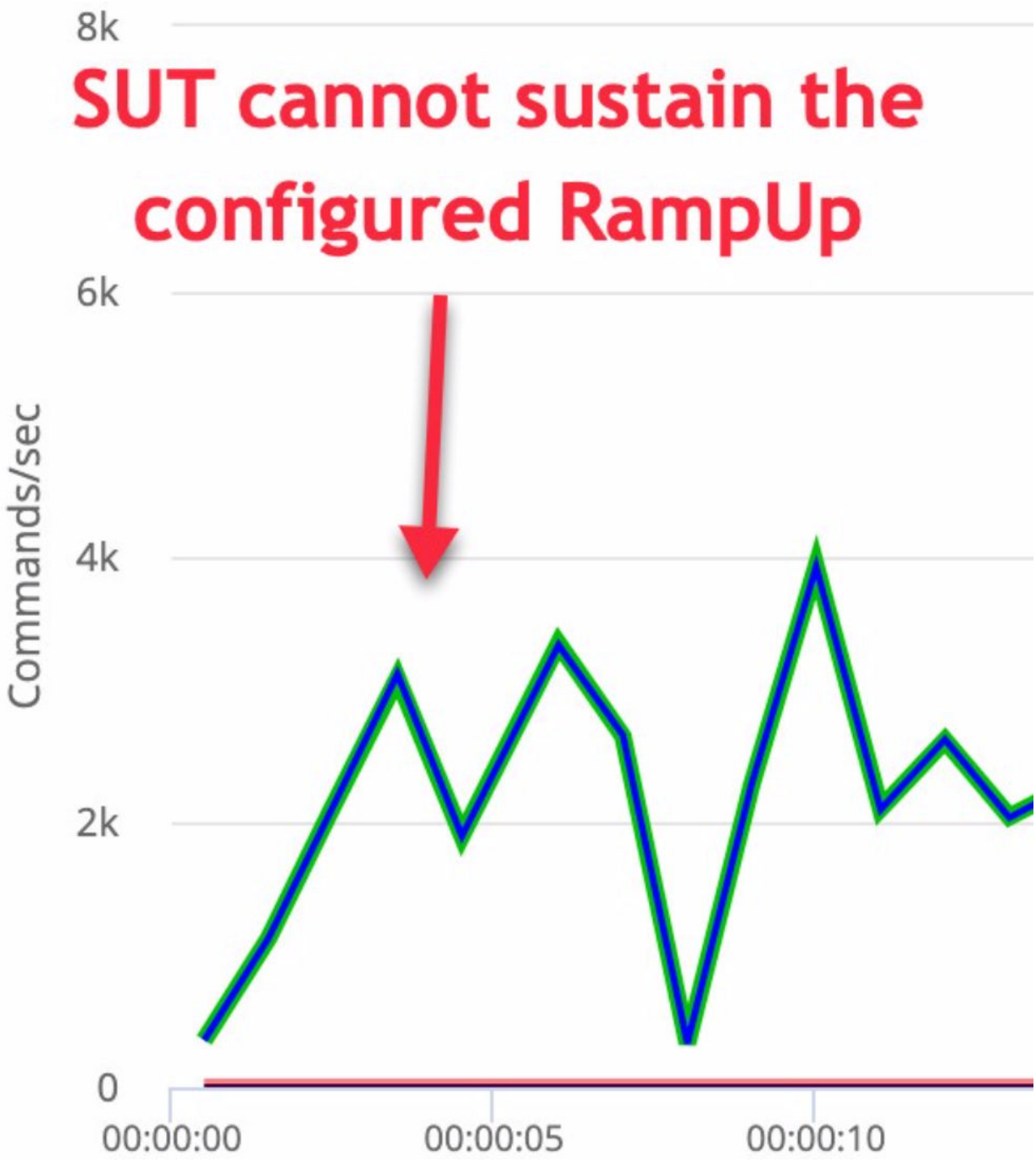


In this example, WorkloadWisdom will attempt to generate up to 10,000 new Actions per Second (essentially IOPS) across 10 concurrent workers, which means 1,000 new IOPS per concurrent worker. A total of 10 seconds will be used to evenly ramp up to the 10,000 new IOPS from the 10 concurrent workers starting with 0 IOPS, which means every second, an additional 1,000 new IOPS will be attempted from the 10 concurrent workers.

During the test run, the Results Dashboard will show the following ramp for this configuration:

IOPS and Workers ramp up behaviors for the Load Properties settings example

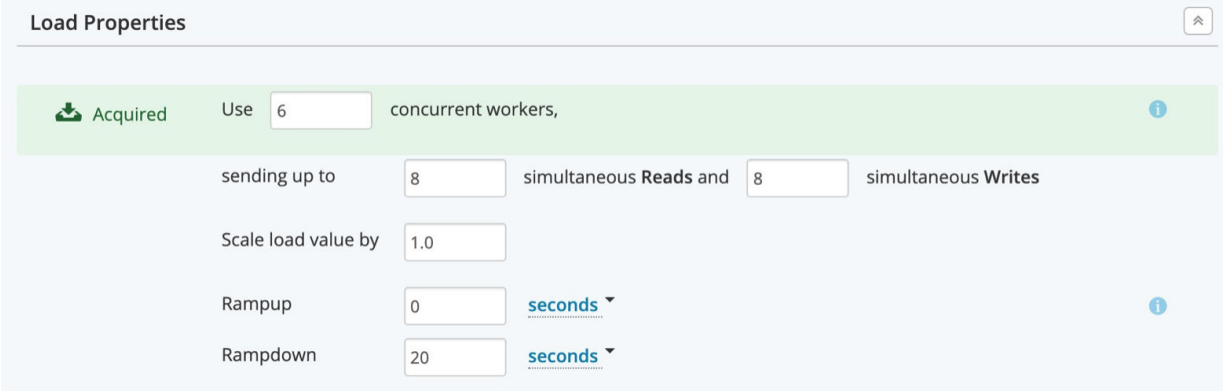
However, whether a perfect RampUp can be achieved or not also depends on the SUT's performance, as well as the Test Bed used. For example, using the same configuration, here is an example showing that the SUT cannot sustain the RampUp, so it is important to set a RampUp value that makes sense for your test environment.



The concept is similar for Throughput. Instead of using Actions per Second as the load target, it uses Throughput as the load target.

Temporal Protocol Workloads

The temporality aspect of the load is automatically generated based on the imported production workload data (via Workload Data Importer). What you can set is a set of related parameters that essentially allow you to “scale up” and “scale down” the imported production workload data.



In this example:

- **Concurrent workers**: Acquired, 6. The value 6 is automatically set based on the resulting analysis of the imported production workload data. To understand where this comes from, you can review the Workload Components View tab from the output of your imported Production Workload Data csv file (using the Workload Data Importer feature). You can change the default value, and when you do, the **Acquired** label will change to **Reset to Acquired** to let you know that the current value is no longer the default value.
- **Simultaneous Reads / Writes**: See **Asynchronous I/Os** section.
- **Scale load value by**: This changes the amplitude of the temporality aspect of the generated load. For example, if the default temporal load profile has 1,000 IOPS in the first minute, 1,500 IOPS in the second minute, and 1,100 IOPS in the third minute, and you set this value to 2.0, then the generated temporal load profile will have 2,000 IOPS in the first minute, 3,000 IOPS in the second minute, and 2,200 IOPS in the third minute.

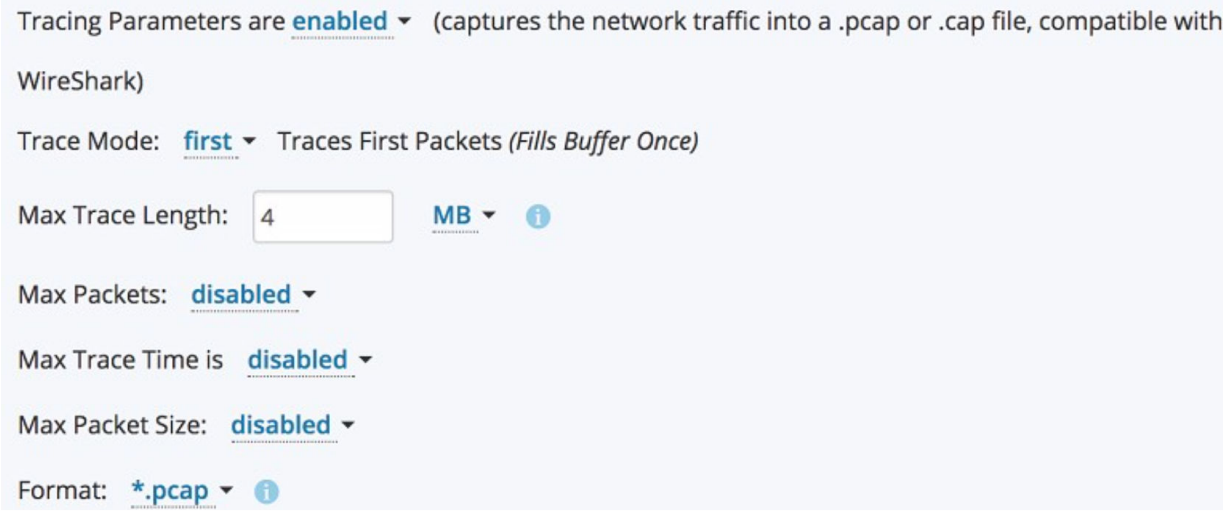
It is important to note that depending on the default temporal load and the test environment you have, it might not be possible to generate the load you specify with just one Workload Generator Port. For example, if the default temporal load profile has 100,000 IOPS and the workload already contains some large block sizes, then it simply might not be possible to generate 1,000,000 IOPS if you set the scale factor to 10.0. Conversely, it is also possible that there is not enough resource to recreate the expected load. For example, if you selected a time slice of a temporal workload that is doing 200 IOPS, and then you set the concurrent workers = 100, simultaneous Reads / Writes = 1, and scale load value = 100, this would require each worker to send 200 IOPS. However,

with this setting, each worker can only send 1 Read and 1 Write simultaneously, which may be insufficient.

A general best practice is to allow for long enough workload run time, and scale up / down proportionally across concurrent workers, simultaneous Reads / Writes, and scale load values.

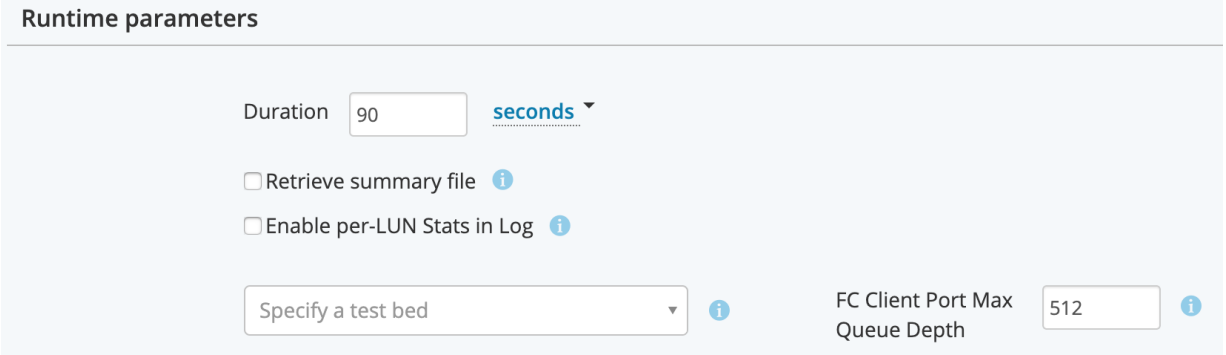
Tracing Parameters

Use tracing parameters to specify whether to capture a pcap or cap network traffic trace and the values to be used to configure the pcap or cap capture size. You can specify the size of the pcap or cap file based on file size, number of bytes, packets and duration in time.



Runtime Parameters

Use runtime parameters to specify the workload test duration, and optionally retrieve the summary file or per-LUN Stats in a Fibre Channel or iSCSI workload test. The summary file is used for diagnostic purposes and should be left unchecked unless requested by Support. The per-LUN Stats option captures LUN usage into the log. These are available for download after a workload test run has stopped.

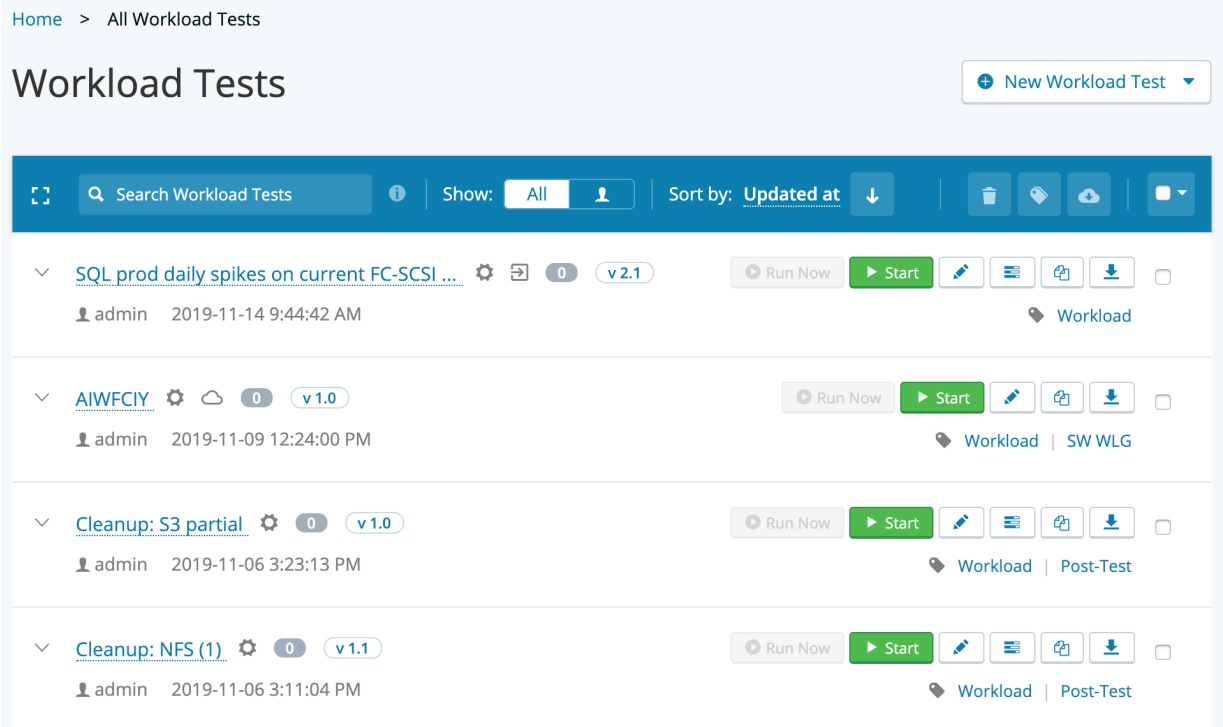


You can specify the test bed to run the workload against by selecting the pre-filtered list of test beds that support that protocol. Once a test bed is selected it is displayed in the workload.

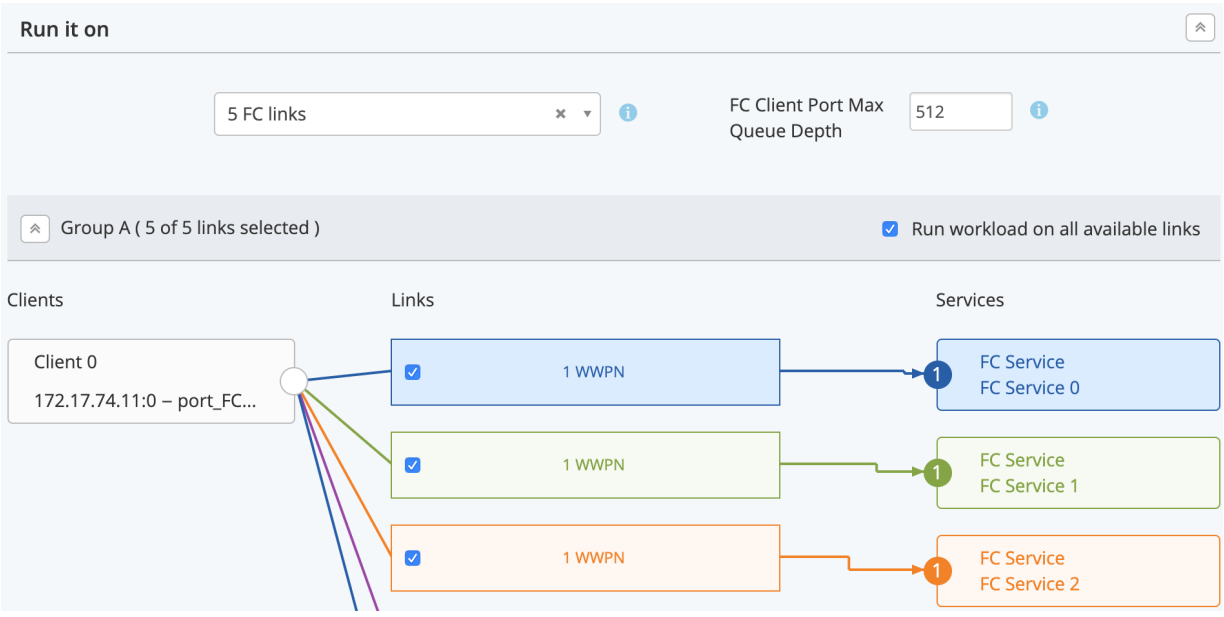
Starting a Workload Test

You can start your workload from the *Workloads Tests* or *Run it on screen*. To start your workload, click **Start** from either screen, or click **Run Now** from the *Workload Tests* screen if the button is activated.

Workload Tests screen:



Run it on screen of a Workload Test Start page.



The **Run Now** button is only activated if the Workload Test or Workload Suite or Iteration Suite has all required parameters configured with valid configurations, and that the selected Test Bed does not contain Ports that are currently in use by others. The difference

between **Run Now** and **Start** is that the Run Now option will start the Test immediately without any review or runtime configuration change, whereas the Start option will take you to the Start page for optional review and optional runtime configuration change before starting.

There can still be corner cases in which the Run Now button is activated, but when you click on it, you encounter an error that the Test Bed is busy. This is because the Test Bed availability status is checked when the Workload Tests page is loaded. After you load the page but before clicking the Run Now button, it is possible that another User has started a Workload Test using the same Test Bed. Other corner cases exist as well. As a best practice, refresh the page right before you intend to click the Run Now button.

Start Parameters

In addition to being able to review the parameters available when editing the workload, there are a few additional options when starting a workload test.

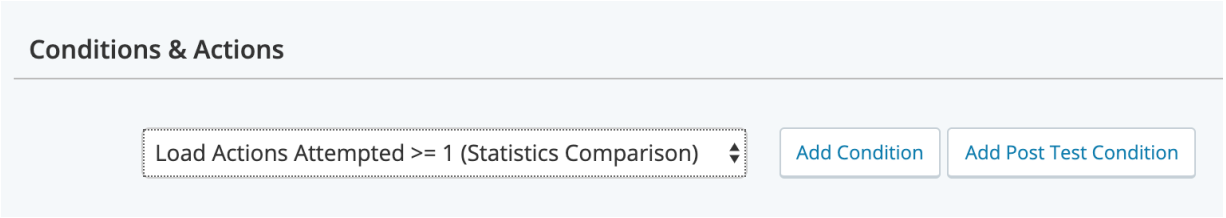
Description

The *Description* at a test's start time is the description for the test run and results. You can use the description to specify the purpose of a test run.

Conditions and Actions

At start time it is possible to configure conditions. You can use conditions to mark a test as passed or failed or to stop a test or series of tests. For more information on conditions, see Conditions [230]

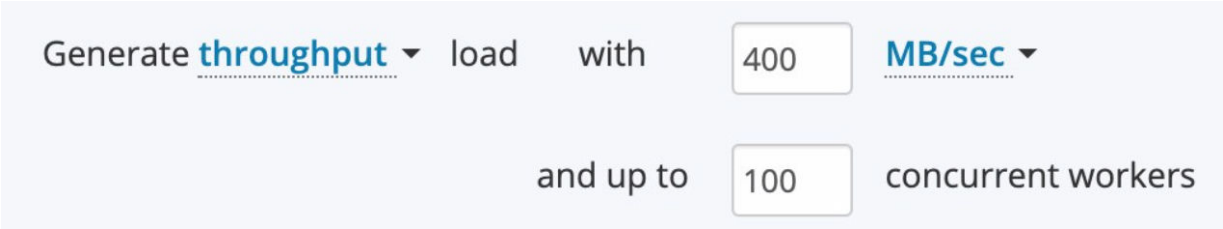
Click the up/down arrows to expand the list of conditions.



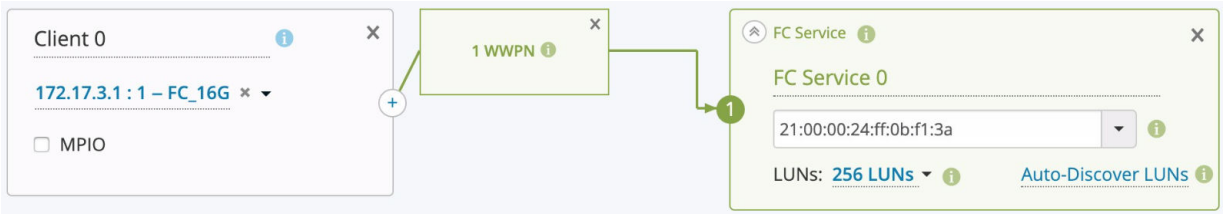
Load Properties, Test Bed Behaviors, and Best Practices

Load Properties (both the load settings and the worker settings) are applied on a per Test Bed Link basis, and each worker works on one LUN / Share at a given time. It is best to think of “load” and “worker” separately. Load is more about controlling the amount of traffic, whereas worker is more about controlling the access to the LUNs / Shares, and that load is divided across the concurrent workers, again on a per Link basis.

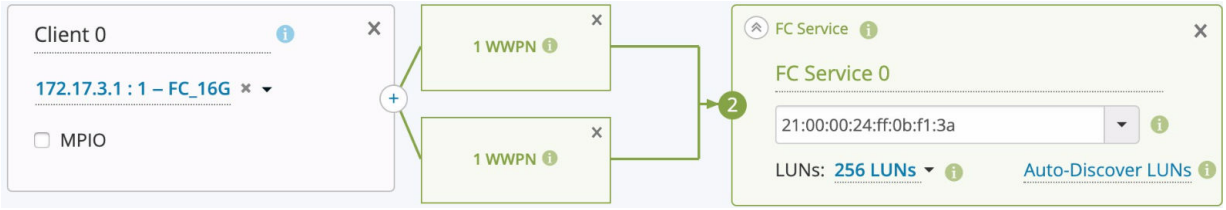
Let’s say in a workload you have the following Load Properties settings:



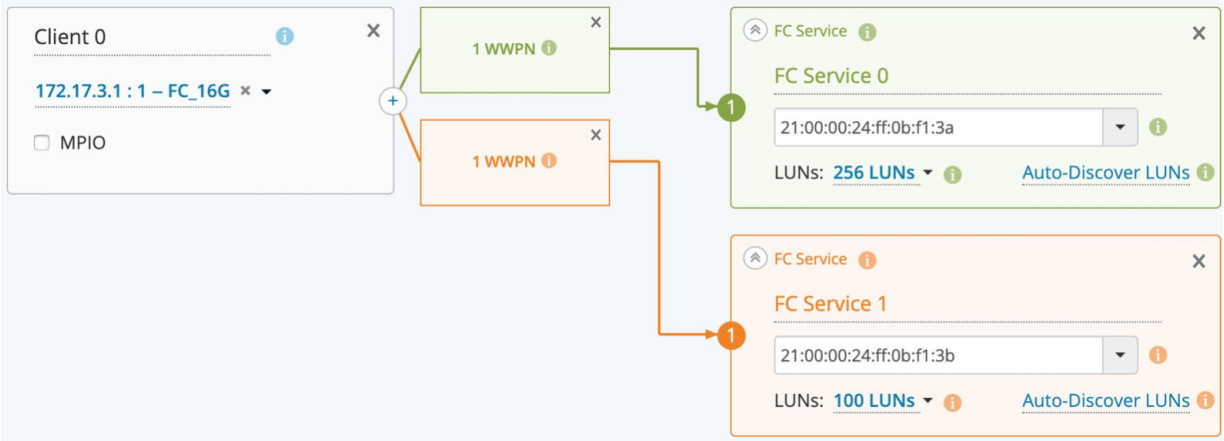
Here are the behaviors for each of the following Test Bed examples.



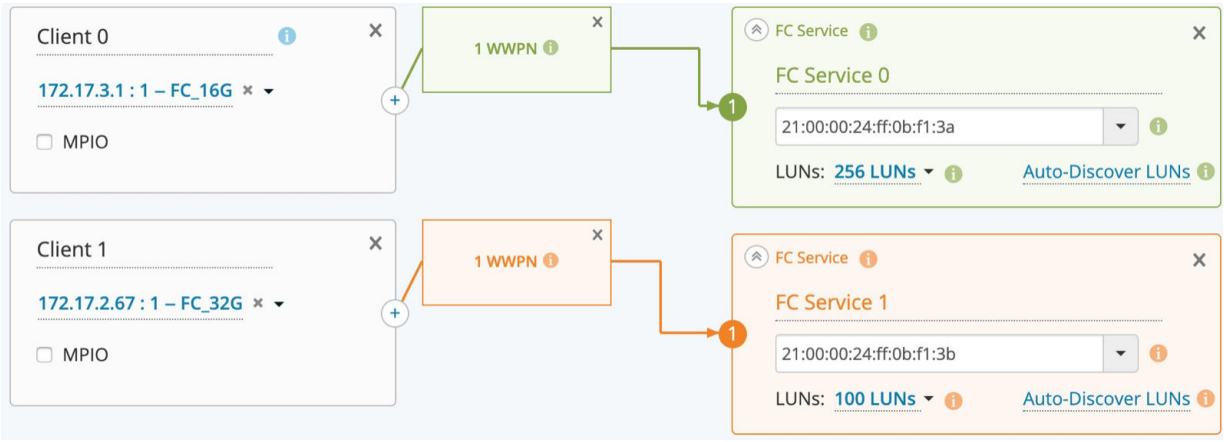
In the above example, a total of 400MB/s of throughput will be generated, and a total of 100 LUNs will be accessed concurrently by 100 workers (1 worker on 1 LUN at a given time).



In the above example, a total of 800MB/s of throughput will be generated, and a total of 100 LUNs will be accessed concurrently by 200 workers (2 workers on the same 1 LUN at a given time). If you want 256 LUNs to be accessed simultaneously, you will need to set concurrent workers to at least 256.



In the above example, a total of 800MB/s of throughput will be generated, and a total of 100 LUNs from the first FC Service will be accessed concurrently by 100 workers, and a total of 100 LUNs from the second FC Service will be accessed concurrently by 100 workers.



In the above example, a total of 800MB/s of throughput will be generated (400MB/s per port), and a total of 100 LUNs from the first FC Service will be accessed concurrently by 100 workers, and a total of 100 LUNs from the second FC Service will be accessed concurrently by 100 workers.

So if you want to make sure that all LUNs / Shares are getting accessed simultaneously at any given time during a workload run, then you should set the concurrent workers to at least equal to the number of unique LUNs / Shares you have in the Test Bed. For example, if you have 1,000 LUNs / Shares across a number of Targets, a total of 4 external ports (i.e. those facing the clients) on the SUT connected to the Workload Generator over 4 Test Bed Links, and you want to generate a total of 400MB/s of load to the SUT, then you should set the concurrent workers to at least 1,000, and the load to 100MB/s.

Note that the “4 Test Bed Links” can be connected to 1 to 4 Workload Generator Ports, because you can have multiple logical Test Bed Links per Port. For example, you may have a non-contiguous set of client IP addresses you can use for testing purposes, but you don’t have a need to require multiple physical Workload Generator Ports, then you can create multiple Test Bed Links between a Workload Generator Port and the SUT, where each Link has a unique block of client IP addresses.

For FC, using multiple Test Bed Links between a single Workload Generator Port and a single FC Service is not common (or necessary), but is possibly required if you enable NPIV and have some strict zoning configurations.

For FC Test Beds, you might have MPIO enabled. Refer to the Block Protocol SCSI Workloads section and the FC Service (Test Bed) section for more information.

File Protocol Workloads Details

File protocol workloads emulate the process of accessing one or more files across a file system on a pool of Network Attached Storage (NAS). The behavior is dependent on the metadata operations, block sizes, and other aspects negotiated by the client and NAS server. There are a number of common elements in a file workload, such as access pattern, file system, and writes and reads. Only the command or metadata differ across file protocol workloads.

For a list of unique commands for each protocol, see the section of this user guide for that particular type of workload.

WorkloadWisdom provides out of the box models for:

- Network File System (NFS), Version 3
- Network File System (NFS), Version 3 temporal workload
- Network File System (NFS), Version 4
- Network File System (NFS), Version 4 temporal workload
- Network File System (NFS), Version 4.1
- Server Message Block (CIFS/SMB 1.0, SMB 2.x and SMB 3.x)
- Server Message Block (SMB 2.x) temporal workload

You can only create temporal workloads using the Workload Data Importer, or by importing a temporal workload file.

Workload Design and Behaviors

The File Workloads (NFS, SMB) use a probability-based design to generate the user specified distribution of commands and load. Factors such as SUT performance, Test Bed, errors, and also runtime duration can cause the actual distribution and load to be very different from the expected distribution and load.

For example, if READ/WRITE ratio = 71%/29%, it means that READ probability is 0.71 and WRITE probability is 0.29 during the workload run. The “larger” your test environment is, the better a longer workload run duration should be configured as from a probability perspective, it is more likely that the actual observation will match the configuration. Some metadata commands have dependencies with each other, such as CLOSE and OPEN commands, READDIR and LOOKUP commands, so it is recommended that you specify “realistic” distributions for these types of commands.

Throughout a workload test run, if you have a single concurrent worker, then it will choose to perform a set of operations from start to finish based on probability on one pass, and then on the next pass it may perform a different set of operations from the previous pass. For example, in the first pass the worker may be doing some LOOKUP and OPENs and sequential Writes on file X, and on the next pass it may be doing some more LOOKUPS and CLOSEs and random Reads on a different file Y. When you have multiple concurrent workers, each worker goes through its own “pass”, so you have many different workers doing different things on different files on the SUT. However, throughout the workload test run, the configured operations will eventually be performed with sufficient runtime duration, in the absence of unexpected errors or performance issues.

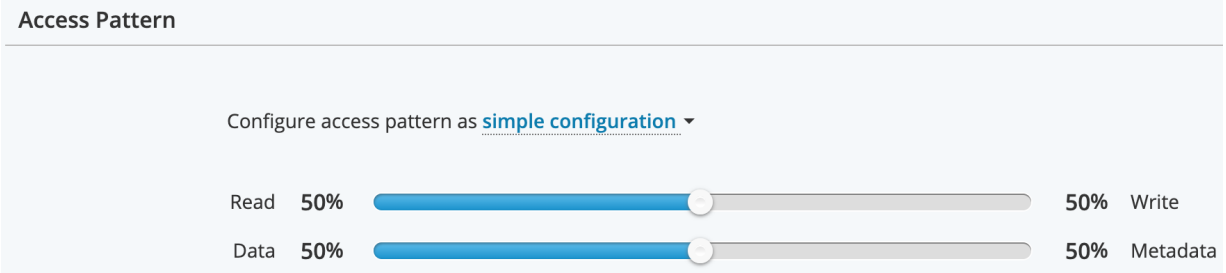
While the protocol commands and some backend mechanisms differ across protocols, the “experience” the SUT goes through is largely similar across the file protocols. For example, SMB workloads use a backend Loop-Thread mechanism, whereas NFS does not, but from the SUT’s perspective, it still sees similar behavior in that many different clients are issuing a variety of metadata and data operations across its shares, folders, and files at the same time.

Common to All File Protocol Workloads

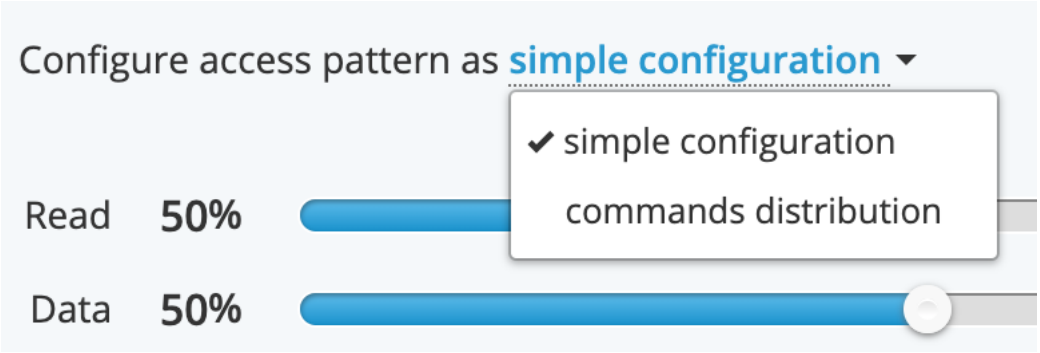
The following sections are common to all file protocol workloads:

Access Pattern

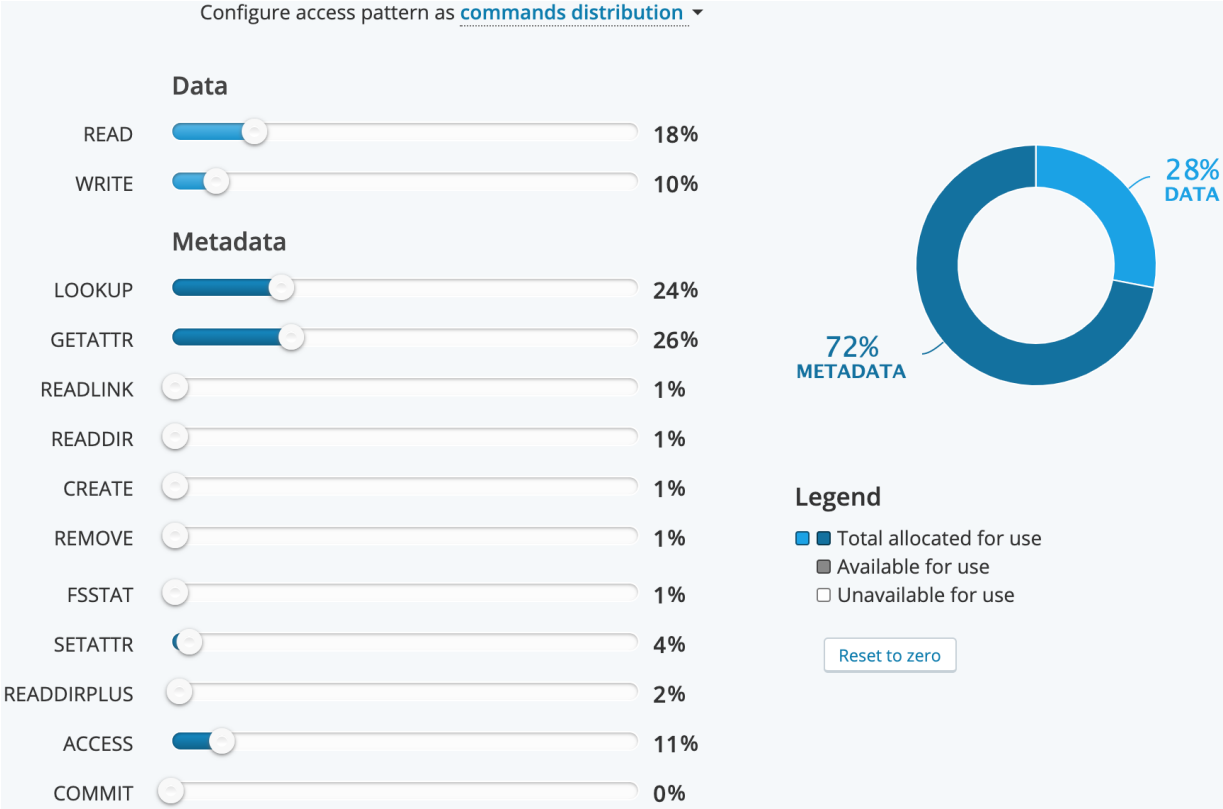
The access pattern can be specified in two ways, a simple configuration (below) that specifies the read/write ratio and data/metadata ratio, or as a specific command distribution. The simple configuration is common to all file workloads, the command distribution is specific to each protocol.



To change the ratio, move the slider to reflect the desired percentage value. If you move the slider to the left for the read/write percentage the read percentage decreases.



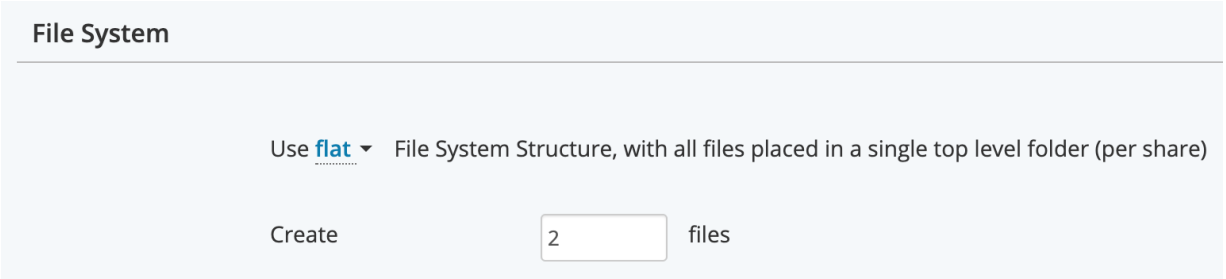
Simple Configuration is the default. To change to specific command distribution, which gives you the ability to individually control each supported command specific to each protocol, select the down arrow and select Commands Distribution. The sliders change to:



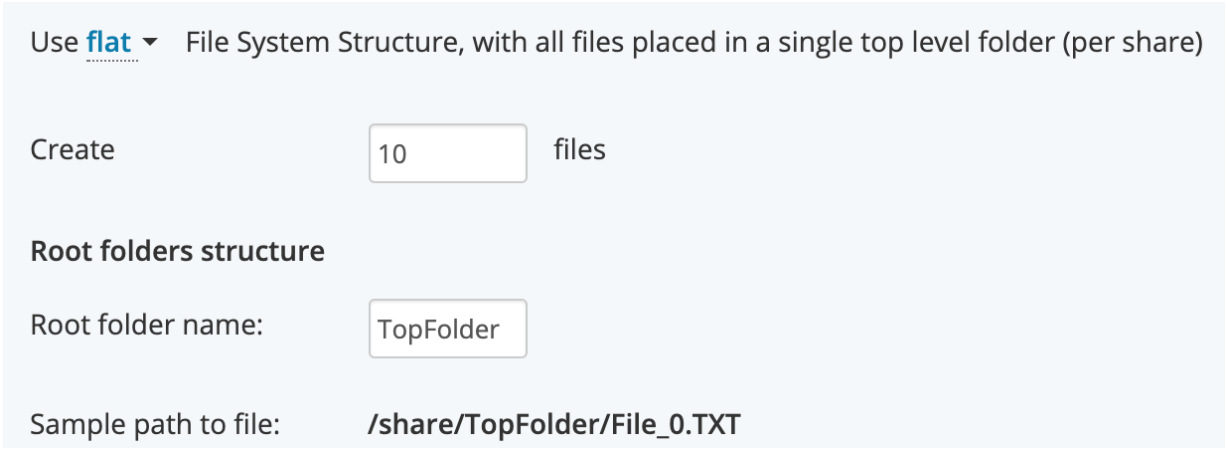
File System

A file protocol workload specifies the workload that it runs against.

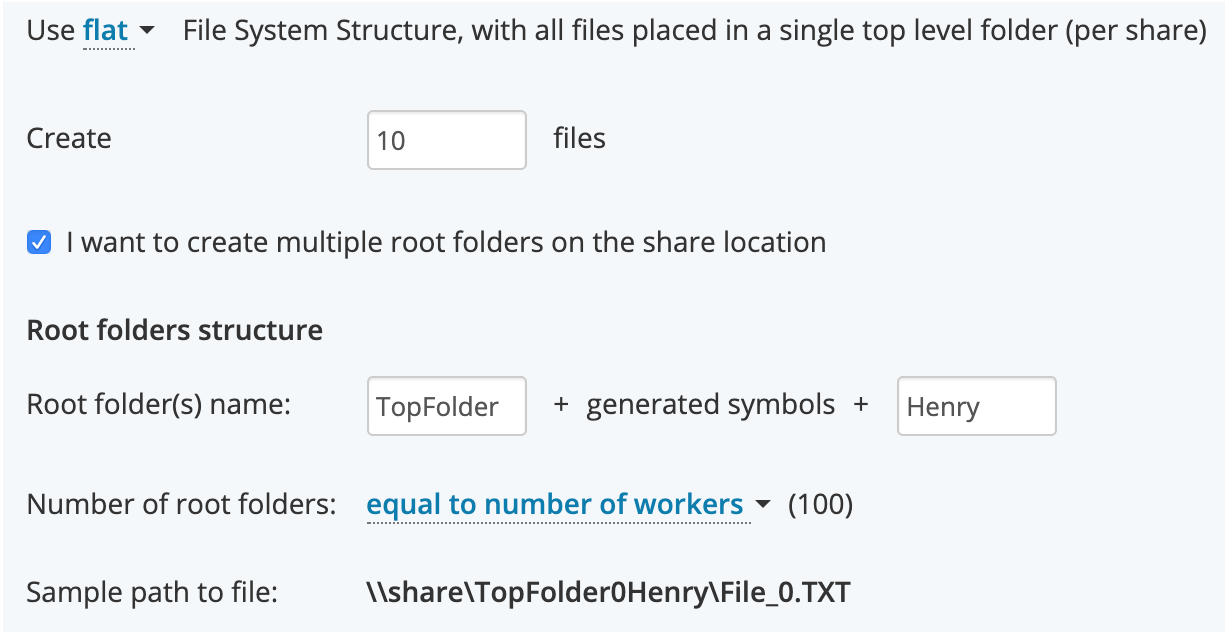
Use the file system section to create a file system based on a flat or tree hierarchy. To change the value, select the down arrow next the current value.



Use *flat hierarchy* to specify the number of files to be created in a single top-level root folder defined by the **Root folder(s) name**. If you leave it blank, the files are created on the default share specified in the test bed, without any folders.



If you want multiple top-level root folders on the share, select the checkbox **I want to create multiple root folders on the share location**, and you will be presented with additional options to specify the **Number of root folders** (all at the same top level on the share, not nested), as well as a **Postfix** to further make it easier to identify the folders and files created from this Workload Test.

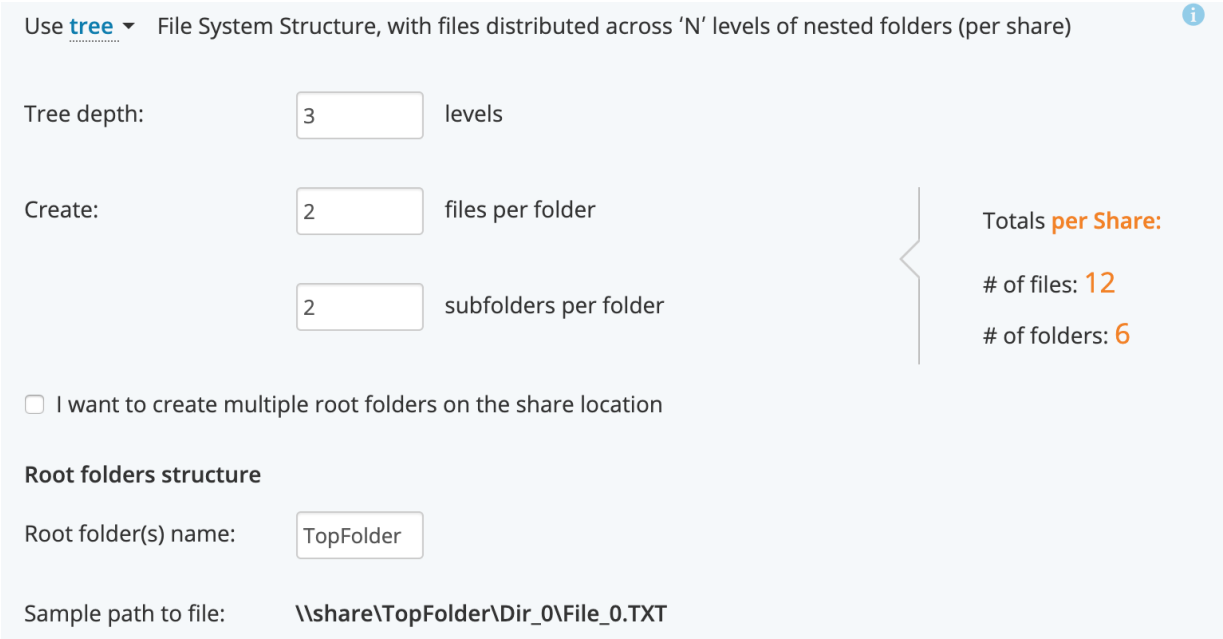


NOTE

Multiple top-level root folders is only available for SMB protocol workloads.

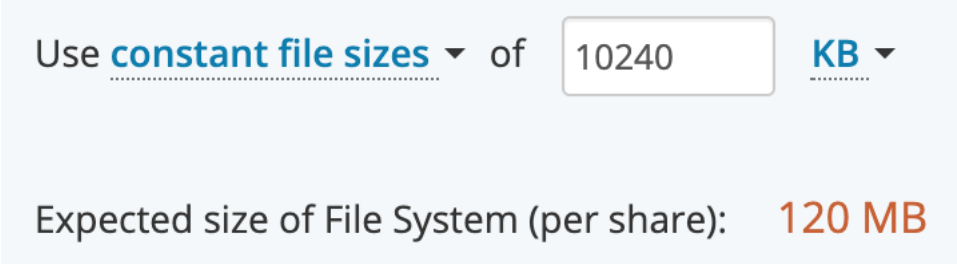
Finally, the **Sample path to file** provides a read-only indication of how your files will be created based on current configurations.

Use *tree hierarchy* to specify the depth (levels) and the breath of the file system (subfolders per folder), and the number of files per folder. The total folders and files are displayed automatically as you modify any of the values. Like the flat hierarchy, you can specify a specific share or leave it blank and it runs against the default share specified in the test bed.



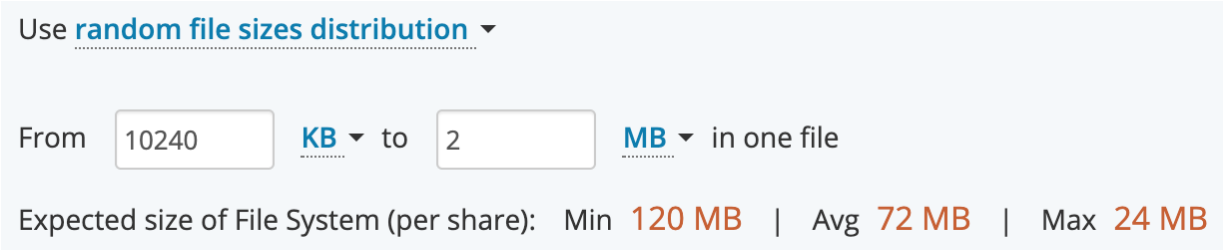
In addition to specifying the file system structure, you can specify the size of the files in the file system using either a constant or fixed size, a random distribution of file sizes or specify a distribution based on set file size bins. The expected size of the file system or range of possible sizes are given based on the file system information above and the file size information.

You can specify *constant file sizes* in bytes, kilobytes (KB), megabytes (MB) or gigabytes (GB).

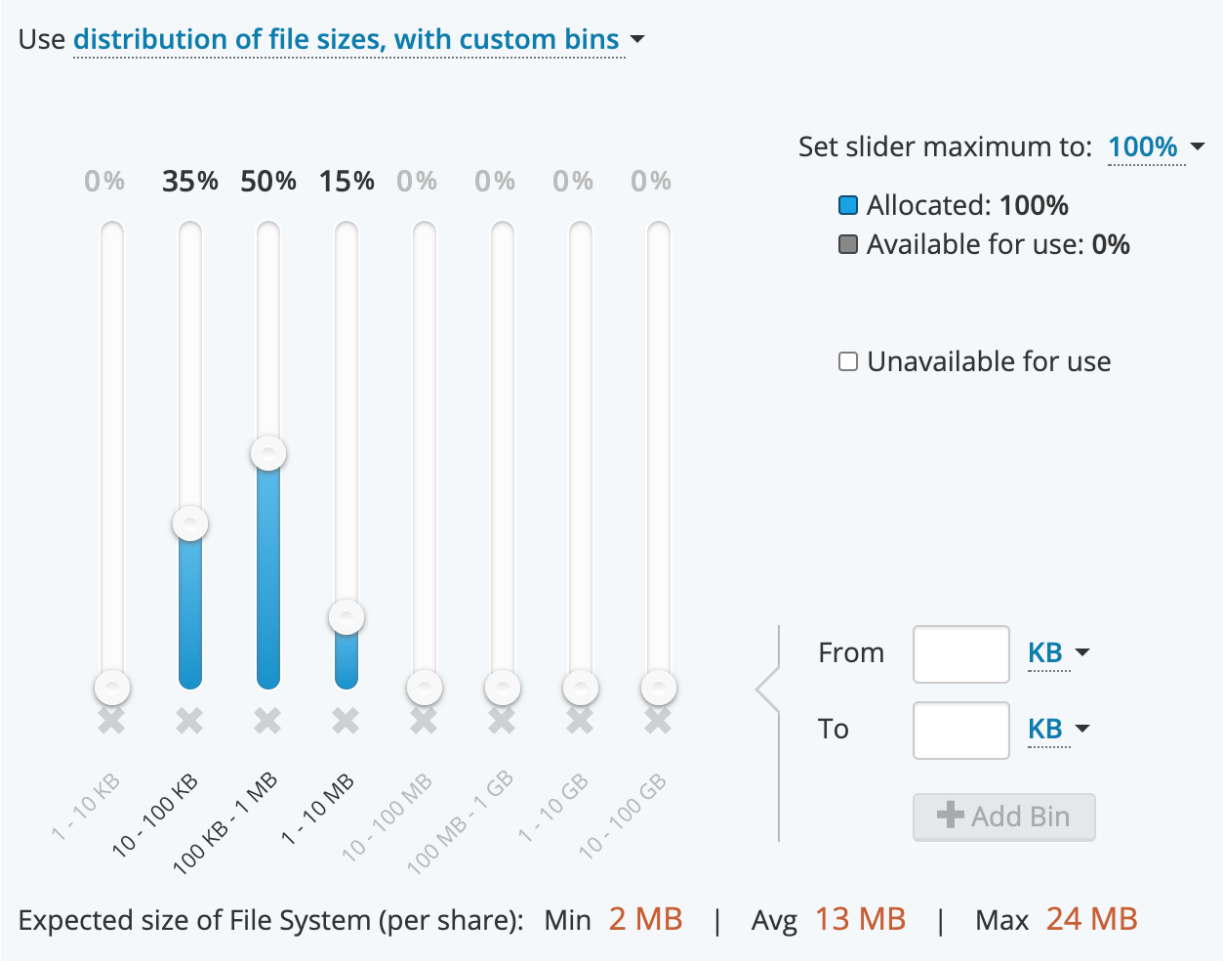


Constant file size is the default. You can change select the file size type from the drop-down list.

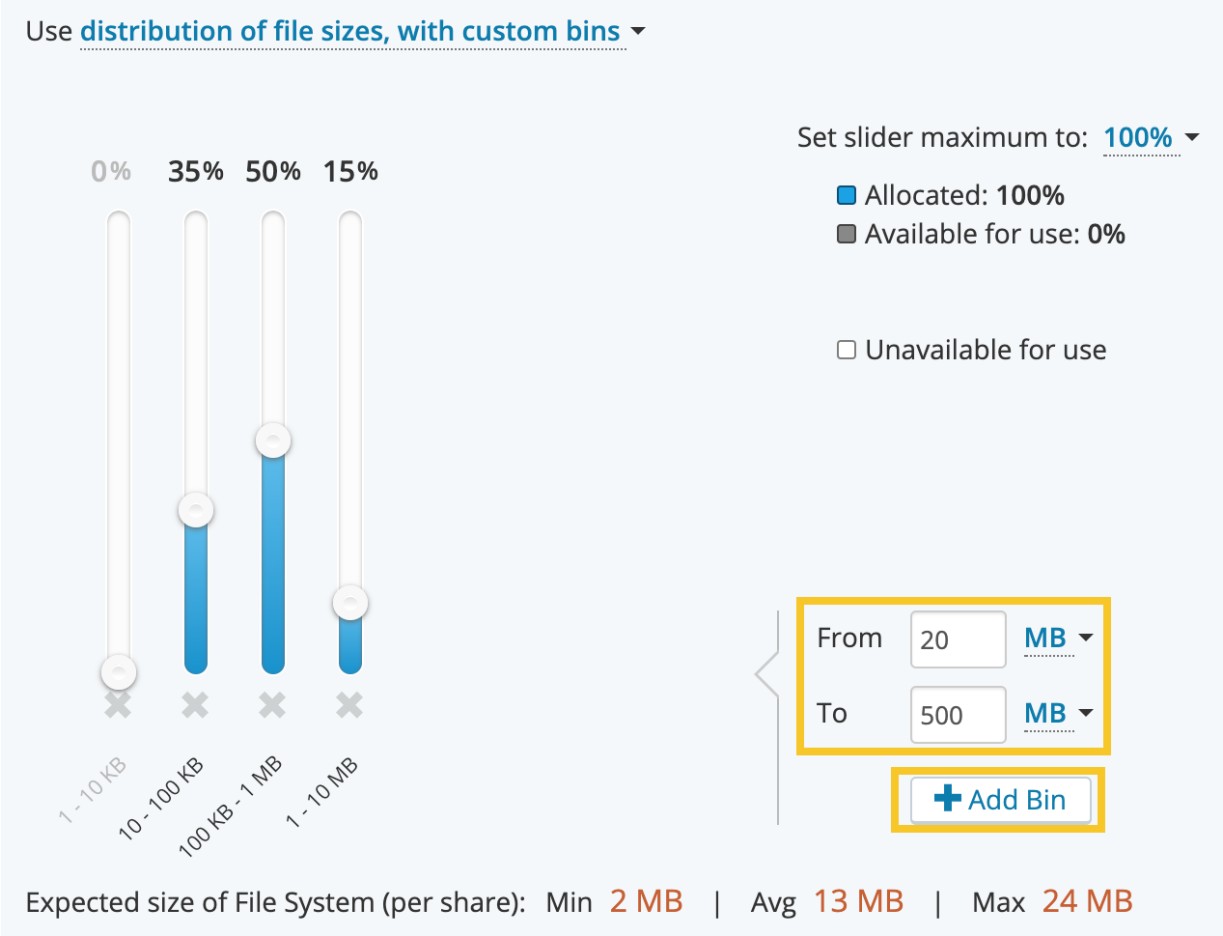
Use *random file size distribution* to randomly create files between the first value (minimum) and the second value (maximum size). You can specify random file size in bytes, kilobytes (KB), megabytes (MB) or gigabytes (GB).



You can specify up to eight custom bins. You can change the ratio of size distributions by moving the sliders up to increase the value or down to decrease. You can set the slider maximum by selecting maximum percentage value the sliders can have in the top righthand corner of the bin distribution section.

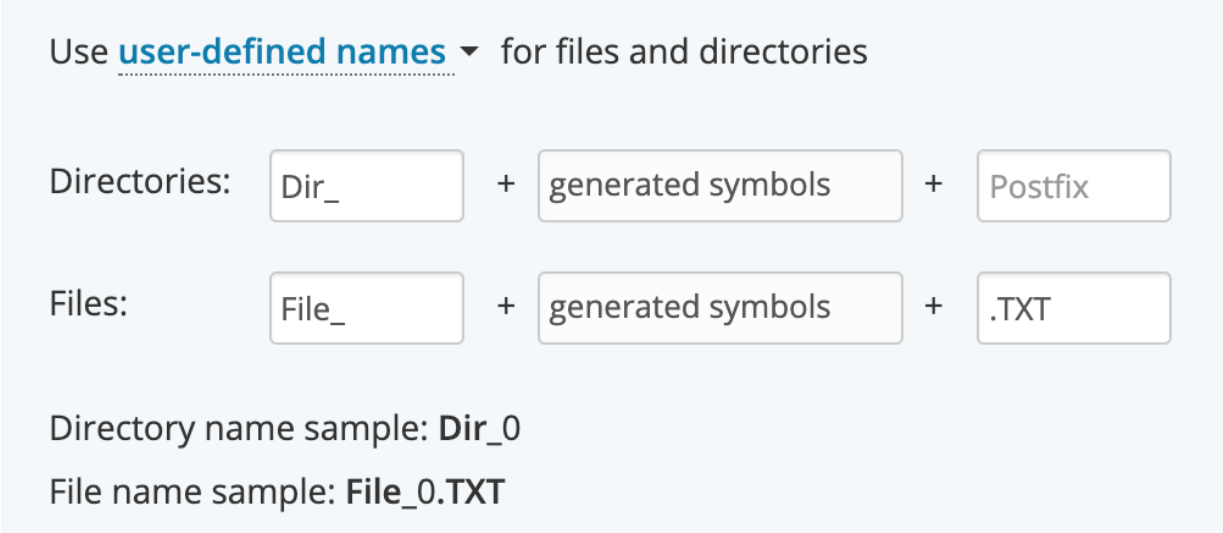


You can remove the default bins and replace with your own custom bins by clicking **+Add Bin**. You can specify the size range by filling in the start (*From*) and end (*To*) sizes in bytes, kilobytes (KB), megabytes (MB) or gigabytes (GB). However, you cannot specify bins that are overlapping in file sizes. When this occurs, the “Add Bin” button is grayed out and an error message displays when you hover over the **Add Bin** button.

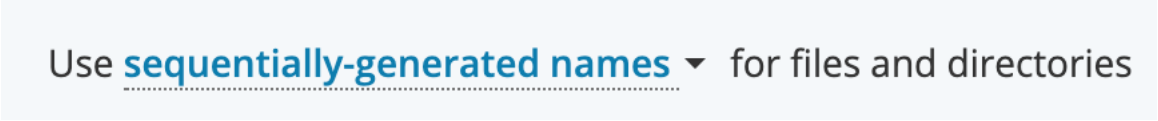


You can also specify the approach to the file names by either choosing *user-defined names* or *sequentially generated names* for files and directories.

Use *user-defined names* to specify a specific pre-fix text string, concatenated by a sequence of automatically generated symbols, and optionally concatenated by a specific postfix text string. You can specify values exceed the length of the text boxes. The generated symbols are not editable.



Use *sequentially-generated names* if you do not want to specify a specific text string for file and directory names.



Writes and Reads

Use the writes and reads section to configure the block sizes, the percentage of random versus sequential behaviors and the number asynchronous I/Os.

Data Parameters

File workloads support both the data deduplication and data compression features that are available under Data Parameters.

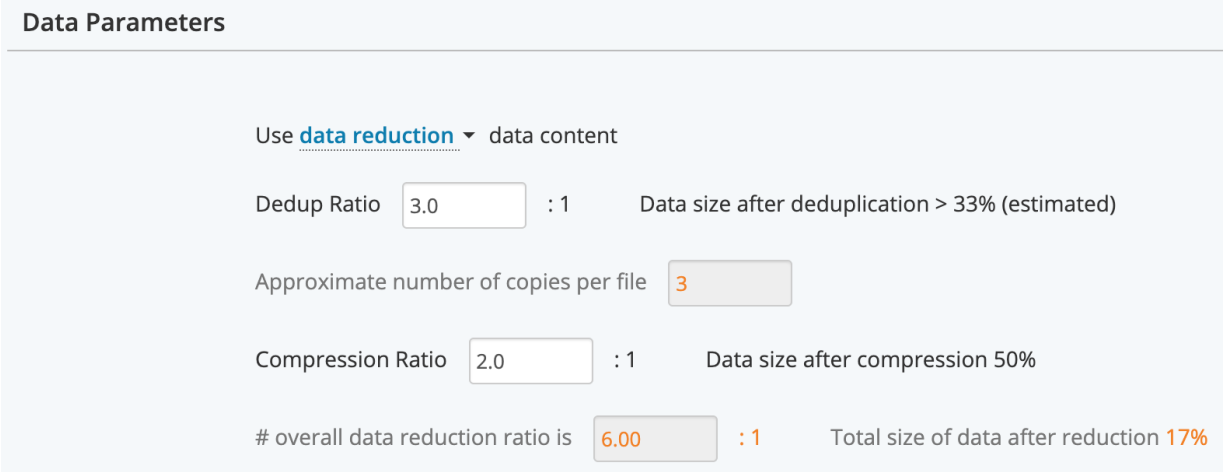
If you are testing with data deduplication, WorkloadWisdom generates the unique files and copies of those files that are required to emulate data deduplication functionality. The number and types of files generated depends on the deduplication ratio selected by the user. As an example, a deduplication ratio of 3:1 indicates that for every 3 files written, only 1 is unique.

For data compression, data content is generated that includes a block of 0's. Each file uses a specific data pattern to emulate data compression, to yield the user-configured data compression ratio. For example, a compression ratio of 3:1 indicates that for every 3 bytes of data, on average 1 is non-zero. The zeros are typically grouped together within the block.

The data deduplication and compression ratios are both rounded to 1 decimal place.

The following settings are available for deduplication and compression:

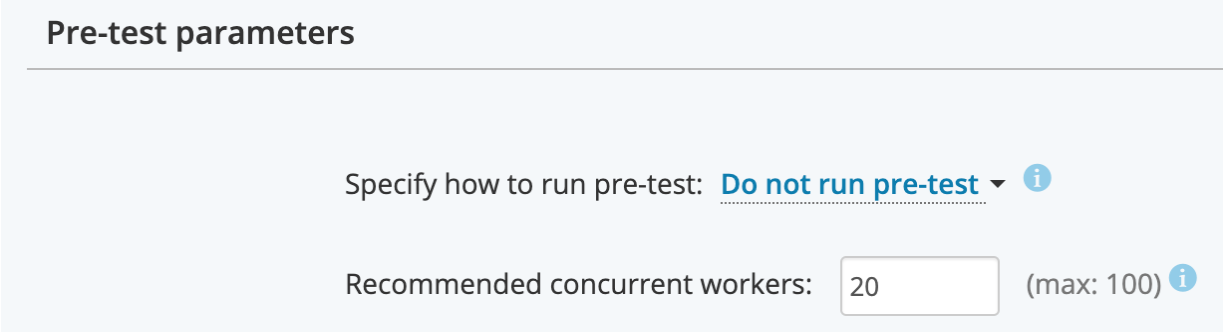
- *Dedup Ratio.* Generates unique files and copies of each unique file over the course of the workload test run that causes the system under test to yield the Dedup Ratio that you specify.
- *Compression Ratio.* Generates a data pattern for each file. The generated data pattern causes the system under test to yield the compression ratio that you specify.



It is recommended that you use at least one hundred files in the workload test, as with a small number of files, some deduplication ratios are not possible. For example, it is not possible to achieve a deduplication ratio of 50% with three files.

Pre-Test

For file workloads, the **Recommended concurrent workers** value is automatically set for you based on your workload's file system configuration, to optimize the file system creation time. You can change the recommended value to a lower number, but you cannot change it to a higher number. This is because there will be one worker dedicated to each folder.



Therefore, if you are creating a filesystem with only one folder or no folder at all (e.g. flat file system with no root folders) then this value must be 1. If you are creating a filesystem with 50 folders, then this value can be 1 to 50. If you are creating a filesystem with 200 folders, then this value can be 1 to 100, because 100 is the current maximum value.

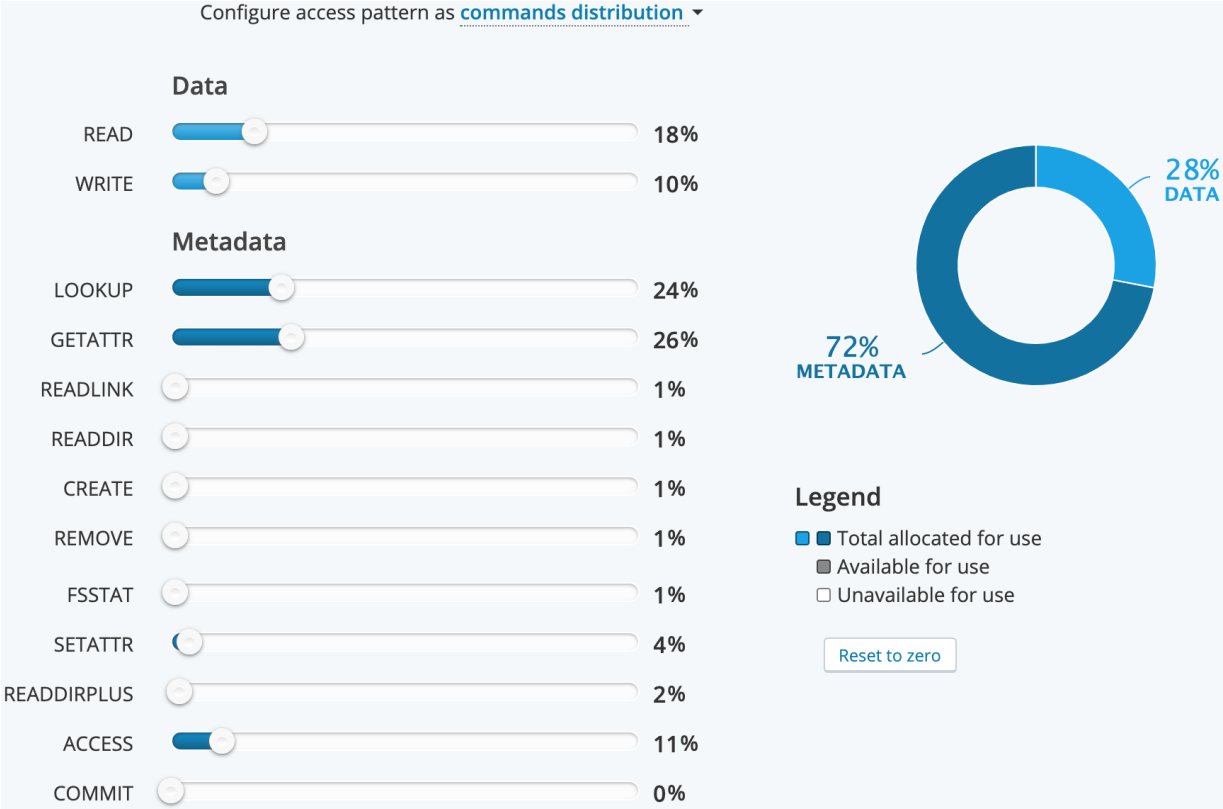
NFS Specifics

NFSv3 Constant Protocol Workload

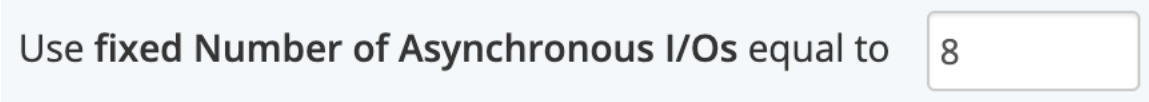
Access Pattern

The NFSv3 commands supported by the NFSv3 constant protocol workload are shown below.

Reset all the sliders to zero or reduce the command percentage by dragging the sliders to the left, to receive the unused percentage. Next, if you drag the sliders to the right, the percentage of that particular command increases.



The donut chart displays the overall data (read/write) versus the metadata ratio.



The **Asynchronous I/Os** parameter allows you to specify up to how many “chunks” of a R/W request can be concurrently outstanding “on the wire” (per worker) for Read and Write operations, respectively.

Let’s say you have a 10KB file you need to read, and the read block size is 1KB, then for each file for each concurrent worker:

- If Asynchronous I/Os = 1: At any given time, there is one 1KB read request on the wire, and the next 1KB read request will not be sent until the currently open 1KB read request has been responded to. So to complete the 10KB file read, there will be ten sequences of one read request – one read response exchanges.
- If Asynchronous I/Os = 8: The NFSv3 client will send out up to eight 1KB read requests even if zero responses have been received. However, it will not send the ninth 1KB read request until at least one read request has been responded to. So there is not a guaranteed sequence of one read request – one read response exchanges, but by the

time the 10KB file read completes, there will still be a total of ten read requests and ten read responses.
- If Asynchronous I/Os = 16: The NFSv3 client will send out up to ten 1KB read requests, because the 10KB file read can be completed with ten 1KB read requests.

Load Properties

The number of concurrent workers needs to be a multiple of the number of volumes access to maintain even workload across volumes. For example, if you have a total of 7 volumes, and you set the number of concurrent workers to 21, then the likelihood for the 7 volumes to receive equal load is the highest. However, it is not guaranteed always that the 7 volumes will have the same exact amount of load at all times depending on how the workload and test bed are configured, and how the SUT behaves.

More information

During an NFS workload run, the NFS client uses LOOKUPs to determine if a directory exists or not, and then makes a decision whether it is necessary to create a directory or not. Therefore, it is normal to see “failed” commands (e.g. failed Lookup response) when reviewing the test run results.

For information on designs and behaviors of the NFSv3 workload that is not specific to the NFSv3 protocol, see the **Common Workload Concepts and Settings** section as it relates to Constant Protocol Workloads, and the **File Protocol Workloads** section.

NFSv3 Temporal Protocol Workload

More information

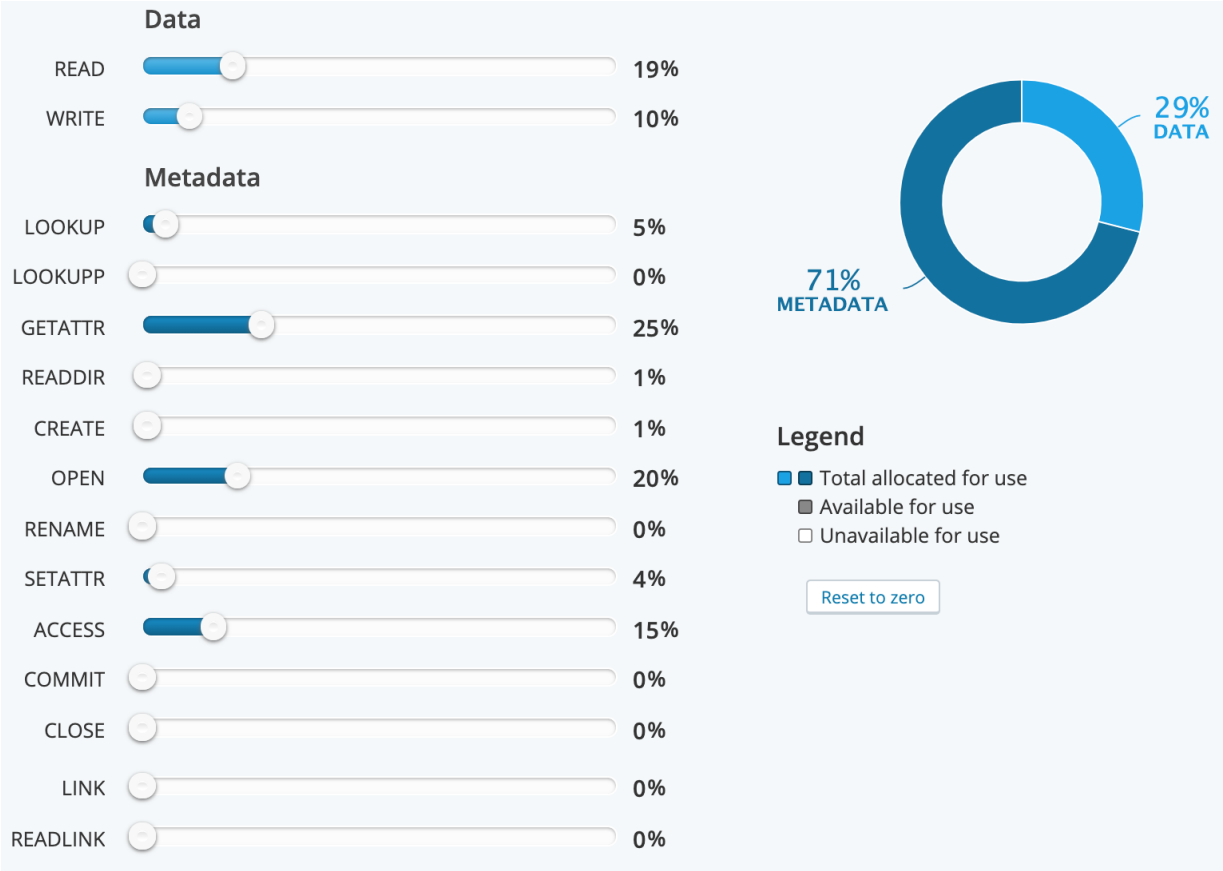
For information on designs and behaviors of the NFSv3 Temporal Protocol Workload, see **Common Workload Concepts and Settings** section as it relates to Temporal Protocol Workloads, the **File Protocol Workloads** section, and the **NFSv3 Constant Protocol Workload** section.

NFS Version 4 and NFS Version 4.1

NFS Version 4 (NFSv4.0 and NFSv4.1) differs from NFSv3 in both behavior and command set. The core commands supported by the NFSv4/NFSv4.1 workloads are displayed in the following example. Many of the commands are the same as NFSv3, although there are some differences.

New commands such as `open` and `close` are related to state management. Other commands, such as `fsstat`, are not available.

The following example displays the default values for NFSv4/NFSv4.1.



Reset all the sliders to zero or reduce the command percentage by dragging the sliders to the left, to receive the unused percentage. Next, if you drag the sliders to the right, the percentage of that particular command increases.

The donut chart displays the overall data (read/write) versus the metadata ratio.

More information

For information on designs and behaviors of the NFSv4/NFSv4.1 workload that is not exclusively specific to the NFSv4/NFSv4.1 protocol, see the Creating a New Workload Test [92] section as it relates to Constant Protocol Workloads, the File Protocol Workloads Details [114] section, and the NFS Specifics [125] section as it relates to NFSv3.

SMB Specifics

Server Message Block (SMB) Constant Protocol Workload

SMB Options

SMB supports CIFS/SMB 1.0, SMB 2 (2.0.2 and 2.1), and SMB 3.0.

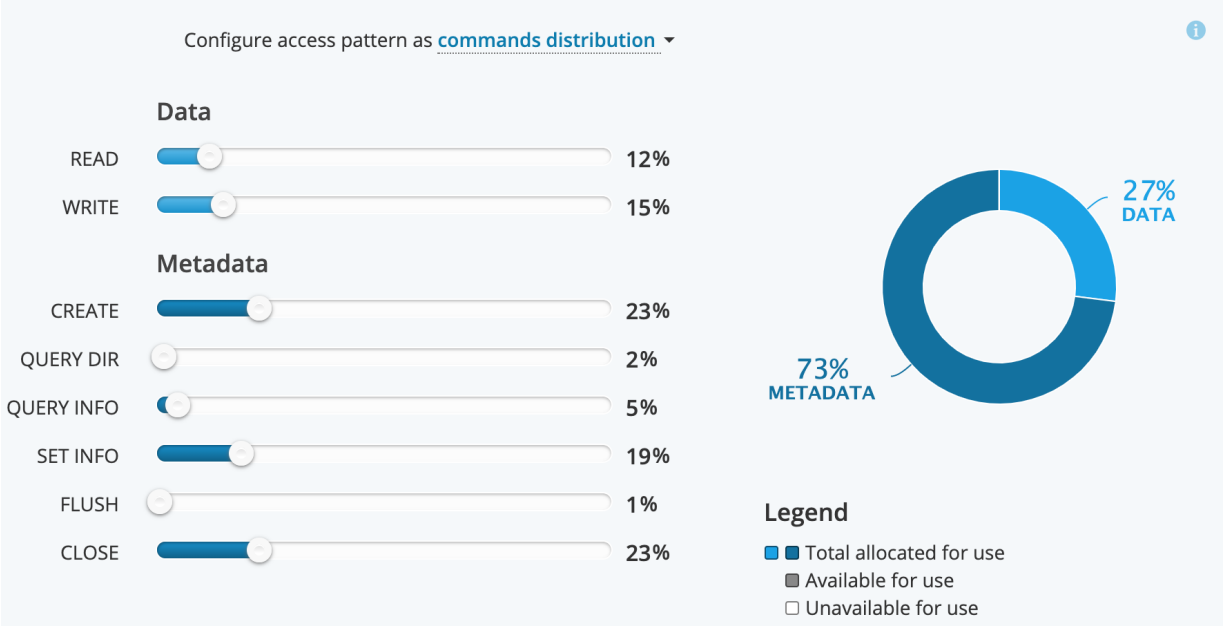
Access Pattern

Depending on the SMB version selected, some options may vary. However, the general concept is the same.

The following example displays the default values for SMB2.

Reset all the sliders to zero or reduce the command percentage by dragging the sliders to the left, to receive the unused percentage. Next, if you drag the sliders to the right, the percentage of that particular command increases.

The donut chart displays the overall data (read/write) versus the metadata ratio.



The **Asynchronous I/Os** parameter allows you to specify up to how many “chunks” of a R/W request can be concurrently outstanding “on the wire” (per worker).

Let’s say you have a 10KB file you need to read, and the read block size is 1KB, then for each concurrent worker:

- If Asynchronous I/Os = 1: At any given time, there is one 1KB read request on the wire, and the next 1KB read request will not be sent until the currently open 1KB read request has been responded to. So to complete the 10KB file read, there will be ten sequences of one read request – one read response exchanges.
- If Asynchronous I/Os = 8: The SMB client will send out up to eight 1KB read requests even if zero responses have been received. However, it will not send the ninth 1KB read request until at least one read request has been responded to. So there is not a guaranteed sequence of one read request – one read response exchanges, but by the time the 10KB file read completes, there will still be a total of ten read requests and ten read responses.
- If Asynchronous I/Os = 16: The SMB client will send out up to sixteen 1KB read requests at the same time. Some parts of a file may be read simultaneously or multiple times.

More information

For information on designs and behaviors of the SMB constant workload that are not exclusively specific to the SMB protocol, see the Creating a New Workload Test [92] section as it relates to Constant Protocol Workloads, and the File Protocol Workloads Details [114] section.

SMB2 Temporal Protocol Workload

The SMB2 temporal workload enables the workload to change access patterns, block sizes, and load to match the workload acquired using the Workload Data Importer. This enables a unique level of realism over other forms of testing and stimulates many aspects of different array architectures that are designed to handle burst and changes in workload behavior.

The workload does not appear in the default library because it can only be created using the Workload Data Importer. You must use the Workload Date Importer to create temporal workloads.

More information

For information on designs and behaviors of the SMB2 temporal workload that are not exclusively specific to SMB2 temporal workload, see Creating a New Workload Test [92] section as it relates to Temporal Protocol Workloads, the File Protocol Workloads Details [114] section, and the SMB Specifics [129] section.

Object Protocol Workloads

Object protocol workloads simulate the process of accessing objects in Amazon S3 or OpenStack-Swift.

Two Object protocol workload models are available in WorkloadWisdom:

- S3 Workload: A purpose-built constant protocol workload model for Amazon S3, with more granular control of the supported Amazon S3 APIs, ability to define short-lived temporary Objects during workload runtime, and more flexibility to work with existing Buckets.
- Object Storage Workload: A general purpose constant protocol workload model that supports both Amazon S3 and OpenStack Swift.

NOTE

This is the original Object protocol workload that was initially created before a purpose-built Amazon S3 Workload was added. No new features will be added to this workload, and it will be removed from WorkloadWisdom over the next few releases.

S3 Workload

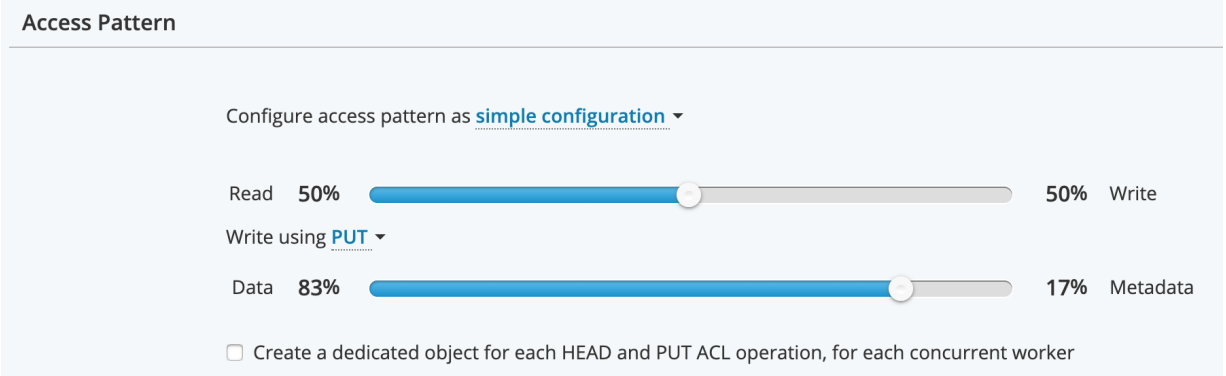
The Amazon S3 constant protocol workload is designed to simulate a variety of applications that use the Amazon S3 APIs. The following Amazon S3 functionalities are supported:

- Object “Read / Write” operations: PUT, POST, GET
- Metadata operations for Buckets and Objects
- Object Versioning
- Lifecycle
- Bucket / Object ACL
- Bucket Policy

In addition, the S3 Workload also supports the ability to create and delete temporary short-lived objects during a workload run (to simulate applications that can generate a large number of objects for temporary use), and is more flexible in working with existing Buckets.

Access Pattern

Two options are available for Access Pattern configuration: *simple configuration* and *command distribution*.



Simple configuration

Under *simple configuration*, you can specify the Read (GET) and Write (PUT, POST) distribution, and the overall Data versus Metadata operations. The Read / Write setting defines the Data operations. In the example above, the resulting workload will have 41.5% Read (GET), 41.5% Write (PUT), and 17% Metadata which is made up of other supported Amazon S3 commands that are not considered Data operations.

If you select the checkbox **Create a dedicated object for each HEAD and PUT ACL operation, for each concurrent worker**, then WorkloadWisdom creates one additional temporary object dedicated for each HEAD operation for each concurrent worker, and one additional temporary object dedicated for each PUT ACL operation for each concurrent worker. The effect of selecting this option is that you create many more Objects on the system under test than you configured in the Object Storage System section of this workload test, but allowing these different operations to be performed on different objects.

The effect of not selecting this option is that you will not end up creating many more Objects on the system under test than what is configured (except for the temporary Objects that must be created for DELETE operations, and additional short-lived temporary Objects if you enable the option **Create and Delete temporary Buckets and Objects**), but some Object systems under test might not be able to process HEAD or PUT ACL operations on the same object while that same object is receiving a PUT or GET operation.

Commands distribution

Under **Commands distribution**, you can individually control the percentage of each supported command. Below are the available commands:

- Data: These are “read / write” commands that transfer payload data to or from Objects in the data store.
  - PUT: S3 PUT Object API
  - POST: S3 POST Object API
  - GET: S3 GET Object API

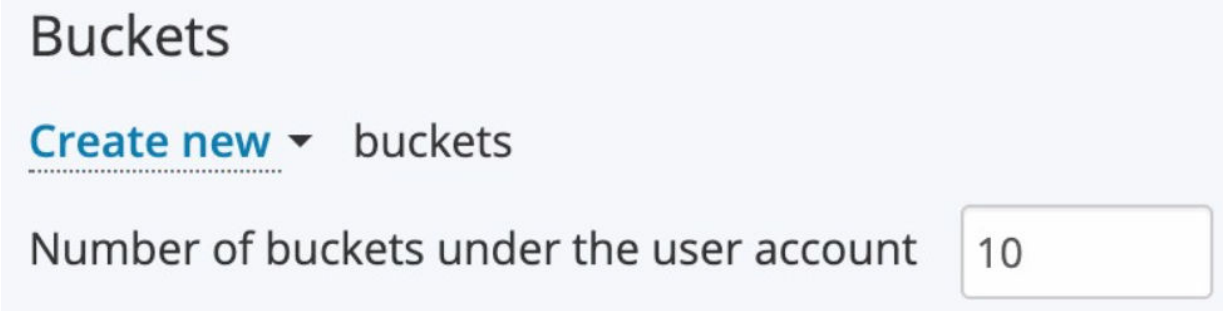
Retrieve older versions option: If set, then the specified percentage of GET commands will first issue a GET Bucket Object Versions command to retrieve the Version IDs of an Object, and then issue a GET Object with the most recent non-current (i.e. IsLatest = false) version of the Object.

- Metadata: These are commands that do not transfer payload data to or from Objects / Buckets in the data store. Unless specified otherwise, these commands do not include any specific Version ID in the requests.
  - HEAD Object
  - GET Object ACL
  - PUT Object ACL: Uses Canned ACL. Sets x-amz-acl to public-read-write by default.
  - HEAD Bucket
  - GET Bucket ACL
  - PUT Bucket ACL: Uses Canned ACL. Sets x-amz-acl to public-read-write by default. If **Number of Unique ACLs** is greater than 1, then the x-amz-acl field cycles through public-read-write, public-read, authenticated-read, bucket-owner-full-control, private.
  - GET Bucket Policy
  - PUT Bucket Policy: Sets Permissions for Object Operations to true by default, and others to false. If **Number of Unique Policies** is greater than 1, then some of the Object Operations will be set to false to give some variation to the Policies set.
  - GET Bucket Lifecycle
  - PUT Bucket Lifecycle: Sets a rule to transition Objects to Glacier (after 3,650 days). If **Number of Unique Lifecycle Rules** is greater than 1, then the aging and storage class transition rules vary to give some variation to the Lifecycle Rules set.
  - GET Bucket Versioning

Object Storage System

Buckets

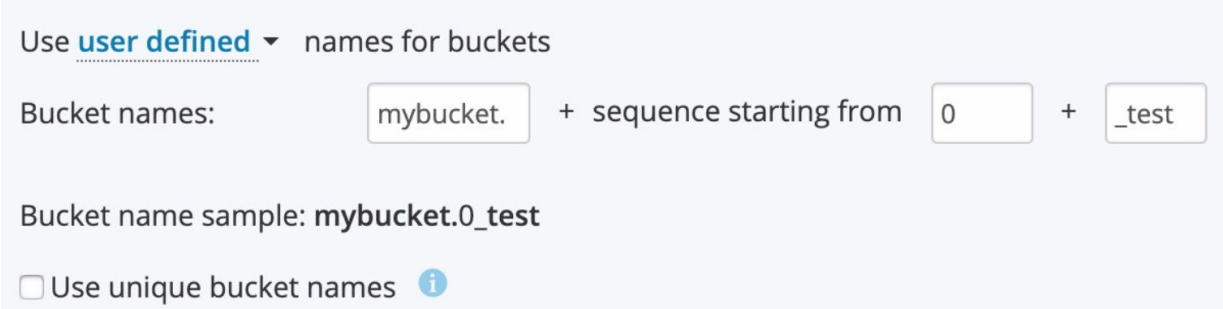
You can either use an existing Bucket to create the Objects needed for your workload, or create new Buckets for the Objects.



To create new Buckets for the Objects, select *Create new* in the drop-down menu, and specify the number of Buckets that should be created per user account (i.e. per S3 credential in the Test Bed). To use existing Buckets, select *Use existing*, and specify the Bucket names that exist on your Object storage system to hold the Objects.

Maximum number of existing Buckets: 100

If you choose to create new Buckets to use, then you can choose *auto generated* or *user defined*, which gives you the ability to specify the Bucket names' prefix and postfix. This is helpful later on when you need to delete Buckets and Objects in bulk.

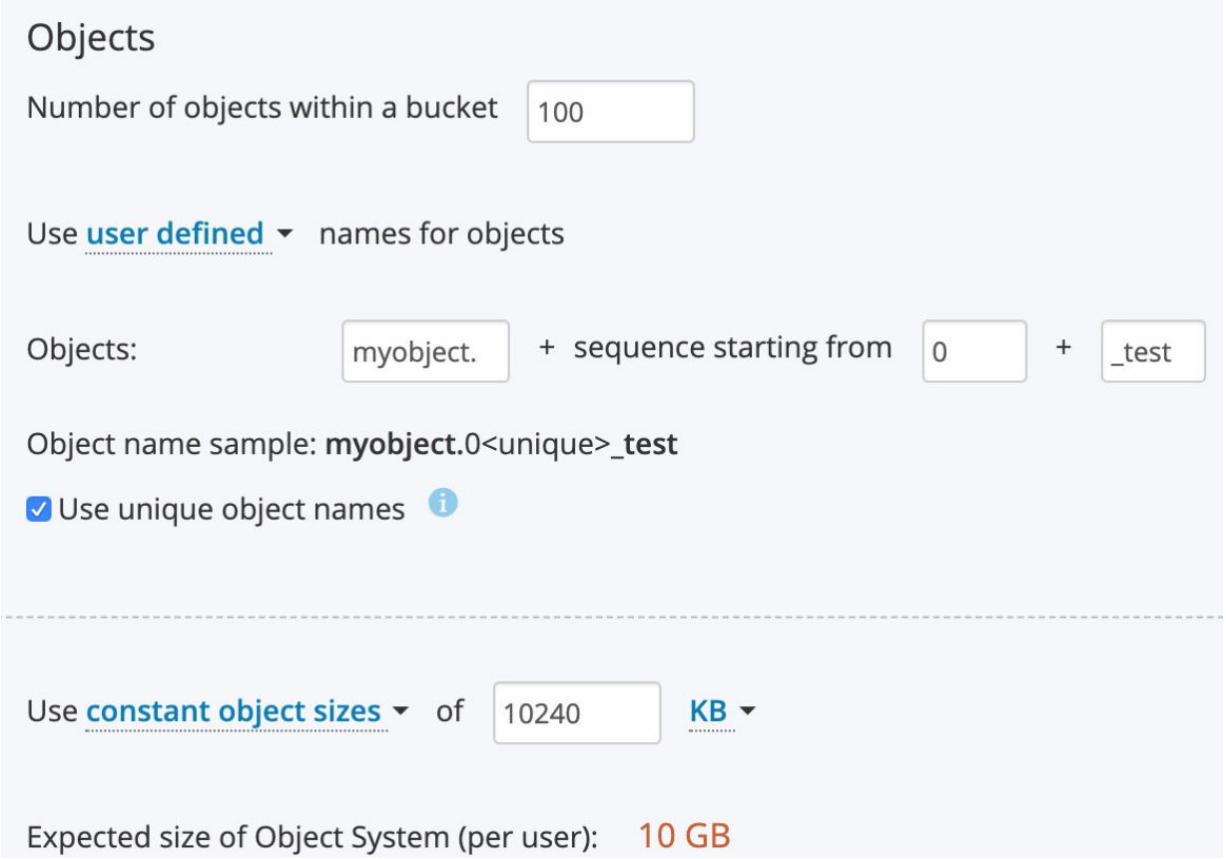


If you select the checkbox *Use unique bucket names*, then a unique ID will be inserted before the postfix. The unique ID will be generated per user account per test.

Optionally, you can **Enable Versioning** on the Buckets that you create. This option is not available if you select to *Use existing* Buckets.

Objects

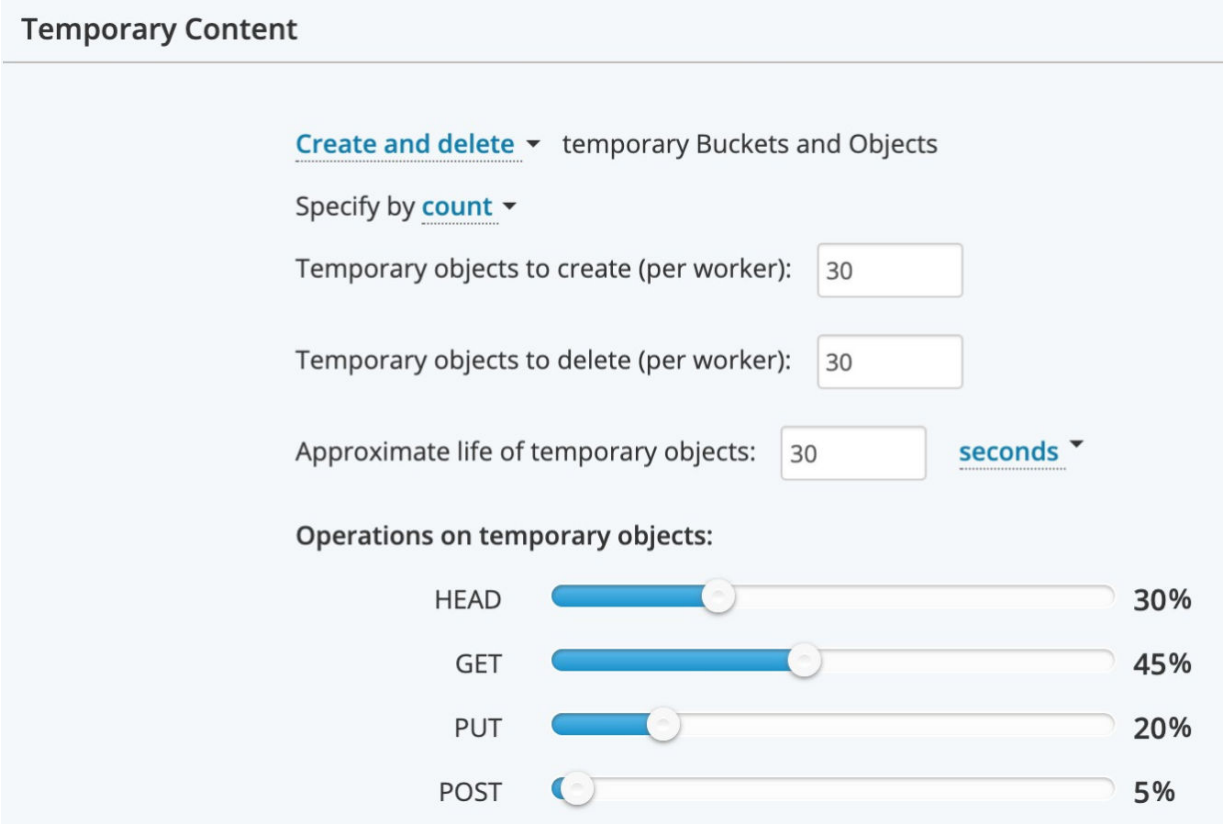
The options available for creating new Objects are similar to those for creating new Buckets. In addition, you need to specify the Object sizes.



As with file workloads, you can specify a *constant object size* or create your own *distribution of object sizes with custom bins*. To change the size distribution method, select the option from the drop-down menu.

Temporary Content

To generate short-lived Objects that are created and deleted throughout a workload run, click on the drop-down menu and change from *Don't create* temporary Buckets and Objects to *Create and delete* temporary Buckets and Objects. Use caution if you have a high number of workers, because the settings from this section are applied to each worker, which can result in a much larger number of Buckets and Objects on the System Under Test than anticipated.



You can control these numbers by *count* or by *percentage*, and the set values are applied to each worker. When you use *percentage*, it is a percentage of the total number of Objects defined in the Object Storage System section, except for Delete. Delete is a percentage of the temporary Objects created. The size of these Objects is the same as those configured in the Object Storage System section.

The location of the temporary Objects depends on your settings from the Object Storage System section as well. If you selected *Create new* Buckets, then each worker will create a new dedicated Bucket will be used to house the temporary Objects. If you selected *Use existing* Buckets, then a random existing Bucket will be used to house the temporary Objects. You can also specify the distribution of the operations performed on the temporary Objects.

For the temporary Objects created, you can specify the approximate life time for them. When the life time of the objects is up, the workload will begin deleting the temporary Objects in pseudo-random order.

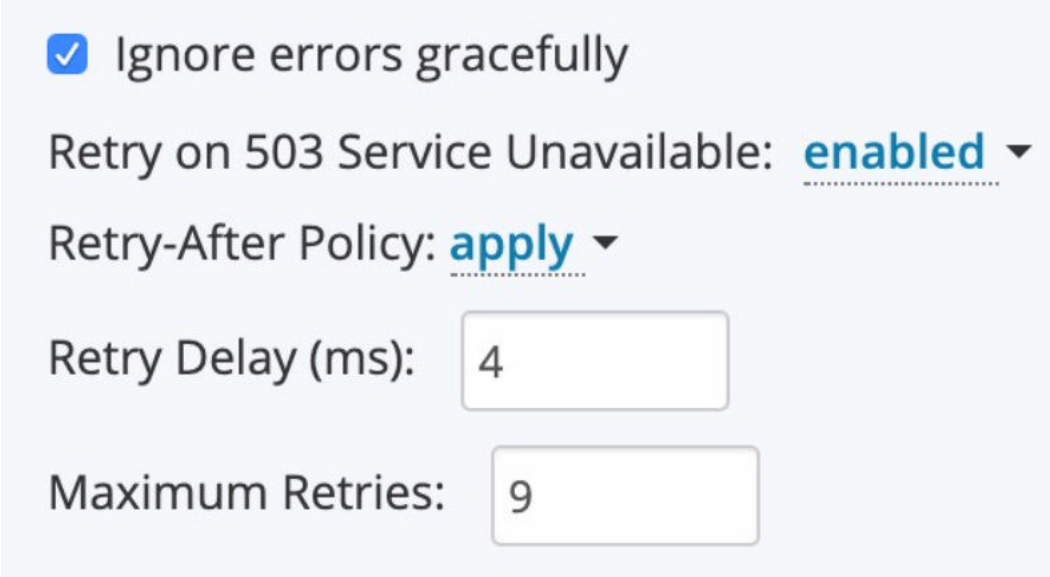
Depending on many factors including workload settings and how the System Under Test responds throughout the test run, the actual observed life of the temporary Objects may differ from the configured value, and that some temporary Objects may not be completely

deleted during the test run. To give an extreme example to illustrate this point, if you configure 1,000,000 temporary Objects with an average life of 5 seconds for the temporary Objects, and set the workload Duration to 5 minutes, and that the System Under Test has an average response time of 100ms, then the workload simply cannot run as expected.

A general recommendation is that the **Approximate life of temporary Objects** should be several times greater than Average Response Time x **Temporary objects to create** and **delete**, and set the workload run Duration to several times greater than the life time value as the workload performs operations on both the temporary Objects as well as the Objects that exist on the Object Storage System from the Pre-Test run.

Response Handling

By default, 400-level and 500-level HTTP Responses and considered errors, and a worker encountering these errors will terminate its workload. You can optionally select the checkbox **Ignore errors gracefully** to force the workload to continue to run when it receives 400-level and 500-level HTTP Responses.



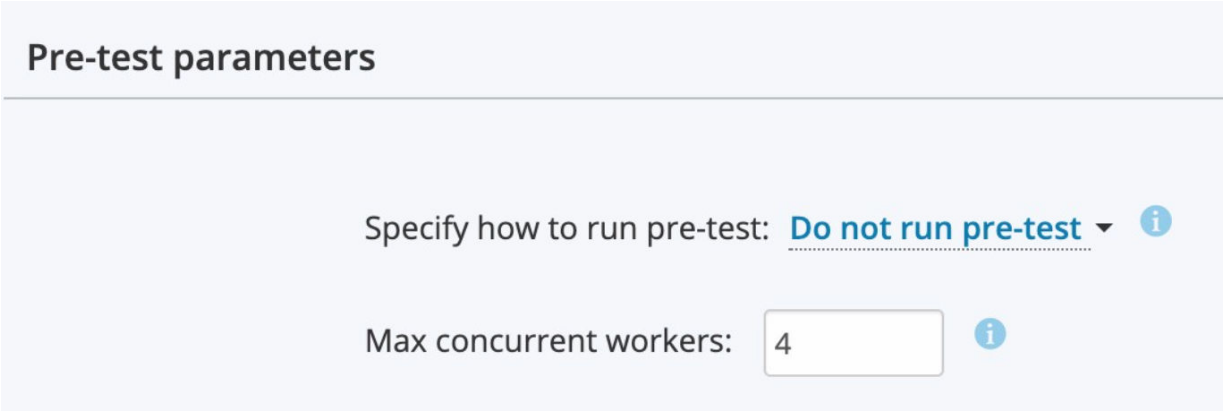
In addition, for 503 HTTP Responses, you can specify that the workload performs a Retry of the HTTP Request that resulted in the 503 HTTP Response. If this is set, then whenever a worker receives a 503 Response to a Request, it performs a retry after **Retry Delay**, until the number of retries exceeds **Maximum Retries**.

Pre-test parameters

The S3 Pre-Test populates the Objects, and optionally Buckets (depending on your workload settings), that are required for your workload to run. When you run an S3

workload on your SUT for the first time, you must run the pre-test at least once. Depending on your workload settings, subsequent runs may or may not require a pre-test run. For example, if you apply the setting **Use unique object names**, then you need to run pre-test because a unique identifier is created at run time and is used as part of the Object name.

The **Max concurrent workers** setting allows you to speed up the Object System population time by using multiple workers to create the configured Buckets and Objects simultaneously. It is important to note that all S3 credentials defined in the Test Bed can be used, so if you have strict access to the S3 SUT, then it is best to either reconfigure the SUT to allow any S3 credential to create new Buckets, or use a Test Bed with a single S3 credential that can create new Buckets, or just use an existing Bucket and create only new Objects (not create new Buckets).



For example, if you have 1 S3 credential defined in the Test Bed, and set the pre-test concurrent workers to 4, then all 4 workers will use the same S3 credential during pre-test. If you have 8 S3 credentials defined in the Test Bed, and set the pre-test concurrent workers to 4, then, all else being equal, each worker will use 2 S3 credentials throughout the pre-test.

More information

For more information on designs and behaviors of the S3 workload that are not exclusively specific to the S3 protocol, see the Workload Tests Concepts [90] section as it relates to the Constant Protocol Workloads section.

Object Storage Workload (will become obsolete)

NOTE

The Object Storage Workload is an obsolete general purpose workload model for Object Storage protocols that include OpenStack Swift and Amazon S3. This workload was built before the new dedicated S3 workload described above, and it contains less functionality and flexibility.

The behavior is dependent on the metadata operations, block sizes, and other aspects, which are negotiated by the client and the object storage server. Unlike file or block workloads, the object enables you to select the protocol within the workload, or, if simple configuration is selected, it defaults to the protocol in the test bed against which it is run. As with file workloads, metadata commands differ between protocols.

WorkloadWisdom provides out of the box models for Amazon S3 and OpenStack-Swift in a single workload: Object Storage Workload. SNIA CDMI is supported by running TDE tests on WorkloadWisdom.

Access Patterns

You can specify access patterns in the following ways:

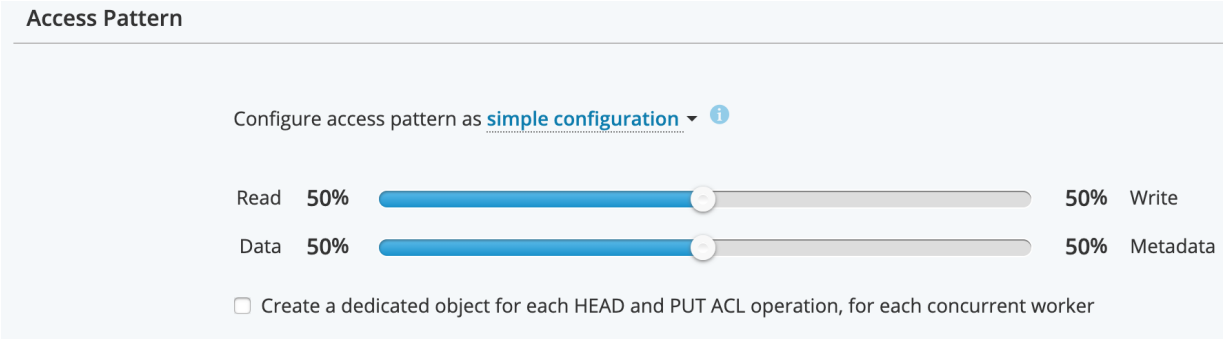
- Simple configuration
- Common configuration
- Amazon S3 commands distribution
- OpenStack-Swift commands distribution

Simple and common configuration automatically map object access concepts to the appropriate protocol command based on the service specified in the test bed. This enables a single workload to be run to compare Amazon S3 versus OpenStack-Swift or against a mixed environment.

Amazon S3 commands distribution and OpenStack-Swift commands distribution provide direct control over the protocol commands that are sent, but can only be used on test bed services for the selected protocol.

Simple Configuration

If you use the *simple configuration*, the protocol that runs is automatically determined by service configured in the test bed.



To change the ratio, move the slider to reflect the desired percentage value. If you move the slide to the left for the read/write percentage the read percentage increases.

Simple configuration is the default. When you use the simple configuration option, you only need to specify read/write ratio and data/metadata ratio, but you cannot control the distribution of different metadata commands. Simple configuration will evenly distribute all available commands.

For example, assume you use *simple configuration* and have the following settings:

Read: 80%

Write: 20%

Data: 40%

Metadata: 60%

The resulting workload will have 32% read operations, 8% write operations, and 60% metadata operations, depending on whether you run this workload on an Amazon S3 test bed or an OpenStack Swift test bed, or use different HTTP methods or APIs. For example, if you run this on an Amazon S3 test bed, the supported “write” operations for Amazon S3 can be PUT NEW, PUT REPLACE, or POST APIs, so the 8% write operations are further divided evenly across the three methods (i.e., 2.67% PUT NEW, 2.67% PUT REPLACE, and 2.67% POST). Similarly, for the metadata operations, there are seven supported APIs, so the 60% of metadata operations are evenly divided by seven (i.e. 8.57% DELETE OBJECT, 8.57% HEAD BUCKET, 8.57% GET BUCKET ACL, etc.). However, in this specific example there are more DELETE OBJECT APIs than PUT NEW APIs which would result in erroneous conditions as there are not enough Objects to be deleted, so the percentage of DELETE

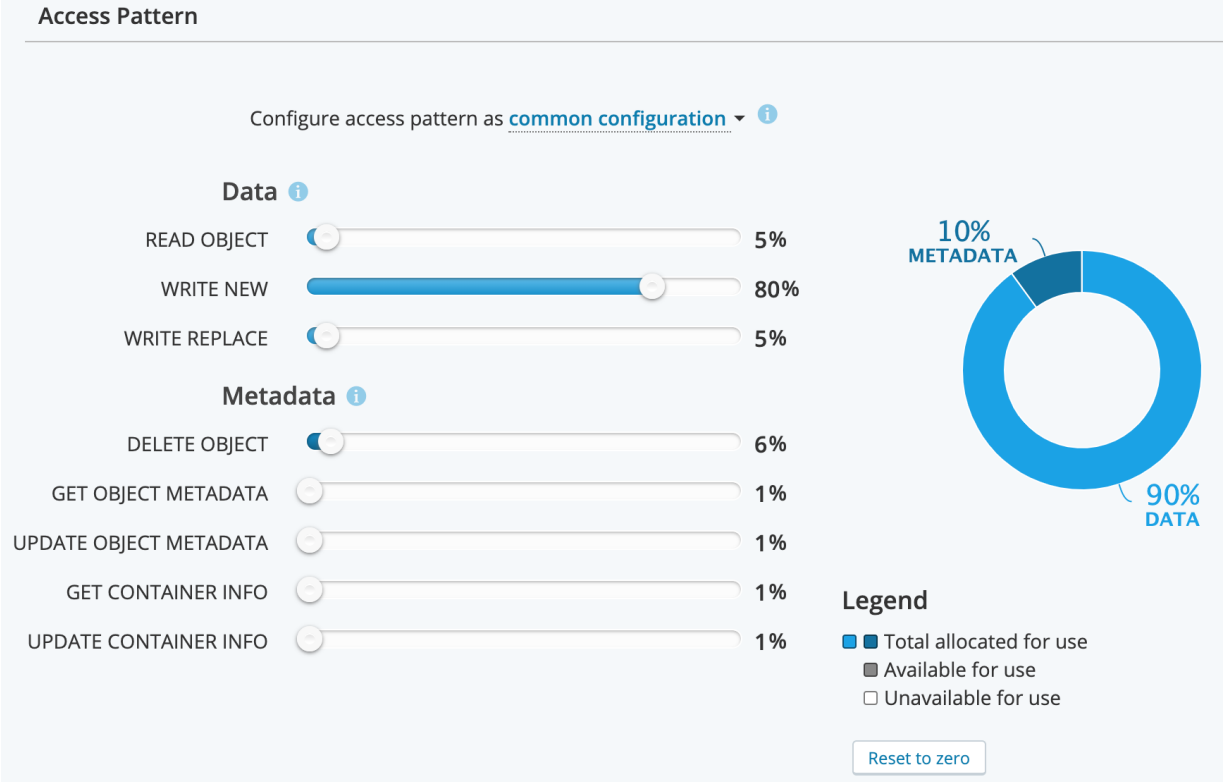
OBJECT APIs is updated to match the percentage of PUT NEW APIs (i.e. 2.67% in this specific example).

The checkbox **Create a dedicated object for each HEAD and PUT ACL operation, for each concurrent worker** is applicable only to Amazon S3 test beds. If you select the checkbox, then WorkloadWisdom creates one additional temporary object dedicated for each HEAD operation for each concurrent worker, and one additional temporary object dedicated for each PUT ACL operation for each concurrent worker.

The effect of selecting this option is that you create many more Objects on the system under test than you configured in the Object Storage System section of this workload test, but allowing these different operations to be performed on different objects. The effect of not selecting this option is that you will not end up creating many more Objects on the system under test than what is configured (except for the temporary Objects that must be created for DELETE operations), but some Object systems under test might not be able to process HEAD or PUT ACL operations on the same object while that same object is receiving a PUT or GET operation.

Common Commands Configuration

Use *Common commands configuration* for more realistic control over the metadata traffic than simple configuration, while still enabling the protocol comparison and mixed protocol test bed use cases. This configuration is not used frequently unless you have a use case to compare how Amazon S3 and OpenStack Swift perform against each other when “similar comparable” Data and Metadata operations are used.



The following table displays how commands are mapped to protocols when applied to a test bed with the specified protocol.

Table 1. Mapping of Common commands to Amazon S3 APIs and OpenStack Swift APIs

Access	S3	OpenStack-Swift
Read Object	Get	Retrieve Object
Write New	Put New	Write Object New
Write Replace	Put Replace	Write Object Replace
Delete Object	Delete	Delete Object
Get Object Metadata	Head Object and Get Object ACL	Retrieve Object Metadata
Update Object Metadata	Put Object ACL	Update Object Metadata
Get Container Info	Get Bucket ACL	Retrieve Container
Update Container Info	Put Bucket ACL	Write Container

`Write New` and `Write Replace` do not indicate different protocol commands but rather whether the object being written is one that was created during the pretest phase (replace) or a new object name. If `Write New` is greater than `Delete Object`, the amount of storage needed grows with each test run and is not be cleaned up at the end of the test. `Write Replace` might also result in increased storage depending on versioning configurations of the storage system.

NOTE

Using `Get Object Metadata` has more overhead on Amazon S3 than on OpenStack-Swift because two operations are performed instead of one. Depending on the workload this might not be what would happen in a real workload and is a worst-case scenario use case for Amazon S3.

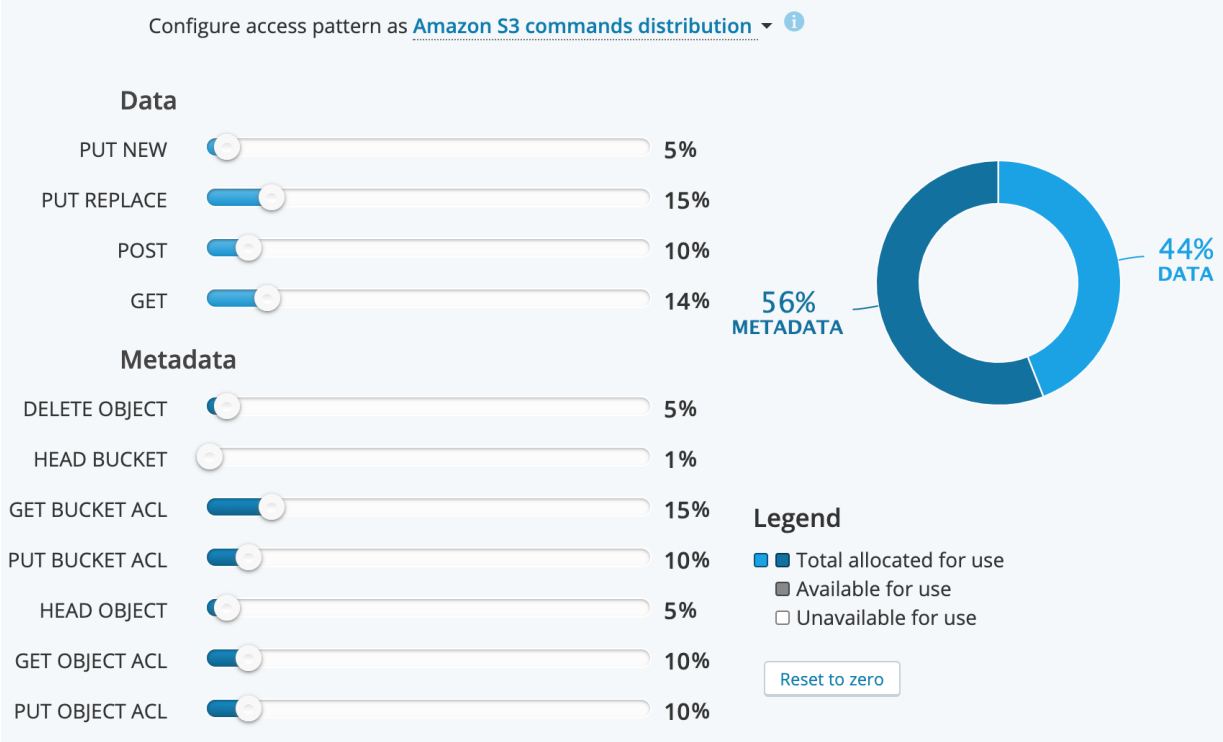
Data Parameters

Object protocol workloads do not support the deduplication feature of data reduction.

Amazon S3 Commands Distribution

When using the *Amazon S3 commands distribution* configuration, the commands (those with at least 1%) are issued the same way as the *simple configuration* when applied on an Amazon S3 Test Bed. The only difference is the percentage of commands issued.

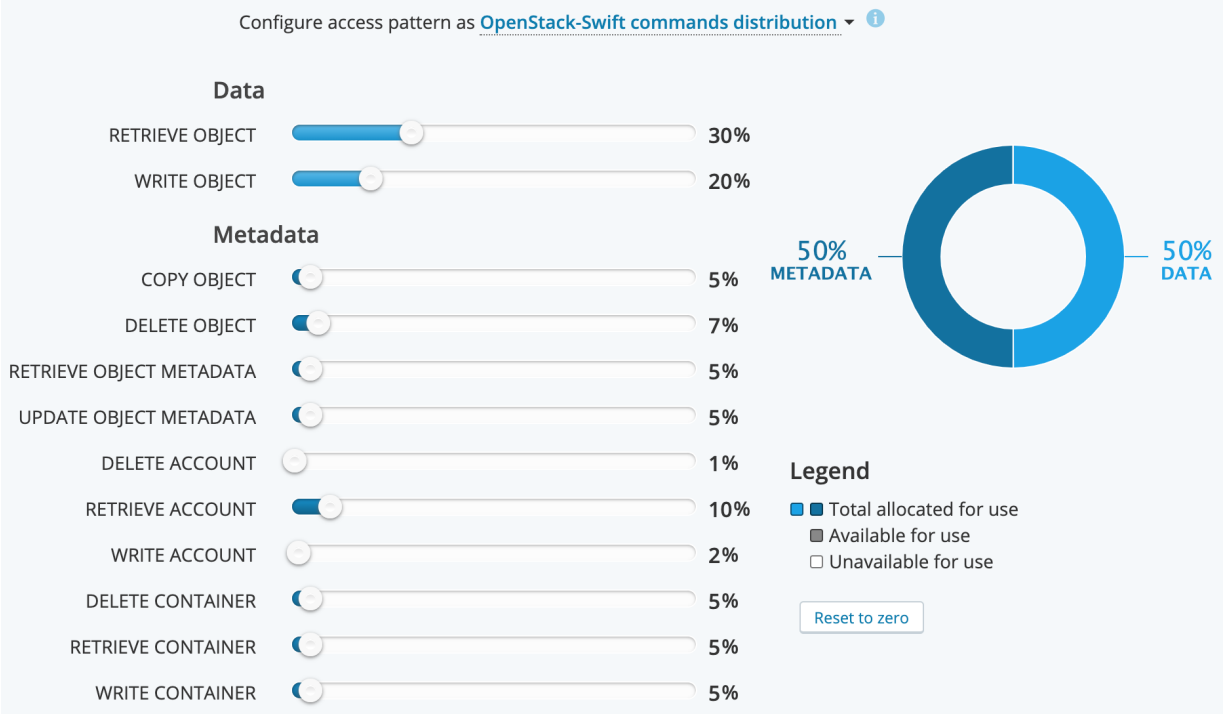
The donut chart displays the overall data (read/write) versus the metadata ratio. Gray is displayed if there is unused capacity.



OpenStack-Swift Commands Distribution

When using the *OpenStack-Swift commands distribution* configuration, the commands (those with at least 1%) are issued the same way as the *simple configuration* when applied on an OpenStack Swift test bed. The only difference is the percentage of commands issued.

The donut chart displays the overall data (read/write) versus the metadata ratio. Gray is displayed if there is unused capacity.



Commmand Reference

The access pattern specified can be mapped to those specified by: `http://developer.openstack.org/api-ref-objectstorage-v1.html`

OpenStack-Swift uses the keystone authentication service within OpenStack and this can be configured in the OpenStack-Swift service in the test bed.

Table 2. OpenStack Swift Command Mapping

Virtana Label	OpenStack Reference
Data	
RETRIEVE OBJECT	Objects Get
WRITE OBJECT	Objects Put
Metadata	
COPY OBJECT	Objects Copy
DELETE OBJECT	Objects Delete
RETRIEVE OBJECT METADATA	Objects Head

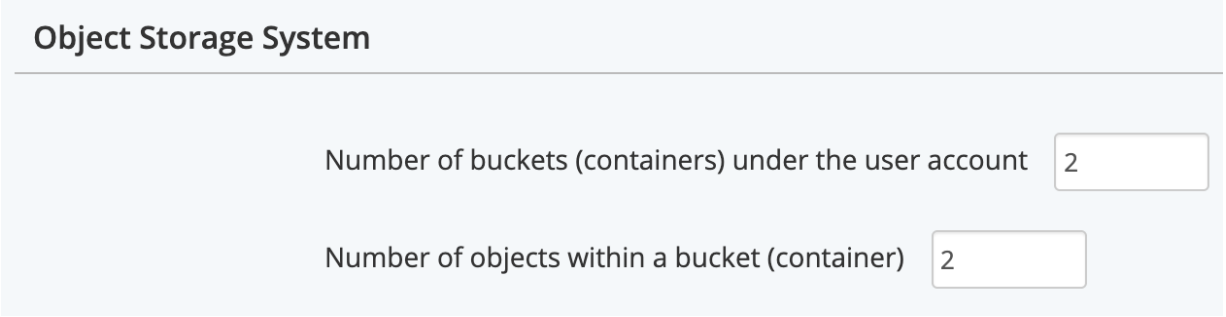
Virtana Label	OpenStack Reference
UPDATE OBJECT METADATA	Objects Post Metadata
DELETE ACCOUNT	Accounts Post to Delete account metadata
RETRIEVE ACCOUNT	Accounts Head
WRITE ACCOUNT	Accounts Post with updates
DELET CONTAINER	Containers Delete
RETRIEVE CONTAINER	Containers Head
WRITE CONTAINER	Containers Post with metadata

Object Storage System Settings

Unlike file systems, object storage is flat and stores objects in containers (OpenStack-Swift) or buckets (Amazon S3). Buckets and containers are conceptually the same and cannot contain other buckets or containers. As a result, you cannot create a container or bucket hierarchy. However, both Amazon S3 and OpenStack Swift support the concept of a “folder” for organization purposes in a manner that is conceptually similar to file systems. For Amazon S3 and OpenStack Swift, this can be accomplished by including a forward slash (/) character in the object name.

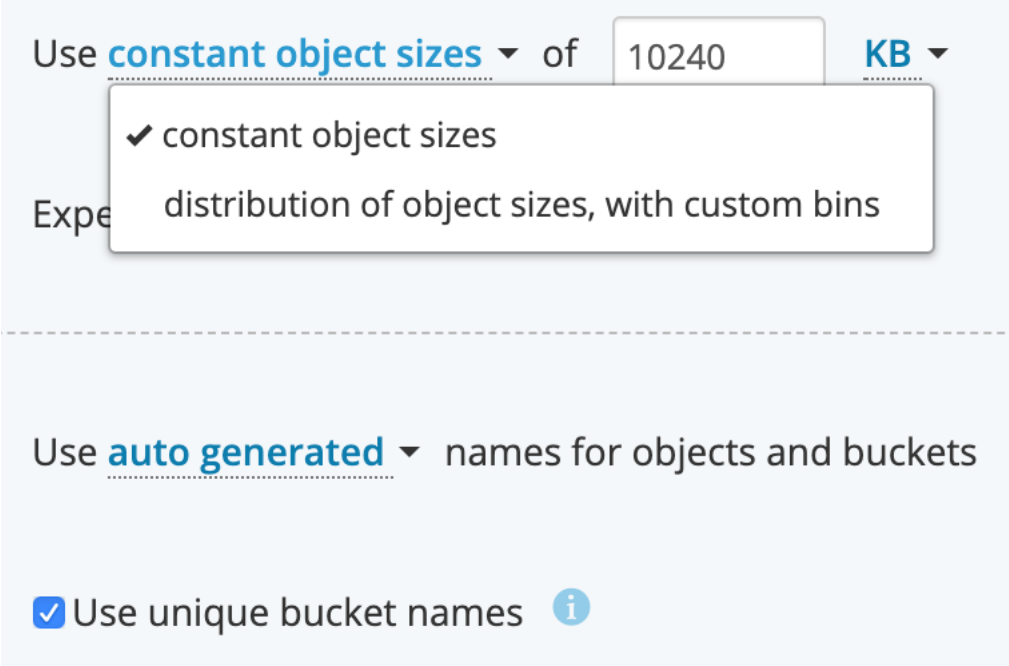
Specify how many containers or buckets you create and the number of objects per container or bucket. When you enable Pre-Test, WorkloadWisdom builds out the object storage system before the main workload run. The total number of objects created during Pre-Test is:

Total number of objects = (number of objects) x (number of buckets)



Note that the main workload that runs after Pre-Test completes may produce additional temporary objects for reasons described earlier in this section.

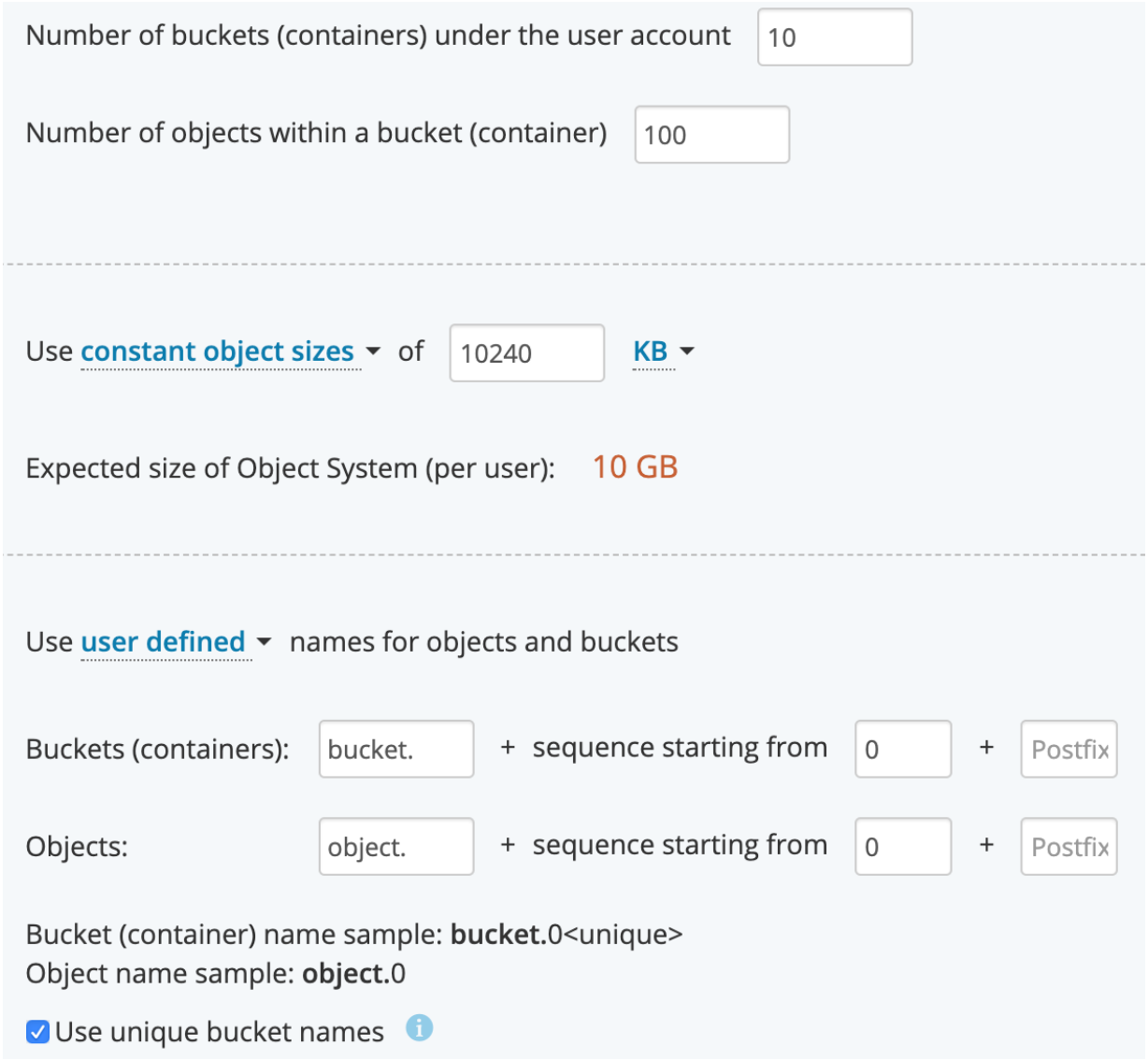
As with file workloads, you can specify a *constant object size* or create your own *distribution of object sizes with custom bins*. To change the size distribution method, select the option from the drop-down menu.



Object names can be *auto-generated* or *user-defined*.

Depending on the object storage system under test and the user privileges, you may or may not have the authorization to create Buckets or Containers on the fly, and you may have to use existing Buckets or Containers. If this is the case, you need to deselect **Use unique bucket names** and use **user defined** names for objects and buckets or containers so that the workload uses predictable bucket or container names per user.

For example, assume that the workload contains the following settings for the object storage system.

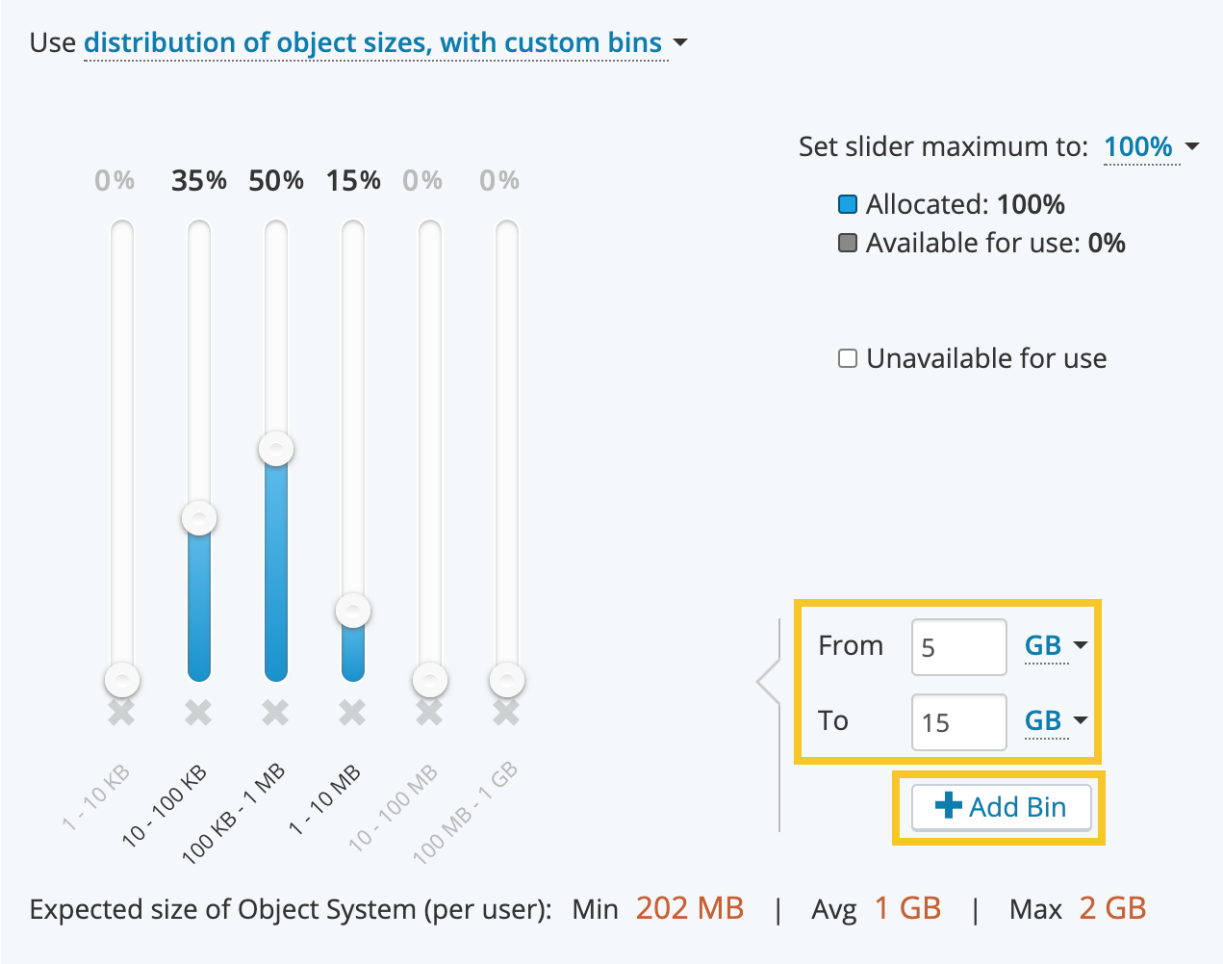


Then the following buckets are used to ensure unique Buckets are created and accessed per user.

- If there are **3 workers for 3 user accounts** then, according to screenshot, example should be:
  - If the *Use unique bucket names* option is checked:
    - Worker 1, user 1: bucket.0<1st unique number>, bucket.1<1st unique number>, ...,bucket.9<1st unique number>
    - Worker 2, user 2: bucket.10<2nd unique number>, bucket.11<2nd unique number>, ...,bucket.19<2nd unique number>

    - Worker 3, user 3: bucket.20<3rd unique number>, bucket.21<3rd unique number>, ...,bucket.29<3rd unique number>
  - If the option *Use unique bucket names* is unchecked:
    - Worker 1, user 1: bucket.0, bucket.1, ...,bucket.9
    - Worker 2, user 2: bucket.10, bucket.11, ...,bucket.19
    - Worker 3, user 3: bucket.20, bucket.21, ...,bucket.29
- If there are **3 workers for 2 user accounts** then, according to screenshot, example should be:
  - If the *Use unique bucket names* option is checked:
    - Worker 1, user 1: bucket.0<1st unique number>, bucket.1<1st unique number>, ...,bucket.4<1st unique number>
    - Worker 2, user 2: bucket.10<2nd unique number>, bucket.11<2nd unique number>, ...,bucket.19<2nd unique number>
    - Worker 3, user 1: bucket.5<1st unique number>, bucket.6<1st unique number>, ...,bucket.9<1st unique number>
  - If the option *Use unique bucket names* is unchecked:
    - Worker 1, user 1: bucket.0, bucket.1, ...,bucket.4
    - Worker 2, user 2: bucket.10, bucket.11, ...,bucket.19
    - Worker 3, user 1: bucket.5, bucket.6, ...,bucket.9

As in the file system section you can use a distribution of object sizes, with up to eight bins supported. Remove the default bins. Next, click **+Add Bin**. You can specify the size range by filling in the start (from) and to (end) sizes in bytes, kilobytes (KB), megabytes (MB) or gigabytes (GB). However, you cannot specify bins that are overlapping in object sizes. When this occurs, the “Add Bin” button appears grayed out and an error message displays when you mouse-over the **Add Bin** button.



You can change the ratio of size distributions by moving the sliders up to increase the value or down to decrease. You can set the slider maximum by selecting maximum percentage value the sliders can have in the top righthand corner of the bin distribution section.

Block Protocol SCSI Workloads

Block workloads simulate the process of accessing SCSI block devices across a Fibre Channel or Ethernet network (iSCSI). For block access, there are far fewer metadata commands comparing to file or object access. This is because the metadata is primarily abstracted by the filesystem or object storage (or data structure for raw storage), so queries to discover file layout turn into actual reads. To characterize block data it is important to understand the basic command mix, if the data is being accessed

sequentially or randomly, the I/O request sizes, any hotspots, data content, and how LUNs are being used.

Workload Design and Behaviors

The SCSI Workloads (FC-SCSI, iSCSI) use a probability-based design to generate the user specified distribution of commands and load. Factors such as SUT performance, Test Bed, errors, and also runtime duration can cause the actual distribution and load to be very different from the expected distribution and load.

For example, if READ/WRITE ratio = 71%/29%, it means that READ probability is 0.71 and WRITE probability is 0.29 during the workload run. More precisely, this concept is applied to into 4 groups: sequential Read, sequential Write, random Read, and random Write. The "larger" your test environment is, the better a longer workload run duration should be configured as, from a probability perspective, it is more likely that the actual observation will match the configuration.

When you have multiple concurrent workers, each worker goes through its own "pass", so you have many different workers doing different things on different files on the SUT. However, throughout the workload test run, the configured operations will eventually be performed with sufficient runtime duration, in the absence of unexpected errors or performance issues.

The Port Queue Depth also can place an upper limit on how many outstanding I/Os you can have. The Port Queue Depth setting is applied on a per Workload Generator Port basis. If MPIO is enabled, then the Ports that belong to the same MPIO Group "share" the Port Queue Depth setting. For example, if you have two Ports that are part of one MPIO Group, and your Port Queue Depth setting is 1,024, then there is not 2,048 for the two Ports combined, because the two Ports that are part of the same MPIO Group are treated as a single interface to the upper layer SCSI workload.

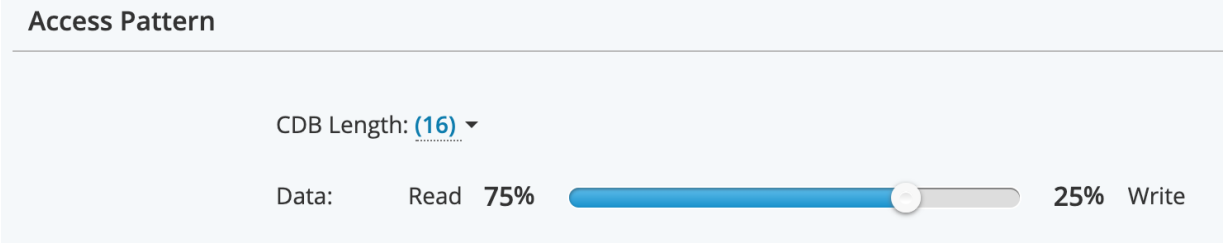
Constant Workloads

Access Pattern

The access pattern has two configurable parameters: The Command Descriptor Block (CDB) length and the read/write ratio.

CDB defines the size of the header block. You can set CDB to 6, 10, 12, or 16 bytes. The default is 10 bytes, which is still observed to be the most common. However, if you are running the workload on a Test Bed that includes a LUN with size greater than 2TB, CDB 16 must be used. To change the CDB Length value, click the **CDB Length** drop-down menu next to the current value and select from the list.

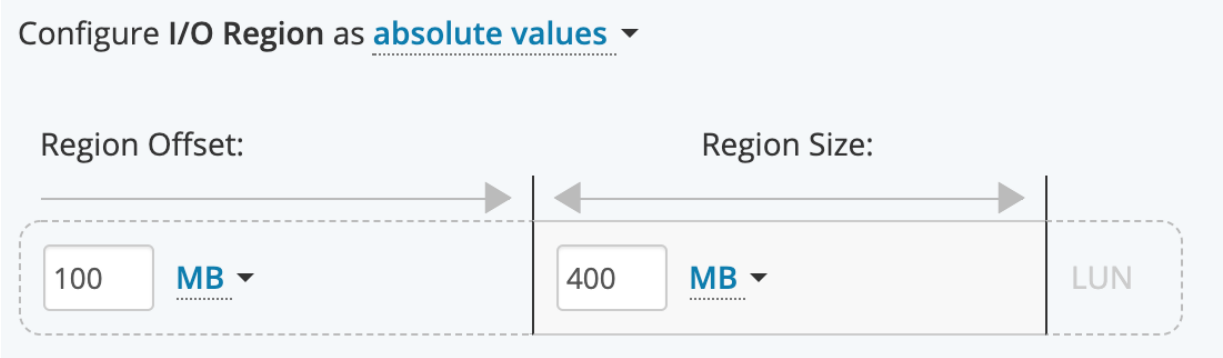
To change the read/write ratio, drag the slider to the right to increase the read percentage or to the left to increase write percentage.

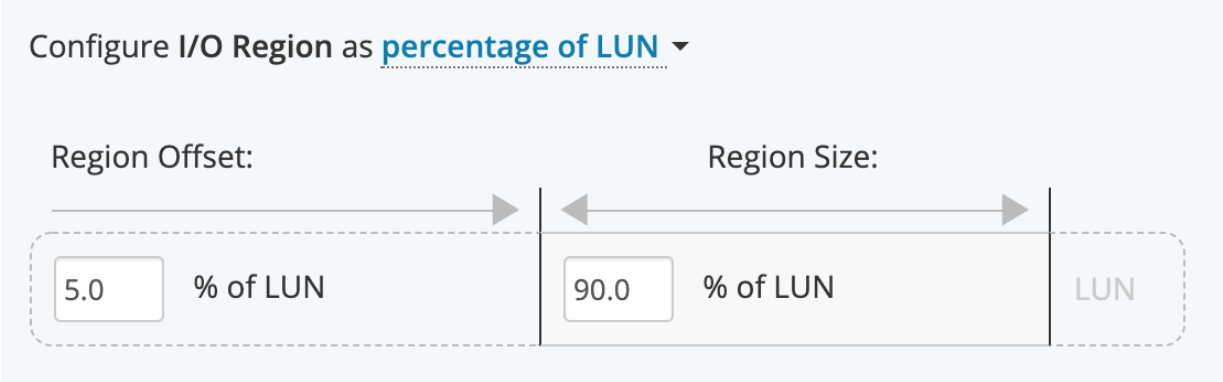


Writes and Reads

Block workloads provide some additional parameters for writes and reads.

Configuration options for writes and reads are described in the Creating a New Workload Test [92] section.





You can specify the I/O region of the LUN using absolute values or a percentage of the LUN for the overall size and offset. To change the method, select the drop-down menu next to the current value.

If you specify an absolute value in bytes, kilobytes (KB), megabytes (MB) or gigabytes (GB), only that sized region starting at specified offset is used. To change the absolute values, edit the offset and region sizes or percentages in the text boxes.

If you specify a percentage, the starting offset and the overall region are calculated by the workload.

If you want to use the same parameters for reads and writes, check the *Use the same parameters as in Writes* checkbox in the *Reads* section.

The **fixed number of Asynchronous I/Os** parameter allows you to specify how many concurrently outstanding I/O requests can be sent for Read and for Write, respectively, for each worker on each Test Bed Link. For example, if you set it to 8 for Writes and 6 for Reads, then you can have up to 8 concurrent asynchronous Writes and up to 6 concurrent asynchronous Reads, totaling up to 14 concurrent asynchronous I/Os for each worker on each Test Bed Link. If you have 2 concurrent workers and 1 Test Bed Link, then you can have up to 28 concurrent asynchronous I/Os. If you have 2 concurrent workers and 2 Test Bed Links, then you can have up to 48 total, but up to 28 per Link.

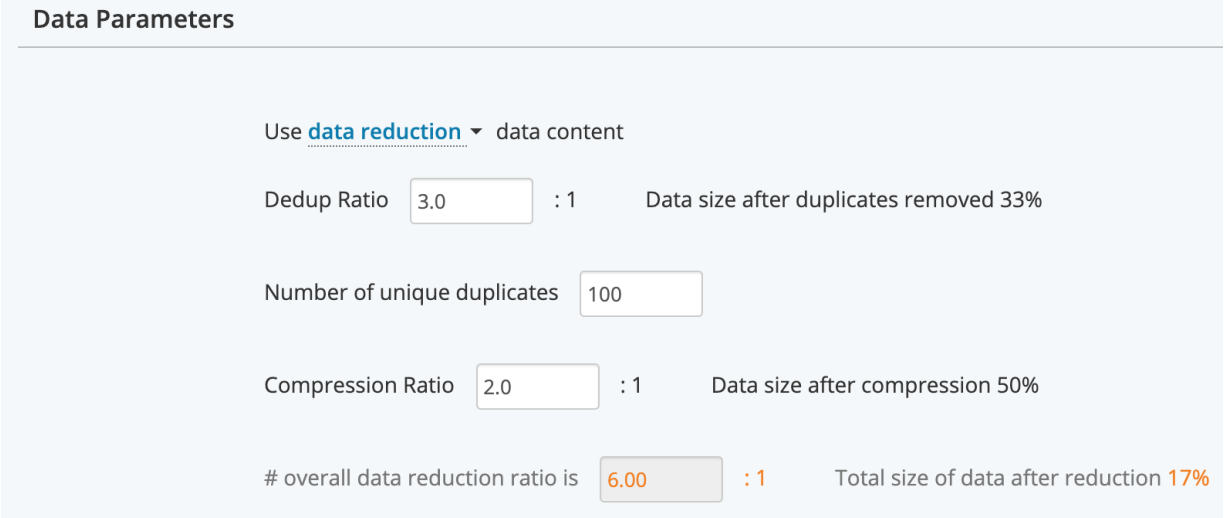
Data Parameters

Block constant workloads support all WorkloadWisdom data parameters options, including the deduplication option of data reduction.

The data deduplication functionality produces unique data content and duplicates of the data content that should yield the user configured data deduplication ratio, intermixed with data content that includes a block of 0's which will be compressible by most algorithms to yield the user configured data compression ratio. A deduplication ratio of 3:1 indicates that for every 3 blocks written only 1 will be unique. Similarly, a 3:1 compression indicates that for every 3 bytes of data on average 1 will be non-zero. The zeros are typically grouped together within the block. Both ratios are rounded to 1 decimal place. The number of unique duplicates determines the size of the pool of reused blocks making up the duplicated data.

The following settings are available for deduplication and compression. The following deduplication option settings control how many duplicate data patterns are used for block workloads:

- *Dedup Ratio.* Ratio of data patterns to be deduplicated versus data patterns that will not be deduplicated.
- *Number of unique duplicates.* Number of unique data patterns in a “pool” of data patterns that are used to draw existing data patterns from to generate duplicated data.
- *Compression Ratio.* Generates a data pattern that causes the system under test to yield the compression ratio that you specify.

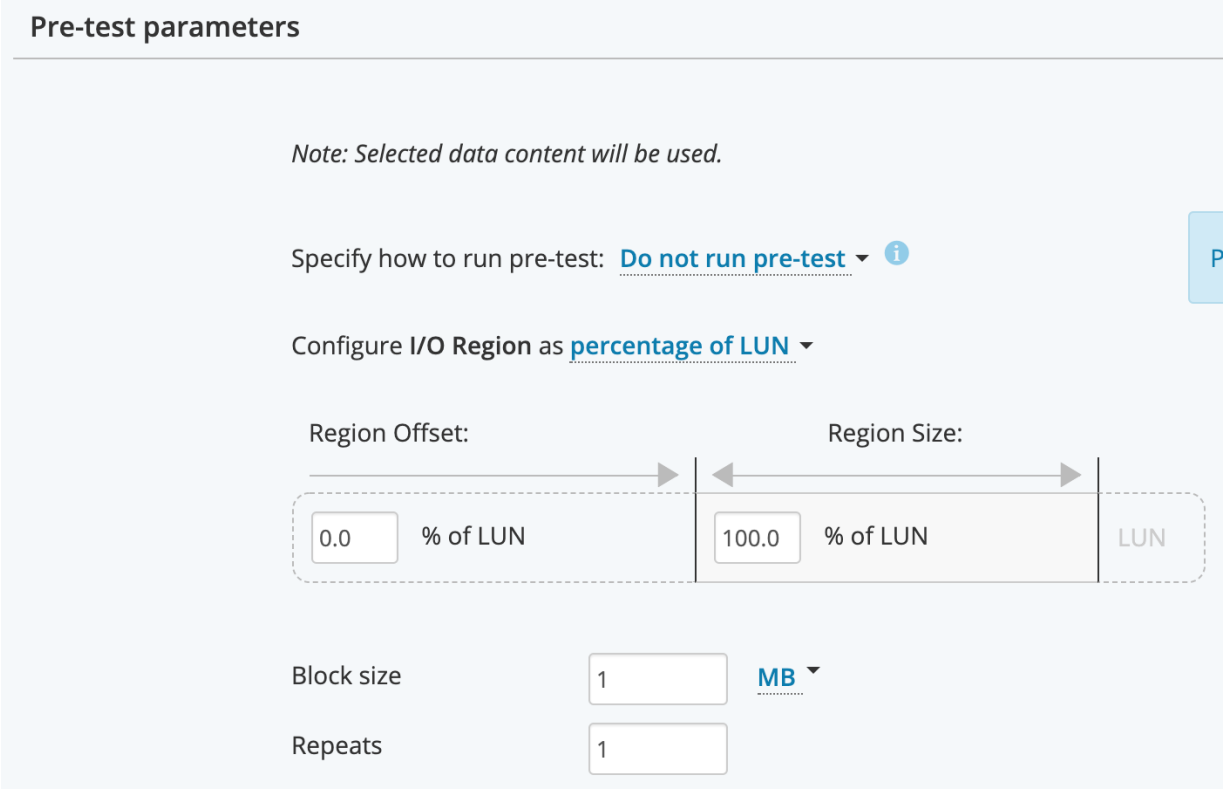


When the workload is running, the generated data pattern is either completely unique (no repeatability) or drawn from a pool of unique, but repeatable data patterns (seeded random sequence) that has been “seen” before to yield the specified *Dedup Ratio*.

For example, if the *Number of Unique Duplicates* is set to `100`, a pool of one hundred unique, but repeatable, data patterns are generated. Throughout the workload test run, whenever a deduplicable data pattern is needed, it is drawn from this pool alongside completely unique data patterns. If you are not sure how many unique duplicates to use, use the default value of `100`.

Pre-test parameters

A pre-test sets up the testing environment and normally includes preparing the file or object system and LUN, so the workload can be run against it. The pre-test section specifies when or it the pre-test runs. For block workloads, it is recommended to set *Do not run pre-test*, and use the preconditioning workload instead. See Preconditioning [57]



The pre-test setting is different for block workloads. Specify the LUN region offset and I/O region. It operates the same way as described in Writes and Reads [0]. You can specify the block size that is used only during the Pre-test, as it may help in speeding up the preconditioning process. There is an additional parameter that you can use to repeat the action for each LUN in the test bed

Fibre Channel

The workload for Fibre Channel supports Multi-Path IO (MPIO). This enables you to test workload load balancing or recovery from a failure scenario.

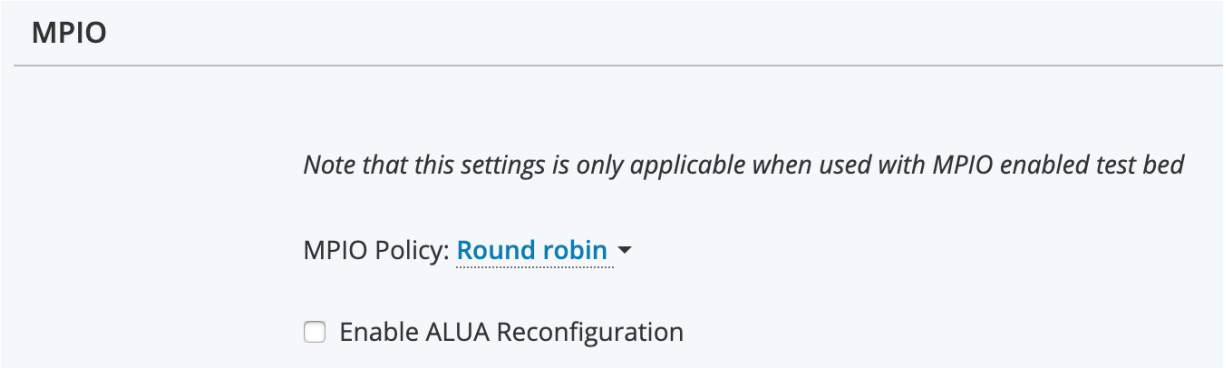
MPIO

MPIO is only available when enabled on the test bed on which the workload is running. There are three MPIO algorithms currently supported:

- *Fail over only*. Ensures the port is redirected to an operational port
- *Round robin*. Distributes the load across all ports participating in MPIO
- *Least queue depth*. Uses the least busy port to send traffic

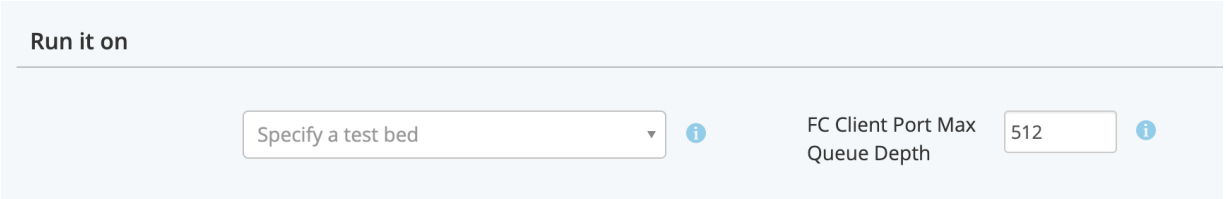
To change the block size approach, select from the drop-down menu next to the current value.

You can enable Asymmetric Logical Unit Access (ALUA) reconfiguration by selecting the *Enable ALUA Reconfiguration* checkbox.



Run it on

Run it on is slightly different for FC workload than for other protocol workloads. It provides the ability to set the client port queue depth. You can use this to determine what is the maximum number of commands that should ever be outstanding on a storage port at any one time. You can also use to find the ideal port *Queue Depth* setting to use for your OS.



The *FC Client Port MaxQueue Depth* is a per port limit of the maximum number of commands enabled onto the network at any point in time. This setting is a limit at the Workload Generator port and applies regardless of MPIO configurations and number of links in the test bed from the port.

Latency is only tracked for commands that are enabled on the network so as not to artificially penalize the storage environment for commands on which it is not enabled to work. You do not have to worry about setting the number of concurrent workers or asynchronous commands too high. You can use the *FC Client Port Max Queue Depth* to control the queueing.

iSCSI

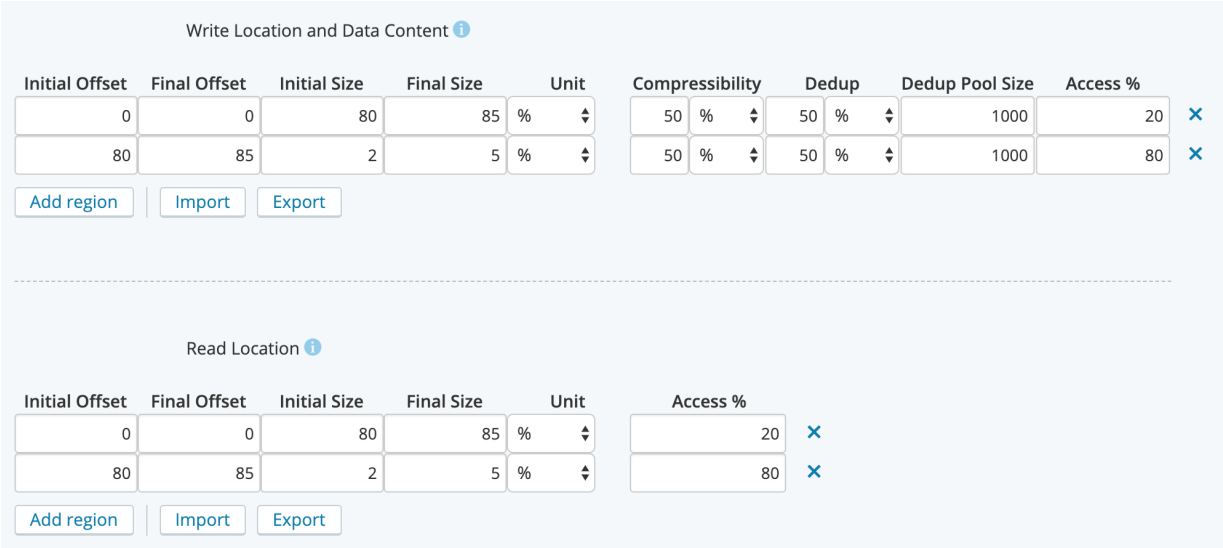
The iSCSI workload model is identical to Fibre Channel, with two exceptions:

- iSCSI workload model does not support MPIO

- iSCSI workload model does not support the *Max Queue Depth* setting

Hotspot Workloads

The FC and iSCSI hotspot workloads are based on the corresponding constant workloads. The hotspot workload adds the ability to specify multiple read and write locations that are accessed.



Each location is accessed the same amount. The same region can be entered more than once, and the locations can overlap. This makes it possible to create hotspot regions within the LUNs. Additionally, the regions move and grow (or shrink) over the duration of the test.

The default example simulates a LUN that 20% of the traffic will initially be to the first 80% of the LUN but will move over the duration of the test to include the first 85% of the test. Another 80% of the access will be initially from 80%-82% of the LUN but then grow and move to 85%-90% of the LUN. This simulates LUN growth with 80% of the activity to the growing region of the LUN. In addition, for the Write portion of the activity, the data content can be specified for the region.

You can export and import the setting for ease of configuration. Use care when importing data as little validation is done when importing. You also should be sure that to the data content matches the required format.

Temporal Workloads

The SCSI temporal workload supports both iSCSI and FC protocols in a single workload. The protocol is based on the test bed configuration. If both FC and iSCSI exist in the same

test bed the default key performance indicator (IOPs, Throughput and Latency) charts are only for the first protocol in the test bed. The other metrics are in the all charts section.

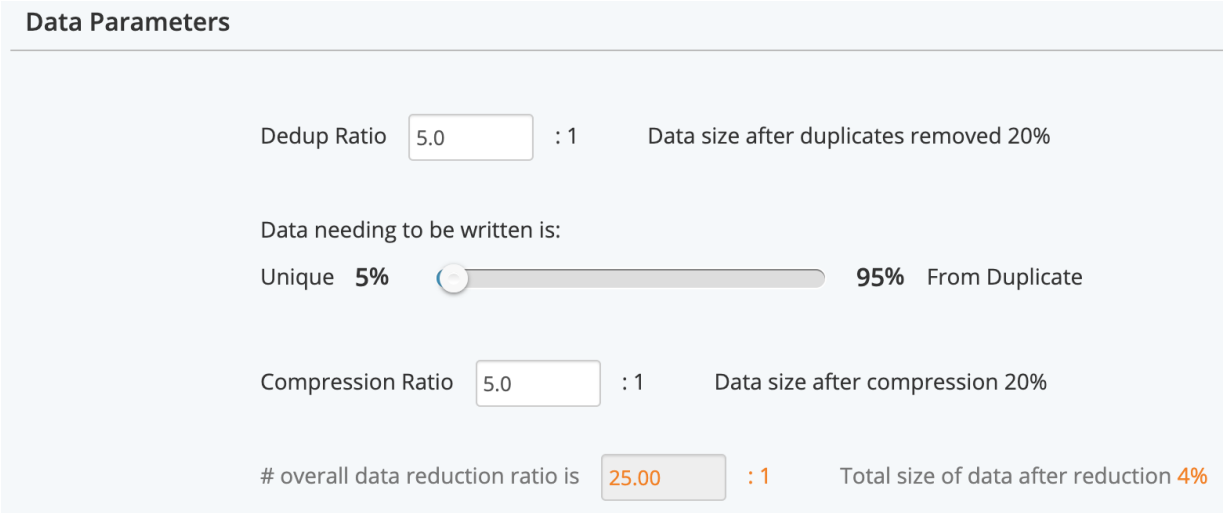
The workload does not appear in the default library because it can only be created from the workload analysis performed on workload data imported into the Workload Data Importer.

Data Parameters

Block temporal workloads support all data parameters available in WorkloadWisdom, including the following implementation for the deduplication option of data reduction.

The following deduplication settings control how many duplicate data patterns are used:

- *Dedup Ratio.* Generates data pattern over the course of the workload test run. Dedup Ratio is unique as it generates data pattern over the course of the test that yields a specified deduplication ratio.
- *Data needing to be written is.* Specifies how much of the data pattern should be completely unique (that is, never repeated) versus the amount of data patterns that are duplicates (that is, repeatable data patterns) throughout the workload test run. File workloads do not have *Data needing to be written is.*



The implementation of the deduplication option for temporal workloads is different from the non-temporal workloads, in that you have the option to have no data pattern that is completely unique by setting the *Unique* percentage to 0%. If it is set to 0, it means you want all data patterns to be repeating patterns (except for the first pass). The workload test generates a pool of 100 unique but repeatable data patterns, and reuses these patterns throughout the workload test run to yield the desired *Dedup Ratio*. When the

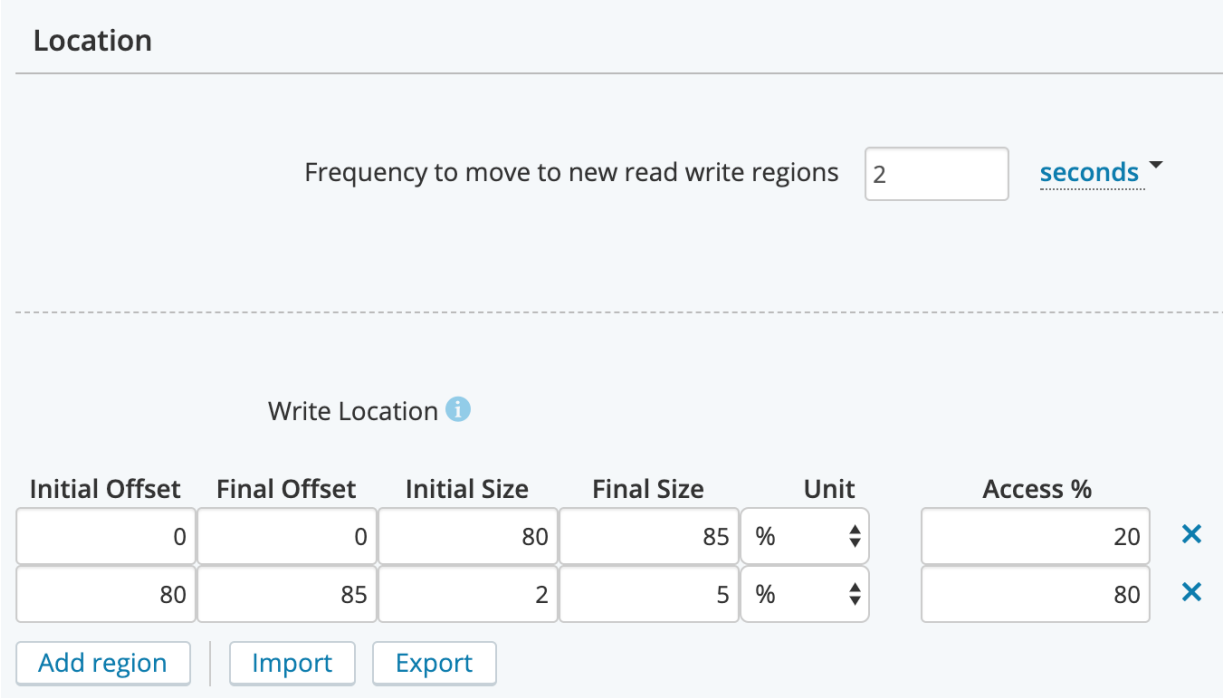
initial pool of 100 unique duplicates is exhausted, another pool of 100 unique duplicates is generated.

For example, if the *Dedup Ratio* is set to 5:1, then the first pool of 100 unique duplicates is generated 5 times to yield a ratio of 5:1, and then a second pool of 100 unique duplicates is generated 5 times to maintain the ratio of 5:1, and so on. If the dedup ratio contains a decimal some of the patterns are written more than the integer portion and others equal to it to arrive at the ratio specified. Setting the *Unique* percentage to 50% with a 5:1 ratio causes each duplicate data pattern to be written 9 times along with one data pattern that is only used once to arrive at a 10:2 or 5:1 ratio.

Location

As with hotspot workloads, the SCSI temporal workload provides the ability to access some portions of the LUNs more frequently than others. It also enables those locations to grow or shrink over time as well as move.

Each concurrent worker is working on only one location at a time and only one LUN for the duration of the test. During the test the worker changes which region it is working on based on the setting to update the offset and size.

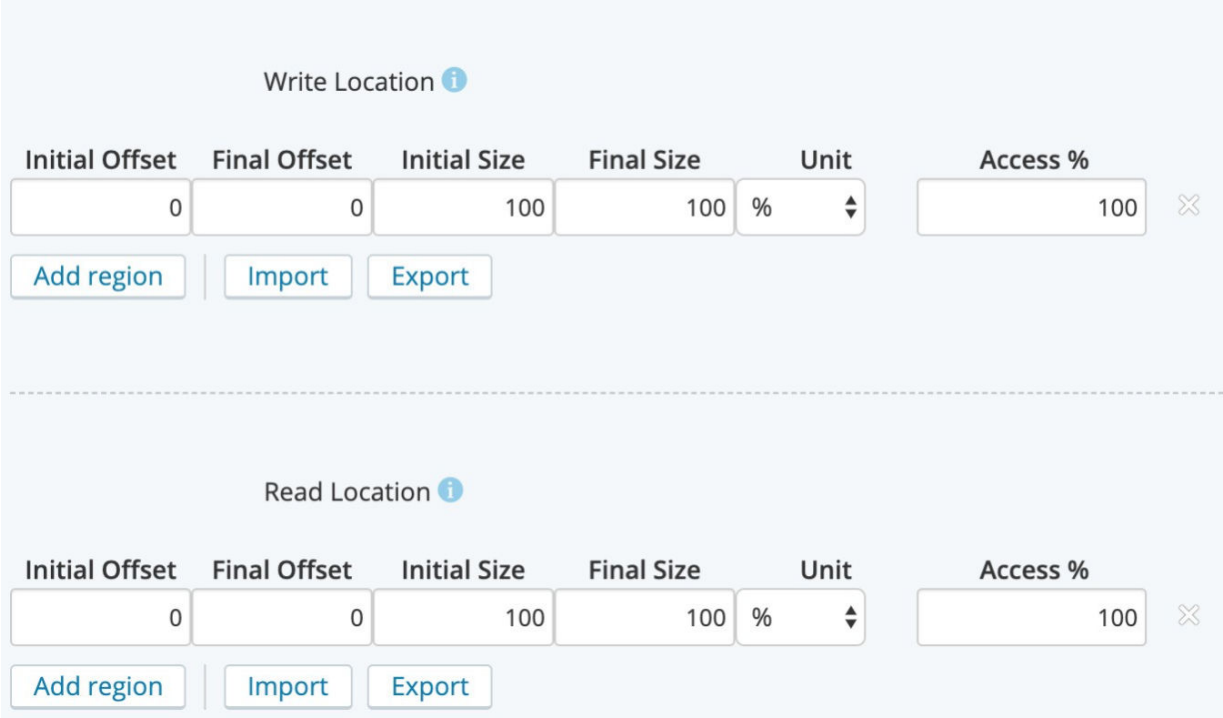


Initial Offset determines where the region begins at the beginning of the test. The offset location for the region moves linearly for the duration of the test until reaching the Final Offset at the end of the test. Similarly, the size of the region is the Initial Size specified and

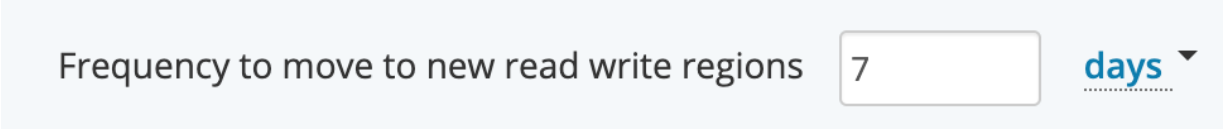
grows or shrink linearly to until the Final Size is reached at the end of the test. Both settings can be either based on a percentage of the LUN or a size specified independent of the LUN size. The Access % determines how frequently the workload is accessing the region as it moves from region to region based on the frequency specified.

Many array architectures are designed to take advantage of hotspot activity that is common to many real-world workloads. Systems that do dynamic tier management rely on this behavior. Some systems might also consider hotspots when doing garbage collection or performing other maintenance tasks. Very few workloads are completely random across and entire LUN so it is important include some level of hotspots when testing.

To run the FC / iSCSI Temporal Workload with the Hot Spots functionality “turned off”, create one Hot Spot entry for Read and for Write, and for each entry, set Initial Offset and Final Offset to 0%, Initial Size and Final Size to 100%, and Access to 100%.



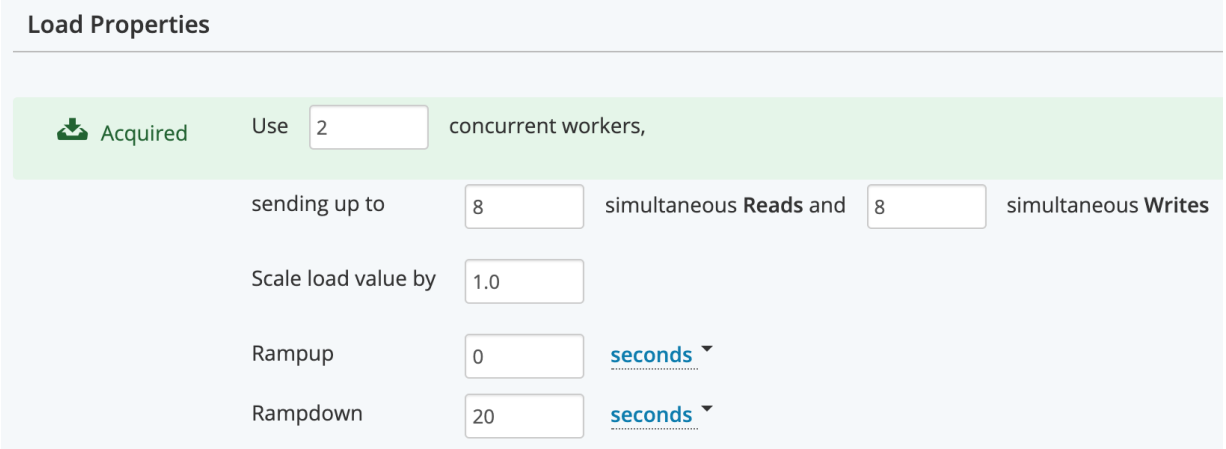
While the **Frequency** setting does not materially make a difference in this case since there are no hot spots, you do have to set it to a value greater than 0.



Load Properties

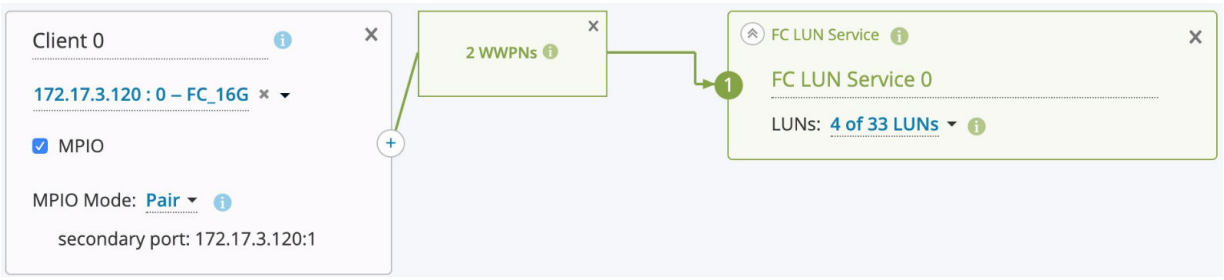
Load Properties for SCSI temporal workloads are somewhat different than for other workloads. The IOPs are automatically changing over time with pattern observed in the production workload. There is no need to set it.

The number of asynchronous requests was also moved to this section as it is closely related to the number of concurrent workers in determining how many requests can be outstanding at any point in time. This queue can be determined by the number of concurrent workers and how many total requests can be set up to 2048. Setting the combined value higher than 2048 artificially increases latency but might be needed to cover all LUNs.



The workload can be scaled for “what if” scenarios and this value can be used in conjunction with the iteration.

When you enable MPIO in a workload (which implies the Test Bed has MPIO enabled), the Load Properties settings will treat the Workload Generator Ports that belong to the same MPIO Group as if they were a single logical Workload Generator Port.



In the above example, each of the 4 LUNs will be accessed by 2 workers concurrently, and if the load is 400MB/s, then a total of 400MB/s will be generated from the MPIO Group.

Exactly how much of the 400MB/s is generated by each of the two physical Workload Generator Ports that make up the same MPIO Group depends on the MPIO Policy that is set. If it is round-robin, then each Port will generate 200MB/s, all else being equal. If it is failover, then the primary port will generate 400MB/s.

Block Protocol FC-NVMe Workloads

FC-NVMe workloads emulate the process of accessing NVMe block devices across a Fibre Channel fabric. For NVMe access, there are far fewer metadata commands relative to SCSI access. This is because the NVMe specification was designed from scratch to be simple, scalable, and optimized for modern flash storage devices. To characterize NVMe data it is important to understand the basic command mix, if the data is being accessed sequentially or randomly, the I/O request sizes, any hotspots, data content, how namespaces are being used, and the effect of NVMe queues.

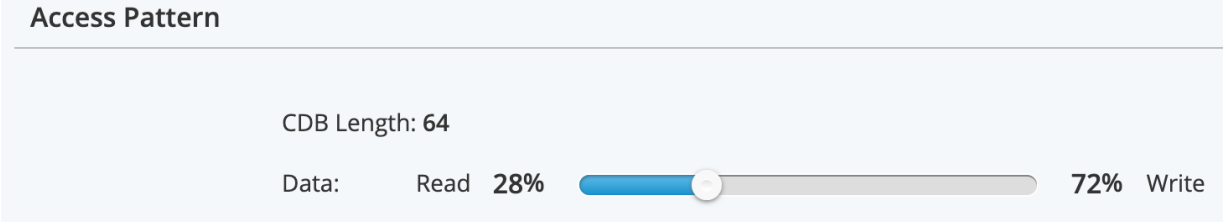
NOTE
You can use FC-NVMe workloads in Composite Workloads, Workload Suites, and Iteration Suites. However, you cannot mix FC-NVMe workloads with non-FC-NVMe workloads in the same Composite Workload or the same Workload Suite.

Constant Workloads

Access Pattern

The access pattern has one configurable parameter: the read/write ratio. The CDB Length is not yet configurable, but it provides a read-only display to confirm the CDB Length that is used for the workload.

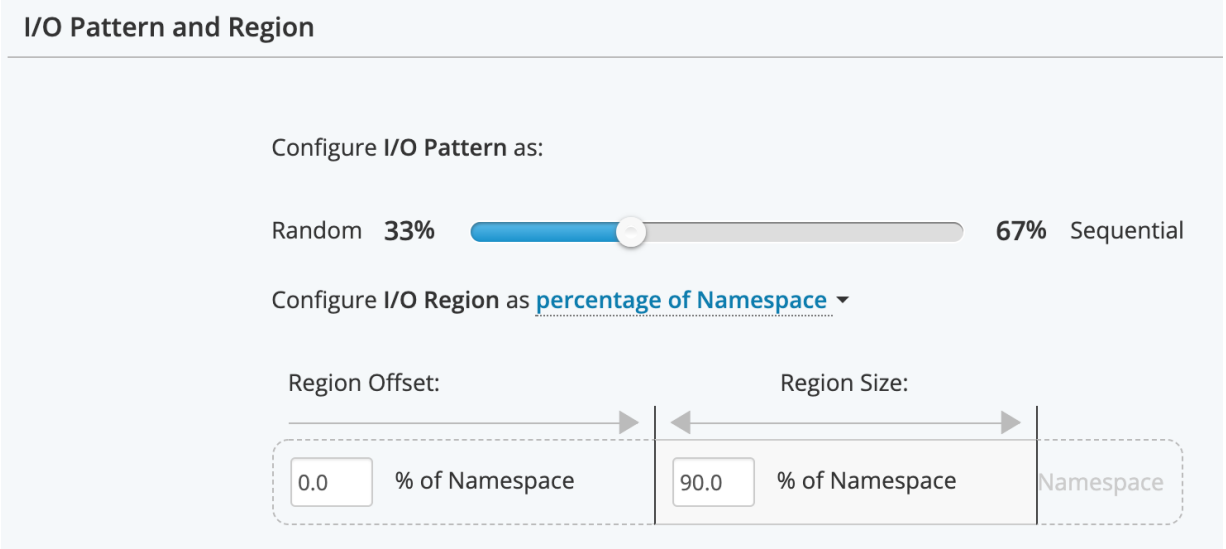
To change the read/write ratio, drag the slider to the right to increase the read percentage or to the left to increase write percentage.



Writes and Reads

Using the writes and reads section, you can configure the block sizes, percentage of random versus sequential behaviors, and several other parameters.

There is a separate section in the UI for writes and reads, the configuration options are similar. See Creating a New Workload Test. FC-NVMe workloads provide some different parameters and terminologies.



You can specify the I/O region of the Namespace using absolute values or a percentage of the Namespace for the overall size and offset. To change the method, select the drop-down menu next **Configure I/O Region as**.

If you specify an absolute value in bytes, kilobytes (KB), megabytes (MB) or gigabytes (GB), only that sized region starting at specified offset is used. To change the absolute values, edit the offset and region sizes or percentages in the text boxes.

If you specify a percentage, the starting offset and the overall region are calculated by the workload.

If you want to use the same parameters for reads and writes, check the *Use the same parameters as in Writes* checkbox in the *Reads* section.

Data Parameters

FC-NVMe constant workloads support the following data parameters options:

- *Constant*: all zeroes

- *Sequential*: incrementing counter of a 32-bit number
- *Random*: randomly generated numbers that are not reproducible
- *Seeded Random*: randomly generated numbers that are reproducible with the same seed value
- Data Reduction
  - *Dedup Ratio*: the desired resulting ratio of transmitted data versus stored data after duplicated data is removed.
  - *Duplicate uniqueness percent*: the desired percentage of duplicate uniqueness, from 0% to 100%.

When writing dedup patterns, duplicates are written in order to meet the desired dedup ratio. These duplicates can be mostly identical (minimal duplicate uniqueness) or can mostly be unique (maximal duplicate uniqueness). A small DupUniq value leads to a small amount of device space being required to store duplicates. A large DupUniq value leads to a large amount of device space being required to store duplicates.

- *Compression Ratio*: the desired resulting ratio of transmitted data versus stored data after compressible data is compressed.

NVMe Settings

Specify the total number of concurrent NVMe Queues and the total depth of each NVMe Queue.

- *Queue Count*: default is 32. Max is 32.
- *Queue Depth*: default is 1,024. Max is 1,024.

Preconditioning

A dedicated Preconditioning workload is not yet available for FC-NVMe. To precondition a device, simply run an FC-NVMe workload test with 100% write and the desired data content, and set the I/O Region to 100% of the Namespace.

MPIO

MPIO is only available when enabled on the test bed on which the workload is running. There are two MPIO algorithms currently supported:

- *Fail over only*. Ensures the I/O is redirected to an operational port.
- *Round robin*. Distributes the load across all ports participating in MPIO.

VDI Workloads

There are two Virtual Desktop Infrastructure (VDI) workload models that simulate key VDI loads:

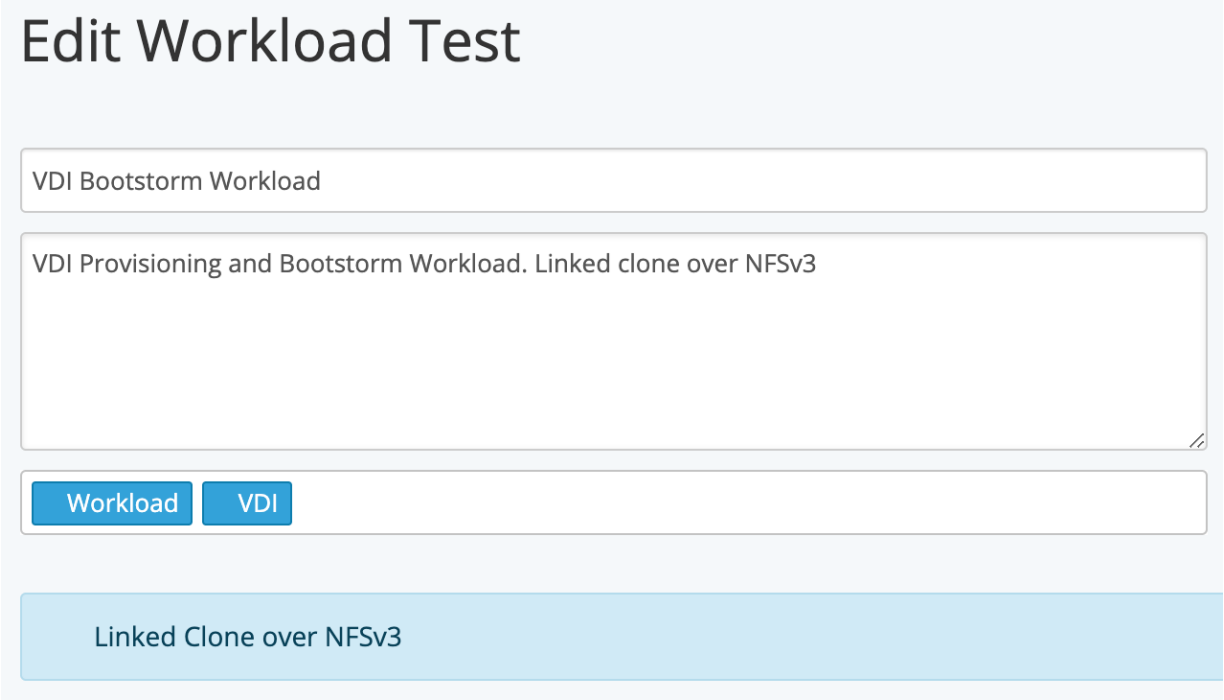
- *VDI Bootstorm*. Simulates booting a number of VDI guests across NFSv3
- *VDI Runtime over both FC & iSCSI*. Simulates the runtime impact of VDI guest.

When a Virtual Machine (VM) is assigned to a desktop pool, a clone (thin size) of the VM is created. The thin clone is called replica. After the replica is created, it starts creating Linked Clones (LC). An LC is an empty disk that, over time, grows, according to block changes requested by the Windows GuestOS. The replica disk is read-only and used as primary disk. The writes and/or block changes are written/read from the replica disk.

VDI workloads are designed to provide an interface that can be easily understood by a VDI administrator. VDI parameters are expressed in terms of pools, replicas and linked clones.

VDI Bootstorm under NFSv3

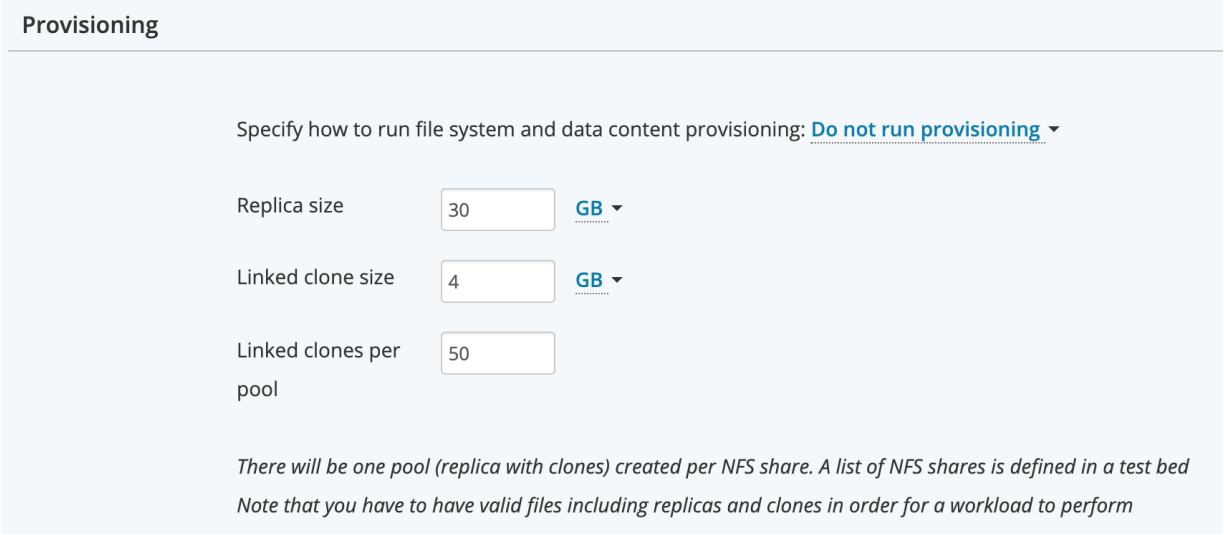
The VDI bootstorm workload provides five sections: workload description, provisioning, bootstorm, tracing parameters, and runtime parameters. For information regarding workload description, tracing parameters and runtime parameters, see Creating a New Workload Test [92].



The workload description highlights the fact that VDI bootstorm is using linked clones over NFSv3.

Provisioning

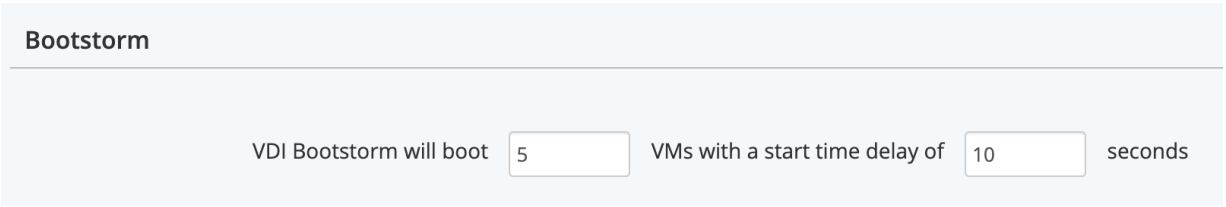
Provisioning functions like the pre-test in a protocol workload and sets up the environment to emulate the bootstorm. You specify when provisioning is run, the Replica (or guest image size), whether it's a linked clone and the number of linked clones per pool. There can only be one pool per NFS share, so if you want to create multiple pools you need to ensure the test bed is configured to use as many pools as you have shares.



To change when provisioning is run, the down arrow next to the current value and select the value from the drop-down list displayed above.

Bootstorm

Use the bootstorm section to specify how many guests boot and the delay between each boot.



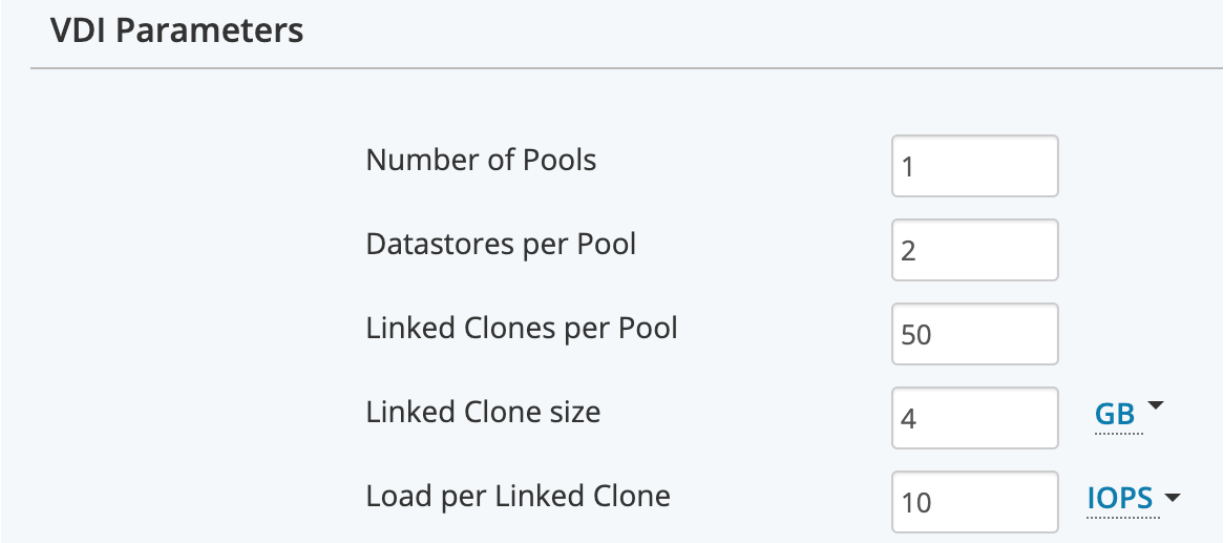
VDI Runtime FC & iSCSI Workload

This workload simulates run time activities of multiple desktops based on the linked clone model and is available for both Fibre Channel and iSCSI.

Linked clones are combined in pools, with 2000 linked clones per pool, maximum. Linked clones are located in datastores which represent a FC or iSCSI LUs in this model. The number of pools is limited by the number of appliance physical ports used in the test bed. There is a 1-1 relationship between the pool and the physical port. This workload generates read/write operations distributed randomly within the region of the LUs occupied by Linked Clones. The size of these operations and their rate vary according to a statistical model trained on experimental data.

VDI Parameters

Use the VDI parameters for the runtime workload to specify the number of VDI pools, datastores per pool, and linked clone characteristics including the load in IOPs per linked clone.



The workload configuration specifies the *Load per Linked Clone*, which is the desktop user activity in IOPS. To change the unit of measurement for *Load per Linked Clone* the unit of measurement is measured select the down-arrow next to the current value. The default of 10 IOPS corresponds to a light user and 80 IOPS corresponds to a heavy user.

Linked Clone Size represents a “delta” between a particular desktop and the replica. The Linked Clones are modeled as non-deduplicable data patterns located on the Datastore.

As an example, consider a VDI setup consisting of 4 pools with 500 Linked Clones. The size of the Linked Clone is 4 GB, the *Datastore per Pool* is 4, and the *Load per Desktop* is 20 IOPS (medium activity).

This configuration should be supported by a test bed with Physical Ports. As a result, 2000 desktops will be simulated generated overall storage traffic of 40,000 IOPS distributed among 4 Ports. The overall storage capacity required to support this test will be 16 LUs, 500 GB each. 500 GB of each LU will be used as the space to which IO commands will be addressed.

Composite Workloads

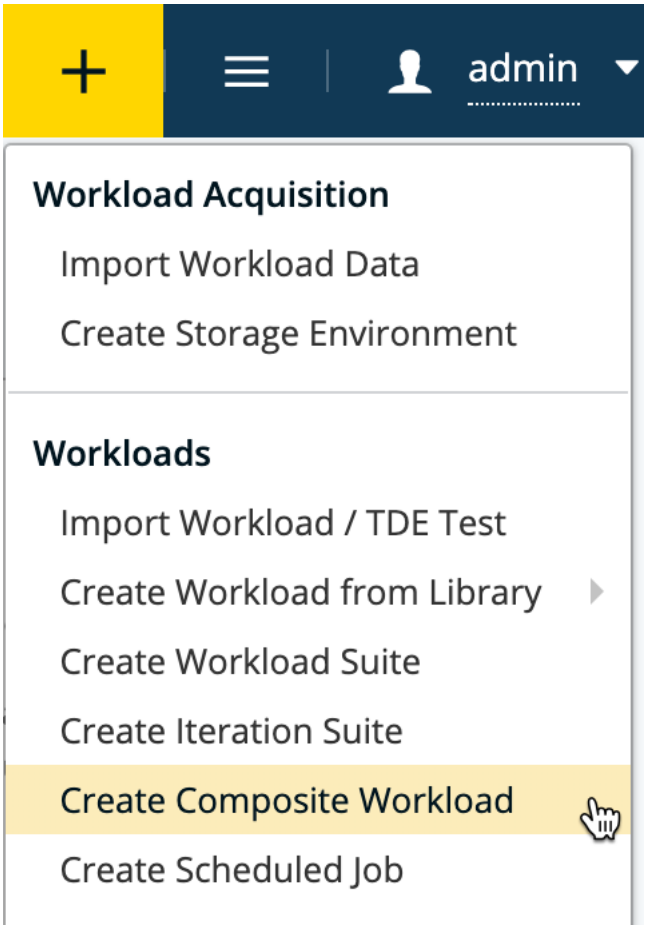
A Composite Workload (CW) is a powerful framework that enables you to run multiple protocol workloads with different workload characteristics at the same time. Good examples of Composite Workloads are:

- Application with several different I/O characteristics or profiles, for example, database workloads.
- Application accessing different IO Regions, for example, VDI infrastructures.
- Multi-protocol workloads, each with its own protocol workload characteristics, accessing one or more SUT Services. A good example of this is a rack of virtualized servers in a data center.

You can use the Composite Workload Editor (CWE) to create a composite workload. The CWE is designed to support up to 100 workloads and up to 100 Links in each Composite Workload (that is, a Composite Workload made up of 100 workloads or 100 links, whichever limit is reached first).

New Composite Workload

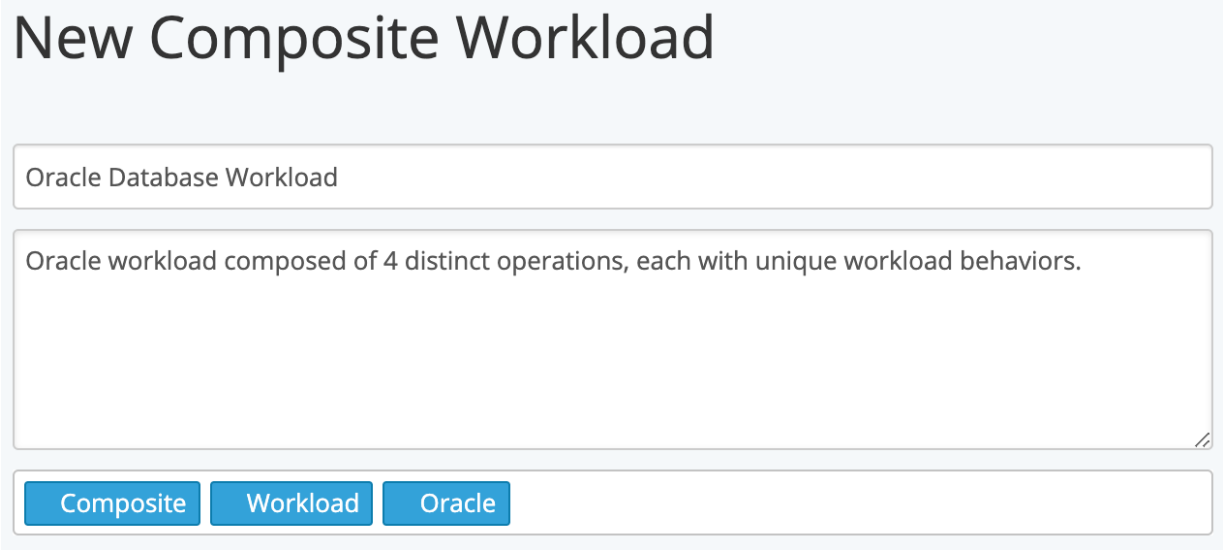
Access the *New Menu* page to create a new Composite Workload:



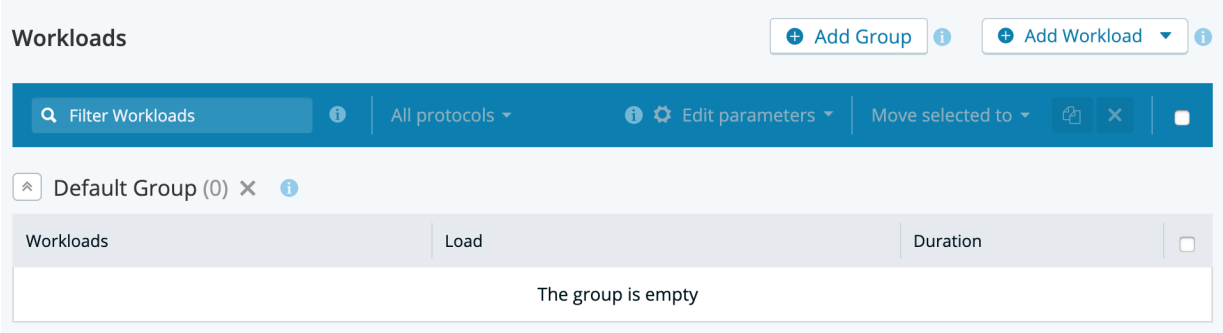
Assign a name to the Composite Workload to easily identify this Composite Workload from the list of all Composite Workloads.

Optionally add a description to the Composite Workload that briefly describes the purpose or content of this Composite Workload.

Optionally add one or more Tags to describe this Composite Workload. Tags are searchable in WorkloadWisdom, in addition to the name of the Composite Workload.



Workloads

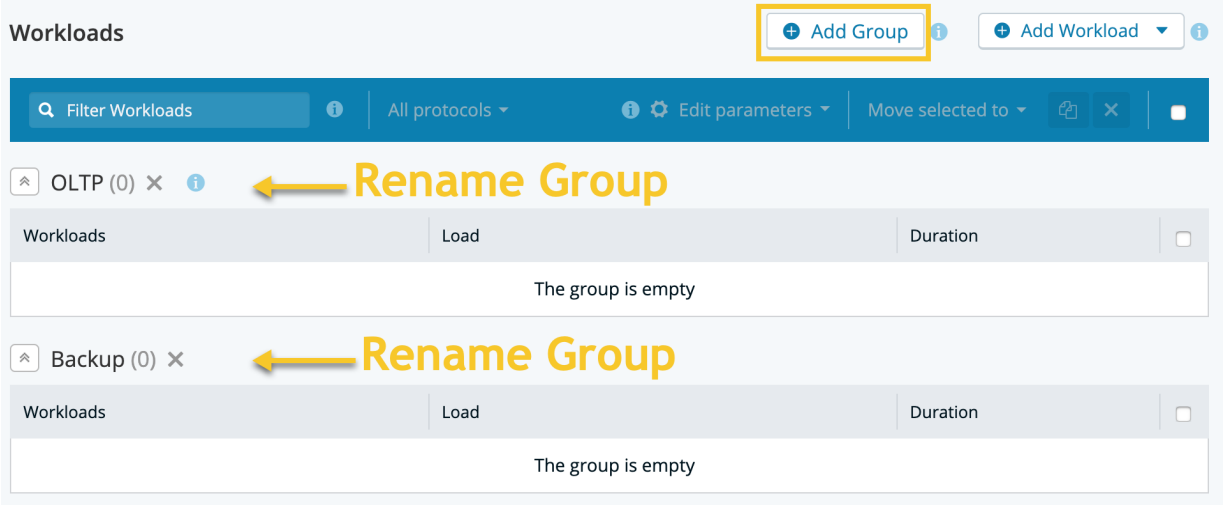


Begin by defining the structure and content of the Composite Workload. The structure of a Composite Workload is one or more logical Groups, in which each Group contains one or more Workloads. Use the logical group to better manage the workloads you have defined for the Composite Workload. For example, if you are creating a Composite Workload that simulates a VDI deployment, you can create two Groups, named “Infrastructure” and “Clients”, and then define a set of workloads that simulate the I/O profiles of the VDI Infrastructure operations and a set of workloads that simulate the I/O profiles of the VDI Clients.

Add Group

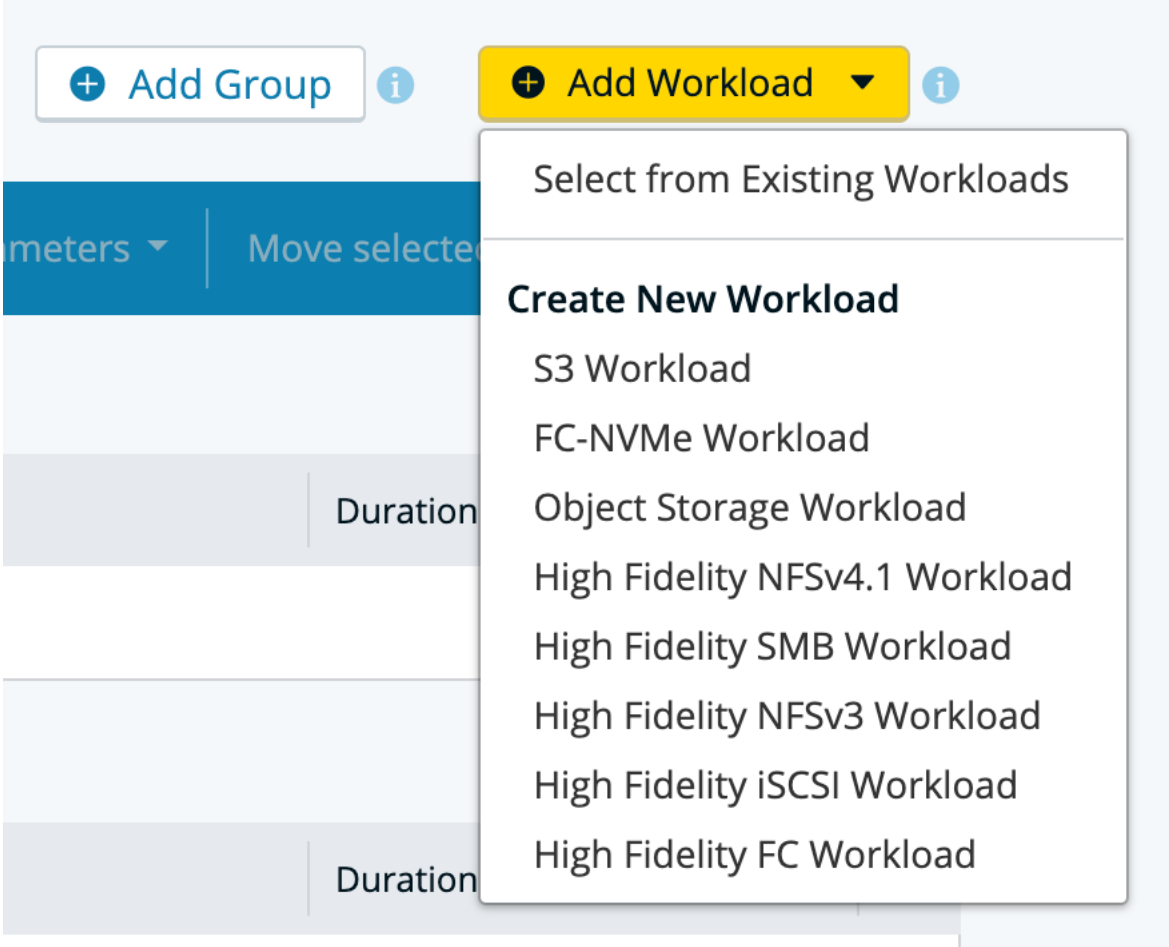
One default Group is created, and it cannot be deleted. Click the Group name to give the Group a name that is meaningful to you.

Optionally, click **Add Group** to create a new logical group.



Add Workload

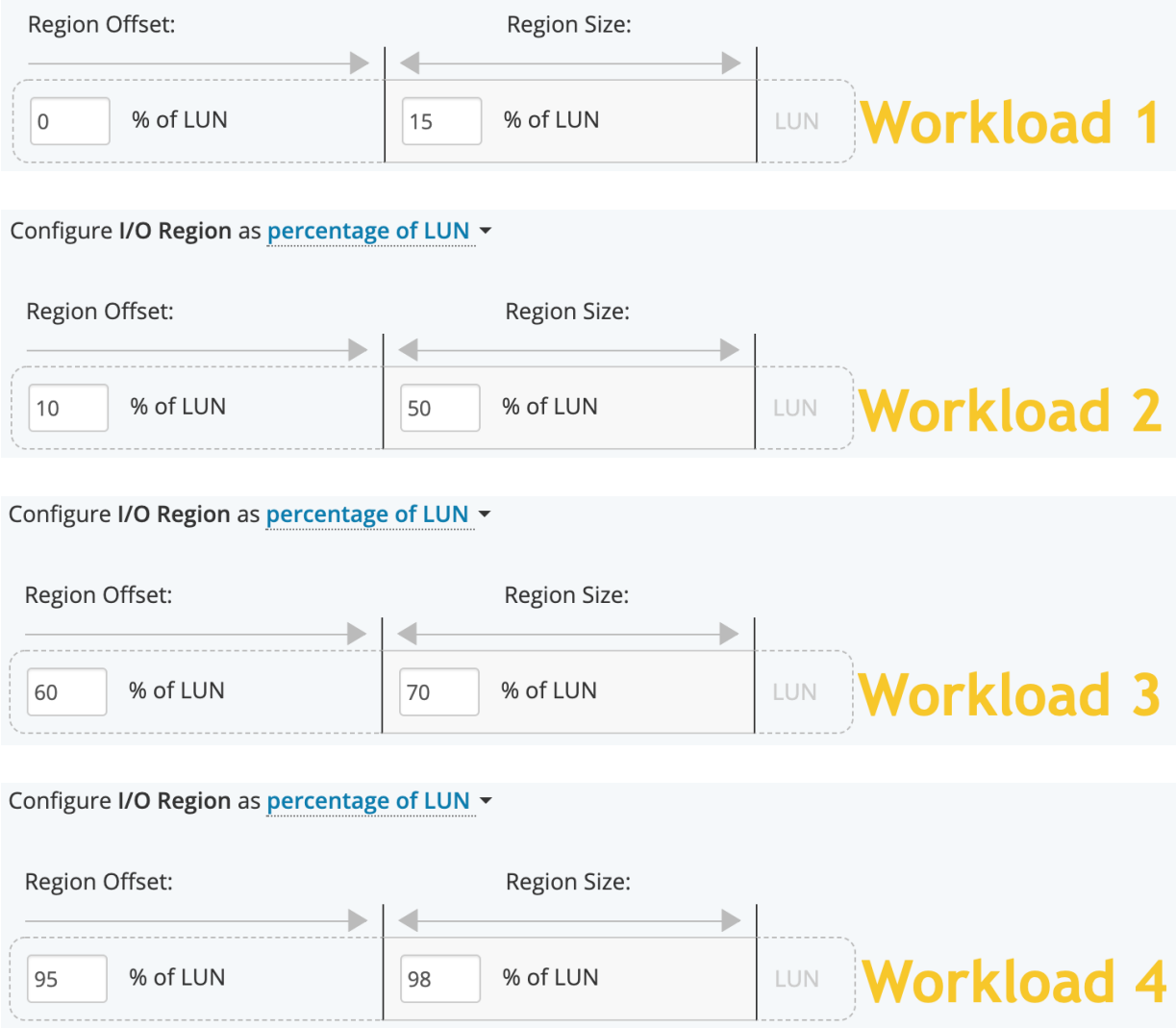
Once you have identified the I/O profiles for all the workloads that make up this Composite Workload, click Add Workload to begin defining the Workloads.



You can add workloads using two methods:

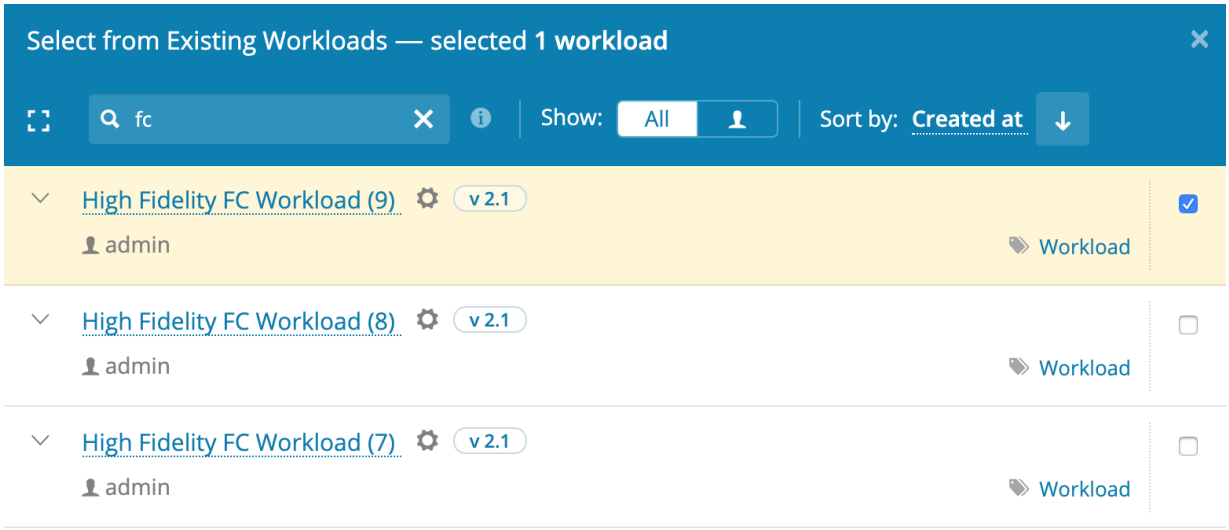
- *Select from Existing Workloads*. Use this option if you have already created the workloads outside of the CWE. This option will bring up a browser where you can select the workloads you want to use and specify the Group to add the selected workloads to.
- *Create New Workload*. Use this option if you have not already created the workloads outside of the CWE. This option will bring up a window where you can Clone from the Workload Library.

In the following example, assume you have already created four workloads outside of the CWE, each with a different load focusing on a different LUN Region, and you want to add the first Workload to the first Group, and the other three workloads to the second Group.

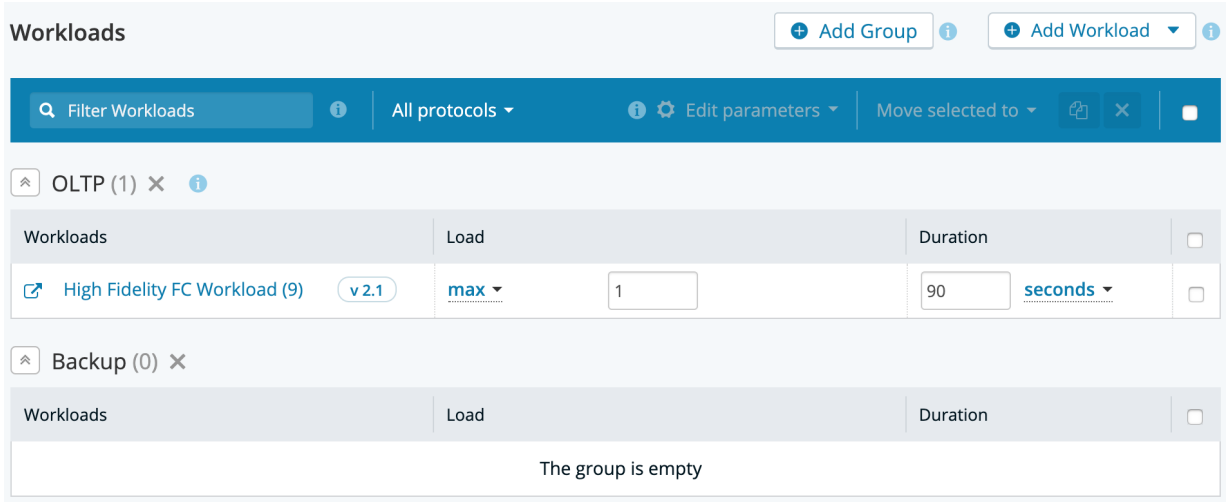


Click *Select from Existing Workloads* to choose existing workloads.

Select the workload for the first hotspot to the first group using the checkboxes and the *Target Group* selector.

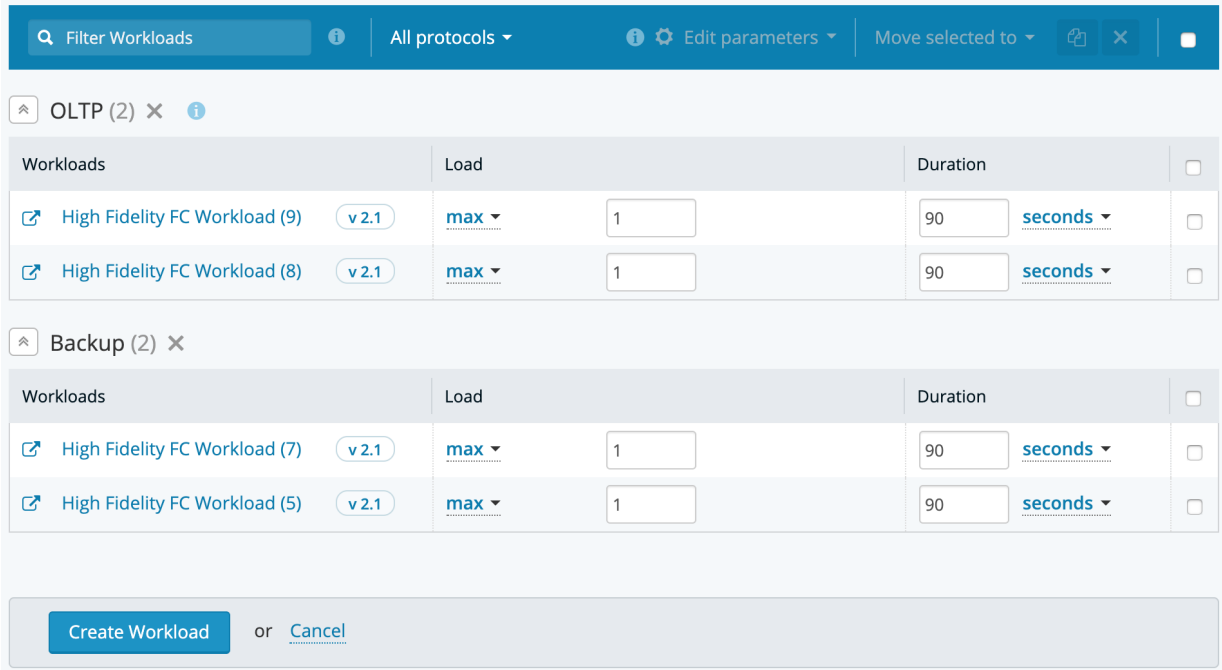


You return to the CWE main page, which displays the changes you made.



Return to the *Select from Existing Workloads* browser to add the remaining three workloads across the two groups. You can also select multiple workloads at the same time.

You should now have a total of two groups and four workloads in your composite workload.



You can also specify and change the load profile for each workload. The Load Profiles you set for each Workload in the Composite Workload does not overwrite the Load Profiles saved in each Workload outside of this Composite Workload. Instead, the changes are within the scope of the Composite Workload itself. The original workloads can be used separately without unknown or unexpected changes by you or other users while working with a Composite Workload that is built from these workloads.

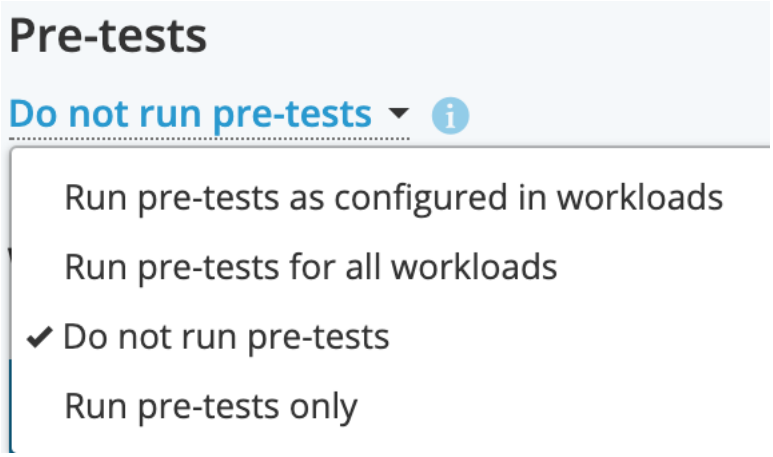
Pre-tests

NOTE

Beginning with version 5.3, Preconditioning Workloads phased out pre-tests across a number of releases. While existing pre-tests remain functional without changes, it is recommended that you use preconditioning workloads if they are available. Preconditioning workloads are available for FC and iSCSI. See Preconditioning [57] for information regarding preconditioning workloads.

Next, define how you want the Pre-tests to run. Again, the concept of a Pre-test is to precondition the LUNs (for Block) and populate folders and files (for file). Pre-test is not

mandatory if you have already preconditioned your LUN or populated the folders and files separately. Pre-test is recommended if you have not preconditioned your LUN or populated the folders and files.

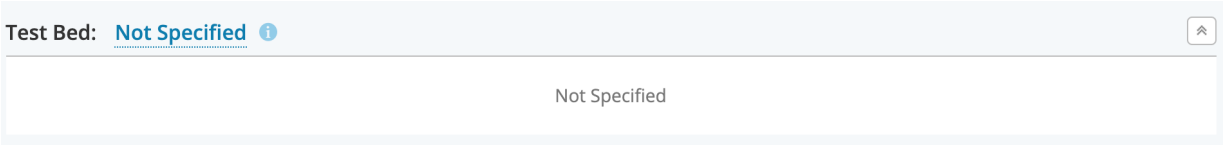


- *Run pre-tests as configured in Workloads*. Use this option if you want some of your workloads to run a Pre-test but not others. In this case, the CW will honor each Workload's Pre-test setting.
- *Run all pre-tests before Composite Workload*. Use this option if you want to force the Pre-test to run for all workloads once prior to running the CW itself. In this case, the CW will not honor each Workload's Pre-test setting, but instead force the Pre-test to run once for all Workloads.
- *Do not run pre-tests*. Use this option if you do not want any Pre-test to run at all, and start running the CW itself.
- *Run pre-tests only*. Use this option if you do not want to run the CW itself, but only the Pre-tests instead.

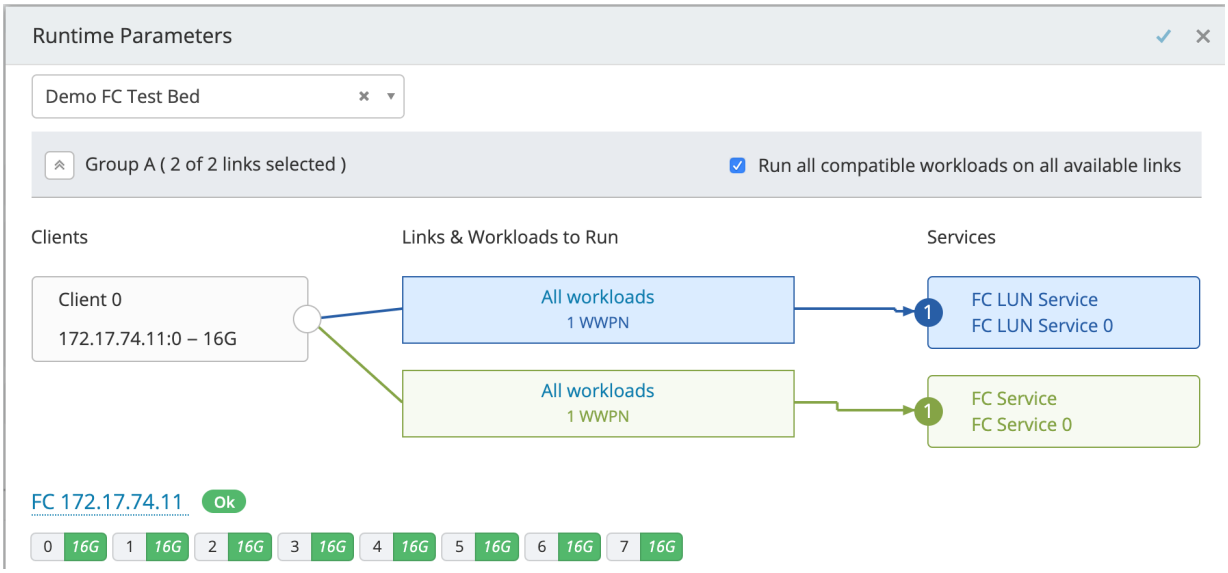
Test Bed

Finally, you need to select the test bed on which the CW runs.

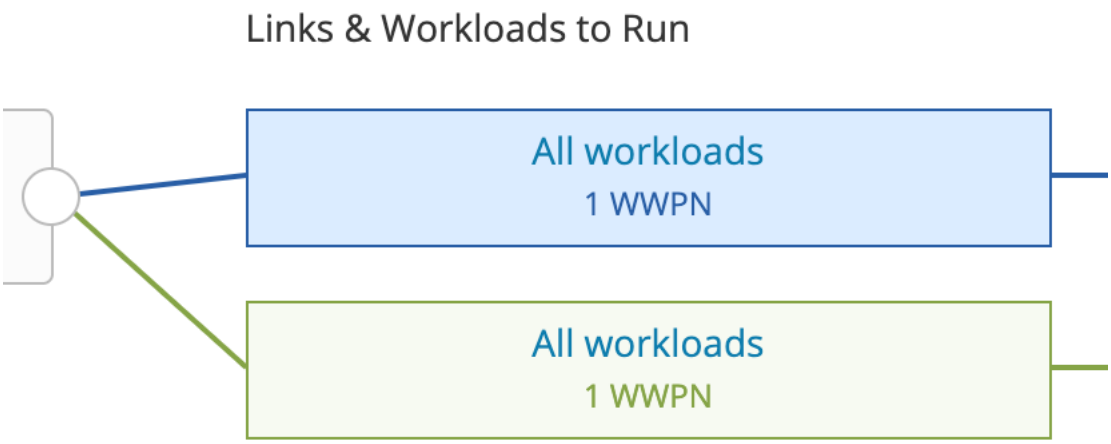
Click the *Test Bed* link, which is not specified by default, to bring up the test bed browser for this CW.



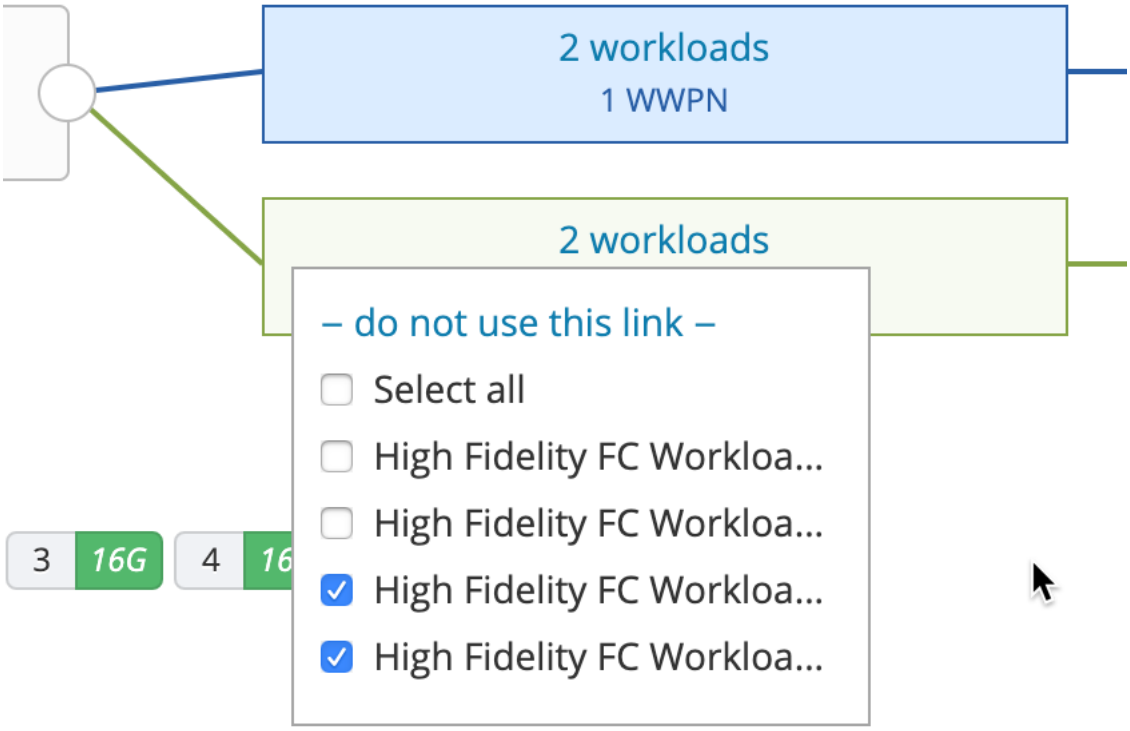
Select an applicable test bed from the drop-down menu.



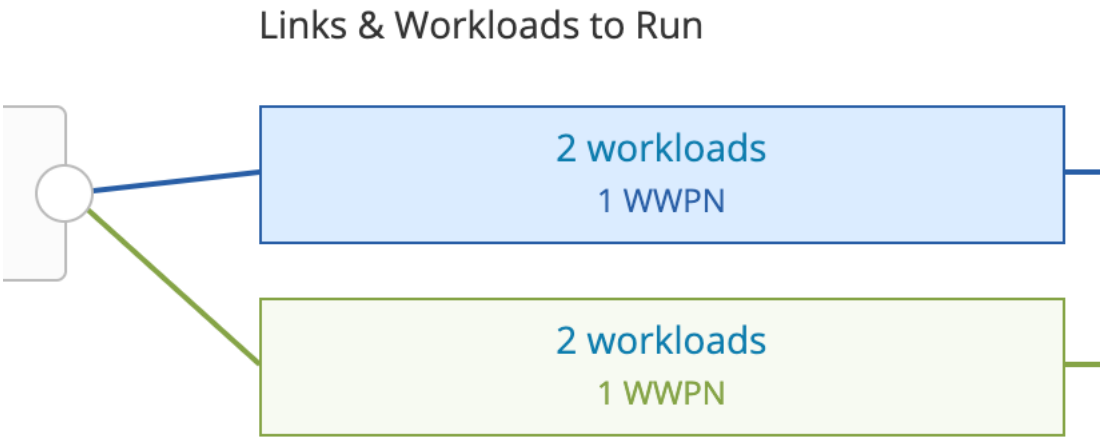
By default, all applicable workloads are applied to all available links in the test bed. In this example, four FC workloads are defined in the CW, and the selected test bed has two FC Links. Therefore by default, all four FC workloads run on both FC Links.



If you want to change the default such that only some workloads run on some links, click the link or links to specify which workloads are applied to each link.



In this example, the workloads for the first two hotspots are applied to the link connected to FC Service 0, and the other two workloads for the other hotspots are applied to the link connected to FC Service 1.



Click *Update Runtime Parameters* to apply the changes.

Start Composite Workload

Click Start to access the Start Composite Workload page, where you can specify runtime options each time. If you do not want to change the runtime options, click the Start.

Description

Optionally add text to describe this instance of the composite workload run. The text entered is displayed and saved in the *Results Dashboard* for this instance of the composite workload run.

Tracing Parameters

You can optionally enable tracing for a composite workload when you run a test. If you set it to disabled, tracing is turned off for all ports used in the composite workload. If you set it to enabled, tracing is turned on for all ports in the composite workload. See Tracing Parameters [108].

Test Configuration

- *Retrieve summary file.* Downloads a proprietary Virtual Instruments summary file at the end of the test run. Only enable this option when requested by Virtual Instruments Support.
- *Enable per-LUN Stats in Log.* Produced a log file at the end of each test run. If you are running an FC or iSCSI Workload, you can optionally enable per-LUN statistics in the log file, which provides statistics on a per-LUN basis. Enabling this has an impact on performance, so it is recommended to use this option only when you are troubleshooting.

Run it on

Select the test bed that you want to run the Composite Workload on. If you have defined a test bed already in the Composite Workload, then that test bed is displayed by default. You can optionally use a different test bed.

Conditions & Actions

Optionally select one or more conditions you have created to apply to the Composite Workload run. See Conditions & Actions [202] for details.

Modifying Composite Workloads

One of the key functionalities of the Composite Workload Editor is to perform a change across multiple workloads at once.

Filters

You can use filters to narrow down available workloads to a smaller set.

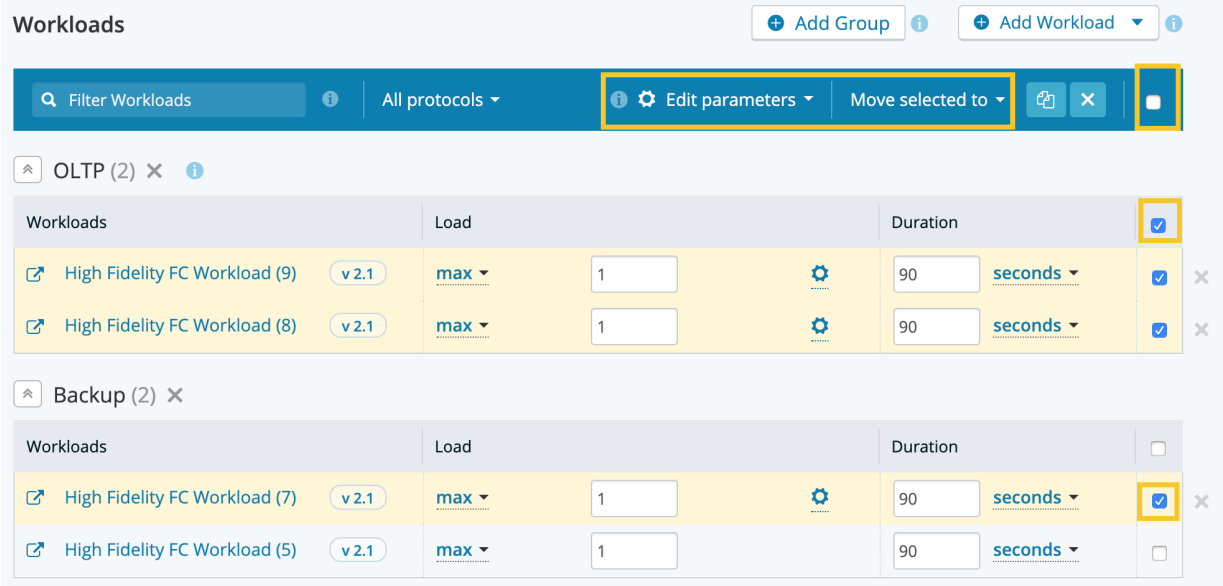
Use the Protocols Filter to filter workloads based on their protocol type. By default, the Protocols Filter is set to All Protocols.

Use the Name Filter to filter workloads based on their name.

Selector

The checkbox Selectors are available at different levels, giving you the ability to select workloads at different granularity. You can select all workloads in the Composite Workload, or all workloads in a Group, or a specific Workload.

The Selector only selects workloads that are available based on the filter settings. If you apply a filter and then use the selector, you might not be selecting all workloads that you intend to select.

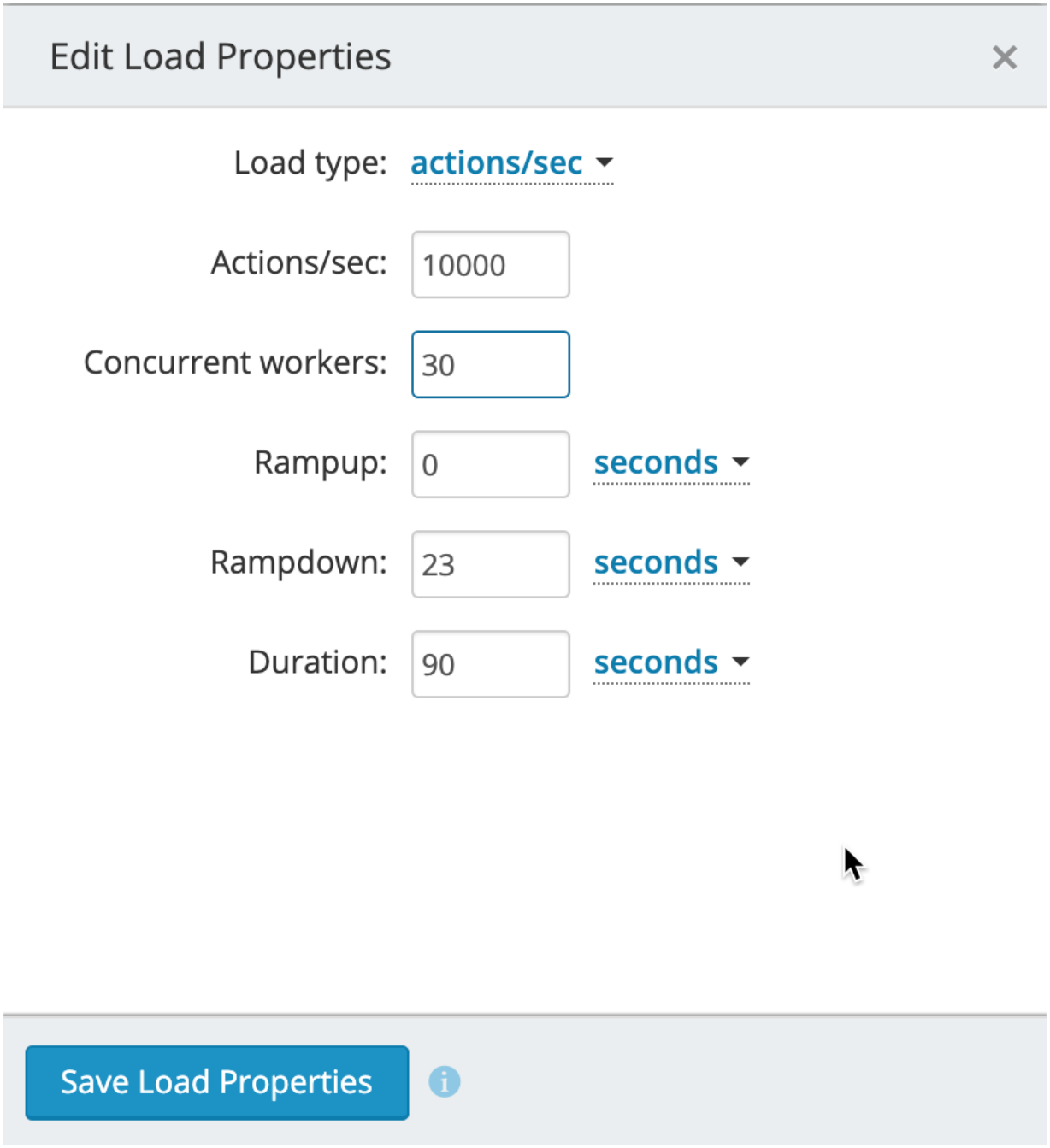


Edit Parameters

Once you have selected one or more workloads, you can perform bulk parameter edits across all selected workloads at once.

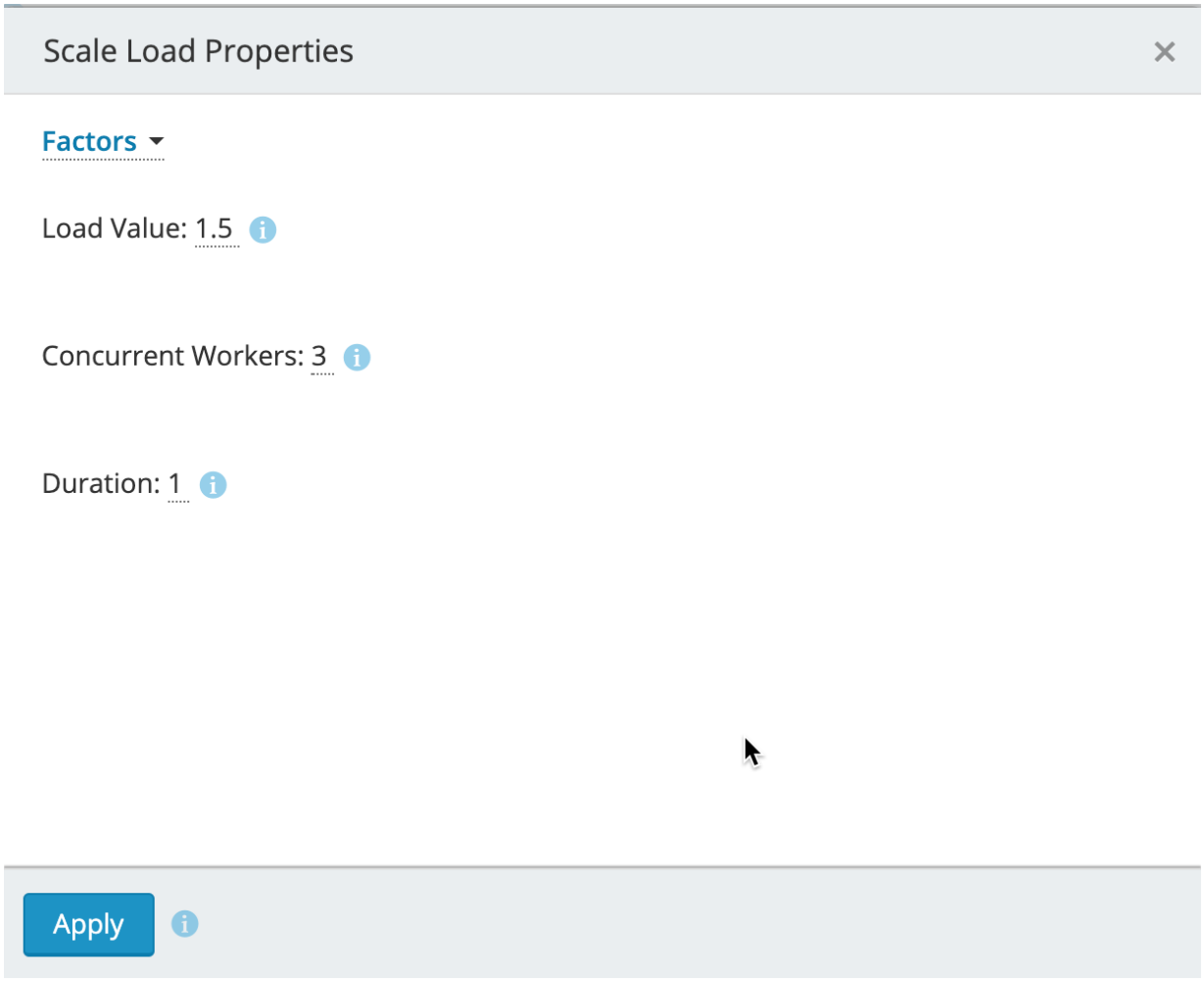
Load Properties

You can use *EditLoad Properties* to set a fixed value across all selected workloads.



Scale Load

You can use *Scale Load Properties* to scale the existing load properties of each workload by a factor or percentage. The parameters you can scale are *Load Value*, *Concurrent Workers*, and *Duration*.



Replay Workloads

Replay Workloads are a special category of workloads that replays the exact sequence of commands recorded by a supported source file. Replay workloads are not workload models, and do not generate IOs based on percentage distributions like the Storage Protocol Workloads.

Replay Workloads can only be created from the *Workload Data Importer* workflow.

Once a supported data source file is imported into *Workload Data Importer*, the output will be a Replay Workload.

Design Limits and Behaviors

The amount of time it takes to analyze the imported per-command trace data increases as the imported file's table size increases. To give some idea of how long it takes to analyze a CSV file of certain size, here are some internal benchmarks that were observed:

- 4MB CSV file: ~1 minute
- 40MB CSV file: ~2 minutes
- 400MB CSV file: ~25 minutes

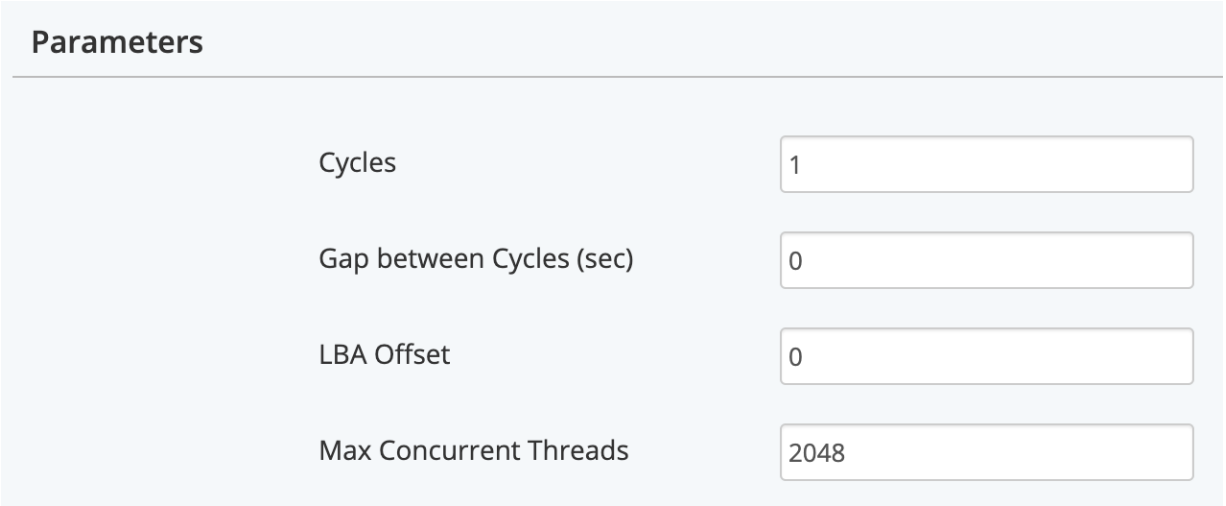
The exact analysis time you experience may be different from what was described above, depending on the exact number of rows in the CSV file, the number of columns, and what else WorkloadWisdom is doing at the time.

A maximum file size of 1GB is recommended, because there is an upper limit of 1GB per Workload Generator Port that stores the entire command sequence that the Port will be responsible for generating. In the event you import a CSV file that is larger than 1GB, there is still a possibility that it will be accepted as the imported data is transformed to a file that gets loaded to each Workload Generator Port, and usually it is a smaller file size than the original imported data file.

There is also a product wide maximum import size of any file into WorkloadWisdom. See Design Limits and Support [12].

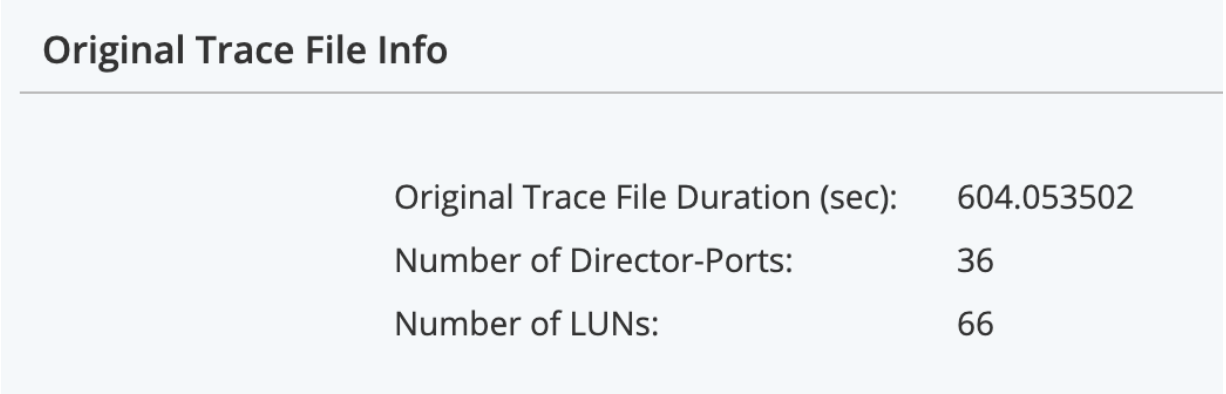
The Duration of the Replay Workload Test Run is automatically calculated based on the information provided in the imported workload data file, and the number of cycles, rounded up to the nearest second. Therefore, you do not need to specify the Duration of the test run like other workloads.

Replay Workload Options



- *Cycles*. The number of times the workload will repeat itself.
- *Gap between Cycles (sec)*. The number of seconds to wait before starting a new cycle.
- *LBA Offset*. The LBA offset, in number of blocks, added to the recorded LBA each time a new cycle begins. Leaving it at 0 can cause the SUT to cache the data as the same data can appear on the same block every cycle. Setting it to a non-zero value will cause the LBA of a command to shift by that non-zero value each cycle. For example, if a recorded command's LBA is 2,048, and the LBA Offset is 2, then during the first cycle, the command will carry an LBA of 2,048 + 2 * block size, then during the second cycle, the command will carry an LBA of 2,048 + 4 * block size, then during the third cycle, the command will carry an LBA of 2,048 + 6 * blocksize.
- *Max Concurrent Threads*. The maximum number of threads that can run concurrently. Each Thread works on a specific LUN. So if you have 10 LUNs, then you should have at least 10 concurrent threads. Generally, it is ok to set it at 2,048, which is the maximum value.

Original Trace File Info



The *Original Trace File Info* section is a read-only section that gives you high level summary of the imported trace file, which will be helpful when selecting a Test Bed ITLs for the Replay Workload Test Run.

Workload Test Run Results Dashboard Page

When you start a workload test or click a test result from the *Completed Workload Tests* page, the *Results Dashboard* page for the selected workload test run displays. The *Results Dashboard* page displays a test run's test results. These results provide Key Performance Indicators (KPI) charts (for example, throughput, IOPS, latency) and a comprehensive set of statistics (thousands of statistics per second per Test Port) in real-time as well as post-test, for all workload test runs.

Within each chart, a wrench tool is provided to perform additional drill-drowns to get more detailed results such as IOPS per command, or switch to a different view such as Log view as well as switching between absolute time (aka wall clock time) or relative time (i.e., relative to test start, which is recorded as time 0). Some options are not available for some protocols.

NOTE
To ensure all statistics are processed timely and without loss, you need ensure that your usage of WorkloadWisdom falls within the specifications outlined in the Design Limits and Support section.

If you have a WorkloadWisdom Virtual deployment, the minimum compute resource allocations are guaranteed. As the number of concurrent tests or concurrent users increase, the time to load charts and statistics increases.

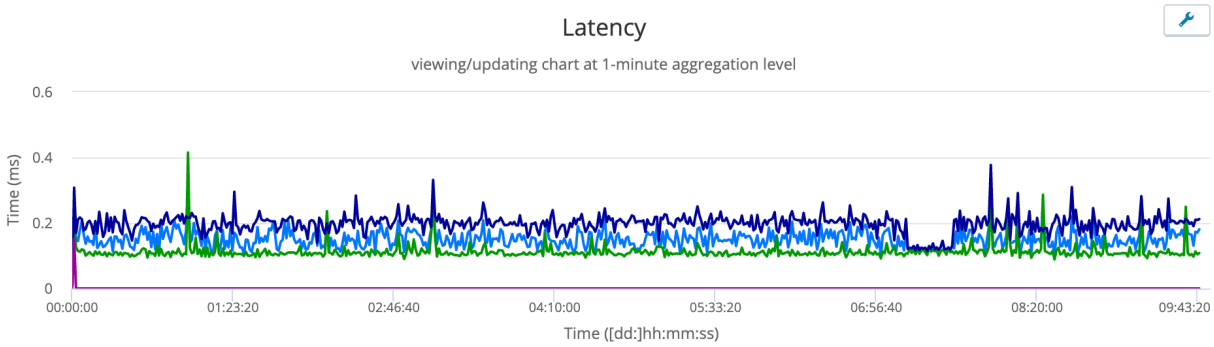
Statistics Engine Overview

For each workload test, whether it is running as a standalone test or as a part of a suite, statistics are being retrieved, stored, calculated, and presented in real-time.

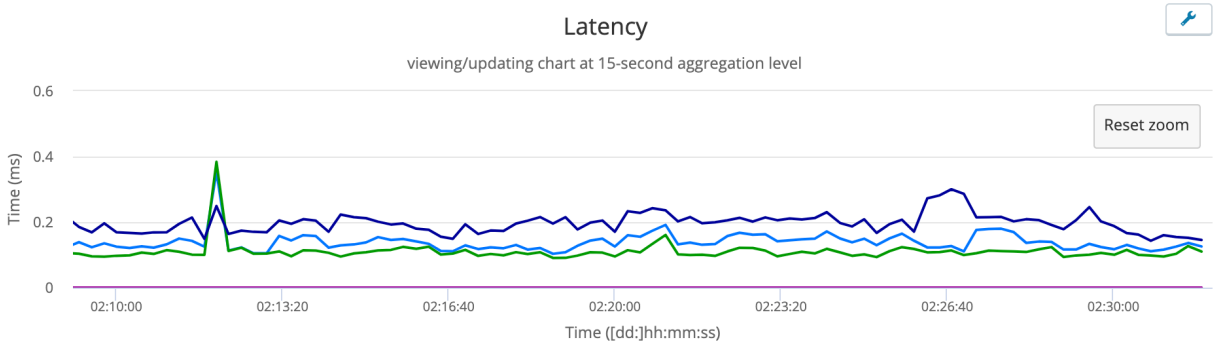
This section gives an overview of the WorkloadWisdom statistics design that is running in the background to generate the charts and tables that you see in the *Results Dashboard*. You might notice a few seconds delay, as some statistics need to be calculated after they have been collected across all test ports.

Two tiers of statistics are processed: KPI statistics and non-KPI statistics. KPI statistics produce the Throughput, IOPS, Latency, and Command Mix Charts for the storage protocol layer (for example, SCSI for FC, and RPC for NFS). Non-KPI statistics are for the most part all statistics below the storage protocol layer (for example, TCP).

All statistics are processed and stored every 1 second. For *Time* charts, data points are plotted every 1 second by default in real-time. As more data points are plotted on the fixed-length chart in a long duration test, the data points are required to be aggregated to coarser intervals (called the aggregation level) to enable the complete data points set to fit on the chart. The aggregation levels are 15 seconds, 1 minute, and 5 minutes. WorkloadWisdom automatically determines the aggregation level required to show the highest fidelity of data possible.

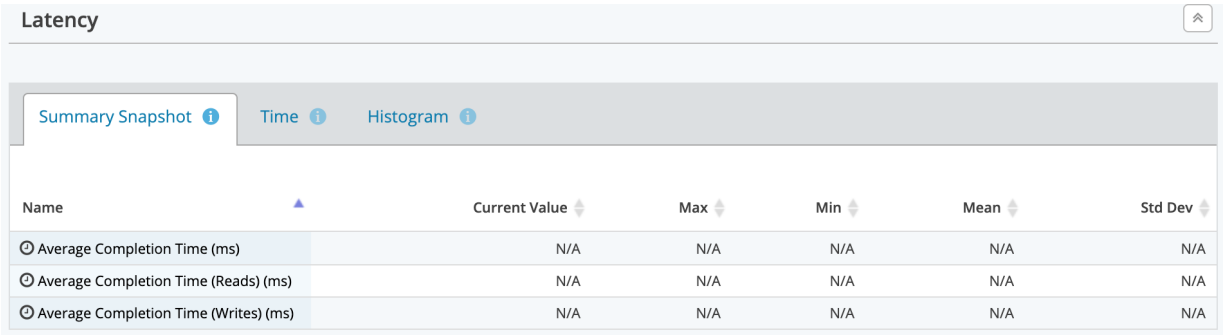


If necessary, you can use the zoom function to see the data points at lower aggregation levels.



The data points are also provided in tabular form in snapshots of time. After the data points are plotted on the Time charts, they are calculated and displayed in the Summary Snapshot tables. The Summary Snapshots tables follow the currently displaying time range of the Time chart, which is from the start of test to the latest data point by default. If you zoom in to select a specific part of the Time chart, the Summary Snapshots table will recalculate.

For long duration tests (for example, multi-day tests), it can take several minutes for all the statistics to be calculated and displayed on the Results Dashboard, as the number of statistics to be processed can reach over a billion, and the size of the data points set can be over several GBs in size. During this time, you might see “N/A” for some statistics along with a clock icon. This indicates that calculation is still in progress.



In some extreme cases, calculation might time out or error out. This is indicated by orange colored text for the affected stats instead of the default black. If you encounter this, refresh the Results Dashboard one or more times.

WorkloadWisdom processes a massive amount of data points in real-time very efficiently. In some extreme cases, for example, multi-port multi-day tests, the number of data points to be processed requires additional time to calculate, summarize, and display.

Results Dashboard

The *Results Dashboard* is structured as follows:

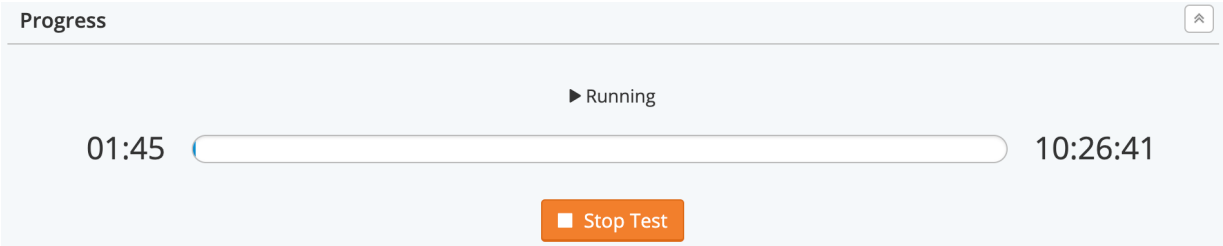
- Test Progress
- Command Mix
- KPI Charts
- Errors Summary
- All Charts
- Test files

Test Progress, All Charts and Test files are available for all workload test runs.

In addition, Command Mix, KPI Charts and Errors Summary are available for all protocol workloads and Application Workloads, but their content varies depending on the protocol and the specific Application Workload.

Test Progress

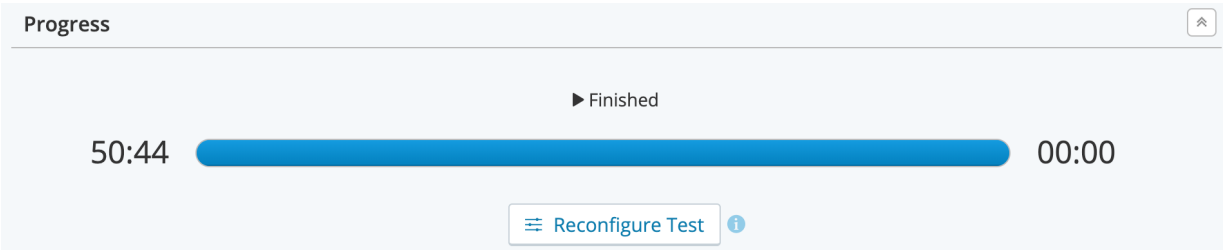
Running Tests



Tests that are actively running are in the Running state.

The time bar indicates the current Workload run Elapsed Time and estimated time remaining.

Completed Tests



Completed Tests show the Finished state.

The time bar indicates the actual Workload run Elapsed Time. For example, if a Workload was configured with a Duration of 3 minutes, but completed within 2 minutes and 15 seconds, the Elapsed Time is 2 minutes and 15 seconds.

Use Reconfigure Test to return to the Workload with the exact configurations used when the Test Result was created.

Other States

Other States might be observed as well during a normal workload test run. Those other States are usually transient and short-lived, unless otherwise specified. Those other States indicate what WorkloadWisdom is processing before and after the Running State or Finished State.

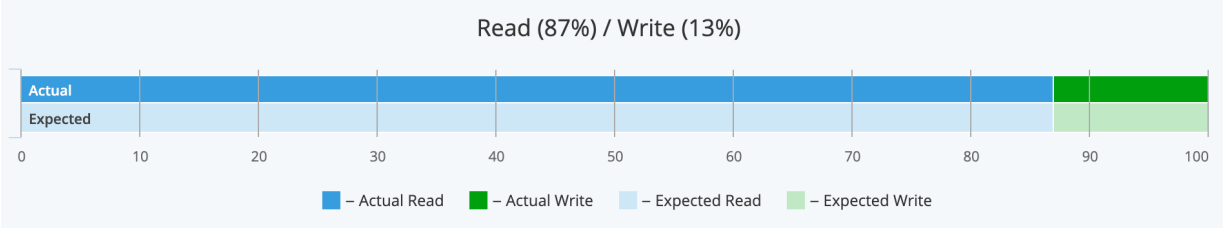
These following States might be observed:

- *Starting*. Content of the workload test is being generated and uploaded to the Workload Generators.
- *Stopping*. Workload test has been requested to stop running. The test ports are being prepared to return to idle.
- *Downloading*. Workload test has finished running. WorkloadWisdom is downloading the test files collected at the end of each workload test run
- *Waiting*: Workload test is queued to run next, when the test ports required are no longer in use.

Command Mix

This section displays the distribution of protocol commands used in the Workload.

Fibre Channel and iSCSI



For Fibre Channel and iSCSI Workloads, a read/write distribution is provided. The percentage of Read requests is displayed.

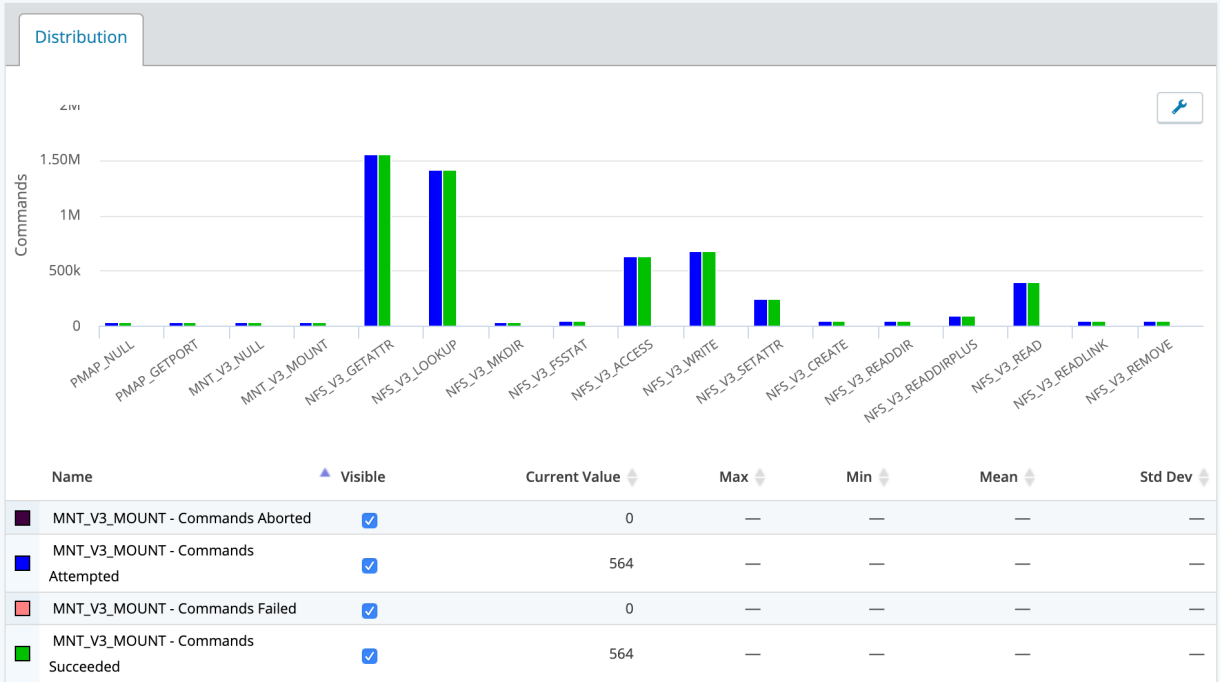
SMB and NFS

For SMB and NFS Workloads, where many metadata commands are required in addition to Read and Write requests, a more detail Command Mix is provided.

Two tabs are available for the Command Mix chart: Distribution and Actual vs Expected.

Distribution

This tab displays the breakdown of actual command mix count observed.



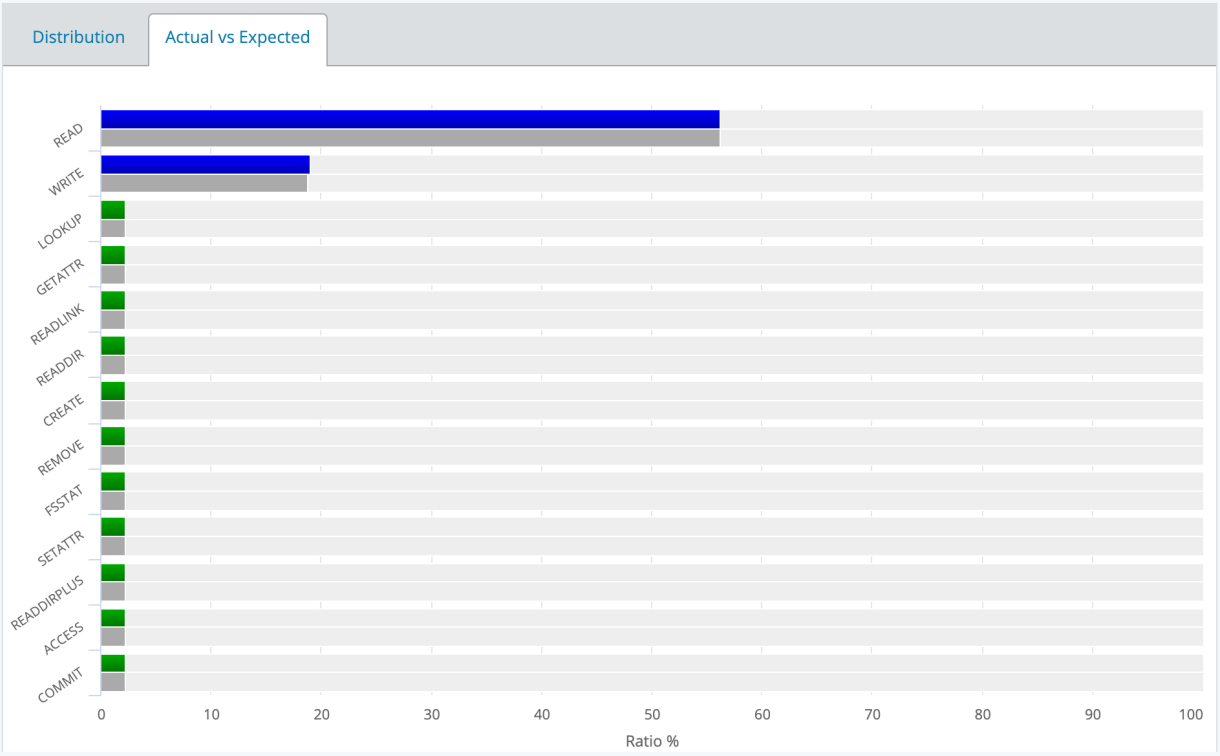
Name	Visible	Current Value	Max	Min	Mean	Std Dev
MNT_V3_MOUNT - Commands Aborted	☑	0	—	—	—	—
MNT_V3_MOUNT - Commands Attempted	☑	564	—	—	—	—
MNT_V3_MOUNT - Commands Failed	☑	0	—	—	—	—
MNT_V3_MOUNT - Commands Succeeded	☑	564	—	—	—	—

The following statistics are collected:

- *Commands Attempted (blue bar)*. Number of times the command was attempted
- *Commands Succeeded (green bar)*. Number of times the command received a positive response. Generally, this is the case where the request was handled successfully.
- *Commands Failed (red bar)*. Number of times the command received a negative response. Generally, this is the case where an error was received for the request.
- *Commands Aborted (brown bar)*. Number of times the command was aborted. Generally, this is the case where no response was received for the request and the request timed out, but it can also be caused by a request that was terminated by the client.

Actual vs Expected

This tab displays the breakdown of actual command mix percentage observed versus the command mix percentage expected.



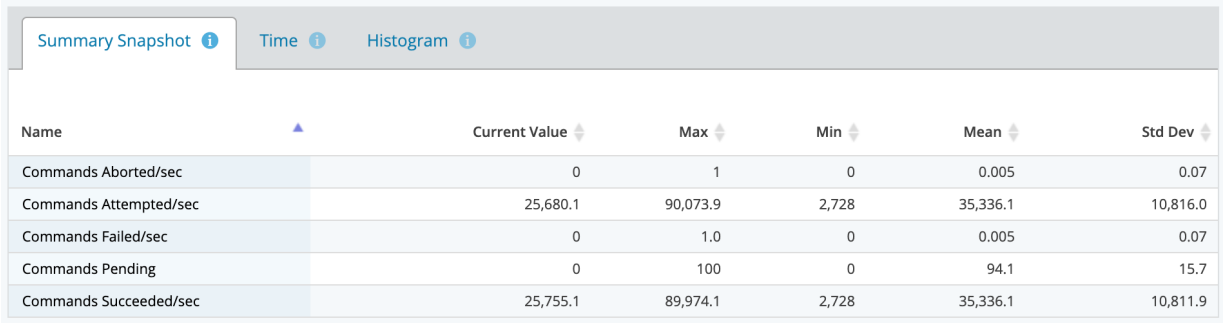
The gray bar in the background for each command displays the expected percentage of the command, based on your Workload configuration. The green bar in the foreground for each command displays the actual percentage of the command from the Workload run.

KPI Charts

The typical KPI Charts for all workloads are IOPS, throughput and latency (response time). Three views are available for each KPI Chart: Summary (tabular), Time and Histogram. By default, Summary view is displayed. Click the tabs to access different views.

Summary View

The Summary View provides summary statistics in a tabular format. The Summary View does not provide Time charts or Histogram charts.

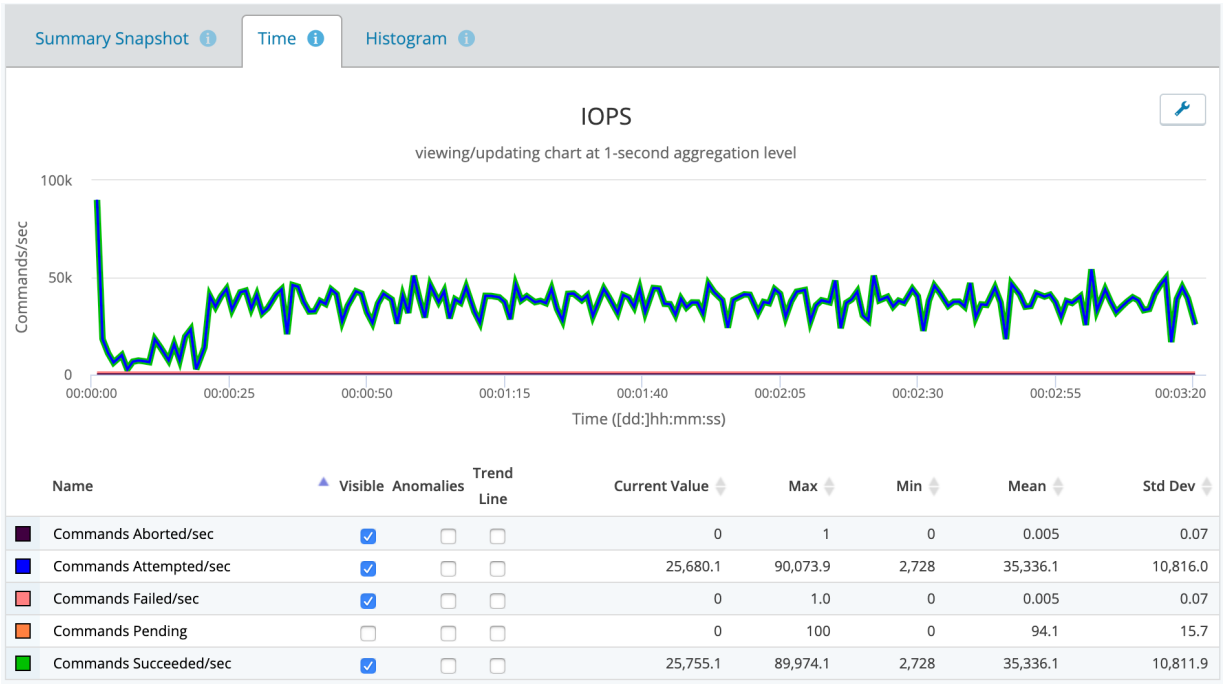


Name	Current Value	Max	Min	Mean	Std Dev
Commands Aborted/sec	0	1	0	0.005	0.07
Commands Attempted/sec	25,680.1	90,073.9	2,728	35,336.1	10,816.0
Commands Failed/sec	0	1.0	0	0.005	0.07
Commands Pending	0	100	0	94.1	15.7
Commands Succeeded/sec	25,755.1	89,974.1	2,728	35,336.1	10,811.9

Time View

The Time View provides charts that show the value recorded (y-axis) for each time interval (x-axis) recorded, from the beginning of the test.

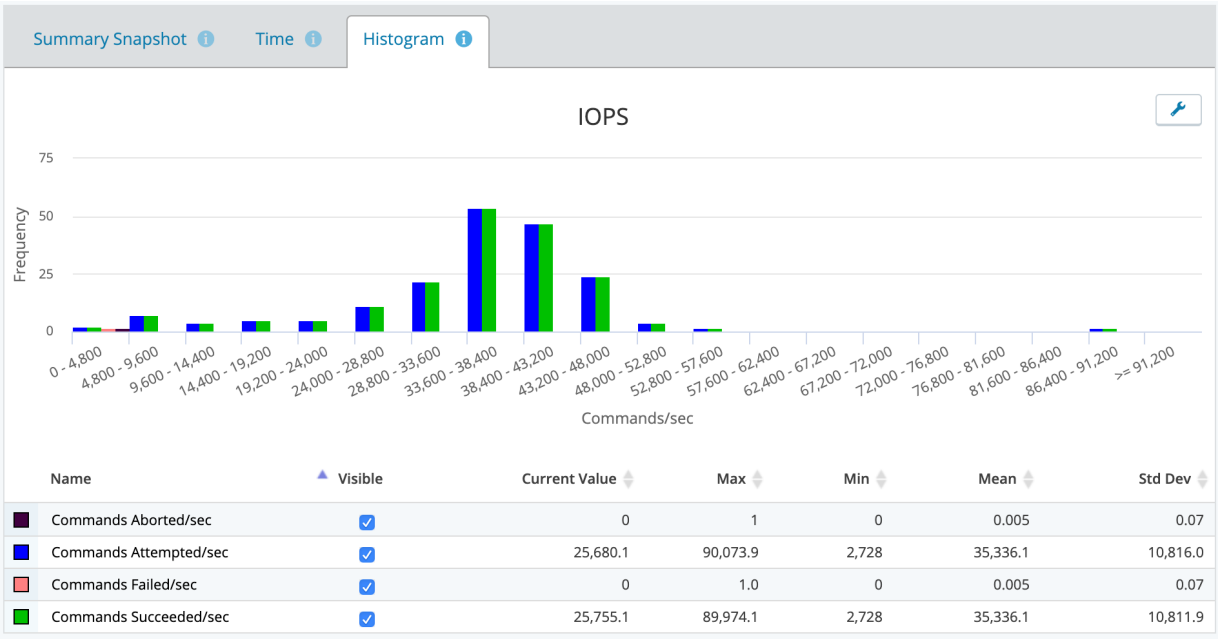
By default, the granularity of the presentation is summarized every second. As the length of the test run increases, the granularity of the presentation increases to every 15 seconds, and every 60 seconds for very long tests. No data is lost, as data for every second is recorded in the database. The granularity of the display changes automatically, based on the test length. For long tests, you can zoom in to any particular timespan on the chart to see data presented at a finer granularity.



Name	Visible	Anomalies	Trend Line	Current Value	Max	Min	Mean	Std Dev
Commands Aborted/sec	☑	☐	☐	0	1	0	0.005	0.07
Commands Attempted/sec	☑	☐	☐	25,680.1	90,073.9	2,728	35,336.1	10,816.0
Commands Failed/sec	☑	☐	☐	0	1.0	0	0.005	0.07
Commands Pending	☐	☐	☐	0	100	0	94.1	15.7
Commands Succeeded/sec	☑	☐	☐	25,755.1	89,974.1	2,728	35,336.1	10,811.9

Histogram View

Histogram view plots the distribution of observed values, to show the concentration of the DUT's performance over the duration of the test. *Histogram* view is useful for latency measurement.



Name	Visible	Current Value	Max	Min	Mean	Std Dev
Commands Aborted/sec	☑	0	1	0	0.005	0.07
Commands Attempted/sec	☑	25,680.1	90,073.9	2,728	35,336.1	10,816.0
Commands Failed/sec	☑	0	1.0	0	0.005	0.07
Commands Succeeded/sec	☑	25,755.1	89,974.1	2,728	35,336.1	10,811.9

Errors

The Errors section provides a high level indication of presence of errors, and once you expand on the Errors section, you will see the Alerts Summary which gives a total count of each error type, along with an Error Inspector for each error type that provides detailed reporting of the reported error in this Workload run instance.

Alerts Summary

The Alerts Summary provides a breakdown of the different errors observed, and the total count of each error. The Alerts Summary table is organized in an OSI-like stack, where the upper layer protocol (e.g. SMB) is at the top of the table, followed by TCP, and so on. At a quick glance, you can quickly get a sense of where the errors are.

For example, if you see no SMB errors and no MAC/VLAN errors, but see a lot of TCP Connection errors, then that is a good indication that there is a TCP transport layer issue or IP networking layer issue, preventing the SMB clients from starting from an SMB protocol perspective. Or vice versa, if you see a high number of SMB errors but few TCP

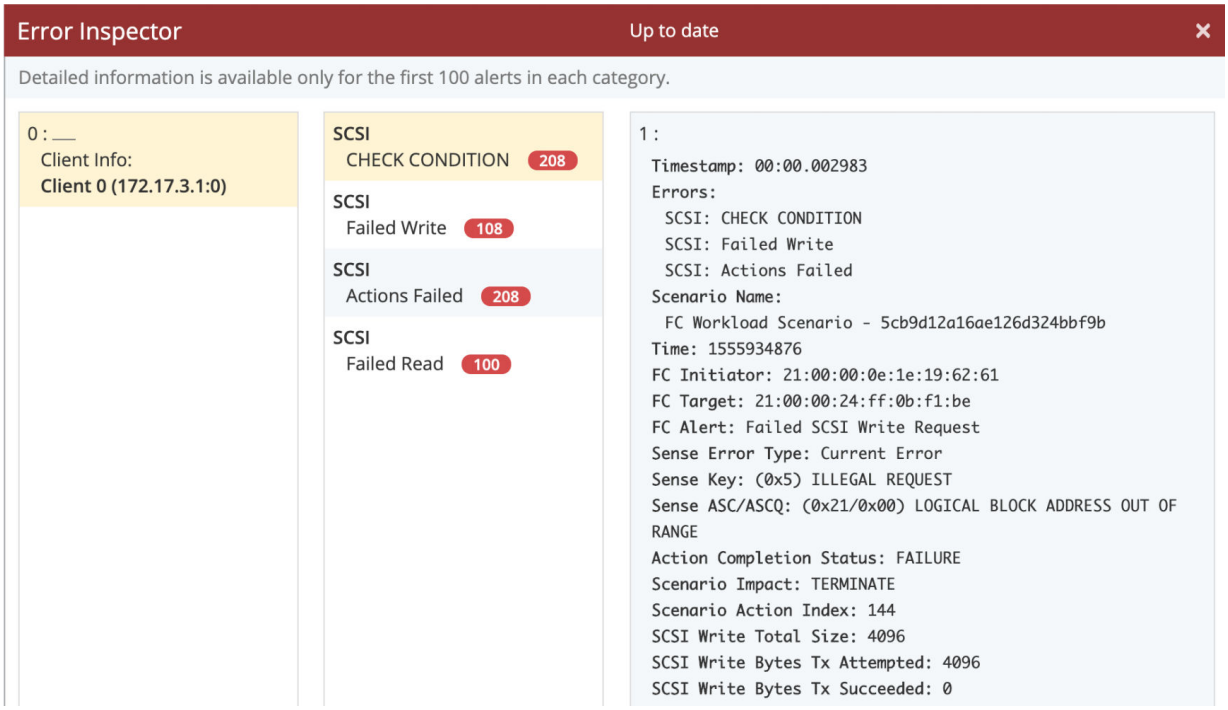
errors, then that is a good indication that the TCP/IP layers are generally ok, but the SUT is not performing so well at the SMB layer.

Alerts Summary	Count	
SMB2		
Actions Failed	3333	Error Inspector
TCP Connections		
Connections Failed to Open	27	Error Inspector
SYN Handshakes Rejected	1	Error Inspector
SYN Handshakes Timeout	26	Error Inspector

Error Inspector

The Error Inspector is a deep error reporting feature that tells you when an error occurred, who was involved, where in the System Under Test the error occurred, what error codes are reported, and more. Here is an example for FC-SCSI.

FC-SCSI example:



We see that the workload run found the following: Essentially, this is a hierarchy of error types and counts observed, and you can think of it in the following structure. Future releases of WorkloadWisdom will improve the layout of this view.

- 208 FC-SCSI Workload Actions (for the most part, Actions are Commands) encountered errors
- 100 were Read Action errors
- 108 were Write Action errors
- The CHECK CONDITION error was observed 208 times

Essentially, this is a hierarchy of error types and counts observed, and you can think of it in the following structure. Future releases of WorkloadWisdom will improve the layout of this view.

In the right column, a detailed reporting of the error code is shown.

- Timestamp: The relative test timestamp when this occurrence of the error was observed (test start time has a timestamp of 0).
- Error type: The types of Errors listed in the middle column that this occurrence of the error contributed to.
- Scenario name: This is mainly for Support use and is not generally visible / meaningful to WorkloadWisdom users. However, if you use TDE as well, then you can see this Scenario name when you export the Workload Test Run and opens it from TDE.
- Time: This is the Unix Epoch time that corresponds to the Timestamp above.
- FC Initiator / Target: This is the Initiator WWPN and Target WWPN found in the SCSI message that contains the reported CHECK CONDITION.
- FC Alert: This is the command that triggered the error from a SCSI protocol perspective.
- Sense information: These are the SCSI error codes that are found in the SCSI message that contains the reported error.
- Action Completion Status: This is mainly for Support use, and is not generally visible / meaningful to WorkloadWisdom users.
- Scenario Impact: By default this is always TERMINATE. However, if the workload supports the **Ignore errors gracefully** option and you have selected the checkbox, then this will report CONTINUE, which means that even though an error was encountered, you wanted the workload to continue to run.
- Scenario Action Index: This is mainly for Support use, and is not generally visible / meaningful to WorkloadWisdom users. However, if you use TDE as well, then you can use this information to find out the exact Action (Command) in the workload Scenario by this index number, as every Action in a Scenario in TDE has a unique index number.
- SCSI Write Total Size / Bytes information: This tells you the attempted total size of the Write command, and exactly how far into the Write (i.e. offset) when the error was reported.

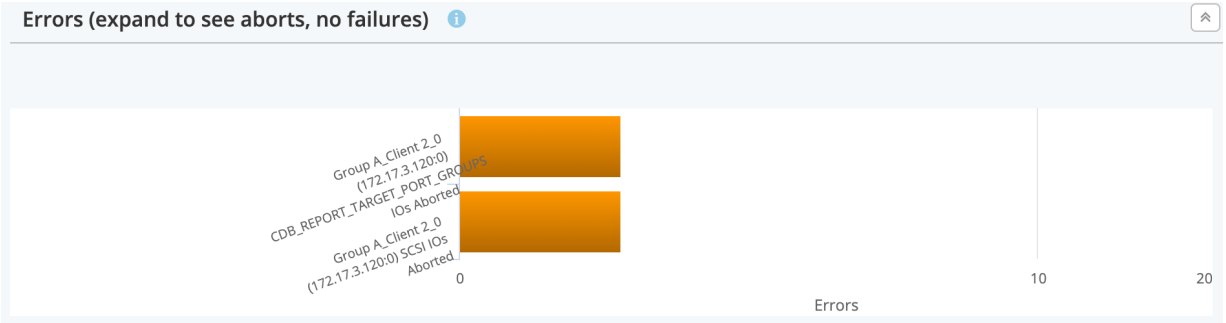
Different protocols offer different levels of details based on what is available from a protocol's error reporting that is observable in a packet on the wire. For example, Error Inspector for iSCSI includes information about sequence numbers found in the iSCSI conversation where the error was reported, and for File protocols like SMB, Error Inspector provides information about the Share and Filename where the error was reported.

For performance reasons, the number of unique error occurrence records stored is the first 100 occurrences of each error type (i.e. error category in the middle column).

Legacy Errors Summary (Workload Test Runs before 6.4)

The Errors Summary section provides a high level indication of errors, as well as accounting of per-command type errors encountered in this Workload run instance.

Two types of errors are reported: fails and aborts. Once expanded, the number of fails and aborts are provided for each statistic.



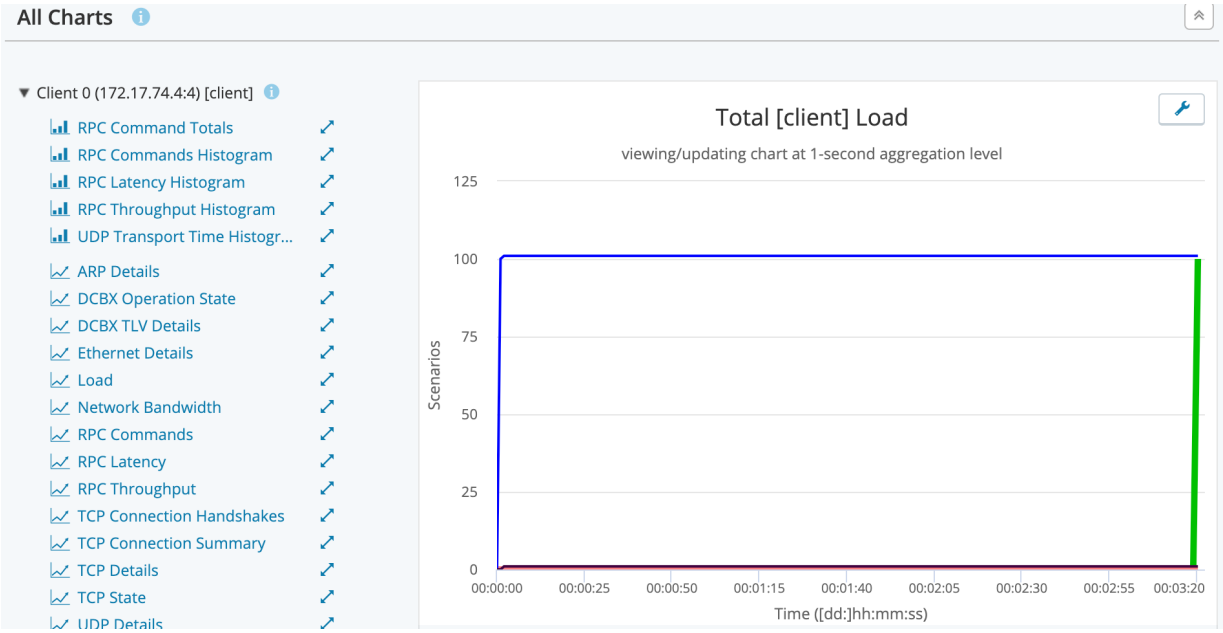
There are cases where 1 error encounter can be effectively registered multiple times.

For example, during the workload run instance, 1 CDB Test Unit Ready request encountered an error, therefore the CDB Test Unit Ready command registered 1 error. At the same time, the CDB Test Unit Ready command is considered a SCSI IO command. Therefore, the SCSI IO error is also registered.

To access the detail errors encountered, navigate the charts in the All Charts section.

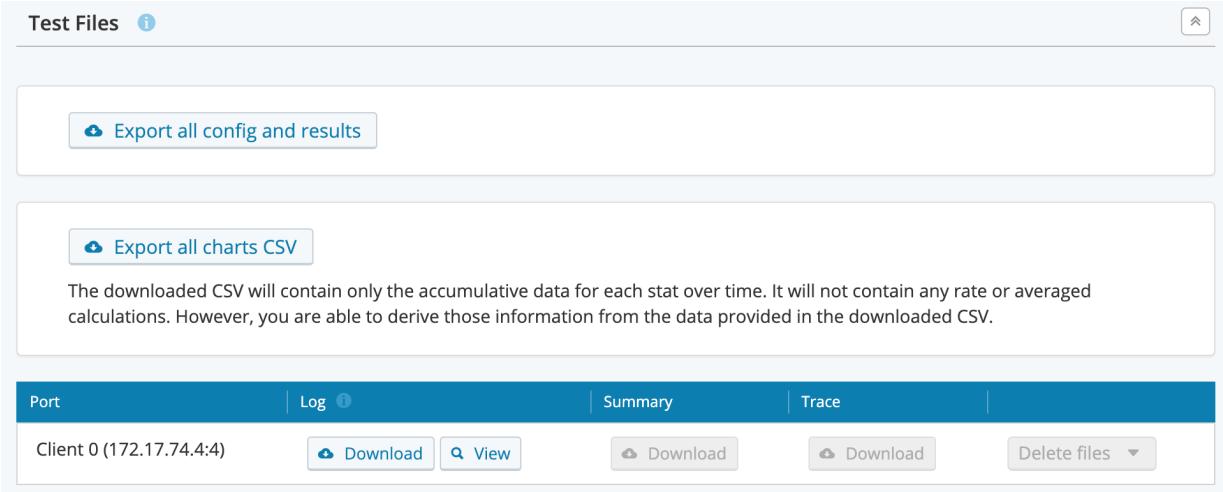
All Charts

This section provides additional charts that can be helpful in further understanding the test results. These additional charts include network layer statistics (for example, ARP), transport layer statistics (for example, TCP), and protocol layer statistics (for example, NFS) on a per-port basis and in aggregation. A set of Scatter Plots is also available.



Test Files

This section provides downloadable content at the end of a test run for a variety of uses, including offline analysis using 3rd party CSV editors, or export to TDE, or more commonly, share files with Support for assistance.



Export all config and results: at a high level, this exports a single .zip file that includes three types of content: Workload Test, Test Bed, and available post-test files (logs, summary files if enabled, alert files, and trace files if enabled) that are collected on a per-port basis. Each of the three types of content is in a .zip file itself.

Note: if you are to import the exported single .zip file into another WorkloadWisdom (for example, sharing with colleagues, customers, or vendors), you need to unzip the .zip file, because upon import, you need to import the Workload Test file using the **import Workload Test** function, and import the Test Bed file using the **import Test Bed** function. The post-test files cannot be imported because they are designed for use outside of WorkloadWisdom.

When you work with Support, it is often and common that you will be requested to use this function to provide the information necessary for Support to provide better assistance.

Export all charts CSV: this exports all cumulative time-series data into a set of CSV files on a per-port basis and (mostly) per-protocol basis.

Per port post-test files: this section provides the ability to collect a specific type of post-test file on a per-port basis. When multiple Workload Generator Ports are used in a Workload Test Run, each port will collect the same set of files individually (if enabled).

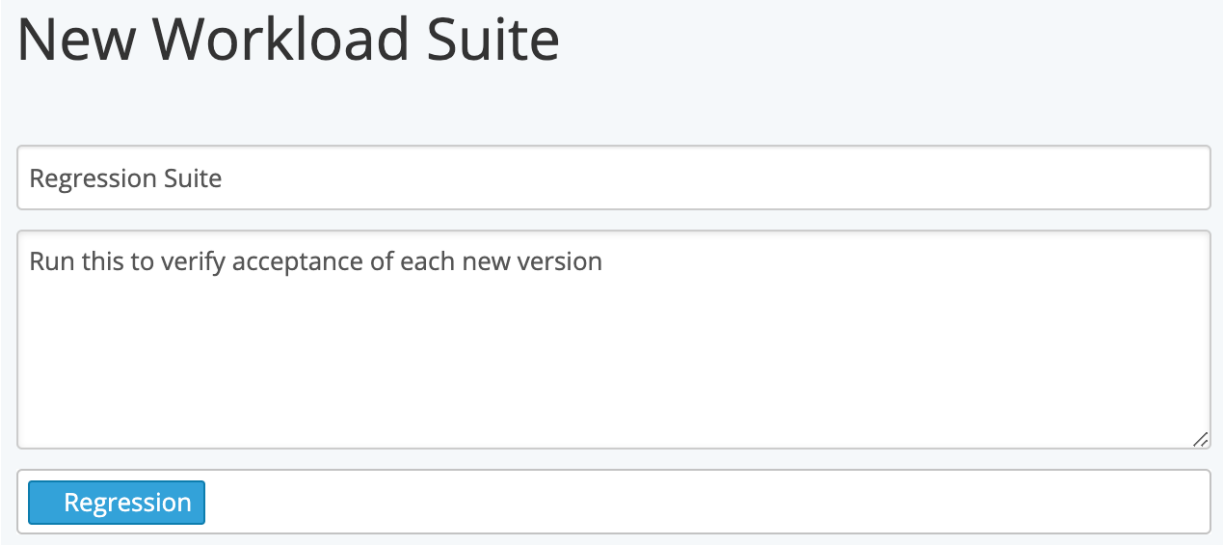
- **Log**: This is a text-based log that the Workload Generator collects, and it provides additional details that are not normally needed by users, but can give insight into specific issues and data that were observed during a test run.
- **Summary**: If you enabled the option **retrieve summary files** when you started a Workload Test Run, this button will be available. Downloading summary files can take a long time, as some summary files can be many GBs large if it was a long test run.
- **Alerts**: This is a proprietary file format that provides the information that is used by the Errors section of the Results Dashboard. Generally there is no need for you to view the Errors details using these post-test Alerts files, as you get the same set of information in the Results Dashboard within WorkloadWisdom. However, when working with Support, this file can be extremely important.
- **Trace**: If you enabled the **Tracing** option when you started a Workload Test Run, the resulting trace file (usually .PCAP) can be downloaded.

Workload Suites

Use *Workload Suites* to define a list of workload tests to run automatically when you run a *Workload Suite*.

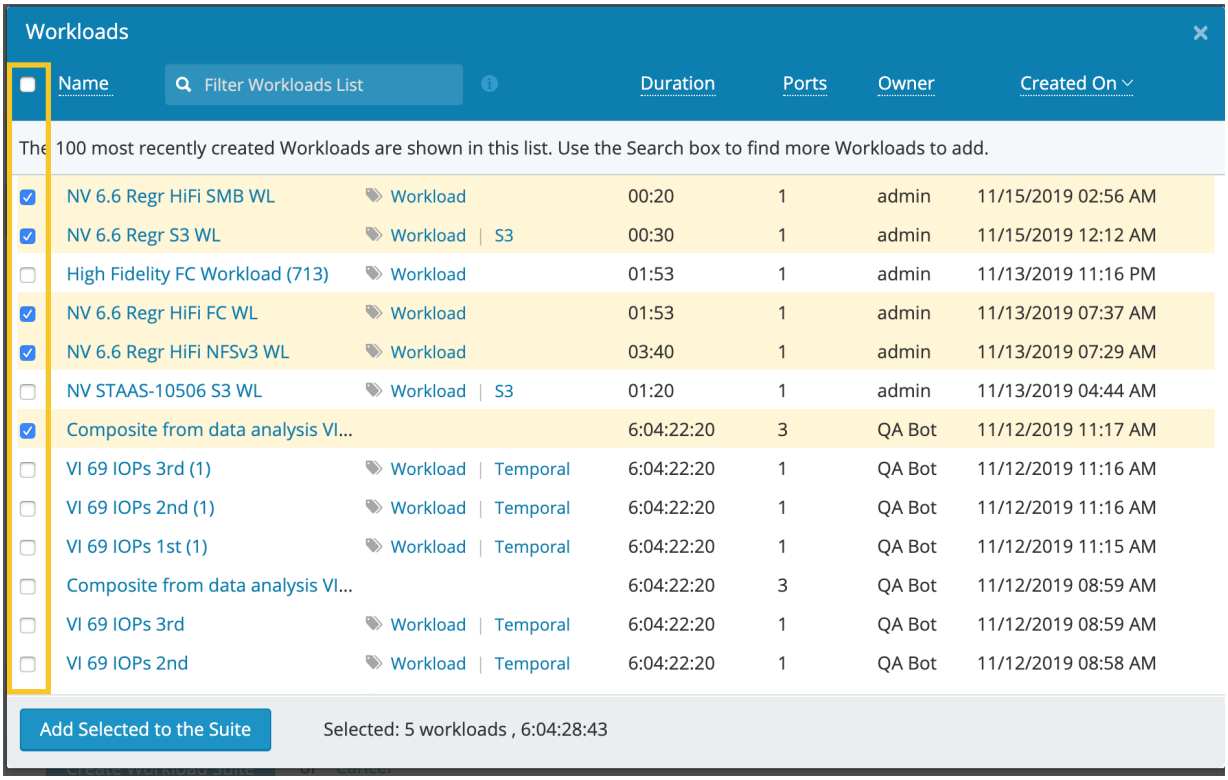
New Workload Suite

Start by creating a new workload suite.

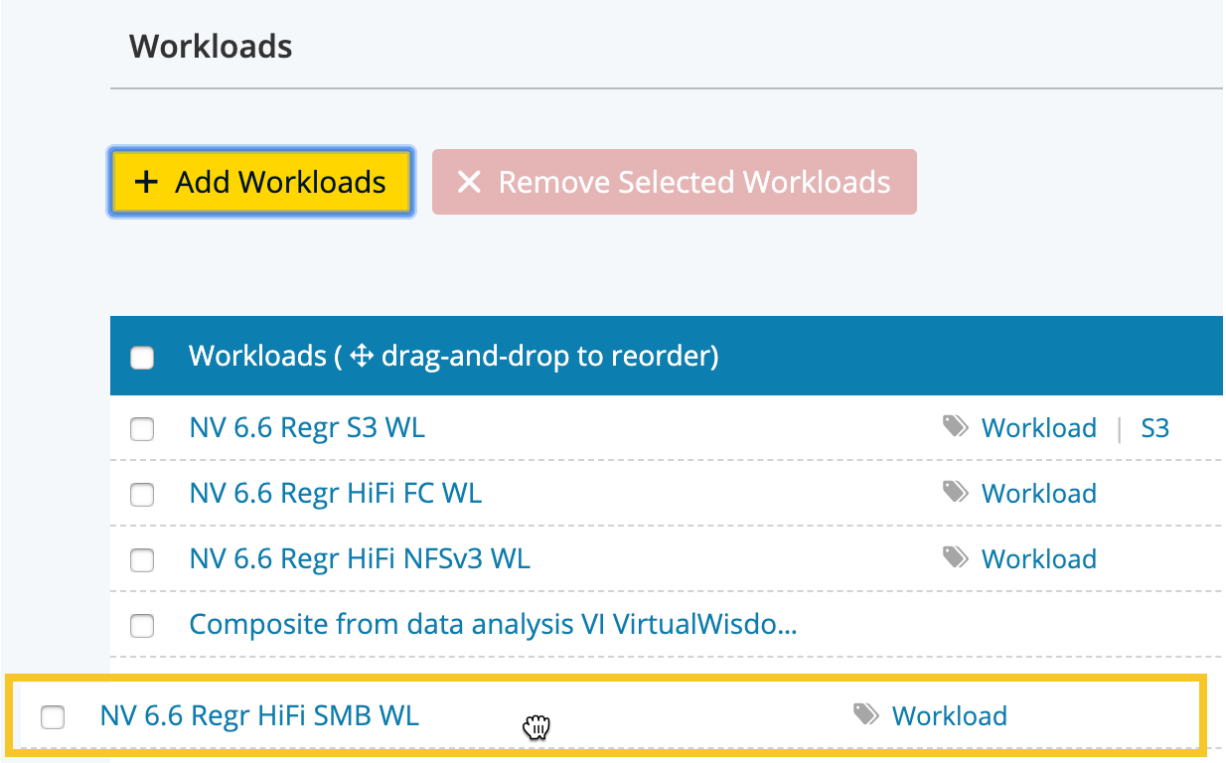


Select Workloads

Add one or more workloads to the workload suite.



After you select the list of Workloads, you can drag-and-drop to reorder the workloads order when the workload suite runs.

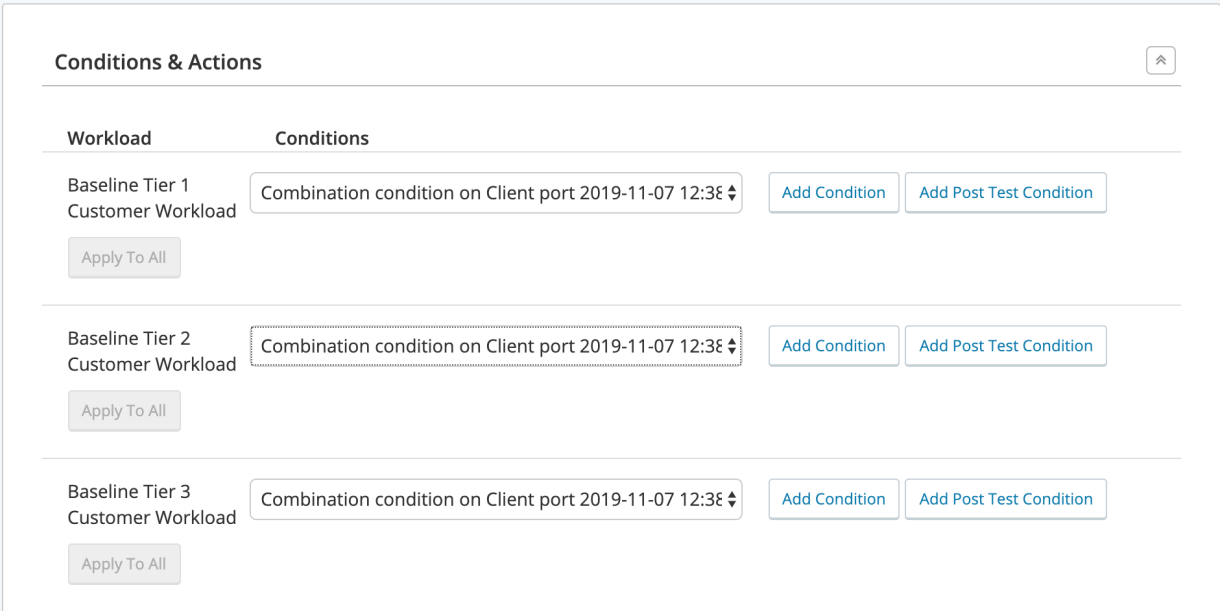


Finally, optionally define a non-zero wait time between the end of the current Workload and the beginning of the next Workload. Usually you want to do this if the SUT needs additional time to process the closing of connections from the current Workload and the influx of connections from the next Workload.

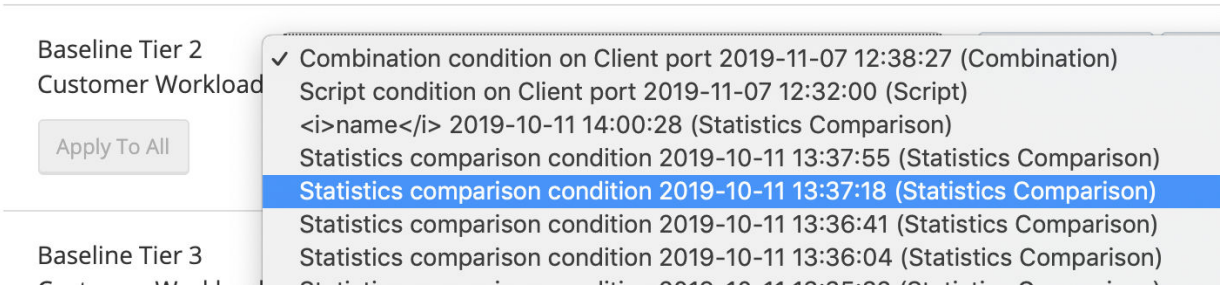
Click *Create Workload Suite* to complete the process.

Conditions & Actions

Optionally select one or more Conditions to apply when running the workload suite. You can select a different Condition for each Workload, or apply the same Condition to all Workloads. In addition, you can apply a Condition as a run-time condition or a post-test condition. You can use both default Conditions and user created Conditions.



For each Workload, click the drop-down menu to select a Condition to apply. Then, apply the Condition as a run-time Condition (Add Condition), or post-test Condition (Add Post Test Condition).



Once applied, you can also click Apply to All to apply the Condition to all workloads in this workload suite.

Running Workload Suites

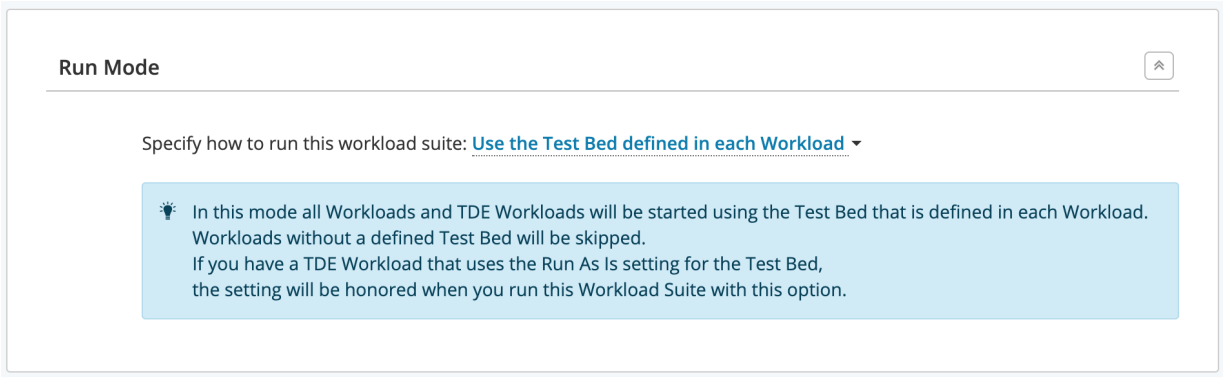
Find the workload suite you want to run, and then click Start to proceed.

Review the list of workloads in the workload suite to make sure the list is correct.

When running the workload suite, you can set it up to use one or more test beds under the Run Mode settings.

Three Run Modes are available:

- *Use the test bed defined in each Workload*. Each Workload runs on the test bed that is specified in the workload setting. For TDE workloads with the *Run As-Is* option configured, the setting is honored. Workloads that do not have a test bed specified are skipped.
- *Run sequentially on One Selected test bed*. Workloads execute sequentially on the specific selected test bed. Use this run mode if you want to control the execution order of the workloads and the specific test bed.
- *Run Simultaneously on Several test beds*. Run as many workloads as possible from the pool of applicable test beds. Use this run mode to minimize the total workload suite execution time. This run mode does not require a specific test bed for a specific workload. To identify the test beds used by this workload suite in simultaneous run mode, assign one or more unique tags to the test beds. In the following example, all test beds with the highlighted tags are applicable.



Iteration Suites

Use the Iteration Suite to configure and run a set of automated tests that iterates over one or more parameters from a selected Workload or TDE Workload. A popular use case is to sweep the system under test with a range of values for one or more parameters to assess the SUT's response to a range of IO characteristics, to determine the Workload characteristics the SUT is optimized for and not optimized for.

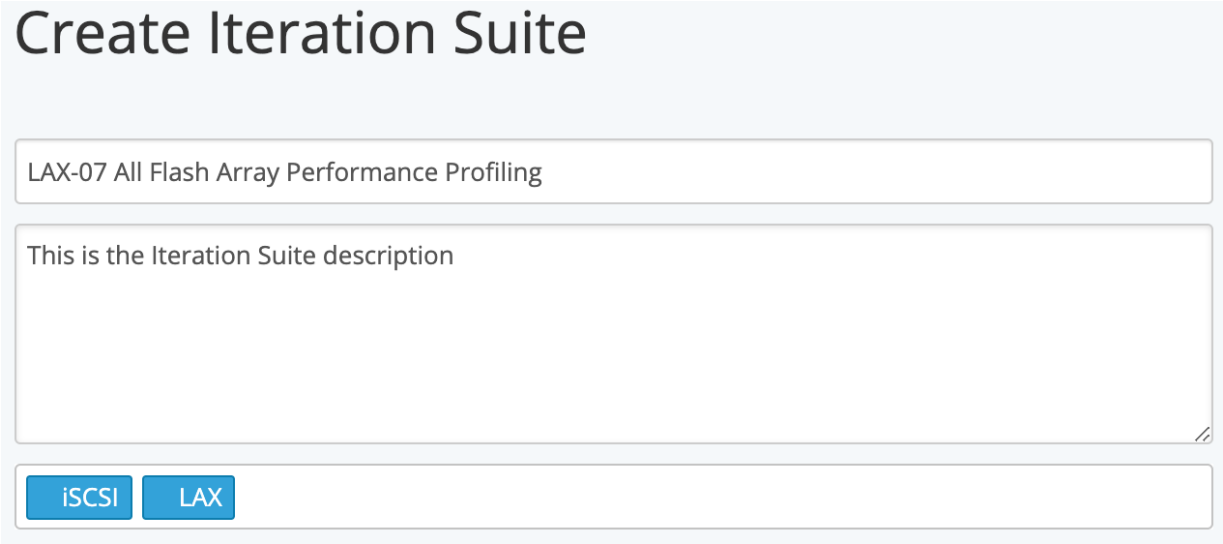
The workflow of the Iteration Suite is as follows:

1. Create a new Iteration Suite
2. Select a Workload
3. Define Iteration Parameters
4. Select Statistics for Iteration Results
5. Define Runtime Parameters

6. Run Iteration Suite and browse results with Iteration Explorer

Create a New Iteration Suite

Access the page to create new Iteration Suites as follows:

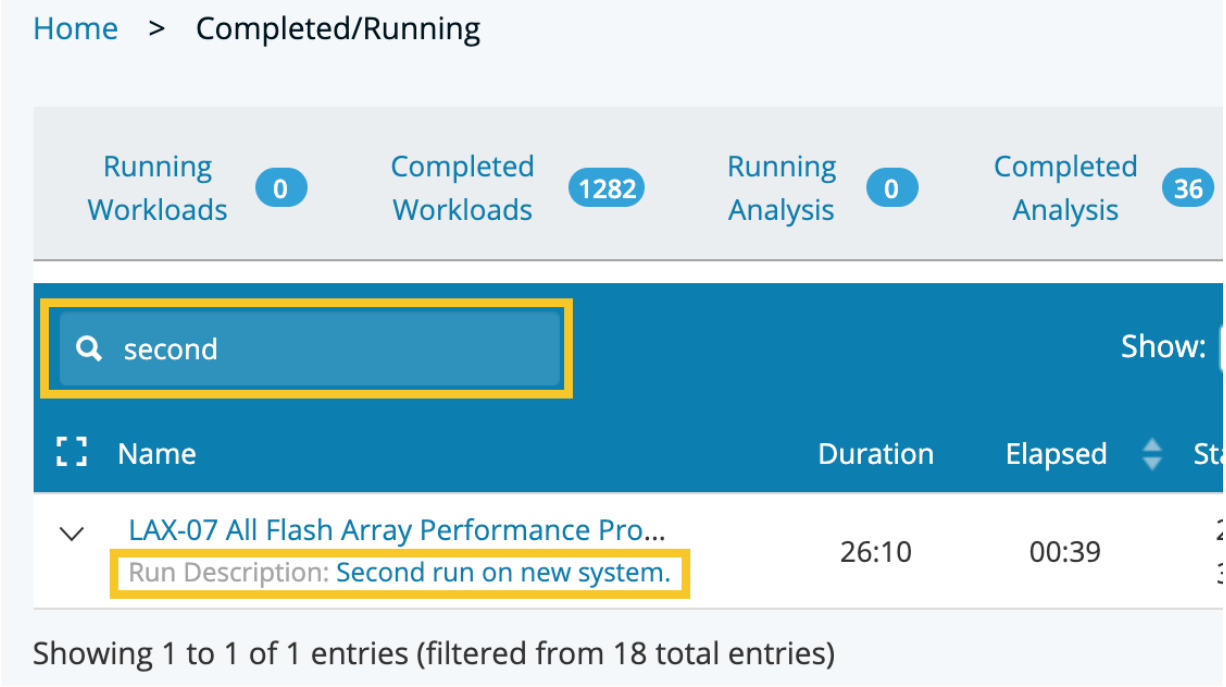


Assign a name to the Iteration Suite that you can easily identify and specify in a search. This name appears in the list of iteration suites to which you have access.

Optionally, add a description to the Iteration Suite. Use the description to specify the objective of the Iteration Suite.

Optionally, add one or more tags to the Iteration Suite.

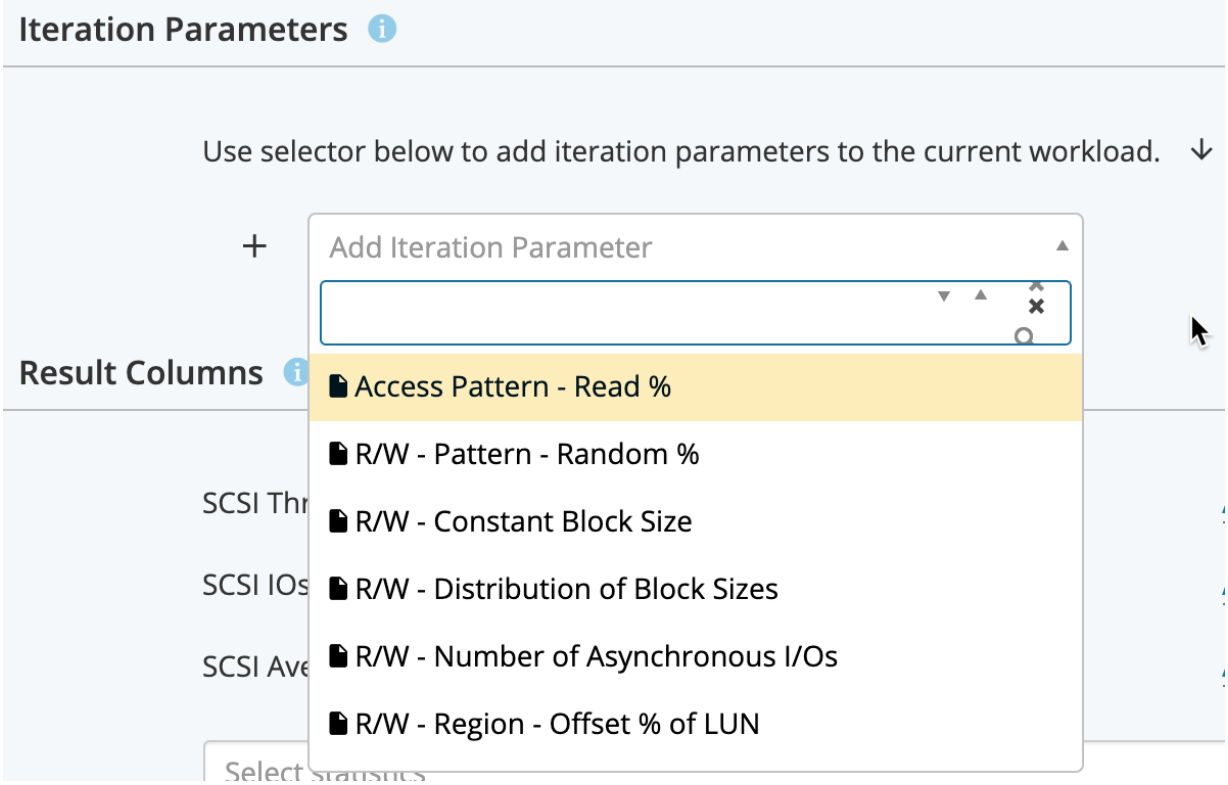
An Iteration Suites tags and names are searchable in WorkloadWisdom.



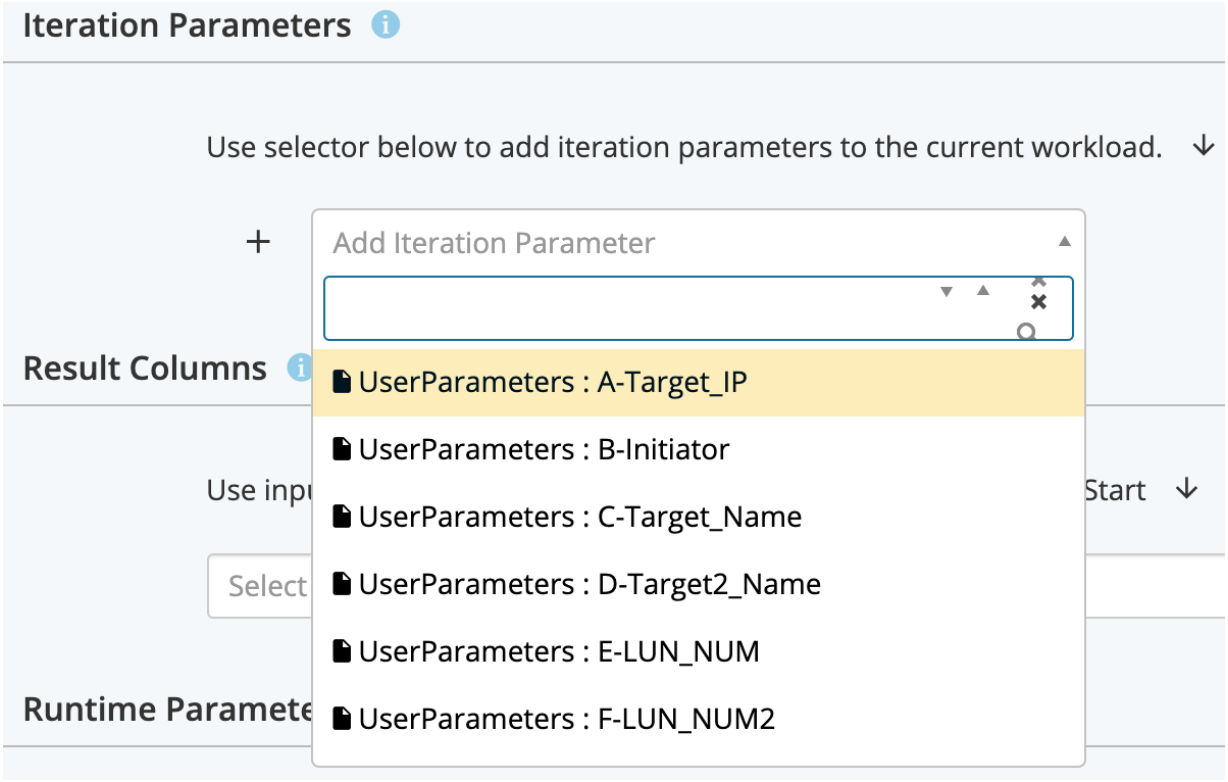
Select a Workload

Click the drop-down menu to select a workload from the list of available workloads that can be used in the Iteration Suite. Two types of workloads are supported by the Iteration Suite: WorkloadWisdom protocol workload models and TDE Workloads.

If you select a WorkloadWisdom protocol workload model (for example, FC Workload), all parameters supported by the selected protocol workload model are available to be used as an Iteration Parameter.



If you select an TDE Workload, all parameters defined in the TDE Workload's User Parameters (UP) file are available to be used as an Iteration Parameter. However, the TDE Workload's UP file must contain one (1) row only.



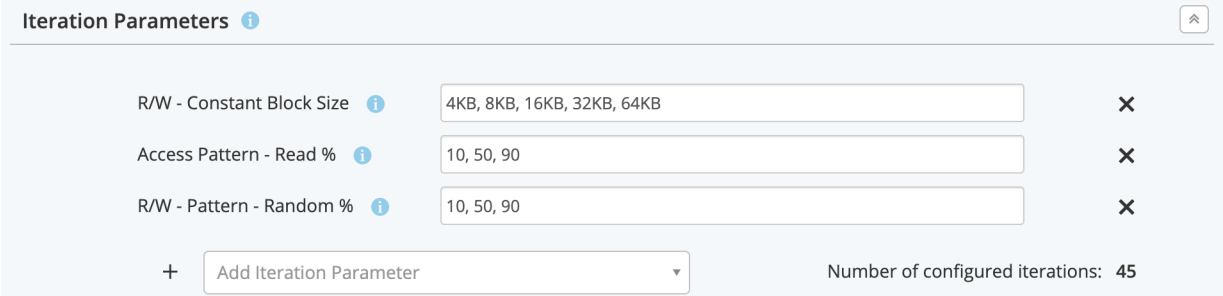
In the Iteration Parameters dropdown, you will see a list of entries that begins with *UserParameters: X-Y* where X is the column index in the User Parameter file, and Y is the custom name you (or the creator of the TDE Workload) provided for each column.

Define Iteration Parameters

From the drop-down menu, select one of the available parameters to iterate over. Once an Iteration Parameter is selected, an additional drop-down menu becomes available for you to optionally select another Iteration Parameter.

For each Iteration Parameter, one or more valid values must be defined.

As you add more Iteration Parameters and define more values for each, the number of Iterations that will run increases.



Multi-value Iteration Parameters

Some parameters require multiple values to be defined for each Iteration. These multi-value parameters require that multiple values are defined to function properly. For example, the parameter I/O – Bin Based Distribution of Request Sizes from the FC Workload requires you to enter multiple Request Sizes as well as the percentage distribution of each Request size for each Iteration.

The following Iteration Parameters are multi-value parameters:

- Data Reduction
- All Bin Based Distribution parameters
- Load – Actions Per Second
- Load – Throughput

Data Reduction

Use the Data Reduction Iteration Parameter to define the *Compression Ratio*, *Deduplication Ratio*, and the *Number of Unique Duplicates* for each iteration.

Syntax:

[C: {Compression Ratio}; D: {Deduplication Ratio}; U: {Number of Unique Duplicates}]

Example:

[C: 2.0; D: 3.5; U: 100], [C: 2.0; D: 4.0; U: 100]

produces the following Iterations:

- Iteration 1
  - Compression Ratio: 2 : 1
  - Deduplication Ratio: 3.5 : 1
  - Number of Unique Duplicates: 100

- Iteration 2
  - Compression Ratio: 2 : 1
  - Deduplication Ratio: 4 : 1
  - Number of Unique Duplicates: 100

All Bin Based Distribution parameters

Use bin-based distribution Iteration Parameters to define a distribution of values and the percentage of each value for each Iteration. Typical examples are block sizes and file sizes.

Syntax:

[{Request Size 1} : {Percentage of Request Size 1} ; {Request Size 2} : {Percentage of Request Size 2}; ... ; {Request Size N} : {Percentage of Request Size N}]

Example:

[4KB : 30; 8KB : 70],[4KB : 50; 8KB : 50]

produces the following Iterations:

- Iteration 1: 30% 4KB Request Size and 70% 8KB Request Size
- Iteration 2: 50% 4KB Request Size and 50% 8KB Request Size

Load – Actions per Second

Use the Load – Actions per Second Iteration parameter to define the maximum Actions per Second (IOPS) and the Number of Concurrent Workers for each Iteration.

Syntax:

[A: {Actions per Second}; W: {Number of Concurrent Workers}]

Example:

[A: 10000; W: 10], [A:20000; W: 15]

produces the following Iterations:

- Iteration 1:
  - Maximum Actions per Second: 10,000
  - Number of Concurrent Workers: 10
- Iteration 2:

- Maximum Actions per Second: 20,000
- Number of Concurrent Workers: 15

Load – Throughput

Use the he Load – Throughput Iteration parameter to define the maximum Throughput and the Number of Concurrent Workers for each Iteration.

Syntax:

[T: {Throughput}; W: {Number of Concurrent Workers}]

Example:

[T: 10000KB; W: 10], [T:20000KB; W: 15]

produces the following Iterations:

- Iteration 1:
  - Maximum Throughput: 10,000 KB/s
  - Number of Concurrent Workers: 10
- Iteration 2:
  - Maximum Throughput: 20,000 KB/s
  - Number of Concurrent Workers: 15

Step Input for Single-value Iteration Parameters

Use Single-value Iteration Parameters to define values in a [min – max: step] 3-tuple instead of individually listing out all values.

Syntax:

[{min value} – {max value}: {step}]

The step must be an integer.

Example:

[5 – 50: 10]

produces

5, 15, 25, 35, 45

Multiplier Input for Single-value Iteration Parameters

Use multiplier input for single-value Iteration parameters to define values in a [min – max: multiplier] 3-tuple instead of individually listing out all values.

Syntax:

[{min value} - {max value}: x{multiplier}]

The multiplier must be an integer.

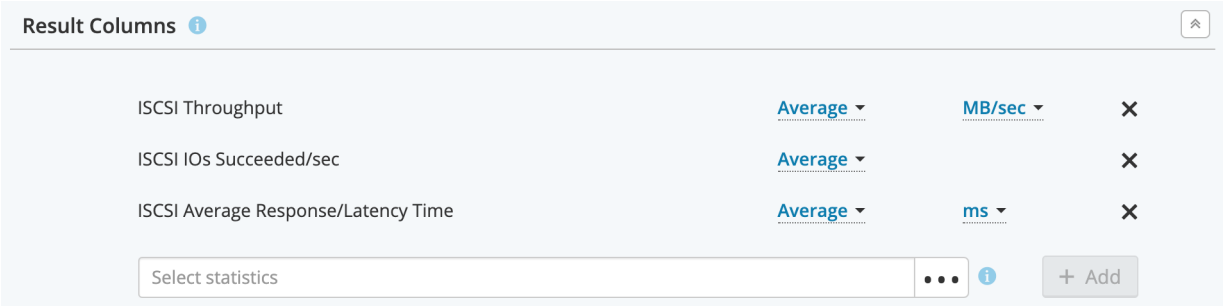
Example:

[1KB – 64KB: x2] produces 1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB

Select Statistics for Iteration Results

When an Iteration Suite is started, an Iteration Results table is provided to display the results, along with a brief description of the Iteration Parameters, for each Iteration. The Iteration Results table can be accessed while the Iteration Suite is running and after the Iteration Suite completes.

The Statistics you select will be displayed in the Iteration Results table. However, WorkloadWisdom collects all Statistics available from the selected Workload even though they are not displayed in the Iteration Results table. You can access all Statistics collected from each Iteration by clicking on the Status for a specific Iteration, which will take you to the full standard Results Dashboard provided for the Selected Workload.



For most Workloads, the default Statistics are protocol Throughput (average), protocol IOs Succeeded/sec (average) and protocol Average Response/Latency Time (average). You can add and/or remove any available statistic. It is recommended that three (3) statistics are selected for display and performance reasons.

Define Runtime Parameters

Select an applicable test bed from the drop-down menu for this Iteration Suite. The selected test bed will be used by the Iteration Suite for all Iterations.

Use iteration duration to define the time duration for each Iteration. The proper duration to use is highly dependent on the selected Workload. For example, if you selected a Workload that creates a huge number of files, then you will want to define a larger duration value. In most cases, a minimum of 30 seconds is recommended.

Use Delay Between Iterations to define a non-zero wait time between the end of the current Iteration and the beginning of the next Iteration. Generally, you want to do this to give the SUT some time to complete processing the closing of all connections from the current Iteration before the new connections are requested from the next Iteration.

Three options are available for Run Pre-Test: do not run, run once, or run all.

- *Do not run*. Pre-Test is not run by the Iteration Suite at all

Use this option if the SUT has already been preconditioned (for FC/iSCSI), or has the required folders and files created (for SMB/NFS)

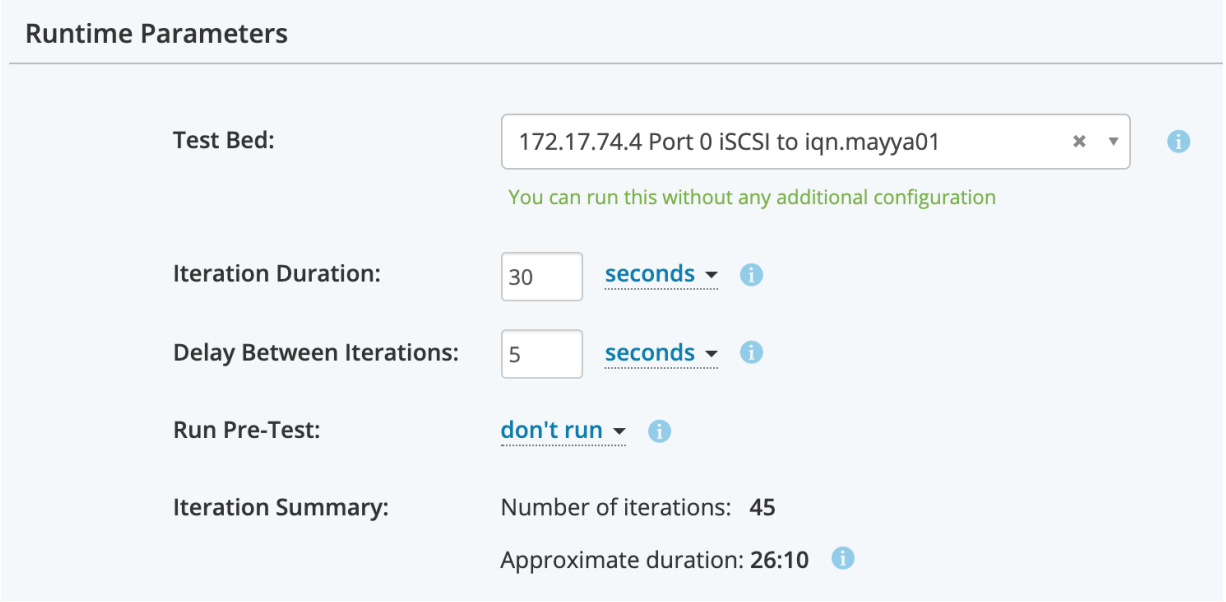
- *Run once*. Pre-Test runs once when the Iteration Suite starts

Use this option if the SUT has not already been preconditioned, or does not have the required folders and files created, and you want to precondition the SUT or populate the required folders and files once before the first Iteration starts

- *Run all*. Pre-Test runs prior to each Iteration.

Use this option if you want to precondition the SUT, or create the required folders and files, prior to the start of each Iteration

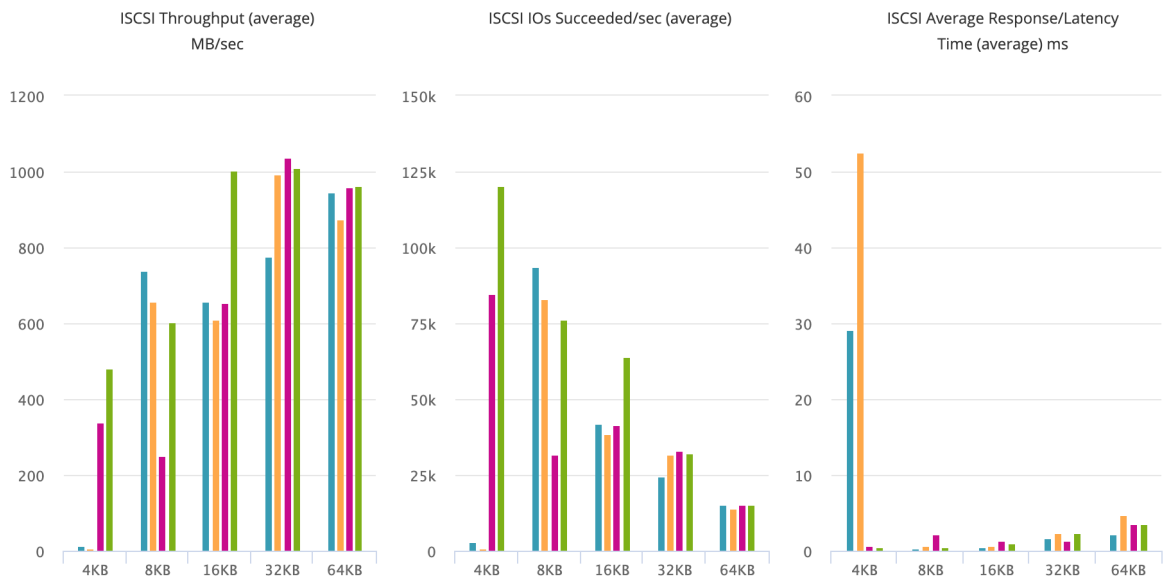
The Iteration Summary provides a read-only summary of the total number of Iterations that will run, and the approximate duration of the entire Iteration Suite.



Run Iteration Suite

When the Iteration Suite is started, you are taken to the Iteration Results page, where you can view the current status of the Iteration Suite, the current status of each Iteration, and the results (if available) from the selected Statistics for each Iteration. The Iteration Explorer is available automatically at the end of the Iteration Suite Run.

Iteration Explorer



Highest/lowest values

#	Status	Duration	R/W - Constant Block Size	Access Pattern - Read %	R/W - Pattern - Random %	ISCSI Throughput (average)	ISCSI IOs Succeeded/sec (average)	ISCSI Average Response/Latency Time (average)
Iteration			Parameters			↑ **Highest values** through all iterations		
41	Finished	00:32	64KB	50	50	**1,569.4 MB/sec**	25074.78	1.2 ms
9	Finished	00:32	4KB	90	90	487.0 MB/sec	**121820.795**	0.4 ms
2	Finished	00:32	4KB	10	50	2.4 MB/sec	590.567	**52.6 ms**
Iteration			Parameters			↓ **Lowest values** through all iterations		
2	Finished	00:32	4KB	10	50	**2.4 MB/sec**	590.567	52.6 ms
2	Finished	00:32	4KB	10	50	2.4 MB/sec	**590.567**	52.6 ms
10	Finished	00:32	8KB	10	10	738.7 MB/sec	93453.914	**0.3 ms**

Iteration Results

Export to CSV | Show by 25 entries

#	Status	Duration	R/W - Constant Block Size	Access Pattern - Read %	R/W - Pattern - Random %	ISCSI Throughput (average)	ISCSI IOs Succeeded/sec (average)	ISCSI Average Response/Latency Time (average)
1	Finished	00:32	4KB	10	10	12.6 MB/sec	3140.85	29.1 ms
2	Finished	00:32	4KB	10	50	2.4 MB/sec	590.567	52.6 ms
3	Finished	00:32	4KB	10	90	185.2 MB/sec	46332.562	0.8 ms
4	Finished	00:32	4KB	50	10	140.7 MB/sec	35192.29	3.0 ms
5	Finished	00:32	4KB	50	50	220.8 MB/sec	55237.236	2.3 ms
6	Finished	00:32	4KB	50	90	396.9 MB/sec	99289.866	1.1 ms
7	Finished	00:32	4KB	90	10	338.5 MB/sec	84679.334	0.7 ms
8	Finished	00:32	4KB	90	50	481.0 MB/sec	120306.006	0.5 ms
9	Finished	00:32	4KB	90	90	487.0 MB/sec	121820.795	0.4 ms
10	Finished	00:32	8KB	10	10	738.7 MB/sec	93453.914	0.3 ms
11	Finished	00:32	8KB	10	50	656.8 MB/sec	83094.358	0.6 ms
12	Finished	00:32	8KB	10	90	349.0 MB/sec	44159.658	0.8 ms
13	Finished	00:32	8KB	50	10	271.7 MB/sec	34379.276	3.0 ms
14	Finished	00:32	8KB	50	50	418.9 MB/sec	52994.567	1.2 ms
15	Finished	00:32	8KB	50	90	693.6 MB/sec	87749.204	1.0 ms

Iteration Explorer

The Iteration Explorer provides KPI Charts on-demand for a selected set of Iteration Runs.

The Iteration Explorer is only available after an Iteration Suite finishes running. While an Iteration Suite is in the Running state, Iteration Explorer will not contain any data.

The Iteration Explorer contains three sections: KPI Charts, Iteration Parameters & Colors Legend, and Iterations Selection.

KPI Charts

KPI Charts are used to gain insight into the KPI trends of the SUT across a range of changing IO characteristics.

KPI Charts are Bar Charts representation of the KPI Stats defined by the user in the Iteration Suite. For each KPI Stat defined in the Iteration Suite, a KPI Chart is generated. Up to four KPI Charts per row are displayed. If more than four KPI Stats are defined, the KPI Charts will appear on subsequent rows.

The amount of data that appears on the KPI Charts depends on the selections made in the Iterations Selection section of Iteration Explorer.

Iteration Parameters & Colors Legend

For each Iteration Run selected to be included in the KPI Charts, a unique color is selected. This section provides a legend of the color and the Iteration Parameter Values associated with each selected Iteration Run.

Iterations Selection

You must select the following for KPI Charts to show data:

- One and only one iteration parameter to use as the X-axis.
- At least one iteration parameter value from each iteration parameter.
- Click **Update Charts** after making your selections.

For best results and presentation, use the following guidelines:

- Use 2~3 KPI Stats and 6 or less Iteration Parameters when defining the Iteration Suite.
- Keep under 100 total Iterations when defining the Iteration Suite.
- Select 3~5 Iteration Parameter Values from the Iteration Parameter that is used as the X-axis.
- Select 32 or fewer Iteration Runs to include in the KPI Charts.
- Select a common “big impact” Iteration Parameter as the X-axis, such as IO Request Size that typically produces consistently higher IOPS as the IO Request Size decreases. Port Queue Depth is also a good “big impact” Iteration Parameter to use as the X-axis.

When the number of Iteration Parameter Values you select produces more than 32 Iteration Runs to include in the KPI Charts, a message appears next to Update Charts, displaying the total number of Iterations you have selected. As suggested above, keep the number to 32 or less for best displaying results, as there is limited space on each KPI Chart.

Iteration Analyst

The Iteration Analyst automatically finds the iterations that produce performance results that stand out.

Highest/Lowest Values Finder

The Highest/Lowest Values Finder displays the Iteration that produces the highest and lowest numerical value for each selected Statistic respectively.

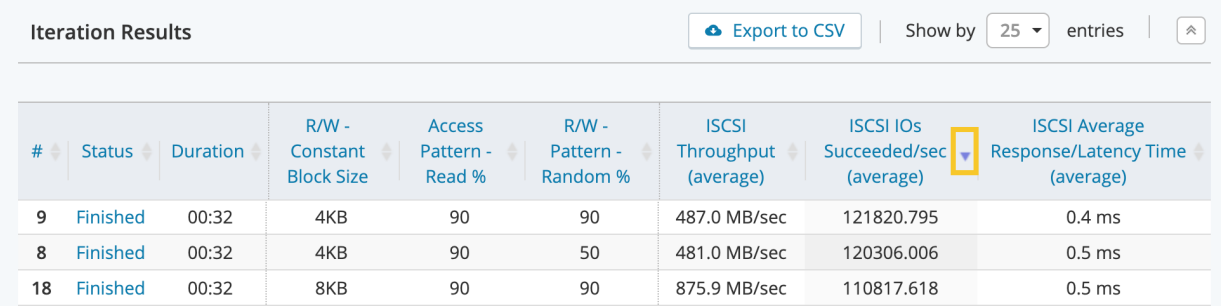
It is important to understand that “highest” does not always imply best, and “lowest” does not always imply worst. For example, highest SCSI Average Response/Latency Time

indicates poor performance as opposed to good performance. Similarly, an Iteration with 8KB Request Size that produces a higher Throughput value does not imply it is better than an Iteration with 1KB Request Size that produces a slightly lower Throughput value.

Additional Analyst options will be added in future releases.

Iteration Results

Each of the column header of the Iteration Results table can be sorted by clicking on the column header. For example, if you want to sort the Iteration Results table by the protocol Throughput (average) from highest to lowest, click the column header protocol Throughput (average) until you see the down arrow next to the column header.



#	Status	Duration	R/W - Constant Block Size	Access Pattern - Read %	R/W - Pattern - Random %	ISCSI Throughput (average)	ISCSI IOs Succeeded/sec (average)	ISCSI Average Response/Latency Time (average)
9	Finished	00:32	4KB	90	90	487.0 MB/sec	121820.795	0.4 ms
8	Finished	00:32	4KB	90	50	481.0 MB/sec	120306.006	0.5 ms
18	Finished	00:32	8KB	90	90	875.9 MB/sec	110817.618	0.5 ms

To access all Statistics collected for a specific Iteration, click the Status for the specific Iteration, and you will be taken to the standard Results Dashboard for the selected Iteration.

Scheduled Jobs

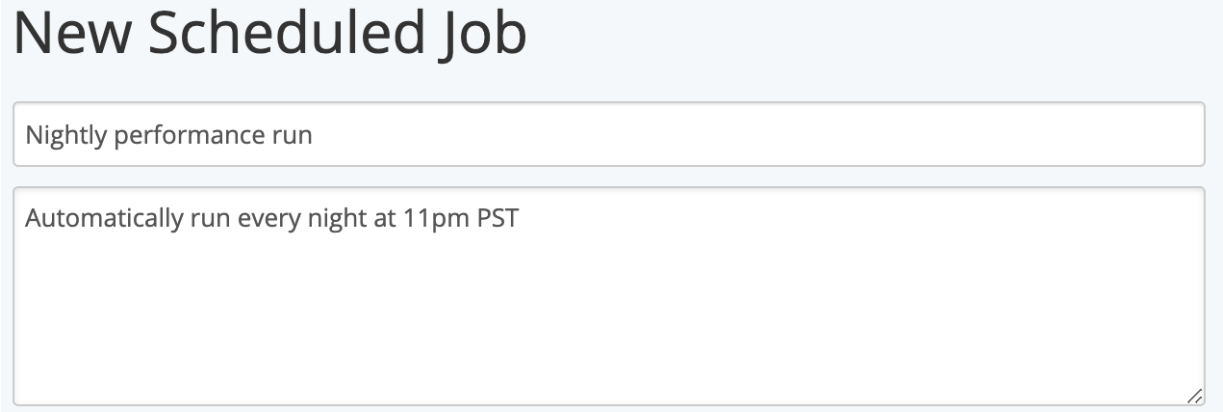
Use *Scheduled Jobs* to define a list of Tasks to run automatically at a future time. Tasks are a set of tests that will run in sequence within the scope of a Scheduled Job. The types of Tasks supported are:

- Workload Tests
- Workload Suites
- Iteration Suites

Each Job that is created is always scheduled in WorkloadWisdom, whether it is scheduled to run now (i.e. the moment you finish creating the Job) or scheduled to run at a future time. Once a Job is started, it can no longer be edited, and can only be stopped. Once a Job is stopped, it is automatically archived. Future releases of WorkloadWisdom will provide additional options for scheduling, editing, and viewing Job Run Summary results.

New Scheduled Job

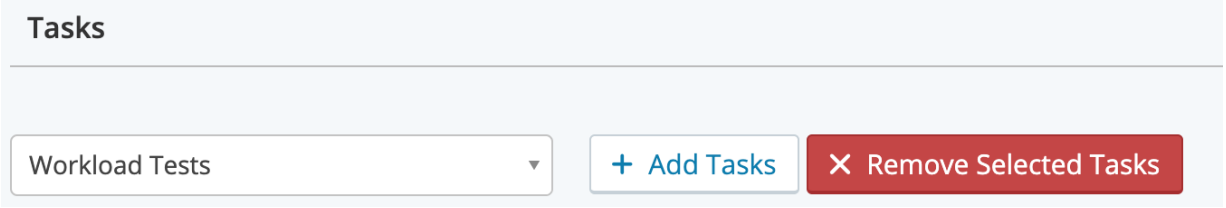
Start by creating a new Scheduled Job by clicking the **New Scheduled Job** button.



On the New Scheduled Job edit page, give it a name, and optionally the purpose of the Scheduled Job in the Description section, as this makes it easier to search for at a later date.

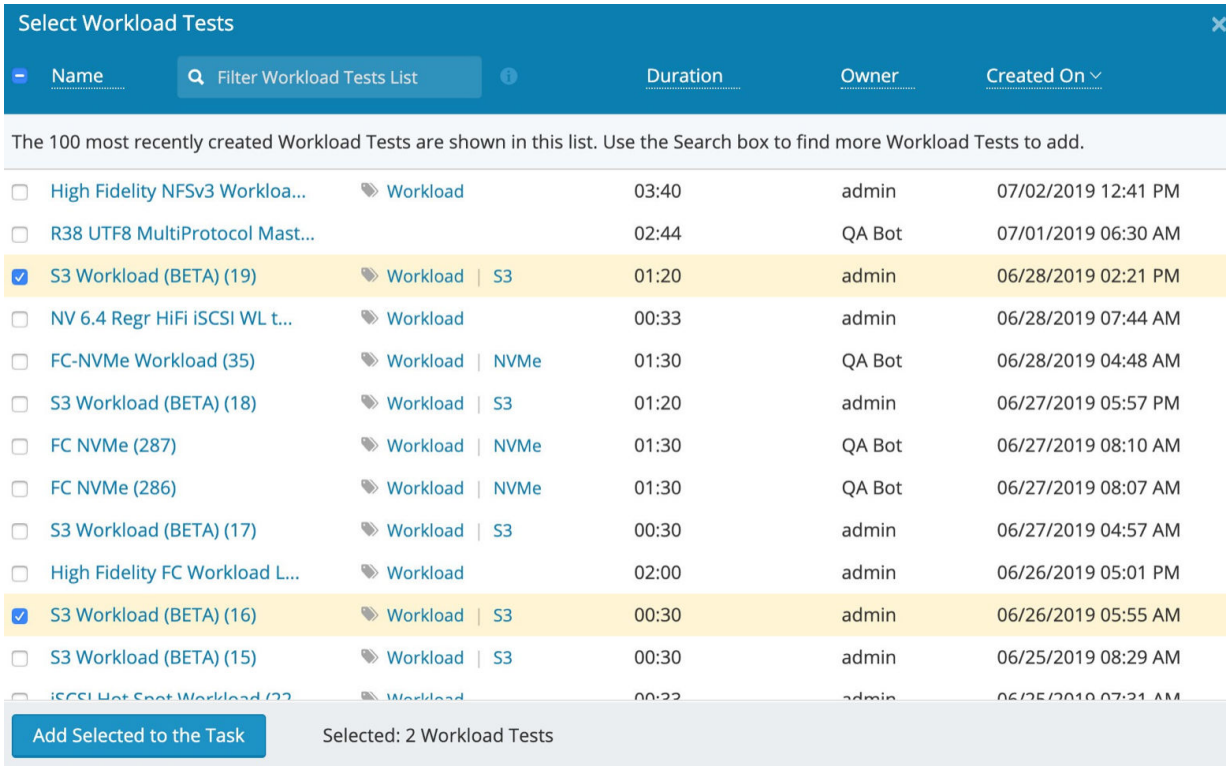
Tasks

Start by adding at least one Task to the Job.

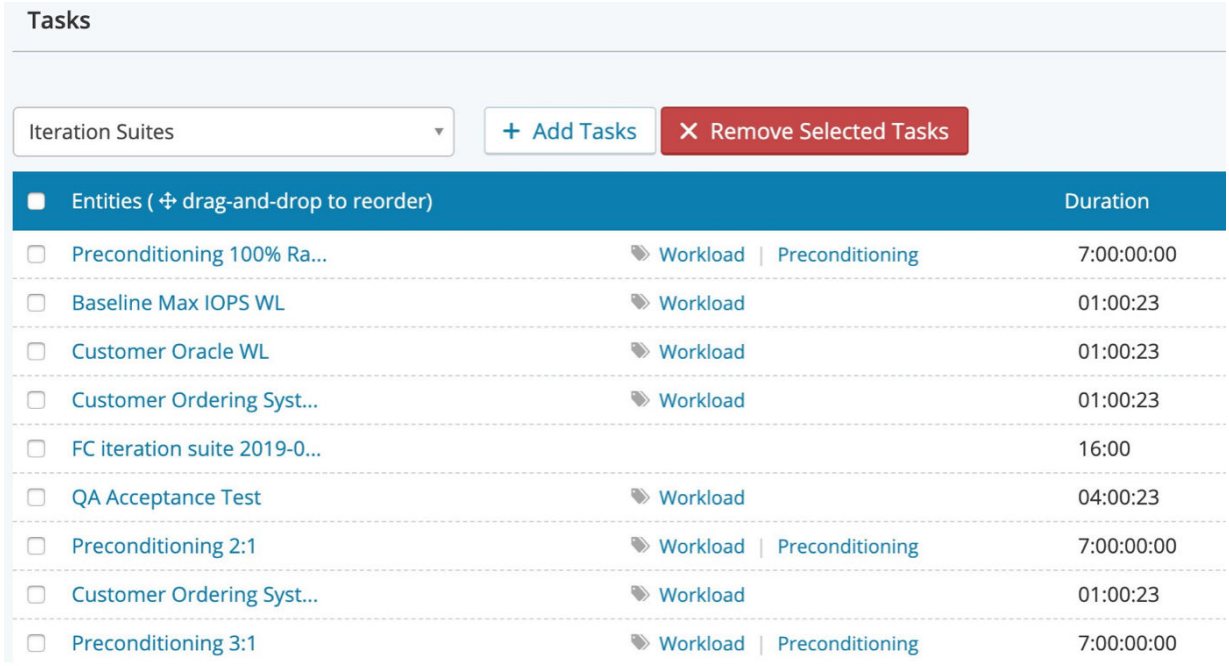


Click the drop-down menu to select a different type of Task to add, and you can mix different types of Tasks in the Job. For example, you can start with a Preconditioning Workload to precondition the SUT a certain way, then run an Iteration Suite that runs 100 iterations that permutate over 100 different combinations of workload profiles, and then run another Preconditioning Workload to precondition the SUT a different way, and then run the same or a different Iteration Suite, and so on. Or some other combination of mixing Workload Tests, Workload Suites, and Iteration Suites.

When you click the **+ Add Tasks** button, a Task selection window opens up. You can scroll and select one or more items, or you can use the search box and / or the available sort options to find your item faster. For performance reasons, the most 100 recently created items that match the current search and sort criteria are shown in the Task selection window.



Once you finish adding the Tasks, you will see the list of Tasks that will run in sequence when the Job starts. For example, you will see a screen similar to this.



If you want to change the order of the Tasks, simply click-and-hold on a Task, then drag-and-drop it to the right position.

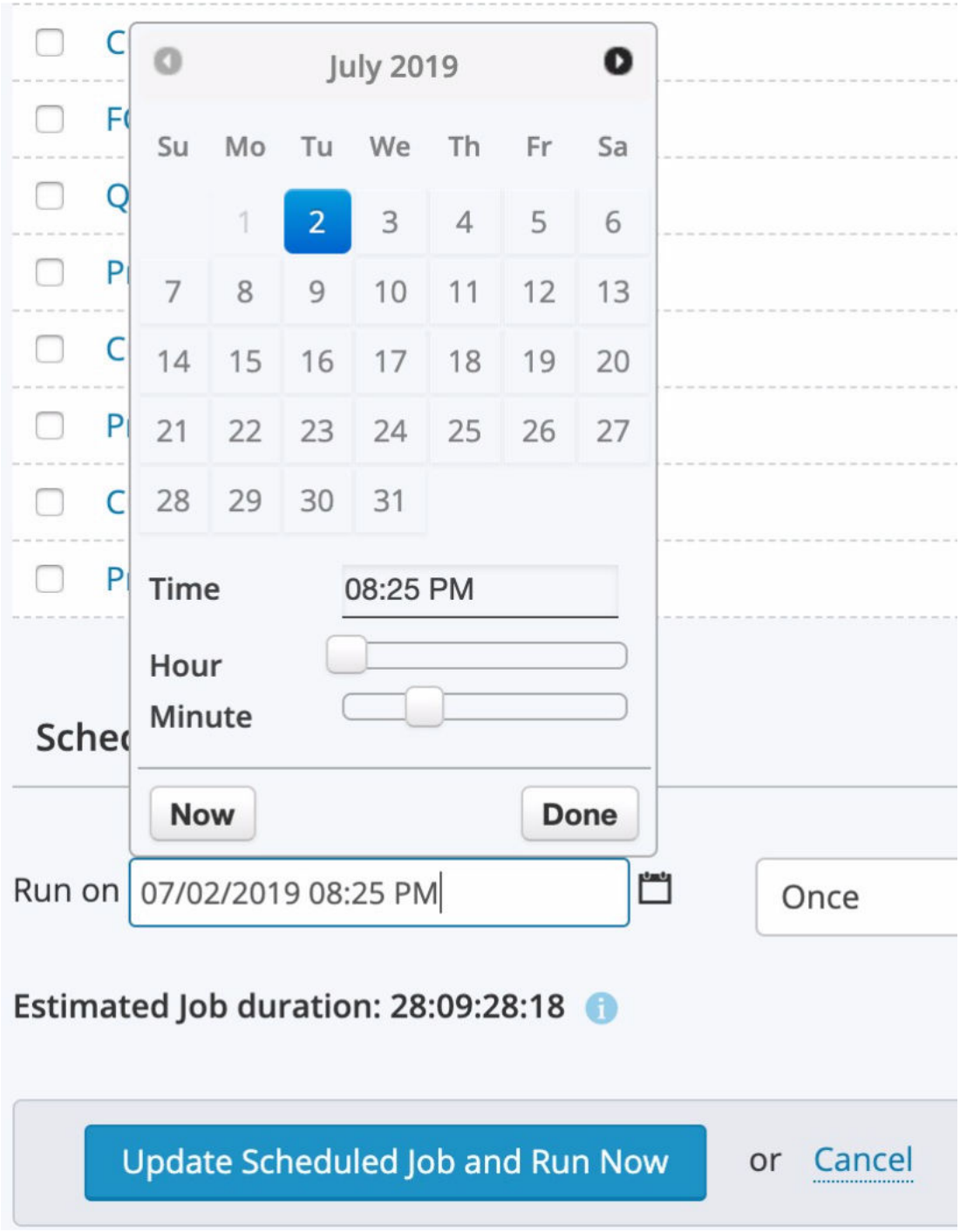
Schedule

A created Job is always scheduled in WorkloadWisdom, whether it is right now (i.e. the moment you want to apply your changes) or at a future time.

When you click on the date time window, a date time browser will open where you can select a future date time for the Job to run, or you can click **Now** to have the Job run immediately when you finish updating the Job. In addition, you can specify that the Job will be scheduled to run Once, or on a recurring basis. Click on the drop-down menu to find these options:

- Once: The Job is scheduled to run one time, and will start at the configured **Run on** date and time
- Daily: The Job is scheduled to run every calendar day, and the first run will start at the configured **Run on** date and time
- Weekly: The Job is scheduled to run every 7 calendar days, and the first run will start at the configured
- **Run on** date and time Monthly: The Job is scheduled to run every 28 calendar days, and the first run will start at the configured **Run on** date and time

When the **Run on** time is the current or before the current time, the button automatically changed to **Create / Update Scheduled Job and Run Now** to make it clear that the Job will run immediately.

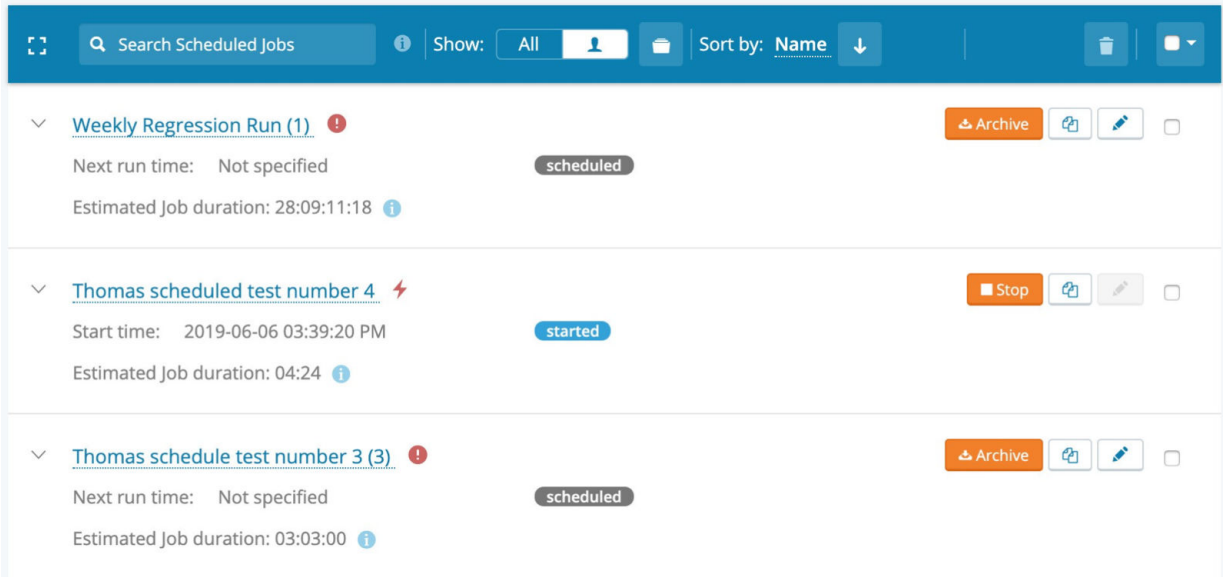


When the **Run on** time is a future time, the button will read **Create / Update Scheduled Job.**

The **Estimated Job duration** is a numerical sum of the configured duration of all Tasks. This is not the enforced or guaranteed Job duration, especially if a Task directly or indirectly contains Pre-Test(s), Preconditioning Workload(s), or Cleanup Workload(s) because these workloads[3] are configured with the maximum upper limit of 7 days as it is not possible to know how long they actually take, but they will terminate as soon as they're completed so in reality, most of these "special" workloads will complete far earlier than the default limit of 7 days. The major contributors that can drastically change the actual Job run duration are resource unavailability (e.g. if a Test Bed is busy) and the performance of the SUT.

Scheduled Jobs List

Click on the **Scheduled Jobs** link from either the homepage or the hamburger menu button from the top right corner of the screen to see a list of Scheduled Jobs.



A Scheduled Job can be in one of the following states:

- Scheduled: This Job is not currently running, but is scheduled to run at a future time
- Started: This Job is currently running
- Archived: This Job has been completed and is no longer scheduled to run at a future time

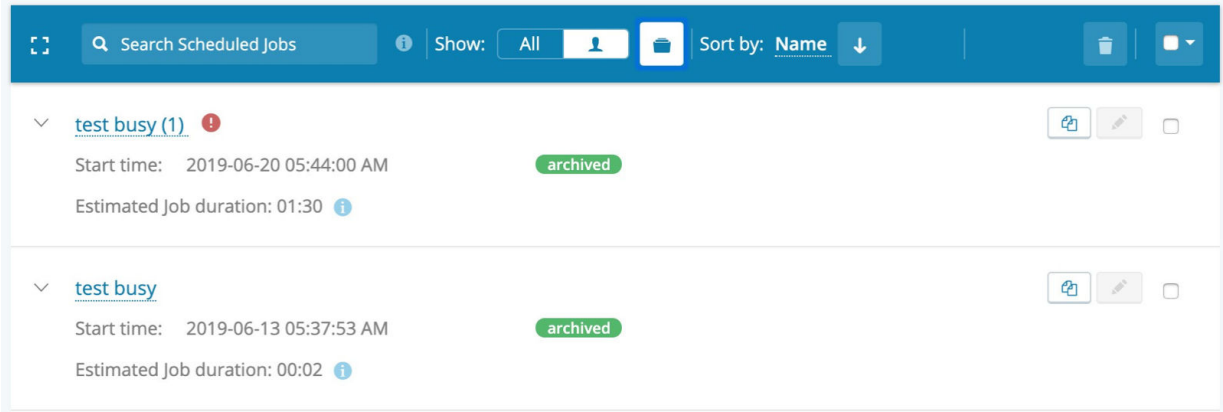
A valid Scheduled Job is always scheduled the moment it is created. Once it is started, you can manually stop it by clicking the **Stop** button. Once a Scheduled Job has finished

[3]The VDI over NFSv3 Workload is configured with a 1 day maximum duration value.

running or has been stopped by the user, then it will go to the Archived state. If the Scheduled Job has future recurring runs, then a new instance of the same Scheduled Job will be created and scheduled. You can also manually choose to discontinue all future runs of a Scheduled Job by clicking the **Archive** button, forcing it to go to the Archived state.

To see a list of archived Jobs, click on the button. A list of archived Jobs will appear. You cannot move a Job out of the Archived state.

However, you can click on the button to clone the archived Job, which makes a copy of the Job and opens up the edit page for you to create a new Job out of the archived Job.



If some issues are detected for a Scheduled Job that prevents it from running as expected, an icon appears next to it.

This Scheduled Job doesn't have a valid configuration, such as missing **Run on** time, missing resource for a Task, or missing Test Bed.

This Scheduled Job currently overlaps with another Scheduled Job's scheduled start time.

When a Job is in the Started state, the Tasks will run in sequence. If a Task cannot run due to resource availability reasons, for example if its Test Bed is currently in use, then it will automatically enter a queue in the system and wait for the resource to become available. Multiple Tasks from multiple Scheduled Jobs can be in the queue, and the queued Tasks will run on a first come first serve basis for the same resource. *For example, if Task A waiting on Test Bed 1 enters the queue before Task B waiting on Test Bed 2 enters the queue, but Test Bed 2 frees up first, then Task B will run before Task A.*

Resources for *Run Workload Tests*

Workload Models

The *Workload Models* page is a library of workload models templates, workload examples, and preconditioning workloads that are shipped with the current version of WorkloadWisdom. By default, the workload models filter is applied. You can view workload examples by clicking *Workload Examples*.

Four filters are available:

- *Workload Templates*. List of protocol workload models and application workload models that you can select from to create a workload test from and specify your own I/O profiles. See Workload Tests Concepts [90].
- *Workload Examples*. List of workload examples with pre-defined I/O profiles.
- *Preconditioning Workloads*. List of "initialization" workloads that are designed to optionally Precondition a storage array before running a workload test.
- *Post-Test Cleanup*. Removes the files and directories in a location specified by a test bed.

Select a workload model and click Clone to add an instance of the selected Workload to the workloads list, where you can modify, run and delete the Workload. The workload models displayed in this list are read-only and cannot be modified.

Workload Templates

The following table includes the Workload Templates that are included in WorkloadWisdom. Many of the templates share common configurations and concepts. The table includes a description of each template and a link to the configurations available in the template. When more than one protocol is listed, the actual protocol used is dependent on the test bed or option in a workload (HiFi SMB WL), and automatically set at runtime for easier workload reuse.

For additional information, refer to Workload Tests Concepts [90].

Table 3. Default Workload Templates in WorkloadWisdom

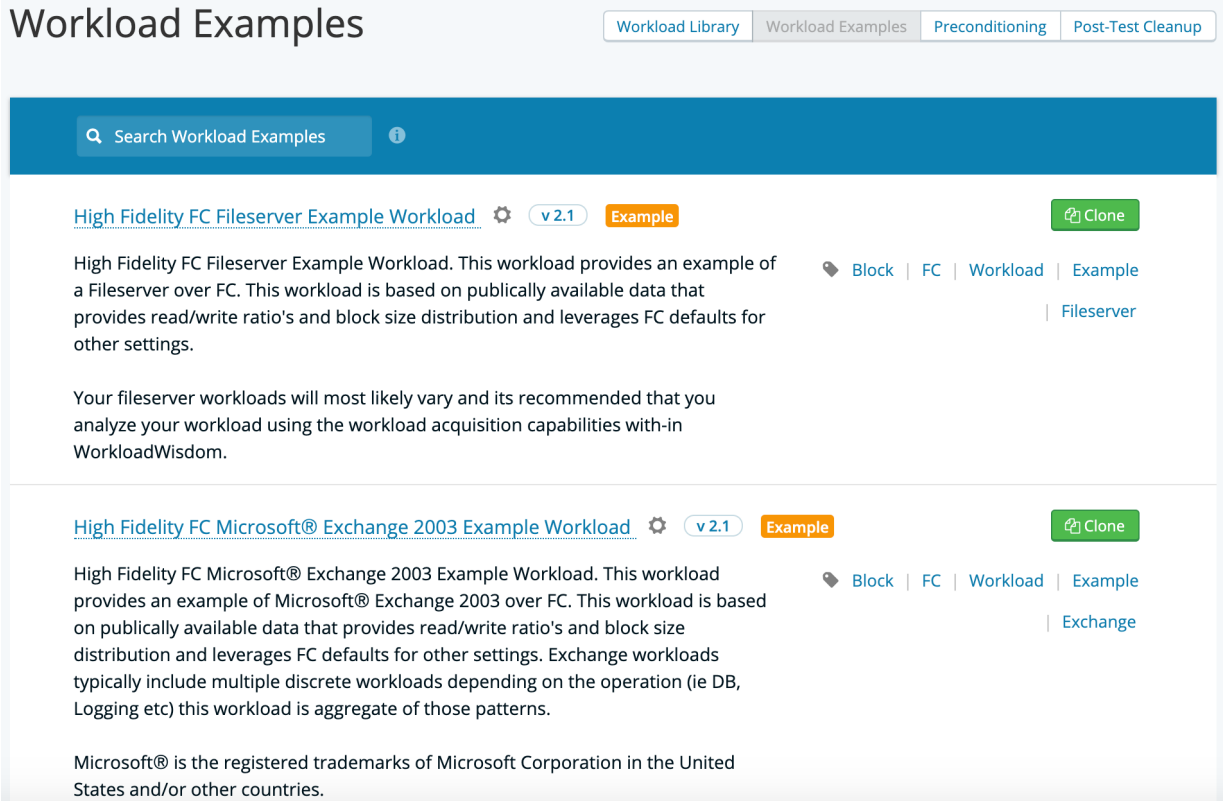
Workload	Protocols	Description	Workload Category
FC Hot Spot Workload	FC-SCSI	High Fidelity FC workload with the additional capability to enable multiple LUN regions to be specified. Most real-world workloads have hotspots though it can be difficult to know to what degree. It is best to compare tests with hotspots present to ones without to determine the impact to performance of hotspots.	Protocol Workload: Constant
High Fidelity FC-SCSI Workload	FC-SCSI	Intended to offer broad coverage of common FC capabilities. It is a constant that is good for testing changes external to the workload or for iteration testing. It simplifies configuration to a single read and write location over the hotspot workloads.	Protocol Workload: Constant
High Fidelity FC-NVMe Workload	FC-NVMe	Intended to offer broad coverage of common FC-NVMe capabilities. It is a constant workload that is good for testing changes external to the workload or for iteration testing.	Protocol Workload: Constant **Note**: Some standard Constant Protocol Workload functionalities do not apply to FC-NVMe Workloads. Refer to the FC-NVMe Block Workload section for details.
High Fidelity NFSv3 Workload	NFSv3	Intended to offer broad coverage of common NFSv3 capabilities. It is a constant that is good for testing changes external to the workload or for iteration testing. It can be combined in using composite workloads to have file hotspots using different file layouts and load for each group of files.	Protocol Workload: Constant

Workload	Protocols	Description	Workload Category
High Fidelity NFSv4 Workload	NFSv4	Intended to offer broad coverage of common NFSv4 capabilities. It is a constant that is good for testing changes external to the workload or for iteration testing. It can be combined in using composite workloads to have file hotspots using different file layouts and load for each group of files.	Protocol Workload: Constant
High Fidelity NFSv4.1 Workload	NFSv4.1	Intended to offer broad coverage of common NFSv4.1 capabilities. It is a constant that is good for testing changes external to the workload or for iteration testing. It can be combined in using composite workloads to have file hotspots using different file layouts and load for each group of files.	Protocol Workload: Constant
High Fidelity SMB Workload	CIFS/SMB, SMB2, SMB3	Intended to offer broad coverage of common SMB capabilities. It is a constant that is good for testing changes external to the workload or for iteration testing. It can be combined in using composite workloads to have file hotspots using different file layouts and load for each group of files.	Protocol Workload: Constant
High Fidelity iSCSI Workload	iSCSI	Intended to offer broad coverage of common iSCSI capabilities. It is a constant that is good for testing changes external to the workload or for iteration testing.	Protocol Workload: Constant
S3 Workload	Amazon S3	Purpose-built workload for Amazon S3 that provides more flexibility and configuration options that are specific and relevant to Amazon S3 use cases.	Protocol Workload: Constant
Object Storage Workload	Amazon S3, OpenStack-Swift	Provides a common workload for Amazon S3 and OpenStack-Swift. A single workload can be configured either to work with both protocols depending on the test bed or only for a specific protocol. This enables more direct control over the commands.	Protocol Workload: Constant
VDI Runtime FC Workload	FC	Simulates Run Time activities of multiple desktops based on the Linked Clone model and is available for FC. Configuration is done in terms familiar to VDI administrators w/o needed to model at a lower storage knowledge level.	Application Workload

Workload	Protocols	Description	Workload Category
VDI Runtime iSCSI Workload	iSCSI	Simulates Run Time activities of multiple desktops based on the Linked Clone model and is available for iSCSI. Configuration is done in terms familiar to VDI administrators w/o needed to model at a lower storage knowledge level.	Application Workload
VDI Workload	NFSv3	Simulates Bootstorm activities of multiple desktops based on the Linked Clone model and is available for NFSv3. Configuration is done in terms familiar to VDI administrators w/o needed to model at a lower storage knowledge level.	Application Workload
iSCSI Hot Spot Workload	iSCSI	Similar to the High Fidelity iSCSI workload except that multiple LUN regions can be specified. Most real-world workloads have hotspots though it can be difficult to know to what degree. It is best to compare tests with hotspots present to ones without to determine the impact to performance of hotspots	Protocol Workload: Constant
NFSv3 Temporal Workloads	NFSv3	Provides the capability to have many aspects of an NFSv3 workload vary over time in a fashion that models a real production environment. It offers a level of realism unique to the testing industry.	Protocol Workload: Temporal
NFSv4 Temporal Workloads	NFSv4	Provides the capability to have many aspects of an NFSv4 workload vary over time in a fashion that models a real production environment. It offers a level of realism unique to the testing industry.	Protocol Workload: Temporal
SCSI Temporal Workloads	FC-SCSI, iSCSI	Provides the capability to have many aspects of an FC/iSCSI workloads vary over time in a fashion that models a real production environment. It offers a level of realism unique to the testing industry. Hotspots including drift can be configured.	Protocol Workload: Temporal
SMB Temporal Workload	SMB2	Provides the capability to have many aspects of a SMB2 workload vary over time in a fashion that models a real production environment. It offers a level of realism unique to the testing industry.	Protocol Workload: Temporal

Workload Examples

Workload Examples provide you with example workloads for file and block protocols, using typical workloads in the enterprise, web servers, Microsoft® Exchange or Databases.



Access *Workload Examples* from the Workload Library by clicking *Workload Examples* from the *Run Workload Tests* section of the *Home* page.

All workloads are provided as example workloads and do not necessary imply the best or most common example. It is always best to use the Workload Data Importer to collect real data from your own environment or from your customer's environment for Workloads. The examples are intended as a starting point.

None of the workloads take into consideration load properties such as typical IOPs/ throughput as these vary greatly in each environment. The examples are based on public industry research available on the internet and do not include direct research by Virtual Instruments unless otherwise stated.

Example File Workloads

The library of Example File Workloads includes versions of each Workload for Network File System (NFS) version 3 and 4.1, and Server Message Block 2 (SMB).

For a more detailed description of the example workloads see Example Workload Descriptions [273]."

Example Block Workloads

The library of Example Block Workloads provides a version of each Workload for both Fibre Channel and iSCSI.

For a more detailed description of the example Workloads see Example Workload Descriptions [273]."

Conditions

Use conditions to define an action trigger-based on comparison against a statistic, a log message or a Boolean value. Conditions can be triggered during or after a workload test.

There are four types of conditions:

- *Statistics Comparison*. Performs a numerical comparison of a WorkloadWisdom statistic to a user-defined value or another WorkloadWisdom statistic.
- *Match Text in Log*. Performs a regular expression (regex) match in the log file produced from each test run.
- *Combination*. Combines two or more Conditions together using logic operations AND, OR, or XOR.
- *Script*. Executes a custom JavaScript code that results in a Boolean value when executed.

The following default conditions are shipped with WorkloadWisdom:

- Scenarios or Actions Failed or Aborted
- Actions Aborted
- Actions Failed
- Scenarios Aborted
- Scenarios Failed

Click **+New Condition** to define a new condition.

Statistics Comparison

Trigger Parameters

Specify the “when” and “where” to apply the Condition to.

- *Trigger on*. Specify the Logical Ports to apply the Condition to in a selected test bed. You can specify the Condition to trigger on all Logical Ports (All), a specific Logical Port, all Client side Logical Ports (Client total), all Server side Logical Ports (Server total), any Client side Logical Port (Client any), any Server side Logical Port (Server any), or any Logical Port (Any, which is the default).
- *From ___ to ___ percent/seconds of test time*. Specify when the Condition can be triggered during a test run. For example:
  - *0 to 100 percent*: the Condition can be triggered at any time during the test run.
  - *25 to 75 percent*: the Condition can be triggered after 25% of the total test duration from the beginning of the test run, and before 75% of the total test duration from the beginning of the test run.
  - *30 to 45 seconds*: the Condition can be triggered after 30 seconds from the beginning of the test run, and before 45 seconds from the beginning of the test run.
- *After triggering, wait ___ seconds before checking again*. Specify a "grace period" after a Condition is triggered before the same Condition can be triggered again. For example, if you have a Condition that is triggered when Average Response Time is above 30ms, and you specify a wait time of 5 seconds, then the earliest the Condition can be triggered for a second time is 5 seconds after the first time the same Condition is triggered, regardless of the observed Average Response Time during that wait time interval, provided that the observation time is within the from ___ to ___ range.

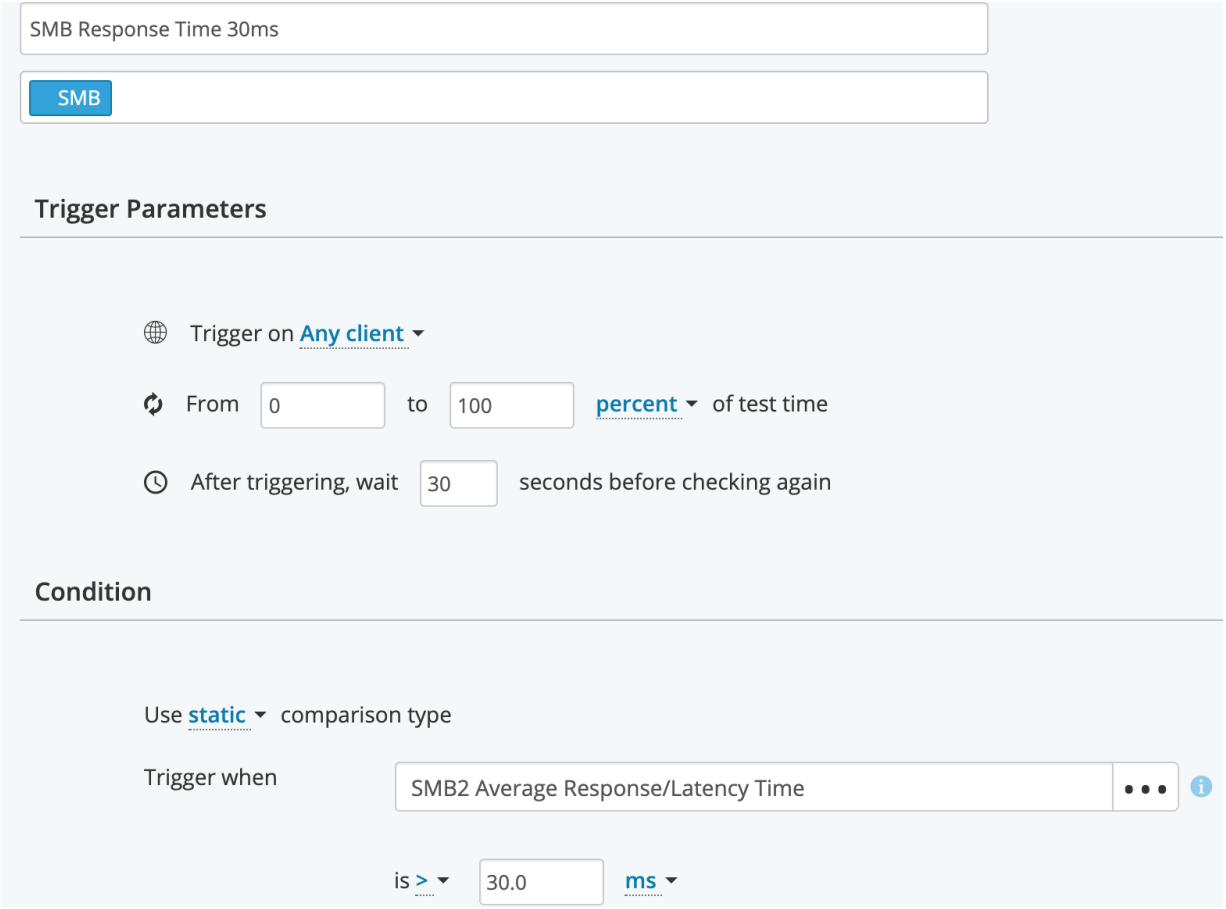
Condition

Specify the statistics that will be compared when this condition is applied.

Two types of Statistics Comparison are supported:

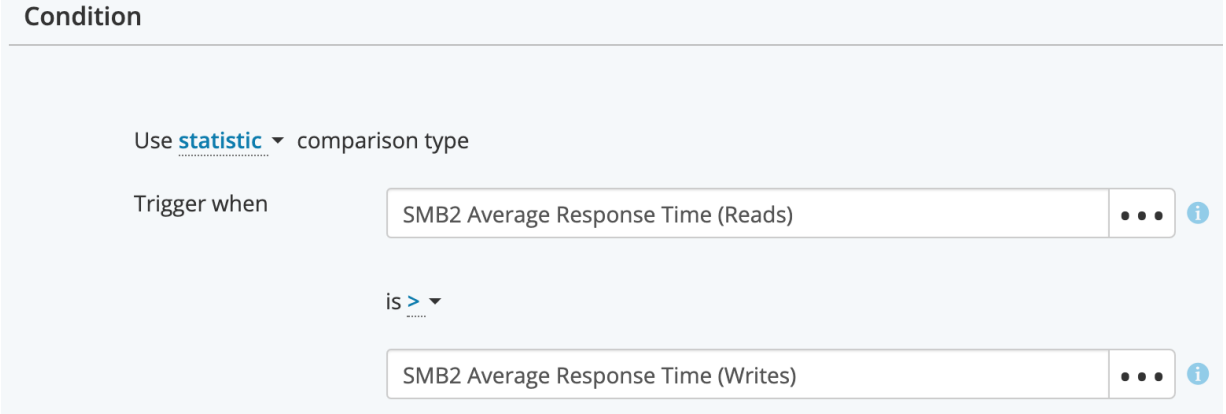
- *Static*. Compares a selected statistic to a statically defined value. Use this option if you have a fixed "red line" that is triggered when the selected statistic crosses the red line.

For example, if you want to confirm that the average response time for SMB2 READ is greater than 30ms, give it a meaningful name and then specify the following:



- *Statistic*. Compares a selected statistic to another selected statistic. Use this option if you have a moving "red line" that is relative to another statistic that is triggered when the selected statistic crosses the red line.

For example, if you want to know when Average Response Time for SMB2 READ is greater than the Average Response Time for SMB2 WRITE, specify the following:



Match Text in Log

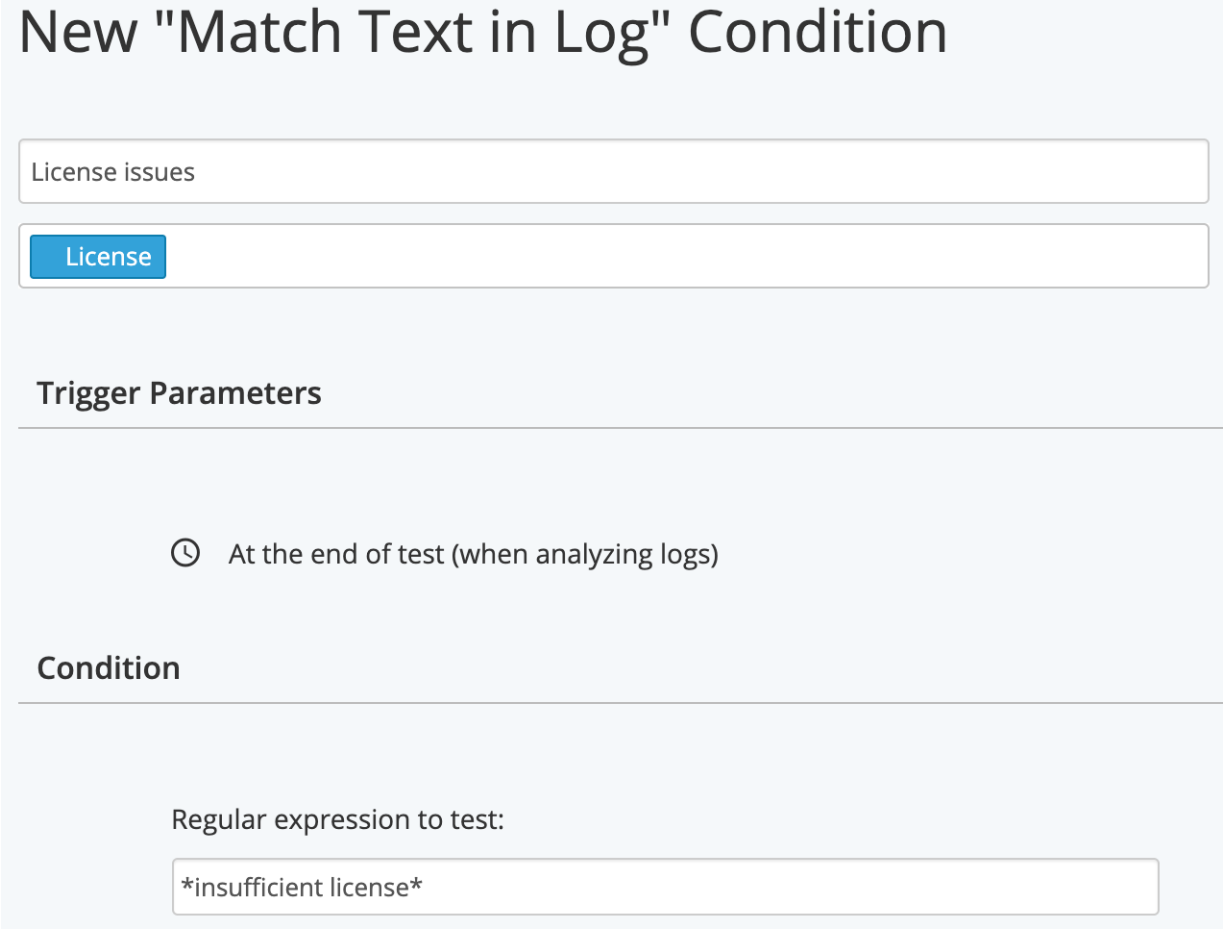
This condition type is for advanced users who are very familiar with the test run log file that is produced by Workload Generators.

Trigger Parameters

The test run log is produced once only after the test run completes. No option is available for trigger parameters when you define a match text in log condition.

Condition

Specify a text string match expression using regular expressions. For example, if you want a condition to be triggered when the log reports “insufficient license”, then specify the following:



There are many websites and articles available on the internet today that provide tutorials and guides to writing regular expressions. For help on regular expressions it's recommended that you search for the term regular expression tutorial.

Combination

Trigger Parameter

See Conditions & Actions [202] for information regarding the statistics comparison condition.

Condition

Use the combination condition to take two or more existing conditions, and join them together using the three logic operations: AND, OR, XOR.

The condition is triggered if the output of the logic operation is True.

Script

Trigger Parameters

See Conditions & Actions [202] for information regarding the statistics comparison condition.

Condition

Use the Script Condition to enter a custom JavaScript code, and the Condition is triggered if the output is True.

For example:

This JavaScript fragment monitors the statistic load.scenarios.attempted.value > 0 is True or False.

```
(function() {

if (load.scenarios.attempted.value > 0)

{a = true;}

return a;

})()
```

Actions

Actions define what happens to a test when an applied condition is triggered. Actions are pre-defined. You cannot add, delete or modify them.

- *Stop Test or Suite*. Stops a Workload test or suite.
- *Pass Test or Suite With Errors*. Sets the test state for a workload test or suite to `passed_with_errors`.
- *Pass Test or Suite*. Sets the test state for a workload test or suite to `passed`.
- *Fail Test or Suite*. Sets the test state for a workload test or suite to `failed`.
- *Pass Suite with Errors*. Suite continues and sets state to passed_with_errors.
- *Pass Test with Errors*. Test continues and sets state to passed_with_errors.
- *Pass Suite*. Suite continues and sets state to `passed`.
- *Pass Test*. Test continues and sets state to `passed`.
- *Fail Suite*. Suite stops and sets state to `failed`.
- *Fail Test*. Test stops and sets state to `failed`.

- *Log*. Logs the result.

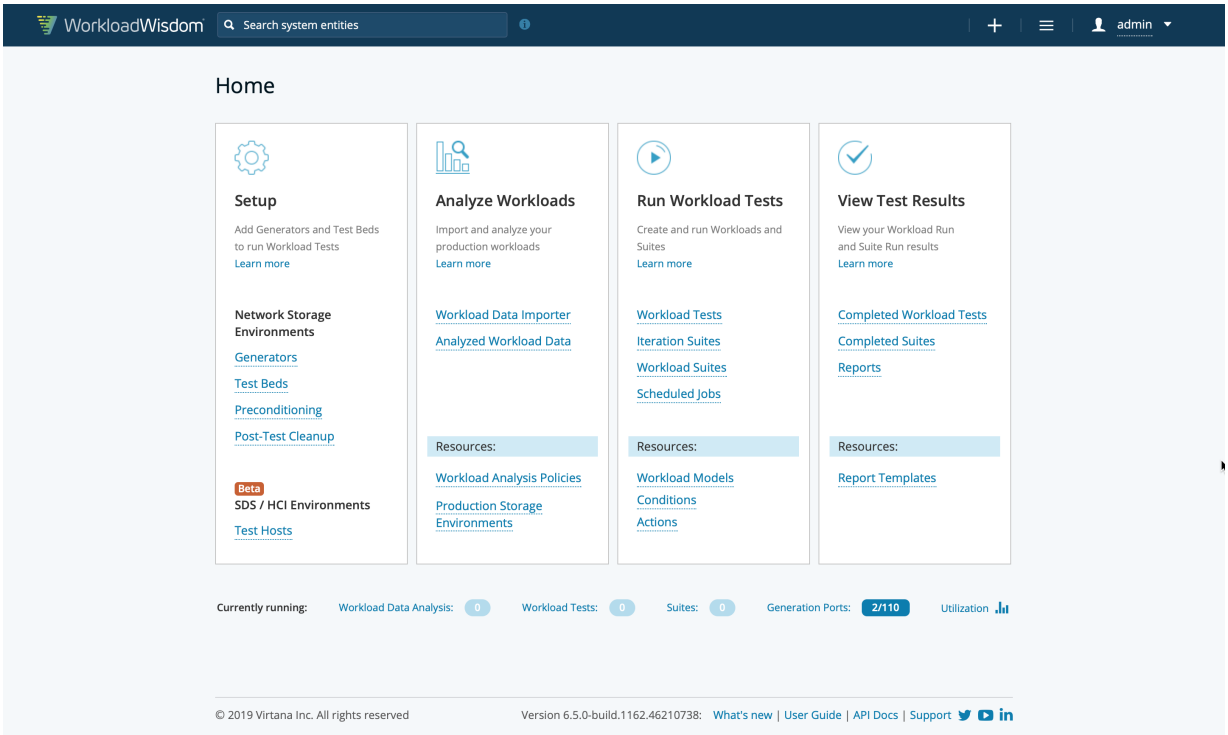
Chapter 6

View Test Results

- **Completed Workload Tests**: View completed Workload Tests.
- **Completed Suites**: View completed Workload Suites and Iteration Suites.
- **Reports**: View and create Reports

Completed Workload Tests and Completed Suites

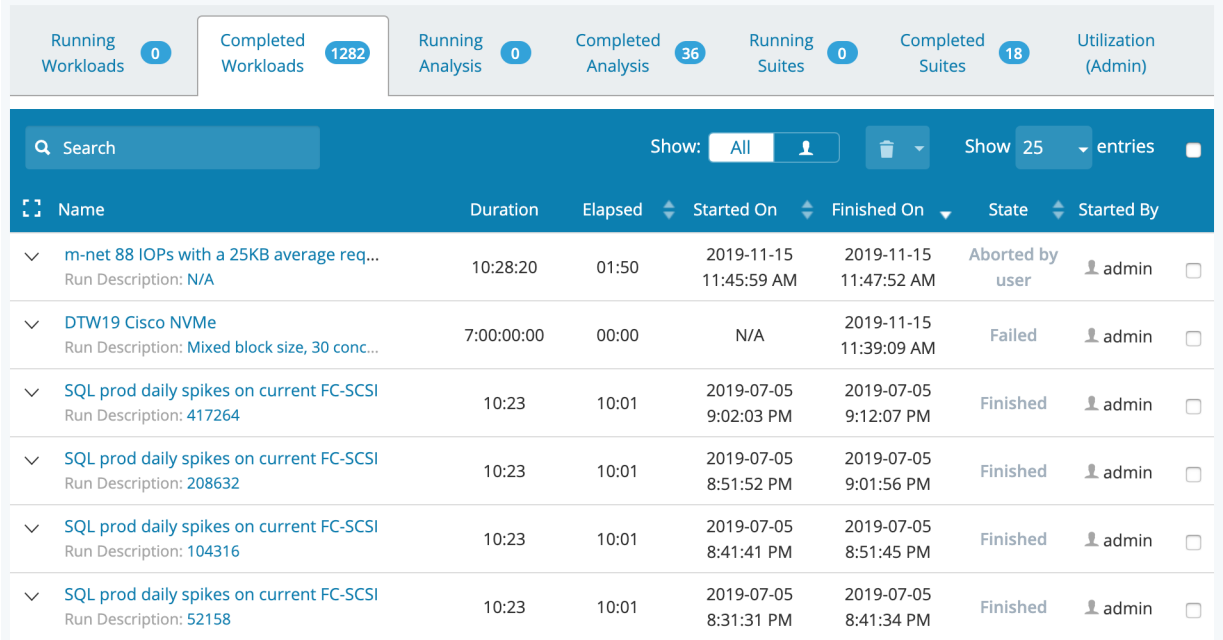
Once you have set up a workload by configuring and starting a test, individually or as part of a suite, you can analyze that test run by navigating to its associated test result. When the workload test finishes, the test result appears on the *Completed Workloads* tab of the *Completed/Running* page.



Click the *Completed Workload Tests* link in the *View Test Results* section of the *Home* page to display a detailed view of all test results produced on your system. These results are displayed on the *Completed Workloads* tab of the *Completed/Running* page.

Click the *Completed Suites* link in the *View Test Results* section of the *Home* page to display the results for completed suites. These results are displayed on the *Completed Suites* tab of the *Completed/Running* page.

The following display is an example completed workloads test display. This display was accessed by clicking the *Completed Workload Tests* link on the *Home* page.



At the top of the page, and depending on what type of user you are, there are seven tabs that you can access.

The number associated with each tab refers to the number of entries associated with the tab. In the *Running Workloads*, *Running Analysis*, or *Running Suites* tabs, the number refers to the number of entities that are presently running on the system. In the *Completed Workloads*, *Completed Analysis*, and *Completed Suites* tabs, the number refers to the number of entities that have ever completed on the system.

- *Running Workloads*. Displays a list of currently running workload tests. A currently running workload test is one that has entered the starting state and has not yet completed. The results displayed regardless of whether the workload test was started manually as a single test run, or automatically as part of a suite.
- *Completed Workloads*. Displays a list of completed workload tests. The `Finished` state indicates the workload test ran to its expected completion. The `Aborted by User` state indicates that the workload test stopped prematurely, either by the user or due to errors.
- *Running Analysis*. Displays a list of currently running analysis of workload data files that were imported for analysis using the Workload Data Importer (WDI).
- *Completed Analysis*. Displays a list of completed analyses of workload data files.
- *Running Suites*. Displays a list of currently running iteration and workload suites.
- *Completed Suites*. Displays a list of completed iteration suites and workload suites. In each of these suites, all workload tests have completed.
- *Utilization (Admin)*. This option is only displayed and available if you are logged as a WorkloadWisdom administrator.

Displays a summary of *Port Utilization*, *Total Number of Tests*, *Total Test Duration*, and *Total Port Hours* for the past year, month, and week. This information tells administrators how much WorkloadWisdom is being used.

In the Port Utilization row, an entry of 0.0 indicates that no one is using WorkloadWisdom. A number of 100 indicates that WorkloadWisdom is being used continuously, and indicates effective, maximum use of the product.

There are some additional parameters available on the *Completed/Running* page to better qualify your results:

Use the drop-down menu to define the number of results entries to display. You can choose to display 10, 25, 50, or 100 entries at a time. The default is 25 entries.

Use the **Collapse All** or **Expand All** buttons to control the visibility of additional details for each row. These details include fields such as *Description*, *Tags*, *Test Bed*, and others, depending on the type of entity (*Test Result*, *WorkloadData Import*, *Suite*), that the row describes. You can edit some of these fields, for example, *Description* and *Tags*, by clicking the *Name* of the entity that you would like to drill down into, and then making changes on the displayed page.

Use the **Show by User** filter to show entries started by All Users or yourself.

Use the **Delete** button to delete the selected result entries. If you select multiple result entries simultaneously, the **Delete Selected** button activates. You can click the **Delete Selected** button to delete multiple result entries at once. The **Delete Selected** function is only available for *Completed Workloads*, *Completed Suites*, and *Completed Analysis*. If there in an associated trace file, the drop-down next to the **Delete** button activates and displays **Delete Trace Files**. Use **Delete Trace Files** to delete the trace files associated with the selected entries.

Use the *Search* box to search for a result entry by matching your search terms to the name of the *Workload Test* or *Workload Suite*. This search only searches the *Description*, *Name*, and *Started Date* fields. A result entry is considered a match if all your search terms are found in the name, regardless of ordering or capitalization. For example, if you entered “vdi workload”, then a result entry with the name “My VDI Workload” and a Result entry with the name “My Workload simulating VDI” would both appear.

Each results entry displays on a (multi-line) row in the output table. Use the checkbox to the left of each row to select the entry.

Descriptions of the column headings associated with each Results entry are:

- *Info*. Controls the visibility of additional details for each row. These additional details include fields such as *Description*, *Tags*, *Test Bed*, and others, depending on the type of

entity (*Test Result*, *Workload Data Import*, *Suite*), that the row describes. You can edit some of these fields, for example, *Description* and *Tags*, by clicking the *Name* of the entity you would like to drill down into, and making changes on the displayed page
- *Name*. Displays the name and description of the *Workload Test* or *Suite*.
- *Duration*. Displays the configured run duration. In the case of a suite, the duration is calculated by adding up the configured run duration of each workload test inside the suite.
- *Elapsed*. Displays the elapsed runtime of the workload test or suite. There are some common cases when the elapsed run time displayed in this column is much shorter than the configured duration run displayed in the *Duration* column. One common case is if you abort the workload test or suite. This carries the state *Aborted by User* in the *State* column. Another common case is when the workload completed, what was configured to do in less time than the configured *Duration* time. When preconditioning a very large array, for example, you might set *Duration* to several days to be sure there is ample time for preconditioning and that the actual preconditioning completed in the configured duration time.
- *Started On*. Timestamp of when the workload test or suite entered the starting state.
- *Finished On*. Timestamp of when the workload test or suite entered the finished state. This is the column that is sorted by default, from most recent to least.
- *State*. Current state of the workload test or suite. Values are: Finished, Aborted by User, Skipped, and Failed. Finished means that the workload test ran to its expected completion. Aborted by User means that the workload test was stopped prematurely either by a user or because do error. Skipped means that an added test encountered problems and was not run. Failed means that an error occurred.
- *Started By*. Username of the person who started the workload test or suite.

Reports

Reports are created by applying a report template to a set of workload test results.

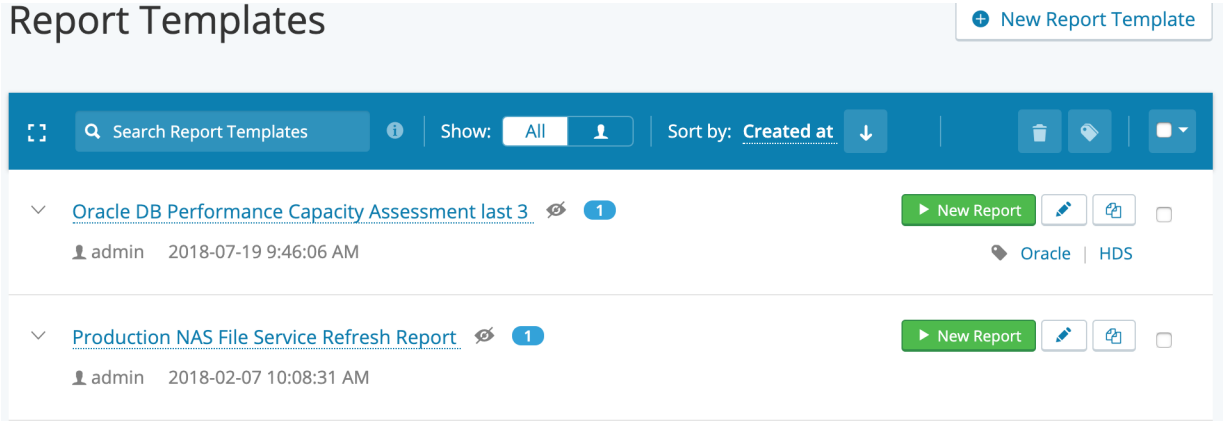
For information regarding how to create a report template, see Report Templates [243].

Create Reports

Once the report template has been created, you need to apply the report template to create a new report. For information regarding how to create a report template, and the different types of report types, see Report Templates [243].

1. Click *Report Templates* in the *View Test Results* section of the *Home* page.

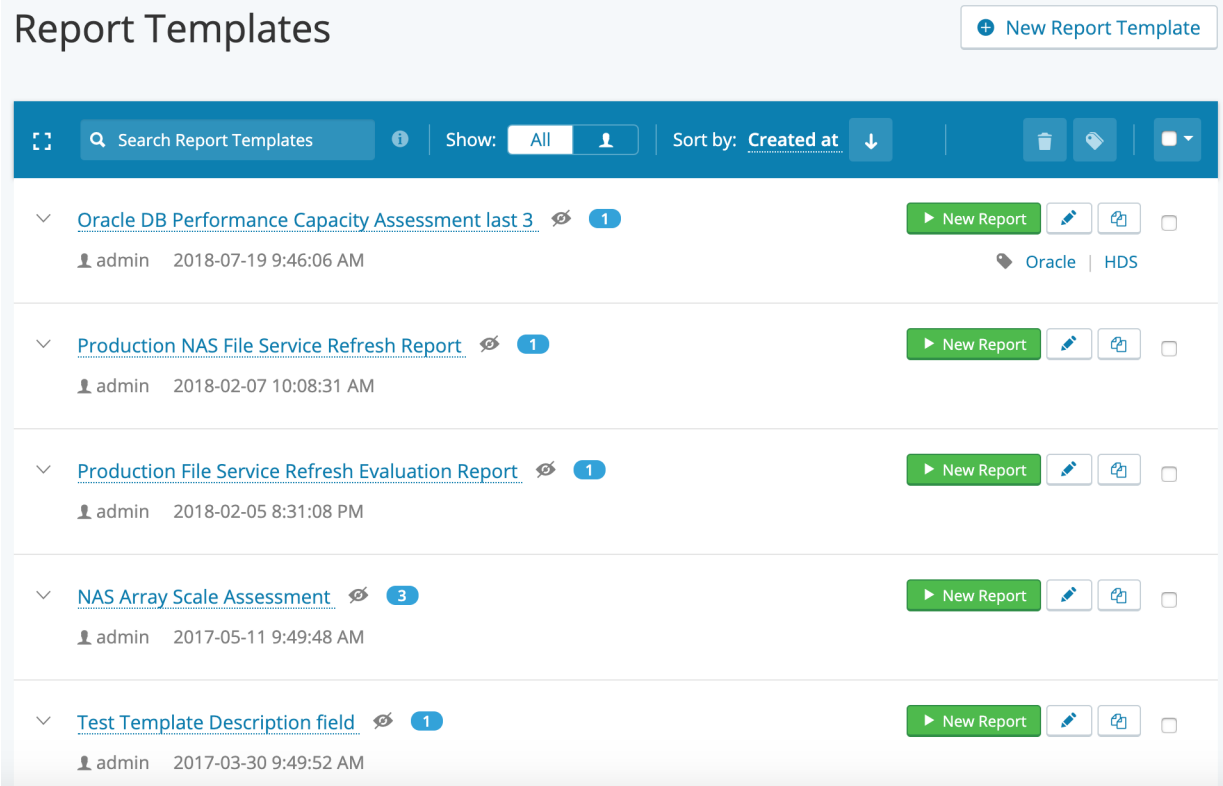
The *Report Templates* page displays.



1. Locate the report template that you want to use.
2. Click **New Report**, next to the report template that you want to use.
3. The *Create Report* page appears. First, select the type of Report you want to create.
   a. **Baseline Reports**
      i. If you are applying a Baseline Report Template, there is nothing else you need to select, since all the required information is defined in the Report Template. Review the report before committing and modify the report name to give it a unique name, and to also make it easier to identify this specific report.
   b. **Multi-Statistic Reports**
      i. If you are applying a multi-statistic report template, select the specific test run or runs to which to apply the report template.
      ii. Review the report before committing and modify the report name to give it a unique name which is easier to identify.
4. Click **Create Report** to complete the process.

Resources for *View Test Results*

You can also create a report by applying a report template to one or more completed workload tests, including those which were run as part of an iteration or workload suite. You can use a report template to define the types of charts and the statistics to be included the report.



Reports create visual and insightful reports from your test runs. Use reports to take one or more supported statistics from your test runs and plot them on charts.

You can create reports that compare how:

- Different storage arrays perform using the same workload
- Storage infrastructure handles a workload as the number of IOPS increases
- Storage infrastructure handles a workload as the block size changes
- Throughput, IOPS and latency trend over time for a workload
- Latency changes as throughput increases

This list displays some of the reports types that you can create using the WorkloadWisdom.

Report Templates

Follow these steps to create a report:

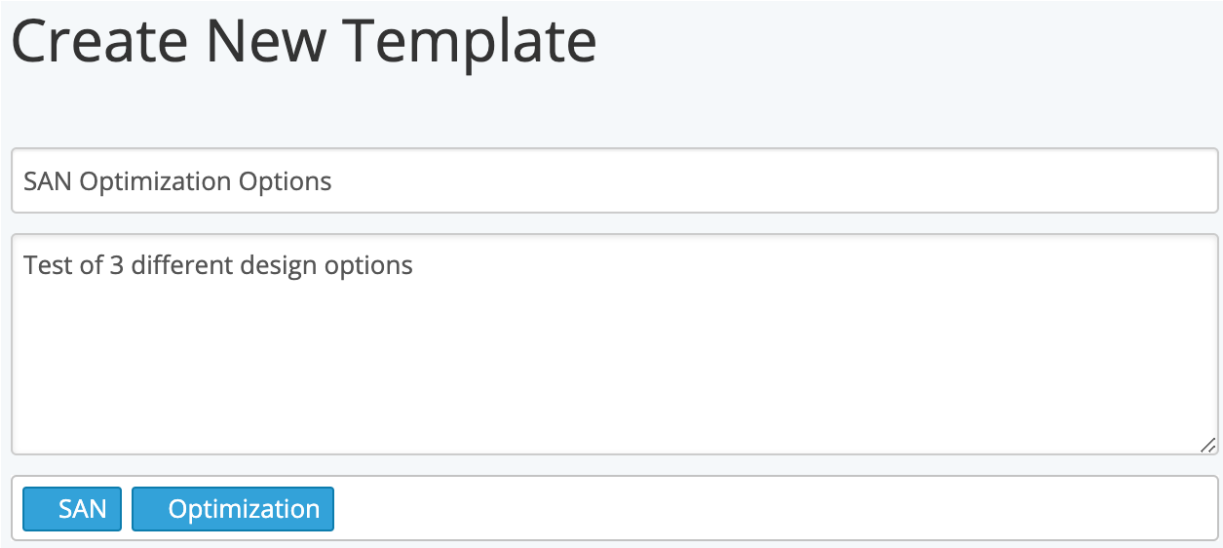
1. Create a report template.
2. Apply the report template to your test results.

Create a New Report Template

Assign a name to the report template that allows you to easily identify and search for the report template. This name appears in the list of report templates to which you have access.

Optionally, add a description to the report template.

Optionally, add one or more tags to describe this report template. Tags are searchable in WorkloadWisdom.



Select Report Type

Define the type of charts to be built by the report template. Baseline and multi-statistic charts are supported.

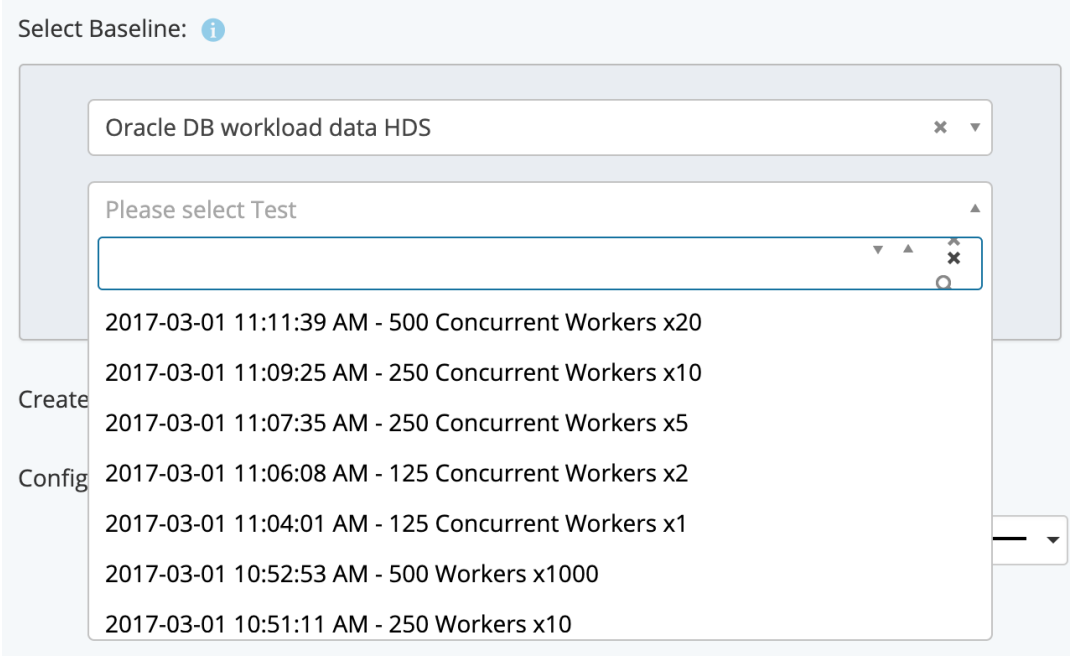
Baseline Chart

Use baseline charts if you want to create a report template that compares the same statistic from different test runs, where one of the test runs is the baseline.

For example, if you are creating a report that compares how 5 different firmware versions perform when handling the same workload, and you want the performance obtained with the current firmware version to be the baseline, you might use a baseline chart. Create a baseline chart and identify the test run performed on the current firmware version to be the baseline result.

Select Baseline

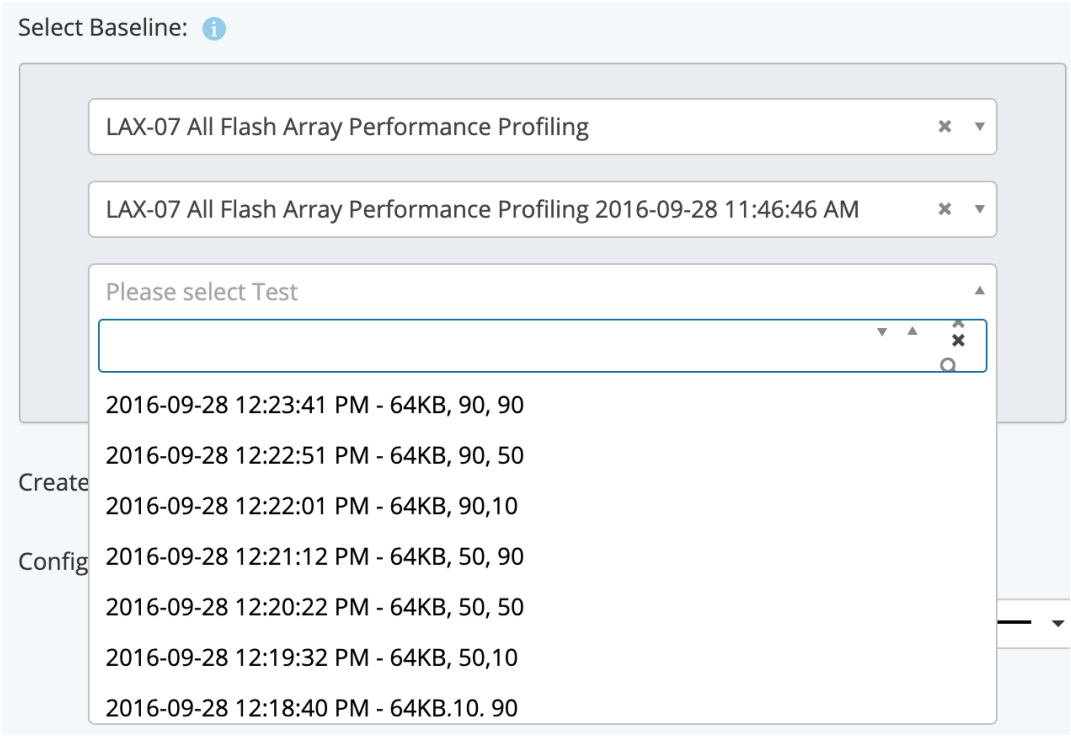
1. Select a Workload Test or Suite from the drop-down
   a. If selecting a Workload Test
      i. Once you select a Workload Test, all available test runs associated with the selected workload that meet the filter criteria and that were not started by a Suite are made available.



> **NOTE**
> Prior to WorkloadWisdom 6.2, if you ran Workload Test X ad-hoc 3 times, and that Workload Test X was also used in an Iteration Suite Y, and the Iteration Suite Y ran 1,000 iterations of Workload Test X, then you would see all 1,003 Workload Test Runs when you select Workload Test X here. This made it difficult to differentiate and find the 3 runs of Workload Test X from the other 1,000 runs. Starting with 6.2, the 3 ad-hoc runs of Workload Test X will appear when you select a Workload Test, and the 1,000 runs of Workload Test X that were started from Iteration Suite Y can be found by selecting a run from Iteration Suite Y, making it easier to find both the 3 runs and the 1,000 runs.

   b. If selecting a Workload Suite or Iteration Suite

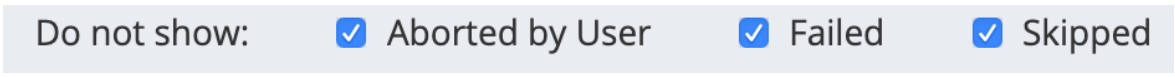
i. Once you select a Workload Suite or an Iteration Suite, an additional drop-down is available for you to select the Suite Run, in which each Suite Run contains one or more Workload Test Runs that you can select in the next step.



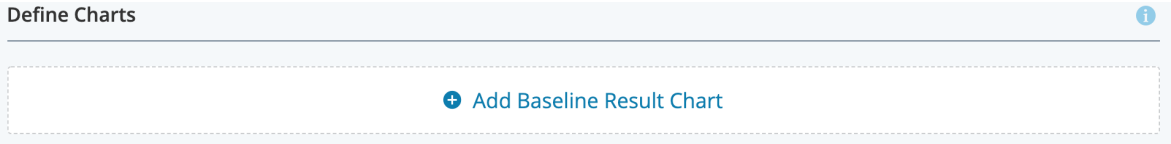
CAUTION

If the number of Workload Tests and Suites is very large (over 1,000), it may take 1 to 2 minutes to load the list in the drop-down.

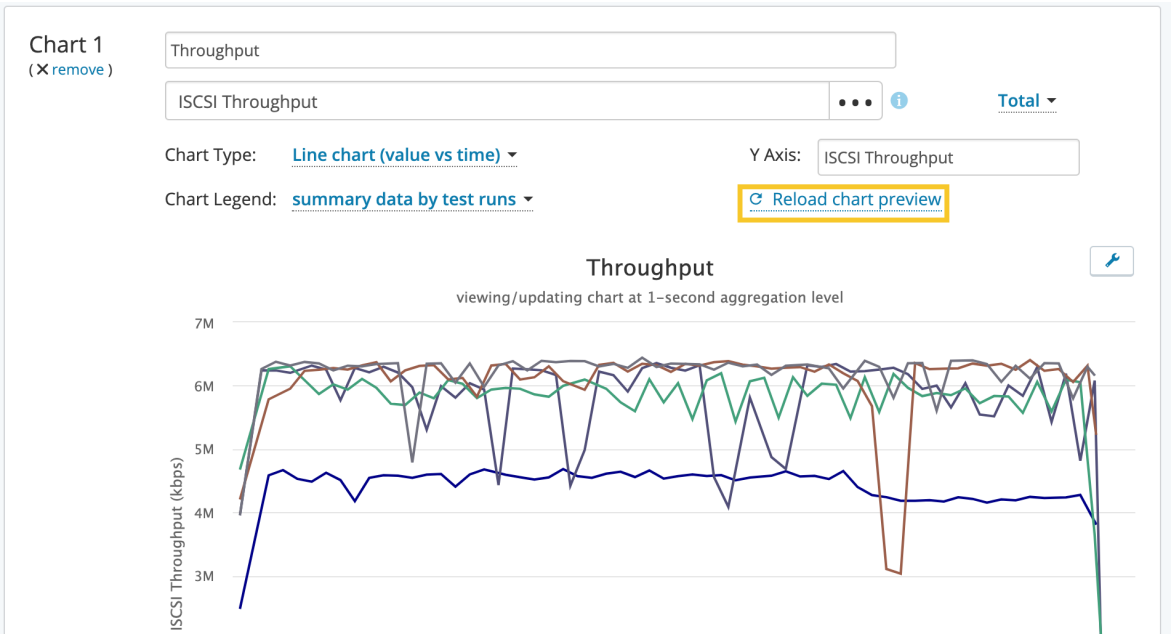
2. Select the Run from the selected Workload Test / Suite Run that you want to be the baseline result.
3. Select the Run from the selected Workload Test / Suite Run that you want to be the baseline result. The Runs are identified by their system generated timestamps in the format YYYY-MM-DD HH:MM:SS AM/PM, followed by their user entered Run Description if you have entered any. In the example below, “2018-10-05 11:34:52 AM” is generated by the system, and “Storage firmware A” is manually entered by the user when viewing the Workload Test Run’s Results Dashboard. By default, Runs that have an Aborted, Failed, or Skipped status are filtered out as these are usually not used for reporting. However, if you have a need to include one of these Runs in the report, you can deselect one or more of these by using the **Do not show** checkboxes.



4. Define the test runs from which you want the report template to collect statistics. Three options are available:
   a. *Most recent test run*. Select “last test run” to collect statistics from the very last test run of the workload when applying this report template.
   b. *Most recent N test runs*. Select “last” to define the number most recent test runs of the workload when applying this report template.
   c. *Test runs from recent days*. Select “test runs during last” to define the maximum number of test runs of the workload from the past days when applying this report template.
5. Define the specific baseline charts to include in the report template.



6. For each baseline chart, open the statistics selection dialog by clicking on the ... button in the screenshot below and select one statistic (for example, NFSv4.1 Commands Succeeded/sec) at a time. This is the statistic that is collected from the applicable test runs to populate this chart when the report template is applied. Optionally configure other settings on the chart. To preview the resulting chart when you create a Report out of this Report Template, click **Reload chart preview**.



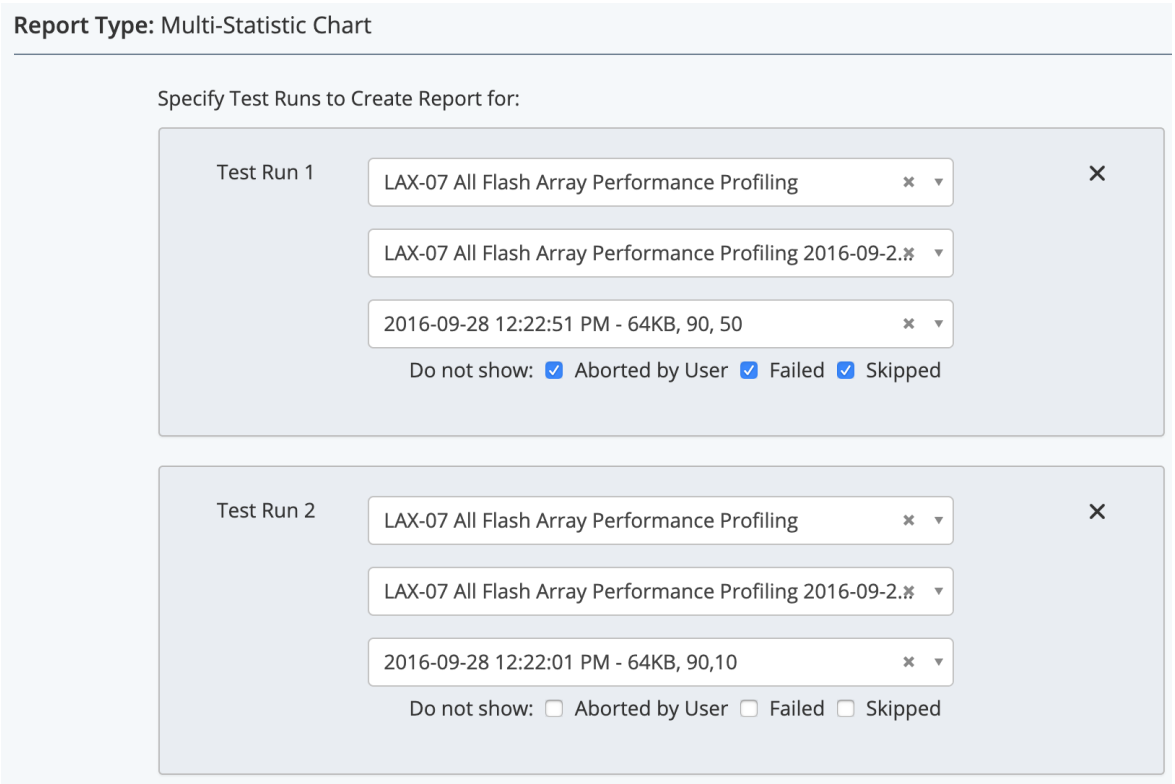
Multi-Statistic Chart

Use multi-statistic charts if you want to create a report template that plots different statistics on the same chart.

For example, suppose that you want to plot throughput, IOPS, and latency on the same time-series chart, to enable you to ascertain how these statistics relate and correlate to each other over the course of a workload. Create a multi-statistic chart, and select the three statistics to be plotted on the same chart.

Specify Test Runs for the Report

1. Optional: Select a Workload Test or Suite from the drop-down that will always be included in the Report when you create it. Follow the same instructions from the Baseline Report section to select a Workload Test or Workload Suite or Iteration Suite. The only difference is that you can have multiple Test Runs.
2. Optionally, select one or more test runs by following the step above for Test Run 2 in the Report Template.

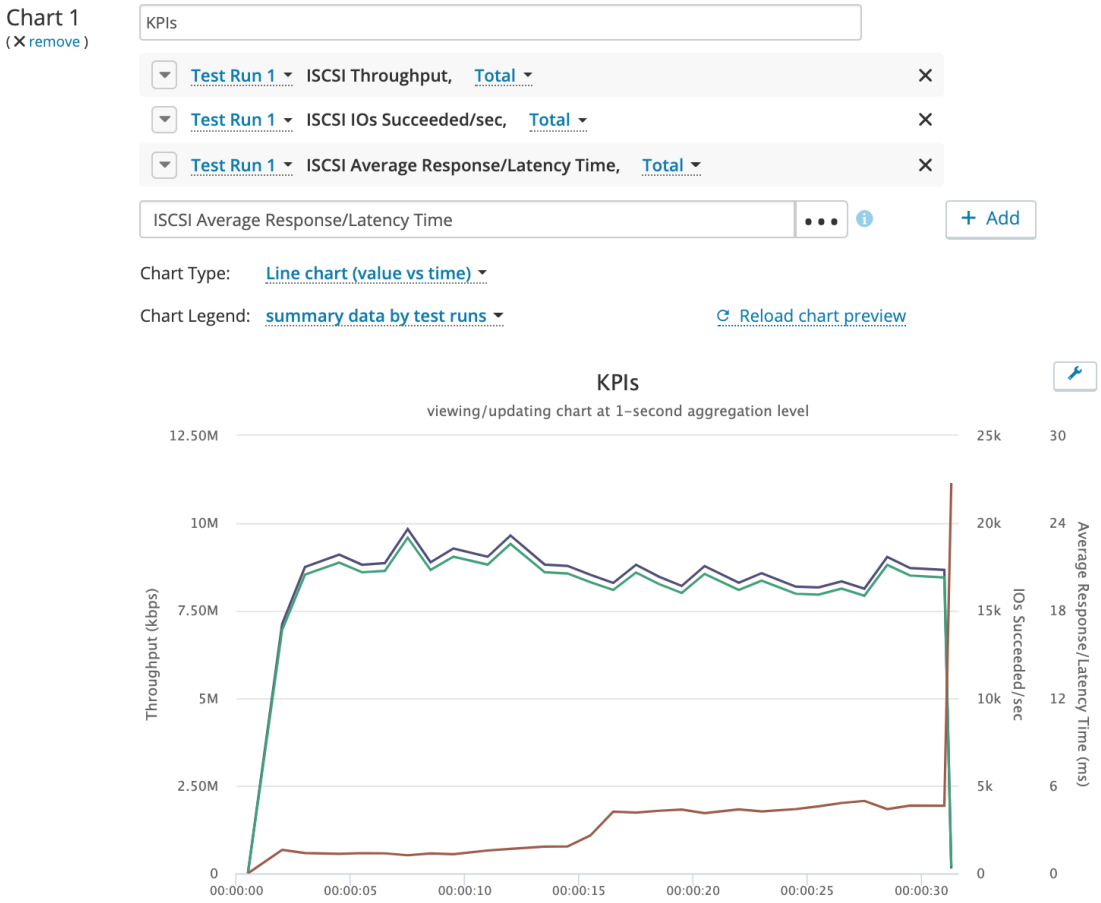


NOTE

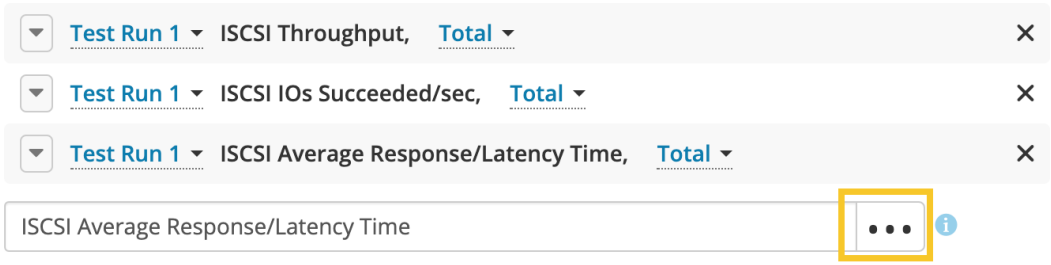
You can select the same Workload Test / Suite or a different one. There is no requirement in Multi-Statistic Charts to use Runs from the same Workload Test or Suite Run as required by Baseline Charts. However, in most cases, you want to use at least the same protocol.

3. Define the specific multi-statistic charts to include in the report template.

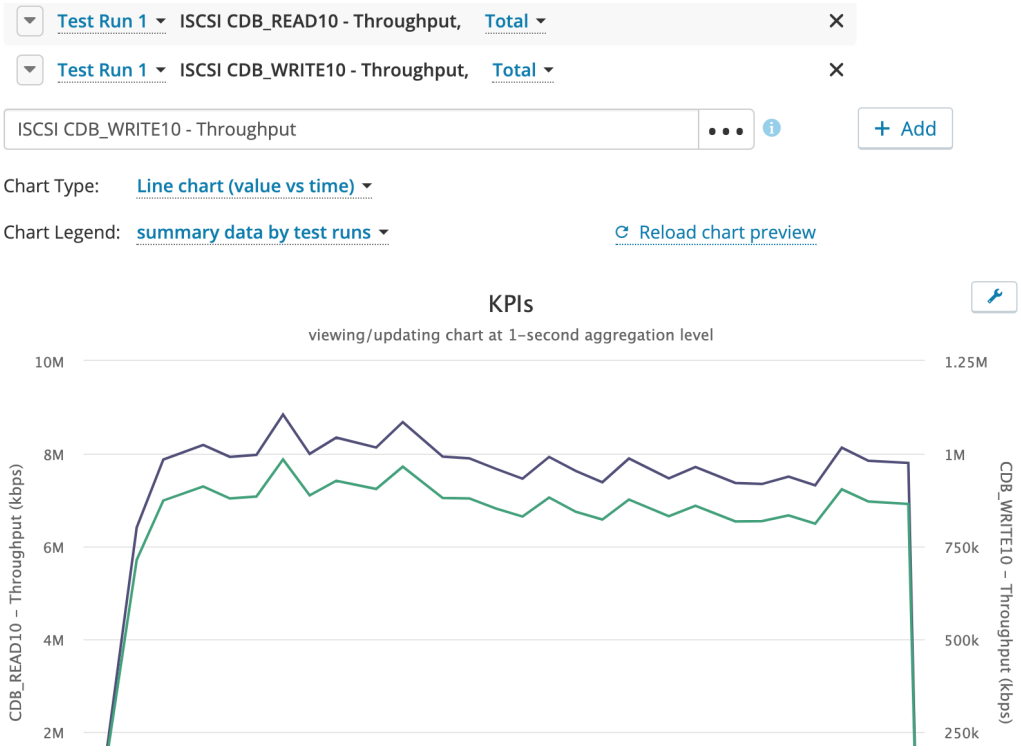
a. **Custom Chart**: use custom chart to define exactly what and how many statistics to plot on the same chart, as well as the presentation of the statistics. For example, you may want a KPI chart where you plot throughput, IOPS, and response time on the same time graph.



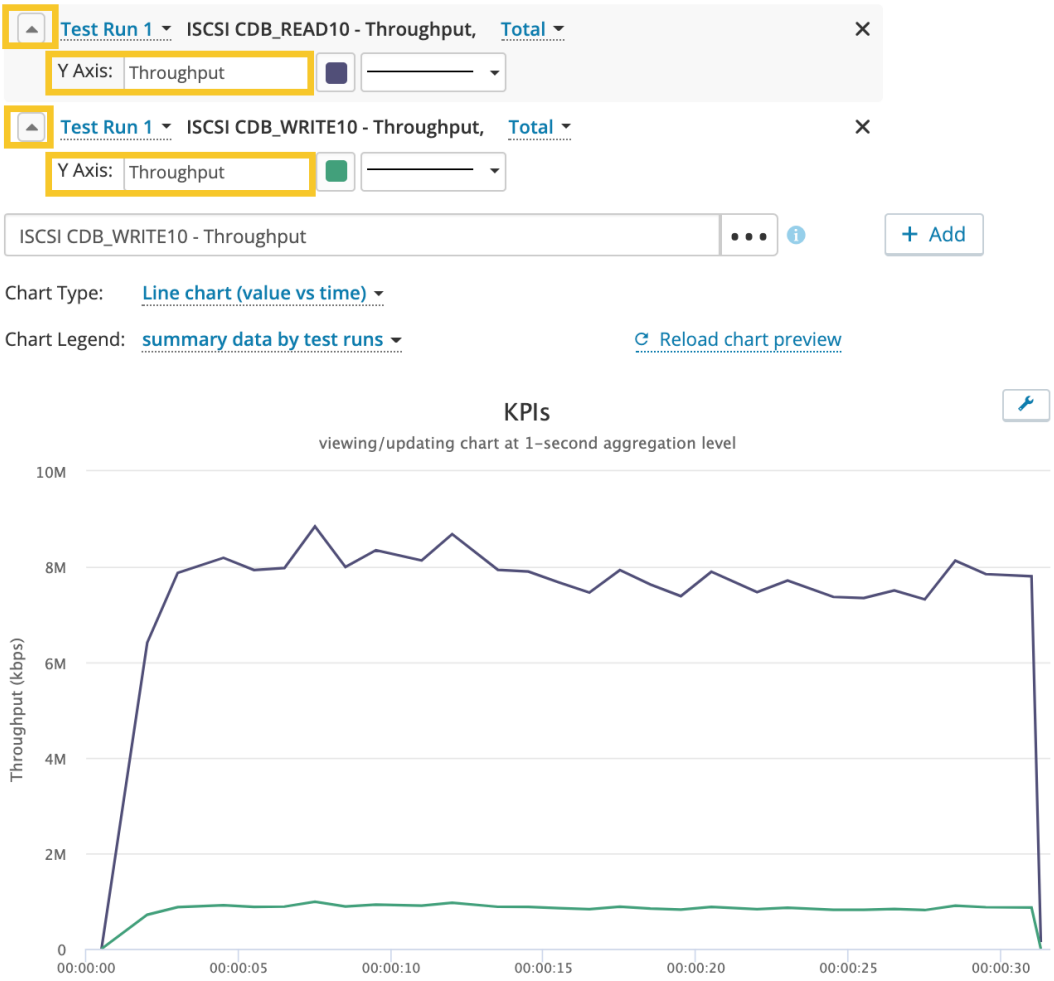
i. For each chart, select one or more statistics to plot on the same chart. The selected statistics are collected from the applicable test runs when you apply the report template.



ii. By default, for each statistic you add to the custom chart, a dedicated Y-axis is generated for that statistic. You might want to have only one Y-axis if multiple statistics share the same unit, for example, read throughput statistic and write throughput statistic.

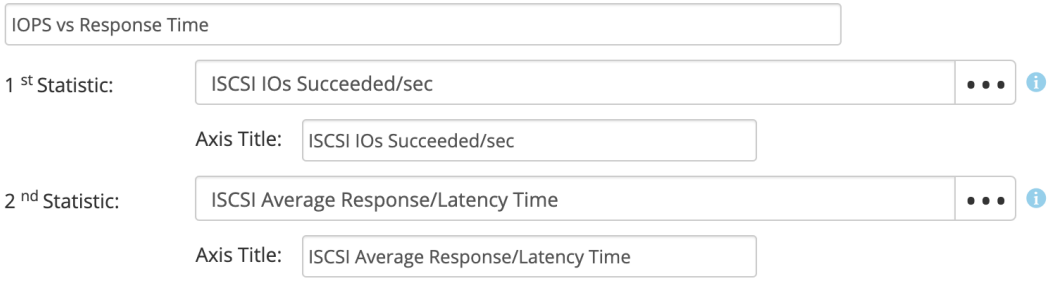


iii. To make that change click the down arrow to open up the Y-axis dialog and enter the same Y-axis name for the statistics you want to plot over the same Y-axis.



b. **Scatter Plot**: Use scatter plot if you want to define which two statistics to plot on a standard scatter plot. For example, you may want to see how sensitive the SUT's Response Time is to a wide range of IOPS.

i. For each chart, select one statistic to be plotted on the x-axis, and one statistic to be plotted on the y-statistic, for each test run.



ii. Click **Reload chart preview**.

Creating a Multi-Statistic Report Template with Different Protocols

If you have different Runs from different protocols that you want to put in the same Report, it is recommended that you apply Test Runs of different protocols to different charts. For example, one Custom Chart for FC for Test Runs 1 and 2, and one Custom Chart for NFSv4.1 for Test Runs 3 and 4. Overall, the workflow is no different between one protocol and multiple protocols.

Chart Presentation Options

Statistics Selection

For each selected statistic, you can define the ports from which to collect the statistics.

The following statistics selections are supported:

- *Total.* Aggregate from all Workload Generator test ports used in the test run. This is the most common statistics selection, as well as the default.
- *Client total.* Aggregate from all Workload Generator client ports used in the test run.
- *Server total.* Aggregate from all Workload Generator server ports used in the test run.
- *Port n.* Where *n* is a specific Workload Generator test port used in the test run.

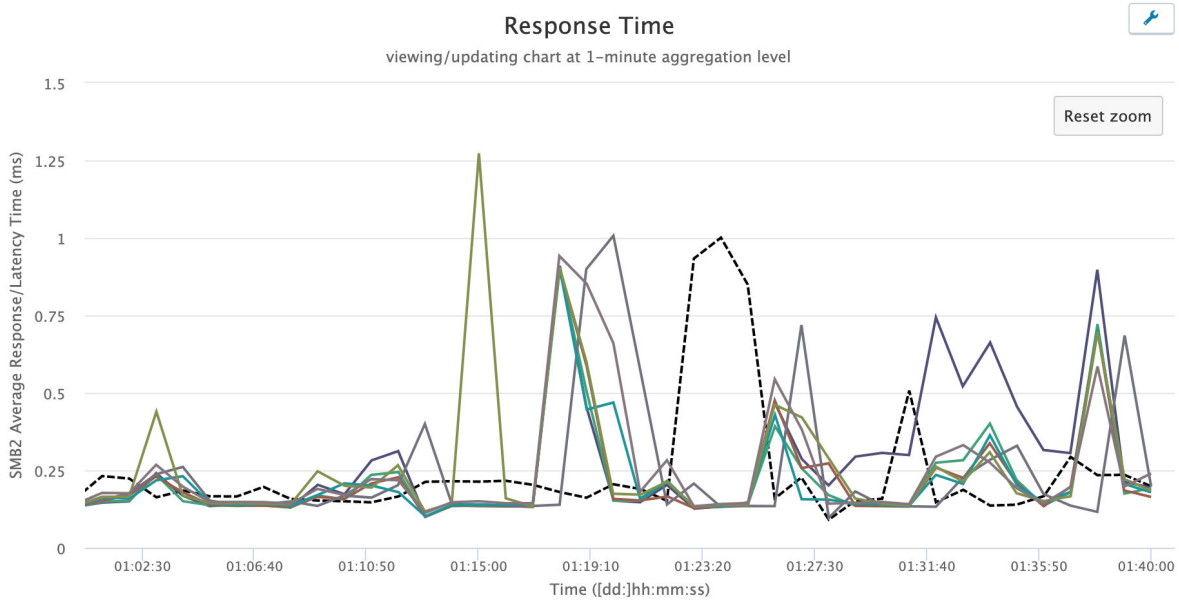
Provide a chart title (for example, Average SCSI Throughput) for the chart in the report template.

Chart Type

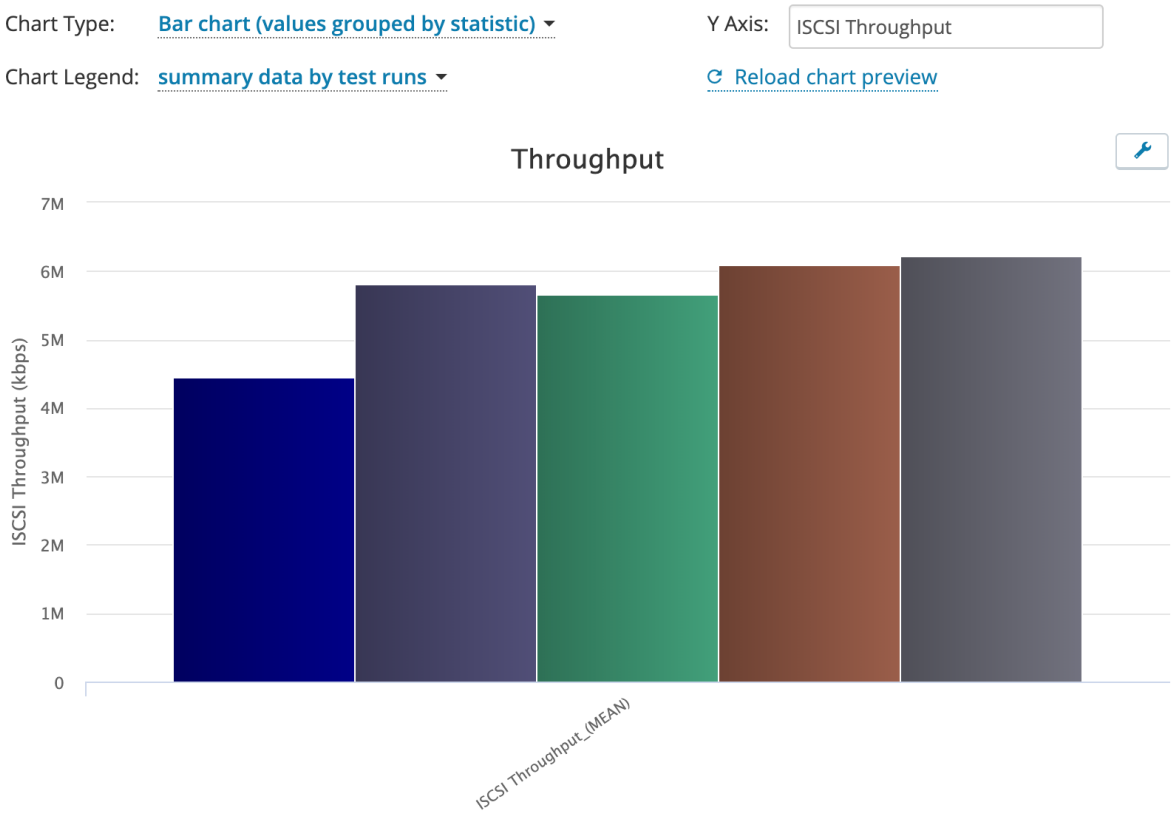
Define the chart type for the chart.

The following chart types are supported:

- *Line chart (value vs time).* Plots the statistic from each test run in a time-series chart.

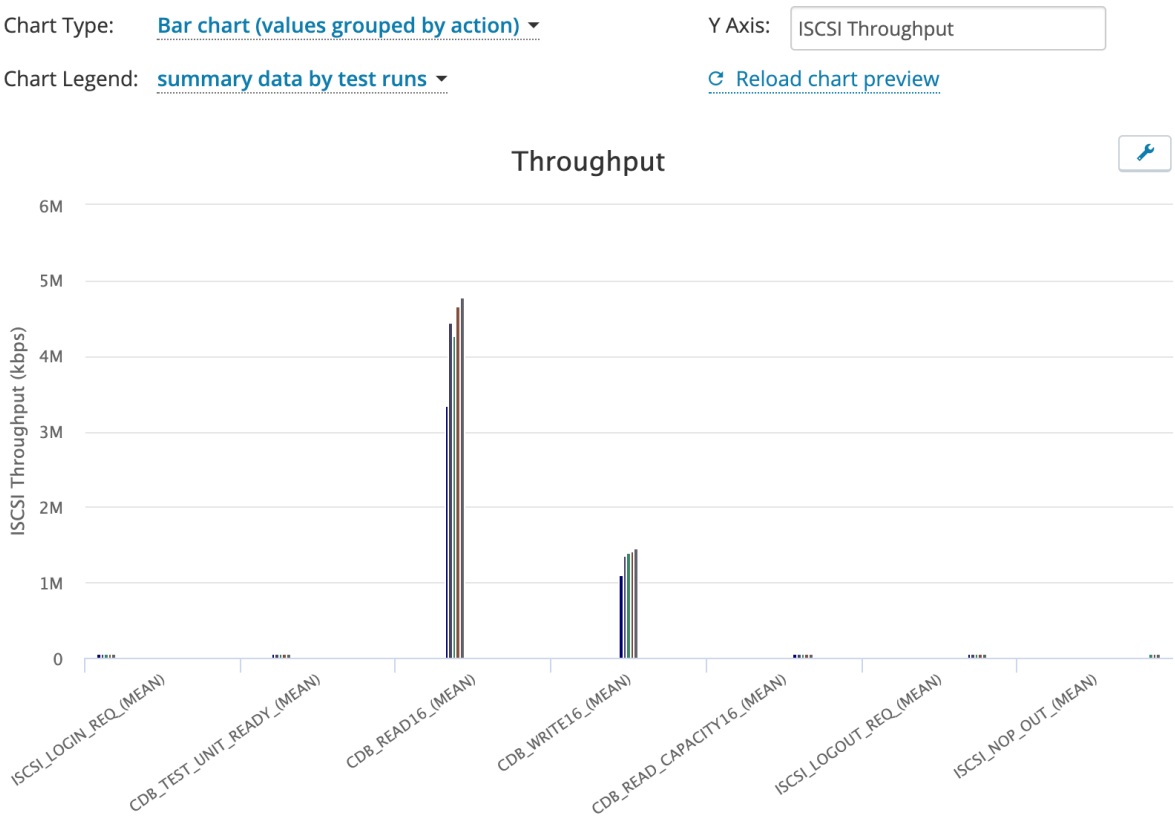


- *Bar chart (values grouped by stat).* Plots the statistic from each test run in a bar chart.

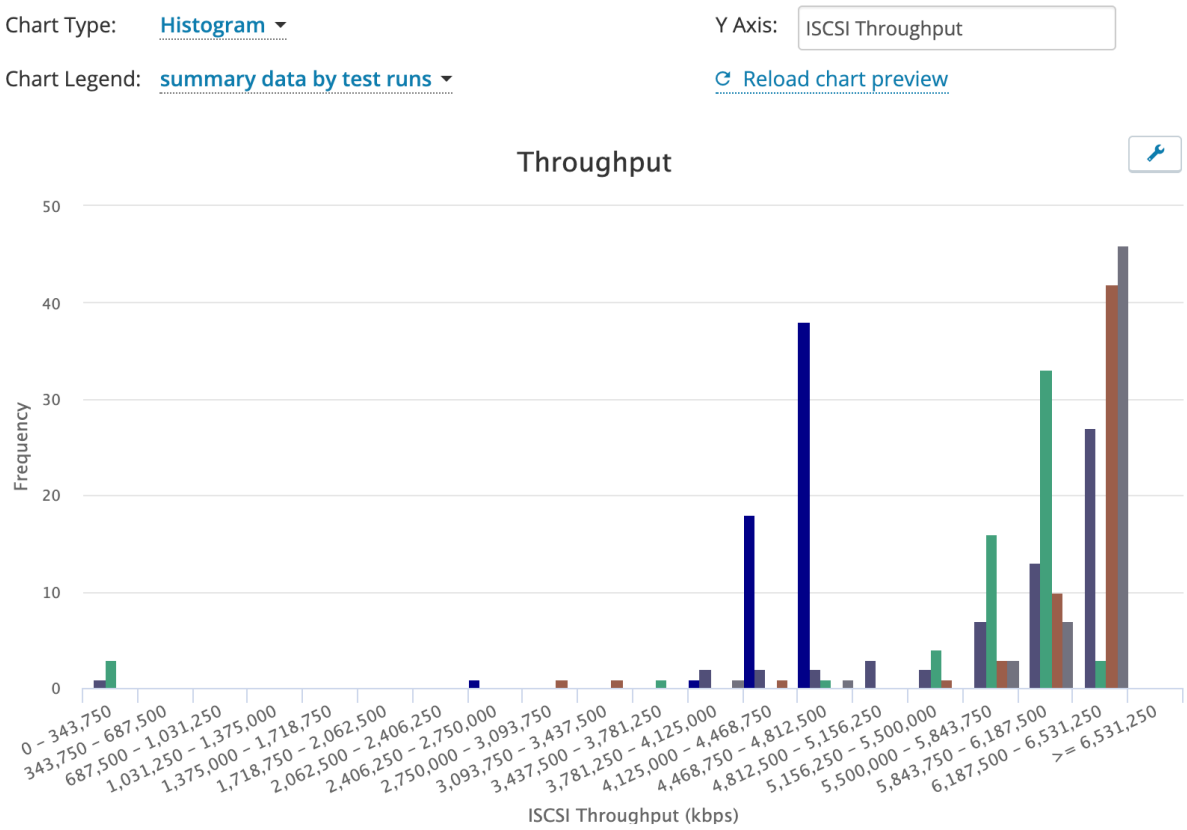


- *Bar chart (values grouped by test run)*. Plots the statistic from each test run in a bar chart.

- *Bar chart (values grouped by action)*. Plots the statistic, broken down to a per command basis if applicable, from each test run in a bar chart.



- *Histogram*: plots the statistic from each test run on a histogram chart



You will find more options under the Wrench Tool menu on each chart, such as changing units, or selecting Max value instead of Mean value (default), and others.

Chart Legend

You can define the content of the legend at the bottom of the charts.

The following three options are available:

- *None*. No legend information displayed.
- *Test runs only*. Display test runs that are applicable to the report template, based on the report type settings.
- *Summary data by test runs*. Display statistics for each test run in tabular format.

Reload Chart Preview

While building the report template, you can click **Reload chart preview**to preview the chart to be created when you formally apply the report template.

Chapter 7

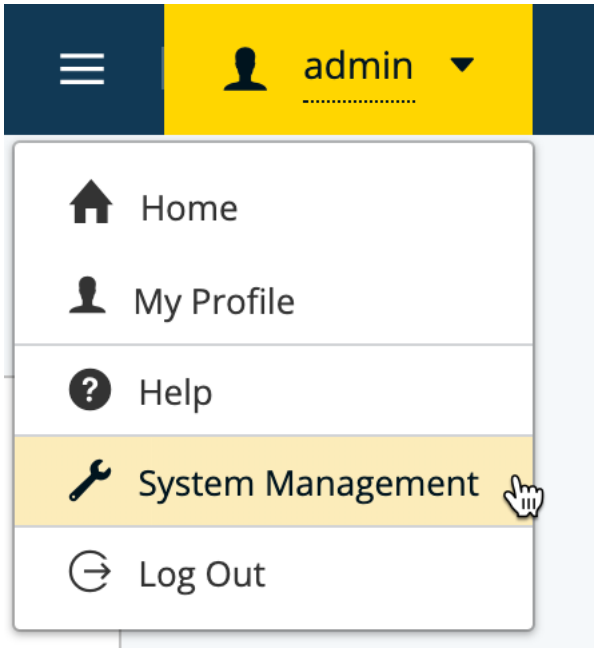
Using System Management

- **WorkloadWisdom System Management**

Understand and use the following system management tools:

- Usage
- CPU Utilization
- Firmware Upgrade
- Upgrade Workloads (BETA)
- Tools
- Backup/Restore
- Authentication
- Users
- User Groups
- Logs
- System Logs

Access the system management tasks by using the drop-down *User* menu at the top right of any page.



WorkloadWisdom System Management

Two types of roles are available: *Administrators* and *Users*. Administrators have full access to WorkloadWisdom and all its objects from all user roles, regardless of the Privacy setting of an object. Users have partial access to System Management, and full access to objects that are not set to Private.

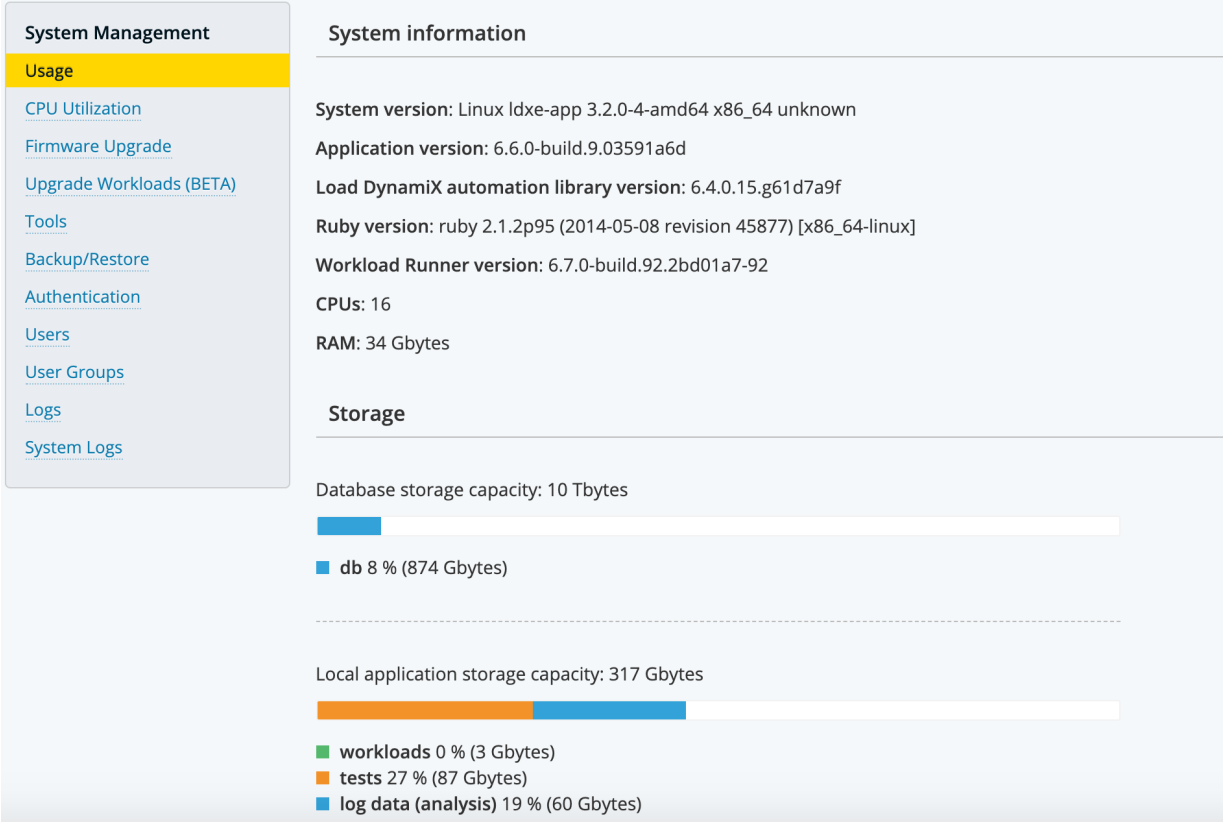
As a WorkloadWisdom Administrator, you have access to many system management tools that apply to the Workload Generator. All Users are affected by changes made in system management. Ensure all Users are notified prior to making changes, as some changes require that WorkloadWisdom to be restarted.

Some of the system management functions are not available to regular Users, such as firmware upgrade. You might be requested by WorkloadWisdom Users to perform certain system management tasks as a WorkloadWisdom administrator.

Usage

Use the *Usage* page to view the following read-only information about WorkloadWisdom. Some of the information is intended to be meaningful only to Virtual Instruments for better supportability.

- System version
- Application version
- Load DynamiX automation library version
- Ruby version
- Workload Runner version
- CPUs
- RAM
- Storage, including database storage and local application storage capacity



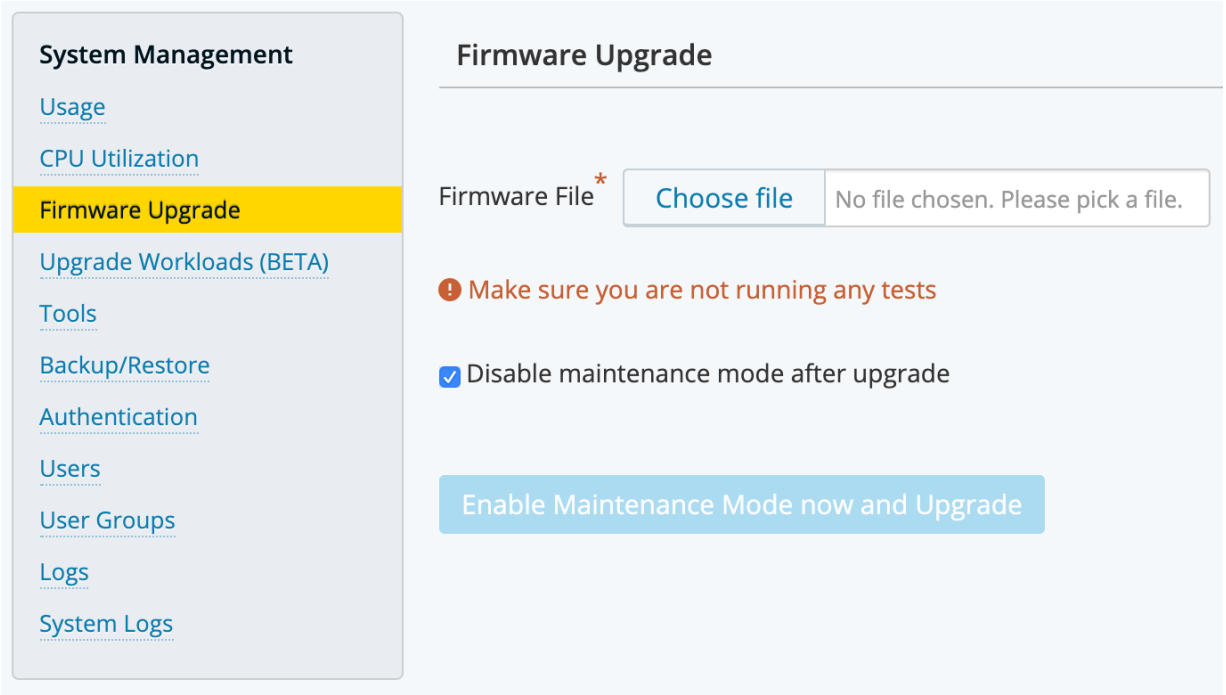
If you are deploying WorkloadWisdom Virtual, ensure that the computing resources (CPUs and RAM) meet the minimum requirements for WorkloadWisdom Virtual. If you cannot find the WorkloadWisdom Virtual documentation, please contact Virtana Support.

CPU Utilization

Use the *CPU Utilization* page to view a read-only view of the current list of PIDs, process names, and the percentage of CPU and memory consumed. While tests are running, it is normal to see certain processes reach 100% CPU% or slightly over.

Firmware Upgrade

Use the *Firmware Upgrade* page to update the Workload Generator version from the current version to the next supported version.



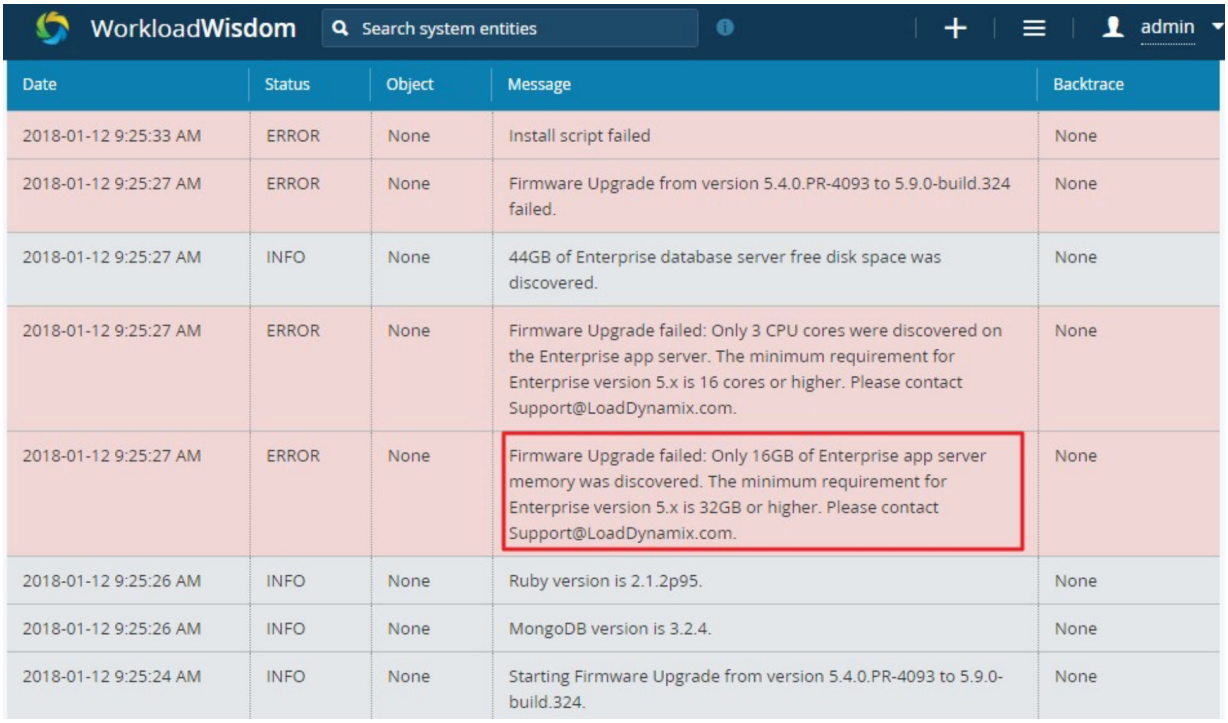
By default, before you perform a firmware upgrade, WorkloadWisdom enables maintenance mode. Upon completion, WorkloadWisdom exits maintenance mode by default.

If you want to keep WorkloadWisdom in maintenance mode after an upgrade, *do not* check the *Disable Maintenance Mode after upgrade* checkbox.

Once you have downloaded the WorkloadWisdom firmware and ready to perform the upgrade, and assuming that you have enabled maintenance mode already, click **Update** to perform the firmware version update. Otherwise, you will click **Enable Maintenance Mode now and Update**.

The *Update* screen displays while WorkloadWisdom is being updated. The *Update* screen displays what is being updated currently, and the estimated time remaining to complete the update.

The system logs contain useful information related to update processes. If you encounter errors during an update, refer to the system logs, sometimes referred to as the WorkloadWisdom logs. See System Logs [271].



Date	Status	Object	Message	Backtrace
2018-01-12 9:25:33 AM	ERROR	None	Install script failed	None
2018-01-12 9:25:27 AM	ERROR	None	Firmware Upgrade from version 5.4.0.PR-4093 to 5.9.0-build.324 failed.	None
2018-01-12 9:25:27 AM	INFO	None	44GB of Enterprise database server free disk space was discovered.	None
2018-01-12 9:25:27 AM	ERROR	None	Firmware Upgrade failed: Only 3 CPU cores were discovered on the Enterprise app server. The minimum requirement for Enterprise version 5.x is 16 cores or higher. Please contact Support@LoadDynamix.com.	None
2018-01-12 9:25:27 AM	ERROR	None	Firmware Upgrade failed: Only 16GB of Enterprise app server memory was discovered. The minimum requirement for Enterprise version 5.x is 32GB or higher. Please contact Support@LoadDynamix.com.	None
2018-01-12 9:25:26 AM	INFO	None	Ruby version is 2.1.2p95.	None
2018-01-12 9:25:26 AM	INFO	None	MongoDB version is 3.2.4.	None
2018-01-12 9:25:24 AM	INFO	None	Starting Firmware Upgrade from version 5.4.0.PR-4093 to 5.9.0-build.324.	None

NOTE

Do not attempt to restart WorkloadWisdom during an update. An update can take many hours, depending on the current version you are running, the size of the database, and in the case of a WorkloadWisdom Virtual deployment, the amount of real computing and networking resources available. It is possible to occasionally see a “web page unavailable” error, or “host unreachable” error, or equivalent errors depending on the web browser you are using. During the update, the WorkloadWisdom web service is shut down periodically. If your update does not complete within one day, contact Virtana Support before attempting to restart WorkloadWisdom.

Upgrade Requirements

WorkloadWisdom supports updating to version X from version X-1, where X is the current GA Release version. WorkloadWisdom does not support updating to version X from version X-2. For example, updating to 3.0 is only supported if your current LDX-E version is 2.6SP1, which is the most recent GA Release version of LDX-E before 3.0 (that is, X-1

version). Updating to 3.0 is not supported if your current LDX-E version is 2.6GA, which is the X-2 GA Release version. To update to 3.0 from 2.6GA, which is two GA Release versions apart, you need to update from 2.6GA to 2.6SP1 first, and then update from 2.6SP1 to 3.0. If you are not sure, contact Virtana Support before you start your update if you aren't sure.

When updating to some releases, the update might require you to perform a system update first, prior to the WorkloadWisdom firmware update. You can request detailed update instructions and download links from Virtana Support.

Upgrade Workloads (BETA)

To upgrade a Versionable Workloads Framework compliant workload model, the Upgrade Workloads functionality allows you to select a template file provided by Virtual Instruments for the workload model, and apply the upgrade for the affected workload system-wide.

Starting with WorkloadWisdom 6.4, a new Versionable Workloads Framework (VWF) was designed to provide better separation of workload content from the underlying platform, which provides two critical capabilities: out-of-band delivery of VWF compatible workload models, and multiple versions of a VWF compatible workload model. For example, if in 6.4 you ran a set of tests using a current workload model in 6.4, and then in 6.6 the workload model was upgraded to include new functions that also altered the workload's behavior, then if it was a VWF compatible workload then you can keep and run both the old and the new versions of the workload model.

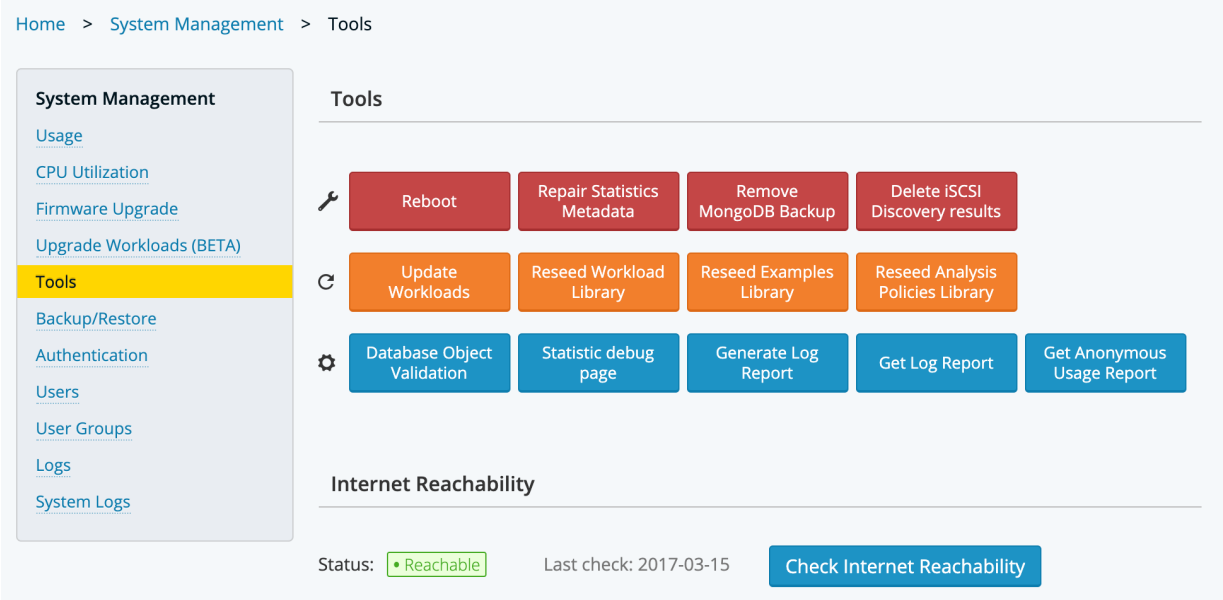
The currently supported VWF compatible workload models are:

- S3 Workload [131]

Tools

Generally, you should only use the *Tools* page when requested by Virtual Instruments Support.

With guidance from Virtual Instruments Support, you can use the *Tools* page to view a list of management tasks that you can execute to bring the Workload Generator up to date, if the update process did not complete successfully. You can also use this page to recover from other unexpected errors.



Reboot

Use the *Reboot* button to restart the WorkloadWisdom application. When WorkloadWisdom is executing tasks, in rare cases, some internal queues can become desynchronized. For example, if you see that a workload is still running, even though the workload should have stopped, based on your workload configurations. Only perform a reboot if there are no other ways to recover besides restarting the Workload Generator.

Repair Statistics Metadata

Use the *Perform Repair Statistics Metadata* button to bring the statistics engine up to date

When the WorkloadWisdom version is being updated, the statistics engine is also updated. If the update process is interrupted or encounters other unexpected errors, the statistics might be incomplete. For example, if you see that some charts are missing when you are running a workload, it is likely that the statistics engine encountered errors during the update.

Delete iSCSI Discovery Results

Use the *Delete iSCSI Discovery results* button to delete the iSCSI Discovery results that are saved to each iSCSI Test Bed that has performed iSCSI Discovery.

Update Workloads

Use the *Perform Update Workloads* page to bring workloads up-to-date.

When the WorkloadWisdom version is being updated, the workloads might also be updated. If the update process is interrupted, or encounters unexpected errors, the workloads might report an Out-of-Date error. This error displays in the workloads list after the update.

Reseed Workload Library

Use the *Reseed Workload Library* page to bring the standard workloads in the library up-to-date.

When your WorkloadWisdom version is being updated, the workload library might also be updated. If the update process is interrupted, or encounters other unexpected errors, the workload library might not be completely up to date.

Reseed Examples Library

Use the *Reseed Examples Library* to bring the example application workloads in the library up-to-date.

When the WorkloadWisdom version is being updated, the examples library might also be updated. If the update process is interrupted, or encounters other unexpected errors, the workload examples might not be completely up to date.

Reseed Analysis Policies Library

Use the *Reseed Analysis Policies Library* page to bring the default workload analysis policies in the library up-to-date.

When the WorkloadWisdom version is being updated, the workload analysis policies might also be updated. If the update process is interrupted, or encounters other unexpected errors, the workload analysis policies library might not be completely up to date.

Database Object Validation

Use the *Database Object Validation* page to check if there are objects that are in an unexpected state.

Each WorkloadWisdom Resource or Object has a set of validation rules that must be met in order for the product to function properly. For example, if a workload has request size distribution profiles that do not add up to 100%, the workload fails workload object validation rules. In rare cases, certain unexpected events result in an object that does not pass validation rules.

Statistic Debug Page

Use the *Statistic Debug Page* as directed by Virtual Instruments Support or Virtual Instruments Engineering.

Generate Log Report

Use *Generate Log Report* to generate a report that analyzes the web services' hits, performance and other information that Virtual Instruments Support and Engineering can use understand the overall health of WorkloadWisdom.

Get Log Report

Use *Get Log Report* to download the generated log report. You can sent the log report to Virtual Instruments Support.

Get Anonymous Usage Statistics Report

Use Get Anonymous Usage Statistics Report to get an anonymous report on the usage frequency of WorkloadWisdom functions and features. No sensitive information is kept in the report, such as workload names, workload characteristics, or test results. The generated report is in plain text format and you can review the information prior to sending to Virtual Instruments Support.

Internet Reachability

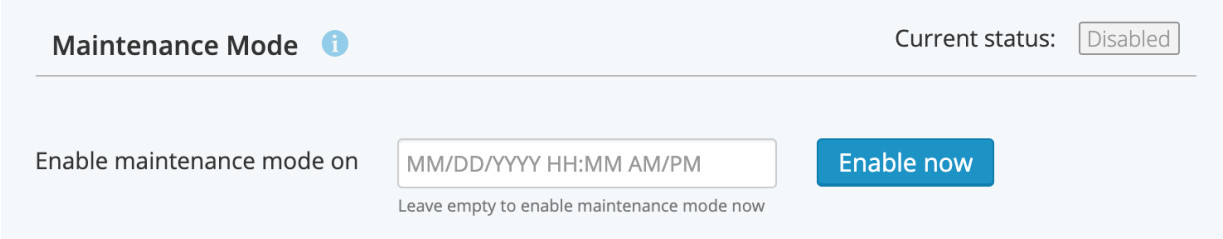
This is for future online product capabilities.

Maintenance Mode

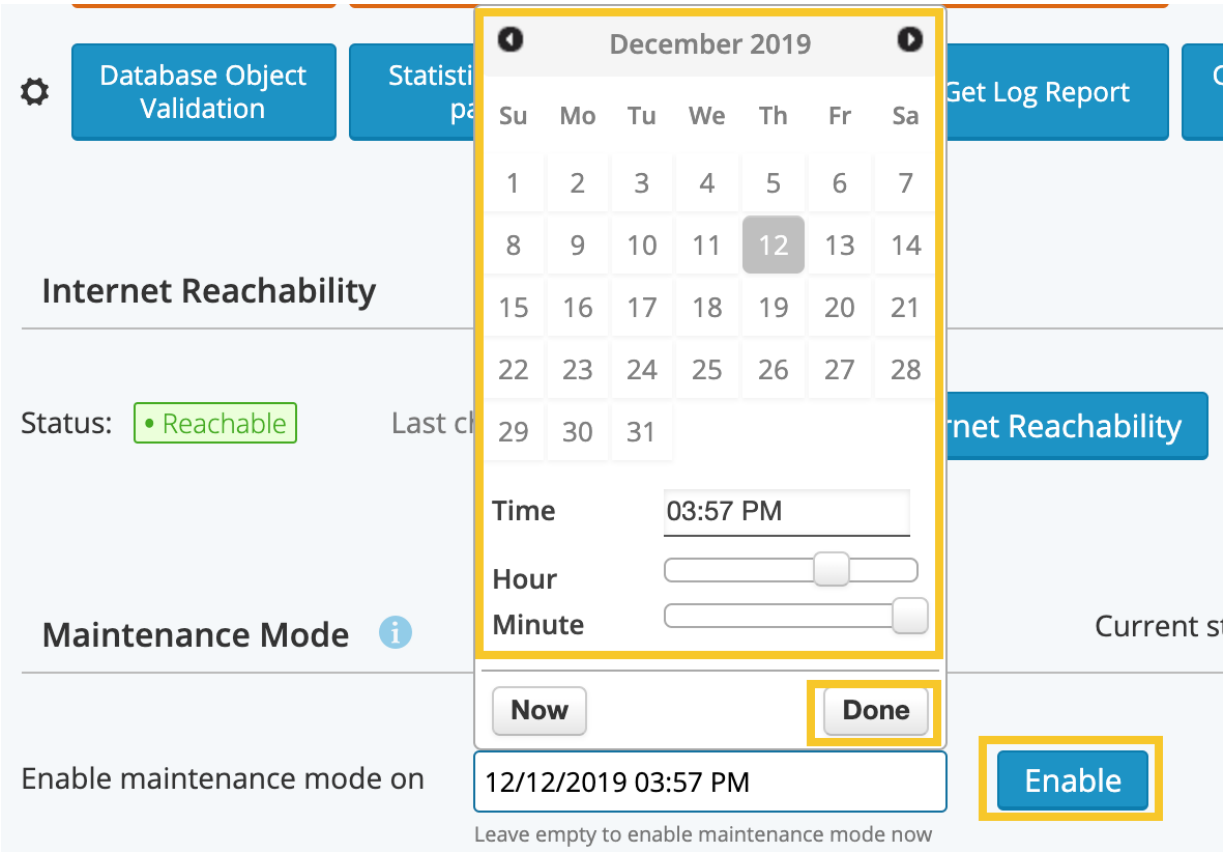
Maintenance Mode allows you, the administrator, to place the WorkloadWisdom in a state that prevents users from creating, removing, or modifying content. As administrator, you still have full access to the WorkloadWisdom.

Maintenance Mode is required when performing firmware updates. Maintenance mode can be useful when performing periodic cleanups, backup/restore, and other administrative tasks in which you need the WorkloadWisdom System in a constant state.

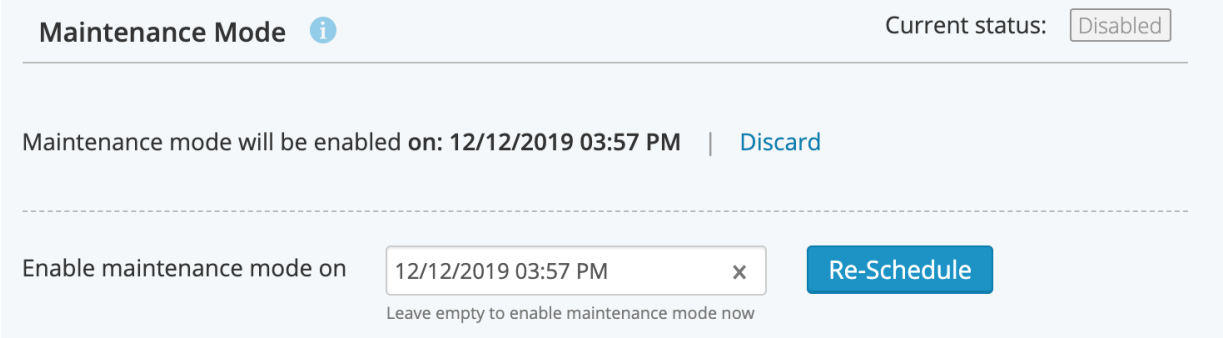
Click Enable Now to enable maintenance mode immediately.



You can also schedule maintenance mode for a specific time in the future. Using the calendar, select the date and time. Click **Done** to confirm the selection and click **Enable** to set it.



Once maintenance mode is scheduled, a log entry appears on the *Maintenance Mode* page, where you can discard or reschedule the date and time. A system maintenance message will also appear at the top of every page to let all users know about the upcoming maintenance event.



Backup/Restore

Use *Backup/Restore* to back up your user-created resources and overwrite all existing resources and data when you perform a restoration. Backup/restore is done using an on-premise FTP server. Resources are user-created items that can be used to produce results, but excluding the results.

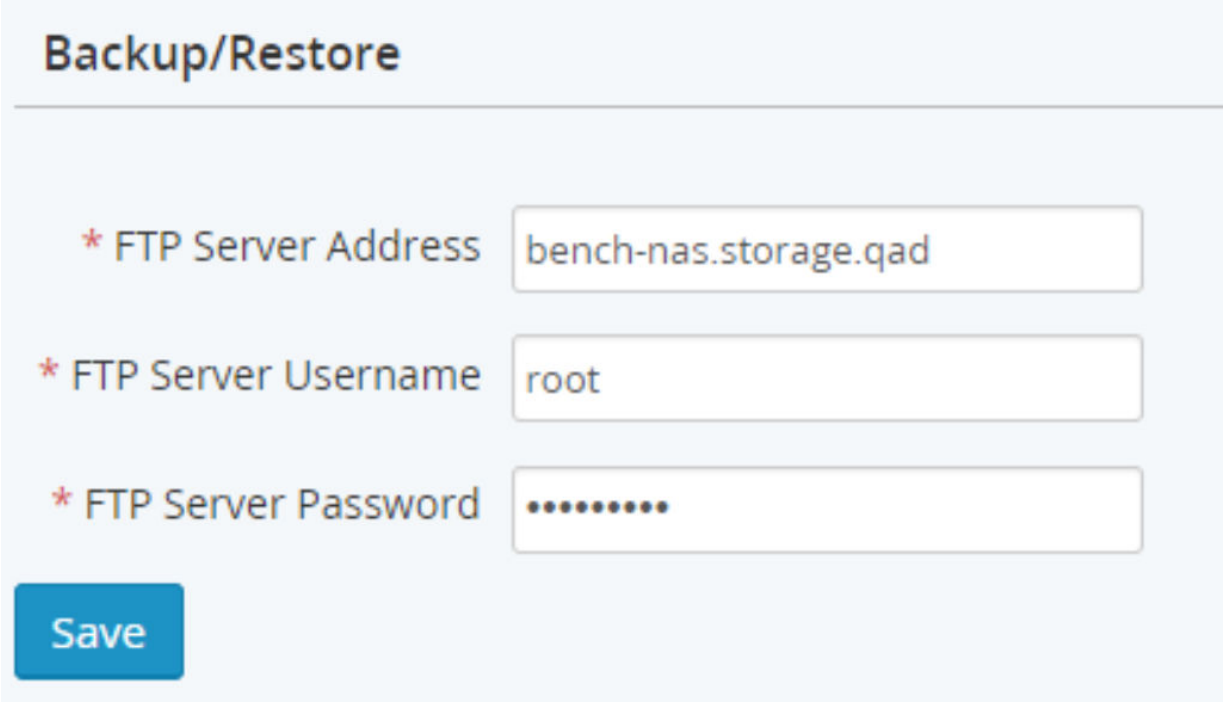
The following is a list of common user-created Resource content in WorkloadWisdom:

- Appliances (that is, Workload Generators)
- Test beds
- User-created production storage environments
- Workload tests (that is, every workload that was cloned from a default workload model)
- Workload suites
- Iteration suites
- Report templates
- User-created conditions

The following is a list of common Data content in WorkloadWisdom:

- Analyzed workload data
- Workload test results
- Suite results
- Reports

Before proceeding with backup/restore, you need to setup an on-premise FTP server that is accessible by the WorkloadWisdom administrator interface. Once the FTP server is up and running, you need to enter the access information in WorkloadWisdom.

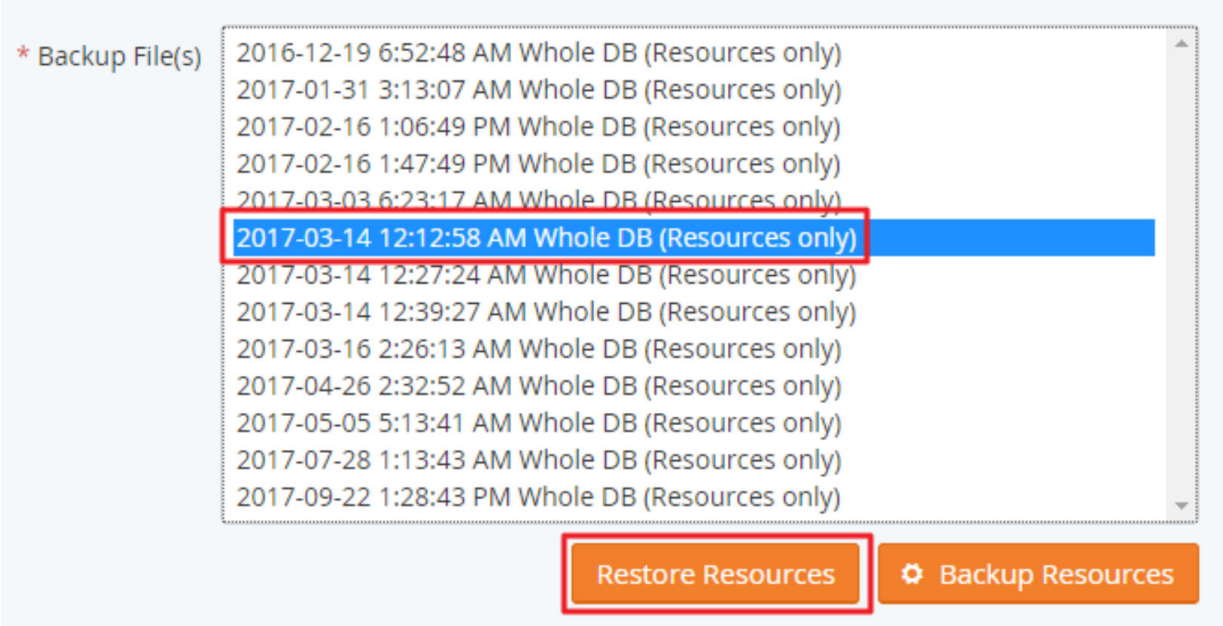


After you enter the *FTP Server Address*, *FTP Server Username*, and *FTP Server Password*, click **Save** to store the information.

NOTE
It is strongly recommended that you put the WorkloadWisdom System in maintenance mode before performing a backup.

To back up the latest resources, click **Backup Resources**. This creates a backup of all resources in a compressed file and copies the file to the FTP Server.

To restore the resources, click **Restore Resources**.



NOTE
When you perform a restore, all existing resources and data are overwritten. This task is irreversible!

Authentication

Use the *Authentication* page to view the current strategy authentication scheme. Additional authentication options might be available in the future.

Users

Use the *Users* page to display the list of users that have access to your Workload Generator. You can also add new users by clicking **+New User**. Only the default administrator account has full access. Full access means that you can perform firmware updates and view all resources. All other users are “regular” users. Regular users can only view resources owned by the specific user or public resources and cannot perform firmware updates.

User Groups

Use the *User Groups* page to display and edit user groups. You can also add new User Groups by clicking on **+New User Group**.

Logs

Use the *Logs* page to display the logged activities performed by WorkloadWisdom, such as “Workload start successful”, “Running Workload”, “Checking ports availability”, and “Saving Workload configuration.” You can also view errors such as “Can’t download pcap”, “Error pulling appliance info”, and “Failed to start test” on this page.

System Logs

Use the *System Logs* to display significant events that are useful for support and troubleshooting. System logs are sometimes referred to as WorkloadWisdom Logs.

You do not need to review the system logs unless requested to do so by Virtana.

Chapter 8

Automating, Integrating, and Using APIs

- **WorkloadWisdom and Your Environment**

Integrate WorkloadWisdom into your test and build environment.

WorkloadWisdom and Your Environment

WorkloadWisdom is designed to be integrated into your existing test and build environment.

WorkloadWisdom provides a set of RESTful APIs that enables you to integrate it into your automation environment and workflows.

Refer to the API Docs in the product for a full set of available APIs and examples of how to use them. At a high level, the available APIs allow you to query Resources such as Workload Generators, start a Workload Test, collect test results, and more.

Click *API Docs* at the bottom right of every screen to access this documentation.

Chapter 9

Example Workload Descriptions

- **File Workload Examples**
- **Block Workload Examples**

File Workload Examples

The library of Example File Workloads provides a version of each workload for Network File System (NFS) Versions 3 and 4.1, and Server Message Block (SMB) 2.

Database OLTP

The (File Protocol) Database OLTP Example is designed to mimic an Online Transaction Processing Database in terms of block sizes, command ratios, and typical file sizes. It does not take into consideration typical ratios of access within files, as the available data does not cover this.

OLTP access is often very read heavy, accessing large files in a flat or shallow file hierarchy. This leads to minimal meta-command activity. The suggested block size can vary dramatically from the configured 8KB size selected for this example.

Fileserver

Based on the File protocol workload model, the Fileserver Example is designed to mimic an in house fileserver in terms of block sizes, command ratios, and typical file sizes. There is no need to take into account access within files because most files are read/written completely rather than in portions.

Fileserver access is often very read heavy, with a medium-large file hierarchy, leading to some meta command activity. The file sizes can vary greatly from one fileserver to another depending on the type of business the fileserver is used to support. This example uses a wide distribution of sizes to simulate a mixed environment.

Photo/Image Server

Based on the File protocol workload model, the Photo/Image Server Example is designed to mimic direct access to a photo server in terms of block sizes, command ratios, and typical file sizes. There is no need to take into account access within files because most files are read/written completely rather than in portions.

This example is for a photo server that is changing quite frequently and where users are frequently searching for content rather than accessing photos directly because they are well indexed else where in a webserver or other portal. If a portal where used in front of this photo server the meta traffic would decrease dramatically. Also many photo servers do not change as frequently as this example where often photos are uploaded but never accessed.

Video Streaming

Based on the File protocol workload model, the Video Streaming Example is designed to mimic a semi-commercial level video-streaming server in terms of block sizes, command ratios, and typical file sizes. The example mimics a server where the there is a cap of 5GB on the videos that are uploaded.

Most videos are being accessed but a few times but not at the level that a commercial video streaming system would. This video system assumes that access is from a decently designed portal that is knowledgeable about the video content so minimal searching is required to find and access the videos.

Linux Server

Based on the File protocol workload model, Linux Server Example is designed to simulate 40 small sized Virtualized Linux Server OS's per Volume/Share with any significant

application data housed elsewhere. The Workload would also simulate a rather high level of activity such as a system that is doing significant disk operations, as the Workloads are not capped.

Windows Server

Based on the File protocol workload model, the Windows Server Example is designed to simulate 40 small sized Virtualized Windows Server OS's per Volume/Share with any significant application data housed elsewhere. The Workload would also simulate a rather high level of activity such as a system that is doing significant disk operations, as the Workloads are not capped.

Webserver

Based on the File protocol workload model, the Webserver Example is designed to mimic an in webserver in terms of block sizes, command ratios, and typical file sizes. There is no need to take into account access within files because most files are read/written completely rather than in portions.

Webserver access is often very read heavy. The amount of metadata content in this example is quite low compared to other example where the amount of meta command access can often be as high as 95% of the traffic. This depends on the selected webservers caching algorithms. Webservers with high amounts of meta traffic are often GetAttr (as high as 62%), Lookups and Access heavy so a "commands distribution" setting should be used if the metadata is increased.

Block Workload Examples

The library of Example Block Workloads provides a version of each workload for both fibre channel and iSCSI.

File Server

Based on the Block protocol workload model, this workload provides an example of a fileserver over FC and iSCSI. This workload is based on publicly available data that provides read/write ratios and block size distribution. It leverages FC/iSCSI defaults for other settings.

MicrosoftExchange 2003-2013

Based on the Block protocol workload model, this workload provides an example of MicrosoftExchange 2003, 2007, 2010, and 2013 over FC and iSCSI. This workload is based on publicly available data that provides read/write ratios and block size distribution. It leverages FC/iSCSI defaults for other settings. Exchange workloads typically include multiple discrete workloads depending on the operation (that is, DB, Logging and so forth). This workload is aggregate of those patterns.

Microsoft SQL Server

Based on the Block protocol workload model, this workload provides an example of Microsoft SQL Server over FC and iSCSI. This workload is based on publicly available data that provides read/write ratios and block size distribution and leverages FC/iSCSI defaults for other settings. Database workloads typically include multiple discrete workloads depending on the operation (that is, OLTP, logging and so forth) and vary by version. This workload is aggregate of those patterns.

Splunk

Based on the Block protocol workload model, this workload provides an example of Splunk®over FC and iSCSI. This workload is based on publicly available data that provides read/write ratios and block size distribution. It leverages FC defaults for other settings.

Oracle

Based on the Block protocol workload model, this workload provides an example of an Oracle®Database over FC and iSCSI. This workload is based on publicly available data that provides read/write ratios and block size distribution. It leverages FC/iSCSI defaults for other settings. Database workloads typically include multiple discrete workloads depending on the operation (that is, OLTP, logging and so forth) and vary by version. This workload is aggregate of those patterns.

Chapter 10

Example Workload Data Import Format

WorkloadWisdom does not require the imported Workload Data (.csv) from storage arrays, or other devices that contain storage workload information, to be in any specific format. The engine is designed to be completely generic.WorkloadWisdom requires a specific set of metrics to perform its analysis, and provides Workload Analysis Policies to map the imported data to the metrics required by WorkloadWisdom to perform analysis. Therefore, WorkloadWisdom requires the imported .csv to contain data, identified by the column label, required by an Analysis Policy.

An Analysis Policy is essentially a “conversion” template that converts the imported data into the specific set of metrics that WorkloadWisdom requires to perform analysis. The metrics required to perform analysis include, but not limited to, read/write percentages, read/write IO sizes, and IOPS.

Analysis Policies are designed to be extensible and customizable. Therefore, you can either edit an Analysis Policy so that it can interpret the Input Data from the .csv, or you can edit the import .csv column labels to so that it can be interpreted by an available Analysis Policy.

As different storage arrays keep track of workload data differently, and produce .csv in different structures, the Analysis Policy also provides mechanisms to perform calculations on the data in the imported .csv to derive the metrics WorkloadWisdom requires to perform analysis on the .csv. For example, if a storage array provides data on Throughput and IOPS, but does not provide data on average Block Size, then the Analysis Policy can derive the average Block Size by dividing Throughput by IOPS.

As an example, here is a truncated example of an imported .csv from a storage array:

Timestamp	LDEV Number	Read I/O /sec	Write I/O /sec	Read Response Rate	Write Response Rate	R+W Response Rate	Read Transfers /sec	Write Transfers /sec	Sequential R+W I/O /sec	Sequential R+W Transfers /sec
04 27 2016 10:10:00	Process1A	0	253.26666	0	98.847404	98.847404	0	0.18333334	192.91667	0.11666667
04 27 2016 10:10:00	Process2A	95.6	5111.4165	1354.9421	85.67831	108.98179	23.783333	17.883333	892.7	25.066668
04 27 2016 10:10:00	Process3A	0	251.23334	0	75.23339	75.23339	0	0.13333334	242.13333	0.11666667
04 27 2016 10:10:00	Process4A	0	249.38333	0	75.19675	75.19675	0	0.13333334	248.71666	0.11666667
04 27 2016 10:10:00	Process4A	0	0	0	0	0	0	0	0	0
04 27 2016 10:10:00	Process4A	0	0	0	0	0	0	0	0	0
04 27 2016 10:10:00	Process4A	0	0	0	0	0	0	0	0	0
04 27 2016 10:10:00	Process4A	0	0	0	0	0	0	0	0	0
04 27 2016 10:10:00	Process1B	136.36667	254	82.01112	97.97386	92.397606	1.3833333	0.18333334	300.81668	1
04 27 2016 10:10:00	Process2B	416.3	5124.7334	987.19916	86.06849	153.77078	101.2	17.916666	1058.8334	64.46667
04 27 2016 10:10:00	Process3B	127.916664	251.26666	78.78954	75.16317	76.38652	1.0666667	0.13333334	285.28333	0.6333333
04 27 2016 10:10:00	Process4B	124.65	251.05	78.22068	75.20325	76.20438	1.05	0.13333334	303.06668	0.8
04 27 2016 10:10:00	Job1	318.83334	46.783333	353.20923	100.50656	320.87408	2.4833333	0.45	1.5333333	0.016666668
04 27 2016 10:10:00	Job2	320.36667	48.583332	355.04257	99.620995	321.40866	2.5	0.46666667	1.1833333	0.016666668
04 27 2016 10:10:00	Job3	323.9	51.433334	348.8821	102.7452	315.15305	2.5333333	0.53333336	2.6666667	0.033333335
04 27 2016 10:10:00	Job4	330.68332	53.316666	343.1538	101.17032	309.5555	2.5833333	0.51666665	7.1	0.06666667
04 27 2016 10:10:00	Job5	324.53333	47.75	351.5707	101.15451	319.4517	2.5333333	0.45	1.95	0.016666668
04 27 2016 10:10:00	Job6	317.06668	49.133335	355.1729	101.94203	321.19672	2.4833333	0.45	1.55	0.016666668
04 27 2016 10:10:00	Job7	319.53333	48.316666	351.61758	103.706825	319.05478	2.5166667	0.46666667	2.5166667	0.033333335
04 27 2016 10:10:00	Job8	331.1	50.116665	352.49545	104.85965	319.94	2.5833333	0.5833333	1.65	0.016666668

And here is a truncated example of an Analysis Policy that is defined to interpret the imported .csv:

Field / Input Parameter	Label	Units	Type	Format
Date and Time	Time	Date Time	Index(time)	%m %d %Y %H:%M:%S
LDEV Number	LUN	N/A	Index(device)	
Read I/O /sec	Read IOPs	N/A	Metric(req)	
Write I/O /sec	Write IOPs	N/A	Metric(req)	
Read Response Rate	Read Latency	µs	Metric(opt)	
Write Response Rate	Write Latency	µs	Metric(opt)	
Read Hit %	Read Hit %	N/A	Metric(req)	
Read Xfer /sec	Read Throughput	MB	Metric(req)	
Write Xfer /sec	Write Throughput	MB	Metric(req)	
Sequential Total I/O /sec	Sequential IOPs	N/A	Metric(opt)	
Random Total I/O /sec	Random IOPs	N/A	Metric(opt)	

For some of the metrics that are not directly available from the imported .csv such as Average Write IO Size, the following Calculation Rules can be applied to derive the Average Write IO Size:

Total IOP Attempted	Read IOPs + Write IOPs	N/A	Metric(req)
Average Read IO Size	(Read Throughput / Read IO...	N/A	Metric(req)
Average Write IO Size	(Write Throughput / Write IO...	N/A	Metric(req)
Average Write IO Size Per De...	(LUN.Scope Value.Write Thro...	N/A	Metric(req)
Average Read IO Size Per De...	(LUN.Scope Value.Read Thro...	N/A	Metric(req)

In this example, the Calculated Metric “Average Write IO Size” is derived by the formula (Write Throughput/Write IOPs) * 1024, where “Write Throughput” is interpreted from the data from the column labeled “Write Xfer /sec” in the imported .csv, and “Write IOPS” is interpreted from the data from the column labeled “Write I/O /sec”.

The Workload Data Importer is extremely flexible, and the fidelity of the analyzed workload depends on the Analysis Policy you define, and most importantly, the amount of data that is available in the imported .csv file.

Chapter 11

Supported Analysis Policies

The following table lists the supported analysis policies, vendors, products, protocols, and workload models.

Table 4. Supported Analysis Policies

Vendor	Product	Protocols	Workload Models	Analysis Policy
Dell EMC™	VMAX®	Fibre Channel	Constant FC	EMC VMAX Specify Blocks creates FC Constant Workload
Dell EMC	VMAX	iSCSI	Constant iSCSI	EMC VMAX Specify Blocks creates iSCSI Constant Workload
Dell EMC	VMAX	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	EMC VMAX Specify Blocks creates SCSI Temporal Workload
Dell EMC	VMAX	Fibre Channel	Constant FC	EMC VMAX v1.6.1.8 creates FC Constant Workload

Vendor	Product	Protocols	Workload Models	Analysis Policy
Dell EMC	VMAX	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	EMC VMAX® v1.6.1.8 creates SCSI Temporal Workload
Dell EMC	VMAX	iSCSI	Constant iSCSI	EMC VMAX v1.6.1.8 creates iSCSI Constant Workload
Dell EMC	VNX®	Fibre Channel	Constant FC	EMC VNX Specify Blocks creates FC Constant Workload
Dell EMC	VNX	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	EMC VNX® Specify Blocks creates SCSI Temporal Workload
Dell EMC	VNX	iSCSI	Constant iSCSI	EMC VNX® Specify Blocks creates iSCSI Constant Workload
Dell EMC	Unisphere® for VNX	Fibre Channel	Constant FC	The Unisphere for VNX v2 Specify Blocks creates FC Constant Workload
Dell EMC	Unisphere for VNX	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	The Unisphere for VNX v2 Specify Blocks creates SCSI Temporal Workload
Dell EMC	Unisphere for VNX	iSCSI	Constant iSCSI	The Unisphere for VNX v2 Specify Blocks creates iSCSI Constant Workload
Dell EMC	Isilon	NFSv3	Temporal NFSv3	Dell-EMC Isilon Create NFSv3 Temporal Workload (note: script required)
Dell EMC	Isilon	SMB	Temporal SMB	Dell-EMC Isilon Create SMB Temporal Workload (note: script required)
HDS	Tuning Manager	Fibre Channel	Constant FC	Hitachi® Tuning Manager Specify Blocks creates FC Constant Workload
HDS	Tuning Manager	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	Hitachi® Tuning Manager Specify Blocks creates SCSI Temporal Workload

Vendor	Product	Protocols	Workload Models	Analysis Policy
HDS	Tuning Manager	iSCSI	Constant iSCSI	Hitachi® Tuning Manager Specify Blocks creates iSCSI Constant Workload
HP	XP Performance Advisor	Fibre Channel	Constant FC	HP XP Performance Advisor Specify Blocks creates FC Constant Workload
HP	XP Performance Advisor	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	HP XP Performance Advisor Specify Blocks creates SCSI Temporal Workload
HP	XP Performance Advisor	iSCSI	Constant iSCSI	HP XP Performance Advisor Specify Blocks creates iSCSI Constant Workload
HPE™	3PAR StoreServ	Fibre Channel	Constant FC	HPE 3PAR StorServ Analysis Policy creates FC Constant Workload
HPE	3PAR StoreServ	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	HPE 3PAR StorServ Analysis Policy creates SCSI Temporal Workload
HPE	3PAR StoreServ	iSCSI	Constant iSCSI	HPE 3PAR StorServ Analysis Policy creates iSCSI Constant Workload
Huawei	OceanStor	Fibre Channel iSCSI	Temporal FC Temporal iSCSI	Huawei pre-parsed stats Analysis Policy produces SCSI Temporal Workload
IBM®	XIV®	Fibre Channel	Constant FC	IBM XIV® Analysis Policy Specify Blocks creates FC Constant Workload
IBM	XIV	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	IBM XIV Analysis Policy Specify Blocks creates SCSI Temporal Workload
IBM	XIV	iSCSI	Constant iSCSI	IBM XIV Analysis Policy Specify Blocks creates iSCSI Constant Workload

Vendor	Product	Protocols	Workload Models	Analysis Policy
NetApp®	OnCommand®	Fibre Channel	Constant FC	NetApp OnCommand® Performance Manager Specify Blocks creates FC Constant Workload
NetApp	OnCommand	iSCSI	Constant iSCSI	NetApp OnCommand Performance Manager Specify Blocks creates iSCSI Constant Workload
NetApp	OnCommand	SMB2	Constant SMB2	NetApp OnCommand Performance Manager Specify Blocks creates SMB Constant Workload
NetApp	OnCommand	NFSv3	Constant NFSv3	NetApp OnCommand Performance Manager Specify Blocks creates NFSv3 Constant Workload
NetApp	OnCommand	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	NetApp OnCommand Performance Manager Specify Blocks creates SCSI Temporal Workload
NetApp	OnCommand	NFSv3	Temporal NFSv3	NetApp OnCommand Performance Manager Specify Blocks creates NFSv3 Temporal Workload
NetApp	LatX	Fibre Channel	Constant FC	NetApp LatX Specify Blocks creates FC Constant Workload
NetApp	LatX	iSCSI	Constant iSCSI	NetApp LatX Specify Blocks creates iSCSI Constant Workload
NetApp	LatX	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	NetApp LatX Specify Blocks creates SCSI Temporal Workload
Oracle®	ZFS	Fibre Channel	Constant FC	Oracle ZFS Storage Appliance Analytics creates FC Constant Workload

Vendor	Product	Protocols	Workload Models	Analysis Policy
Oracle	ZFS	iSCSI	Constant iSCSI	Oracle ZFS Storage Appliance Analytics creates iSCSI Constant Workload
Oracle	ZFS	NFSv3	Temporal NFSv3	Oracle ZFS Storage Appliance Analytics creates NFSv3 Temporal Workload
Oracle	ZFS	Fibre Channel, iSCSI	Temporal FC, Temporal iSCSI	Oracle ZFS Storage Appliance Analytics creates SCSI Temporal Workload
Pure Storage	Purity	Fibre Channel, iSCSI	Temporal FC, Temporal iSCSI	Pure Storage Array creates SCSI Temporal Workload (note: script required)
Pure Storage	Purity	Fibre Channel	Constant FC	Pure Storage Array creates FC Constant Workload (note: script required)
Pure Storage	Purity	iSCSI	Constant iSCSI	Pure Storage Array creates iSCSI Constant Workload (note: script required)
Virtual Instruments	VirtualWisdom	NFSv3	Temporal NFSv3	VI VirtualWisdom ProbeNAS Analysis Policy creates NFSv3 Temporal Workload
Virtual Instruments	VirtualWisdom	SMB2	Temporal SMB2	VI VirtualWisdom ProbeSMB Analysis Policy creates SMB Temporal Workload
Virtual Instruments	VirtualWisdom	Fibre Channel	Constant FC	VI VirtualWisdom4® Analysis Policy creates FC Constant Workload
Virtual Instruments	VirtualWisdom	iSCSI	Constant iSCSI	VI VirtualWisdom4 Analysis Policy creates iSCSI Constant Workload

Vendor	Product	Protocols	Workload Models	Analysis Policy
Virtual Instruments	VirtualWisdom	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	VI VirtualWisdom4 Analysis Policy creates SCSI Temporal Workload
Virtual Instruments	LDX LAB	Fibre Channel	Constant FC	LDX LAB Specify Blocks creates FC Constant Workload
Virtual Instruments	LDX LAB	Fibre ChanneliSCSI	Temporal FCTemporal iSCSI	LDX LAB Specify Blocks creates SCSI Temporal Workload
Virtual Instruments	LDX LAB	iSCSI	Constant iSCSI	LDX LAB Specify Blocks creates iSCSI Constant Workload
Virtual Instruments	WorkloadWisdom (custom)	Fibre Channel	Constant FC	Default Template Specify Blocks creates FC Constant Workload
Virtual Instruments	WorkloadWisdom (custom)	iSCSI	Constant iSCSI	Default Template Specify Blocks creates iSCSI Constant Workload
Virtual Instruments	WorkloadWisdom (custom)	Fibre Channel iSCSI	Temporal FC, Temporal iSCSI	Default Template Specify Blocks creates SCSI Temporal Workload

Chapter 12

Contact Information

Sales Inquiries

To speak with a sales representative:

Complete the form at virtana.com/contact-us/.

Call us at +1-888-522-2557.

Support for WorkloadWisdom and Xangati

Online Support

WorkloadWisdom and Xangati support is available Monday-Friday.

www.virtana.com/support

Technical Support

workloadwisdom.support@virtana.com

Feedback

We appreciate your input to help us improve the quality of our products and documentation. Send your suggestions, comments, and questions about Virtana products and documentation to:

Product Feedback

feedback@virtana.com

Documentation Feedback

techpubs@virtana.com

Chapter 13

WorkloadWisdom Copyright and Trademarks

Virtual Instruments reserves the right to revise these specifications without notice or penalty.

Trademarks

Amazon is a trademark of Amazon Services LLC and/or its affiliates.

Dell Technologies, Dell, EMC, Dell EMC, Unisphere, Unity, VMAX, and VNX and other trademarks are trademarks of Dell Inc. or its subsidiaries.

HPE is a trademark of Hewlett-Packard Development Company, L.P.

Hitachi is a registered trademark of Hitachi Ltd.

Huawei is a registered trademark of Huawei Technologies Co. Ltd.

IBM and IBM XIV are registered trademarks of IBM in the United States, other countries, or both.

Load DynamiX is a registered trademark of Load Dynamix, Inc.

NetApp and OnCommand are registered trademarks of NetApp, Inc.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Pure Storage and the Pure Storage logo is a trademark or registered trademark of Pure Storage, Inc. in the United States and other countries.

Splunk is a registered trademark of Splunk Inc. in the United States and other countries.

Hyper-V is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Virtual Instruments, VirtualWisdom, and WorkloadWisdom are registered trademarks of Virtual Instruments Corporation (d/b/a Virtana).