Turbonomic 7.22.5 Target Configuration Guide v2

Turbonomic, Inc

500 Boylston St, 7th floor
Boston, MA 02116 USA
Phone: (844) 438-8872
www.turbonomic.com

COPYRIGHT

END-USER LICENSE AGREEMENT

https://cdn.turbonomic.com/wp-content/uploads/Turbonomic_Click_Through_Customer-License.pdf

Contents

Target Configuration

A target is a service that performs management in your virtual environment. Turbonomic uses targets to monitor workload and to execute actions in your environment. Target Configuration specifies the ports Turbonomic uses to connect with these services. You must install Turbonomic on a network that has access to the specific services you want to set up as targets.

For each target, Turbonomic communicates with the service via the management protocol that it exposes — The REST API, SMI-S, XML, or some other management transport. Turbonomic uses this communication to discover the managed entities, monitor resource utilization, and execute actions.

You can assign instances of the following technologies as Turbonomic targets:

- Applications and Databases
  - Microsoft SQL Server 2008 R2, 2012, 2014, and 2016
  - AppDynamics 4.1+
  - DynaTrace 1.1+
  - AppInsights 4.1+
- Cloud Native Targets
  - OpenShift 3.3+
  - Kubernetes
- Fabric and Network
  - Cisco UCS Manager 3.1+
  - HPE OneView 3.00.04+
- Guest OS Processes
  - SNMP
  - WMI: Windows versions 2019, 2016, 2012 / 2012 R2, 2008 R2, 10, 8 / 8.1, and 7
- Hyperconverged
  - Nutanix Community Edition
- Hypervisors
  - VMware vCenter 5.1, 5.5, 6.0, 6.5, 6.7, and 7.0
  - Microsoft Hyper-V 2008 R2, Hyper-V 2012, and Hyper-V 2012 R2

- Orchestrator Targets
  - Action Script
- Private Cloud Managers
  - Microsoft System Center 2012 Virtual Machine Manager and System Center 2012 R2 Virtual Machine Manager
- Public Cloud Managers
  - Amazon AWS
  - Amazon AWS Billing
  - Microsoft Azure
  - Microsoft Enterprise Agreement
- Storage Managers
  - Pure Storage F-series and M-series arrays
  - NetApp Cmode/7mode using ONTAP 8.0+ (excluding AFF and SolidFire)
  - EMC VMAX using SMI-S 8.1+
  - EMC ScaleIO 2.x and 3.x
  - HPE 3PAR InForm OS 3.2.2+, 3PAR SMI-S, 3PAR WSAPI
- Virtual Desktop Infrastructure
  - VMware Horizon

Transport Layer Security Requirements

Turbonomic requires Transport Layer Security (TLS) version 1.2 to establish secure communications with targets. Most targets should have TLSv1.2 enabled. However, some targets might not have TLS enabled, or they might have enabled an earlier version. In that case, you will see handshake errors when Turbonomic tries to connect with the target service. When you go to the Target Configuration view, you will see a Validation Failed status for such targets.

If target validation fails because of TLS support, you might see validation errors with the following strings:

- `No appropriate protocol`

  To correct this error, ensure that you have enabled the latest version of TLS that your target technology supports. If this does not resolve the issue, please contact Turbonomic Technical Support.
- `Certificates do not conform to algorithm constraints`

  To correct this error, refer to the documentation for your target technology for instructions to generate a certification key with a length of 1024 or greater on your target server. If this does not resolve the issue, please contact Turbonomic Technical Support.

Adding and Removing Targets

The target services your Turbonomic installation will manage appear in the Target Configuration list. You can add, remove, and edit entries in this list. Note that the target service's account must be configured with privileges that support the Turbonomic activities you want to perform. For example, the following list shows how vCenter privileges correspond to activities Turbonomic can perform:

- **Read Only** — Enables Turbonomic monitoring and simulation (what-if scenarios) only

- **VCenter Administrator** — Enables Turbonomic monitoring, simulation (what-if scenarios), and automation functions
- **Enable Datastore Browse** — Enabling this property for the account gives Turbonomic the privileges it needs to enable its storage management functionality

Adding Targets

To add a target service, click the **Target Configuration** button, provide the requested information, and click **Apply** to validate those targets and start a new discovery.

> **NOTE:**
> As you add targets, be sure not to add duplicate entries for the same target.

Typical information you provide includes:

- Target Type — Choose among the supported technologies

  After you choose the technology, then choose the specific target type for that technology. For example, for Cloud Management, you can choose AWS.
- Hostname or IP address — The address of the target service you want to add
- User Name — A valid account username for the target service
- Password — A password for the target service account

Removing Targets

To remove a target, select the entry in the list and then click **Delete**.

Hypervisor Targets

A hypervisor is a service that creates and runs virtual machines (VMs), providing the VMs compute and storage resources. When you connect Turbonomic to hypervisor targets in your environment, Turbonomic controls your environment, assuring application performance while also utilizing resources as efficiently as possible.

One of the first steps in any Turbonomic deployment is to connect to the hypervisors within your environment. Once connected, Turbonomic discovers the VMs, the physical machines that host the VMs, the datastores that provide storage resources to the physical machines, and the virtual datastores that provide storage resources to the VMs.

Supply Chain

Each hypervisor requires a physical machine (host) and one or more datastores to provide compute and storage resources. Virtual machines (VMs) run on those physical resources, and the VMs in turn provide resources to applications.

At the bottom of the supply chain, physical machines consume resources from data centers.

If your environment includes SAN technologies such as disk arrays, then the storage consumes resources from that underlying technology. If you add these storage targets, then Turbonomic extends the supply chain analysis into the components that make up the disk array. For more information, see Storage Manager Targets *(on page 50)*.

Actions

Turbonomic recommends actions for the hypervisor supply chain as follows.

> **NOTE:**
> This is a general list of actions for entities discovered for hypervisors. Detailed actions per target are described in each target section.

Entity Type	Action
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine
Physical Machines	• Start Physical Machine • Provision Physical Machine • Suspend Physical Machine
Storage	• Start Storage • Provision Storage • Suspend Storage • Move (only with Storage Targets configured) • Resize (only with Storage Targets configured)
Consumer Virtual Datacenters	• Resize Consumer vDC • Provision Consumer vDC

Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB)

Entity Type	Commodity
	• Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore

Entity Type	Commodity
	Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Datacenter	**NOTE:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the physical machines in the datacenter. • Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared of memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Kilobytes (KB)
Provider Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Provider vDC.

Entity Type	Commodity
	Measured in Kilobytes (KB)
Consumer Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Consumer vDC. Measured in Kilobytes (KB)

Hyper-V

If you have a small number of Hyper-V hosts in your environment, you can add them individually as Turbonomic targets. Also, if you have deployed the Hyper-V hosts in a clustered domain (for example as a failover cluster), you can specify one Hyper-V host as a target and Turbonomic automatically add the other members of that cluster.

Note that for large Hyper-V environments, it's typical to manage the hosts via System Center Virtual Machine Manager (VMM). You can specify the VMM server as a target and Turbonomic will use it to discover and manage its child Hyper-V hosts. If you use VMM, you should not add individual Hyper-V hosts as targets. For information about adding VMM targets, see Adding Virtual Machine Manager Targets *(on page 30)*.

> **NOTE:**
> Even if you manage Hyper-V using a VMM target, you must still configure remote management on each Hyper-V server. This Hyper-V topic includes instructions to configure remote management — see Enabling Windows Remote Management *(on page 130)*.

For accurate SMB storage calculations, Turbonomic requires a VMM target.

Prerequisites

- Create a user account that Turbonomic can use to connect to your Hyper-V servers. See Creating a Service User Account in Hyper-V *(on page 15)*
- Configure remote management on each Hyper-V server. Refer to Enabling Windows Remote Management *(on page 130)*
- Your Hyper-V environment must not use Server Message Block (SMB) storage.

  To manage SMB storage, Turbonomic requires a VMM target, and that VMM instance must manage the Hyper-V hypervisors and the SMB storage that they use.

  Managing a Hyper-V plus SMB environment via Hyper-V targets will result in incorrect data collection for SMB storage.

Adding Hyper-V Targets

Once you've enabled remote management, you can add your Hyper-V hosts as targets. To add Hyper-V targets, select the **Hypervisors > Hyper-V** option on the Target Configuration page and provide the following information:

- Address

  The FQDN of the Hyper-V host. If you're using the "Discover Host Cluster" below to add an entire cluster, enter the name of any one of the Hyper-V hosts in the cluster.

  Note that you can enter an IP address for the host, but you must first configure an SPN on the host. Turbonomic recommends that you use the FQDN in this field.
- Port number

  The port number for the remote management connection. The default HTTP port is 5985; the default HTTPS port is 5986.
- Secure connection

  Select this option to use a secure connection (HTTPS). Make sure the required certificate is configured for use on the host.
- Full domain name

  The full domain name of the cluster to which the host belongs.
- Discover Host Cluster

  Turbonomic discovers and adds all Hyper-V hosts in the named cluster if this option is checked. Note that each server must be configured to allow remote management. You may find it helpful to configure WinRM using a GPO so new servers are configured automatically (see Enabling WinRM Via a GPO *(on page 131)*).
- Username

  The username of a user account Turbonomic can use to connect to the Hyper-V host. If you checked "Discover Host Cluster" in the field above, use an account that is valid for all Hyper-V hosts in that cluster.
- Password

  Password for account used.

> **NOTE:**
> If your Hyper-V hosts are running in a Nutanix environment, you must understand pinning a Nutanix Controller VM. For more information, see Pinning Controller VMs in Generic Hypervisor Mode *(on page 114)*.

Exporting Hyper-V Virtual Machines

In Hyper-V environments, you must be sure that all VMs have unique IDs.

Hyper-V supports the export of a VM, so that you can create exact copies of it by importing those exported files. The Copy import type creates a new unique ID for the imported VM. When importing VMs in your environment, you should always use the Copy import type.

Turbonomic uses the unique ID to discover and track a VM. If your environment includes multiple VMs with the same ID, then discovery will assume they are the same VM. As a result, the counts for VMs will be incorrect.

Supported Actions

For each discovered entity within the hypervisor supply chain, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations Only
Virtual Machine	Start, Move, Suspend, Resize Down, Resize Up	Terminate, Provision, Reconfigure
Physical Machine	Start, Suspend	Terminate, Provision
Storage		Provision

Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters

Entity Type	Commodity
	Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB)
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Datacenter	**NOTE:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the physical machines in the datacenter. • Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared of memory among VMs running on the host. ESX-only

Entity Type	Commodity
	Measured in Kilobytes (KB) • CPU Ready The utilization of the PM's allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Kilobytes (KB)

Creating A Service User Account

The service account Turbonomic uses to connect to a Hyper-V host must be an Active Directory domain account. The account must have full access to the cluster. To create such an account, execute the following command at a PowerShell prompt:

`Grant-ClusterAccess <domain>\<service_account> -Full`

Additionally, the service account must have specific local access rights on each host. The easiest way to grant Turbonomic the access it requires is to add the domain account to the `Local Administrators` group on each Hyper-V server.

Some enterprises require that the service account does not grant full administrator rights. In that case, you can create a restricted service account on every Hyper-V host.

> **NOTE:**
> Turbonomic does not support Restricted User Accounts on Windows 2012 Hyper-V nodes.

To create a restricted service account on your Hyper-V hosts:

1. Add the service account to each of the following local groups:
   - `WinRMRemoteWMIUsers__` (or `Remote Management Users`)
   - `Hyper-V Administrators`
   - `Performance Monitor Users`

   > **NOTE:**
   > These groups are examples, only. If your version of Windows Server does not include these groups, contact Technical Support for assistance.
2. Grant permissions to the service account.

   In the WMI Management console, grant the following permissions to the service account:
   - `Enable Account`
   - `Remote Enable`
   - `Act as Operating System` (For Windows 2016)
3. Configure the WinRM security descriptor to allow access by the service account:
   - At a PowerShell prompt, execute `winrm configSDDL default`.
   - In the "Permissions for Default" dialog box, grant the service account Read and Execute access.

vCenter Server

VMware vCenter Server provides a centralized management platform for VMware hypervisors. To manage your VMware environment with Turbonomic, you specify a vCenter Server instance as a target. Turbonomic discovers the associated infrastructure and performs intelligent workload management.

Prerequisites

- A user account Turbonomic can use to connect to your vCenter and execute actions (see Creating a User Account In vCenter *(on page 20)*).

General Considerations Before Target Addition

- **Linked vCenters**:

  For linked vCenters, you must add each vCenter separately. Turbonomic communicates with each vCenter through a separate API endpoint, but aggregates the data from those vCenters and makes cross-vCenter actions possible.
- **Restricting Turbonomic Access to Specific Clusters**:

  When you add a vCenter target, Turbonomic discovers all of the connected entities that are visible based on the account used to connect to the vCenter target. If there are clusters or other entities you want to exclude, you can do this by setting the role for the Turbonomic account to `No access` in the vSphere management client.
- **Shared Datastores**:

  If you add more than one vCenter target that manages the same datastore, you may see conflicts in the metadata Turbonomic maintains for each vCenter. For example, datastore browsing may display a conflict between active and wasted files, or each vCenter may define the same datastore as a member of a different storage cluster. Turbonomic recommends that you do not add multiple vCenter targets that manage the same datastore.
- **VSAN Permissions**:

  In order to enable VSAN support and discover groups based on storage profiles, you must ensure that the user role Turbonomic is assigned has the `Profile-driven storage view` permission enabled. This permission is *disabled* in the built-in `readonly` role.

Adding vCenter Targets

To add vCenter targets, select the **Hypervisors > vCenter** option on the Target Configuration page and provide the following information:

- Address

  The name or IP address of the vCenter server.
- Username/Password

  Credentials for the user account Turbonomic can use to connect to the vCenter Server. Include the domain if required (`<domain>\<username>`).
- Enable Datastore Browsing

  Enable datastore browsing so that Turbonomic can discover wasted storage.
- Enable Guest Metrics

  Enable this to collect advanced Guest memory metrics. This can increase the accuracy of the VMEM data that Turbonomic uses for analysis of VMs.

  Note that VMware Tools must be installed and running on the affected VMs. The **Hypervisor VMEM for Resize** option must also be turned on for the VMs discovered by the affected target. You make this setting in the Actions

section of the automation policy for the given scope of VMs. This is turned on by default. For more information, see "Hypervisor VMEM for Resize" in the *User Guide*.

> **NOTE:**
> If your VMware hypervisors are running in a Nutanix environment, you must understand pinning a Nutanix Controller VM. For more information, see Pinning Controller VMs in Generic Hypervisor Mode *(on page 114)*.

vCenter Post-Addition Imports

In addition to discovering entities managed by the hypervisor, Turbonomic also imports a wide range of vSphere settings, such as Host and Storage DRS rules, annotations, Resource Pools, and DRS HA settings (See Other Information Imported From vCenter *(on page 21)*).

VMware vSphere 6.0 introduced the ability to move VMs between vCenters. If you enabled this feature in your VMware environment, you can configure Turbonomic to include cross vCenter vMotions in its recommendations.

To configure Turbonomic to support cross vCenter vMotion recommendations, you must create a Workload Placement Policy that merges the datacenters on the different vCenters, and then another policy to merge the given clusters. Also note that the merged clusters must use the same network names in the different datacenters. To create a Merge Policy:

1. In the Policy Management Tab, select **Placement Policy**.
2. For `policy type`, select **Merge**.
3. For `MERGE`, choose the merge type, and click **Select**.

   To merge datacenters choose Datacenter, to merge Host clusters choose Cluster, and for storage choose StorageCluster.
4. Choose the specific datacenters or clusters to merge in this policy, then click **Select**.
5. Click **Save Policy**.

> **NOTE:**
> Since Turbonomic can only execute vMotions between clusters that use the same switch type (VSS or VDS), make sure any clusters you merge use the same switch type. Although Turbonomic will not initiate VSS → VDS vMotions, vSphere may do so. If this happens, Turbonomic displays a compliance violation notification.

Supported Actions

For each discovered entity within the hypervisor supply chain, Operations Manager can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations Only
Virtual Machine	Start, Move, Suspend, Storage Move, Resize Down, Resize Up	Terminate, Provision, Reconfigure
Physical Machine	Start, Suspend	Terminate, Provision
Storage		Provision

Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB)

Entity Type	Commodity
	• CPU Ready The utilization of the PM's allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Datacenter	**NOTE:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the physical machines in the datacenter. • Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared of memory among VMs running on the host. ESX-only Measured in Kilobytes (KB)

Entity Type	Commodity
	• CPU Ready The utilization of the PM's allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Kilobytes (KB)
Provider Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Provider vDC. Measured in Kilobytes (KB)
Consumer Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Consumer vDC. Measured in Kilobytes (KB)

Creating A Service User Account In vCenter

The service account you use must have specific permissions on the vCenter. The easiest way to grant Turbonomic the access it requires is to grant full administrator rights.

Some enterprises require that the service account does not grant full administrator rights. In that case, you can create a restricted service account that grants the following permissions to enable the required Turbonomic activities:

Turbonomic Activity	Required Privileges
Monitoring	Read-only for all entity types
Recommend Actions	Read-only for all entity types
Wasted Storage Reporting	Datastore > Browse Datastore
Execute VM Move (vMotion)	Resource > Migrate Resource > Query Vmotion

Turbonomic Activity	Required Privileges
	Resource > Modify Resource Pool Resource > Assign VM to Resource Pool
Execute VM Storage Move (svMotion)	Datastore > Allocate Space Datastore > Browse Datastore Resource > Assign VM to Resource Pool Resource > Migrate Resource > Modify Resource Pool Resource > Move Resource Pool Resource > Query VMotion Virtual Machine > Configuration > Change Resource Virtual Machine > Configuration > Swap File Placement
Execute VM Resize	Virtual Machine > Configuration > Change CPU Count Virtual Machine > Configuration > Change Resources Virtual Machine > Configuration > Memory Virtual Machine > Interaction > Reset Virtual Machine > Interaction > Power Off Virtual Machine > Interaction > Power On
Discover tags	Global > Global tags In addition, you must open ports 10443 and 7443 on the target server

Other Information Imported from vCenter

In addition to discovering entities managed by the vSphere hypervisors and their resources, Turbonomic:

- Imports any vSphere Host DRS rules when DRS is enabled, and displays them on the **Policy > Workload Placement** view under **Imported Placement Policies**. Imported rules are enabled by default, but you can disable specific rules if you want.

  > **NOTE:**
  > In vCenter environments, Turbonomic does not import DRS rules if DRS is disabled on the hypervisor. Further, if Turbonomic did import an enabled DRS rule, that somebody subsequently disables that DRS rule, then Turbonomic will discover that the rule was disabled and will remove the imported placement policy.
- Imports any custom annotations and displays related groupings in the **Inventory > Groups** tree view, under **VC Annotations**. The service account must enable the **Global > Global tag** privilege, and the target server must open ports 10443 and 7443.
- For vCenter Server versions 5.5 and later, discovers Virtual Machine Storage Profiles and displays them as groups anywhere that you can set scope. The groups appear under **VC Storage Profiles**. You can use these discovered storage profiles the same as any other groups — For example, to scope dashboards, or to set the scope for specific action policies.
- Discovers resource pools and displays them as folders in the Inventory tree and as components in the Supply Chain Navigator. If you have the Cloud Control Module license, Turbonomic manages resource pools as Virtual Datacenters

(VDCs) and can recommend resize actions. Root resource pools appear as Provider VDCs in the supply chain, whereas child resource pools appear as Consumer VDCs.

- Imports vSphere HA cluster settings and translates them into CPU and memory utilization constraints. These are displayed as cluster-level overrides under **Folders** on the **Policy > Analysis > Host** view.

Private Cloud

A private cloud manages resources in pools to support multi-tenancy and self-service provisioning of virtual workloads. Turbonomic manages these resource pools in real time as demand fluctuates. This includes demand across resource pools, virtual datacenters (VDCs), and tenants.

On the private cloud, you can use Turbonomic to:

- Set up charge-back and show-back for private cloud or service-provider scenarios
- For service-providers, set up scoped views to limit exposure to the customer base
- Plan hardware requirements — the planning scenarios takes cloud architectures into account

Supply Chain

For private clouds, Turbonomic discovers resource partitions that are managed by the cloud manager, as well as the workload running on these partitions (the VMs and applications) and, where applicable, the supply that hosts workload (the physical machines and storage). Turbonomic represents these partitions as the following types of Virtual Datacenters (VDCs):

- Provider VDC

  A collection of physical resources (PMs and datastores) within a private cloud. The cloud administrator has access to these resources, and defines the datacenter members. Administrators allocate Provider VDCs to manage resources that will be allocated to external customers through one or more Consumer VDCs.
- Consumer VDC

  A collection of resources that are available for customers to perform self-service management of workload through the cloud. It is an environment customers can use to store, deploy, and operate virtual systems. Consumer VDCs use the resources supplied by a Provider VDC.

Actions

Turbonomic recommends actions for private cloud infrastructures as follows:

Entity Type	Action
Provider Virtual Datacenters	Turbonomic does not recommend actions to perform on a Provider VDC. Instead, it recommends actions to perform on the devices running in the datacenter.
Consumer Virtual Datacenters	• Resize Consumer vDC • Provision Consumer vDC

Monitored Resources

Turbonomic monitors the following private cloud infrastructure resources:

Entity Type	Commodity
Provider Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Provider vDC. Measured in Kilobytes (KB)
Consumer Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Consumer vDC. Measured in Kilobytes (KB)

OpenStack

To manage your OpenStack® environment, Turbonomic connects to the Keystone identity service endpoint. Through this connection, Turbonomic discovers the other services it needs to control your OpenStack environment. If it discovers the necessary services, then it considers the OpenStack target to be validated.

Turbonomic has been tested with hypervisors running OpenStack on Red Hat Enterprise Linux and Red Hat RDO.

Prerequisites

Turbonomic supports OpenStack version Icehouse or later.

You must have the following services installed and enabled, and Turbonomic must have access to each service at its administrative endpoint:

Service	Purpose
Keystone	Discovery of OpenStack tenants
Nova	Management of cloud computing systems

Service	Purpose
Cinder	Management of block storage
Ceilometer	Telemetry
Gnocchi	Telemetry- extended Port: 8041

To verify that these services are enabled, Open the Horizon dashboard to the Admin/System page, or execute the following command at the OpenStack command line:

```
[root@openstack ~(keystone_admin)]# keystone service-list
```

You should see each of the required services in the list.

In addition, you must have the following meters enabled for Turbonomic to collect metrics for your OpenStack VMs. These meters are cumulative. For example, if you are running OpenStack Newton, you need the meters listed under Mitaka as well as those for Newton. For more information about each meter, please refer to the official OpenStack documentation:

Release	Meters
Mitaka or Earlier	• compute.node.cpu.percent • cpu_util • disk.device.usage • disk.device.read.requests.rate • disk.device.write.requests.rate • disk.read.bytes.rate • disk.write.bytes.rate • disk.read.requests.rate • disk.write.requests.rate • hardware.cpu.util • hardware.disk.size.used • hardware.memory.buffer • hardware.memory.cached • hardware.memory.used • hardware.network.incoming.bytes.rate • hardware.network.outgoing.bytes.rate • hardware.system_stats.io.incoming.blocks.rate • hardware.system_stats.io.outgoing.blocks.rate • memory.resident • memory.usage • network.incoming.byte • ns.rate • network.outgoing.bytes.rate
Newton	• cpu_l3_cache • memory.bandwidth.total • memory.bandwidth.local

Release	Meters
	• perf.cpu.cycles • perf.instructions • perf.cache.references • perf.cache.misses
Ocata	• network.incoming.packets.drop • network.outgoing.packets.drop • network.incoming.packets.error • network.outgoing.packets.error
Pike	• memory.swap.in • memory.swap.out
Queens	• disk.device.read.latency • network.outgoing.packets.drop • network.incoming.packets.error • network.outgoing.packets.error

Adding OpenStack Targets

To add OpenStack targets, select **Cloud Management > OpenStack** on the Target Configuration page, and provide the following information:

- Hostname or IP Address

  Provide the public URL of the Keystone service. The default port is 5000 — Do not provide a port if you want to use the default. For the default port, validation assumes a standard HTTP connection. If you provide a port value (for a port other than 5000), validation assumes a secure HTTPS connection unless you specify the protocol.

  For example, `10.10.123.45:5001` will use HTTPS, while `http://10.10.123.45:5775` will use HTTP.
- Username

  The account must have an administrator role on the specified tenant. This account must be authenticated by OpenStack.
- Password

  The password for the administrator account.
- Tenant Name

  The organizational structure within the Compute service that you want to manage. In a basic OpenStack installation this tenant is usually named `admin`.

Enabling Reservations

When you add a valid OpenStack target, Turbonomic can perform its analysis, recommend actions, and perform actions to assure performance and efficiency in your environment. To support these actions, you do not need to perform other configuration.

Turbonomic also includes reservations — deployment capabilities that can act as workload orchestration, or be integrated into an existing orchestration system (see the Deploy View in the Turbonomic user interface). These capabilities include:

- Calculate optimal placement for new workload
- Reserve resources for proposed workload, and include the reservations in real-time and planning analysis

An OpenStack user can use the Turbonomic placement proposals to deploy workload to the optimal locations.

To enable these capabilities, you must install the Turbonomic Nova Scheduler plugin that matches your version of OpenStack. Turbonomic delivers the following versions of this plugin on the Turbonomic Github repository:

- Icehouse

  https://raw.githubusercontent.com/turbonomic/nova/stable/icehouse/nova/scheduler/vmt_scheduler.py
- Juno

  https://raw.githubusercontent.com/turbonomic/nova/stable/juno/nova/scheduler/vmt_scheduler.py
- Kilo

  https://raw.githubusercontent.com/turbonomic/nova/blob/stable/kilo/nova/scheduler/vmt_scheduler.py
- Mitaka / Newton

  https://raw.githubusercontent.com/turbonomic/nova/stable/mitaka/nova/scheduler/vmt_scheduler.py

To fetch the scheduler plugin you want, execute the following commands on the Nova controller (substituting the url to the version of scheduler plugin that you want):

```
cd /usr/lib/python2.6/site-packages/nova/scheduler/

curl -O <URL TO CORRECT VERSION OF vmt_scheduler.py>
```

After you execute these commands to add the plugin to your controller, add the following entries to the file, `/etc/nova/nova.conf` under the `[DEFAULT]` section, where you provide the IP address of your Turbonomic server, and credentials for a Turbonomic user account that has administrator privileges:

```
scheduler_driver = nova.scheduler.vmt_scheduler.VMTScheduler

vmturbo_rest_uri = <VMTurbo_IPAddress>

vmturbo_username = <VMTurbo_UserName>

vmturbo_password = <VMTurbo_Password>
```

After you restart the Nova scheduler, it can use the plugin to communicate with your Turbonomic instance.

Collecting Physical Machine Metrics

To collect full metrics from physical machines in your environment and display them in Turbonomic, the following actions must be taken:

- `/etc/ceilometer/pipeline.yaml` must be modified to match the following snippets:

```
- name: meter_snmp
       interval: 600
       resources:
       - snmp://[Compute Node FQDN]
```

```
            - snmp://[Compute Node FQDN]
            - snmp://[Compute Node FQDN]

            meters:
            - "hardware.cpu*"
            - "hardware.memory.used"
            - "hardware.disk.size.used"
            sinks:
            - meter_sink
            - name: meter_snmp_rate
            interval: 600
            resources:
            - snmp://[Compute Node FQDN]
            - snmp://[Compute Node FQDN]
            - snmp://[Compute Node FQDN]
            meters:
            - "hardware.network*"
            - "hardware.system_stats*"
            sinks:
            - snmp_sink

     - name: snmp_sink
            transformers:
            - name: "rate_of_change"
            parameters:
            source:
            map_from:
            name: "hardware\\.(system_stats.io|network)\\.(incoming|outgoing)\\.(block
    s|bytes)"
            unit: "(blocks|B)"
            target:
            map_to:
            name: "hardware.\\1.\\2.\\3.rate"
            unit: "\\1/s"
            type: "gauge"
            publishers:
            - notifier://
```

- After modifying this file, you will need to create a `snmpd.conf` file. For information about creating this file, see Sample OpenStack SNMP Configuration File for PM Metric Collection *(on page 133)*.
- Upload the file you just created to `/etc/snmp`.
- Enable and start the SNMP service with the follow commands:
  - `systemctl enable snmpd.service`
  - `systemctl start snmpd.service`

After these modifications are made, OpenStack will collect all available metrics for physical machines.

Actions

Turbonomic recommends actions for the cloud target supply chain as follows.

Entity Type	Action
Virtual Machines	• Provision additional resources (VMem, VCPU)

Entity Type	Action
	• Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine
Physical Machines	• Start Physical Machine • Provision Physical Machine • Suspend Physical Machine
Consumer Virtual Datacenters	• Resize Consumer vDC • Provision Consumer vDC

Monitored Resources

Turbonomic monitors the following resources for the cloud target supply chain:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters

Entity Type	Commodity
	Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM's allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
Datacenter	**NOTE:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the physical machines in the datacenter. • Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared of memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready

Entity Type	Commodity
	The utilization of the PM's allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Kilobytes (KB)
Provider Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Provider vDC. Measured in Kilobytes (KB)
Consumer Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Consumer vDC. Measured in Kilobytes (KB)

Virtual Machine Manager

In a VMM environment, the VMM management server processes commands and controls communications with the Hyper-V hosts. To manage VMM, you set the management server as a target. Turbonomic communicates with that target, and also with the Hyper-V hosts that the VMM server manages. For this reason, if you add a VMM target, you should not also add individual Hyper-V hosts as targets. You must grant Turbonomic access to the VMM management server, and also to all the associated Hyper-V machines.

Prerequisites

- VMM Dynamic Optimization disabled (recommended) or set to Low aggressiveness, with a frequency of at least 60 minutes.
- Configure remote management on the VMM management Hyper-V server. Refer to Enabling Windows Remote Management *(on page 130)*
- Apply necessary hot fixes on the VMM host

For example, if you are running VMM Server on a Windows Server version earlier than Windows Server 2012 R2, you must apply the hotfix referenced in the Microsoft Knowledge Base article #2842230 (http://support.microsoft.com/kb/2842230).

- Apply time synchronization

  The VM that hosts Turbonomic must be synchronized with each target VMM management server. The Turbonomic Installation Guide includes instructions for synchronizing the clock on the Turbonomic server.
- PowerShell execution must be enabled on the VMM management server.
- Configure port access

  WinRM uses ports 5985 and 5986 for standard and secure communications, respectively. The firewall on your VMM server must open these ports.
- Preexisting Hyper-V Targets

  If any of the hosts that make up the VMM target were added separately as Hyper-V targets (and you do not plan to exclude these hosts), these targets must be deleted. Failure to do so will create duplicate entities in the market, which will negatively impact Turbonomic performance.

Adding VMM Targets

Turbonomic uses the address and credentials you provide to discover the VMM target. From the VMM target, Turbonomic gets the list of managed Hyper-V instances. It then uses that list to discover each Hyper-V instance. The Hyper-V credentials you provide must be valid for all of these machines.

Turbonomic will also import your Availability Sets, representing them as placement policies for the affected infrastructure.

To add VMM targets, select the **Cloud Management > VMM** option on the Target Configuration page, and provide the following information:

- The IP address or host name of the VMM management server
- Which port to use for the WSMan connection

  For a standard connection (HTTP) use 5985. For a secure connection (HTTPS) use 5986.
- Enable or disable a secure connection

  If you enable a secure connection, then you must configure a certificate, and you must configure Turbonomic to communicate over HTTPS. For more information, see Secure Setup for WSMan *(on page 132)*.

  Note that setting a secure connection for VMM does not also set secure connections for the underlying Hyper-V hosts. Any communications between Turbonomic and VMM will be secure. To configure secure connections to the underlying Hyper-V hosts, you must specify secure connections on each one.
- Full domain name for the user account

  This domain name identifies the user account for Active Directory authentication.
- Login credentials for the Hyper-V servers that are managed by the VMM target (Optional)

  Turbonomic must log into the Hyper-V servers that the VMM server manages. If you leave the Hyper-V credentials blank, then it will use the same credentials that VMM uses. If you provide Hyper-V credentials, then it will use that service account to log into every Hyper-V managed by the VMM.

  Note that the service account Turbonomic uses to log into a Hyper-V host must satisfy certain requirements. For more information, see Creating a Service User Account in Hyper-V *(on page 15)*.
- Hosts in the VMM target that you would like to exclude from Turbonomic.

Turbonomic will not discover or manage any hosts you enter in this field. These hosts do not participate in the market, and Turbonomic does not generate actions that involve these hosts. Enter multiple hosts in a comma-separated list. This field accepts both host names, and FQDNs (Fully Qualified Domain Names), and allows the use of the wildcard characters * and ?.

Microsoft VMM and Hyper-V VLANs

Windows Server Hyper-V provides support for VLANs on host and VM partitions. If your Hyper-V environment makes use of this VLAN support, then your VM moves must be sensitive to which hosts provide networking access to your defined VM networks. If a VM is a member of a given VM network, then any move of that VM must be to a host that has access to the same network.

For Hyper-V targets in a VMM environment, the Cloud Control Module is aware of the VM networks, and ensures that a move is to a host that provides connectivity over the given VM network. For example, if Host 1 and 2 provides connectivity to VM Network A, and Host 3 and 4 provide connectivity to VM Network B, Turbonomic will never recommend a VM with access to Network A, residing on Host 1, to move to Host 3. That would render the VM unable to communicate on Network A.

Configuring SMB 3.0 File Shares Discovery

With VMM, Turbonomic can discover SMB 3.0 shares as datastores, assuming these shares have been properly added to your VMM service center. When you add shares to your VMM environment, be sure to:

- Use the Fully Qualified Domain Name of the file server

  As you the associated file server to your VMM environment (via the Add Storage Devices Wizard), be sure to specify the FQDN of the file server on the Specify Discovery Scope page of the wizard. Do not use the file server's IP address.
- Ensure that file server names are unique

  Do not specify file servers with the same name, even if they belong to different domains. Turbonomic requires the file server names to be unique.

For information about setting up SMB 3.0 shares, please refer to your Microsoft documentation. For example, see "How to Assign SMB 3.0 File Shares to Hyper-V Hosts and Clusters in VMM".

Actions

Turbonomic recommends actions for the cloud target supply chain as follows.

Entity Type	Action
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine
Physical Machines	• Start Physical Machine • Provision Physical Machine • Suspend Physical Machine
Consumer Virtual Datacenters	• Resize Consumer vDC

Entity Type	Action
	• Provision Consumer vDC

Monitored Resources

Turbonomic monitors the following resources for the cloud target supply chain:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon

Entity Type	Commodity
	The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
Datacenter	**NOTE:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the physical machines in the datacenter. • Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared of memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Kilobytes (KB)
Provider Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz)

Entity Type	Commodity
	• Storage The utilization of the storage attached to the Provider vDC. Measured in Kilobytes (KB)
Consumer Virtual Datacenter	• Memory (Mem) The utilization of the Datacenter's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the Datacenter's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the Consumer vDC. Measured in Kilobytes (KB)

Public Cloud

The public cloud provides compute, storage, and other resources on demand. You can run all of your infrastructure on a public cloud, or you can set up a hybrid environment where you burst workload to the public cloud as needed. Turbonomic can analyze the performance of applications running on the public cloud, and provision more instances as demand requires. For a hybrid environment, Turbonomic can provision copies of your application VMs on the public cloud to satisfy spikes in demand, and as demand falls off it can suspend those VMs if they're no longer needed.

With public cloud targets, you can use Turbonomic to:

- Scale VMs and Databases
- Change storage tiers
- Purchase VM Reservations
- Locate the most efficient workload placement within the hybrid environment, while assuring performance
- Detect unused storage volumes

Cloud-based datacenters support scalability, resource pooling, multi-tenancy, and self-service management of virtual resources. Turbonomic supports the following cloud technologies:

Supply Chain

For public clouds, Turbonomic discovers Regions and Zones, and adds them as Datacenter Entities. Regions and zones divide the public cloud into managed subsets. A region is typically associated with the geographic location of the cloud resources, and a zone is some division within the region. One region contains multiple zones.

Amazon Web Services

Amazon Web Services (AWS) provides a reliable and scalable infrastructure platform in the cloud. You gain access to this infrastructure through a subscription account with the appropriate organization API permissions. To specify an AWS target, you provide the credentials for that account and Turbonomic discovers the resources available to you through that account.

In order to discover RI utilization, you must provide Turbonomic with access to the S3 bucket that contains the AWS Cost and Usage report. Without this access, Turbonomic's purchase and scale decisions will be made without consideration of this data.

Adding AWS Targets

For Turbonomic to manage an AWS account, you provide the credentials via access key or IAM role that you use to access that account. For information about getting an Access Key for an AWS account, see the Amazon Web Services documentation.

The AWS Target has different target addition requirements based on connection via key or IAM role.

To add an AWS target *without* an IAM Role, specify the following:

- Custom Target Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- Access Key

  Provide the **Access Key** for the account you want to manage.
- Secret Access Key

  Provide the **Access Key Secret** for the account you want to manage.
- Proxy Host

  The IP of your Proxy Host
- Proxy Port

  The port required for the proxy above
- Proxy User

  The username required for the proxy above
- Proxy Password

  The password required for the proxy above

To add an AWS target *with* an IAM Role, specify the following:

- Custom Target Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- IAM Role ARN

  Provide the **ARN** for the IAM role used to access the AWS billing information.
- Proxy Host

  The address of the proxy used for this target. Only fill out proxy information if you connect to the Dynatrace server via a proxy.
- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

  The username for the account to log into the proxy specified above.

- Proxy Password

  The password to use with the proxy specified above.

Firewall and Proxy Access

To use AWS with a proxy or firewall, it must be configured to allow unrestricted access to the following URLs:

Functionality	AWS Endpoint
Price List	`api.pricing.us-east-1.amazonaws.com`
AWS Bill	`{bucket-name}.s3.{region-containing-report}.amazonaws.com`
CloudWatch	`monitoring.{region-id}.amazonaws.com`
CloudWatch Events	`events.{region-id}.amazonaws.com`
CloudWatch Logs	`logs.{region-id}.amazonaws.com`
EC2	`ec2.{region-id}.amazonaws.com`
Elastic Load Balancing	`elasticloadbalancing.{region-id}.amazonaws.com`
IAM	`iam.amazonaws.com`
Organizations	`organizations.{region-id}.amazonaws.com`
Relational Database Service (RDS)	`rds.{region-id}.amazonaws.com`
Resource Groups	`resource-groups.{region-id}.amazonaws.com`
Service Catalog	`servicecatalog.{region-id}.amazonaws.com`
S3	`s3.{region-id}.amazonaws.com`
Storage Gateway	`storagegateway.{region-id}.amazonaws.com`

Whitelisting AWS Regions

While Turbonomic is discovering your AWS environment, if it fails to reach one or more AWS regions, then AWS discovery will fail for that target.

There may be policy decisions that prevent Turbonomic from reaching all AWS regions. For example, if you operate Turbonomic behind a firewall, you might not be able to reach all the regions that are available to your AWS account. In that case, you need to specify which regions you want Turbonomic to discover.

For information about how to specify the regions that you want Turbonomic to discover, contact your support representative.

Cost and Usage Report

In order for Turbonomic to display month-to-day spend, you must create a cost and usage report in AWS and store it in an S3 bucket.

For more information, see Displaying AWS Spend In Turbonomic.

Enabling Collection of Memory Statistics

For Turbonomic to collect memory statistics in AWS, you must set up CloudWatch to enable the collection of these statistics on the VMs in your environment. For more information, see Enabling Collection of Memory Statistics: AWS *(on page 126)*.

IAM Authentication

To connect to an AWS target via IAM, Turbonomic requires an IAM User for authentication. This user must have the following permissions:

> **NOTE:**
> Turbonomic recommends that you set up IAM access via an IAM group that has the necessary permissions. After you create this group, create a user that is a member of it and specify that user for Turbonomic to access your AWS environment. For more information on IAM Groups, see `http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html`.

The Turbonomic user account must have the following permissions.

For read access:

Use these permissions to gain insight into your environment, including utilization statistics and costs for workloads and services. Turbonomic can recommend actions, but this account cannot execute them.

- AmazonRDSReadOnlyAccess
- AmazonEC2ReadOnlyAccess
- AmazonS3ReadOnlyAccess
- organizations:List*
- organizations:Read*

For write access (Action Execution):

Use these permissions to automate actions based on Turbonomic analysis.

- AmazonRDSFullAccess
- AmazonEC2FullAccess
- AmazonS3ReadOnlyAccess
- kms:CreateGrant*
- organizations:List*
- organizations:Read*

Turbonomic also supports logging in to AWS targets via IAM Roles. To enable using Roles, you must run the Turbonomic software on an EC2 instance in the AWS cloud, and you must have the Turbonomic instance run as the IAM Role, and connect to the AWS target accounts with the appropriate IAM Role. To perform these actions, see https://greencircle.vmturbo.com/docs/DOC-5593-role-based-target-access. For troubleshooting, contact Turbonomic Technical Support.

Actions

Turbonomic recommends actions for the cloud target supply chain as follows.

Entity Type	Action
Virtual Machines (Cloud)	• Scale up to template (based on VMem / VCPU) • Scale down to template (based on VMem / VCPU) • Move Virtual Machine (intra-cloud) **NOTE:** This is a destructive move- data / applications are not preserved. This action also requires both a Turbonomic merge policy and the moved VM to be a Linux VM with template configuration.
Zone	• Start VM on the Zone • Suspend VM on the Zone

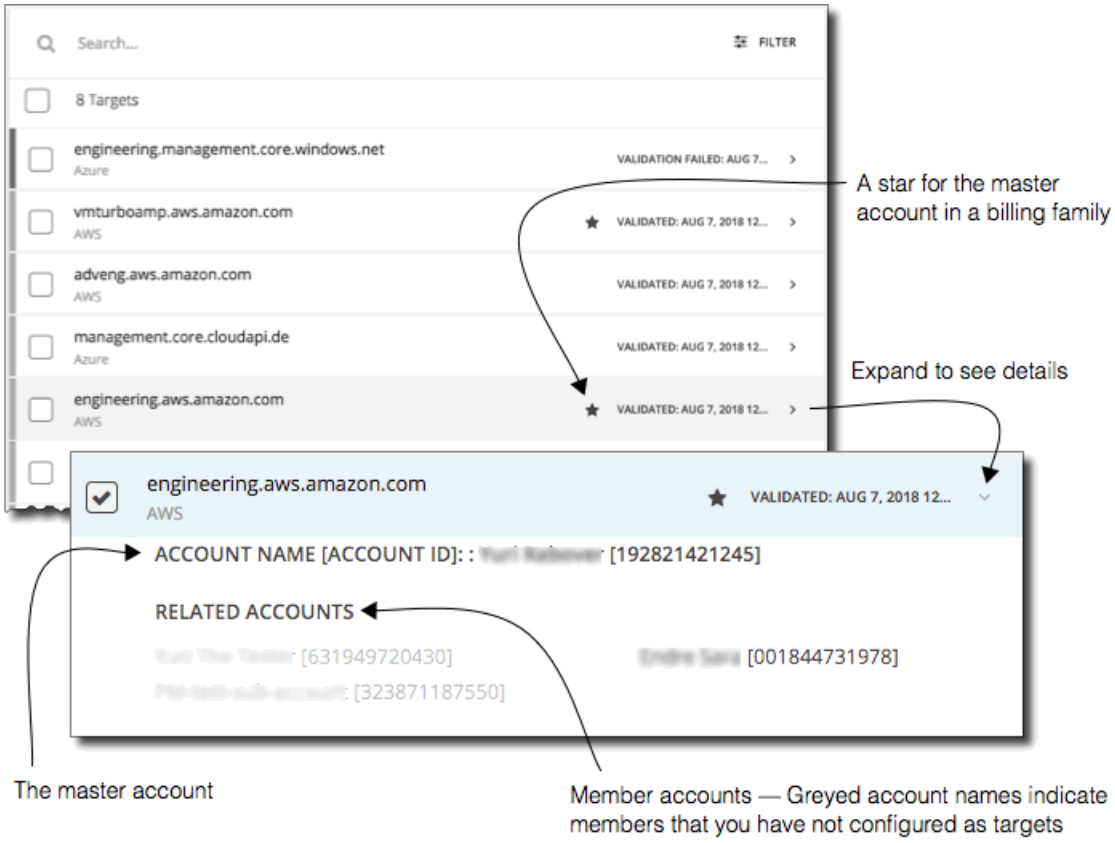
Monitored Resources

Turbonomic monitors the following resources for the cloud target supply chain:

Entity Type	Commodity
Virtual Machine (Cloud)	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Net Throughput Rate of message delivery over a port Measured in KB/s • I/O Throughput The throughput to the underlying storage for the entity Measured in KB/s • Latency The utilization of latency allocated for the VStorage on the VM

Entity Type	Commodity
	Measured in milliseconds (ms)
Database Server	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • I/O Throughput The throughput to the underlying storage for the entity Measured in KB/s • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)

AWS Billing Families



As you configure AWS targets, Turbonomic discovers AWS accounts that are consolidated into *billing families*. A billing family has one *master* account, and zero or more *member* accounts. By recognizing billing families, Turbonomic more accurately calculates cloud investments and savings, and makes more accurate recommendations for RI coverage.

In the Targets user interface, master accounts appear in bold, with a star next to them. You can expand the account entry to see the related member accounts. If you expand the entry for a member account, then the related accounts includes the family master, indicated by a star.

For RI purchases, different accounts in a billing family can share the same RI resources. At the same time, accounts in other billing families cannot use those RIs. This adds flexibility to your RI coverage, while maintaining order over the billing.

In Turbonomic, if you enable Billing Family Recognition, then you can see the billing family master and member accounts in the Targets user interface, and Turbonomic can recommend proper RI purchases within the correct billing families.

To enable Billing Family Recognition, ensure the following as you configure your AWS targets:

- Use the proper role for each AWS target

  To properly discover billing family information for a target, you must give Turbonomic credentials for an AWS role that includes the permission, `organizations:DescribeOrganization`. With that permission, Turbonomic can:

  - Discover master accounts and member accounts in different billing families
  - Display the account names in the user interface
  - Discover billing information for each family and account
  - Recommend RI actions that respect billing family boundaries

- Configure targets for the complete billing family

  One billing family can consolidate a number of AWS accounts. For Turbonomic to include these accounts in its analysis, you must configure each one as a separate target. If you do not configure all the accounts in a billing family, then Turbonomic cannot discover complete billing information for that family, and its analysis will be based on incomplete information.

  Turbonomic displays member accounts that have been configured as targets in regular text. For members that Turbonomic discovers but have not been configured as targets, Turbonomic displays their names in grayed text.

If you have enabled Billing Family Recognition, you should keep the following points in mind:

- Billing families can grow

  Turbonomic regularly checks the membership of your billing families. If it discovers a new member account, it adds that account to the list of members. If you have already configured the account as a target, then Turbonomic includes the new member in its analysis of billing families. If the new member is not already a target, then Turbonomic lists the new member in grayed text.
- You can configure discounts per billing family

  Turbonomic includes a feature to set a discount for a billing group, and to override that discount for specific template families within that scope. For more information, see "Cloud Discounts" in the *User Guide* and "Discount Override: AWS" in the *User Guide*.
- You might see master accounts that have no member accounts

  AWS treats every account you create as a part of a billing family. Assume you created an account, but you had no reason to consolidate its billing with any other accounts. In that case, the account appears in the Turbonomic user interface as a master account, but it has no member accounts.

AWS Billing

The Turbonomic AWS Billing target allows users to grant access to a bill which is used to discover billing family relationships. It does not provide access to any operational concern of an AWS account.

Adding an AWS Billing Target

The AWS Billing Target has different target addition requirements based on connection via key or IAM role.

To add an AWS Billing target *without* an IAM Role, specify the following:

- Custom Target Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- Access Key

  Provide the **Access Key** for the account you want to manage.
- Secret Access Key

  Provide the **Access Key Secret** for the account you want to manage.
- Cost and Usage Report Bucket

  Name of the S3 bucket that contains the AWS Cost and Usage report.
- Cost and Usage Report Path

  Path in the S3 bucket to the AWS Cost and Usage report.

- Cost and Usage Report Region

  Region of the S3 bucket that contains the AWS Cost and Usage report.
- Proxy Host

  The address of the proxy used for this target. Only fill out proxy information if you connect to the Dynatrace server via a proxy.
- Proxy Port

  The port to use with the proxy specified above. By default, this is 8080.
- Proxy Username

  The username for the account to log into the proxy specified above.
- Proxy Password

  The password to use with the proxy specified above.

To add an AWS Billing target *with* an IAM Role, specify the following:

- Custom Target Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- IAM Role ARN

  Provide the **ARN** for the IAM role used to access the AWS billing information.
- Cost and Usage Report Bucket

  Name of the S3 bucket that contains the AWS Cost and Usage report.
- Cost and Usage Report Path

  Path in the S3 bucket to the AWS Cost and Usage report.
- Cost and Usage Report Region

  Region of the S3 bucket that contains the AWS Cost and Usage report.
- Proxy Host

  The address of the proxy used for this target. Only fill out proxy information if you connect to the Dynatrace server via a proxy.
- Proxy Port

  The port to use with the proxy specified above. By default, this is 8080.
- Proxy Username

  The username for the account to log into the proxy specified above.
- Proxy Password

  The password to use with the proxy specified above.

Actions

Turbonomic does not recommend actions for AWS Billing targets. However, the billing information will be used in conjunction with the AWS target to make informed decisions.

Monitored Resources

Turbonomic does not monitor resources for AWS Billing targets. However, the billing information will be used in conjunction with the AWS target to make informed decisions.

Microsoft Azure

Microsoft Azure is Microsoft's infrastructure platform for the public cloud. You gain access to this infrastructure through a service principal target. To specify an Azure target, you provide the credentials for that account and Turbonomic discovers the resources available to you through that account.

Azure service principal targets will automatically discover the subscriptions to which the service principal has been granted access in the Azure portal. This in turn will create a derived target for each subscription that inherits the authorization provided by the service principal (e.g. contributor). Derived subscription targets are not directly modifiable but otherwise behave like any other target that may be validated and the inventory discovered.

Prerequisites

- External access via App Registration
- Azure Resource Manager

> **NOTE:**
> Turbonomic will not discover Azure Classic virtual machines, as they do not utilize the Azure Resource Manager.
>
> You must allow at least 30 minutes of discovery time for Turbonomic to allow full discovery of the Resource Group information.

> **NOTE:**
> In Azure environments, when you first configure an Azure target, under some circumstances the target has `No Quotas Available`, and so Turbonomic cannot discover the available templates. This can happen when you initially set up the Azure account and you have not enabled any providers. If this occurs, you can install a single VM in your cloud subscription to make quotas available. Or you can navigate to the Azure Subscriptions Blade and select the subscription you want. Then for the resource providers, register the `Microsoft.Compute` option. For more information, see the following Microsoft article: Resolve errors for resource provider registration.

External Access via App registration

The administrator of an Azure Active Directory (Tenant) can register an application with the tenant — This app registration gives an external application access to the tenant's resources. Turbonomic connects to an Azure target via an App registration.

To create an App registration in your tenant:

- Log into the Azure Management Portal
- Add an App registration to an available tenant — The tenant ID will correspond with the tenant ID that you set for the target.
- From the newly-created App registration blace in the Management Portal, go to the Overview blade

  This blade displays the generated Directory (tenant) ID and Application (client) ID for this app.
- From the newly created App registration blade in the Management Portal, go to the Certificates and Secrets blade. This blade displays previously created certificates and client secrets, as well as provides the ability to create them.

> **NOTE:**
> The administrator who creates the App registration must also create a Client secret key. This secret key must be recorded on creation. The administrator can return to the Management Portal to see the Application (client) ID, but the portal only shows the Client secret key once during creation.

For more information, refer to Microsoft's article, How to: Use the portal to create an Azure AD application and service principal that can access resources.

Accessing Reservations

To manage the use of Azure Reservations, the App registration for this target must have permissions to manage the reservations. In most cases, `Reader` permissions are sufficient. In the case of reservations that are scoped to specific subscriptions, you must you must provide the app permissions to the reservation order, per the Microsoft article, Manage Reservations for Azure resources. Specifically, review the section titled, *Add or change users who can manage a reservation*.

Adding Azure Targets

To add Azure targets, select **Cloud Management > Azure** on the Target Configuration page, and provide the following information:

> **NOTE:**
> For information about how to get these credentials, see the Green Circle article, "Cloud Control: Azure Target Setup"

- Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- Tenant ID

  The ID of the tenant that contains subscriptions to be managed with Turbonomic.
- Client App ID

  The Client/App ID of the App Registration that gives Turbonomic access to resources in your Azure subscription.
- Client Secret Key

  The secret key for the App Registration.

Firewall and Proxy Access

To use Azure with a proxy or firewall, it must be configured to allow unrestricted access to the following URLs:

- `ratecard.azure-api.net`
- `management.azure.com`
- `login.microsoftonline.com`
- `*.core.windows.net`

Memory Statistics

For Turbonomic to collect memory statistics in Azure, you must enable the collection of these statistics on the VMs in your environment. You can do this as you deploy your VMs, or you can enable the counters after the fact on VMs you have already deployed. For more information, see Enabling Collection of Memory Statistics: Azure *(on page 129)*.

Actions

Turbonomic recommends actions for the cloud target supply chain as follows.

Entity Type	Action
Virtual Machines (Cloud)	• Scale up to template (based on VMem / VCPU) • Scale down to template (based on VMem / VCPU) • Move Virtual Machine (intra-cloud) **NOTE:** This is a destructive move- data / applications are not preserved. This action also requires both a Turbonomic merge policy and the moved VM to be a Linux VM with template configuration.

Monitored Resources

Turbonomic monitors the following resources for the cloud target supply chain:

Entity Type	Commodity
Virtual Machine (Cloud)	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Net Throughput Rate of message delivery over a port Measured in KB/s • I/O Throughput The throughput to the underlying storage for the entity Measured in KB/s • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Database Server	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM

Entity Type	Commodity
	Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • I/O Throughput The throughput to the underlying storage for the entity Measured in KB/s • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)

Microsoft Enterprise Agreement

You can configure Turbonomic to manage Azure subscriptions within the context of an Enterprise Agreement (EA). An EA target enables Turbonomic to use custom pricing and discover reserved instances. When you configure an EA target, Turbonomic uses that richer pricing information to calculate workload size and RI coverage for your Azure environment.

To enable Turbonomic management of Azure EA environments, you must configure both an EA target and at least one service principle target. For more information about service principal targets, see Adding Azure Targets *(on page 44)*.

Prerequisites

- Microsoft Azure EA access key
- You must ensure that **DA View Charges** and **AO View Charges** are both enabled in the EA Portal (located under **Manage**). If you just enabled these settings, it can take up to 24 hours for the changes to take effect. For more information, see Troubleshoot enterprise cost views in the Microsoft Azure documentation.

Adding Microsoft Enterprise Agreement Targets

To add a Microsoft Enterprise Agreement target, select the **Cloud Management > Microsoft Enterprise Agreement** option on the Target Configuration page and provide the following information:

- Target Name

  A user-friendly name that will identify the target
- Enrollment Number

  Enterprise Agreement enrollment number (found in your EA admin account at ea.azure.com)
- API Key

  The API Access Key for the Enterprise Agreement (found in your EA admin account at ea.azure.com)
- Proxy Host

The IP address of the proxy server used, if any

- Proxy Port

  The port number of the proxy server

- Proxy User

  The username of the proxy user used to authenticate

- Proxy Password

  The password of the proxy user used to authenticate

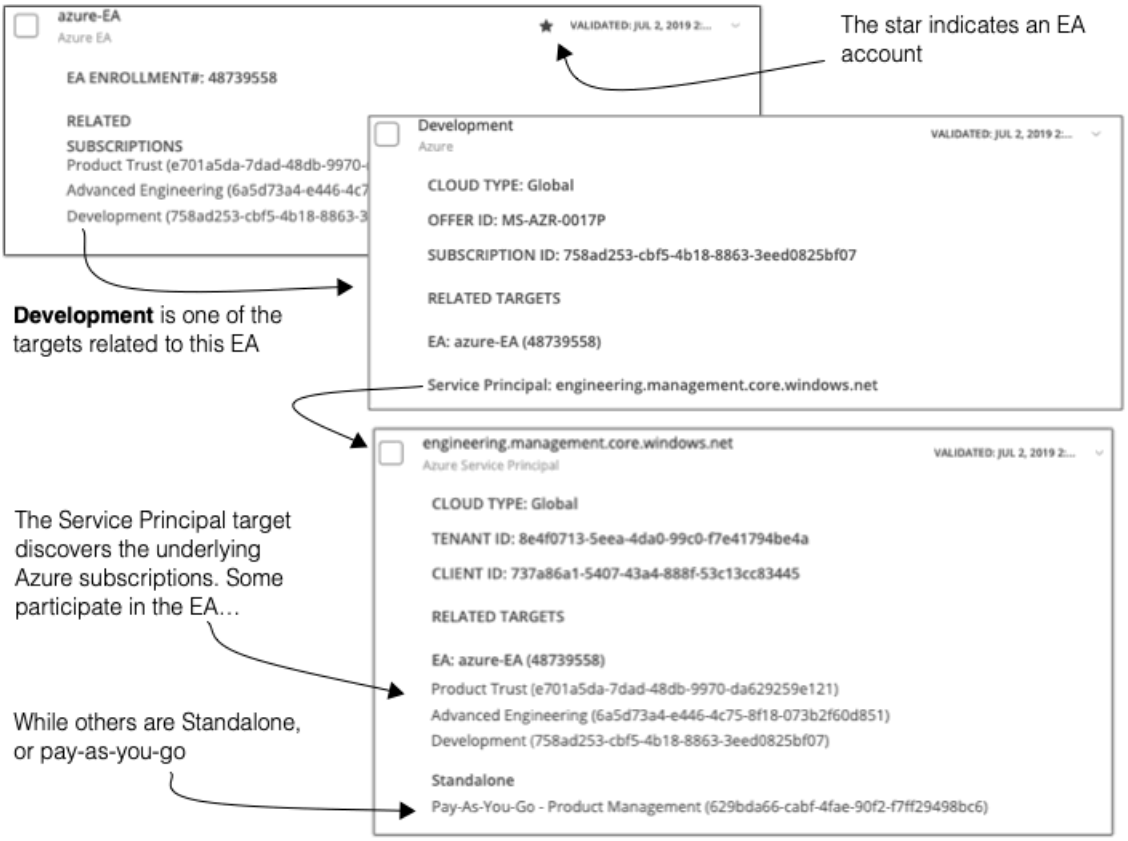
When you add the target and it validates, Turbonomic:

- Recognizes any existing Azure targets in your environment that are part of the EA
- Updates these targets with custom prices from the EA
- Discovers RIs in these targets

Note that this can take up to 24 hours, as target stitching occurs after the next bill processing cycle.

> **NOTE:** Turbonomic does not generate actions on the EA target specifically, but for the underlying service principal targets. For information about actions and monitored resources for Azure targets, see Adding Azure Targets *(on page 44)*.

Azure Enterprise Agreements



You can configure Turbonomic to manage Azure subscriptions within the context of an Enterprise Agreement (EA). An EA defines specific pricing, including the pricing for Reserved Instances (RIs). When you configure an EA target, and set the EA key to your Azure targets, Turbonomic uses that richer pricing information to calculate workload placement and RI coverage for your Azure environment.

To enable Turbonomic management of Azure EA environments, you must configure:

- One Microsoft Enterprise Agreement target
- At least one Service Principal target that can discover the underlying Azure subscriptions

For information about Azure targets, see Microsoft Azure *(on page 44)*.

In the Targets View, you can identify the targets related to Azure EA as follows:

- EA Targets

  The target that discovers the EA to track pricing and RI information. You can have one EA target per Turbonomic deployment. The EA target appears with a star next to the validation date. Expand the entry to see the EA enrollment number, and the Azure subscriptions that participate in the EA.
- Azure Subscription Targets

  The targets that manage the workloads in your Azure environment. These are discovered by Service Principal targets. Note that not all subscription targets *necessarily* participate in the EA. Expand these entries to see the related Service Principal target. For members of the EA, you can see the related EA target as well.

  Subscriptions that do not participate in the EA appear as Standalone targets.

  > **NOTE:**
  > In rare circumstances, you can have a subscription that is not in use – The subscription has no workloads associated with it. In this case, Turbonomic identifies the subscrition as Standalone. This is because the target cannot discover any cost or usage information that would relate the subscription to its EA.
- Service Principal Targets

  The Azure target that you configure to discover Azure subscription targets. Expand the entry to see the discovered targets. If you have configured an EA target, the entry lists that as well, along with the EA enrollment number.

Reserved Instances and Azure EA

For Azure envirnments, Turbonomic can only discover and use RIs if you have configured a Microsoft Enterprise Account target, and if one or more subscriptions participate in that EA.

To discover and manage RIs in Azure environments, Turbonomic uses both the EA target and the associated subscription targets. On its own, a subscription target exposes costs for pay-as-you-go pricing. The EA target discovers pricing for the available RI instance types. Turbonomic combines this information to track:

- RI utilization
- RI coverage
- Virtual machine costs (accounting for RIs)

> **NOTE:**
> This release of Turbonomic does not support RI discovery and management for Classic VMs and Classic Cloud Services. Also, it does not support RI discovery and management for Suppressed Core virtual machines.

Cost Calculations for Azure Environments

To understand the reported costs in your Azure environment, consider these points:

- For targets that participate in the EA, Turbonomic uses the terms of the given EA, and bases costs on the Offer ID that is effective for the given subscription.
- For VMs in Azure, RI pricing does not include the cost of the OS license. However pricing for on-demand VMs does include the license cost.

  > **NOTE:**
  > For Miscrosoft Azure EA environments, the projected cost for RI Purchase actions might not match associated costs you find in the Microsoft Pricing Calculator.
  >
  > Turbonomic actions can recommend RI purchases. For these recommendations, the action assumes a free Linux OS, so the cost estimate does not include the OS cost. However, The Microsoft Pricing Calculator does include costs for OS licenses. As a result, when you compare the Turbonomic cost estimates to the values in the Pricing Calculator, it's likely that the two estimates will not match. This difference also affects the Break Even Point that appears in the Recommended RI Purchases chart. Because the recommended purchases do not include Azure costs for OS licenses, the listed Break Even Point can be optimistic.
- For workloads you migrated from on-prem to the Azure cloud, Turbonomic recognizes Azure Hybrid Benefit (AHUB) savings for RIs and on-demand workloads. The costs you see in Turbonomic charts include this benefit. However, remember that recommended actions do not include any license cost, so the actions will not reflect any proposed AHUB savings (see above).
- Turbonomic supports rate cards for workloads that use the pricing set by a Cloud Solution Provider. Note that rate cards only apply to on-demand workloads, and they do not include RI pricing data. For more information, see " Azure Rate Cards" in the *User Guide*.

Storage Targets

Adding a storage Target enables Turbonomic to connect to your storage subsystem through an SMI-S provider API or through the controller's native API. Turbonomic uses the target's API to access information about each of the underlying disk arrays, and uses this information to set disk performance characteristics according to the type and capacity of storage. This leads to improved workload placement. Similarly, Turbonomic knows the relationships between storage controllers and disk arrays, and about the location of datastores within those arrays. This information also helps optimize workload placement.

The section below describes the storage supply chain. For information on how to add specific storage targets, the resources Turbonomic can monitor for the various supply chain entities, and the actions it can take to optimize the environment, refer to the target configuration instructions for your specific storage type.

Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities in turn host Storage entities (datastores).

Entity Mapping

Turbonomic Mapping	EMC VMAX	HPE 3Par	NetApp	Pure
Storage	Volume (Regular, Thin, Meta)	Virtual Volume	Volume	Volume
Disk Array	Disk Group or Thin Pool	CPG	Aggregate	Shelf Array
Storage Controller	VMAX Array	Controller	Controller / Filer	Controller

Actions

Turbonomic recommends actions for storage targets as follows.

> **NOTE:**
> This is a general list of actions for storage managed by storage controllers. Specific actions Turbonomic can recommend, and which actions it can automate depends on the actual technology — Not all actions make sense for all types of storage. For example, Turbonomic can automate a datastore move across disk arrays or storage controllers for NetApp in C mode, but not for other storage technologies.
>
> You can see how actions differ per technology in each section that describes adding a specific type of Storage Manager target.

Entity Type	Action
Storage	• Start Storage • Provision Storage • Suspend Storage • Move (only with Storage Targets configured) • Resize (only with Storage Targets configured)
Disk Arrays	• Provision Disk Array • Start Disk Array • Suspend Disk Array • Move Disk Array (for NetApp Cluster-Mode, only) • Move Virtual Machine • Move Datastore
Storage Controller	• Provision Storage Controller (recommendation only)

Monitored Resources

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned

Entity Type	Commodity
	The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Storage Controller	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • CPU The utilization of the Storage Controller's CPU in use Measured in Megahertz (MHz) • Storage Amount The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller Measured in Megabytes (MB)

EMC ScaleIO

EMC ScaleIO is an example of Software-Defined Storage for the datacenter. It creates a Virtual SAN overlaying commodity infrastructure that consists of multiple LAN-connected Servers with locally attached commodity Storage. It presents a standard Block Storage interface to Applications accessing the Virtual SAN.

Turbonomic communicates with the EMC ScaleIO system via the REST API Gateway.

Prerequisites

- EMC ScaleIO 2.x or 3.x
- A service account that Turbonomic can use to connect to the ScaleIO Gateway.

Adding EMC ScaleIO Targets

To add EMC ScaleIO targets, select the **Storage > EMC ScaleIO** option on the Target Configuration page and provide the following information:

- Address

  The IP or host name of the Gateway.
- Username

  The Username for the Gateway service account.
- Password

  The Password for the Gateway service account.

Entity Comparison

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in EMC ScaleIO to those used in Turbonomic:

EMC ScaleIO Name	Turbonomic Entity
Volume	Storage
Storage Pool	Disk Array
Protection Domain	Storage Controller

Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations only
Storage	Provision (Cloning)	Resize (Disabled by default)
Disk Array		Resize Disk Array
Protection Domain		Provision (Cloning)

Monitored Resources

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Storage Controller	• Storage Amount The utilization of the storage controller's capacity. Measured in Megabytes (MB)

EMC VMAX

Turbonomic supports management of VMAX2 and 3 Series storage arrays. The VMAX series is a family of enterprise storage arrays designed for SAN environments. Turbonomic connects to VMAX storage systems via an EMC SMI-S provider that has the disk arrays added to it. A single SMI-S provider can communicate with one or more disk arrays. When you specify an SMI-S provider as a target, Turbonomic discovers all the added disk arrays.

Turbonomic will create Storage Groups based on the SLO levels defined in VMAX3 Targets. By default, Storage Move actions will respect these SLO levels based on the configured response time.

Prerequisites

- EMC SMI-S Provider V8.x
- A service account that Turbonomic can use to connect to the EMC SMI-S Provider (typically the default `admin` account)

Adding VMAX Targets

To add VMAX targets, select the **Storage > VMAX** option on the Target Configuration page and provide the following information:

- Address

  The IP or host name of the SMI-S provider.
- Username

  The Username for the SMI-S provider.
- Password

  The Password for the SMI-S provider.

Entity Comparison

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in EMC VMAX to those used in Turbonomic:

EMC VMAX Name	Turbonomic Entity
Volume (Regular, Thin, Meta)	Storage
Storage Resource Pool (VMAX3) / Thick Provisioned Pool (earlier)	Disk Array
Storage Group (VMAX3) / Thin Provisioned Pool (earlier)	Logical Pool
VMAX Array	Storage Controller

Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations only
Storage	Provision (Cloning), Delete	Resize (V-Volumes only)
Logical Pool		Resize
Disk Array		Provision

Monitored Resources

When calculating available storage, Turbonomic excludes disks devoted to the VMAX operating system by default. If these disks are assigned to new raid groups or storage pools, the capacity of those disks will then be considered when calculating the capacity of the Storage Controller.

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Logical Pool	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the logical pool's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the logical pool's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the logical pool. Measured in Operations per second • Latency The utilization of latency on the logical pool. Measured in milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity.

Entity Type	Commodity
	Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Storage Controller	• Storage Amount The utilization of the storage controller's capacity. Measured in Megabytes (MB)

HPE 3PAR

HPE 3PAR StoreServ systems use controller nodes to manage pools of storage resources and present a single storage system to consumers. Turbonomic communicates with the HPE 3PAR system via both the WSAPI and SMI-S providers that are installed on the 3PAR controller node.

Prerequisites

- SMI-S Provider enabled and configured on the controller node.
- WSPAI Provider enabled and configured on the controller node.
- A service account on the controller node that Turbonomic can use to connect to the SMI-S and WSPAI providers.

> **NOTE:**
> For discovery and monitoring, the Turbonomic service account must have the `Browse` permission on all monitored domains. To exclude domains from monitoring, the Turbonomic service account must have no permissions on those domains. For action execution, Turbonomic requires the `Edit` permission.

Setting Up the SMI-S Provider

The HPE 3PAR SMI-S Provider should be installed on the controller node. It is disabled by default — you must ensure that it is installed properly and running on the controller node.

To enable the SMI-S provider:

1. Log into the HPE 3PAR Command Line Interface (CLI).

   Open a secure shell session (ssh) on the controller node. Default credentials are `3paradm/3pardata`.
2. Check the current status of the SMI-S provider.

   In the shell session, execute the command, `showcim`.

3. If the CIM service is not running, start it.

   Execute the command `startcim` to enable the CIM service and the SMI-S provider.

To stop the SMI-S provider, execute the command `stopcim -f -x`.

Setting Up the WSAPI Provider

The HPE 3PAR WSAPI Provider should be installed on the controller node.

To enable the WSAPI provider:

1. Log into the HPE 3PAR Command Line Interface (CLI).

   Open a secure shell session (ssh) on the controller node. Default credentials are `3paradm/3pardata`.
2. Check the current status of the WSAPI provider.

   In the shell session, execute the command, `showwsapi`.
3. If the WSAPI service is not running, start it by executing the command `startwsapi`.

   Execute the command `set wsapi -http enable` to allow only insecure connections, or `set wsapi -https enable` to allow only secure connections.

To stop the WSAPI provider, execute the command `stopwsapi -f`.

Adding HPE 3PAR Targets

To add an HPE 3PAR target, select the **Storage > HPE 3Par** option on the Target Configuration page and provide the following information:

- Address

  The name or IP address of the 3PAR controller node.

  By default, the controller provides SMI-S data over port 5988 (HTTP) or port 5989 (HTTPS). If your installation uses a different port for SMI-S, include the port number in the Address field.
- Username/Password

  Credentials for a user account on the controller node.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in HPE 3PAR to those used in Turbonomic:

HPE 3PAR Name	Turbonomic Entity
Virtual Volume	Storage
CPG	Disk Array
AO Configuration	Logical Pool
Controller	Storage Controller

Supply Chain

Storage targets (storage controllers) add Storage Controller, Logical Pool and Disk Array entities to the supply chain. Logical Pool and Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory Storage Supply Chain *(on page 50)*.

3Par Adaptive Optimization

Adaptive Optimization (AO) for HPE 3Par enables management of data storage across two or three tiers. AO places storage regions on the appropriate tier in response to periodic analysis that AO performs.

To work with the storage in an AO group, Turbonomic:

- Discovers each Common Provisioning Group (CPG) as a disk array

  In the Turbonomic user interface, these disk arrays do not host storage — They appear empty. Turbonomic will not recommend storage moves between these disk arrays, because such moves would conflict with AO block-level placement.
- Creates a single logical pool that hosts all the datastores in an AO group

  This logical pool represents the AO group, and it includes all the member CPGs. Turbonomic considers this single logical pool when it performs analysis — It can recommend moving storage into or out of the AO group. Also, Turbonomic aggregates resource capacity in this logical pool. For example, the IOPS capacity for the AO logical pool is a combination of IOPS capacity for the constituent CPGs.

You can see the AO logical pool in the Turbonomic user interface. The display name for this logical pool is the name of the AO Configuration.

Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations only
Storage	Provision, Resize Up/Down, Delete	Move
Disk Array	Provision, Resize Up/Down, Delete	
Logical Pool		Provision, Resize Up/Down, Delete
Storage Controller		Provision

Monitored Resources

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second

Entity Type	Commodity
	• Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Logical Pool	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the logical pool's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the logical pool's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the logical pool. Measured in Operations per second • Latency The utilization of latency on the logical pool. Measured in milliseconds (ms)
Storage Controller	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • CPU The utilization of the Storage Controller's CPU in use

Entity Type	Commodity
	Measured in Megahertz (MHz) • Storage Amount The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller Measured in Megabytes (MB)

NetApp

The Storage Control Module adds support for NetApp filers running the Data ONTAP operating system. NetApp storage controllers are Storage Virtual Machines that manage storage arrays — Vfilers for 7-Mode and Vservers for C-Mode. Turbonomic connects to these storage controllers to support NetApp targets in 7-Mode and Cluster-Mode (C-Mode).

Prerequisites

- Storage Control Module license
- Transport Layer Security (TLS) is enabled
- A service account Turbonomic can use to connect to the NetApp target

Enabling TLS

Starting with version 5.4, by default Turbonomic requires Transport Layer Security (TLS) version 1.2 to establish secure communications with targets. NetApp filers have TLS disabled by default, and the latest version they support is TLSv1. If your NetApp target fails to validate on Turbonomic 5.4 or later, this is probably the cause.

If target validation fails because of TLS support, you might see validation errors with the following strings:

- No appropriate protocol

  To correct this error, ensure that you have enabled the latest version of TLS that your target technology supports. If this does not resolve the issue, please contact Turbonomic Technical Support.
- Certificates does not conform to algorithm constraints

  To correct this error, refer to your NetApp documentation for instructions to generate a certification key with a length of 1024 or greater on your target server. If this does not resolve the issue, please contact Turbonomic Technical Support.

For information about enabling TLS, see the Data ONTAP **System Administration Guide** for sections on the SSL protocol.

Service User Account — Administrator Role

To discover and fully manage NetApp disk arrays, Turbonomic must have a service account that grants privileges to execute commands through the NetApp filer's OnTap API (ontapi). In most cases, you can provide a user account with Administrator privileges:

- **NetApp 7-Mode**: Create the administrator account from the NetApp command line — For example:

  ```
  useradmin user add Turbonomic -g Administrators
  ```
- **NetApp C-Mode**: Create the administrator account via the NetApp OnCommand System Manager, or from the NetApp command line — For example:

```
security login create -role admin -username Turbonomic -application ontapi -authmethod
password
```

If you prefer not to grant full administrator rights, see Creating Restricted Service Accounts In NetApp *(on page 64)*

Adding NetApp Targets

To add a NetApp target, select the **Storage > NetApp** option on the Target Configuration page and provide the following information:

- Address

  The name or IP address of the NetApp Storage Controller.

  7-Mode: Enter the storage controller address.

  Cluster-Mode (C-Mode): Enter the cluster management address.
- Username/Password

  Credentials for the NetApp service user account that you have configured for Turbonomic to use.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in NetApp to those used in Turbonomic:

NetApp Name	Turbonomic Entity
Volume	Storage
Aggregate	Disk Array
Controller / Filer	Storage Controller

Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory Storage Supply Chain *(on page 50)*.

Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations only
Storage	Move (C-Mode only)	Move (7-Mode), Provision, Resize Up
Disk Array		Resize Up, Move (C-Mode only), Provision (C-Mode only)
Storage Controller		Provision

Note that for NetApp in C-Mode, Turbonomic can automate moving a datastore to a disk array on the same storage controller, as well as moves to a disk array on a different storage controller.

Monitored Resources

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Storage Controller	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • CPU The utilization of the Storage Controller's CPU in use Measured in Megahertz (MHz) • Storage Amount

Entity Type	Commodity
	The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller Measured in Megabytes (MB)

Restricted Service Accounts In NetApp

While Turbonomic prefers a NetApp service account with administrator rights, it is possible to create an account that has limited access, by following the steps outlined below, depending on NetApp mode.

NetApp 9.x Restricted Service Account Setup

If you prefer to use a service account that does not have full administrator rights:

1. Log into the NetApp filer from a command shell.
2. Create a role and assign it permission to execute each of the following commands:

   For example:

   ```
   security login role create -role RoleName -cmddirname “storage aggregate show” -vserver Cluster-Name
   ```

   The required capabilities are listed below:

   - `cluster identity modify`
   - `cluster identity show`
   - `lun create`
   - `lun igroup create`
   - `lun igroup modify`
   - `lun igroup show`
   - `lun mapping create`
   - `lun mapping delete`
   - `lun mapping show`
   - `lun modify`
   - `lun show`
   - `network interface create`
   - `network interface delete`
   - `network interface modify`
   - `network interface show`
   - `statistics show`
   - `storage aggregate create`
   - `storage aggregate modify`
   - `storage aggregate show`
   - `storage disk show`
   - `system controller flash-cache show`
   - `system node modify`

- `system node show`
- `version`
- `volume create`
- `volume modify`
- `volume move modify`
- `volume move show`
- `volume move start`
- `volume qtree create`
- `volume qtree show`
- `volume show`
- `volume snapshot create`
- `volume snapshot modify`
- `volume snapshot show`
- `vserver create`
- `vserver fcp nodename`
- `vserver iscsi nodename`
- `vserver modify`
- `vserver options`
- `vserver show`

3. For execution privileges, execute the following commands for the given role, where `Role-Name` is the name of the role you are creating, and `Cluster-Name` identifies the cluster you want the role to affect. You must execute these commands individually to set privileges that affect each individual cluster:
   - `security login role create -role Role-Name -access all -cmddirname "volume offline" -vserver Cluster-Name`
   - `security login role create -role Role-Name -access all -cmddirname "volume unmount" -vserver Cluster-Name`
   - `security login role create -role Role-Name -access all -cmddirname "volume move" -vserver Cluster-Name`
   - `security login role create -role Role-Name -access all -cmddirname "volume delete" -vserver Cluster-Name`
4. Create a user that will use the newly-created role.

   For example:

   ```
   security login create -User-Name Turbonomic -r TurbonomicRole
   ```
5. Enter a password for the new user when prompted.
6. Give the user access to the `ssh` and `ontapi` applications by using the following commands, replacing `Role-Name` and `User-Name` with the role and user you created:

   ```
   security login create -role Role-Name -username User-Name -application ontapi -authmethod password
   ```

   ```
   security login create -role Role-Name -username User-Name -application ssh -authmethod password
   ```

NetApp 7-Mode Restricted Service Account Setup

If you prefer to use a service account that does not have full administrator rights:

1. Log into the NetApp filer from a command shell.
2. Create a role with API privileges.

   For example:

   `useradmin role add TurbonomicRole <capabilities>]`

   where <*capabilities*> is a comma-separated list of capabilities assigned to the role. The required capabilities are listed below:

   - `api-aggr-list-info`
   - `api-disk-list-info`
   - `api-fcp-node-get-name`
   - `api-flash-device-list-info`
   - `api-igroup-list-info`
   - `api-iscsi-node-get-name`
   - `api-lun-initiator-list-map-info`
   - `api-lun-map-list-info`
   - `api-lun-list-info`
   - `api-net-ifconfig-get`
   - `api-nfs-exportfs-list-rules-2`
   - `api-options-list-info`
   - `api-system-get-info`
   - `api-system-get-version`
   - `api-volume-list-info`
   - `api-snapshot-list-info`
   - `api-perf-object-get-instances`
   - `api-perf-object-instance-list-info`
   - `api-perf-object-counter-list-info`
   - `api-qtree-list`
   - `security-api-vfiler`
   - `api-vfiler-list-info`
   - `api-volume-options-list-info`
   - `login-http-admin`
   - `login-*`

   Note that the last login capability (`login-*`) may be necessary for external users.

   Execution capabilities:

   - `api-volume-create`
   - `api-volume-size`
   - `api-volume-offline`
   - `api-volume-online`
   - `api-volume-destroy`
   - `api-aggr-add`

- `api-aggr-create`
- `api-aggr-offline`
- `api-aggr-online`
- `api-aggr-destroy`

3. Create a group and assign the role.

   For example:

   ```
   useradmin add TurbonomicGroup -r TurbonomicRole
   ```

4. Create a user that is a member of the group.

   For example:

   ```
   useradmin user add Turbonomic -g TurbonomicGroup
   ```

5. Enter a password for the new user when prompted.

NetApp C-Mode Restricted Service Account Setup

If you prefer to use a service account that does not have full administrator rights:

1. Log into the NetApp filer from a command shell.
2. Create a role and assign it permission to execute each of the following commands:

   - `aggr-get-iter`
   - `igroup-get-iter`
   - `cluster-identity-get`
   - `lun-map-get-iter`
   - `net-interface-get-iter`
   - `storage-disk-get-iter`
   - `system-get-node-info-iter`
   - `volume-get-iter`
   - `vserver-get-iter`
   - `fcp-node-get-name`
   - `flash-device-get-iter`
   - `iscsi-node-get-name`
   - `options-list-info`
   - `qtree-list-iter`
   - `system-get-version`
   - `lun-get-iter`
   - `snapshot-get-iter`
   - `perf-object-get-instances`
   - `volume-get-iter`
   - `volume-move-get-iter`
   - `volume-move-start`

   For example, to enable volume offline, execute the following:

   ```
   security login role create -role TurbonomicRole -access all -cmddirname "volume
   offline" -vserver <cluster_name>
   ```

3. Create a user based on the role you create.

Give the user access to the `ssh` and `ontapi` applications. For example:

```
security login create -role TurbonomicRole -username Turbonomic -application ontap
i -authmethod password

security login create -role VMTurboRole -username VMTurbo -application ssh -authmet
hod password
```

Pure Storage

Turbonomic supports management of Pure Storage FlashArray systems. Note that one Pure Storage target manages a single Pure Storage FlashArray instance. The storage devices in the array are all flash storage — to analyze IOPS capacity, Turbonomic discovers your Pure storage controller model and maps the appropriate IOPS capacity to it.

Because of the improved performance of Pure Storage arrays, Turbonomic intelligently moves more demanding workloads to these datastores. Turbonomic analysis is also able to incorporate Pure Storage deduplication and compression when recommending actions.

Prerequisites

- A service account Turbonomic can use to connect to the FlashArray

  This account needs privileges to execute commands through the Pure Storage API — Typically the default `pureuser` administrative account.

Adding Pure Storage Targets

To add a Pure Storage target, select the **Storage > Pure Storage** option on the Target Configuration page and provide the following information:

- Address

  The name or IP address of the Pure Storage FlashArray.
- Username/Password

  Credentials for the service account Turbonomic can use to connect to the FlashArray. The Username must not contain the domain. For example, `Username=jjsmith` is correct, while `Username=myDomain\jjsmith` will result in a failure to validate.
- Secure connection

  When checked, uses SSL to connect to the Pure target. Most Pure installations do not accept insecure connections. If you receive an error when adding the target with secure connections disabled, try re-adding with this option enabled.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in Pure to those used in Turbonomic:

Pure Name	Turbonomic Entity
Volume	Storage
Shelf Array	Disk Array

Pure Name	Turbonomic Entity
Controller	Storage Controller

Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory Storage Supply Chain *(on page 50)*.

Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations only
Storage		Resize Up
Disk Array		
Storage Controller		Provision

Pure Storage assigns all the disks managed by a storage controller to a single array, with a fixed form-factor. There are no actions to perform for an array — For example, there is no action to move a disk array from one storage controller to another. Likewise, there are no actions to move or provision volumes because of the fixed form-factor.

Monitored Resources

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount

Entity Type	Commodity
	The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Storage Controller	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • CPU The utilization of the Storage Controller's CPU in use Measured in Megahertz (MHz) • Storage Amount The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller Measured in Megabytes (MB)

Application and Database Targets

Turbonomic supports the following Guest OS Process targets:

- SNMP (Simple Network Management Protocol)
- AppDynamics 4.1+
- AppInsights 4.1+
- DynaTrace 1.1+
- NewRelic 4.1+
- SNMP (Simple Network Management Protocol)
- WMI (Windows Management Instrumentation)

Supply Chain

Applications and Databases targets add Business Application, Business Transaction, Service, Application Component, and Database entities to the supply chain. You can navigate to the associated target page to see how these entities map to the target nomenclature.

AppDynamics

Turbonomic supports workload management of the application infrastructure monitored by AppDynamics, via adding the AppDynamics instance to Turbonomic as a target.

The Turbonomic integration with AppDynamics provides a full-stack view of your environment, from application to physical hardware. With information obtained from AppDynamics, Turbonomic is able to make recommendations and take actions to both assure performance and drive efficiency with the full knowledge of the demands of each individual application.

In its default configuration, the AppDynamics target will collect up to 1100 AppDynamics nodes within the default collection period when a proxy is used, and up to 5000 nodes when no proxy is required. Larger AppDynamics environments are expected to take longer than one cycle to collect complete data.

Prerequisites

- A valid AppDynamics user account.

  For all types of application instances, the service account must have the `Read Only User` role. For monitoring database instances, this user must also have the `DB Monitoring User` role.

  > **NOTE:**
  > In newer versions of AppDynamics where these roles are available, they should be used instead:
  > - Applications and Dashboards Viewer
  > - Dashboards Viewer
  > - DB Monitoring User
  > - Server Monitoring

AppDynamics Database Servers

AppDynamics also monitors database servers. In order for your database servers to be correctly stitched to the rest of your environment, you must:

- Enable enhanced metric collection.

  For Hyper-V hosts, you must install Hyper-V Integration Services on the target VM hosting the database. For more information, please refer to the following integration services TechNet article:

  `https://technet.microsoft.com/en-us/library/dn798297%28v=ws.11%29.aspx`

  For VMware hosts, you must install VMware Tools on the target VMs.
- Ensure that the database name in AppDynamics is resolvable to an IP address by the Turbonomic instance.

  You may need to make changes to your DNS or the file `/etc/resolv.conf` on the Turbonomic instance.

Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The following table describes nomenclature congruency between the target and Turbonomic:

AppDynamics	Turbonomic
Business Application	Business Application
Business Transaction	Business Transaction

AppDynamics	Turbonomic
Tier	Service
Node	Application Component
Database	Database Server
Server	Virtual Machine, Physical Machine

Adding an AppDynamics Target

> **NOTE:**
> It is possible to monitor certain applications or database servers with both AppDynamics and Turbonomic, but this must be avoided as it will cause the entities to appear duplicated in the market.
>
> If an application is monitored by AppDynamics, do not add it as a separate Turbonomic application target.

To add an AppDynamics instance as a target, specify:

- Hostname or IP Address

  The host name or IP Address of the AppDynamics controller instance.
- Port

  the port used to connect to the AppDynamics controler. By default, this is set to ports 80 (HTTP) and 443 (HTTPS).

  > **NOTE:** For SaaS-based AppDynamics instances, you must use port 443.
- Username@Tenant

  Username and account ID with the necessary role(s). The format must be *Username@**Tenant***, and the user must have the "Read Only User" and "DB Monitoring User" permissions. This username can be found on the "License > Account" page in AppDynamics. For oAuth authentication, the username must be a user defined as an API Client.

  > **NOTE:**
  > The username and password cannot contain any of the following special characters:
  >
  > `\ / " [ ] : | < > + = ; , ? * , ' tab space @`
- Password

  Password for the account used to connect to the AppDynamics instance. For oAuth, this will be the client secret key.

  > **NOTE:**
  > The username and password cannot contain any of the following special characters:
  >
  > `\ / " [ ] : | < > + = ; , ? * , ' tab space @`
- Secure Connection

  When checked, Turbonomic will connect via HTTPS. Make sure the required certificate is configured for use on the host.
- Proxy Host

  The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.
- Proxy Port

  The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

  The username to use with the proxy specified above.
- Proxy Password

  The password to use with the proxy specified above.

For more information about creating API client users, see the AppDynamics Documentation.

Actions

Turbonomic recommends actions for the AppDynamics supply chain as follows.

Entity Type	Action
Application / Application Server	• Resize Heap Recommendation only. • Resize Thread Pool Recommendation only. • Resize Connection Capacity Recommendation only. • Suspend VM Recommendation only. • Provision VM Recommendation only.

Monitored Resources

> **NOTE:**
> The exact resources monitored will differ based on application type. This list includes all resources you may see.

Turbonomic monitors the following resources for the AppDynamics supply chain:

Entity Type	Commodity
Business Transaction	• Transactions The utilization of the allocated transactions per second for the given virtual application Measured in transactions per second • Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms)
Business Applications	• Transactions The utilization of the allocated transactions per second for the given virtual application Measured in transactions per second • Response Time The utilization of the server's allocated response time

Entity Type	Commodity
	Measured in Milliseconds (ms)
Service	• Transactions The utilization of the allocated transactions per second for the given virtual application Measured in transactions per second • Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms)
Application Component	• Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) **NOTE:** This commodity is collected for Java and Node.js applications only. • Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) **NOTE:** This commodity is collected for Java and Node.js applications only. • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second • Heap The utilization of the application server's heap Measured in Kilobytes (KB) **NOTE:** This commodity is collected for Java applications only. • Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms) • Connections The utilization of the connection capacity. Only applicable to database servers Measured in Connections **NOTE:** This commodity is **NOT** collected for WebSphere applications. • Collection Time The percentage of server uptime spent garbage collecting. Available when the JVM profiler is enabled. Measured in percentage of uptime (%) **NOTE:** This commodity is **NOT** collected for WebSphere applications.

Entity Type	Commodity
	• Threads The utilization of the server's thread capacity Measured in number of Threads > **NOTE:** This commodity is **NOT** collected for WebSphere applications.
Database	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) > **NOTE:** This commodity requires a machine agent present, and database hardware monitoring to be enabled. • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) > **NOTE:** This commodity requires a machine agent present, and database hardware monitoring to be enabled. • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second • Heap The utilization of the application server's heap Measured in Kilobytes (KB) > **NOTE:** This commodity is collected for MS SQL and Mongo databases only. • DBMem The memory utilized by the database, as a of the memory capacity that is allocated to the database. Note that this resource is more accurate than the VMem resource on the hosting VM. With this resource, Turbonomic can drive resize and move actions based on the memory consumed by the database, not the memory consumed by the VM > **NOTE:** This commodity is collected for MS SQL and Mongo databases only. • Transactions The utilization of the allocated transactions per second for the given virtual application Measured in transactions per second • Connections The utilization of the connection capacity. Only applicable to database servers Measured in Connections > **NOTE:** This commodity is **NOT** collected for WebSphere applications. • Collection Time

Entity Type	Commodity
	The percentage of server uptime spent garbage collecting. Available when the JVM profiler is enabled. Measured in percentage of uptime (%) **NOTE:** This commodity is **NOT** collected for WebSphere applications. • Threads The utilization of the server's thread capacity Measured in number of Threads **NOTE:** This commodity is **NOT** collected for WebSphere applications. • Transaction Log The utilization of the server's capacity for storage devoted to transaction logs Measured in Kilobytes (KB) • DB Cache Hit Rate The percentage of accesses that result in cache hits. Measured as a percentage of hits vs total attempts (%)

Application Insights

Turbonomic supports workload management of the application infrastructure monitored by Application Insights via the Turbonomic integration, which provides a full-stack view of your environment, from application to hosting server. With information obtained from Application Insights, Turbonomic is able to make recommendations and take actions to both assure performance and drive efficiency with the full knowledge of the demands of each individual application.

Prerequisites

- A valid Application Insights user account with the same permissions detailed for Microsoft Azure *(on page 44)* targets.

Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The following table describes nomenclature congruency between the target and Turbonomic:

Application Insights	Turbonomic
Application	Application, Application Server

Adding an Application Insights Target

> **NOTE:**
> If an application is monitored by Application Insights or Azure, do not add it as a separate Turbonomic application target.

To add Application Insights as a target, specify:

- Unique Target Description

  A user-created name that will appear in the Turbonomic UI.
- Tenant Name

  The tenant associated to the Azure subscription associated to Application Insights.
- Azure Subscription ID

  The ID of the Azure subscription with access to the Azure target associated to Application Insights.
- Client App ID

  The Client/App ID of the App registration that gives Turbonomic access to resources in your Azure subscription.
- Client Secret Key

  The secret key for the App registration.
- Offer ID

  If applicable, the Azure Offer ID related to the Azure subscription.
- Enrollment Number

  If an Azure EA account, the enrollment number associated to the Azure subscription.

Stitching Application Insights Applications to the Turbonomic Environment

Each application monitored by Application Insights can have several associated applications, and each of these applications can be deployed on a different hosting server. Likewise, a hosting server may host multiple groups or partial groups of multiple applications.

In order to accurately stitch metrics from Application Insights, the host name or IP address of the hosting server must be discoverable through the Application Insights instance. For most monitored application instances, this is automatic. If the hosting server is not discoverable, Azure tags on the application can be provided to indicate the IP address or hostname.

You can provide an Azure tag in the following name : value format:

`Turbonomic-Host-Name : RoleInstance=hostname;RoleInstance=hostname;RoleInstance=hostname;`

In the preceding example, `RoleInstance` is the name of the application instance, and `hostname` is the hosting server. For example: `Turbonomic-Host-Name : cluster-app-a=120.120.120.10;cluster-app-b=120.120.120.11;cluster-app- c=120.120.120.12;`.

> **NOTE:** If these tags are provided, they will replace any values discovered through the Application Insights API.

Actions

Turbonomic recommends actions for the Application Insights supply chain as follows.

Entity Type	Action
Application / Application Server	• Resize Heap Recommendation only. • Resize Thread Pool Recommendation only. • Resize Connection Capacity

Entity Type	Action
	Recommendation only. • Suspend VM Recommendation only. • Provision VM Recommendation only.

Monitored Resources

> **NOTE:**
> The exact resources monitored will differ based on application type. This list includes all resources you may see.

Turbonomic monitors the following resources for the Application Insights supply chain:

Entity Type	Commodity
Application	• Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second • Heap The utilization of the application server's heap Measured in Kilobytes (KB) **NOTE:** This commodity is collected for Java applications only. • Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms) • SLA Commodity The number of requests made divided by the duration of the request.

New Relic

Turbonomic supports workload management of the application infrastructure monitored by New Relic via the Turbonomic integration, which provides a full-stack view of your environment, from application instance to host. With information obtained from New Relic, Turbonomic is able to make recommendations and take actions to both assure performance and drive efficiency with the full knowledge of the demands of each individual application.

Prerequisites

- A valid New Relic user account that includes either APM or infrastructure monitoring.

Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The following table describes nomenclature congruency between the target and Turbonomic:

New Relic	Turbonomic
APM: Key Transactions	Business Transaction
APM: Application	Service
APM: Application Instance	Application Component
-Infra: Database	Database Server
Infra: Host	Virtual Machine

Supported Applications

Turbonomic supports the following Application languages and associated commodities:

Application Language	Commodities
Java	Virtual CPU, Virtual Memory, Response Time, Transactions, Heap, Collection Time, Threads
.NET	Virtual CPU, Virtual Memory, Response Time, Transactions
Node.js	Virtual CPU, Virtual Memory, Response Time, Transactions, Heap, Collection Time
Python	Virtual CPU, Virtual Memory, Response Time, Transactions
Python	Virtual CPU, Virtual Memory, Response Time, Transactions
PHP	Virtual CPU, Virtual Memory, Response Time, Transactions
GO	Virtual CPU, Virtual Memory, Response Time, Transactions

Supported Databases

Turbonomic supports the following Database types and commodities:

> **NOTE:** Database commodities are exposed only if the New Relic account used to connect to Turbonomic has a `New Relic Infrastructure Pro` subscription.

Database	Commodities
MySQL	Cache Hit Rate
MS SQL	Cache Hit Rate, Virtual Memory, Transactions
OracleDB	Cache Hit Rate, Transactions, Response Time
MongoDB	Virtual Memory, Connections

Adding a New Relic Target

> **NOTE:**
> If an application is monitored by New Relic, do not add it as a separate Turbonomic application target.

To add New Relic as a target, specify:

- Account ID

  The New Relic Account ID.
- REST API Key

  The REST API Key *provided by the New Relic platform*. For more information, see Understand New Relic API Keys.
- GraphQL API Key

  The REST API Key *provided by the GraphQL service*. This is not identical to the `REST API Key` above. For more information, see Generate a new API key in the NerdGraph GraphiQL Explorer.
- EU Region

  If checked, Turbonomic will use the EU API endpoints.
- Proxy Host (Optional)

  The IP of the Proxy Host.
- Proxy Port (Optional)

  The port required by the proxy.
- Proxy Username (Optional)

  The username required by the proxy.
- Proxy Password (Optional)

  The password required by the proxy.

Actions

Turbonomic recommends actions for the New Relic supply chain as follows.

Entity Type	Action
Application / Application Server	• Resize Heap Recommendation only. • Resize Thread Pool Recommendation only. • Resize Connection Capacity Recommendation only. • Suspend VM Recommendation only. • Provision VM Recommendation only.
Virtual Machines	• Provision additional resources (VMem, VCPU) • Reconfigure Virtual Machine

Monitored Resources

> **NOTE:**
> The exact resources monitored will differ based on application type. This list includes all resources you may see.

Turbonomic monitors the following resources for the New Relic supply chain:

Entity Type	Commodity
Application	• Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second • Heap The utilization of the application server's heap Measured in Kilobytes (KB) • Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms) • Connections The utilization of the connection capacity. Only applicable to database servers Measured in Connections • Collection Time The percentage of server uptime spent garbage collecting. Available when the JVM profiler is enabled. Measured in percentage of uptime (%) • Threads The utilization of the server's thread capacity Measured in number of Threads
Database	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM

Entity Type	Commodity
	Measured in Megahertz (MHz) • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second • Heap The utilization of the application server's heap Measured in Kilobytes (KB) • DBMem The memory utilized by the database, as a of the memory capacity that is allocated to the database. Note that this resource is more accurate than the VMem resource on the hosting VM. With this resource, Turbonomic can drive resize and move actions based on the memory consumed by the database, not the memory consumed by the VM • Transactions The utilization of the allocated transactions per second for the given virtual application Measured in transactions per second • Connections The utilization of the connection capacity. Only applicable to database servers Measured in Connections • Collection Time The percentage of server uptime spent garbage collecting. Available when the JVM profiler is enabled. Measured in percentage of uptime (%) • Threads The utilization of the server's thread capacity Measured in number of Threads • Transaction Log The utilization of the server's capacity for storage devoted to transaction logs Measured in Kilobytes (KB) • DB Cache Hit Rate The percentage of accesses that result in cache hits. Measured as a percentage of hits vs total attempts (%)

Dynatrace

Turbonomic supports discovery of applications that are managed by the Dynatrace platform. Turbonomic includes the discovered information about these applications in its calculations for VM actions.

Prerequisites

- A Dynatrace Server instance running in your environment.

  This instance must be configured to monitor applications running in your environment.
- Managed VMs that are running applications managed by Dynatrace.

  For Turbonomic to discover applications through Dynatrace, the applications must be running on VMs in your environment. Also, these VMs must be managed by Turbonomic targets such as hypervisors or public cloud targets.
- A valid Dynatrace user account.

  The account must provide read-only access to the application data that is gathered by the Dynatrace server. It must enable the following switches for the access scope of the Dynatrace API token:

  - `Access problem and event feed, metrics, and topology`
  - `Read content log`
  - `Read synthetic monitors, locations, and nodes`
  - `Read configuration`
  - `User sessions`
  - `Anonymize user session data for data privacy reasons`
  - `Read audit logs`

Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The following table describes nomenclature congruency between the target and Turbonomic:

Dynatrace	Turbonomic
Application	Business Application
Service	Service
Process	Application Component, Database Server
-	Container
Host	Virtual Machine

Adding a Dynatrace Target

> **NOTE:**
> It is possible to monitor certain applications or database servers with both Dynatrace and Turbonomic. You should avoid such a configuration because it can cause Turbonomic to generate duplicate entities in the market.
>
> If you monitor an application via a Dynatrace server, and you configure that Dynatrace server as a Turbonomic target, then be sure you have not added that application as a separate application target in Turbonomic.

To add a Dynatrace server instance as a target, specify:

- Hostname or IP Address

  The host name or IP and endpoint, separated by a slash. For example, `10.10.10.10/e/b70e3eb2-e82b-4c13-a5a4-560d9865841r`
- API Token

The token that Turbonomic will use to authenticate its calls to the Dynatrace API. This token must have permission to execute GET methods via the Dynatrace API.

- Proxy Host

  The address of the proxy used for this target. Only fill out proxy information if you connect to the Dynatrace server via a proxy.

- Proxy Port

  The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

  The username for the account to log into the proxy specified above.

- Proxy Password

  The password to use with the proxy specified above.

Actions

Turbonomic does not recommend actions for Dynatrace applications. However, it considers resource utilization by these applications when recommending actions for the underlying VMs.

Monitored Resources

> **NOTE:**
> The exact resources monitored will differ based on application type. This list includes all resources you may see.
>
> Database Servers are discovered only for MySQL and MSSQL databases.

Turbonomic monitors the following resources for the Dynatrace supply chain:

Entity Type	Commodity
Business Application	• Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms) • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second
Service	• Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms) • Transactions The utilization of the allocated transactions per second for the given entity Measured in transactions per second
Application Component	• Collection Time The percentage of server uptime spent garbage collecting. Available when the JVM profiler is enabled.

Entity Type	Commodity
	Measured in percentage of uptime (%) **NOTE:** This commodity is collected for Java applications only. **NOTE:** This commodity is **NOT** collected for WebSphere applications. • Heap The utilization of the application server's heap Measured in Kilobytes (KB) **NOTE:** This commodity is collected for Java applications only.
Container	• VMem The of memory capacity the container utilizes Measured in Megabytes (MB) • VCPU The of CPU capacity the container utilizes Measured in Gigahertz (Ghz)
Container	• VMem The of memory capacity the container utilizes Measured in Megabytes (MB) • VCPU The of CPU capacity the container utilizes Measured in Gigahertz (Ghz)

SNMP

Turbonomic will discover application and operating system resources using SNMP (Simple Network Management Protocol) on Linux hosts in your entire environment, or a portion of your environment, based on scope.

Prerequisites

- Underlying VM host targets added to Turbonomic (for discovery)
- SNMP service enabled/configured on target VMs

> **NOTE:**
> For Hyper-V hosts, you must install Hyper-V Integration Services on the target VMs. For more information, please refer to the following integration services TechNet article:
>
> `https://technet.microsoft.com/en-us/library/dn798297%28v=ws.11%29.aspx`
>
> For VMware hosts, you must install VMware Tools on the target VMs.

Adding SNMP Targets

To add SNMP targets, select the **Guest OS Processes > SNMP** option on the Target Configuration page and provide the following information:

- Target Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- Scope

  Turbonomic will search for SNMP applications using the port entered above on the VMs found within the set scope. This scope can be set to the entire environment, single or multiple clusters, or particular virtual machines. Select this option and choose the scope for application discovery.

  > **NOTE:**
  > A single scope (target) must not contain more than 500 virtual machines. Turbonomic recommends utilization of multiple WMI targets for environments exceeding this limit.
- Community

  The SNMP v2c community string Turbonomic will use to connect to the application
- Port number

  The Port number Turbonomic will use to connect to the virtual machines hosting the application
- Full Validation

  When selected, Turbonomic will require every VM in the selected scope to be a valid target. If Turbonomic is unable to authenticate a VM in the scope, the target will not validate and data will not be collected.
- Enable SNMPv3

  When checked, Turbonomic will use SNMPv3 to connect to the virtual machines in the selected scope
- SNMPv3 Username/Securityname

  The Username or Securityname that Turbonomic will use to connect to the virtual machines hosting the application
- SNMPv3 Enable Privacy

  When checked, Turbonomic will encrypt using the privacy password to keep the connection private.
- SNMPv3 Authentication Password

  The Authentication Password Turbonomic will use to connect to the virtual machines in the selected scope. This allows requests to be authenticated, confirming the sender's identity.
- SNMPv3 Privacy Password

  The Privacy Password Turbonomic will use to connect to the virtual machines in the selected scope. For requests to be encrypted, the SNMP manager and the SNMP agent must share knowledge of the privacy password associated with the username.
- SHA-1 For Authentication

  When checked, Turbonomic will use SHA-1 Authentication to connect to the virtual machines in the selected scope. When unchecked, Turbonomic uses MD5
- AES For Privacy

  When checked, Turbonomic will use AES 128 to connect to the virtual machines in the selected scope. When unchecked, Turbonomic uses DES

> **NOTE:**
> Turbonomic does not support AES 192 or 256 for SNMP targets.

Supported Actions

For each discovered entity within the application supply chain, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Action
Applications	Without discovered Guest OS Processes or Application Servers, Turbonomic doesn't perform actions on applications. Instead, it performs resize actions on the host VMs. If host utilization is high enough on the physical machine running the application VM, Turbonomic may also recommend provisioning a new host. For specific application servers, see the individual application entry.
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine

Monitored Resources

Turbonomic monitors the following resources for the application supply chain:

Entity Type	Commodity
WMI/SNMP Application	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz)
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS)

Entity Type	Commodity
	The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)

Enabling SNMP

Turbonomic requires that SNMP is enabled and configured in order to discover Guest Processes. While these steps will change slightly between OS versions, this topic gives you the general instructions:

Enabling SNMP v2

1. Obtain and install the SNMP server package for your version of Linux. This is commonly called `net-snmp` in most package managers. Also install the `net-snmp-utils` package if it is available
2. Configure the SNMP daemon by editing `/etc/snmp/snmpd.conf` or running `snmpconf -i` at the command line in order to verify that:
   - SNMPv2c is enabled
   - A read-only community name has been set. This community name will be used by Turbonomic to communicate with the VM

   > **NOTE:**
   > The community name you select must have at least read-only access to all OIDs in the system.
3. Configure the daemon to listen on a public interface. Most default installations only listen on `127.0.0.1`
4. Start the SNMP daemon process

These long form and short form sample `snmp.conf` files illustrate the necessary changes.

Long form, VACM:

```
# First, map the community name "mycommunity1" into a "security name"
        # sec.name  source          community
        com2sec notConfigUser  default      mycommunity1
        # Second, map the security name into a group name:
        #   groupName securityModel securityName
        group   notConfigGroup v2c           notConfigUser
        # Finally, grant the group read-only access to any UUID.
        #   group context sec.model sec.level prefix read   write  notif
        access  notConfigGroup ""      any       noauth    exact  all none none
```

Short form, older syntax:

```
rocommunity mycommunity1 default system
```

Verify Your SNMP Setup

Verify that your SNMP setup is successful by using the following command from a remote machine, replacing the community string and IP address:

```
snmpwalk -Os -v 2c -c COMMUNITY_STRING IP_ADDRESS iso.3.6.1.2.1.1.1
```

If successful, the command will return the kernel version of the machine (similar to the output of `uname -a`)

Enabling SNMP v3

1. Obtain and install the SNMP server package for your version of Linux.

   Most package managers call this package `net-snmp`. To verify your configuration in step 5 below, you should also install the `net-snmp-utils` package on the Turbonomic VM.
2. Stop the SNMP service.
3. Create the SNMPv3 user.

   Execute the following command:

   ```
   net-snmp-config --create-snmpv3-user [-ro] [-A authpass] [-X privpass] [-a MD5|SHA] [-x DES|AES] [username]
   ```

   For example, the command:

   ```
   i. net-snmp-create-v3-user -ro -A snmpv3authPass -a SHA -X snmpv3encPass -x AES snmpv3user
   ```

   results in output similar to:

   ```
   adding the following line to /var/lib/net-snmp/snmpd.conf:
   ```

   ```
   createUser snmpv3user SHA "snmpv3authPass" AES snmpv3encPass adding the following line to /etc/snmp/snmpd.conf:
   ```

   ```
   rouser snmpv3user
   ```
4. Secure the SNMP daemon.

   Use a host firewall to only allow requests from source IP addresses you know, and which need to query the system.
5. Verify your SNMP setup.

   To verify the configuration, you can execute the following command on the Turbonomic VM:

   ```
   snmpwalk -v3 -u snmpv3user -A snmpv3authPass -a SHA -X snmpv3encPass -x AES -l authPriv <TargetIP>
   ```

WMI

Turbonomic will discover application and operating system resources using WMI (Windows Management Instrumentation) in your entire environment, or a portion of your environment, based on scope.

Prerequisites

- Underlying VM host targets added to Turbonomic (for discovery)
- A WMI user account Turbonomic can use to connect to the WMI targets. This account can either be an administrator user, or a non-administrator belonging to certain local user groups. For a full list of requirements, see Creating a WMI User Account *(on page 91)*
- WMI enabled on target VMs. For assistance in enabling WMI, see Enabling WMI *(on page 92)*.

> **NOTE:**
> For Hyper-V hosts, you must install Hyper-V Integration Services on the target VMs. For more information, please refer to the following integration services TechNet article:
>
> `https://technet.microsoft.com/en-us/library/dn798297%28v=ws.11%29.aspx`
>
> For VMware hosts, you must install VMware Tools on the target VMs.

Adding WMI Targets

To add WMI targets, select the **Guest OS Processes > WMI** option on the Target Configuration page and provide the following information:

- Target Name

  The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.
- Username

  The username Turbonomic will use to connect to the WinRM service on the Virtual Machine hosting the application. This should not include the Active Directory domain.
- Password

  The password Turbonomic will use to connect to the WinRM service on the Virtual Machine hosting the application.
- Scope

  Turbonomic will search for Windows applications on the VMs found within the set scope. This scope can be set to the entire environment, single or multiple clusters, or particular virtual machines. Select this option and choose the scope for application discovery.

  > **NOTE:**
  > A single scope (target) must not contain more than 500 virtual machines. Turbonomic recommends utilization of multiple WMI targets for environments exceeding this limit.
- Domain Name

  The Active Directory domain used by Turbonomic in conjunction with the Username for authentication. Leave blank for local accounts.
- Full Validation

  When selected, Turbonomic will require every VM in the selected scope to be a valid target. If Turbonomic is unable to authenticate a VM in the scope, the target will not validate and data will not be collected.

Supported Actions

For each discovered entity within the application supply chain, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Action
Applications	Without discovered Guest OS Processes or Application Servers, Turbonomic doesn't perform actions on applications. Instead, it performs resize actions on the host VMs. If host utilization is high enough on the physical machine running the application VM, Turbonomic may also recommend provisioning a new host.

Entity Type	Action
	For specific application servers, see the individual application entry.
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine

Monitored Resources

Turbonomic monitors the following resources for the container supply chain:

Entity Type	Commodity
WMI/SNMP Application	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz)
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)

Creating a WMI User Account

The service account Turbonomic uses to connect to WMI should be an Active Directory domain account.

Some enterprises require that the service account does not grant full administrator rights. In that case, you can create a restricted service account for WMI:

1. Add the account to each of the following local groups:
   - `WinRMRemoteWMIUsers__` (or `Remote Management Users`)
   - `Performance Monitor Users`

   > **NOTE:**
   > These groups are standard Windows Server 2012 security groups. If you are using an earlier version of Windows Server and do not see these groups, contact Turbonomic Support for assistance.
2. Grant permissions to the account.

   In the WMI Management console, grant the `Enable Account` and `Remote Enable` advanced security permissions to the service account:
   - Open the WMI Management console (`wmimgmt`).
   - Right-click **WMI Control (Local)** and choose **Properties**.
   - Go to the **Security** tab and then click **Security** to display the **Security for Root** dialog.
   - Click **Advanced**, select the service account, and click **Edit**.
   - Confirm that **This namespace and subnamespace** is selected.
   - Select **Enable Account** and **Remote Enable** and click **OK**.

Enabling WMI

Enabling Discovery

To enable Turbonomic discovery of Windows-based Guest Processes, you must configure and enable WMI with WinRM, usually by using a group policy for AD Domains.

When local credentials are provided to Turbonomic, NTLM is the authentication mechanism. When domain credentials are provided, either NTLM or Kerberos can be specified.

> **NOTE:**
> If Kerberos is specified, the IP address of the target must be resolvable to a host name using DNS.

While these steps will change slightly between Windows Server versions, the general instructions are:

1. Open an elevated PowerShell prompt on the virtual machine(s) to be discovered as a WMI Target.
2. Enable the WS-Management protocol and set the default configuration for remote management.

   Execute `winrm quickconfig`.
3. Enable the WinRM *Negotiate* authentication scheme.

   This policy is enabled by default. To enable this policy if it is disabled, execute `winrm set winrm/config/service/auth '@{Negotiate="true"}'`.
4. Set the WinRM `AllowUnencrypted` property to `true` if non-SSL connections are preferred.

   This property must be set on both the server and the client. Note that setting this value to `true` does not mean that WMI passes sensitive data in an unencrypted form. It will send only the content of the SOAP messages as plain text:
   - Server setting:

```
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
```

- Client setting:

```
winrm set winrm/config/client '@{AllowUnencrypted="true"}'
```

5. Restart the Remote Registry service.

After restarting the Remote Registry service, Turbonomic will discover the WMI targets.

SSL Connections

To connect using SSL, then a certificate must be assigned to WinRM, and an SSL listening socket enabled. Turbonomic will accept self-signed certificates. Complete details can be found in the official WinRM documentation.

Some example SSL-related commands:

- List certificates on the system:

```
ls cert://localmachine/my
```

- Find Thumbprint for the default certificate:

```
7B56C33F029E7605D5C58E5597D0A077FE1D7F1C CN=winsql-server1.corp.mycorp.com
```

- Enable SSL listener:

```
winrm create winrm/config/listener?Address=*+Transport=HTTPS @{Hostname="winsql-
server1.corp.mycorp.com";CertificateThumbprint="7B56C33F029E7605D5C58E5597D0A077FE1D7F1C";Por
```

SQL Server

Turbonomic supports Microsoft SQL Server 2012, 2014, 2016, 2017, and 2019.

> **NOTE:**
> SQL Server clusters are not supported by this version of Turbonomic.

Prerequisites

- A user account with SQL permissions including `Connect SQL` and `View Server State` on the database
- The following services must be running, and set to enabled:
  - Net.Tcp Listener Adapter
  - Net.Tcp Port Sharing Service
- TCP/IP is enabled on the port used for Turbonomic discovery
- To enable dynamic port discovery, the port used by the SQL Browser Service

Creating a Service User Account

The user account that Turbonomic uses for its service login must include the following:

- The account must exist in the Security folder within the SQL Server Object Explorer, with the following properties:
  - Enable **SQL Server Authentication**
  - Disable **Enforce password policy**
- The account's security properties must include:
  - Permission to connect to the database through SQL

- Permission to view the server state

Adding a SQL Server Database to Turbonomic

You can add an individual database server as a target, or you can add all matching targets within a given scope.

To add a database server as a target, you specify:

- Target Name

  Name displayed in the Turbonomic UI
- Username

  Username for the account. This username must not include the AD domain
- Password

  Password for the account. This username must not include the AD domain
- Scope

  A cluster or group of VMs that host SQL servers
- Browsing Service Port

  The port used to communicate with the browsing service. Turbonomic will obtain the SQLServer port for each instance running on each VM in the scope.
- SQLServer Port

  The SQL remote port. Turbonomic will use this port if there is no browsing service port specified, or if the browsing service is not available during discovery.

  > **NOTE:**
  > Turbonomic will connect to the port specified for the SQL browsing service first. If that connection fails, Turbonomic will connect using the SQLServer Port.
- Domain Name

  The Active Directory domain used by Turbonomic in conjunction with the Username for authentication. Leave blank for local accounts.
- Full Validation

  When selected, Turbonomic will require every VM in the selected scope to be a valid target. If Turbonomic is unable to authenticate a VM in the scope, the target will not validate and data will not be collected.

For additional information about adding SQL Server targets, see the Green Circle article, "How to Target SQL Servers in VMTurbo (5.4 and later)"

Actions

Turbonomic recommends actions for the application supply chain as follows.

Entity Type	Action
Applications	Without discovered Guest OS Processes or Application Servers, Turbonomic doesn't perform actions on applications. Instead, it performs resize actions on the host VMs. If host utilization is high enough on the physical machine running the application VM, Turbonomic may also recommend provisioning a new host.

Entity Type	Action
	For specific application servers, see the individual application entry.
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine

Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

Entity Type	Commodity
Database Server	• DBMem The memory utilized by the database, as a of the memory capacity that is allocated to the database. Note that this resource is more accurate than the VMem resource on the hosting VM. With this resource, Turbonomic can drive resize and move actions based on the memory consumed by the database, not the memory consumed by the VM • Transactions The utilization of the allocated transactions per second for the given virtual application Measured in transactions per second • Response Time The utilization of the server's allocated response time Measured in Milliseconds (ms) • Connections The utilization of the connection capacity. Only applicable to database servers Measured in Connections • TransactionLog The utilization of the server's capacity for storage devoted to transaction logs Measured in Kilobytes (KB) • Cache Hit Rate The percentage of accesses that result in cache hits. Measured in a percentage of hits vs total attempts (%)
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz)

Entity Type	Commodity
	• Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)

Kubernetes Platform Targets

If you are running the Cloud Native Edition of Turbonomic, it will discover entities related to container platforms in your environment, recommend resize, placement, and scale actions, and display discovered entities. To use this functionality, you must add a container platform target.

Prerequisites

- Running Kubernetes 1.8+ Cluster

Adding a Kubernetes Platform Target

In order to add this target, you deploy a pod running the Kubeturbo container image, which will self-register a Cloud Native target type that represents your Kubernetes cluster. Kubeturbo will use a configuration file with your Turbonomic credentials and a service account RBAC model to discover and monitor the entities in your Kubernetes cluster.

For more information on different deployment options, see the Kubeturbo GitHub Repository.

> **NOTE:** You must install Kubeturbo on each cluster you want Turbonomic to discover.

Supply Chain

Turbonomic adds several entities to the supply chain: Services, Containers, Container Pods, Container Specs, Workload Controllers, Namespaces, Volumes, and VirtualMachines. Each entity represents key components of your containerized application running in Kubernetes.

Actions

Turbonomic recommends actions for the Kubernetes container platform supply chain as follows.

Entity	Action
Service	**None**

Entity	Action
	No actions are recommended at this level of the supply chain. Instead, actions that affect the service are generated and executed on underlying entities.
Application Component	**Suspend** Application components are suspended while due to a node (virtual machine) suspension **APM Actions** Application components may also receive other actions as part of APM integration related to those use cases. For example, a `Resize Heap` action from an underlying AppDynamics integration. See the Target Configuration Guide documentation for the appropriate technology to discover what actions may be available.
Container	**Resize Container Up/Down** With `Merged Actions` enabled, individual Container actions will be recommend only and the resize will be reflected as an action on the Workload Controller entity. **Suspend** Containers are suspended while due to a node (virtual machine) suspension
Container Pod	**Move Pod** Pods will be moved across nodes (Virtual Machines). **Suspend** Container Pods are suspended while due to a node (virtual machine) suspension
Container Spec	**None** No actions are recommended at this level of the supply chain. This entity maintains the history of all replicas, or instances of pods for this container specification.
Workload Controller	**Resize Container** With `Merged Actions` enabled, this is a single resize action representing all resize actions for containers associated to a specific workload controller.
Namespace	**None** No actions are recommended at this level of the supply chain. Namespace Quotas are constraints to container resizing actions.
Virtual Machine (Node)	**Provision Additional Resources** The following resources may be provisioned: • VMem • VCPU • VMem Requests • VCPU Requests • Number of Consumers **Suspend** Nodes (virtual machines) may be suspended.

Entity	Action
	Infrastructure-dependent Actions Depending on the technology the node is stitched to for underlying infrastructure, there may be additional actions: On-prem VMware: • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine

Monitored Resources

Turbonomic monitors the following resources for the Kubernetes container platform supply chain:

Entity	Commodity
Service	**Response Time** Response time of the service, measured in ms. This commodity is populated via APM or DIF integrations. **Transactions** Transaction utilization, measured in transactions per second. This commodity is populated via APM or DIF integrations.
Application Component	**Various Commodities** The commodities monitored and the values received for those commodities at the application component level is dependent on the APM integration used. See the Target Configuration Guide documentation for the appropriate technology to discover what data will be reported.
Container	**VMem** The virtual memory utilized by the container against the memory limit (if no limit is set, then node capacity is used). Measured in Megabytes (MB) **VCPU** The virtual CPU utilized by the container against the CPU limit (if no limit is set, then node capacity is used). Measured in Megahertz (Mhz) **VMemRequest** If applicable, the virtual memory utilized by the container against the memory request. Measured in Megabytes (MB) **VCPURequest** If applicable, the virtual CPU utilized by the container against the CPU request. Measured in Megahertz (Mhz)
Container Pod	**VMem**

Entity	Commodity
	The virtual memory utilized by the pod against the node physical capacity. Measured in Megabytes (MB) **VCPU** The virtual CPU utilized by the pod against the node physical capacity. Measured in Megahertz (Mhz) **VMemRequest** The virtual memory request allocated by the pod against the node allocatable capacity. Measured in Megabytes (MB) **VCPURequest** The virtual CPU request allocated by the pod against the node allocatable capacity. Measured in Megahertz (Mhz) **VMemRequestQuota** If applicable, The amount of virtual memory request the pod has allocated against the namespace quota. Measured in Megabytes (MB) **VCPURequestQuota** If applicable, The amount of virtual CPU request the pod has allocated against the namespace quota. Measured in Megahertz (Mhz) **VMemLimitQuota** If applicable, The amount of virtual memory limit the pod has allocated against the namespace quota. Measured in Megabytes (MB) **VCPULimitQuota** If applicable, The amount of virtual CPU limit the pod has allocated against the namespace quota. Measured in Megahertz (Mhz)
Container Spec	**VMem** The virtual memory historically utilized by any containers run for this workload against the memory limit (if no limit is set, then node capacity is used). Measured in Megabytes (MB) **VCPU** The virtual CPU historically utilized by any containers run for this workload against the CPU limit (if no limit is set, then node capacity is used). Measured in Megahertz (Mhz) **VMemRequest** If applicable, the virtual memory historically utilized by any containers run for this workload against the memory request. Measured in Megabytes (MB) **VCPURequest** If applicable, the virtual CPU historically utilized by any containers run for this workload against the CPU request. Measured in Megahertz (Mhz)
Workload Controller	**VMemRequestQuota**

Entity	Commodity
	If applicable, The amount of virtual memory request the pod has historically allocated for this workload against the namespace quota. Measured in Megabytes (MB) **VCPURequestQuota** If applicable, The amount of virtual CPU request the pod has historically allocated for this workload against the namespace quota. Measured in Megahertz (Mhz) **VMemLimitQuota** If applicable, The amount of virtual memory limit the pod has historically allocated for this workload against the namespace quota. Measured in Megabytes (MB) **VCPULimitQuota** If applicable, The amount of virtual CPU limit the pod has historically allocated for this workload against the namespace quota. Measured in Megahertz (Mhz)
Namespace	**VMemRequestQuota** The total amount of virtual memory request for all pods allocated to the namespace against the namespace quota. Measured in Megabytes (MB) **VCPURequestQuota** The total amount of virtual CPU request for all pods allocated to the namespace against the namespace quota. Measured in Megahertz (Mhz) **VMemLimitQuota** The total amount of virtual memory limit for all pods allocated to the namespace against the namespace quota. Measured in Megabytes (MB) **VCPULimitQuota** The total amount of virtual CPU limit for all pods allocated to the namespace against the namespace quota. Measured in Megahertz (Mhz)
Virtual Machine (Node)	**VMem** The virtual memory utilized by the node against the memory allocated to the hosting virtual machine. Measured in Megabytes (MB) **VCPU** The virtual CPU utilized by the node against the CPU allocated to the hosting virtual machine. Measured in Megahertz (Mhz) **VMemRequest** The total amount of virtual memory allocated to pods with memory request against the allocatable capacity of the node. Measured in Megabytes (MB) **VCPURequest** The total amount of virtual CPU allocated to pods with CPU request against the allocatable capacity of the node. Measured in Megahertz (Mhz) **Number Consumers** The total number of pods running on the node against the maximum number of pods allowed. Measured in Pods (#)

Entity	Commodity
	Infrastructure-dependent Commodities Depending on the technology the node is stitched to for underlying infrastructure, there may be additional commodities, or more granular data reported to existing commodities. See the Target Configuration Guide documentation for the appropriate technology to discover what data will be reported.

Fabric Targets

A fabric target is a service that unites compute, network and storage access into a cohesive system. When you connect Turbonomic to fabric targets, it monitors the performance and resource consumption of your fabric interconnects, IO modules, chassis, and Blade Servers to assure application performance while also utilizing resources as efficiently as possible.

Once connected, Turbonomic discovers the blades that host the VMs, the chassis and datastores that provide resources to the blades, the IO modules and fabric interconnects that provide network resources, and the virtual datastores that provide storage resources to the VMs.

As part of this process, Turbonomic will stitch information from the fabric target and the associated hypervisor targets together, to provide deeper insight into the state of the hardware, and information related to the applications and VM's running on the hypervisor-stitched blades.

Supply Chain

Fabric targets add IO Module, Fabric Interconnect, Domain, and Chassis entities to the supply chain. The Chassis entities host physical machines (blade servers) — The physical machines also consume network connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network, and also hosts the UCS Manager for UCS Targets. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect.

Supported Fabric Targets

Turbonomic supports the following fabric targets:

- Cisco UCS

Monitored Resources

Turbonomic monitors the following resources for the fabric supply chain:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz)

Entity Type	Commodity
	• Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Blade	• Net The utilization of data through the Blade's network adapters Measured in Kilobytes per second (KB/s) • Treated as a Physical Machine of the underlying Hypervisor (see below) CPU, Mem, etc.
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only

Entity Type	Commodity
	Measured in Megahertz (MHz)
I/O Module	• NetThroughput Rate of message delivery over a port Measured in Megabits per second (Mb/s)
Fabric Interconnect	• NetThroughput Rate of message delivery over a port Measured in Mb/s • PortChannel Amalgamation of ports with a shared net throughput and utilization Measured in Mb/s

Cisco UCS Manager

The Cisco Unified Computing System (UCS) Manager is a management solution that participates in server, fabric, and storage provisioning, device discovery, inventory, configuration, diagnostics, monitoring, fault detection, auditing, and statistics collection.

UCS integrates all of these resources in a scalable multi-chassis platform to converge administration onto a single point. Managing these various entities on a network fabric with Turbonomic enables automation at the hardware level, including automated provisioning of hosts.

Prerequisites

- A service account Turbonomic can use to connect to UCS Manager

Adding UCS Targets

To add a UCS target, select the **Fabric** category and choose one of the UCS Fabric options to match the version of UCS you want to manage. Then provide the following information:

- **Address:** The IP address of the UCS Manager

  This gives access to the Fabric Manager that resides on the interconnect.

  Turbonomic connects to the UCS Manager via the HTTPS protocol by default. In order to force the HTTP protocol, the Address must be entered in one of two ways. For example, an IP of 8.8.8.8 must be entered as http://8.8.8.8 or by using a specific HTTP port, such as 8.8.8.8:80.
- **Username/Password:** The credentials of the account Turbonomic will use to connect to UCS Manager.

  specify the IP address and credentials for UCS Manager. Turbonomic discovers the fabric interfaces associated with that manager.

> **NOTE:**
> When providing a username, if the account is managed in Active Directory you must include the domain in case-sensitive spelling. For example,`MyDomain\john` is not the same as `mydomain\john`. For local user accounts, just provide the username.

Supply Chain

Fabric targets add IO Module, Fabric Interconnect (Switch), Domain, and Chassis entities to the supply chain. The Chassis entities host physical machines — The physical machines also consume network connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network, and also hosts the UCS Manager. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect.

Actions

Turbonomic recommends actions for the various entities of the UCS Fabric Network as follows:

Entity Type	Action
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine
Physical Machines	• Start Physical Machine • Provision Physical Machine • Suspend Physical Machine
Chassis	• Provision New Chassis
Fabric Interconnect	• Add Port to Port Channel • Remove Port from Port Channel • Add Port
DPod (if Network Flow target is present)	• Provision new DPod

Monitored Resources

Turbonomic monitors the following commodities of the UCS target:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage)

Entity Type	Commodity
	The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
Chassis	• Power Electricity being consumed by the Chassis Measured in Watts (W) • Temperature Temperature of the internals of the Chassis

Entity Type	Commodity
	Measured in degrees Celsius (C)
I/O Module	• NetThroughput Rate of message delivery over a port Measured in Megabits per second (Mb/s)
Fabric Interconnect	• NetThroughput Rate of message delivery over a port Measured in Mb/s • PortChannel Amalgamation of ports with a shared net throughput and utilization Measured in Mb/s
DPod (if Network Flow target is present)	• Memory (Mem) The utilization of the DPod's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the DPod's CPU reserved or in use Measured in Megahertz (MHz) • Storage The utilization of the storage attached to the DPod Measured in Kilobytes (KB) • Flow The utilization of the network flow capacity utilized by the DPod. This is divided into Flow1 (Low Cost) and Flow2 (Medium Cost) utilization Measured in Kilobytes per second (KB/s)

HPE OneView

HPE OneView is a management solution that streamlines provisioning and lifecycle management across compute, storage, and fabric. Through a unified API, infrastructure can be configured, monitored, updated, and re-purposed.

HPE OneView integrates all of these resources in a scalable multi-enclosure platform to converge administration onto a single point. Managing these various entities on a network fabric with Turbonomic enables automation at the hardware level, including automated provisioning of hosts.

Prerequisites

- A service account Turbonomic can use to connect to HPE OneView.
- HPE OneView 2.0 and compatible hardware.

Adding HPE OneView Targets

To add a HPE OneView as a target, select the **Fabric** category and choose the HPE OneView radio button. Then provide the following information:

- **Address:** The IP address of the HPE OneView target

  This gives access to the Fabric Manager that resides on the VM.

  Turbonomic uses the HTTPS protocol by default. In order to force the HTTP protocol, the Address must be entered in one of two ways. For example, an IP of 8.8.8.8 must be entered as http://8.8.8.8 or by using a specific HTTP port, such as 8.8.8.8:80.
- **Username/Password:** The credentials of the account Turbonomic will use to connect to the HPE OneView target.

  specify the IP address and credentials for HPE OneView. Turbonomic discovers the fabric interfaces associated with that instance.

> **NOTE:**
> When providing a username, if the account is managed in Active Directory you must include the domain in case-sensitive spelling. For example, `MyDomain\john` is not the same as `mydomain\john`. For local user accounts, just provide the username.

Supply Chain

Fabric targets add IO Module, Fabric Interconnect (Switch), Domain, and Chassis entities to the supply chain. The Chassis entities host physical machines — The physical machines also consume network connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect.

> **NOTE:**
> For HPE OneView targets, the "Fabric Interconnect" entity exists as a false "Switch", and only as a pass-through for network resources. Unlike other fabric targets, such as UCS, there is no physical hardware that serves this function.

Actions

Turbonomic recommends actions for the various entities of the HPE OneView Fabric Network as follows:

Entity Type	Action
Virtual Machines	• Provision additional resources (VMem, VCPU) • Move Virtual Machine • Move Virtual Machine Storage • Reconfigure Storage • Reconfigure Virtual Machine
Physical Machines	• Start Physical Machine • Provision Physical Machine • Suspend Physical Machine
Fabric Interconnect	• Add Port to Port Channel • Remove Port from Port Channel

Entity Type	Action
	• Add Port
DPod (if Network Flow target is present)	• Provision new DPod

Monitored Resources

Turbonomic monitors the following commodities of the HPE OneView target:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB) • Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space

Entity Type	Commodity
	Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM's allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
I/O Module	• NetThroughput Rate of message delivery over a port Measured in Megabits per second (Mb/s)
Fabric Interconnect	• NetThroughput Rate of message delivery over a port Measured in Mb/s • PortChannel Amalgamation of ports with a shared net throughput and utilization Measured in Mb/s
DPod (if Network Flow target is present)	• Memory (Mem) The utilization of the DPod's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the DPod's CPU reserved or in use Measured in Megahertz (MHz)

Entity Type	Commodity
	• Storage The utilization of the storage attached to the DPod Measured in Kilobytes (KB) • Flow The utilization of the network flow capacity utilized by the DPod. This is divided into Flow1 (Low Cost) and Flow2 (Medium Cost) utilization Measured in Kilobytes per second (KB/s)

Hyperconverged Targets

A hyperconverged target is a service that unites compute, network and storage access into a cohesive system. When you connect Turbonomic to hyperconverged targets, it monitors the performance and resource consumption of your stitched infrastructure, enhancing the data used to assure application performance while also utilizing resources as efficiently as possible.

As part of this process, Turbonomic will stitch information from the hyperconverged target and the associated hypervisor and fabric targets together, to provide deeper insight into the state of the hardware, and information related to the applications and workload running in your environment.

Supported Hyperconverged Targets

Turbonomic supports the following hyperconverged targets:

- Cisco HyperFlex
- Nutanix

Monitored Resources

Turbonomic monitors the following resources for the hyperconverged supply chain, once stitched to your hypervisor and other associated targets:

Entity Type	Commodity
Virtual Machine	• Virtual Memory (VMem) The utilization of the VMem allocated to the hosting VM Measured in Kilobytes (KB) • Virtual CPU (VCPU) The utilization of the VCPU allocated to the hosting VM Measured in Megahertz (MHz) • Virtual Storage (VStorage) The utilization of the virtual storage capacity allocated for the VM Measured in Kilobytes (KB)

Entity Type	Commodity
	• Storage Access Operations Per Second (IOPS) The utilization of IOPS allocated for the VStorage on the VM Measured in IOPS • Latency The utilization of latency allocated for the VStorage on the VM Measured in milliseconds (ms)
Blade	• Net The utilization of data through the Blade's network adapters Measured in Kilobytes per second (KB/s) • Treated as a Physical Machine of the underlying Hypervisor (see below) CPU, Mem, etc.
Physical Machine	• Memory (Mem) The utilization of the PM's memory reserved or in use Measured in Kilobytes (KB) • CPU The utilization of the PM's CPU reserved or in use Measured in Megahertz (MHz) • IO The utilization of the PM's IO adapters Measured in Kilobytes per second (KB/s) • Net The utilization of data through the PM's network adapters Measured in Kilobytes per second (KB/s) • Swap The utilization of the PM's swap space Measured in Kilobytes (KB) • Balloon The utilization of shared memory among VMs running on the host. ESX-only Measured in Kilobytes (KB) • CPU Ready The utilization of the PM’s allocated ready queue capacity (measured in Kbytes) that is in use, for 1, 2, and 4 CPU ready queues. ESX-only Measured in Megahertz (MHz)
I/O Module	• NetThroughput Rate of message delivery over a port

Entity Type	Commodity
	Measured in Megabits per second (Mb/s)
Fabric Interconnect	• NetThroughput Rate of message delivery over a port Measured in Mb/s • PortChannel Amalgamation of ports with a shared net throughput and utilization Measured in Mb/s
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)

Nutanix

Nutanix products provide hyperconverged platforms that include VM hosting and a distributed storage fabric. The platform presents storage in two tiers — Local HDD storage and server-attached flash (hot storage).

Nutanix environments include:

- One or more Nutanix appliances

  An appliance contains up to four server nodes.
- Nutanix nodes

  Servers that expose compute and storage resources. Each node provides local HDD and hot storage. Nodes combine to form a unified cluster that pools resources.
- Controller VMs

  Each node includes a Controller VM that manages the node's resources within the cluster pool. To minimize storage latency, the Controller VM keeps the most frequently accessed data in the hot storage.

Turbonomic supports management of Nutanix fabrics, where the supply chain treats a Nutanix Storage Pool as a disk array. Turbonomic recognizes Nutanix storage tiers when calculating placement of VMs and VStorage. In addition, Turbonomic can recommend actions to scale flash capacity up or down by adding more hosts to the cluster, or more flash drives to the hosts.

To specify a Nutanix target, provide the Cluster External IP address. This is a logical IP address that always connects to one of the active Controller VMs in the cluster. In this way, you can specify a Nutanix target without having to specify an explicit Controller VM.

> **NOTE:**
> The Controller VM must remain *pinned* to its host machine — You must not move the Controller VM to a different host. If the Nutanix cluster uses the Nutanix Acropolis OS to manage VMs, Turbonomic automatically pins the Controller VMs. However, if you use vCenter Server or Hyper-V to manage VMs on the hosts, you must configure a group to pin the Controller VMs. For more information, see Pinning Nutanix Controller VMs in Generic Hypervisor Mode *(on page 116)*.

Prerequisites

- A service account with administrator rights on the Nutanix target

Finding the Cluster External IP Address

To set a Nutanix target, provide the Cluster External IP address for the given Nutanix cluster.

The Cluster External IP address is a logical IP that resolves to the cluster's Prism Element Leader. If the Prism Element Leader fails, then the Cluster External IP address will resolve to the newly elected Prism Element Leader.

To find this IP address, open the Web Console (the Prism Element) on the cluster and navigate to the **Cluster Details** view. In this view you can see the **Cluster External IP** address. If there is no IP address specified, you can specify the address at this time. For more information, see the Nutanix documentation.

Operating Modes

A Nutanix node is a server that hosts VMs — In this sense the node functions as a hypervisor. A cluster of nodes can host VMs using the following Hypervisor technologies:

- Nutanix Acropolis

The native Nutanix host platform, which combines software-defined storage with built-in virtualization.

- VMware ESXi
- Microsoft Hyper-V

Turbonomic supports Nutanix cluster management in the Generic Hypervisor Mode (ESXi or Hyper-V). In this mode you:

- Add each Hyper-V host or vCenter as a hypervisor target — This enables VM workload control for the respective hypervisor technologies
- Specify the Nutanix Cluster External IP address as the target address — This adds the cluster as a Storage Controller target to enable Turbonomic storage control

Controller VM Pinning

Each Nutanix node hosts a Controller VM that runs the Nutanix software and manages I/O for the hypervisor and all VMs running on the host. Each Controller VM must remain on its host node —The Controller VM must be *pinned* to that host, and must not be moved to any other host.

For more information about how to pin the Controller VM, see Pinning Nutanix Controller VMs in Generic Hypervisor Mode *(on page 116)*.

Adding Nutanix Targets

> **NOTE:**
> This describes how to add a Nutanix cluster to Turbonomic as a target. The steps are the same no matter which operating mode you use (Standalone or Generic Hypervisor). Before you add the cluster as a target, you should know which operating mode you intend. If you want Standalone mode, then you will have to enable that operating mode after adding the cluster. If you want Generic Hypervisor mode, then you will have to add the hypervisors as targets after you have added the Nutanix cluster as a target. For more information, see Hypervisor Targets *(on page 7)*.

To add Nutanix targets, select the **Hyperconverged > Nutanix** option on the Target Configuration page and provide the following information:

- Address

  The Cluster External IP address for the Nutanix cluster.
- Port Number

  The port that the cluster listens on.
- Secure Connection

  Enable this to use a secure connection.
- Username/Password

  Credentials for an account on the Nutanix cluster.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in Nutanix to those used in Turbonomic:

Nutanix Name	Turbonomic Entity
Container	Storage
Storage Pool	Disk Array
Nutanix Cluster	Storage Controller

Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

Entity Type	Can Be Automated	Recommendations only
VM (a Nutanix VM)	Move, Resize Resize actions require the VM to power down, and power back on again.	
Datastore ("Storage")	Provision, Resize Up, Resize Down, Suspend	
Disk Array		
Storage Controller		Provision

Monitored Resources

Turbonomic monitors the following storage resources:

Entity Type	Commodity
Storage	• Storage Amount The utilization of the datastore's capacity Measured in Megabytes (MB) • Storage Provisioned The utilization of the datastore's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS) The summation of the read and write access operations per second on the datastore Measured in Operations per second • Latency The utilization of latency on the datastore Measured in Milliseconds (ms)
Disk Array	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • Storage Amount The utilization of the Disk Array's capacity. Measured in Megabytes (MB) • Storage Provisioned The utilization of the Disk Array's capacity, including overprovisioning. Measured in Megabytes (MB) • Storage Access Operations Per Second (IOPS)

Entity Type	Commodity
	The summation of the read and write access operations per second on the disk array Measured in Operations per second • Latency The utilization of latency, computed from the latency of each device in the disk array. Measured in milliseconds (ms)
Storage Controller	**NOTE:** Not all targets provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data. • CPU The utilization of the Storage Controller's CPU in use Measured in Megahertz (MHz) • Storage Amount The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller Measured in Megabytes (MB)

Pinning Nutanix Controller VMs in Generic Hypervisor Mode

Each Nutanix node hosts a Controller VM that runs the Nutanix software and manages I/O for the hypervisor and all VMs running on the host. Each Controller VM must remain on its host node —The Controller VM must be *pinned* to that host, and must not be moved to any other host.

For a cluster in Generic Hypervisor mode (using vCenter or Hyper-V hypervisors), you must use Turbonomic policies to pin the Controller VMs to their respective nodes. To do this, you will create a dynamic group of Nutanix Controller VMs, and then disable move actions for all members of this group.

To pin the Controller VMs:

1. Create a group of Controller VMs.

   In Turbonomic you can create dynamic groups based on VM name — All VMs with matching names automatically belong to the group. Nutanix uses the following naming convention for Control VMs:

   `NTNX-<SerialNumber>-A-CVM`, where `<SerialNumber>` is the serial number of the Controller VM.

   - Create a new group

     In Turbonomic go to the **Policy > Group Management** view and create a new group that groups VM entities by criteria.
   - Add a filter to match the VM names

     Add a filter that matches names by the regular expression, `NTNX.*CVM`. This regular expression will match the Nutanix Controller VMs.

   Be sure to save the group. All the Nutanix Controller VMs will automatically become members of this group.
2. Disable moves for all VMs in this group.

   - In Turbonomic go to the Policy > Action > VM view

- Set the scope to the group you made

  In the **Scope** column, expand **My Groups** and select the group you just made.
- Disable moves for this group

  In the **Parameter** column under **Action Mode Settings**, set the value to **Disabled**. This will override the global action mode.
- Save the action mode settings

  Be sure to click **Apply Settings Change**.

Orchestrator Targets

Turbonomic supports the Action Script orchestrator target.

Turbonomic uses remote servers to execute action scripts. Managing the processes remotely means that you do not install custom code on the Turbonomic server, which eliminates associated security risks at that point.

However, you are responsible for maintaining the security of your action script server, to ensure the integrity of your custom code.

Action Script Server

Resource Requirements for the Server

The remote server can be a VM or a container. The capacity you configure for the server depends entirely on the processes you intend to run on it. Turbonomic does not impose any special resource requirements on the server.

Configuring Command Execution

To support execution of your scripts, you must install any software that is necessary to run the scripts. This includes libraries, language processors, or other processes that your scripts will invoke.

Turbonomic invokes the scripts as commands on the server. The server must run an SSH service that you have cnfigured to support command execution and SFTP operations. At this time, Turbonomic has tested action scripts with the OpenSSH sshd daemon.

The standard port for SSH is 22. You can configure a different port, and provide that for admins who configure the server as an Action Script target.

Note that an action script can invoke any process you have deployed on the remote server. You do not have to run scripts per se. However, you must be able to invoke the processes from the command line. The script manifest gives Turbonomic the details it needs to build the command line invocation of each script.

Configuring the Action Script User Account

To execute the scripts on your server, Turbonomic logs on via a user account that is authorized to execute the scripts from the command line. You provide the user credentials when you configure the Action Script target. To support this interaction, the user account must meet the following requirements:

- Public Key

  The user must have a public key in the `.ssh/authorized_keys` file. When you configure the Action Script target, you provide this as the Private Token for the target.
- Security for the `.ssh` Directory

  The Action Script User should be the only user with authorized access. You should set file permissions to `600`.
- Supported Shells

  The Action Script User shell can be either the Bourne shell (usually at `/bin/sh`) or the Bourne-Again shell (usually at `/bin/bash`). Turbonomic passes parameters as it invokes your scripts. At this time it only supports script execution through these shells.

Handling Action Script Timeouts

Turbonomic limits script execution to 30 minutes. If a script exceeds this limit, Turbonomic sends a `SIGTERM` to terminate the execution of the process.

Note that Turbonomic does not make any other attempt to terminate a process. For example you could implement the script so it traps the `SIGTERM` and continues to run. The process should terminate at the soonest safe opportunity. However, if the process does not terminate, then you must implement some way to terminate it outside of Turbonomic. Note that a runaway process continues to use its execution thread. This can block other processes (action scripts or primary processes) if there are no more threads in the pool.

Adding Action Script Targets to Turbonomic

> **NOTE:** At this time, Action Script Targets must be added using the API. The Action Script target appears in the UI as a selection, but should not be used. For more information about the Turbonomic API and how to use it to add targets, see the Turbonomic API Guide.

The TargetApiInputDTO for this target has the following parameters:

nameOrAddress

IP or FQDN of the script execution server.

port

Port used to connect to the script execution server.

userid

Username used to connect to the script execution server.

privateKeyString

Path to the SSH private token corresponding to the user used to connect to the script execution server.

manifestPath

Path to the Action Script manifest file on the script execution server.

hostKey *(Optional)*

Path to the public key presented by the SSH server for host authentication; if not provided, the presented key will be accepted and integrated into the target definition for future use.

Use the `POST https://10.10.10.10/api/v3/targets` request to add this target. Send the constructed TargetApiInputDTO as the body of the request.

Example Input:

```
{
   "category":"Orchestrator",
   "inputFields":[
      {
         "name":"nameOrAddress",
         "value":"10.10.10.10"
      },
      {
         "name":"port",
         "value":"22"
      },
      {
         "name":"userid",
         "value":"testuser1"
      },
      {
         "name":"privateKeyString",
         "value":"path/to/private/key"
      },
      {
         "name":"manifestpath",
         "value":"/home/testuser1/action-script/manifest.json"
      },
      {
         "name":"hostKey",
         "value":"path/to/host/public/key"
      }
   ],
   "type":"Action Script"
}
```

The API will return the TargetApiDTO for the Action Script target upon success, or an error message upon failure.

Example Response:

```
{
   "category":"Orchestrator",
   "displayName":"10.10.10.10-/home/testuser1/action-script/manifest.json",
   "inputFields":[
      {
         "description":"IP or FQDNS for the Script Execution Server",
         "displayName":"Name or Address",
         "isMandatory":true,
         "isSecret":false,
         "isTargetDisplayName":false,
```

```
         "name":"nameOrAddress",
         "value":"10.10.10.10",
         "valueType":"STRING",
         "verificationRegex":".*"
      },
      {
         "defaultValue":"22",
         "description":"Port to use for the Script Execution Server",
         "displayName":"Port",
         "isMandatory":false,
         "isSecret":false,
         "isTargetDisplayName":false,
         "name":"port",
         "value":"22",
         "valueType":"STRING",
         "verificationRegex":".*"
      },
      {
         "description":"Userid to use to execute command on the Script Execution Serve
r",
         "displayName":"User ID",
         "isMandatory":true,
         "isSecret":false,
         "isTargetDisplayName":false,
         "name":"userid",
         "value":"testuser1",
         "valueType":"STRING",
         "verificationRegex":".*"
      },
      {
         "description":"SSH Private Token corresponding to the Userid",
         "displayName":"Private Token",
         "isMandatory":true,
         "isSecret":true,
         "isTargetDisplayName":false,
         "name":"privateKeyString",
         "valueType":"STRING",
         "verificationRegex":".*"
      },
      {
         "description":"File Path to the Action Script manifest file on the Execution S
erver",
         "displayName":"Script Path",
         "isMandatory":true,
         "isSecret":false,
         "isTargetDisplayName":false,
         "name":"manifestPath",
         "value":"/home/testuser1/action-script/manifest.json",
         "valueType":"STRING",
         "verificationRegex":".*"
      },
      {
         "description":"Public key presented by the SSH server for host authenticaion;
 if not provided, the presented key will be accepted and integrated into the target def
inition for future operations",
```

```
         "displayName":"Public Host Key",
         "isMandatory":false,
         "isSecret":false,
         "isTargetDisplayName":false,
         "name":"hostKey",
         "valueType":"STRING",
         "verificationRegex":".*"
      }
   ],
   "lastValidated":"2020-03-17T16:38:35Z",
   "readonly":false,
   "status":"Validated",
   "type":"Action Script",
   "uuid":"73383824653536"
}
```


ServiceNow

You can configure Turbonomic policies that log Turbonomic actions in your ServiceNow instance, and that submit actions for approval in ServiceNow workflows.

For more information concerning orchestration, see "Action Orchestration" in the *User Guide*.

> **NOTE:**
> When creating the action orchestration policy as explained in the section above, the scope of the policy must match the scope of the ServiceNow target.

Prerequisites

- A ServiceNow user with the `web_service_admin` role and the custom role `x_turbo_turbonomic.user` that is created during installation that can communicate with Turbonomic via the REST API.

Adding ServiceNow Targets

To add ServiceNow targets, select the **Orchestration > ServiceNow** option on the Target Configuration page and provide the following information:

- Address

  Hostname of the ServiceNow instance without the `http` or `https` protocols. For example, `dev-env-266.service-now.com`.
- Username

  Username for the account Turbonomic will use to connect to the ServiceNow instance
- Password

  Password for the account Turbonomic will use to connect to the ServiceNow instance
- Client ID

  The Client ID Turbonomic will use if `Use oAuth` is checked
- Client Secret

  The password Turbonomic will use if `Use oAuth` is checked

- Port

  Port used to access the ServiceNow Instance
- Use oAuth

  When checked, Turbonomic will use oAuth authentication to connect to the ServiceNow target
- Proxy Host

  IP address of the proxy server
- Proxy Port

  Port used to access the proxy
- Proxy User

  Username for the account Turbonomic will use to connect to the proxy
- Port

  Port used to access the ServiceNow Instance

Virtual Desktop Infrastructure Targets

A virtual desktop infrastructure target is a service that delivers digital workspaces to multiple application users through a single portal. When you connect Turbonomic to VDI targets, it monitors the performance and resource consumption of the business users utilizing the virtual desktops and the virtual desktop infrastructure itself, enhancing the data used to assure application performance while also utilizing resources as efficiently as possible.

As part of this process, Turbonomic will use this information to provide deeper insight into the state of the hardware, and information related to the users and virtual desktops running in your environment.

Supported Virtual Desktop Infrastructure Targets

Turbonomic supports the following virtual desktop infrastructure targets:

- VMware Horizon

Monitored Resources

Turbonomic monitors the following resources for the VDI supply chain:

Entity Type	Commodity
Business User	• ImageCPU CPU utilization, as a percentage of CPU capacity for the user's desktop image or images. • ImageMem Memory utilization, as a percentage of Memory capacity for the user's desktop image or images. • ImageStorage Storage utilization, as a percentage of storage capacity for the user's desktop image or images.
Desktop Pool	• Pool CPU The CPU available to the pool that is in use by active sessions.

Entity Type	Commodity
	• Pool Memory The memory available to the pool that is in use by active sessions. • Pool Storage The storage capacity available to the pool that is in use by active sessions. • Active Sessions How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.

Horizon

VMware Horizon provides end users access to all of their virtual desktops, applications, and online services through a single digital workspace..

With the information provided by Horizon, Turbonomic efficiently utilizes the VDI infrastructure to avoid inappropriately placed or sized desktops, ensure users are matched to desktop images that meet their demand, while gaining valuable insight into the interconnected nature of your environment.

Prerequisites

- Horizon 7.0 or higher
- The version of the vCenter target associated to the Horizon target must be 5.0+

Adding Horizon Targets

> **NOTE:**
> For each Horizon target, you must add only one connection server servicing that site. Adding multiple connection servers for the same site will result in inconsistent and/or duplicate actions.

To add a Horizon target, select the **Virtual Desktop Infrastructure > Horizon** option on the Target Configuration page and provide the following information:

- Address

  The IP address of the Horizon connection server.
- Username

  Username for the user account Turbonomic will use to connect to the Horizon target. This account must be an Administrator (Read-Only) account.
- Password

  Password for the user account Turbonomic will use to connect to the Horizon target.
- vCenter Username

  Username for the user account Turbonomic will use to connect to the vCenter target related to Horizon. Leave this blank if the Horizon and vCenter credentials are identical.
- vCenter Password

  Password for the user account Turbonomic will use to connect to the vCenter target related to Horizon. Leave this blank if the Horizon and vCenter credentials are identical.

- LDAP Server

  The address of the specific LDAP server Turbonomic will use in conjunction with the Horizon target. By default, Turbonomic will do a lookup based on the domain entered in the Username and select a healthy DNS server from the response.
- LDAP Server Port

  The port Turbonomic will use to connect to the specified LDAP server. By default, Turbonomic will use port 389 for unsecure connections, and port 636 for secure connections.
- Secure LDAP Connection

  When checked, Turbonomic will use SSL to communicate to the LDAP Server
- LDAP Username

  The username Turbonomic will use to connect to the LDAP Server, in the user principal name format to authenticate in the directory service through Kerberos. For example, `user.name@domain.com`
- LDAP Password

  The password Turbonomic will use to connect to the LDAP Server

Actions

Under some circumstances, you can have users who need smaller or larger desktop images. This appears as users with low or high utilization of the image resources, respectively. You may also need to alleviate congestion on the desktop pool at certain times of day, based on usage. In these cases, Turbonomic can recommend moving the business user to a different desktop pool that serves up larger images.

To support these actions, you must configure a Workload Placement policy that merges the desktop pools. To do this:

- Create a new Workload Placement policy
- Choose Merge as the policy type
- For the consumer type to merge, choose Desktop Pool
- Choose the pools that you want to merge
- Save the policy

For more information, see Creating Placement Policies in the Turbonomic User Guide.

Turbonomic recommends actions for the various entities of the Horizon target as follows:

Entity Type	Action
Business User	• Move Business User due to image congestion • Move Business User due to desktop pool congestion
Desktop Pool	Turbonomic does not recommend actions to perform on a desktop pool. Instead, it recommends actions to perform on the business users running active sessions in the pool.
View Pod	Turbonomic does not recommend actions to perform on a view pod. Instead, it recommends actions to perform on the business users running active sessions in the view pod.

Monitored Resources

Turbonomic monitors the following commodities of the Horizon target:

Entity Type	Commodity
Business User	• ImageCPU CPU utilization, as a percentage of CPU capacity for the user's desktop image or images. • ImageMem Memory utilization, as a percentage of Memory capacity for the user's desktop image or images. • ImageStorage Storage utilization, as a percentage of storage capacity for the user's desktop image or images.
Desktop Pool	• Pool CPU The CPU available to the pool that is in use by active sessions. • Pool Memory The memory available to the pool that is in use by active sessions. • Pool Storage The storage capacity available to the pool that is in use by active sessions. • Active Sessions How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.
View Pod	• Active Sessions How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.

Appendix — Target Configuration

This appendix contains topics that are related to configuring Turbonomic targets.

Cisco Unified Computing System

UCS Blade Provisioning

When managing a UCS fabric target, Turbonomic can provision any blade servers that are installed in a chassis but not currently in operation. If the workload demands more physical compute resources, Turbonomic can automatically direct UCS to provision a blade, or it can recommend that you provision a blade and you can execute the action from the To Do list. To enable this capability, you must perform two basic steps:

- Configure the way UCS and vCenter Server manage information as blades are provisioned

To enable Turbonomic to perform automatic provisioning of UCS blades, you must configure UCS to work with vCenter Server as they both manage resources such as server pools, policies, and Service Profile Templates. This ensures that as Turbonomic directs the UCS Manager to provision a new blade, vCenter Server will recognize that the new physical host is available. Once vCenter Server can recognize the new blade, Turbonomic can direct vCenter Server to move workloads onto the server.

Turbonomic provisions new blades based on the service profiles of operating blades. To enable this, the configuration must include Service Profile Templates, and the operating blades must be bound to these templates.

For information about configuration that enables automated provisioning of blades, see the Cisco Communities post, “UCS PowerTool and VMware PowerCLI automated management of Auto-deploy” at the following location:

https://communities.cisco.com/community/technology/datacenter/ucs_management/cisco_ucs_developed_integrations/blog/2013/09/16/ucs-powertool-and-vmware-powercli-automated-management-of-auto-deploy

This post includes a video that shows “a joint PowerShell integration utilizing both Cisco UCS PowerTool and VMware PowerCLI.” You can also download the scripts from this post and modify them as necessary for your deployment.

- Set the Host Provision action to Automate or Manual for the blade servers. By default, Turbonomic sets the Host Provision action to Recommend.

  For any hosts other than blade servers managed by UCS, Turbonomic cannot provision hosts automatically or manually. Instead, it recommends that you provision a host, and you then install the physical machine and wire it to the network.

  In a UCS deployment you can have blade servers installed in the chassis and ready to provision. In that case, Turbonomic can direct UCS to provision a new blade to meet workload demands. For these servers, you can set the Host Provision action to Automated or Manual.

  > **NOTE:**
  > It's important that you only set Automated or Manual host provisioning to UCS blades. If you set Host Provision to Automated for other types of hosts, Turbonomic will attempt to perform the action and fail. As a result, you might never see the recommendation to provision a new host of that type.

  Turbonomic groups blade servers by chassis. To restrict Automated or Manual settings to blade servers, use this group. You can set the action mode for all blade servers in your environment, or you can set the mode differently for individual chassis.

Enabling Collection of Memory Statistics: AWS

So Turbonomic can collect memory statistics in your AWS environment, you must set up your VMs to publish statistics via CloudWatch. The steps to do this are different depending on the OS running on your VMs.

AWS Memory Statistics: Linux VMs

To enable memory statistics on Linux VMs, you must set up your VMs to publish metrics to CloudWatch. You must meet the following requirements:

- Your Linux image must have the SSM Agent installed

  By default, Linux AMIs dated 2017.09 and later include an installed SSM Agent.

- Access to the CloudWatch service

  Your AWS Instance must have internet access or direct access to CloudWatch so it can push data to CloudWatch.
- Access to AWS System Manager

  The user account must include an attacher AmazonEC2RoleforSSM Policy. As a minimum, the policy must have the following permissions:

  - AmazonEC2ReadOnlyAccess
  - AmazonS3ReadOnlyAccess
  - AmazonRDSReadOnlyAccess
  - AmazonEC2RoleforSSM
  - CloudWatchFullAccess
  - AmazonSSMFullAccess

In addition, you must install the CloudWatch agent on your VMs. To configure the agent, add the following JSON as the EC2 Parameter Store:

```
{
  "metrics": {
    "namespace": "Windows System",
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "aggregation_dimensions" : [ ["InstanceId"] ],
    "metrics_collected": {
      "Memory": {
        "measurement": [
          {"name": "% Committed Bytes In Use", "rename": "MemoryUsed", "unit": "Bytes"},
          {"name" : "Available Bytes", "rename": "MemoryAvailable", "unit": "Bytes"}
        ],
        "metrics_collection_interval": 60
      },
      "Paging File": {
        "measurement": [
          {"name": "% Usage", "rename": "paging_used"}
        ],
        "metrics_collection_interval": 60,
        "resources": [
          "*"
        ]
      }
    }
  }
}
```

For more information about enabling Windows memory statistics for AWS, see the Green Circle article, "Setting up CloudWatch with your Linux Instances on AWS".

AWS Memory Statistics: Windows VMs

To enable memory statistics on Windows VMs, you must enable AWS SSM:

- Ensure the AWS SSM Agent is installed on your VMs
- Create an IAM Role that supports AWS SSM. The role must have the following permissions at a minimum:
  - AmazonEC2ReadOnlyAccess
  - AmazonS3ReadOnlyAccess
  - AmazonRDSReadOnlyAccess
  - AmazonEC2RoleforSSM
  - CloudWatchFullAccess
  - AmazonSSMFullAccess
- Assign the IAM role to the VMs that you want to manage.

After you enable AWS SSM, you must deploy the following JSON file to configure CloudWatch.

```
        "IsEnabled": true,
        "EngineConfiguration": {
        "Components": [
        {
        "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.Performance
CounterInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "PerformanceCounter",
        "Parameters": {
        "CategoryName": "Memory",
        "CounterName": "Committed Bytes",
        "DimensionName": "InstanceId",
        "DimensionValue": "{instance_id}",
        "InstanceName": "",
        "MetricName": "MemoryUsed",
        "Unit": "bytes"
        }
        },
        {
        "FullName": "AWS.EC2.Windows.CloudWatch.PerformanceCounterComponent.Performance
CounterInputComponent,AWS.EC2.Windows.CloudWatch",
        "Id": "PerformanceCounter2",
        "Parameters": {
        "CategoryName": "Memory",
        "CounterName": "Available Bytes",
        "DimensionName": "InstanceId",
        "DimensionValue": "{instance_id}",
        "InstanceName": "",
        "MetricName": "MemoryAvailable",
        "Unit": "bytes"
        }
        },
        {
        "FullName": "AWS.EC2.Windows.CloudWatch.CloudWatch.CloudWatchOutputCompone
nt,AWS.EC2.Windows.CloudWatch",
        "Id": "CloudWatch",
        "Parameters": {
```

```
"AccessKey": "",
"NameSpace": "Windows System",
"Region": "",
"SecretKey": ""
}
}
],
"Flows": {
"Flows": [
"(PerformanceCounter,PerformanceCounter2), CloudWatch"
]
},
"PollInterval": "00:05:00"
}
}
```

For more information about enabling Windows memory statistics for AWS, see the Green Circle article, "AWS Memory Stats - Windows".

Enabling Collection of Memory Statistics: Azure

For Turbonomic to collect memory statistics in Azure, you must enable the collection of these statistics on the VMs in your environment. You can do this as you deploy your VMs, or you can enable the counters after the fact on VMs you have already deployed. For each VM, open the Azure Portal and navigate to Diagnostics Settings. Then enable the metrics for your VMs:

To enable the collection of memory statistics in Azure environments, open the Azure Portal, and then navigate to **Diagnostic Settings**. Then enable the metrics for your VMs:

- For Windows VMs

  Navigate to **Performance Counters**, display **Basic**, and enable the performance counters for the VM.
- For Linux VMs

  For supported Linux versions, Azure automatically deploys the Linux Diagnostics Extension v2.3 to gather these metrics. Refer to Microsoft Azure documentation for supported Linux OS versions. For unsupported versions, you can enable the statistics manually:

  1. Set **Status** to ON.
  2. For **Storage Account**, specify the storage that will retain the metric data.
  3. Enable **Basic Metrics** and then click **Save**.
  4. Navigate to **Metrics** in the Azure Portal and enable the metrics to collect.

For more information about enabling memory statistics for Azure, see the Green Circle article, "Enable Basic Metrics on Azure - Manually".

Enabling Windows Remote Management

Turbonomic communicates with your Hyper-V servers using Web Services Management (WS-Management), which is implemented on Microsoft platforms using Windows Remote Management (WinRM). The following steps show how to enable WinRM on a single host, using the command line.

1. Ensure Windows Firewall is running on the host.

   For you to configure WinRM successfully, Windows Firewall must be running on the host. For more information, see the Microsoft Knowledge Base article #2004640 (http://support.microsoft.com/kb/2004640).

2. Set up an SPN for the host machine.

   The machine must have an SPN of the form, `protocol/host_address`. For example, `WSMAN/10.99.9.2`.

   To get a list of SPNs for the machine, execute the following in the command window:

   ```
   setspn -l <vmm-server-name>
   ```

   If there is no valid SPN in the list, create one by running the command:

   `setspn -A protocol/host-address:port` where `port` is optional

   For example, `setspn -A WSMAN/10.99.9.2:VMM-02`

3. Set up the Windows Remote Management (WinRM) service to run on startup.

   Run the `quickconfig` utility to set up the WinRM service. The `quickconfig` utility:

   - Configures the WinRM service to auto-start
   - Configures basic authentication and disables unencrypted traffic
   - Creates a firewall exception for the current user profile
   - Configures a listener for HTTP and HTTPS on any IP address
   - Enables remote shell access

   To run `quickconfig`, log into a command window as Administrator on the host machine. Then execute the following commands:

   ```
   winrm quickconfig
   ```

   Enter `y` to accept the `quickconfig` changes

4. Set permissions on the host machine.

   Execute the following commands in the command window to modify the settings made by `quickconfig`:

   - To set the memory capacity for remote shells:

     ```
     winrm set winrm/config/winrs @{MaxMemoryPerShellMB="1024"}
     ```

   - To set up an unsecured HTTP connection:

     ```
     winrm set winrm/config/service @{AllowUnencrypted="true"}
     ```

     ```
     winrm set winrm/config/service/Auth @{Basic="true"}
     ```

These steps showed you how to enable WinRM for a single host. Some users find the following methods useful for enabling WinRM on multiple hosts:

- EnablingWinRmViaGlobal Policy Objects *(on page 131)*
- EnablingWinRMViaPowerShell *(on page 132)*

Enabling WinRM Via Global Policy Objects

You can configure WinRM for all of your Hyper-V targets by creating and linking a Global Policy Object (GPO) within the Hyper-V domain and applying the GPO to all servers.

Follow the steps below to enable Windows Remote Management (WinRM) for your Hyper-V targets.

For additional information, you can also see the Green Circle article, Application Discovery: GPO for necessary Registry Changes

1. On the AD domain controller, open the Group Policy Management Console (GPMC). If the GPMC is not installed, see https://technet.microsoft.com/en-us/library/cc725932.aspx.
2. Create a new Global Policy Object:
   a. In the GPMC tree, right-click **Group Policy Objects** within the domain containing your Hyper-V servers.
   b. Choose **Create a GPO in this domain,** and link it here.
   c. Enter a name for the new GPO and click **OK**.
3. Specify the computers that need access:
   a. Select the new GPO from the tree.
   b. On the **Scope** tab, under **Security Filtering**, specify the computer or group of computers you want to grant access. Make sure you include all of your Hyper-V targets.
4. Right-click the new GPO and choose **Edit** to open the Group Policy Management Editor.
5. Configure the WinRM Service:
   a. In the Group Policy Management Editor, select **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Management (WinRM) > WinRM Service**.
   b. Double-click each of following settings and configure as specified:

Allow automatic configuration of listeners (“Allow remote server management through WinRM” on older versions of Windows Server):	Enabled IPv4 filter: *
Allow Basic authentication:	Enabled
Allow unencrypted traffic:	Enabled

6. Configure the WinRM service to run automatically:
   a. In the Group Policy Management Editor, expand **Computer Configuration > Preferences > Control Panel Settings**.
   b. Under Control Panel Settings, right-click Services and choose **New > Service**.
   c. In the New Service Properties window, configure the following settings:

Startup:	Automatic
Service name:	WinRM
Service option:	Service start

7. Enable Windows Remote Shell:
   a. In the Group Policy Management Editor, select **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Shell**.
   b. Double-click the following setting and configure as specified:

Allow Remote Shell Access:	Enabled

8. Add a Windows Firewall exception:
   a. In the Group Policy Management Editor, open **Computer Configuration > Windows Settings > Security Settings > Windows Firewall > Windows Firewall**.
   b. Under Windows Firewall, right-click **Inbound Rules** and choose **New > Rule**.
   c. In the New Inbound Rule Wizard, select **Predefined: Windows Remote Management and Allow the connection**.

The new group policy will be applied during the next policy process update. To apply the new policy immediately, execute the following command at a Powershell prompt:

```
gpupdate /force
```

Enabling WinRM Via PowerShell

Using PsExec, you can run quickconfig on all your Hyper-V servers and change the default settings remotely. PsExec is a component of PsTools, which you can download from https://technet.microsoft.com/en-us/sysinternals/bb897553.aspx.

1. Create a text file containing the Hyper-V host names, for example:

   ```
   hp-vx485
   ```

   ```
   hp-vx486
   ```

2. Since Turbonomic requires changes to the default quickconfig settings, create a batch file containing the following command:

   ```
   @echo off Powershell.exe Set-WSManQuickConfig -Force Powershell.exe Set-Item WSMan:
   \localhost\Shell\MaxMemoryPerShellMB 1024
   ```

   > **NOTE:**
   > If you are connecting via HTTP, you must include the following command:
   >
   > ```
   > Powershell.exe Set-Item WSMan:\localhost\Service\AllowUnencrypted -Value $True
   > ```

3. Use PsExec to enable WinRM on the remote servers:

   ```
   .\PsExec.exe @<hosts_file_path> -u <username> -p <password> -c <batch_file_path>
   ```

   > **NOTE:**
   > If you get an error message when executing this command, add the `-h` option (`.\PsExec.exe -h`).

Secure Setup of WSMan

Turbonomic provides a secure option for Hyper-V/VMM Targets which requires that WSMan be set up securely. Use PowerShell to generate a self-signed certificate, and create an HTTPS WinRM listener.

> **NOTE:**
> For clustered Hyper-V targets, you do not need to create a listener on each host. Only create a listener on the host that is being added to the "Address" field in the Target Configuration.

To enable secure WSMan on your Hyper-V host:

1. Generate a self-signed certificate using the following command:

   ```
   New-SelfSignedCertificate -CertstoreLocation Cert:\LocalMachine\My -DnsName
   "myhost.example.org"
   ```

2. Find the thumbprint for the certificate for the host:

   ```
   Get-childItem cert:\LocalMachine\My
   ```

3. Create an HTTPS WinRM listener for the host with the thumbprint you've found:

   ```
   winrm create winrm/config/Listener?Address=*+Transport=HTTPS
   '@{Hostname="myhost.example.org"; CertificateThumbprint="THUMBPRINT_YOU_FOUND"}'
   ```

4. Verify the presence of configured listeners:

   ```
   Get-WSManInstance –ResourceURI winrm/config/listener –Enumerate
   ```

Sample OpenStack SNMP Configuration File for PM Metric Collection

> **NOTE:** This is a sample configuration file for the process found here *(on page 26)*. Please perform those steps before creating this file.

To use this configuration file, you must edit at least the following items:

- `<YourLocationName>`
- `<ContactName>`
- `<ContactEmailID>`

In addition, the line `disk /var/lib/nova/instances 15%` should be edited to `disk / 15%` if the VMs are on local storage.

```
###############################################################################
#
# snmpd.conf:
#   An example configuration file for configuring the ucd-snmp snmpd agent.
#
###############################################################################
#
# This file is intended to only be as a starting point.  Many more
# configuration directives exist than are mentioned in this file.  For
# full details, see the snmpd.conf(5) manual page.
#
# All lines beginning with a '#' are comments and are intended for you
# to read.  All other lines are configuration commands for the agent.

###############################################################################
# Access Control
###############################################################################

# As shipped, the snmpd demon will only respond to queries on the
# system mib group until this file is replaced or modified for
```

```
# security purposes.  Examples are shown below about how to increase the
# level of access.

# By far, the most common question I get about the agent is "why won't
# it work?", when really it should be "how do I configure the agent to
# allow me to access it?"
#
# By default, the agent responds to the "public" community for read
# only access, if run out of the box without any configuration file in
# place.  The following examples show you other ways of configuring
# the agent so that you can change the community names, and give
# yourself write access to the mib tree as well.
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.

####
# First, map the community name "public" into a "security name"

#       sec.name  source          community
com2sec local     default           public
com2sec mynetwork default      public

####
# Second, map the security name into a group name:

#       groupName      securityModel securityName
group MyRWGroup v1         local
group MyRWGroup v2c        local
group MyRWGroup usm        local
group MyROGroup v1         mynetwork
group MyROGroup v2c        mynetwork
group MyROGroup usm        mynetwork

####
# Third, create a view for us to let the group have rights to:

# Make at least  snmpwalk -v 1 localhost -c public system fast again.
#       name           incl/excl     subtree         mask(optional)
view all    included  .1                               80

####
# Finally, grant the group read-only access to the systemview view.

access MyROGroup ""      any       noauth    exact  all    none   none
access MyRWGroup ""      any       noauth    exact  all    all    none


###############################################################################
# System contact information
#

# It is also possible to set the sysContact and sysLocation system
# variables through the snmpd.conf file:
```

```
#syslocation Unknown (edit /etc/snmp/snmpd.conf)
#syscontact Root <root@localhost> (configure /etc/snmp/snmp.local.conf)

syslocation <YourLocationName>
syscontact <ContactName> <ContactEmailID>

# Example output of snmpwalk:
#   % snmpwalk -v 1 localhost -c public system
#   system.sysDescr.0 = "SunOS name sun4c"
#   system.sysObjectID.0 = UUID: enterprises.ucdavis.ucdSnmpAgent.sunos4
#   system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
#   system.sysContact.0 = "Me <me@somewhere.org>"
#   system.sysName.0 = "name"
#   system.sysLocation.0 = "Right here, right now."
#   system.sysServices.0 = 72

###############################################################################
# Logging
#

# We do not want annoying "Connection from UDP: " messages in syslog.
# If the following option is commented out, snmpd will print each incoming
# connection, which can be useful for debugging.

dontLogTCPWrappersConnects yes

# ----------------------------------------------------------------------------

###############################################################################
# disk checks
#

# The agent can check the amount of available disk space, and make
# sure it is above a set limit.

# disk PATH [MIN=100000]
#
# PATH:  mount path to the disk in question.
# MIN:   Disks with space below this value will have the Mib's errorFlag set.
#        Default value = 100000.

# Check the / partition and make sure it contains at least 10 megs.
# Note: Use '/' if the VMs are on local storage and
# '/var/lib/nova/instances' if the VMs are on shared storage

#disk / 15%
disk /var/lib/nova/instances 15%
```