



Turbonomic 7.17.1 Installation Guide

Turbonomic, Inc

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

COPYRIGHT

Copyright © 2010 - 2019 Turbonomic, Inc. All rights reserved

END-USER LICENSE AGREEMENT

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

Contents

Introduction	4
Minimum Requirements	5
Installing Turbonomic	7
Installing on VMware Systems.....	7
Deploying Turbonomic Kubernetes Nodes	9
Initial Deployment Configuration.....	15
Enabling Optional Components.....	16
Manual Deployment Using Helm.....	16
General Configuration Options	18
Specifying a Static IP Address (Optional).....	18
Synchronizing Time (Recommended).....	19
License Installation and First-time Login	21
Single Sign-On Authentication	23
Example of IdP Metadata.....	26
Disabling Single Sign-On.....	27
Support for Single Logout.....	28
Updating Turbonomic to a New Version	30
Appendix A: Turbonomic Components	32
Appendix B: vmtctl Command	34
Appendix C: What Are the Typical Settings for an IdP?	37



Introduction

Thank you for choosing Turbonomic, the Intelligent Workload Management solution for Cloud and Virtualized Environments. This guide gives you information you need to install Turbonomic in your virtual environment, install your license, and get started managing your resources.

If you have any questions, please see our support site at <https://support.turbonomic.com>.

Sincerely:

The Turbonomic Team



Minimum Requirements

The requirements for running a Turbonomic instance depend on the size of the environment you are managing. Turbonomic keeps a real-time representation of your environment in memory. The greater the number of entities to manage, and the greater the relationships between them, the more resources you need for the VM that runs Turbonomic. And as the VM requirements increase, so do the requirements for the physical machine that hosts the VM.

The requirements listed here are recommendations that you should keep in mind as you plan your Turbonomic deployment. After deploying, if you find that you need to change memory capacity for the VM, you can shut it down, make changes, and then power it up again to use the new capacity. If you need more compute resources to support the VM, you can always move it to a different host.

In most cases you can run Turbonomic on a host that meets the following minimum requirements:

Supported Hypervisors		Storage Requirements	CPUs
VMware	vCenter versions 5.5, 6.0, 6.5, and 6.7	1 TB or greater NOTE: Can be thin provisioned depending on the storage requirements.	4 vCPUs

The amount of memory the VM requires depends on the estimated number of entities that the Turbonomic instance will control. The following table gives some guidelines to estimate the memory requirement:

Number of Entities	Memory
15,000	16 GB
25,000	24 GB
50,000	32 GB
100,000	48 GB
200,000	64 GB

Turbonomic requires a browser capable of displaying HTML5 pages.

Turbonomic supports DHCP or static IP addressing. For information about using static IP addresses, see [Specifying a Static IP Address \(Optional\) \(page 18\)](#).



Installing Turbonomic

As you get started with Turbonomic, please note that for this release the only supported hypervisor for installation is vCenter.

Installing on VMware Systems

This download of the Turbonomic instance is in the `.OVA 1.0` format. vCenter Server versions 4.0 and earlier do not support the import of `.OVA 1.0` files.

If you must use one of these earlier versions of vCenter Server, please contact Turbonomic Support to receive the installation package in a compatible format.

To install Turbonomic:

1. Download the Turbonomic installation package.

The email you received from Turbonomic includes links to the Turbonomic download pages. You can get the installation package from there.

The installation package includes the `turbonomic_x1.ova` file, which deploys as a VM with the Turbonomic components already installed.

2. Import the OVA file into your datacenter.

Use the vCenter Server client to import the OVA into your environment.

3. Deploy the Turbonomic VM.

Create the VM using the `turbonomic_x1.ova`. Ensure that the physical machine hosting the VM meets the minimum requirements (see [Minimum Requirements \(page 5\)](#)).

After you have deployed the VM, power it on. If your environment supports DHCP, vCenter Server reports the VM's IP address. If vCenter does not display the IP address, open a Virtual Machine Console on the VM, and execute the command, `ifconfig eth0`. Make a note of this address for future reference.

NOTE:

When Turbonomic deploys, DHCP assigns it an IP address. You can complete the installation and log into the VM using that IP address. After you complete the installation, if you want you can then reconfigure the VM to use a static IP address.

If your environment does not support DHCP, then you must assign a static IP to the VM before you can complete the Installation procedure.

For instructions to set a static IP address, see [Specifying a Static IP Address \(Optional\) \(page 18\)](#).

4. Connect to the VM using HTTPS.

Using the provided IP address or the static IP address (if you configured one), navigate a browser to the VM. For example, if the IP is 10.10.10.10, navigate to `https://10.10.10.10`. This opens the Turbonomic initial configuration page.

5. Set up the Turbonomic System Administrator account.

To manage the Turbonomic VM, you need a system administrator account. When you first navigate to the VM, it posts a form for you to specify the name and password of the account that can log into shell sessions on the VM. Provide the username and password for this Turbonomic System Administrator account, and note them for future reference.

NOTE:

Be sure to save the System Administrator account name and password in a safe place. If you lose these account credentials, you will not be able to log into a shell session on the Turbonomic VM. For security reasons, this is the only account that can access and configure the Turbonomic VM.

In addition, you must provide a viable account name. You *cannot* use the account name, `root`. Also, the account name must begin with a lowercase letter.

After you submit the account credentials, Turbonomic begins to power up all the system components.

6. Log in as a user, and set the administrator user account password.

After the components power up, your browser redirects to `https://[MyIPAddress]/app/index.html#/authentication/login`. This is the login page for Turbonomic users.

Turbonomic includes a default user account named `administrator`. You cannot delete this account, and you must set your own password for it. Enter the information as required, and make a note of it - this is the account you will use to access the Turbonomic GUI with administrator permissions.

NOTE:

The initial login is always for the `administrator` account. This is an administration *user* account. Do not confuse this with the Turbonomic System Administrator account that you previously set up to log into shell sessions on the VM itself.

After you have logged in as `administrator`, you can create other user accounts, and you can give them various roles. For more information about user accounts and roles, see the *Turbonomic User Guide*.

7. Optionally, synchronize the system clock.

For more information, see [Synchronizing Time \(Recommended\) \(page 19\)](#).



Deploying Turbonomic Kubernetes Nodes

The first step for deploying Turbonomic on Kubernetes is to deploy one or more Kubernetes nodes that will host pods to run the Turbonomic components. The Turbonomic Kubernetes node deploys as a VM. The script to deploy and initialize the node also deploys the Kubernetes pods that make up the Turbonomic application.

To configure and deploy a Kubernetes node:

1. Acquire the OVA for the Kubernetes Turbonomic pod.

To get the Turbonomic Kubernetes node, contact your Turbonomic representative and request the OVA.

2. Install the OVA in your environment.

Use normal procedures to install the OVA. Make a note of the VM's IP address.

3. Initiate SSH access to the node VM.

After you deploy the node VM, log into an SSH session using the credentials:

- Username: `turbo`
- Password: `vmturbo`

After you log in, the session requires you to change the password. After you change the password, the VM ends the session. Log into a new SSH session for the `turbo` user, and give the new password you just created.

4. Set up the deployment configuration to suit your environment.

To install the node VM, you will run a script from the command line. Before you run that script, you must configure the setup to specify options such as:

- The IP address of your node VM
- A custom cluster storage device, if you want it
- Registry options

To edit the configuration file, open `/opt/local/etc/turbo.conf` in a text editor. For an example of the configuration file, with explanations for the settings, see [Initial Deployment Configuration \(page 15\)](#).

5. Initialize the Kubernetes node and deploy the pods.

After you have modified `turbo.conf`, you can then execute the initialization script. In the SSH session, execute the command, `/opt/local/bin/t8c.sh`.

This initializes the Kubernetes node and deploys the pods that make up the Turbonomic application. The script output should be similar to:

NAME	STATUS	ROLES	AGE	VERSION
node1	Ready	master,node	25d	v1.12.3

6. Verify that the the deployment succeeded.

Verify your deployment by executing the command, `helm list`. You should see output similar to the following:

```
xl-release 1 Thu Jan 31 22:32:20 2019 DEPLOYED xl-1.1.0 7.12 turbonomic
```

7. Verify that the Turbonomic application installed correctly.

To verify the installation of the application, execute the command:

```
kubectl get pods -n turbonomic -o wide.
```

You should see output similar to the following:

NAME	IP	NODE	READY	STATUS	RESTARTS	AGE
action-orchestrator-74b4c7964d-1l526	...	node1	1/1	Running	0	...
api-778b55cd6f-78qm4	...	node1	1/1	Running	0	...
arangodb-d768f8594-ws56j	...	node1	1/1	Running	0	...
auth-9bbf849bd-cjmgh	...	node1	0/1	Running	0	...
clustermgr-6d6694cc44-95pp2	...	node1	1/1	Running	3	...
consul-844b99f5fb-4hrd4	...	node1	1/1	Running	0	...
cost-58594fb888-2mf9f	...	node1	1/1	Running	0	...
db-54c879f44f-zswsz	...	node1	1/1	Running	0	...
grafana-7c78d8f454-jq5v6	...	node1	1/1	Running	0	...
group-cbfb69db4-pwcw6	...	node1	0/1	Running	1	...
history-64fbfc7ff5-cnggs	...	node1	1/1	Running	0	...
kafka-5867666f7d-97xgb	...	node1	1/1	Running	1	...

market-75575cb466-hvdvw	1/1	Running	0	...
... node1				
mediation-aws-7d77bc6d59-6jw9b	1/1	Running	2	...
... node1				
mediation-azure-67745dfcc9-95xzc	1/1	Running	1	...
... node1				
mediation-vcenter-5b64b548bc-5mxtw	1/1	Running	1	...
... node1				
nginx-7d66b9cf6b-2cgxj	1/1	Running	0	...
... node1				
plan-orchestrator-778748676d-fqgdw	1/1	Running	0	...
... node1				
prometheus-alertmanager-55fccf7648-57pjh	2/2	Running	0	...
... node1				
prometheus-kube-state-metrics-5544f68d59-cr7xh	1/1	Running	0	...
... node1				
prometheus-node-exporter-b25xv	1/1	Running	0	...
... node1				
prometheus-pushgateway-5d7dcdd5dc-s276h	1/1	Running	0	...
... node1				
prometheus-server-585c85c469-fkxdd	2/2	Running	0	...
... node1				
reporting-867c8b569-jdkzp	0/1	Running	2	...
... node1				
repository-6cc44bd8bc-fgfh4	1/1	Running	0	...
... node1				
rsyslog-554c5f5578-bvfv5	1/1	Running	0	...
... node1				
topology-processor-6c877459f-nj2wr	0/1	Running	2	...
... node1				
zookeeper-54794688df-tq2kb	1/1	Running	0	...
... node1				

8. Verify that the Load Balancer has installed correctly.

To verify the presence of the Load Balancer, execute the command:

```
kubectl get services -n turbonomic | grep LoadBalancer
```

You should see output similar to the following:

```

      nginx      LoadBalancer    10.10.10.10    10.10.10.11    443:32669/
TCP,80:32716/TCP    17h
    
```

9. Enable mediation.

For Turbonomic to manage your IT environment, it must attach to targets in your environment so it can perform discovery and execute actions. The combination of the processes of discovery and action execution is *mediation*. This release of Turbonomic on Kubernetes supports mediation through the following targets:

- Application Servers
 - IBM WebSphere Application Server 8.5+

- Oracle WebLogic 12c
- JBoss Application Server 6.3+
- Apache Tomcat 7.x, 8.x, and 8.5.x
- JVM 6.0+
- Cloud Managers
 - CloudStack 4.3 — 4.9
 - Microsoft System Center 2012 Virtual Machine Manager and System Center 2012 R2 Virtual Machine Manager
 - VMware vCloud Director 5.x — 8.2
 - OpenStack Havana — Newton
 - Amazon AWS
 - Microsoft Azure
 - IBM SoftLayer
- Database Servers
 - Microsoft SQL Server 2008 R2, 2012, 2014, and 2016
 - Oracle 11g R2 and 12c
 - MySQL all 5.6 and 5.7 releases
- Fabric Managers
 - Cisco UCS 3.1+
 - Cisco UCS Central
 - HPE OneView 3.00.04+
- Guest OS Processes
 - AppDynamics 4.1+
 - Docker API 1.20+
- Hyperconverged
 - Nutanix Community Edition
 - Hyperflex 2.5, 2.6
- Hypervisors
 - Citrix XenServer 5.6.x and 6.x
 - IBM PowerVM 8.3+
 - Microsoft Hyper-V 2008 R2, Hyper-V 2012, and Hyper-V 2012 R2
 - RHEV-M (RedHat Enterprise Virtualization Manager) versions 3.x

- VMware vCenter 5.1, 5.5, 6.0, 6.5, and 6.7
- Load Balancers
 - Citrix NetScaler NS10.5
- Microsoft Applications
 - Microsoft Exchange 2012
- Network Flow Collectors
 - NetFlow/sFlow: NFDUMP — Turbonomic provides an OVA download with NFDUMP preconfigured for NetFlow and sFlow collection
 - Arista Command eAPI
 - Cisco APIC
 - Cisco Tetration 2.2.1.31
- Orchestrator Targets
 - UCS Director 6.0 — 6.5
- PaaS Targets
 - CloudFoundry Pivotal / Stackato
 - OpenShift 3.3+
 - Mesos 1.0+
- Storage Managers
 - NetApp Cmode/7mode using ONTAP 8.0+ (excluding AFF and SolidFire)
 - EMC VMAX using SMI-S 8.1+
 - EMC VNX (Celera) using VNX Control Station
 - EMC VNX (Clarrion) using SMI-S 4.6.x
 - EMC VPLEX Local Architecture with 1:1 mapping of virtual volumes and LUNs
 - EMC XtremIO XMS 4.0+
 - Pure Storage F-series and M-series arrays
 - HP 3PAR InForm OS 3.2.2+, 3PAR SMI-S, 3PAR WSAPI
 - Dell Compellent Enterprise Manager 2014-2016R3 using Dell Compellent SMI-S
 - HDS (Embedded) with VSP, HUS VM, G600, G800, G1000
 - HDS (Shared) with HUS110, HUS 130, HUS 150
 - HDS Tuning Manager 9.x
- Turbonomic Targets (Classic UI Only)

To configure an aggregated deployment of Turbonomic, you can assign Turbonomic servers as targets. The versions of target instances must match the version of the aggregating instance.

For information about these targets, see the Turbonomic Target Configuration Guide.

Note that some of these targets have dependencies on others. To get the benefit of Turbonomic mediation, you must enable all of these mediation components. To enable mediation, `cd` to `/opt/turbonomic/kubernetes/helm` and then execute the following command:

```
helm upgrade xl-release xl --namespace turbonomic --set vcenter.enabled=true \
--set hyperv.enabled=true --set actionscript.enabled=true \
--set netapp.enabled=true --set pure.enabled=true --set oneview.enabled=true \
--set ucs.enabled=true --set hpe3par.enabled=true --set vmax.enabled=true \
--set vmm.enabled=true --set appdynamics.enabled=true --set snmp.enabled=true \
--set-string global.externalIP=[INSTANCE IP]
```

Where:

- `--namespace` is always `turbonomic`
- `global.externalIP` is the IP address of the node VM

10. Enable optional components.

This release of Turbonomic includes observability components to enhance the collection of metrics. For instructions to install these components, see [Enabling Observability Components \(page 16\)](#).

Deleting the Turbonomic Installation

To delete the Turbonomic installation, execute the following command:

```
helm delete --purge xl-release
```

Installing Multiple Instances of Turbonomic on a Cluster

In order to run multiple installations of XL on the cluster, execute the following command:

```
helm install xl --name xl-customer2 --namespace customer2
```

NOTE:

`--name` and `--namespace` must be unique for each installation.

Initial Deployment Configuration

Before you run the script that initializes the Kubernetes environment and deploys the production pods, you must edit the configuration file. See below for explanation and a sample file:

```
#####
#Turbonomic Configuration Parameters for Kubernetes Setup#
#####

# This will be read by the /opt/local/bin/t8c.sh script to
# setup Turbonomic xl installation

# Is this a new install, yes/no
installAnswer="yes" <== Currently the only option

# Enter the amount of nodes for the Kubernetes Cluster
nodeAnswer="1" <== Currently only a single node deploy is available

# Node Ip Addresses, if multiple, use space separator:
node="10.0.2.15" <== This must be edited to use the ip of the environment. TODO: Stati
c vs Dynamic IP

# Device added for cluster storage:
device="/dev/sdb" <== glusterfs, default 2nd disk. May be changed if using a different
shared storage

# Registry Information for pulling images from turbonomic (Do not modify)
dockerEmail="turbodeploy@localhost.localdomain"
dockerUserName="turbodeploy" <== Account to pull from registry.turbonomic.com
dockerPassword="vmturbo" <== Password for accounts

# Namespace
namespace="turbonomic" <== Should not be changed

# Registry options: localhost or registry.turbonomic.com
registry="registry.turbonomic.com" <== Use localhost if you have no access to the int
ernet. This will run the registry locally.

# Version
turboVersion="7.13.0" <== Version. Ensure that this matches what you expect.

# Environment "prod or dev"
turboEnv=prod <== Only relevant to developers
```

Enabling Optional Components

For your deployment of Turbonomic you can optionally deploy components to enhance metrics observation and data security.

Enabling Observability Components

Observability components enable enhanced collection of metrics among the entities in your environment. This release of Turbonomic on Kubernetes supports the following observability toolkits:

- Prometheus
- Grafana

To enable an observability component, execute the following command:

```
helm install xl --name xl-release --namespace turbonomic \
  --set imageCredentials.username=turbodeploy \
  --set imageCredentials.password=XXXXXX \
  --set prometheus.enabled=false --set grafana.enabled=false \
  --set-string global.externalIP=[INSTANCE IP]
```

Where:

- `--name` is the current release number of your Turbonomic application
- `--namespace` is always `turbonomic`
- `imageCredentials.username` is always `turbodeploy`
- `imageCredentials.password` is the password you assigned to your account on the node VM
- `prometheus.enabled` (or `grafana`) is `true` for the observability components you want to enable
- `global.externalIP` is the IP address of the node VM

To access these components, execute the command, `netstat -tulnp` and note the port number for the relevant process.

Manual Deployment Using Helm

When you initially deploy the Turbonomic Kubernetes node VM, the initialization script also executes helm to deploy the pods that make up the application. This section describes that deployment via helm.

To manually install Turbonomic via helm:

1. Navigate to the parent directory of the helm application manager.

On the node VM, navigate to `/opt/turbonomic/kubernetes/helm`.

2. Initialize helm.

To initialize helm, by executing the following commands:

```
kubectl create sa --n kube-system tiller
kubectl create clusterrolebinding tiller --clusterrole cluster-admin --
serviceaccount=kube-system:tiller
kubectl patch deploy --n kube-system tiller-deploy -p '{"spec":{"templ
ate":{"spec":{"serviceAccount":"tiller"}}}}'
helm init
```

3. Deploy Turbonomic

Execute the following using one of the following helm commands:

```
# Deploy without any customization
helm install xl --name xl-release --namespace turbonomic
```

```
# Deploy using a custom registry without authentication
helm install xl --name xl-release --namespace turbonomic --set global.repository=r
egistry.company.com/turbonomic
--set global.tag=7.13.0-SNAPSHOT
```

```
# Deploy using a specific external IP (on-prem)
$ helm install xl --name xl-release --namespace turbonomic --set-string global.ext
ernalIP=10.10.10.10
```

4. After you execute the deployment, verify the results. For information, see the verification steps in [Deploying Kubernetes Nodes \(page 9\)](#).



General Configuration Options

After you install Turbonomic, you can assign a static IP address to the VM if you did not do so during the installation process. You should also configure the machine's clock.

Specifying a Static IP Address (Optional)

Many installations use DHCP for dynamic IP address allocation. You can also specify a static address via the virtual machine's IP configuration. If you want to configure a static IP address before you have completed the installation process, you can log into the VM through the hypervisor console, using the default `neteditor` account.

1. Start a secure session on your Turbonomic VM.

If you have completed installation, log into the VM via SSH or the hypervisor console. If you already set up the Turbonomic System Administrator account as part of the installation steps, log in using those credentials.

If you have not completed the installation of your Turbonomic VM, open up a session on the VM in the hypervisor console. To log in, use the default `neteditor` account. Note that you will replace this account with your own credentials as part of the installation process. This default account is set up so you can configure a static IP address before you complete the installation process. To log into the `neteditor` account, use the credentials:

- User: `neteditor`
- Password: `vmturbo`

2. Start the NetworkManager session.

After you log into the `neteditor` account, the session prompts you with the option:

```
Do you need to edit the connection [y/n]:
```

Choose `y` to start up the NetworkManager user interface.

If you do not receive this prompt, execute the command, `nmtui` to start up the NetworkManager user interface.

3. Edit the `eth0` ethernet connection.

In NetworkManager, select the `Edit . . .` option and select `<OK>` to open the Edit a Connection dialog.

Then select `Ethernet` from the list of options and fill out the following information (values given are examples only).

- Profile Name: `eth0`
- Device: `eth0`

Then show `IPv4 CONFIGURATION` and fill out the configuration subsettings based on your environment:

- IPv4 Configuration: `<Manual>`
- Addresses: Give the IP address you want, with the subnet mask on the same line. For example:
`10.0.254.10/24`
- Gateway, DNS Servers, and Search domains: Give values that are valid for your network environment.

When you are satisfied with your settings, select `<OK>` to return to the configuration list. Verify that the connection you just created is present.

4. Exit NetworkManager.

Select `<Quit>` to return to the command line.

5. Restart the Turbonomic VM.

To finish setting up the static IP address, restart the VM. The network service should restart successfully and your Turbonomic VM should now be reachable at the IP address that you configured. To confirm this, ping the static IP address.

After you complete your installation and create your System Administrator account, you can open a secure session on the VM and execute the command, `ifconfig eth0` (where `eth0` is your new connection).

Synchronizing Time (Recommended)

It is important that you synchronize the clock on the Turbonomic instance with the devices on the same network. You will specify the servers that Turbonomic will use to synchronize its clock.

You should also set the system clock to your current time zone. Turbonomic runs regular data maintenance processes, and to minimize performance impact it runs these processes at night. To ensure that these processes run at the proper local time, you should synchronize the VM with your local time zone.

To synchronize the clock on the Turbonomic instance:

1. If it exists, back up your existing localtime config file.

Execute the command: `sudo mv /etc/localtime /etc/localtime.bak`

2. Choose the zoneinfo file that matches your timezone.

Select the file from the `/usr/share/zoneinfo` directory that matches your timezone. To see the files, execute the command: `sudo ls usr/share/zoneinfo`.

3. Link to the timezone file to your localtime directory.

Create a symlink between your selected file (America/Chicago, in this example) and the localtime file. Execute the command:

```
sudo ln -s -f /usr/share/zoneinfo/America/Chicago /etc/localtime
```

4. Verify that the new time settings have taken effect.

Execute the `date` command. You should see results similar to:

```
Thu Feb 2 14:25:45 CST 2018
```

To set up chrony on your Turbonomic instance:

1. Open an SSH terminal session to your Turbonomic instance.

2. Open the chrony configuration file.

For example, execute the command: `vi /etc/chrony.conf`

3. Replace the timeservers at the bottom of the file with your timeservers or use the default CentOS servers.

4. Save the file.

5. Restart the chrony service.

Execute the command: `sudo systemctl restart chronyd`

6. Verify that your time is still correct.

Execute the `date` command. You should see results similar to:

```
Thu Feb 2 14:25:45 CST 2018
```



License Installation and First-time Login

Before you begin, make sure you have your full or trial license key file that was sent to you in a separate email. Save the license file on your local machine so you can upload it to your Turbonomic installation.

To use Turbonomic for the first time, perform the following steps:

1. Type the IP address of your installed Turbonomic instance in a Web browser to connect to it.
2. Log in to Turbonomic.
 - Use the default credential for **USERNAME**: administrator.
 - Type a password for **PASSWORD**.
 - Type the password again to verify it for **REPEAT PASSWORD**.
 - Click **CONFIGURE**.
3. Decide whether to enable Usage Data and Analytics.
Click **AGREE** or **No**.
You can always change this setting later. For information, see "Administrative Tasks" in the *Turbonomic User Guide*.
4. Continue setting up your Turbonomic installation.
Click **LET'S GO**.
5. Open the **Enter License** fly-out.
Click **UPDATE LICENSE**.
6. Upload your license key file.
 - a. In the Enter License fly-out, you can upload the license in one of the following ways:
 - Drag the license key file into the Enter License fly-out.
 - Browse to the license key file.

Be sure to upload only .xml or .lic files.

- b. Click **SAVE**.

Depending on which license you have installed, the license enables either a trial or a full unlimited license for Turbonomic.



Single Sign-On Authentication

If your company policy supports Single Sign-On (SSO) authentication, Turbonomic enables SSO authentication by using Security Assertion Markup Language (SAML) 2.0.

At a high-level, the process involves:

- Creating external groups or at least one external user for SSO. See "Managing User Accounts" in the *Turbonomic User Guide*.
- Configuring Turbonomic to connect to the SAML Identity Provider (IdP). See [Configuring Single Sign-On \(page 24\)](#).

When SSO is enabled, use your SSO credentials to log in to your Turbonomic instance. Do not use your local or Active Directory (AD) credentials for the login. The Identity Provider (IdP) will perform the authentication.

NOTE:

When you enable SSO, Turbonomic only accepts authentication from the IdP you configure. Remote requests via the Turbonomic REST API do not use SSO. For security reasons, those REST API requests will not execute while SSO is enabled. As a result, integrations which use the REST API will not work on the Turbonomic instance where SSO is enabled.

Prerequisites

Before you begin, make sure the IdP is set up for SSO. You can use a proprietary or public IdP. For examples of settings for a public Okta IdP, see [What Are the Typical Settings for an IdP? \(page 37\)](#).

Configuring Single Sign-On

To configure Single Sign-On, perform these steps:

1. (Required) Create external groups or at least one external user for SSO.

IMPORTANT:

When SSO is enabled, Turbonomic only permits logins via the SSO IdP. Whenever you navigate to your Turbonomic installation, it redirects you to the SSO Identity Provider (IdP) for authentication before displaying the Turbonomic user interface.

Before you enable SSO for your Turbonomic installation, *you must configure at least one SSO user with Turbonomic administrator privileges*. If you do not, then once you enable SSO you will not be able to configure any SSO users in Turbonomic. To authorize an SSO user as an administrator, use **EXTERNAL AUTHENTICATION** to do one of the following:

- Configure a single SSO user with administrator authorization.
Add an external user. The username must match an account that is managed by the IdP.
- Configure an SSO user group with administrator authorization.
Add an external group. The group name must match a user group on the IdP, and that group must have at least one member.

For information about creating external groups or external users for SSO, see "Managing User Accounts" in the *Turbonomic User Guide*.

2. Obtain a JSESSIONID cookie and make a record of it.

To obtain a JSESSIONID cookie, use this curl command:

```
curl -k -v 'https://<IP_Address>/vmturbo/rest/login' --data  
'username=administrator&password=<my_password>'
```

where <IP_Address> is the IP address or host name of your Turbonomic instance and <my_password> is the password for your Turbonomic administrator user account.

For example: `curl -k -v 'https://10.10.10.123/vmturbo/rest/login' --data 'username=administrator&password=welcome57!'`

3. (Required) Ensure that chrony is configured and the system time on your Turbonomic instance is correct.

For instructions, see [Synchronizing Time \(Recommended\) \(page 19\)](#).

4. Obtain the metadata from your IdP.

- a. Contact your security administrator to obtain the metadata from IdP.
- b. Save the metadata file in a directory on your local machine.
- c. Compare your metadata to the sample provided in [Example of IdP Metadata \(page 26\)](#).

If your metadata includes optional attribute tags that are not listed in the example, remove those optional attribute tags since they are not supported.

d. (Required) Name the metadata file: `metadata.xml`

5. Navigate to the local directory where you saved the `metadata.xml` file.

6. Upload the IdP metadata to import it.

Run this curl command from the local directory where you saved the `metadata.xml` file:

```
curl -v -k --cookie 'JSESSIONID=<jsessionid_value>' -X POST -F file=@metadata.xml
https://<IP_Address>/vmturbo/rest/saml/idpmetadata
```

where `<jsessionid_value>` is the JSESSION cookie that you previously obtained and `<IP_Address>` is the IP address or host name of your Turbonomic instance.

For example: `curl -v -k --cookie 'JSESSIONID=node0p61kwex749dcw5ipbne4rm970.node0' -X POST -F file=@metadata.xml https://10.10.10.123/vmturbo/rest/saml/idpmetadata`

7. Obtain a key store to support IdP.

Generate a key store for SAML that supports IdP by using a security utility like Java SDK keytool.

NOTE:

During the process to generate a key store, you also create an alias. For the alias password, enter the same password that you used for the key store.

8. Save the key store file in the same local directory where you saved the `metadata.xml` file.

9. Obtain a certificate from IdP.

Contact your security administrator to obtain a certificate from IdP.

10. Import the IdP certificate into the key store.

11. Install the key store.

Run the curl command from the local directory where you saved the key store file:

```
curl -v -k --cookie 'JSESSIONID=<jsessionid_value>' -X POST -F
file=@<keystore_file_for_saml> https://<IP_Address>/vmturbo/rest/saml/keystore
```

where `<jsessionid_value>` is the JSESSION cookie that you obtained, `<keystore_file_for_saml>` is your key store file for SAML, and `<IP_Address>` is the IP address or host name of your Turbonomic instance.

For example: `curl -v -k --cookie 'JSESSIONID=node0p61kwex749dcw5ipbne4rm970.node0' -X POST -F file=@samlKeystore.jks https://10.10.10.123/vmturbo/rest/saml/keystore`

12. Update the SAML configuration.

Use curl to update the values for the SAML configuration:

```
curl -k 'https://<IP_Address>/vmturbo/rest/saml' -H 'cookie:
JSESSIONID=<jsessionid_value>' -H 'content-type: application/json' --data-binary
'{"entityId": "<IdP_Audience_Restriction>", "enabled": true, "externalIP": "<IP_Address>",
"password": "<keystore_password>", "alias": "<priv_key_alias>"}
```

where `<IP_Address>` is the IP address or host name of your Turbonomic instance. For the `--data-binary` parameters, specify the following values:

- `entityId`: The IdP's Audience Restriction property
- `enabled`: Type `true` to enable SAML.

- externalIP: The host name or IP address of your Turbonomic instance
- password: The key store password
- alias: The alias for the private key

For example:

```
curl -k 'https://10.10.10.123/vmturbo/rest/saml' -H 'cookie:
JSESSIONID=node0p6lkwex749dcw5ipbne4rm970.node0' -H 'content-type: application/
json' --data-binary '{"entityId": "turbo", "enabled": true, "externalIP": "10.10.10.123",
"password": "nalle123", "alias": "apollo"}'
```

13. Restart the API component.

- Open an SSH terminal session to your Turbonomic instance.

- Use sudo as root.

```
sudo bash
```

- Restart your API component.

```
turboctl restart api
```

14. Verify that the configuration is successful.

- Navigate to the Turbonomic User Interface.

You will be automatically redirected to your IdP for authentication.

- Log in with the username that is a member of the external group or external user previously configured.

Example of IdP Metadata

This section provides an example of IdP metadata which may be useful when you are examining the optional attributes in your metadata.

If your metadata includes optional attribute tags that are not listed in the example, remove those optional attribute tags since they are not supported.

```
<?xml version="1.0" encoding="UTF-8"?>
  <md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    entityID="http://www.okta.com/exkexl6xc9MhzqiC30h7">
    <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <md:KeyDescriptor use="signing">
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <ds:X509Data>
            <ds:X509Certificate>
              MIIDpDCCAoygAwIBAgIGAWMnhv7cMA0GCSqGSIb3DQEBCwUAMIGSMQswCQYDVQQGEwJVUzETMBEG
              A1UECAwKQ2FsaWZvcmlkZXIxEzARBgnVBAMMCMRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
              MBIGAlUECwwLU1NPUHJvdmlkZXIxEzARBgnVBAMMCMRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
              DWluZm9Ab2t0YS5jb20wHhcNMjgwNTAzMTk0MTI4WhcNMjgwNTAzMTk0MjI4WjCBKjELMAKGA1UE
              BhMCVVMxEzARBgnVBAGMCKNhbg1mb3JuaWEwExFjAUBgnVBACMDVNHbiBGcmFuY2lzY28xDTALBgNV
```

```

BAoMBE9rdGExFDASBgNVBAsMC1NTTlByb3ZpZGVyMRMwEQYDVQDDApkZXYtNzcxMjAyMRwwGgYJ
KoZlhvcNAQkBFglpbmZvQG9rdGEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
ugxQGqHAXpjVQZwsO9n8l8bFCoEevH3AZbz7568XuQm6MK6h7/09wB4C5oUYddemt5t2Kc8GRhf3
BDXX5MVZ8G9AUpG1MSqe1CLV2J96rMnwMIJSKeRrXr01LYxv/J4k jnktpOC389wmcy2fE4RbPoJne
P4u2b32c2/V7xsJ7UEjPPSD4i8l2QG6qsUkkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
PTpzek8NS5M5rvF8yaD+eElzS0I/HicHbPOVvLal0JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
nn4XW9Cqjk3equSCJPo5o1Msj8v1LrJYVarqhwIDAQABMA0GCSqGS Ib3DQEBCwUAA4IBAQC26kYe
LgqjIkF5rvxB2QzTgcd0LVzXOuiVVTZr8Sh5714jjqbDoIgvAQrxRSQzD/X+hcmhuwdp9s8zPHS
JagtUJXiypwNtrzb6M71trWB9sdNrqc99dlgOVRr0Kt5pLTaLe5kkq7dRaQoOIVIJhX9wgynaAK
HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70EOwp
6TaluRLQh6jCCOCWRuZbbz2T3/sOX+sibC4rLilwfyTkUopF/bTSdWwknORskK4dBekFcvN9N+C
p/qaHYcQd6i2vyor888DLHDPXhSKWhpG
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:Name
eIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:Name
ameIDFormat>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POS
T"
Location="https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkex
16xc9MhzqiC30h7/sso/saml"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Red
irect"
Location="https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkex
16xc9MhzqiC30h7/sso/saml"/>
</md:IDPSSODescriptor>
</md:EntityDescriptor>

```

Disabling Single Sign-On

If for some reason you no longer want to use SSO, you can disable it for your Turbonomic installation. To disable Single Sign-On, perform these steps:

1. Update the SAML configuration to disable it.

Use curl to update the values for the SAML configuration:

```

curl -k 'https://<IP_Address>/vmturbo/rest/saml' -H 'cookie:
JSESSIONID=<jsessionid_value>' -H 'content-type: application/json' --data-binary
'{"entityId": "<IdP_Audience_Restriction>", "enabled": false, "externalIP": "<IP_Address>",
"password": "<keystore_password>", "alias": "<priv_key_alias>"}'

```

where <IP_Address> is the IP address or host name of your Turbonomic instance.

For the `--data-binary` parameters, specify the following values:

- `entityId`: The IdP's Audience Restriction property
- `enabled`: Type `false` to disable SAML.
- `externalIP`: The host name or IP address of your Turbonomic instance
- `password`: The key store password
- `alias`: The alias for the private key

For example:

```
curl -k 'https://10.10.10.123/vmturbo/rest/saml' -H 'cookie:
JSESSIONID=node0p61kwex749dcw5ipbne4rm970.node0' -H 'content-type: application/json'
--data-binary '{"entityId":"turbo","enabled":false,"externalIP":"10.10.10.123",
"password":"nalle123", "alias": "apollo"}'
```

2. Restart the API component.

- a. Open an SSH terminal session to your Turbonomic instance.

- b. Use `sudo` as root.

```
sudo bash
```

- c. Restart your API component.

```
turboctl restart api
```

3. Verify that the configuration is successful.

- a. Navigate to the Turbonomic User Interface.

You will no longer be redirected to your IdP for authentication. You will be redirected to the default Turbonomic login screen.

- b. Log in with a local account or an Active Directory (AD) account.

Support for Single Logout

If you are using the SSO feature, Turbonomic supports the Single Logout feature provided by Security Assertion Markup Language (SAML) 2.0. When you click **Logout** in the Turbonomic session that has SSO enabled, the SAML 2.0 Single Logout feature terminates the Turbonomic session, the browser session, the Identity Provider (IdP) session, and sessions at other Service Providers (SP) connected to the same IdP session.

If you want to use this feature, contact your security administrator to configure it.

The following are requirements:

- The `Single Logout` setting must be enabled on the IdP.
- The IdP needs to trust the Turbonomic SAML key store certificate.

If the IdP does not enable or support Single Logout, you need to manually log out from the IdP to fully log out from Turbonomic.

If you close the browser without clicking **Logout** or if your browser session times out, you can log in again provided the Turbonomic or the IDP session is valid.



Updating Turbonomic to a New Version

NOTE:

Starting with version 7.3 of Turbonomic, you can perform updates to new versions as they become available. If you are running an earlier version (7.2 or earlier), you cannot perform an update – To move to a newer version, you must perform a full installation. For installation instructions, see [Installing Turbonomic \(page 7\)](#).

Turbonomic continually and rapidly innovates and improves all aspects of this product. This means that Turbonomic periodically releases newer versions of this product. You should check regularly to see if a new version is available.

When a new version is available, it is important to properly update your existing installed instance, rather than just install a new one. When you first installed Turbonomic, you put into place sophisticated data collection and analysis processes. Internal to the installation is an integrated database that retains performance data from across your virtual environment. Turbonomic uses this historical data for right-sizing, projecting trends, and other analysis. This means that the database is important to Turbonomic *and becomes more so over time*. Properly updating your installation of Turbonomic preserves the database for continued use.

Before you begin the update procedure, make sure you have the email that Turbonomic sent to you with links to the Turbonomic .OVA file and to the ISO image.

To perform an update of your Turbonomic installation:

1. Save a snapshot of your current Turbonomic VM.

Before updating, you should properly shut down (not power off) the Turbonomic VM and perform a snapshot (or clone the VM). This provides a reliable restore point you can turn to in the event that trouble occurs during the update. After you have the snapshot, bring the VM back online.

2. Attach the ISO image to the VM that runs Turbonomic.

Refer to the email you received from Turbonomic for links to the Turbonomic .OVA file and to the ISO image.

3. Log in to the Turbonomic VM.

Use SSH to log in to the Turbonomic VM or log in using the vCenter console.

4. Execute the command to be a super user.

```
sudo bash
```

5. Change to the `/docker` directory.

```
cd /etc/docker
```

6. Execute the script to update Turbonomic.

```
./turboupgrade.py start
```

If you receive a message that a newer version of the tool is available, then run the script again.

Wait until the script is finished. You should see the message:

```
XL upgrade finished successfully
```

7. Refresh your browser.

After refreshing, you have full access to Turbonomic features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Lists of Pending Actions will not show any actions until after a full market cycle.

8. Notify other users to refresh their Turbonomic browser sessions.



Appendix A: Turbonomic Components

The following listings show the components that Turbonomic creates as part of the installation process.

Core Components For Turbonomic

- `rsyslog`
- `nginx`
- `consul`
- `db`
- `zoo1`
- `kafka1`
- `auth`
- `clustermgr`
- `api`
- `market`
- `action-orchestrator`
- `topology-processor`
- `arangodb`
- `repository`
- `group`
- `history`
- `plan-orchestrator`

- reporting

Mediation Components For Turbonomic

- mediation-vcenter
- mediation-hyperv
- mediation-vmm
- mediation-vmax
- mediation-ucs
- mediation-netapp



Appendix B: vmtctl Command

Turbonomic utilizes a custom management wrapper for `docker-compose` and `docker` functions. To invoke the command, run `vmtctl` as `sudo`. This resolves the `docker` dependencies, unlike the basic `docker-compose` command.

The `vmtctl` command has the following options:

- `vmtctl info`
Displays information about the installed XL distribution.

Sample output:

```
Turbonomic distribution
-----
Built on: Fri Jan 27 17:32:44 EST 2017
Version: XL 1.0
Build #: 49998

Images distributed
-----
e81f78f6989a9e90c2a6fe6ae26e9870e55de335550eebd41f3173810f25 docker-compo
se.yml
2a8c96a4f64e54370f535dcd72750a6778c23544842f4974507b5b9aedd action-orchestra
tor.tgz
ea43944a57100eb0a357affca995784dec2229f021bd54fab562f318158a api.tgz
a94c029c5ad46ee99bb07204ff051d7461687501ecab7cd06ff363bb6882 arangodb.tgz
58a51b27d3ea5dced9bba39891c77e40849bb5fc8c0a8eac17ea5d0acc75 auth.tgz
9be8e261198d274dd25623b9077417f761fa7e2e99529d4c5bca66aeb122 clustermgr.tgz
2df8406d279becf75dd4deff3b46f7d10e31e2b848a93d40a0ed71fb45a5 consul.tgz
024af3fbbf4dc2a6627720ab56048d8e9793cc9bd476c63a4d02084ee9e1 db.tgz
61266bf127a92ed50a912206600f94dff845d05b9dbb9bd0333ddb525d2f group.tgz
61ab89cbaee476920ebde85abf8a7489a11869034ab28367b30e32f99cba history.tgz
7fcbdfa000a0425b7f7720ec57a3a46cf46b63e0bd77f48c1042646a622f market.tgz
c1ec1f7dfd15435a7228c6f565972468facd7bc235183e43da5ac4bcc391 mediation_vcent
er.tgz
daacb8ea74def95466508bd6d1c9533fbee237f7701a59cd7e57544a0490 plan_orchestrato
r.tgz
```

```
51c111acfe02ebe654c5d6b96aafe5fdcc990b7dd90d1d3b01ca7a2c7e95 repository.tgz
46368d791e3b4e2006e471fcd33430859efe7c12d58c6a30064b4616b7e9 topology-process
or.tgz
```

- `vmtctl stats`

Displays usage statistics for all running containers.

Sample output:

CONTAINER	NET I/O	CPU %	MEM USAGE / LIMIT	MEM %
docker_api_1	194.2 MB / 32.5 MB	0.25%	686.3 MiB / 15.51 GiB	4.32%
docker_mediation-vcenter_1	664 MB / 106.3 MB	0.25%	854.6 MiB / 15.51 GiB	5.38%
docker_history_1	260.7 MB / 148.2 MB	0.34%	590 MiB / 15.51 GiB	3.71%
docker_group_1	98 MB / 18.79 MB	0.24%	544.2 MiB / 15.51 GiB	3.43%
docker_action-orchestrator_1	179.4 MB / 21.82 MB	0.62%	580 MiB / 15.51 GiB	3.65%
docker_plan-orchestrator_1	194.4 MB / 20.84 MB	0.14%	626.3 MiB / 15.51 GiB	3.94%
docker_repository_1	135.7 MB / 153.7 MB	0.65%	586.2 MiB / 15.51 GiB	3.69%
docker_market_1	78.01 MB / 62.14 MB	0.33%	569.2 MiB / 15.51 GiB	3.58%
docker_topology-processor_1	121.1 MB / 122.3 MB	0.33%	2.378 GiB / 15.51 GiB	15.33%
docker_auth_1	76.8 MB / 19.79 MB	0.33%	570.2 MiB / 15.51 GiB	3.59%
docker_arangodb_1	133.8 MB / 10.85 MB	0.44%	605.5 MiB / 15.51 GiB	3.81%
docker_db_1	127.9 MB / 23.8 MB	0.16%	719.6 MiB / 15.51 GiB	4.53%
docker_clustermgr_1	278.4 kB / 118.4 kB	0.25%	457.8 MiB / 15.51 GiB	2.88%
docker_consul_1	212.5 MB / 1.249 GB	1.23%	34.74 MiB / 15.51 GiB	0.22%

- `vmtctl logs`

Tails the Turbonomic logfile.

Sample Output:

```
arangodb_1 | 2017-01-29T22:53:07Z [1] INFO In database 'topolog
y-777777-71804101010176': state standalone/init, tasks setupGraphs, createModules, c
reateRouting,
topology-processor_1 | at java.net.Inet4AddressImpl.lookupAllHostAddr(Native Metho
d)
mediation-vcenter_1 | </soapenv:Envelope>
```

```
api_1          | 2017-01-29 22:27:10,100 INFO [main] [CachingOperationNameG
erator] : Generating unique operation named: invokeUsingGET_12
market_1      | 2017-01-29 22:33:08,153 INFO [market-runner-0] [Ede] : Pla
n completed idleVM placement with 0 actions.
consul_1     | 2017/01/29 22:25:37 [INFO] agent: Synced check 'servi
ce:group-1'
```

- `vmtctl shell {containerName}`

Will log you into a specific container for troubleshooting.

- `vmtctl up {componentName}`

Will start a specific container, for troubleshooting purposes.



Appendix C: What Are the Typical Settings for an IdP?

Before you begin configuring Single Sign-On (SSO), you need to make sure the IdP is set up for SSO.

Here are typical settings for a public Okta IdP which may be useful when you set up your IdP.

SAML Settings: GENERAL	
Setting	Example
Single Sign On URL	<code>https://10.10.10.123/vmturbo/saml/SSO</code>
Recipient URL	<code>https://10.10.10.123/vmturbo/saml/SSO</code>
Destination URL	<code>https://10.10.10.123/vmturbo/saml/SSO</code>
Audience Restriction	<code>urn:test:turbo:markharm</code>
Default Relay State	
Name ID Format	Unspecified
Application username	The username for the account that is managed by Okta
Response	Signed
Assertion Signature	Signed

SAML Settings: GENERAL	
Setting	Example
Signature Algorithm	RSA_SHA256
Digital Algorithm	SHA256
Assertion Encryption	Unencrypted
SAML Single Logout	Enabled
Single Logout URL	https://10.10.10.123/vmturbo/rest/logout
SP Issuer	turbo
Signature Certificate	Example.cer (CN=apollo)
authnContextClassRef	PasswordProtectedTransport
Honor Force Authentication	Yes
SAML Issuer ID	http://www.okta.com/\$(org.externalKey)

SAML Settings: GROUP ATTRIBUTE STATEMENTS		
Name	Name Format	Filter
group	Unspecified	Matches regex:.*admin.*.