

TURBONOMIC AND OPENSTACK

Use Cases, compatibility & prerequisites

Turbonomic with OpenStack® on KVM or a hypervisor of your choice provides autonomic performance control across the OpenStack environment. This document describes the use cases, technical requirements and functionality of the integration, as well as the value-added instrumentation Turbonomic has contributed to the OpenStack project.

Scenario	Turbonomic Advantage	Method
App Performance Lifecycle Control	Maintain Desired State from initial VM deployment until retirement	Nova scheduler instrumentation
Tiered Service Offerings	Assure performance of selected workloads during periods of peak utilization while increasing overall resource efficiency	Flavor CPU shares / Provisioned Amount
Minimize Software Licensing Costs	Limit workload placement to specific Hosts to minimize socket-based licensing costs	Flavor Extra-Spec / Cluster
Workload Affinity Rules	Reduce network latency between customer-identified n-tier workloads	Server Group / Segment
Workload HA, Anti-Affinity Rules	Enable VM redundancy for High Availability and Load Balancing for high performance and scalability	Server Group / Segment
Shared Nothing Live Migration	Use inexpensive local storage for ephemeral-backed VMs, enabling Live Migration for performance, efficiency or Host maintenance	Turbonomic Group Shopping
Host Maintenance Mode	Evacuate all VMs from a Host to enable operational maintenance	Turbonomic Rest API
Forced Host Selection	Enable ongoing operational management during emergencies	Nova scheduler instrumentation

Software Versions

Supported Turbonomic Versions

Any Turbonomic instance can be used with OpenStack, however Version 5.0 (November 2014) or above is recommended. Instances are available on the Turbonomic download site in QCOW format for installation within OpenStack / KVM environments. Turbonomic Cloud Control functionality is required to validate OpenStack targets.

Supported OpenStack Versions and Packaging

OpenStack versions are named alphabetically. Recent versions include Icehouse (April 2014), Juno (October 2014), Kilo (April 2015), Liberty (October 2015) and Mitaka (April 2016). The OpenStack community operates around a six-month time based release cycle. Turbonomic was initially tested in a Red Hat Icehouse environment, and is fully compatible with any version Icehouse or above.

Supported Hypervisors

Turbonomic has been tested with Red Hat Enterprise Linux OpenStack Platform and with Red Hat RDO, both of which are Red Hat packaged versions of OpenStack, KVM and Linux. OpenStack supports many hypervisors for the compute nodes in the environment (e.g., KVM, ESXi™, Hyper-V®, XenServer®), most customers have chosen to use only KVM hypervisors with their OpenStack deployments. Turbonomic supports OpenStack with KVM from any provider, as well as with all OpenStack-ready hypervisors other than KVM.

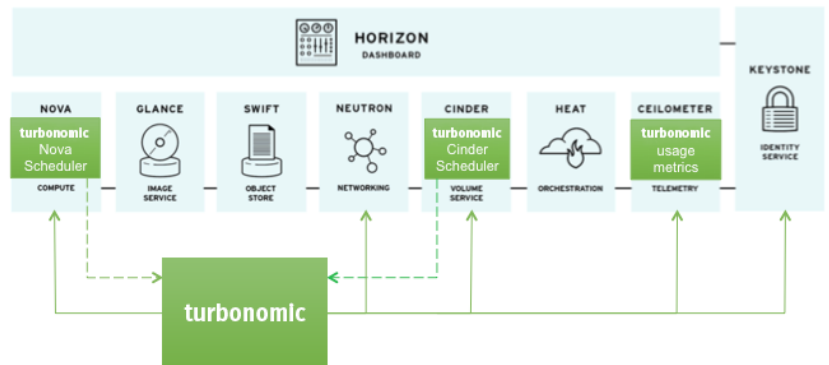
Operational Requirements

Required OpenStack Services

Turbonomic uses four OpenStack services:

1. Identity (*Keystone*) - discovery of OpenStack Tenants
2. Compute (*Nova*) - discovery of hypervisors
3. Block Storage (*Cinder*) - discovery of storage
4. Telemetry (*Ceilometer*) - monitoring of resource usage

All of these services must be installed and enabled, and Turbonomic must be able to access each service at its administrative endpoint (provided by Keystone during Target validation).



TURBONOMIC AND OPENSTACK

Use Cases, compatibility & prerequisites

OpenStack Target Validation

OpenStack targets added to Turbonomic are validated to meet these requirements:

1. All required OpenStack services (listed on previous page) are enabled.
2. The target URL is the public URL of the Keystone service. Specify the port unless it is 5000. A secure "https" connection is assumed for all ports other than 5000; specify the protocol to override this default. Turbonomic must be able to access this URL.
3. The User ID must be authenticated by OpenStack and must be authorized with an Admin role for the specified Tenant.
4. Turbonomic must be able to access each OpenStack service endpoint at its administrative endpoint.

Resource Monitoring Requirements

Host and VM CPU Utilization

Ceilometer automatically collects VM CPU utilization statistics; Turbonomic automatically retrieves them from Ceilometer. Ceilometer can also collect Host CPU utilization, however, the Nova service must be specifically configured to collect Host CPU utilization. This requires a simple change to the Nova.config file on each Host, adding the line: `compute_monitors = ComputeDriverCPUMonitor` and then restarting the Nova compute service: `sudo service openstack-nova-compute restart` after which, Turbonomic will automatically retrieve Host CPU utilization from Ceilometer.

With this minimum set of statistics (Host and VM CPU utilization), Turbonomic has enough information to analyze the OpenStack deployment health from a CPU utilization standpoint and recommend actions (e.g., VM resizing and live migration) to assure VM performance and maximize Host efficiency.

Host and VM Memory Utilization

OpenStack distributions do not currently provide Host Memory utilization metrics. Although OpenStack provides VM Memory utilization for some hypervisors, it does not provide it for KVM. Therefore, Turbonomic has developed code to provide VM Memory utilization and Host Memory utilization for OpenStack with KVM. Turbonomic contributed these metrics to the open source community for eventual distribution with OpenStack. Until then, they are available on Turbonomic's Github repository (www.github.com/Turbonomic). To enable these metrics, customers should download the code (or patch files created from the code) and install them on each Host in the OpenStack datacenter. Once installed on each Host, no additional configuration of Turbonomic is necessary to automatically obtain the metrics from Ceilometer.

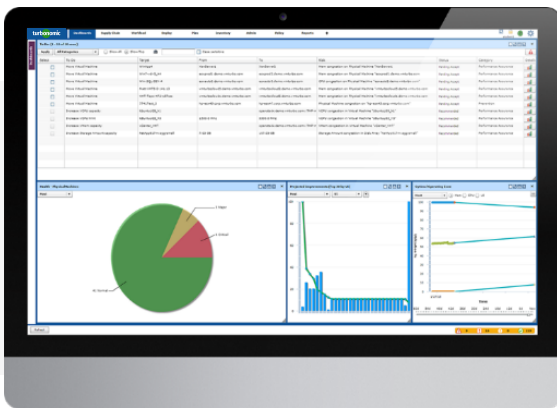
Additional Metrics

To deliver the vision of autonomic performance control to OpenStack deployments, Turbonomic has developed additional metrics beyond those natively distributed in the OpenStack base. These metrics include Host Net Throughput, as well as Host IO Throughput. They have been contributed to the OpenStack project and made available on the Turbonomic Github repository until they are incorporated into the base OpenStack distributions.

Automated Control Requirements

Turbonomic will decide to resize and live migrate VM as dictated by application workload resource demand. These actions can be taken manually, or automated entirely to provide command and control of the environment. This control requires no additional configuration beyond the availability of the metrics specified in OpenStack Target Validation above.

Additionally, Turbonomic provides a Turbonomic-specific Nova scheduler plugin that automates placement decisions for OpenStack VM deployments. This Nova scheduler plugin interfaces with Turbonomic to obtain workload deployment recommendations. The plugin is also available on the Turbonomic Github repository, is open source and can be downloaded and installed on the OpenStack Nova controller.



Try Turbonomic

- Download a free trial of Turbonomic for 30 days, at Turbonomic.com/download
- For more information, visit Turbonomic.com