Red Hat Hyperconverged Infrastructure for Virtualization 1.5

Automating RHHI for Virtualization deployment

Use Ansible to deploy your hyperconverged solution without manual intervention

Red Hat Hyperconverged Infrastructure for Virtualization 1.5 Automating RHHI for Virtualization deployment

Use Ansible to deploy your hyperconverged solution without manual intervention

Laura Bailey
lbailey@redhat.com

Legal Notice

Abstract

Read this for information about using Ansible to deploy Red Hat Hyperconverged Infrastructure for Virtualization without needing to watch and tend to the deployment process. This deployment method is provided as a Technology Preview. Technology Preview features are provided with a limited support scope, as detailed on the Customer Portal:
link:https://access.redhat.com/support/offerings/techpreview/[Technology Preview Features Support Scope].

Table of Contents

CHAPTER 1. ANSIBLE BASED DEPLOYMENT WORKFLOW

You can use Ansible to deploy Red Hat Hyperconverged Infrastructure for Virtualization without needing to watch and tend to the deployment process.

> **IMPORTANT**
>
> This deployment method is provided as a Technology Preview.
>
> Technology Preview features are provided with a limited support scope, as detailed on the Customer Portal: Technology Preview Features Support Scope.

The workflow for deploying RHHI for Virtualization using Ansible is as follows.

1. Verify that your planned deployment meets the requirements: Support requirements

2. Install the physical machines that will act as virtualization hosts: Installing host physical machines

3. Configure key-based SSH authentication without a password to allow automatic host configuration: Configuring public key based SSH authentication without a password

4. On Red Hat Enterprise Linux based hosts, install Ansible packages and roles: Installing Ansible packages and roles

5. Edit the variable file with details of your environment: Setting deployment variables

6. Execute the Ansible playbook to deploy RHHI for Virtualization: Executing the deployment playbook

7. Verify your deployment.

CHAPTER 2. SUPPORT REQUIREMENTS

Review this section to ensure that your planned deployment meets the requirements for support by Red Hat.

2.1. OPERATING SYSTEM

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) uses Red Hat Virtualization Host 4.2 as a base for all other configuration. The following table shows the the supported versions of each product to use for a supported RHHI for Virtualization deployment.

Table 2.1. Version compatibility

RHHI version	RHGS version	RHV version
1.0	3.2	4.1.0 to 4.1.7
1.1	3.3.1	4.1.8 to 4.2.0
1.5	3.4 Batch 1 Update	4.2.0 to current

See Requirements in the Red Hat Virtualization *Planning and Prerequisites Guide* for details on requirements of Red Hat Virtualization.

2.2. PHYSICAL MACHINES

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) requires **at least 3 physical machines**. Scaling to 6, 9, or 12 physical machines is also supported; see Scaling for more detailed requirements.

Each physical machine must have the following capabilities:

- at least 2 NICs (Network Interface Controllers) per physical machine, for separation of data and management traffic (see Section 2.4, "Networking" for details)

- for small deployments:

  - at least 12 cores

  - at least 64GB RAM

  - at most 48TB storage

- for medium deployments:

  - at least 12 cores

  - at least 128GB RAM

  - at most 64TB storage

- for large deployments:

  - at least 16 cores

- at least 256GB RAM

- at most 80TB storage

2.3. HOSTED ENGINE VIRTUAL MACHINE

The Hosted Engine virtual machine requires at least the following:

- 1 dual core CPU (1 quad core or multiple dual core CPUs recommended)

- 4GB RAM that is not shared with other processes (16GB recommended)

- 25GB of local, writable disk space (50GB recommended)

- 1 NIC with at least 1Gbps bandwidth

For more information, see Requirements in the Red Hat Virtualization 4.2 *Planning and Prerequisites Guide*.

2.4. NETWORKING

Each node requires 3 x 1 Gigabit Ethernet ports. To enable high availability, these must be split across two network switches. Ensuring that switches have separate power supplies further improves fault tolerance.

Fully-qualified domain names that are forward and reverse resolvable by DNS are required for all hosts and for the Hosted Engine virtual machine.

Client and management traffic in the cluster must use separate networks. This ensures optimal performance. Red Hat recommends two separate networks:

A front-end management network

This network is used by Red Hat Virtualization and virtual machines.

- This network should be capable of transmitting at Gigabit Ethernet speeds.

- IP addresses assigned to this network can be selected by the administrator, but must be on the same subnet as each other.

- IP addresses assigned to this network must not be in the same subnet as the back-end storage and migration network.

A back-end storage network

This network is used for storage and migration traffic between storage peers.

- Red Hat recommends a 10Gbps network for the back-end storage network.

- Red Hat Gluster Storage requires a maximum latency of 5 milliseconds between peers.

Network fencing devices that use Intelligent Platform Management Interfaces (IPMI) require a separate network.

If you want to use DHCP network configuration for the Hosted Engine virtual machine, then you must have a DHCP server configured prior to configuring Red Hat Hyperconverged Infrastructure for Virtualization.

If you want to use geo-replication to store copies of data for disaster recovery purposes, a reliable time source is required.

Determine or decide on the following details before you begin the deployment process:

- IP address for a gateway to the virtualization host that responds to pings

- IP address of the front-end management network

- Fully-qualified domain name (FQDN) for the Hosted Engine virtual machine

- MAC address that resolves to the static FQDN and IP address of the Hosted Engine

2.5. STORAGE

A hyperconverged host stores configuration, logs and kernel dumps, and uses its storage as swap space. This section lists the minimum directory sizes for hyperconverged hosts. Red Hat recommends using the default allocations, which use more storage space than these minimums.

- **/** (root) - 6GB

- **/home** - 1GB

- **/tmp** - 1GB

- **/boot** - 1GB

- **/var** - 22GB

- **/var/log** - 15GB

- **/var/log/audit** - 2GB

- **swap** - 1GB (for the recommended swap size, see https://access.redhat.com/solutions/15244)

- Anaconda reserves 20% of the thin pool size within the volume group for future metadata expansion. This is to prevent an out-of-the-box configuration from running out of space under normal usage conditions. Overprovisioning of thin pools during installation is also not supported.

- **Minimum Total - 52GB**

2.5.1. Disks

Red Hat recommends Solid State Disks (SSDs) for best performance. If you use Hard Drive Disks (HDDs), you should also configure a smaller, faster SSD as an LVM cache volume.

4K native devices are not supported with Red Hat Hyperconverged Infrastructure for Virtualization, as Red Hat Virtualization requires 512 byte emulation (512e) support.

2.5.2. RAID

RAID5 and RAID6 configurations are supported. However, RAID configuration limits depend on the technology in use.

- SAS/SATA 7k disks are supported with RAID6 (at most 10+2)

- SAS 10k and 15k disks are supported with the following:

  - RAID5 (at most 7+1)

  - RAID6 (at most 10+2)

RAID cards must use flash backed write cache.

Red Hat further recommends providing at least one hot spare drive local to each server.

2.5.3. JBOD

As of Red Hat Hyperconverged Infrastructure for Virtualization 1.5, JBOD configurations are fully supported and no longer require architecture review.

2.5.4. Logical volumes

The logical volumes that comprise the **engine** gluster volume must be thick provisioned. This protects the Hosted Engine from out of space conditions, disruptive volume configuration changes, I/O overhead, and migration activity.

When VDO is not in use, the logical volumes that comprise the **vmstore** and optional **data** gluster volumes must be thin provisioned. This allows greater flexibility in underlying volume configuration. If your thin provisioned volumes are on Hard Drive Disks (HDDs), configure a smaller, faster Solid State Disk (SSD) as an lvmcache for improved performance.

Thin provisioning is not required for the **vmstore** and **data** volumes if VDO is being used on these volumes.

2.5.5. Red Hat Gluster Storage volumes

Red Hat Hyperconverged Infrastructure for Virtualization is expected to have 3–4 Red Hat Gluster Storage volumes.

- 1 **engine** volume for the Hosted Engine

- 1 **vmstore** volume for virtual machine boot disk images

- 1 optional **data** volume for other virtual machine disk images

- 1 **shared_storage** volume for geo-replication metadata

A Red Hat Hyperconverged Infrastructure for Virtualization deployment can contain at most 1 geo-replicated volume.

2.5.6. Volume types

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) supports only the following volume types:

- Replicated volumes (3 copies of the same data on 3 bricks, across 3 nodes).

- Arbitrated replicated volumes (2 full copies of the same data on 2 bricks and 1 arbiter brick that contains metadata. across three nodes).

- Distributed volumes (1 copy of the data, no replication to other bricks).

All replicated and arbitrated-replicated volumes must span exactly three nodes.

Note that arbiter bricks store only file names, structure, and metadata. This means that a three-way arbitrated replicated volume requires about 75% of the storage space that a three-way replicated volume would require to achieve the same level of consistency. However, because the arbiter brick stores only metadata, a three-way arbitrated replicated volume only provides the availability of a two-way replicated volume.

For more information on laying out arbitrated replicated volumes, see Creating multiple arbitrated replicated volumes across fewer total nodes in the Red Hat Gluster Storage *Administration Guide*.

2.6. VIRTUAL DATA OPTIMIZER (VDO)

A Virtual Data Optimizer (VDO) layer is supported as of Red Hat Hyperconverged Infrastructure for Virtualization 1.5.

The following limitations apply to this support:

- VDO is supported only on new deployments.

- VDO is compatible only with thick provisioned volumes. VDO and thin provisioning are not supported on the same device.

2.7. SCALING

Red Hat Hyperconverged Infrastructure for Virtualization is supported for one node, and for clusters of 3, 6, 9, and 12 nodes.

The initial deployment is either 1 or 3 nodes.

There are two supported methods of horizontally scaling Red Hat Hyperconverged Infrastructure for Virtualization:

1. Add new hyperconverged nodes to the cluster, in sets of three, up to the maximum of 12 hyperconverged nodes.

2. Create new Gluster volumes using new disks on existing hyperconverged nodes.

You cannot create a volume that spans more than 3 nodes, or expand an existing volume so that it spans across more than 3 nodes at a time.

2.8. EXISTING RED HAT GLUSTER STORAGE CONFIGURATIONS

Red Hat Hyperconverged Infrastructure for Virtualization is supported only when deployed as specified in this document. Existing Red Hat Gluster Storage configurations cannot be used in a hyperconverged configuration. If you want to use an existing Red Hat Gluster Storage configuration, refer to the traditional configuration documented in Configuring Red Hat Virtualization with Red Hat Gluster Storage.

2.9. DISASTER RECOVERY

Red Hat strongly recommends configuring a disaster recovery solution. For details on configuring geo-replication as a disaster recovery solution, see *Maintaining Red Hat Hyperconverged Infrastructure for Virtualization*: https://access.redhat.com/documentation/en-us/red_hat_hyperconverged_infrastructure_for_virtualization/1.5/html/maintaining_red_hat_hyperconverged_backup-recovery.

2.9.1. Prerequisites for geo-replication

Be aware of the following requirements and limitations when configuring geo-replication:

One geo-replicated volume only

Red Hat Hyperconverged Infrastructure for Virtualization (RHHI for Virtualization) supports only one geo-replicated volume. Red Hat recommends backing up the volume that stores the data of your virtual machines, as this is usually contains the most valuable data.

Two different managers required

The source and destination volumes for geo-replication must be managed by different instances of Red Hat Virtualization Manager.

2.9.2. Prerequisites for failover and failback configuration

Versions must match between environments

Ensure that the primary and secondary environments have the same version of Red Hat Virtualization Manager, with identical data center compatibility versions, cluster compatibility versions, and PostgreSQL versions.

No virtual machine disks in the hosted engine storage domain

The storage domain used by the hosted engine virtual machine is not failed over, so any virtual machine disks in this storage domain will be lost.

Execute Ansible playbooks manually from a separate master node

Generate and execute Ansible playbooks manually from a separate machine that acts as an Ansible master node.

2.10. ADDITIONAL REQUIREMENTS FOR SINGLE NODE DEPLOYMENTS

Red Hat Hyperconverged Infrastructure for Virtualization is supported for deployment on a single node provided that all Support Requirements are met, with the following additions and exceptions.

A single node deployment requires a physical machine with:

- 1 Network Interface Controller

- at least 12 cores

- at least 64GB RAM

- at most 48TB storage

Single node deployments cannot be scaled, and are not highly available.

CHAPTER 3. INSTALL HOST PHYSICAL MACHINES

Install Red Hat Virtualization Host 4.2 on your three physical machines.

See the following section for details about installing a virtualization host:
https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.2/html/installation_guide/red_hat_virtualization_hosts.

Ensure that you customize your installation to provide the following when you install each host:

- Increase the size of **/var/log** to 15GB to provide sufficient space for the additional logging requirements of Red Hat Gluster Storage.

CHAPTER 4. CONFIGURE PUBLIC KEY BASED SSH AUTHENTICATION WITHOUT A PASSWORD

From the first virtualization host, configure Public Key based SSH authentication for the root user without a password to all virtualization hosts using the FQDN associated with the management network. This includes authentication from the first host to itself.
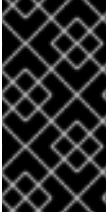
IMPORTANT

RHHI for Virtualization expects key-based SSH authentication without a password between these nodes for both IP addresses and FQDNs. Ensure that you configure key-based SSH authentication without a password between these machines for the IP address and FQDN of all storage and management network interfaces.

See the Red Hat Enterprise Linux 7 **Installation Guide** for more details:
https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/s1-ssh-configuration.html#s2-ssh-configuration-keypairs.

CHAPTER 5. INSTALLING ANSIBLE PACKAGES AND ROLES

If you are installing from Red Hat Enterprise Linux 7 instead of from Red Hat Virtualization 4.2, you need to install some additional packages.

1. Register your machine to Red Hat Network.

   ```
   # subscription-manager register --username=<username> --password=
   <password>
   ```

2. Enable the channels required for Ansible.

   ```
   # subscription-manager repos --enable=rhel-7-server-rpms  --
   enable=rh-gluster-3-for-rhel-7-server-rpms  --enable=rhel-7-server-
   rhv-4-mgmt-agent-rpms --enable=rhel-7-server-ansible-2-rpms --
   enable=rhel-7-server-rhvh-4-rpms
   ```

3. Install ansible and the required roles.

   ```
   # yum install ansible gluster-ansible-roles ovirt-ansible-hosted-
   engine-setup ovirt-ansible-repositories ovirt-ansible-engine-setup
   ```

   Installing the gluster-ansible-roles package places a number of files in the
   /usr/share/doc/gluster.ansible/playbooks directory.

CHAPTER 6. SETTING DEPLOYMENT VARIABLES

1. Make a backup copy of the playbooks directory.

   ```
   # cp -r playbooks playbook-templates
   ```

2. Edit the inventory file.
   Make the following updates to the **playbooks/gluster_inventory.yml** file.

   a. Add host FQDNs to the inventory file

   - On line 4, replace **host1** with the FQDN of the first host.

     Lines 3-4 of gluster_inventory.yml

     ```
     # Host1
     servera.example.com:
     ```

   - On line 71, replace **host2** with the FQDN of the second host.

     Lines 70-71 of gluster_inventory.yml

     ```
     #Host2
     serverb.example.com:
     ```

   - On line 138, replace **host3** with the FQDN of the third host.

     Lines 137-138 of gluster_inventory.yml

     ```
     #Host3
     serverc.example.com:
     ```

   - On line 237 and 238, replace **host2** and **host3** with the FQDN of the second and third host respectively.

     Lines 235-238 of gluster_inventory.yml

     ```
     gluster:
      hosts:
        serverb.example.com:
        serverc.example.com:
     ```

   b. If you want to use VDO for deduplication and compression

   i. Uncomment the **Dedupe & Compression config** and **With Dedupe & Compression** sections by removing the # symbol from the beginning of the following lines.

      ```
      #gluster_infra_vdo:
         #- { name: 'vdo_sdb1', device: '/dev/sdb1', logicalsize:
      '3000G', emulate512: 'on', slabsize: '32G',
            #blockmapcachesize:  '128M', readcachesize: '20M',
      readcache: 'enabled', writepolicy: 'auto' }
      ```

```
     #- { name: 'vdo_sdb2', device: '/dev/sdb2', logicalsize:
'3000G', emulate512: 'on', slabsize: '32G',
       #blockmapcachesize:  '128M', readcachesize: '20M',
readcache: 'enabled', writepolicy: 'auto' }


#gluster_infra_volume_groups:
  #- vgname: vg_sdb1
    #pvname: /dev/mapper/vdo_sdb1
  #- vgname: vg_sdb2
    #pvname: /dev/mapper/vdo_sdb2
```

    ii. Comment out the **Without Dedupe & Compression** section by adding a # to the beginning of each line.

```
# Without Dedupe & Compression
gluster_infra_volume_groups:
  - vgname: vg_sdb1
    pvname: /dev/sdb1
  - vgname: vg_sdb2
    pvname: /dev/sdb2
```

3. Edit the hosted engine variables file.
   Update the following values in the **playbooks/he_gluster_vars.json** file.

he_appliance_password

    The root password of the host machine.

he_admin_password

    The password for the root account of the Administration Portal.

he_domain_type

    glusterfs - There is no need to change this value.

he_fqdn

    The fully qualified domain name for the Hosted Engine virtual machine.

he_vm_mac_addr

    A valid MAC address for the Hosted Engine virtual machine.

he_default_gateway

    The IP address of the default gateway server.

he_mgmt_network

    The name of the management network. The default value is **ovirtmgmt**.

he_host_name

    The short name of this host.

he_storage_domain_name

    HostedEngine

he_storage_domain_path

    /engine

he_storage_domain_addr

    The IP address of this host on the storage network.

he_mount_options

backup-volfile-servers=\<host2-ip-address\>:\<host3-ip-address\>, with the appropriate IP addresses inserted in place of **\<host2-ip-address\>** and **\<host3-ip-address\>**.

he_bridge_if

The name of the interface to be used as the bridge.

he_enable_hc_gluster_service

true

CHAPTER 7. EXECUTING THE DEPLOYMENT PLAYBOOK

1. Change into the playbooks directory on the first node.

   ```
   # cd /usr/share/doc/gluster.ansible/playbooks
   ```

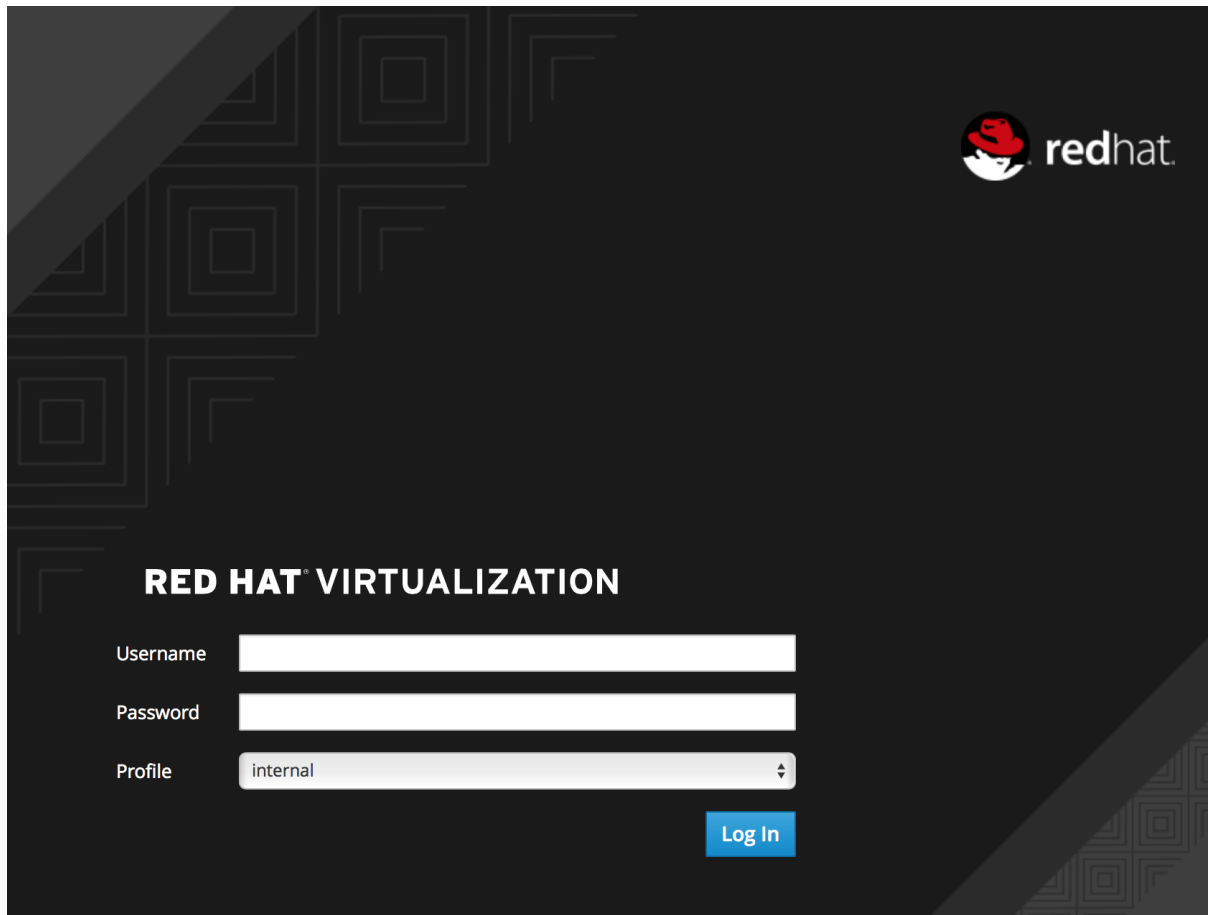2. Run the following command as the root user to start the deployment process.

   ```
   # ansible-playbook -i gluster_inventory.yml hc_deployment.yml --
   extra-vars='@he_gluster_vars.json'
   ```

CHAPTER 8. VERIFY YOUR DEPLOYMENT

After deployment is complete, verify that your deployment has completed successfully.

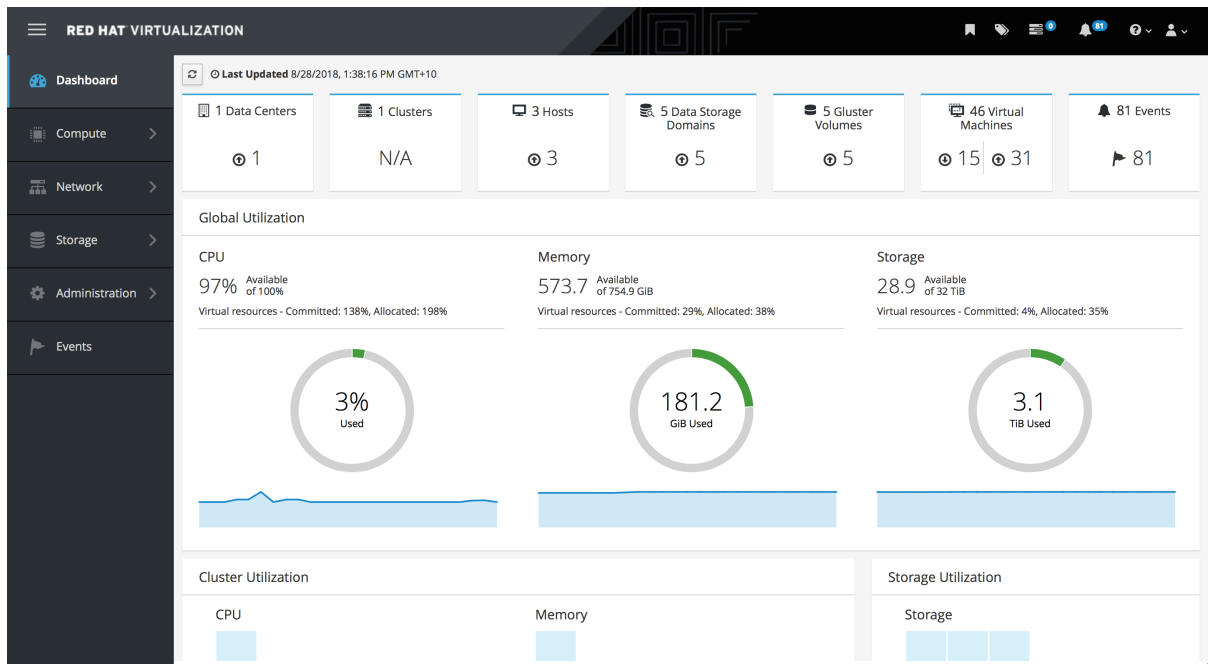1. Browse to the engine user interface, for example, http://engine.example.com/ovirt-engine.

   Administration Console Login

2. Log in using the administrative credentials added during hosted engine deployment. When login is successful, the Dashboard appears.

   Administration Console Dashboard

3. Verify that your cluster is available.

Administration Console Dashboard - Clusters

4. Verify that at least one host is available.
If you provided additional host details during Hosted Engine deployment, 3 hosts are visible here, as shown.

Administration Console Dashboard - Hosts

a. Click **Compute** → **Hosts**.

b. Verify that all hosts are listed with a **Status** of **Up**.

Administration Console - Hosts

Compute » Hosts

		Name	Comment	Hostname/IP	Cluster	Data Center	Status	Vi
▲ !		rhsdev-grafton2.lab.eng.b		rhsdev-grafton2.lab.en...	Default	Default	Up	
▲ !		rhsdev-grafton3.lab.eng.b		rhsdev-grafton3.lab.en...	Default	Default	Up	
▲ !		rhsdev-grafton4.lab.eng.b		rhsdev-grafton4.lab.en...	Default	Default	Up	

5. Verify that all storage domains are available.

   a. Click **Storage** → **Domains**.

   b. Verify that the `Active` icon is shown in the first column.

   Administration Console - Storage Domains

   Storage » Storage Domains

		Domain Name	Comment	Domain Type	Storage Type	Format	Cross Data Center Status	Total Space	Free Space
▲		data		Data	GlusterFS	V4	Active	4998 GiB	4563 GiB
▲		hosted_storage		Data (Master)	GlusterFS	V4	Active	99 GiB	88 GiB
▲		vmstore		Data	GlusterFS	V4	Active	9998 GiB	9284 GiB