Hewlett Packard Enterprise

Deployment Guide

Rancher 2.0 Container Platform on Red Hat Enterprise Linux 7.6 for both Virtual machine and Bare-metal

Implementing a Highly Available Rancher 2.0 on-premises container solution

July 2019, Version 3.0

Contents

Overview 3
Introduction 3
Why Rancher? 3
Solution Layout 4
Requirements for deploying rancher 5
Deployment environment 5
Edit the rhost file for you ansible playbook deployment 7
For Virtual Machine Deployment 8
Creating a RHEL 7.6 VM Template 8
Update OS and install the necessary tools and package 19
Disabling IPv6 support in Red Hat Enterprise Linux 7 20
Convert the VM as a template 20
Ansible scripts for Virtual machines 21
Common steps for either Virtual Machine or Bare metal deployment 22
Disabling IPv6 support in Red Hat Enterprise Linux 7 22
Prepare the host for Rancher and load balancer deployment 22
Install Docker on servers for Rancher 23
Creating the Kubernetes cluster 23
Create service account for tiller and test helm 24
Deploy and configure HAProxy for SSL Pass-through 24
Install Rancher 25
Post installation task forRancher 25
Appendix A- HPE 3PAR StoreServ integration Rancher 26
Resources and additional links (Required) 26

July 2019, Version 3.0

Overview

This document describes the steps required to create an highly available Rancher 2.0 Container Platform environment running Red Hat Enterprise Linux 7.6 . It is meant to be used in conjunction with files and Ansible playbooks found https://github.hpe.com/eaj/Anisble_Rancher.git It is recommended that the installer review this document in its entirety and understand all prerequisites prior to beginning an installation.

Introduction

Containers have dramatically risen in popularity because they provide a consistent way to package application components and their dependencies into a single object that can run in any environment. By packaging code and its dependencies into containers, a development team can use standardized units of code as consistent building blocks. The container will run the same way in any environment and can start and terminate quickly, allowing applications to scale to any size.

Kubernetes can be deployed in almost any environment and is able to cope with the presence of multiple types of hardware and software in the enterprise, differing network technologies, or even competing desires between DevOps and IT teams. However, it only works at peak efficiency when properly managed through an orchestration platform like Rancher.

Rancher is a complete software stack for teams adopting containers. It addresses the operational and security challenges of managing multiple Kubernetes clusters across any infrastructure, while providing DevOps teams with integrated tools for running containerized workloads. Rancher Kubernetes Engine (RKE) is a CNCF-certified Kubernetes distribution that runs entirely within Docker containers. The only software a host needs to run is Docker, which opens the door for running Kubernetes in small-footprint, secure locations. It builds a cluster from a single command in only a few minutes, and its declarative configuration makes Kubernetes upgrades atomic and safe.

Why Rancher?

Kops and Kubespray exist to deploy Kubernetes clusters but can only work with a handful of providers. Both focus on deploying infrastructure and then installing Kubernetes, and then they leave you to figure out what to do next. If something fails during the deploy, their complexity makes it difficult to debug what went wrong. Rancher deploys instances and installs Kubernetes, and it does so with more than a dozen providers. It also deploys hosted Kubernetes clusters with providers like GKE , AKS, and EKS. Rancher is more than an installer. After the clusters are up and running, Rancher manages them with role-based access control (RBAC), deploys workloads onto them, monitors them, notifies you of failures, connects the clusters to your CI/CD system, and gives you a complete solution for using Kubernetes.

July 2019, Version 3.0

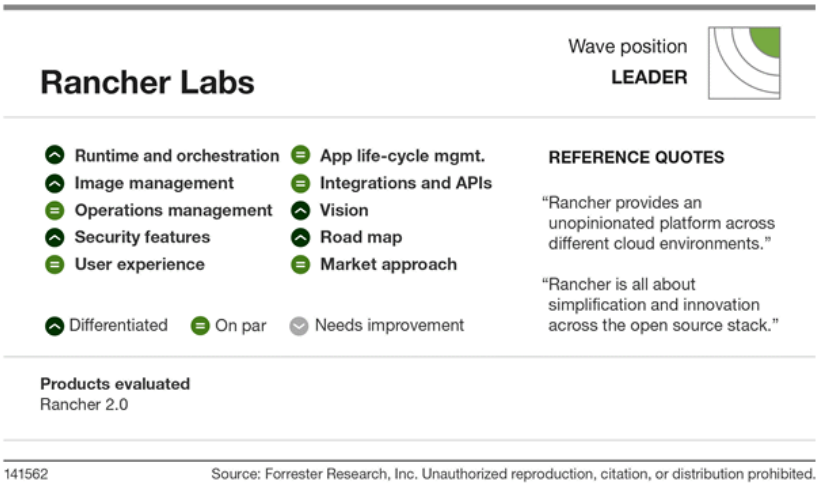


Figure 1. Rancher Summary - Forrester

Solution Layout

This solution comprises of four compute nodes running RHEL 7.6 can be physical or virtual machines running on VMware shown as follows:

One Load Balancer (HAproxy)

Three Rancher Nodes (Kubernetes + Docker)

Another compute is required to runs ansible engine(RHEL 7.6) for solution deployment and CLI management of the Rancher cluster. The ansible playbook reference in this document can modified to deploy additional rancher nodes if required

July 2019, Version 3.0

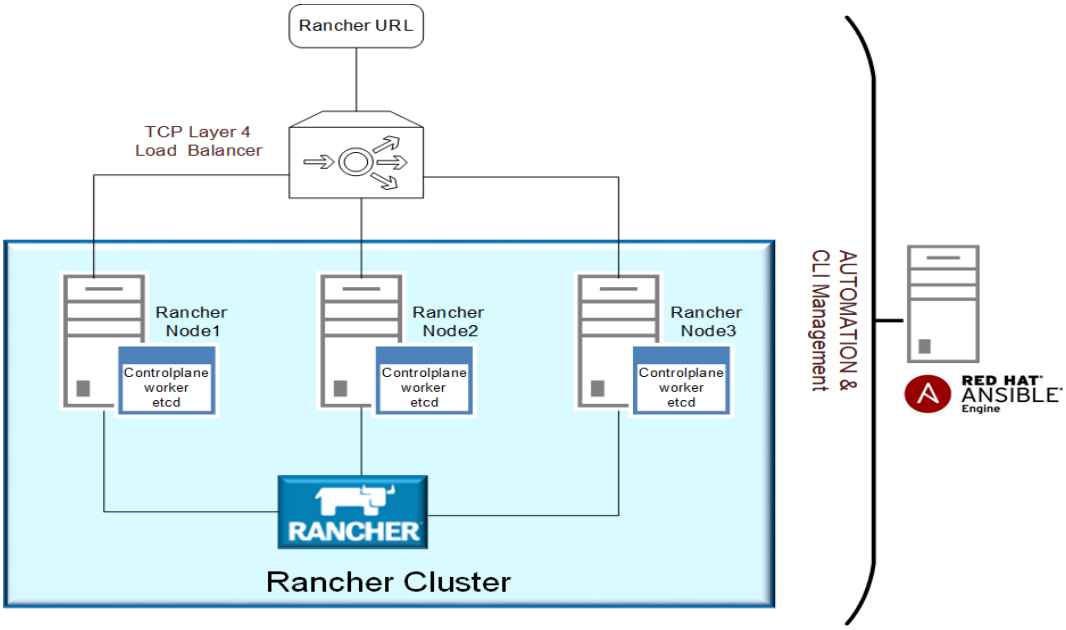


Figure 2. Solution Layout

Note

Scripts and files are provided as is and are examples of how to build out your infrastructure. It is expected that they will need to be adapted to work in the deployment environment. Please change the working directory to Anisble_Rancher to execute the plays discussed in this document. This document assumes you have root access to all the 4 required rancher servers and the Ansible engine host. This document wil install rancher using self-signed certificate

Requirements for deploying rancher

For deploying Rancher 2.0 refer to the rancher documentation https://rancher.com/docs/rke/latest/en/os/

Deployment environment

This document makes assumptions about services and networks available within the implementation environment. This section discusses those assumptions and, where applicable, provides details on how they should be configured. If a service is optional it will be noted in the description.

July 2019, Version 3.0

Services

Table 1 disseminates the services utilized in the creation of this solution and provides a high-level explanation of their function and whether or not they are required.

Table 1. Services used in the creation of this solution.

Service	Required/Optional	Description/Notes
DNS	Required	Provides name resolution on management and data center networks, optionally on iSCSI networks.
DHCP	Required	Provides IP address leases on PXE, management and usually for data center networks. Optionally used to provide addresses on iSCSI networks.
NTP	Required	Required to insure consistent time across the solution stack
Active Directory/LDAP	Optional	May be used for authentication functions on various networks. This solution utilized local authentication.

DNS

Name services must be in place for management and data center networks. Once a host has become active ensure that both forward and reverse lookups are working on the management and data center networks.

DHCP

DHCP services must be in place for the PXE and management networks. DHCP services are generally in place on data center networks. As a convenience it may be useful to have them in place on iSCSI networks. Because Virtual Connect exposes the MAC address of the network interfaces before installation has begun it is easy to create address reservations for the hosts. A reservation is required for a single adapter on the management network of each physical server. This facilitates post-deployment configuration over SSH as well as a secure communication channel for running Ansible scripts. If DHCP services are present on the iSCSI networks reservations can simplify post-deployment configuration of the host on those networks.

NTP

A Network Time Protocol server should be available to hosts within the solution environment.

Ansible Engine

This document assumes that Ansible Engine exists within the deployment environment and is accessible to the installer building this solution using Ansible version 2.7.2

The following repositories need to be enabled on the Ansible Engine host.

rhel-7-server-extras-rpms

rhel-7-server-rpms

rhel-7-server-ose-3.11-rpms

July 2019, Version 3.0

rhel-7-server-ansible-2.7-rpms

rhel-7-rhv-4-mgmt-agent-rpms

Additional tools are required to be installed on the ansible engine host for Rancher deployment

kubectl - Kubernetes command-line tool.

rke - Rancher Kubernetes Engine, cli for building Kubernetes clusters.

helm - Package management for Kubernetes.

These tool can be download and added to the path on the ansible engine host manually or you could use the Toolslocal.yml. playbook from the Asnible_Rancher repository @ https://github.hpe.com/eaj/Anisble_Rancher.git

ansible-playbook Toolslocal.yml

Edit the rhost file for you ansible playbook deployment

The host file for the ansible playbook execution is sown below. This file must be edited to match your environment setup.

```
[all]
rancher01.tennet.local
rancher02.tennet.local
rancher03.tennet.local
lb01.tennet.local

[ansible]
localhost

[server]
rancher01.tennet.local vm_name=rancher01 ipv4addr=10.0.8.91 ansible_connection=ssh ansible_ssh_user=root
ansible_ssh_pass="Password!234"
rancher02.tennet.local vm_name=rancher02 ipv4addr=10.0.8.92 ansible_connection=ssh ansible_ssh_user=root
ansible_ssh_pass="Password!234"
rancher03.tennet.local vm_name=rancher02 ipv4addr=10.0.8.92 ansible_connection=ssh ansible_ssh_user=root
ansible_ssh_pass="Password!234"

[lb]
lb01.tennet.local vm_name=lb01 ipv4addr=10.0.8.94 ansible_connection=ssh ansible_ssh_user=root
ansible_ssh_pass="Password!234"
```

For Virtual Machine Deployment

These steps are tested against VMware Vsphere version 6.7. this document assume that VMware 6.7 is up and running with vcenter configured with required Datastore's and networks for VM deployment and public network access

The following components need to be installed on the Ansible Engine host.

Python 2.7.9 and above

For virtualized deployment Python SDK for the VMware vSphere

PyVmomi is the Python SDK for the VMware vSphere API that allows you to manage ESX, ESXi, and vCenter.

- Execute the command "pip install PyVmomi" in the command line to install PyVmomi

Creating a RHEL 7.6 VM Template

In this solution, we are making use of a VM template with the RHEL 7.6 operatng system with the VMware tools installed in it. The below section illustrates the steps that needs to be completed in order to create a RHEL 7.6 VM template.

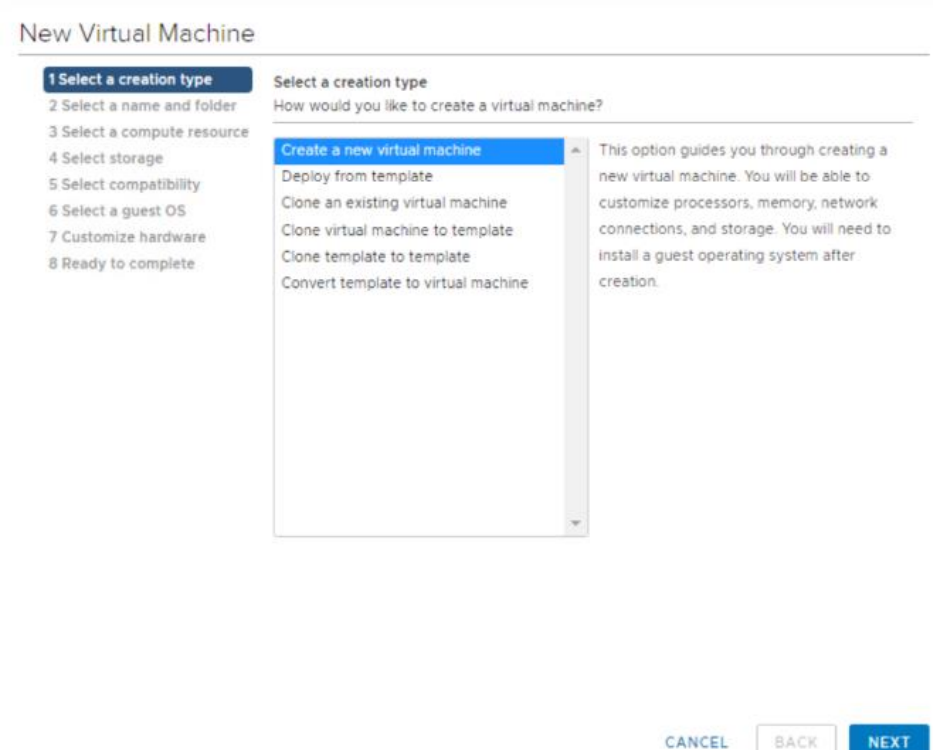
Overview of the tasks

1. Create a VM with RHEL 7.6 OS
2. Install the necessary tools and packages
3. Convert the VM as a template

Create a VM with RHEL 7.6 Operating System

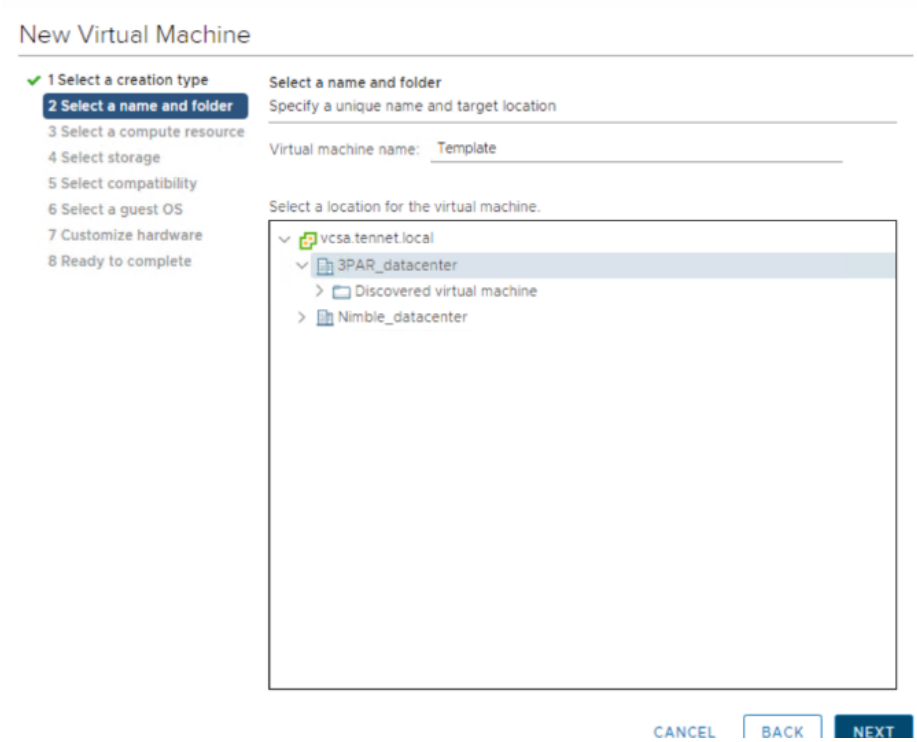
4. Login to vCenter and Right Click on the Cluster name to find the New Virtual Machine to open a new Virtual Machine Wizard
5. In the creation type, Choose "Create a new virtual machine" and Click on Next.

July 2019, Version 3.0



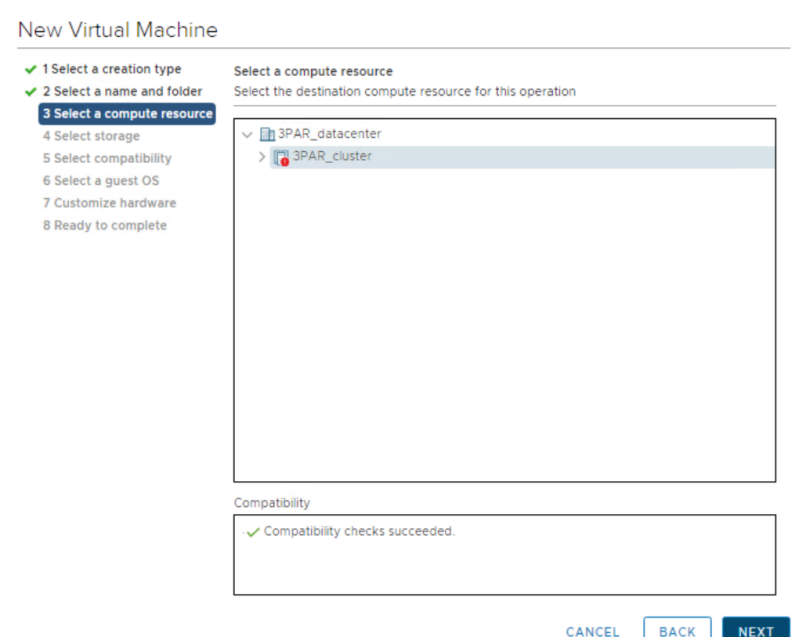
6. Enter a unique name for the Virtual Machine and Select Datacenter to deploy the VM. Click on Next.

July 2019, Version 3.0

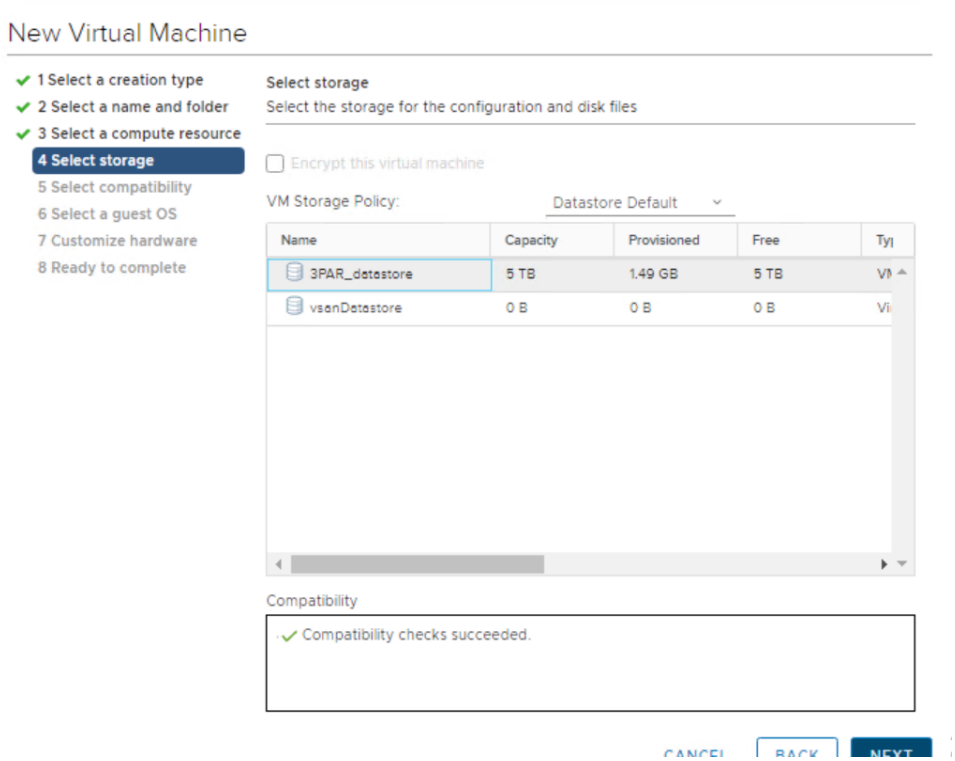


7. Select Cluster where the VM will be deployed.

July 2019, Version 3.0

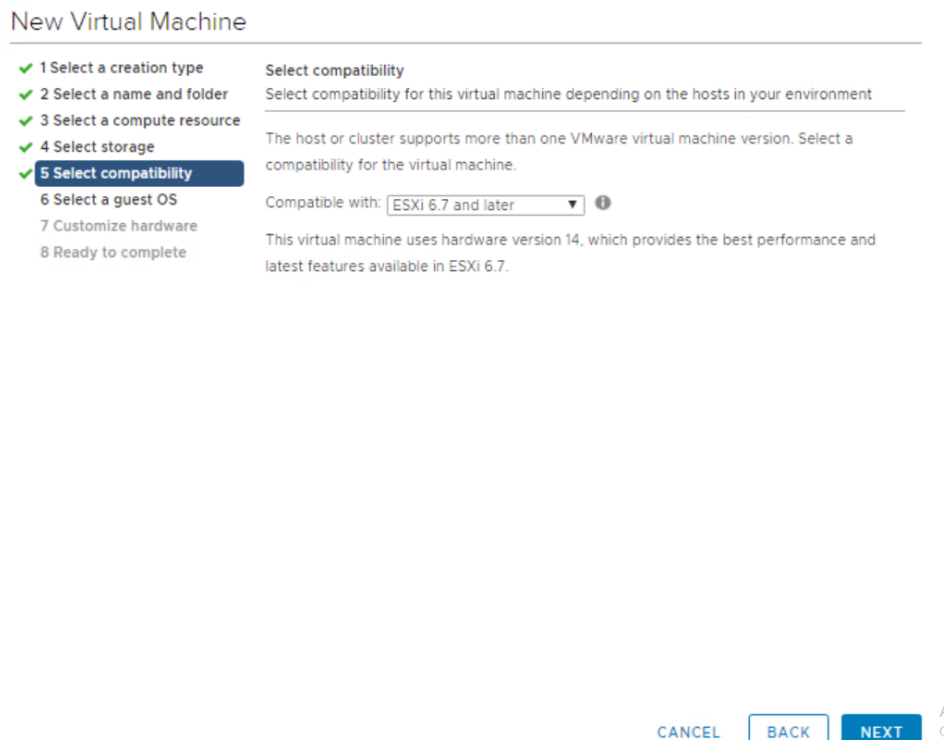


8. Select Datastore in which you would like to store the Virtual Machine configuration files and all of the virtual disks. Click on Next.

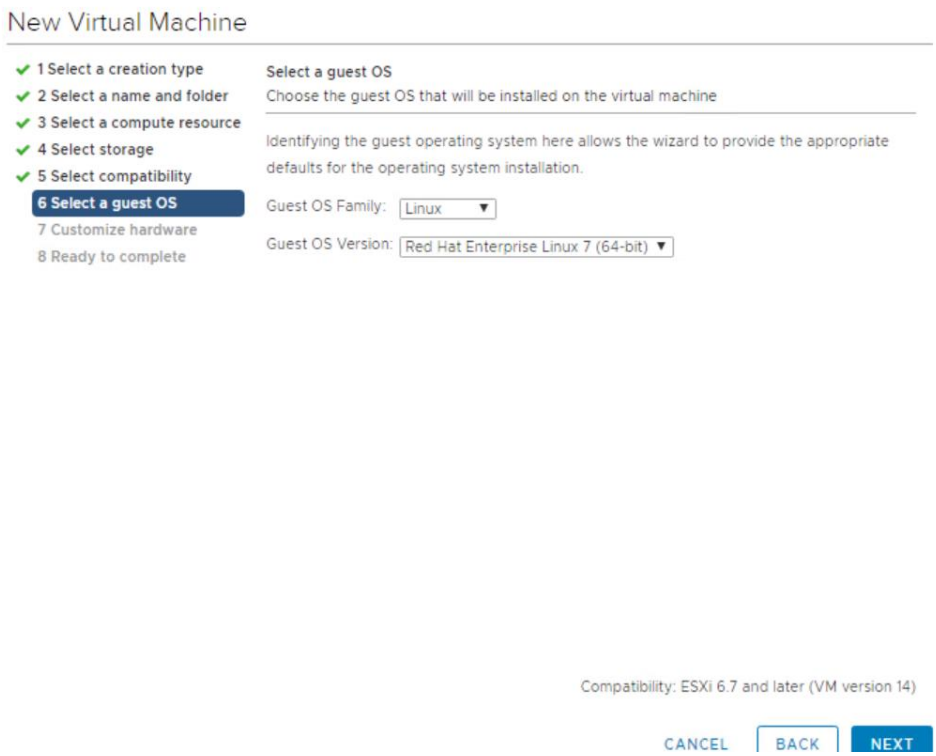


9. Select Compatibility for this Virtual Machine as ESXi 6.7 and later and Click on Next.

July 2019, Version 3.0

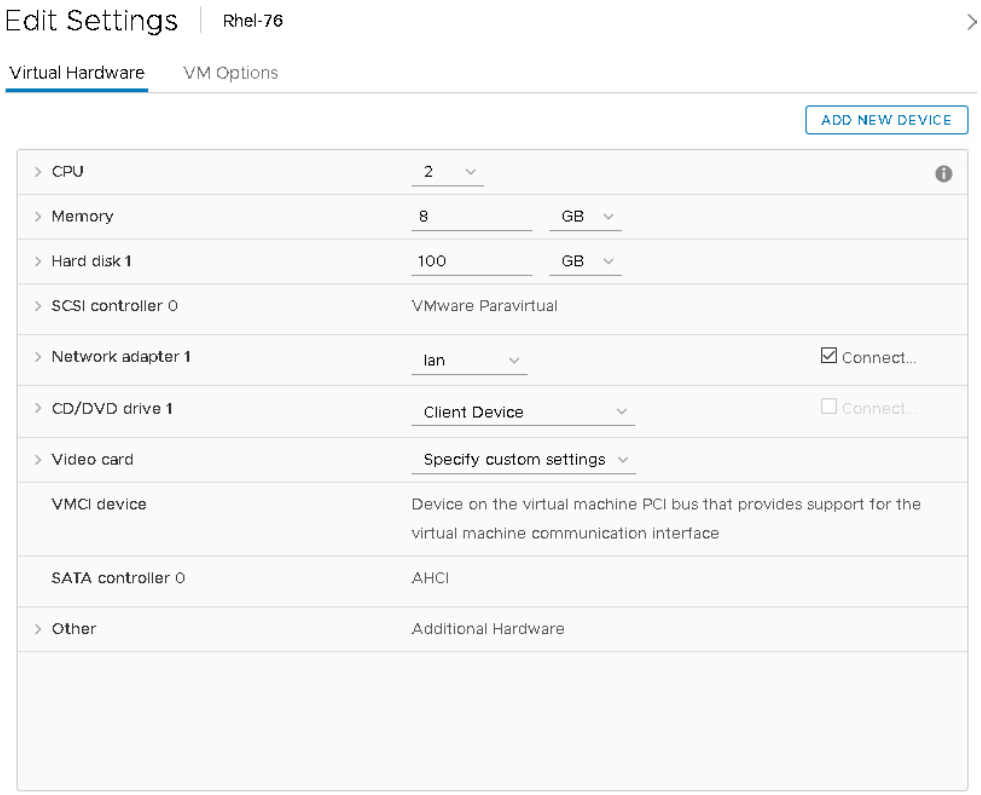


10. On Select a Guest OS page, Choose the Guest OS family as Linux and Guest OS Version as Red Hat Enterprise Linux 7 (64-bit)



11. Configure the Virtual Machine Hardware with 8 GB Memory, dual hard disk of 100 GB each and 2 CPU. Attach the Operating System from the datastore and check the "connect at power on" button and Click on Next.

July 2019, Version 3.0



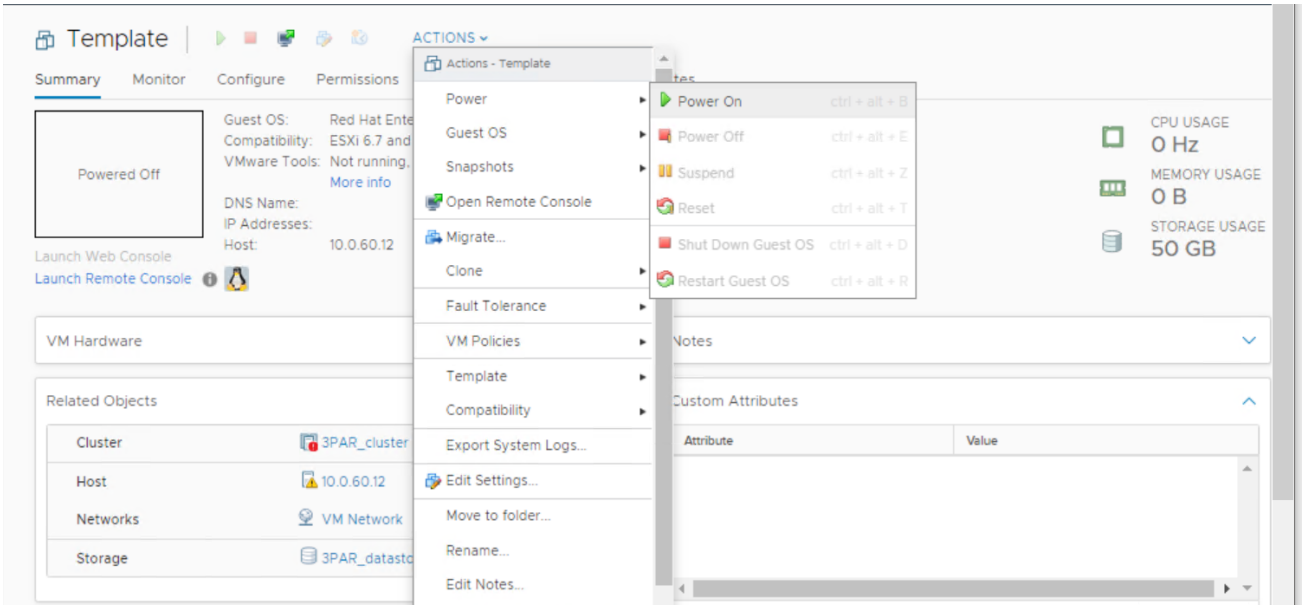
12. Review the virtual machine before you deploy the virtual machine. Click on Finish to Complete the VM creation.

The Creating Virtual Machine Wizard only provisions a New Virtual Machine in Environment. We need to install Operating System to complete the creation procedure. To Install a guest OS from a CD/DVD or using an image, follow the below Process:

13. Right Click on Virtual Machine and go to Edit Settings.

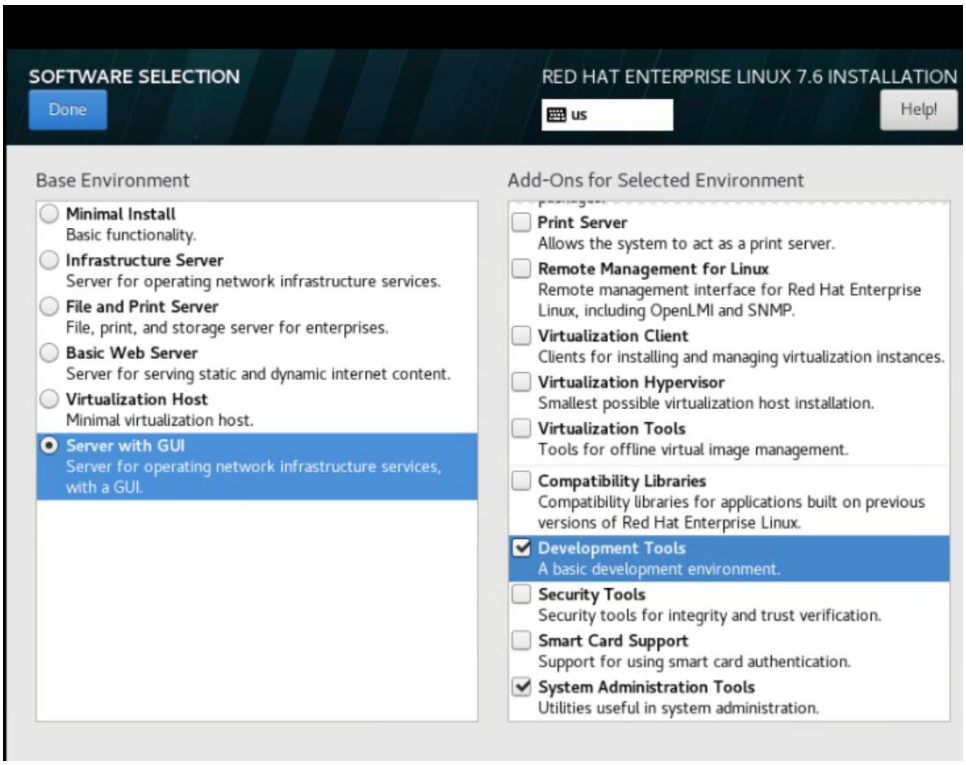
14. Power On Virtual Machine

July 2019, Version 3.0



15. Start your virtual machine by clicking Start.
16. The installation will launch after the virtual machine is started, when asked, press enter to begin the installation process.
17. Select the language and click next.
18. On the Installation Summary page, click Software Selection and select Server with GUI option, select the Add-Ons like Development Tools and click next.

July 2019, Version 3.0



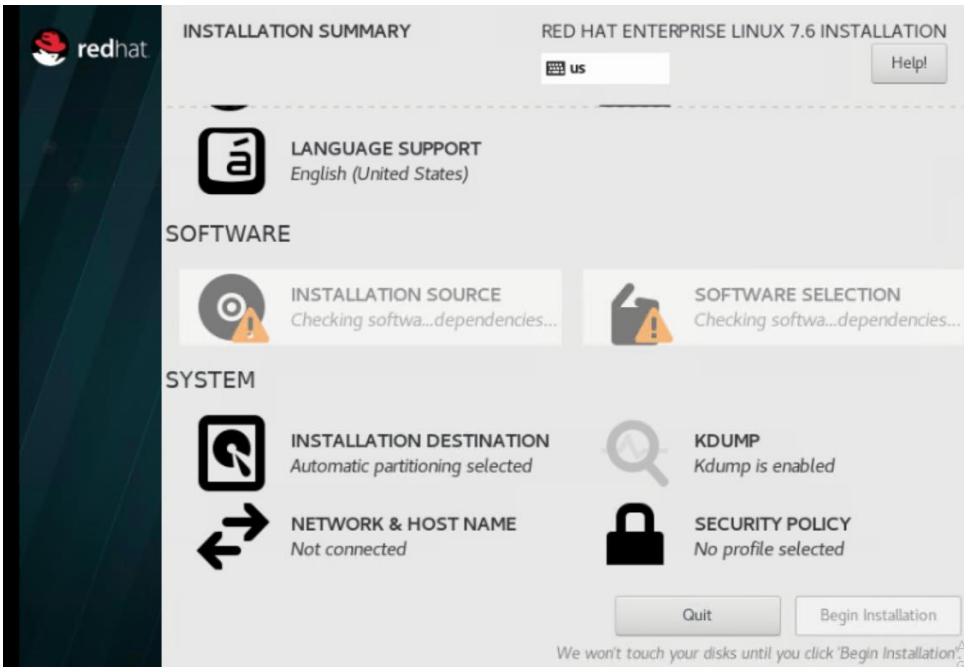
19. In the Installation Summary Screen

    a. Click on the Installation Destination, on the Device Selection screen, accept the "Automatically configure partitioning" option and click Done.

    b. In the Network & Host Name tab, configure the IP address with subnet mask, gateway, search domain and DNS IP address

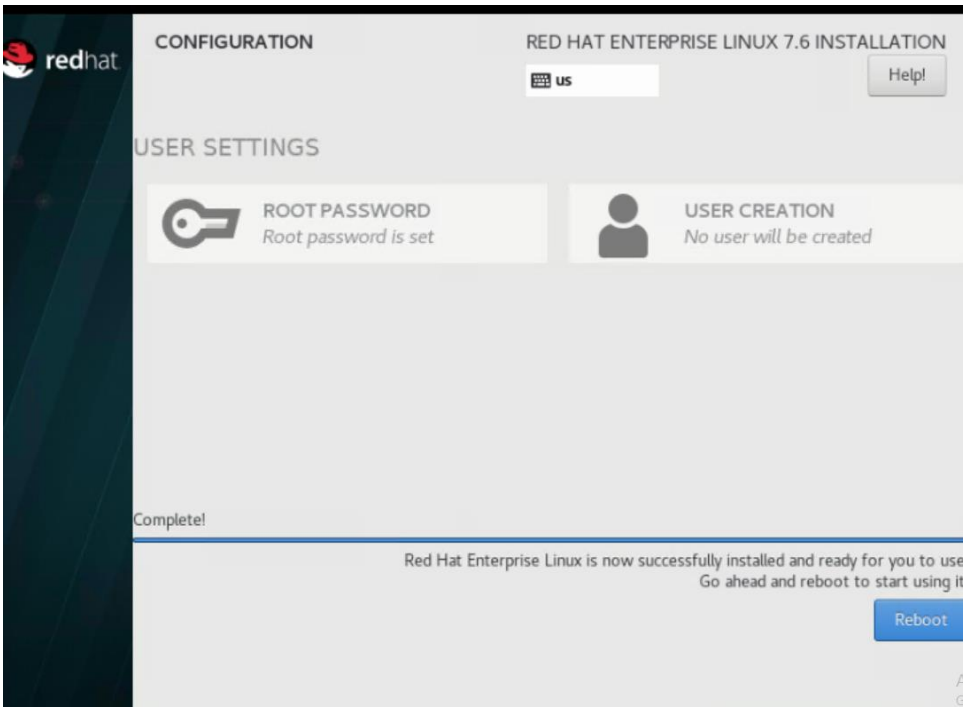
    c. Click Begin Installation.

July 2019, Version 3.0



20. On the configuration screen, set the desired root password. You may also create additional users during the installation process if you wish.

July 2019, Version 3.0



21. The installation process once begun may take 10-15 minutes. Wait until the installation process completes and then click reboot.
22. After the VM reboots, in the End User License Agreement page, Accept the license terms and finish the installation.
23. Once the system is rebooted, follow the on-screen instruction to set the basic settings such as keyboard language and time zone.
24. On the Login screen, login with root user the RHEL 7.6 server is ready to use.

Update OS and install the necessary tools and package

25. Update the OS of the VM to be used as a template. Execute the below command in the terminal.

```
# yum update
```

26. Install VMtools

- Mount the CD image of VMware tools to the VM that is intended to be converted as a template. Execute the following

```
# mkdir -p /mnt/cd /tmp/vmware
```

```
# mount -o loop /dev/sr0 /mnt/cd
```

27. Copy the VMware tools archive from the mounted CD to local partition

```
# cp /mnt/cd/< VMware tools tar.gz filename>  /tmp/vmware
```

28. Extract the content

```
# cd /tmp/vmware/
# tar zxvf <VMware tools tar.gz filename>
```

29. Register to your RedHat subscription using the below command

```
# Subscription-manager register -username=<your username> --password=<your password> --auto-attach
```

30. Ensure open-vm-tools is uninstalled and dependency packages exist on the guest OS

```
# yum remove open-vm-tools
# yum install policycoreutils-python
```

31. Install VMware tools

```
# cd vmware-tools-distrib/
# perl vmware-install.pl -d -f
```

32. Confirm the service is up and running

```
# systemctl status vmware-tools
```

33. once the required packages are installed, you need to unregister your Subscription to the RedHat account using the below command

```
# subscription-manager unregister
```

Disabling IPv6 support in Red Hat Enterprise Linux 7

The following document walks through disabling ipv6 on RHEL 7.6 https://access.redhat.com/solutions/8709#rhel7disable

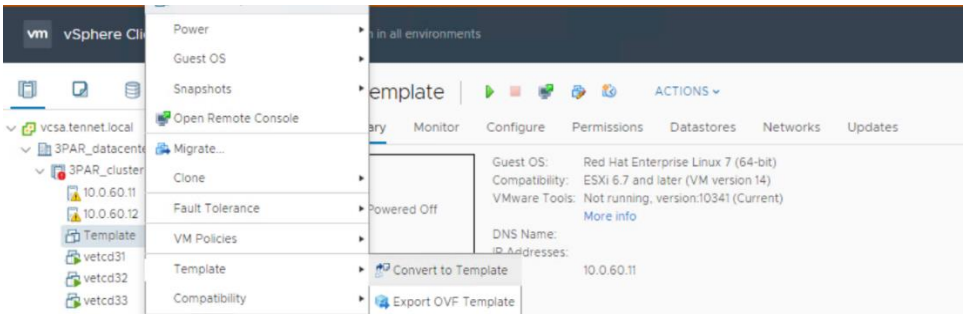
Convert the VM as a template

34. Login to vCSA and select the VM created to be convert to a Template

35. If the VM is powered on, go to actions > power > power off the VM

36. Go to actions and select the template option within that, Click the convert to template option

July 2019, Version 3.0



37. The Template is now ready to be deployed as a VM.

Ansible scripts for Virtual machines

The following ansible script deploy three rancher servers and one virtual machine for Haproxy. All these VM are running RHEL 7.6. To begin deploying the Virtual machines the desired configuration for these virtual machine such as hostname, hardware configuration etc. must be editing in the Vmscripts/VirtualMachineVars.yml. Once the variables are setup as your environment. For the Ansible_Rancher folder run the following play:

```
ansible-playbook -i rhost Vmscripts/DeployrancherVMs.yml
```

Once the virtual machine are created verify the setting and customization setup before proceeding if anything is not as expected you can power down these VM's and deleted them by running the following ansible play:

Power-off

```
ansible-playbook -i rhost Vmscripts/PowerOffVM.yml
```

Delete VM

```
ansible-playbook -i rhost Vmscripts/DeleteVM.yml
```

Once the Virtual machines are created we good to proceed with scripts that are common for both Virtual machine and bare metal deployment

Common steps for either Virtual Machine or Bare metal deployment.

These steps are applicable for the host required for running Rancher sever could be virtual machine or bare metal host. For a Bare metal deployment it assumes that the server are provisioned with required hardware and network and storage connectivity.

Disabling IPv6 support in Red Hat Enterprise Linux 7

The following document walks through disabling ipv6 on RHEL 7.6 https://access.redhat.com/solutions/8709#rhel7disable

Prepare the host for Rancher and load balancer deployment

The following ansible script deploy three rancher servers and one virtual machine for Load Balancing using Haproxy. The variable for this play this are located HostPrepareVars.yml modify the information to match your environment.

```
---
#####Subscription details ########

rhsm_user: <Redhat User id>
rhsm_pass: <Redhat password>
vault_rhsub_user: <Redhat User id>
vault_rhsub_pass: <Redhat password>
ansible_ssh_user: root
ansible_ssh_pass: <Root password>

######## subscription pool_id details#################
pool_ids_vms:
        - pool id for registration
                - pool id for registration
```

Once the variable file is populated with the required information. Run the following ansible-playybook to register your virtual machine or server to Redhat subscriptions services and prepare for rancher deployment:

ansible-playbook -i rhost HostPrepare.yml

July 2019, Version 3.0

Install Docker on servers for Rancher

Using this play we setup Docker CE for rancher servers. This play setup the firewall rules for Kubernetes deployment. This play also setup the sudo user for DockerRoot access on RHEL 7 for Kubernetes deployment. The variable file for this playbook has only one variable this drive you want to install docker to.

```
---
#Patch to the second disk for Docker local storage
second_disk_vms: sda
```

Once the variable file is populated with the appropriate drive value run the following ansible playbook

```
ansible-playbook -i rhost Docker_setup.yml
```

Creating the Cluster Configuration File

RKE uses a cluster configuration file, referred to as cluster.yml to determine what nodes will be in the cluster and how to deploy Kubernetes. There are many configuration options that can be set in the cluster.yml. shown below is sample cluster.yml included with the ansible plays.

```
---
nodes:
  - address: 10.0.8.91
    user: rkeuser
    role: [controlplane,worker,etcd]
  - address: 10.0.8.92
    user: rkeuser
    role: [controlplane,worker,etcd]
  - address: 10.0.8.93
    user: rkeuser
    role: [controlplane,worker,etcd]

services:
  etcd:
    snapshot: true
    creation: 6h
    retention: 24h
```

Creating the Kubernetes cluster

Populate the cluster.yml with your enviroment details as shown above. Once the information is updated run the play Kubernetes-rke.yml this play will generate certificate for kubernetes that self-signed. Then it will bring up the kubernetes cluster.

```
anisble-playbook  Kubernetes-rke.yml
```

Save a copy of the following files in a secure location:

July 2019, Version 3.0

cluster.yml: The RKE cluster configuration file.

kube_config_cluster.yml: The Kubeconfig file for the cluster, this file contains credentials for full access to the cluster.

cluster.rkestate: The Kubernetes Cluster State file, this file contains credentials for full access to the cluster.

The Kubernetes Cluster State file is only created when using RKE v0.2.0 or higher.

Set the kubeconfig path on the ansible host
Using the following command
export KUBECONFIG=<path>/kube_config_cluster.yml

example
export KUBECONFIG=/etc/ansible/rancher/Ansible_Rancher/kube_config_cluster.yml

Test kubernetes connectivity from anisble host
Run
kubectl get pods --all-namespaces

Create service account for tiller and test helm

Once connectivity is established from the ansible host setup the service account tiller within kubernetes. Then create a role binding within kubernetes. Post this initialize helm with the new service account and finally test connectivity. To do all this run the helm-tiller.yml playbook

ansible-playbook helm-tiller.yml

Deploy and configure HAProxy for SSL Pass-through

This playbook runs against the [lb] group in rhost file and can be used to configure a single RHEL server as Haproxy load balancer for rancher. Setup the Haproxy configuration as per your environment by editing /templates/haproxy.cfg file. Sample shown below for rancher specific settings

---

```
#---------------------------------------------------------------------
frontend localhost
    bind *:443
    bind *:80
    option tcplog
    mode tcp
    default_backend nodes

backend nodes
```


July 2019, Version 3.0

```
mode tcp
balance roundrobin
option ssl-hello-chk
server rancher01 10.0.8.91:443 check
server rancher02 10.0.8.92:443 check
server rancher03 10.0.8.93:443 check
```

To execute the play book run the following:
ansible-playbook -i rhost Haproxy_setup.yml

Install Rancher

Rancher installation is managed using the Helm package manager for Kubernetes. Use helm to install the prerequisite and charts to install Rancher.

Prior to running the ansible script update the url for Load balancer under the following task highlighted in yellow

```
- name: install latest version of rancher
  command: "helm install rancher-stable/rancher --name rancher --namespace cattle-system  --set hostname=lb01.tennet.local"
  ignore_errors: no
```

save the changes to playbook then Run the Install_rancher.yml play.

ansible-playbook Install_rancher.yml

Verify the deployment by running “kubectl -n cattle-system get deploy rancher” on the ansible host

Post installation task forRancher

Once Rancher is completed successfully, login via load blancer set the password for the admin user for rancher webconsole then provide public routable name for the accessing Ranncher webconsole UI external. If you don’t have public domain yet enter possible fictitious .com domain name example for this setup we used lb01.tennnnet.com

Your Rancher Highly Available environment is up and running

July 2019, Version 3.0

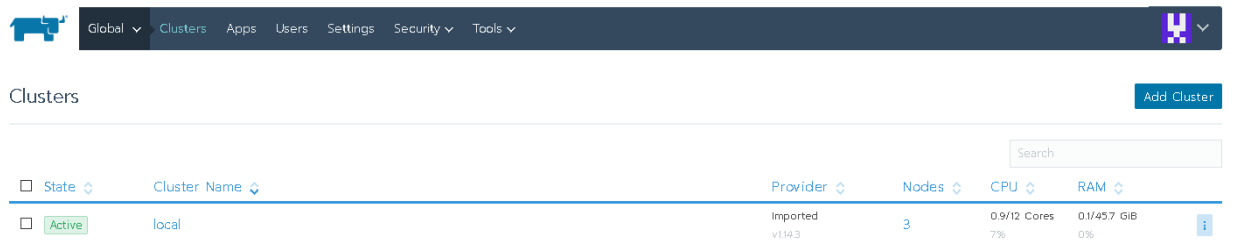


Figure 3. Rancher Web UI

Table A1 below describes the variables used with the VM template.

Appendix A- HPE 3PAR StoreServ integration Rancher

The integration with 3Par for persistent volume requires a separate ETCD cluster this will required additional compute (Virtual Machine or Physical server). To setup the etcd cluster setup do the following:

1. Install and configure the operating system
2. Setup DNS records
3. Setup it up as machine as spate etcd using the instructions mentioned below

https://github.com/hpe-storage/python-hpedockerplugin/blob/master/docs/manual_install_guide_hpe_3par_plugin_with_rancher_kubernetes.md

Resources and additional links (Required)

Red Hat, https://www.Red HatRed HatRed Hat.com

Rancher 2.0 https://rancher.com/docs/rancher/v2.x/en/

HPE Synergy, https://www.hpe.com/info/synergy

HPE 3PAR StoreServ Storage, https://www.hpe.com/info/3PAR

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

July 2019, Version 3.0