

Rancher Deployment Guide

Overview

This document describes the steps required for deployment of Rancher 2.x on virtual and bare metal nodes with RedHat 7.6 and Ubuntu 18.04 OS.

This document is to be used in conjunction with the RA whitepaper available at:

URL - <https://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=a50000789enw>

The above whitepaper describes the RA architecture and other details extensively.

The Deployment Environment

Before executing the solution deployment steps described in this document, ensure that the following pre-requisites are met:

- The compute, storage and network infrastructure is set up
- DNS, DHCP (optional), VMware vCenter are installed
- Ansible Server needs to be deployed with the following configuration:
 - o Ansible Engine installed on the Ansible server
 - o The following repos are enabled on the Ansible server if using RedHat 7.6 OS:
 - redhat-7-server-extras-rpms
 - redhat-7-server-rpms
 - redhat-7-server-ose-3.xx-rpms
 - redhat-7-server-ansible-2.x-rpms (2.7 preferably)

Additional tools required to be installed on Ansible server are:

- rke : Rancher Kubernetes Engine, CLI for building Kubernetes cluster
- kubectl : Kubernetes command line tool
- helm : Package management for Kubernetes

The above tools can be downloaded manually or deployed through automation by using the Toolslocal.yml playbook available in this repository

- ansible-playbook Toolslocal.yml

The Host file for the environment

Edit the Rhost file per your environment before running any playbooks. The sample Rhost file is available in this repo.

The FQDN and ip addresses will depend upon your setup.

Creating the Virtual Machines

The virtual machines can be created using VMware vCenter 6.7 manually using the standard procedure described in VMware documents or they can be created by running the following playbooks.

The following scripts create or delete the virtual machines used for the Rancher and user clusters as well as the load balancer described in this RA:

For VM creation:

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

For deleting VMs:

- `ansible-playbook -i Rhost Vmscripts/DeleteVMs.yml`

For VM power-off:

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

After virtual machines are created, the following steps can be executed for both virtual and bare metal deployments.

Prepare the hosts for Rancher deployment

Run the following playbook for registering the virtual machines and bare metal servers with Redhat subscription services and also preparing for Rancher deployment. Edit the HostPrepareVars.yml per your environment before running the playbook.

- `ansible-playbook -i Rhost HostPrepare.yml`

Install the Docker on VMS and servers

Run the following playbook to install Docker on VMs and Servers. This playbook also sets up firewall rules for Kubernetes deployment as well as sudo user for root access to RedHat OS.

The variable file used in this playbook has only one Drive set up for Docker installation. Therefore setup another Drive for Docker storage using the following:

Patch to second disk for Docker local storage#

`Second_disk_vms : sda`

Run the playbook:

- `Ansible-playbook -i Rhost Docker_setup.yml`

Creating cluster configuration

RKE uses the configuration file `cluster.yml` for setting up the Kubernetes cluster with the following roles assigned to the nodes in the cluster. The roles are: ControlPlane, Worker, etcd.

Run the following playbook for Kubernetes cluster deployment:

- `ansible-playbook Kubernetes-rke.yml`

This will generate the Kubernetes cluster configuration file called `kube_config_cluster.yml`. Using this file the `kubectl` can communicate with the Kubernetes cluster.

Save the `kube_config_cluster.yml` to certain location and set the path.

Set path on ansible server for the kube cluster configuration file:

`export KUBECONFIG= <path to kubeconfig file>/kube_config-cluster.yml`

You may check the connectivity of Kubernetes cluster from the Ansible server:

- `kubectl get pods --all-namespaces`

Create Service account for tiller

Run the following playbook to install the service account tiller within Kubernetes and initialize helm with this new service account.

- `ansible-playbook helm-tiller.yml`

Configure the Load-balancer

Configure the HAProxy load balancer to distribute the traffic on the Kubernetes cluster nodes. Edit the `/templates/haproxy.cfg` file per your environment and run the following:

- `ansible-playbook -i Rhost Haproxy_setup.yml`

Install Rancher

Rancher installation is managed by helm package manager for Kubernetes. Use helm to install pre-requisite and charts to install Rancher.

Prior to running the playbook, update the url of the load balancer:

- name : install latest version of Rancher
- command : "helm install rancher-stable/rancher --name rancher --namespace cattle-system --set hostname = <url_FQDN_of_load_balancer>
- ignore_errors : no

save the above changes to the Install_rancher.yml playbook and run the following:

- ansible-playbook install_rancher.yml

Verify the deployment by running the following on the Ansible server:

- kubectl -n cattle-system get deploy rancher

Access the Rancher web-console

After successful deployment of the Rancher, login through the load balancer on Rancher and provide the externally accessible url name. Use the Web browser to access Rancher Web-console using the load-balancer url that is referred to as Rancher server url. You will see the Rancher Web-console opened on your screen with all the Rancher cluster and user clusters details.