Rancher

25 January 2021 15:40

```
[root@localhost ~]# hostname
rkewstation
[root@localhost ~]# cat /etc/hosts
192.168.1.74 rkewstation.lab.rancher.com rkewstation
192.168.1.75 rkeserver.lab.rancher.com rkeserver
```

yum update -y

Rancher on Docker

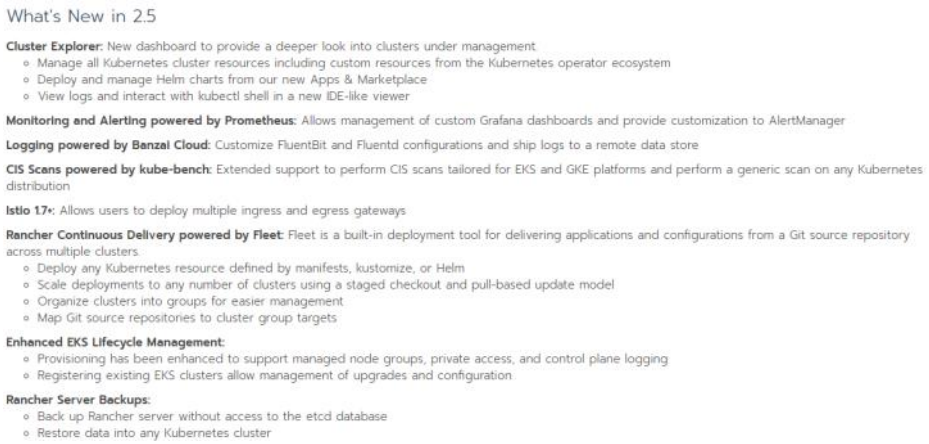
- *curl https://releases.rancher.com/install-docker/19.03.sh | sh*
- *docker run -d --restart=unless-stopped -p 80:80 -p 443:443 --privileged rancher/rancher*

```
[root@rkewstation ~]# docker run -d --restart=unless-stopped -p 80:80 -p 443:443 --privileged rancher/rancher
Unable to find image 'rancher/rancher:latest' locally
latest: Pulling from rancher/rancher
f22ccc0b8772: Downloading [==================================================>  ]  26.12MB/26.71MB
3cf8fb62ba5f: Download complete
e80c964ece6a: Download complete
177bd5a25689: Download complete
```

```
[root@rkewstation ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
                    NAMES
67bf52b9b6b9        rancher/rancher     "entrypoint.sh"     About a minute ago   Up About a minute   0.0.0.0:80->80/tcp,
0:443->443/tcp   fervent_yonath
[root@rkewstation ~]#
```

- *# firewall-cmd --add-service=https --permanent; firewall-cmd --add-service=http --permanent; firewall-cmd --reload*

Access Rancher on your VM IP address



What's New in 2.5

Cluster Explorer: New dashboard to provide a deeper look into clusters under management
- Manage all Kubernetes cluster resources including custom resources from the Kubernetes operator ecosystem
- Deploy and manage Helm charts from our new Apps & Marketplace
- View logs and interact with kubectl shell in a new IDE-like viewer

Monitoring and Alerting powered by Prometheus: Allows management of custom Grafana dashboards and provide customization to AlertManager

Logging powered by Banzai Cloud: Customize FluentBit and Fluentd configurations and ship logs to a remote data store

CIS Scans powered by kube-bench: Extended support to perform CIS scans tailored for EKS and GKE platforms and perform a generic scan on any Kubernetes distribution

Istio 1.7+: Allows users to deploy multiple ingress and egress gateways

Rancher Continuous Delivery powered by Fleet: Fleet is a built-in deployment tool for delivering applications and configurations from a Git source repository across multiple clusters
- Deploy any Kubernetes resource defined by manifests, kustomize, or Helm
- Scale deployments to any number of clusters using a staged checkout and pull-based update model
- Organize clusters into groups for easier management
- Map Git source repositories to cluster group targets

Enhanced EKS Lifecycle Management:
- Provisioning has been enhanced to support managed node groups, private access, and control plane logging
- Registering existing EKS clusters allow management of upgrades and configuration

Rancher Server Backups:
- Back up Rancher server without access to the etcd database
- Restore data into any Kubernetes cluster

RKE cluster in Production

- **RKE in kubernetes to cover HA**

GCP

workstation - 2 vCPU + 4 GB
rkeserver - 4 vCPU + 16 GB

On-Prem

RKE Setup:

- **Workstation**
  - To run rke, kubectl, and helm
- **Server**
  - RKE, Rancher Server

On workstation:

- **Download** https://github.com/rancher/rke/releases/download/v1.1.0/rke_linux-amd64 for **linux from the RKE GitHub repository's releases.**
- **Rename the Binary to rke in MacOS/Linux or rke.exe in Windows.**
- **Activate the Binary**
  a. Copy the binary to /usr/local/bin/
  b. Use chmod to change the permission of the binary
  c. Test the functionality
     [root@rke ~]# rke --version
     rke version v1.1.0

```
[root@rkewstation ~]# mv rke_linux-amd64.1 rke
[root@rkewstation ~]# cp rke /usr/local/bin/
[root@rkewstation ~]# chmod 755 /usr/local/bin/rke
[root@rkewstation ~]# rke --version
rke version v1.1.0
[root@rkewstation ~]#
```

- **Create rke user, set the password**
  # useradd rke; echo -e "redhat\nredhat" |passwd rke
  # echo "rke ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers.d/rke
  # su - rke

$ sudo -l
$ sudo yum install wget -y

```
[root@rkewstation ~]# cat /etc/hosts
192.168.1.74 rkewstation.lab.rancher.com rkewstation
192.168.1.75 rkeserver.lab.rancher.com rkeserver
```

- **Install kubectl**
  - curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
  - sudo mv kubectl /usr/local/bin/ && sudo chmod 755 /usr/local/bin/kubectl

- **Password less authentication for rke user**
  $ ssh-keygen -t rsa
  $ ssh-copy-id rke@rkeserver

On Server:
(Node Preparation)

1. **Download and install docker**
   a. sudo curl https://releases.rancher.com/install-docker/18.09.2.sh | sh

2. **Create rke user, set the password**
   # useradd rke; echo -e "redhat\nredhat" |passwd rke
   # echo "rke ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers.d/rke
   # su - rke
   $ sudo -l
   $ sudo yum install wget -y

3. **Enable all required modules - use root user**
   # for module in br_netfilter ip6_udp_tunnel ip_set ip_set_hash_ip ip_set_hash_net iptable_filter iptable_nat iptable_mangle i ptable_raw nf_conntrack_netlink nf_conntrack nf_conntrack_ipv4 nf_defrag_ipv4 nf_nat nf_nat_ipv4 nf_nat_masquerade_ipv4 nfnetlink udp_tunnel veth vxlan x_tables xt_addrtype xt_conntrack xt_comment xt_mark xt_multiport xt_n at xt_recent xt_set xt_statistic xt_tcpudp; do modprobe $module ; done

```
[root@rkeserver ~]# for module in br_netfilter ip6_udp_tunnel ip_set ip_set_hash_ip ip_set_hash_net iptable_filter
iptable_nat iptable_mangle iptable_raw nf_conntrack_netlink nf_conntrack nf_conntrack_ipv4   nf_defrag_ipv4 nf_nat
nf_nat_ipv4 nf_nat_masquerade_ipv4 nfnetlink udp_tunnel veth vxlan x_tables xt_addrtype xt_conntrack xt_comment xt_
mark xt_multiport xt_nat xt_recent xt_set  xt_statistic xt_tcpudp; do modprobe $module ; done
[root@rkeserver ~]#
```

1. **Disable swap and set sysctl parameters** - use root user
   $ sudo swapoff -a
   $ sudo tee -a /etc/sysctl.d/99-kubernetes.conf <<EOF
   net.bridge.bridge-nf-call-iptables = 1
   net.ipv4.ip_forward = 1
   net.bridge.bridge-nf-call-ip6tables = 1
   EOF

   $ sudo sysctl -p

2. **Enable the docker service**
   $ sudo systemctl enable --now docker

3. **Check Docker version**
   $ sudo docker version --format '{{.Server.Version}}'

4. **Configure user rke to be part of docker group**
   $ sudo usermod -aG docker rke

5. **Enable firewall ports**
   # for i in 22 80 443 179 5473 6443 8472 2376 8472 2379-2380 9099 10250 10251 10252 10254 30000-32767; do sudo firewall-cmd --add-port=${i}/tcp --permanent; done
   # for i in 8285 8472 4789 30000-32767; do sudo firewall-cmd --add-port=${i}/udp --permanent; done
   # sudo firewall-cmd --reload

6. **Allow SSH TCP Forwarding**
   sudo vi /etc/ssh/sshd_config
   AllowTcpForwarding yes
   $ sudo systemctl reload sshd

On workstation @ deploy cluster

- **RKE Config**
  - $ rke config
  - $ rke up
  - $ export KUBECONFIG=./kube_config_cluster.yml

```
[rke@rkewstation ~]$ rke config
[+] Cluster Level SSH Private Key Path [~/.ssh/id_rsa]:
[+] Number of Hosts [1]:
[+] SSH Address of host (1) [none]: 192.168.1.75
[+] SSH Port of host (1) [22]:
[+] SSH Private Key Path of host (192.168.1.75) [none]:
[-] You have entered empty SSH key path, trying fetch from SSH key parameter
[+] SSH Private Key of host (192.168.1.75) [none]:
[-] You have entered empty SSH key, defaulting to cluster level SSH key: ~/.ssh/id_rsa
[+] SSH User of host (192.168.1.75) [ubuntu]: rke
[+] Is host (192.168.1.75) a Control Plane host (y/n)? [y]:
[+] Is host (192.168.1.75) a Worker host (y/n)? [n]: y
[+] Is host (192.168.1.75) an etcd host (y/n)? [n]: y
[+] Override Hostname of host (192.168.1.75) [none]: rkeserver
[+] Internal IP of host (192.168.1.75) [none]: 192.168.1.75
[+] Docker socket path on host (192.168.1.75) [/var/run/docker.sock]:
[+] Network Plugin Type (flannel, calico, weave, canal) [canal]:
[+] Authentication Strategy [x509]:
[+] Authorization Mode (rbac, none) [rbac]:
[+] Kubernetes Docker image [rancher/hyperkube:v1.17.4-rancher1]:
[+] Cluster domain [cluster.local]:
[+] Service Cluster IP Range [10.43.0.0/16]:
[+] Enable PodSecurityPolicy [n]:
[+] Cluster Network CIDR [10.42.0.0/16]:
[+] Cluster DNS Service IP [10.43.0.10]:
[+] Add addon manifest URLs or YAML files [no]:
[rke@rkewstation ~]$
[rke@rkewstation ~]$
```

```
[rke@workstation ~]$ rke config
[+] Cluster Level SSH Private Key Path [~/.ssh/id_rsa]:
[+] Number of Hosts [1]:
[+] SSH Address of host (1) [none]: 10.128.0.19
[+] SSH Port of host (1) [22]:
[+] SSH Private Key Path of host (10.128.0.19) [none]:
[-] You have entered empty SSH key path, trying fetch from SSH key parameter
[+] SSH Private Key of host (10.128.0.19) [none]:
[-] You have entered empty SSH key, defaulting to cluster level SSH key: ~/.ssh/id_rsa
[+] SSH User of host (10.128.0.19) [ubuntu]: rke
[+] Is host (10.128.0.19) a Control Plane host (y/n)? [y]: y
[+] Is host (10.128.0.19) a Worker host (y/n)? [n]: y
[+] Is host (10.128.0.19) an etcd host (y/n)? [n]: y
[+] Override Hostname of host (10.128.0.19) [none]: rkeserver.lab.example.com
[+] Internal IP of host (10.128.0.19) [none]:
[+] Docker socket path on host (10.128.0.19) [/var/run/docker.sock]:
[+] Network Plugin Type (flannel, calico, weave, canal) [canal]:
[+] Authentication Strategy [x509]:
[+] Authorization Mode (rbac, none) [rbac]:
[+] Kubernetes Docker image [rancher/hyperkube:v1.17.4-rancher1]:
[+] Cluster domain [cluster.local]:
[+] Service Cluster IP Range [10.43.0.0/16]:
[+] Enable PodSecurityPolicy [n]:
[+] Cluster Network CIDR [10.42.0.0/16]:
[+] Cluster DNS Service IP [10.43.0.10]:
[+] Add addon manifest URLs or YAML files [no]:
[rke@workstation ~]$
```

```
INFO[0332] [addons] Setting up user addons
INFO[0332] [addons] no user addons defined
INFO[0332] Finished building Kubernetes cluster successfully
[rke@rkewstation ~]$
```

```
[rke@rkewstation ~]$ export KUBECONFIG=./kube_config_cluster.yml
[rke@rkewstation ~]$ kubectl get nodes
NAME         STATUS   ROLES                      AGE     VERSION
rkeserver    Ready    controlplane,etcd,worker   2m50s   v1.17.4
[rke@rkewstation ~]$
```

```
[rke@rkewstation ~]$ kubectl get pods -A
NAMESPACE       NAME                                      READY   STATUS      RESTARTS   AGE
ingress-nginx   default-http-backend-67cf578fc4-mgdfw      1/1     Running     0          8m24s
ingress-nginx   nginx-ingress-controller-hd5hk            1/1     Running     0          8m24s
kube-system     canal-r5db9                               2/2     Running     0          9m38s
kube-system     coredns-7c5566588d-pd7wx                  1/1     Running     0          9m33s
kube-system     coredns-autoscaler-65bfc8d47d-8xgd7       1/1     Running     0          9m32s
kube-system     metrics-server-6b55c64f86-t9skh           1/1     Running     0          9m27s
kube-system     rke-coredns-addon-deploy-job-7hqkg        0/1     Completed   0          9m36s
```

***** RKE cluster is built**

Rancher Server on RKE cluster

- **Deploy helm**

  $ wget https://get.helm.sh/helm-v3.2.4-linux-amd64.tar.gz
  $ tar zxvf helm-v3.2.4-linux-amd64.tar.gz
  $ sudo mv linux-amd64/helm /usr/local/bin/ && chmod +x /usr/local/bin/helm

- **Deploy cert-manager for self-sign certs**

  $ kubectl apply --validate=false -f https://github.com/jetstack/cert-manager/releases/download/v1.0.4/cert-manager.crds.yaml

```
[rke@rkewstation ~]$ kubectl apply --validate=false -f https://github.com/jetstack/cert-manager/releases/download/v1.0.4/cert-m
anager.crds.yaml

customresourcedefinition.apiextensions.k8s.io/certificaterequests.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/certificates.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/challenges.acme.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/clusterissuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/issuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/orders.acme.cert-manager.io created
[rke@rkewstation ~]$
```

[rke@workstation ~]$ kubectl create namespace cert-manager
namespace/cert-manager created
[rke@workstation ~]$ helm repo add jetstack https://charts.jetstack.io
"jetstack" has been added to your repositories
[rke@workstation ~]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jetstack" chart repository
...Successfully got an update from the "rancher-latest" chart repository
Update Complete. å Happy Helming!å

```
[rke@rkewstation ~]$  kubectl create namespace cert-manager
namespace/cert-manager created
[rke@rkewstation ~]$ helm repo add jetstack https://charts.jetstack.io
"jetstack" has been added to your repositories
[rke@rkewstation ~]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jetstack" chart repository
...Successfully got an update from the "rancher-latest" chart repository
Update Complete. □ Happy Helming!□
[rke@rkewstation ~]$
```

$ helm install cert-manager jetstack/cert-manager --namespace cert-manager --version v1.0.4

```
[rke@rkewstation ~]$ helm install cert-manager jetstack/cert-manager --namespace cert-manager --version v1.0.4
NAME: cert-manager
LAST DEPLOYED: Mon Jan 25 11:32:39 2021
NAMESPACE: cert-manager
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
cert-manager has been deployed successfully!
```

[rke@workstation ~]$ kubectl get pods --namespace cert-manager

```
[rke@rkewstation ~]$ kubectl get pods --namespace cert-manager
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-85c9b9bb44-cqc76              1/1     Running   0          72s
cert-manager-cainjector-78fc9bb777-m5ms5   1/1     Running   0          72s
cert-manager-webhook-695f8b56cd-bc8s5      1/1     Running   0          72s
[rke@rkewstation ~]$
```

[rke@workstation ~]$ kubectl create namespace cattle-system
namespace/cattle-system created
[rke@workstation ~]$

```
[rke@rkewstation ~]$ kubectl create namespace cattle-system
namespace/cattle-system created
[rke@rkewstation ~]$ kubectl  get ns
NAME              STATUS   AGE
cattle-system     Active   5s
default           Active   3h18m
ingress-nginx     Active   3h16m
kube-node-lease   Active   3h18m
kube-public       Active   3h18m
kube-system       Active   3h18m
[rke@rkewstation ~]$
```

[rke@workstation ~]$ helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
"rancher-latest" has been added to your repositories

```
[rke@rkewstation ~]$ helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
"rancher-latest" has been added to your repositories
[rke@rkewstation ~]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "rancher-latest" chart repository
Update Complete. □ Happy Helming!□
[rke@rkewstation ~]$
```

$ helm install rancher rancher-latest/rancher --namespace cattle-system --set hostname=rkeserver.lab.rancher.com

```
[rke@rkewstation ~]$ helm install rancher rancher-latest/rancher --namespace cattle-system --set h
ostname=rkeserver.lab.rancher.com
NAME: rancher
LAST DEPLOYED: Mon Jan 25 11:38:58 2021
NAMESPACE: cattle-system
STATUS: deployed
REVISION: 1
```

[rke@workstation ~]$ kubectl -n cattle-system get deploy rancher

[rke@workstation ~]$ kubectl -n cattle-system rollout status deploy/rancher

```
[rke@rkewstation ~]$ kubectl -n cattle-system rollout status deploy rancher
Waiting for deployment "rancher" rollout to finish: 0 of 3 updated replicas are available...
Waiting for deployment "rancher" rollout to finish: 1 of 3 updated replicas are available...
Waiting for deployment "rancher" rollout to finish: 2 of 3 updated replicas are available...


deployment "rancher" successfully rolled out
```