Deploy NONRTRIC in Minikube

This article helps you to deploy the nonrtric components in your local VM in minikube environment.

- Prerequisite
- Install Minikube
- Start minikube
- Install Kubectl
- Install Helm
- Install Nonrtric
- Troubleshoot
    - Install Conntrack
    - Turn off swap
    - Increase the CPU
    - Install socat
- Version Details
- Setup
- Alternative to Deploy NONRTRIC in Minikube
    - Prerequisite
    - Install Ubuntu 22.04
    - Install Minikube
    - Start Minikube
    - Open the Kubernetes Dashboard

Prerequisite

- *Docker*

Install Minikube

Login into your vm and run the below command as sudo user in the terminal. This will install the latest version of minikube in your vm.

minikube install

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \
  && chmod +x minikube

sudo mkdir -p /usr/local/bin/
sudo install minikube /usr/local/bin/
```

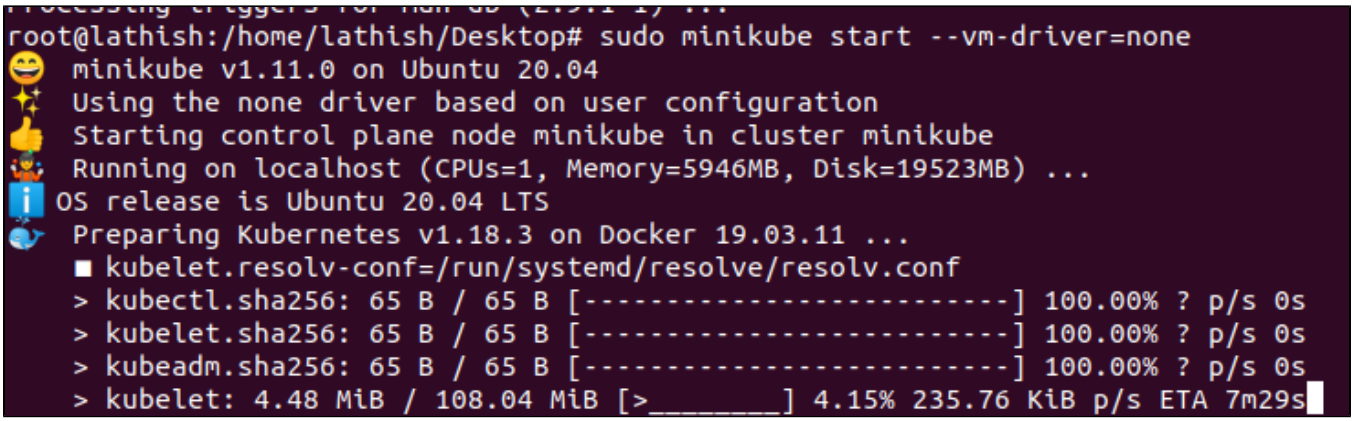
Start minikube

You can start the minikube with the below command. Check Troubleshoot section for any potential errors.

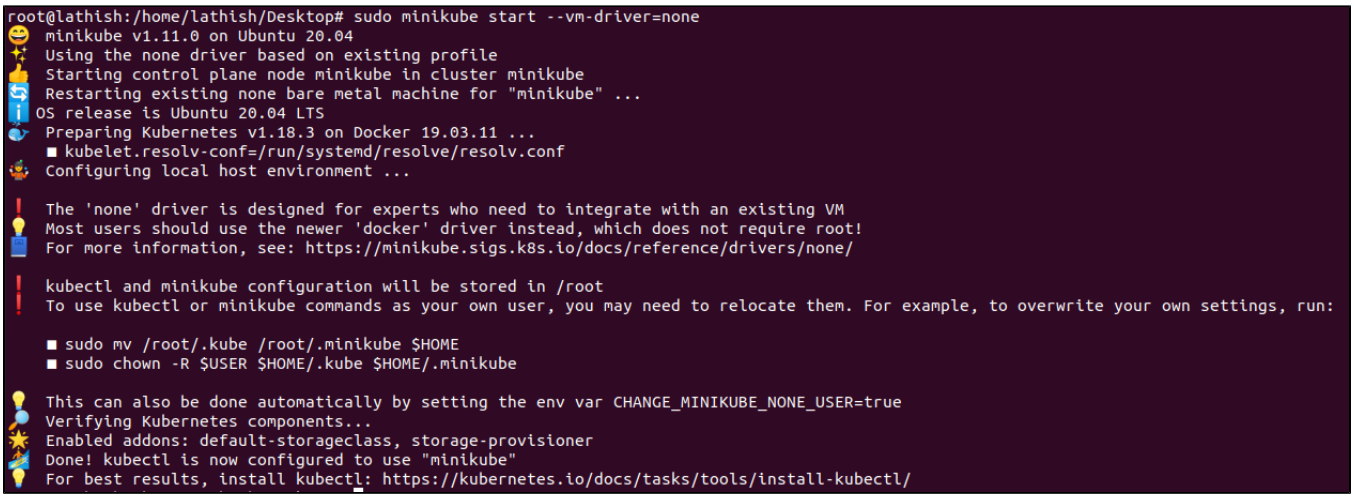
start minikube

```
sudo minikube start --vm-driver=none
```

If the above command is successful, you should see the below logs in your terminal,



If there are no issues then you could see below success message,



Install Kubectl

Kubectl command enables us to run commands against k8's cluster.

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

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

kubectl version --client
Client Version: version.Info{Major:"1", Minor:"18", GitVersion:"v1.18.3", GitCommit:"
2e7996e3e2712684bc73f0dec0200d64eec7fe40", GitTreeState:"clean", BuildDate:"2020-05-20T12:52:00Z", GoVersion:"
go1.13.9", Compiler:"gc", Platform:"linux/amd64"}
```

Install Helm

Run the below command to install helm version 2.16,

install Helm

```
curl -L https://git.io/get_helm.sh | bash
```

Run the below command to install latest helm, Create yaml file to define the service account & cluster role binding for helm,

helm tiller

```
cat > tiller-serviceaccount.yaml << EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: tiller-clusterrolebinding
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: ""
EOF
```

Create ServiceAccount

```
sudo kubectl create -f tiller-serviceaccount.yaml
```

helm init

```
helm init --service-account tiller --upgrade
```

After initializing helm you can call helm version and you should see below response,

```
root@lathish:/home/lathish/dev/code# helm version
Client: &version.Version{SemVer:"v2.16.7", GitCommit:"5f2584fd3d35552c4af26036f0c464191287986b", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.16.7", GitCommit:"5f2584fd3d35552c4af26036f0c464191287986b", GitTreeState:"clean"}
```

Install Nonrtric

Clone IT/dep repo and go to /dep/bin and run below command,

Install nonrtric

```
./deploy-nonrtric -f ../nonrtric/RECIPE_EXAMPLE/example_recipe.yaml
```

If the deployment is successful, you could below logs,

```
NAME:   r2-dev-nonrtric
LAST DEPLOYED: Mon Jun  8 22:00:39 2020
NAMESPACE: nonrtric
STATUS: DEPLOYED

RESOURCES:
==> v1/ConfigMap
NAME                              DATA  AGE
controlpanel-configmap            1     0s
policymanagementservice-configmap  1     0s

==> v1/Deployment
NAME                    READY  UP-TO-DATE  AVAILABLE  AGE
a1controller            0/1    1           0          0s
controlpanel            0/1    1           0          0s
db                      0/1    1           0          0s
policymanagementservice  0/1    1           0          0s

==> v1/Pod(related)
NAME                                    READY  STATUS             RESTARTS  AGE
a1-sim-osc-0                            0/1    ContainerCreating  0         0s
a1-sim-std-0                            0/1    Pending            0         0s
a1-sim-osc-0                            0/1    ContainerCreating  0         0s
a1-sim-std-0                            0/1    Pending            0         0s
a1controller-556c8f4f96-5484v           0/1    Pending            0         0s
controlpanel-549684cbb7-c2dct           0/1    ContainerCreating  0         0s
db-6fcb9d97c7-c5glb                     0/1    ContainerCreating  0         0s
policymanagementservice-5d8d4845cc-6b2m9  0/1    ContainerCreating  0         0s

==> v1/Service
NAME                    TYPE       CLUSTER-IP      EXTERNAL-IP  PORT(S)                        AGE
a1-sim                  ClusterIP  None            <none>       8085/TCP,8185/TCP              0s
a1controller            ClusterIP  10.98.106.46    <none>       8282/TCP,8383/TCP              0s
controlpanel            NodePort   10.107.124.215  <none>       8080:30091/TCP,8081:30092/TCP  0s
dbhost                  ClusterIP  10.109.70.36    <none>       3306/TCP                       0s
policymanagementservice  NodePort   10.98.191.144   <none>       9080:30093/TCP,9081:30094/TCP  0s
sdnctldb01              ClusterIP  10.98.88.78     <none>       3306/TCP                       0s

==> v1/StatefulSet
NAME        READY  AGE
a1-sim-osc  0/2    0s
a1-sim-std  0/2    0s
```

Note: It may take a while for A1 controller image to be downloaded.

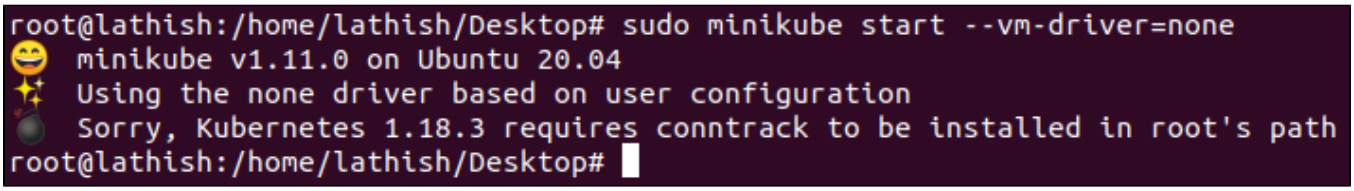
Kubectl -n nonrtric get pod, will show pod status

```
root@lathish:/home/lathish/dev/code/dep/bin# kubectl -n nonrtric get pod
NAME                                    READY   STATUS    RESTARTS   AGE
a1-sim-osc-0                            1/1     Running   0          8m37s
a1-sim-osc-1                            1/1     Running   0          3m10s
a1-sim-std-0                            1/1     Running   0          8m37s
a1-sim-std-1                            1/1     Running   0          2m24s
a1controller-556c8f4f96-mqgnb           1/1     Running   1          8m37s
controlpanel-549684cbb7-5ltjx           1/1     Running   0          8m37s
db-6fcb9d97c7-th58k                     1/1     Running   0          8m37s
policymanagementservice-5d8d4845cc-68mqq  1/1     Running   0          8m37s
```

Troubleshoot

Install Conntrack

1. When you run minikube start you may get below error. Basically this is connection tracking module needed for kubernetes.



Install conntrack,

install conntrack

```
sudo apt-get install conntrack
```

Turn off swap

Once k8's commands are installed , minikube will start the initialization process and you may get below error,

Simply turn off the swap,

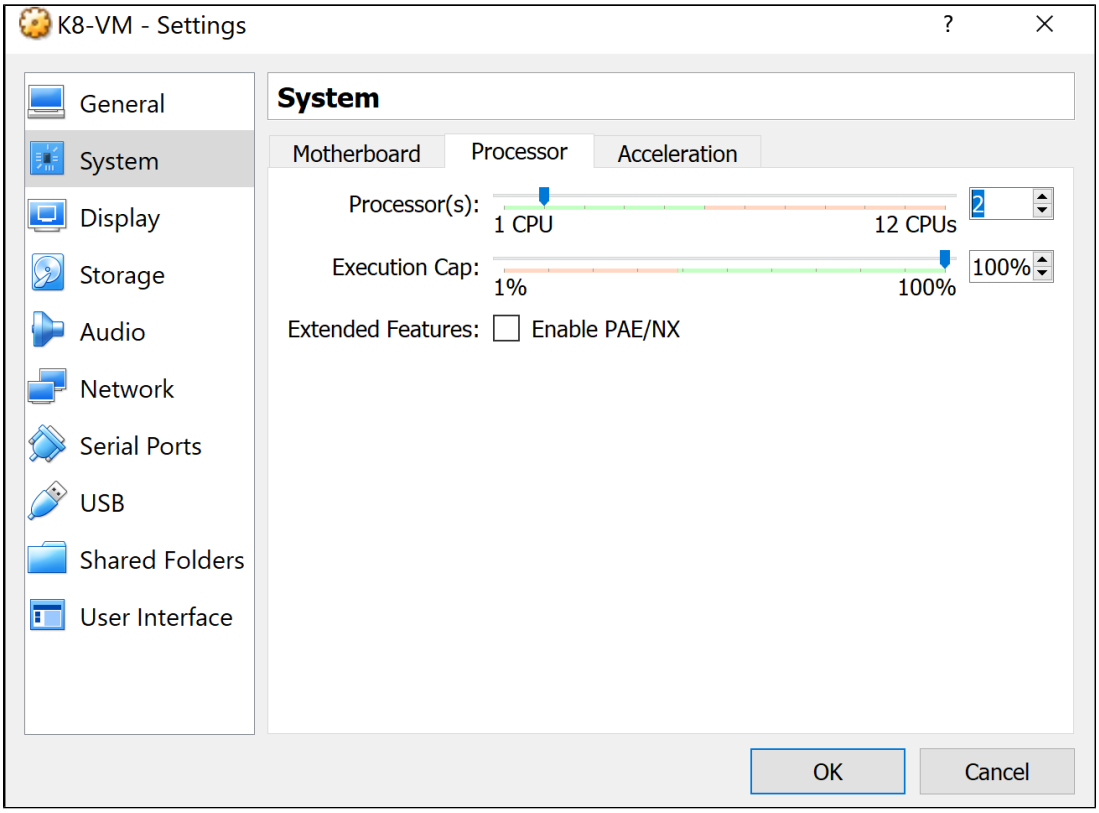
swap-turnoff

```
sudo swapoff -a
```

Increase the CPU

You need to increase the processors of the VM. To this go to the virtual box settings and increase it to 2.

```
  Error starting cluster: run: /bin/bash -c "sudo env PATH=/var/lib/minikube/binaries/v1.18.3:$PATH kubeadm init --config /var/tmp/minikube/kubeadm.yaml  --ignore-preflight-errors=DirAvailable--etc-kubernetes-manifests,DirAvailable--var-lib-minikube,DirAvailable--var-lib-minikube-etcd,FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml,FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml,FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml,FileAvailable--etc-kubernetes-manifests-etcd.yaml,Port-10250,Swap": exit status 1
stdout:
[init] Using Kubernetes version: v1.18.3
[preflight] Running pre-flight checks

stderr:
W0608 16:41:18.670438   15763 configset.go:202] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io]
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
        [WARNING FileExisting-ethtool]: ethtool not found in system path
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher


  minikube is exiting due to an error. If the above message is not useful, open an issue:
  https://github.com/kubernetes/minikube/issues/new/choose

  failed to start node: startup failed: run: /bin/bash -c "sudo env PATH=/var/lib/minikube/binaries/v1.18.3:$PATH kubeadm init --config /var/tmp/minikube/kubeadm.yaml  --ignore-preflight-errors=DirAvailable--etc-kubernetes-manifests,DirAvailable--var-lib-minikube,DirAvailable--var-lib-minikube-etcd,FileAvailable--etc-kubernetes-manifests-kube-scheduler.yaml,FileAvailable--etc-kubernetes-manifests-kube-apiserver.yaml,FileAvailable--etc-kubernetes-manifests-kube-controller-manager.yaml,FileAvailable--etc-kubernetes-manifests-etcd.yaml,Port-10250,Swap": exit status 1
stdout:
[init] Using Kubernetes version: v1.18.3
[preflight] Running pre-flight checks

stderr:
W0608 16:41:18.670438   15763 configset.go:202] WARNING: kubeadm cannot validate component configs for API groups [kubelet.config.k8s.io kubeproxy.config.k8s.io]
        [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
        [WARNING FileExisting-ethtool]: ethtool not found in system path
        [WARNING FileExisting-socat]: socat not found in system path
        [WARNING Service-Kubelet]: kubelet service is not enabled, please run 'systemctl enable kubelet.service'
error execution phase preflight: [preflight] Some fatal errors occurred:
        [ERROR NumCPU]: the number of available CPUs 1 is less than the required 2
[preflight] If you know what you are doing, you can make a check non-fatal with `--ignore-preflight-errors=...`
To see the stack trace of this error execute with --v=5 or higher


  minikube is exiting due to an error. If the above message is not useful, open an issue:
  https://github.com/kubernetes/minikube/issues/new/choose
```



Install socat

```
Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users' policy.
To prevent this, run `helm init` with the --tiller-tls-verify flag.
For more information on securing your installation see: https://v2.helm.sh/docs/securing_installation/
root@lathish:/home/lathish/dev/code# helm version
Client: &version.Version{SemVer:"v2.16.7", GitCommit:"5f2584fd3d35552c4af26036f0c464191287986b", GitTreeState:"clean"}
E0608 17:18:13.352752   10901 portforward.go:400] an error occurred forwarding 43175 -> 44134: error forwarding port 44134 to pod 8d248ca8434e20215b4df6605bec199c0991f8bdab261c636de5567279b10a5f,
 uid : unable to do port forwarding: socat not found
E0608 17:18:14.361204   10901 portforward.go:400] an error occurred forwarding 43175 -> 44134: error forwarding port 44134 to pod 8d248ca8434e20215b4df6605bec199c0991f8bdab261c636de5567279b10a5f,
 uid : unable to do port forwarding: socat not found
E0608 17:18:15.685543   10901 portforward.go:400] an error occurred forwarding 43175 -> 44134: error forwarding port 44134 to pod 8d248ca8434e20215b4df6605bec199c0991f8bdab261c636de5567279b10a5f,
 uid : unable to do port forwarding: socat not found
^C
```

Install socat

```
apt install socat
```

Version Details

- *docker - 19.03*
- *helm - 2.16*
- *minikube - 1.11*
- *kubernetes - 1.18*

Setup

All the above setup is done in Oracle Virtual box VM and you can try it in different OS's and feel free to update this page

You have Oracle Virtual box installed with Ubuntu VM.

For more information, refer

https://www.virtualbox.org/

https://ubuntu.com/download/desktop

Alternative to Deploy NONRTRIC in Minikube

There is an alternative way, described in the procedure below to run NonRT-RIC kubenetes cluster in Minikube running on Oracle Virtual Box with Ubuntu 22.04.

Via an installation script that brings up Minikube and allow to deploy NonRT-RIC cluster on it. Besides, the Kubernetes GUI could be started to see overall deployment.

Prerequisite

VirtualBox version 6.1. (Can be downloaded from this page Download VirtualBox)

Install Ubuntu 22.04

Ubuntu version 22.04. (Can be downloaded from this page Download Ubuntu)

Basic configuration of Ubuntu on Virtual Box can be found below.

General

Name: ubunut2204
Operating System: Ubuntu (64-bit)

System

Base Memory: 16384 MB
Processors: 4
Boot Order: Hard Disk, Optical, Floppy
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 128 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: ubunut2204.vhd (Normal, 128.66 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders

Shared Folders: 1

Description

None

Install Curl and Git

```
###Install Curl###
$ sudo apt install curl

###Install Git###
$ sudo apt install git
```

Install Minikube

Minikube installer on Ubuntu 22.04

```
#!/bin/bash

# Script to run as root in Ubuntu 22.04 running on Oracle Virtual Box.
# This will bring up MiniKube running. For further Non-RT RIC installation, the related block must uncomment
and run.
# NOTE: Do 'sudo su' before running the script
# NOTE: Before start to installation, ensure that Curl, Git and Wget are ready to use!

###Log all cmds to stdout###
set -x

###Fetch and install Docker###
curl -fsSL https://get.docker.com -o get-docker.sh

sh get-docker.sh

###Install Conntrack###
apt-get install conntrack
```

```
###Install Socat###
apt install socat

###Install GO Lang###
wget https://storage.googleapis.com/golang/getgo/installer_linux
chmod +x ./installer_linux
./installer_linux
source ~/.bash_profile

###Install cri-dockerd###
git clone https://github.com/Mirantis/cri-dockerd.git
cd cri-dockerd
mkdir bin
go build -o bin/cri-dockerd
mkdir -p /usr/local/bin
install -o root -g root -m 0755 bin/cri-dockerd /usr/local/bin/cri-dockerd
cp -a packaging/systemd/* /etc/systemd/system
sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service
systemctl daemon-reload
systemctl enable cri-docker.service
systemctl enable --now cri-docker.socket
cd ..

###Install Crictl###
VERSION="v1.25.0"
wget https://github.com/kubernetes-sigs/cri-tools/releases/download/$VERSION/crictl-$VERSION-linux-amd64.tar.gz
sudo tar zxvf crictl-$VERSION-linux-amd64.tar.gz -C /usr/local/bin
rm -f crictl-$VERSION-linux-amd64.tar.gz

###Fetch and install Minikube###
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \
 && chmod +x minikube
mkdir -p /usr/local/bin/
install minikube /usr/local/bin/

###Fetch and 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
chmod +x ./kubectl
mv ./kubectl /usr/local/bin/kubectl
kubectl version --client #Just print the version

###Fetch Helm 3###
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh

###exit root###
exit

###Add user to docker group###
sudo usermod -aG docker $USER && newgrp docker


################################################################################
## Clone dep repo and deploy Non-RT RIC components                           ###
## Uncomment the following lines to install Non-RT RIC (can also be done later)###
################################################################################
#git clone "https://gerrit.o-ran-sc.org/r/it/dep"
#cd dep/bin
#sudo ./deploy-nonrtric -f ../nonrtric/RECIPE_EXAMPLE/example_recipe.yaml
```

Start Minikube

Start Minikube

```
###Start minikube###
minikube start

###Check status###
$ minikube status
$ kubectl get pods -n kube-system
```

```
thor@thor:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

thor@thor:~$ kubectl get pods -n kube-system
NAME                               READY   STATUS    RESTARTS       AGE
coredns-6d4b75cb6d-nt7fw           0/1     Running   3 (53s ago)    22h
etcd-minikube                      1/1     Running   3 (47m ago)    22h
kube-apiserver-minikube            1/1     Running   3 (47m ago)    22h
kube-controller-manager-minikube   1/1     Running   3 (47m ago)    22h
kube-proxy-9sjwh                   1/1     Running   3 (47m ago)    22h
kube-scheduler-minikube            1/1     Running   3 (47m ago)    22h
storage-provisioner                1/1     Running   10 (47m ago)   22h
```

Open the Kubernetes Dashboard

Open the Kubernetes Dashboard

```
###Start dashboard###
$ minikube dashboard
```