kops create cluster --name=telugugcp.xyz \

--state=s3://telugugcp.xyz --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=2 --node-size=t3.medium --master-size=t3.medium \

--master-volume-size 10 --node-volume-size 10 \

--dns-zone=telugugcp.xyz --kubernetes-version 1.19.9 --yes

ku create deploy nginx01 --image=sreeharshav/rollingupdate:v5 --replicas 3

ku expose deployment nginx01 --port=80 --target-port=80 --type=LoadBalancer

ubuntu@ip-192-168-1-100:/tmp$ kops upgrade cluster --state=s3://sreek8s

Using cluster from kubectl context: k8sb04.k8s.local

ITEM PROPERTY OLD NEW

Cluster KubernetesVersion 1.19.9 1.20.6

----------------------------------------------------------------

kops upgrade cluster --state=s3://telugugcp.xyz

kops upgrade cluster --state=s3://telugugcp.xyz --yes

kops update cluster --name telugugcp.xyz --yes *[sync state to Kops state bucket]*

kops update cluster --name telugugcp.xyz --yes --admin=87600h

kops rolling-update cluster --name telugugcp.xyz --yes *[start updating the nodes]*

# Upgrading Kubernetes Cluster with Kops:

<https://gist.github.com/stevenc81/e8d86d68a2aff69b7268938fa1f711fe>

<https://ahmet.im/blog/mastering-kubeconfig/>

**KOPS ROLLBACK to 1.19.9:**

kops edit cluster

Change the value of kubernetesVersion as below. Save and Exit.

kubernetesVersion: 1.19.9

kops update cluster k8sb04.k8s.local --state=s3://sreek8s --yes

kops rolling-update cluster --state=s3://sreek8s --yes

Merging in Linux:

KUBECONFIG=config:config2

KUBECONFIG=config:config2 kubectl config view --merge --flatten > out.txt

Export KUBECONFIG=out.txt

Merging with Powershell:

<https://www.mrjamiebowman.com/software-development/kubernetes/merge-kube-config-in-powershell/>

# merge both kube config files

$ENV:KUBECONFIG = "C:\Users\mrjamiebowman\.kube\config;C:\Users\mrjamiebowman\.kube\config-devbox"

# verify that the variable is set

$ENV:KUBECONFIG

# output to temp file

kubectl config view --flatten > config-merged

# verify that config-merged is correct

kubectl --kubeconfig=config-merged config get-clusters

# delete backup

rm config

# move merged file to config

mv config-merged config

# remove (optional)

rm config.bak

root@ip-192-168-1-100:~/.kube# KUBECONFIG=out.txt kubectx

k8sb04.k8s.local

Trainingk8s.xyz

root@ip-192-168-1-100:~/.kube# kubectl config view | grep -i cluster

clusters:

- cluster:

- cluster:

- cluster:

cluster: k8sb04.k8s.local

cluster: trainingk8s.xyz

cluster: twok8sb04.k8s.local

kubectl config use-context <cluster-name>

Or add lias as below

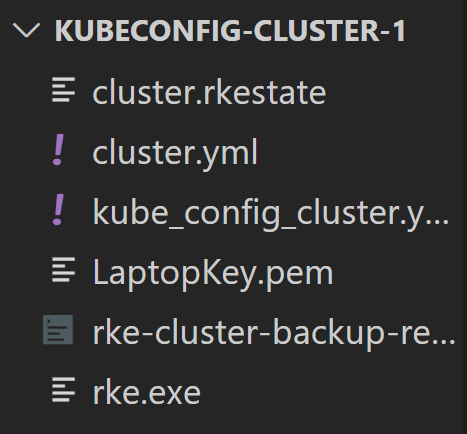
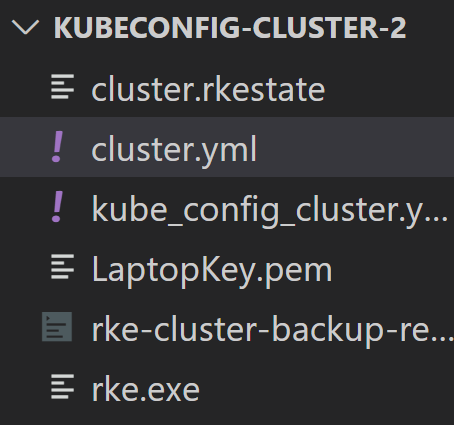
alias trainctx='kubectl config use-context trainingk8s.xyz'

alias trainctx='kubectl config use-context k8sb04.k8s.local'

alias trainctx='kubectl config use-context twok8sb04.k8s.local

Deploying multiple K8S Clusters using RKE:

1. Create two folder as production & development.
2. Copy RKE files as shown below in both folders.

1. Create two EC2 Ubuntu 20.04 t2.medium server and give following in the userdata.

#!/bin/bash

sudo curl https://get.docker.com | bash

sudo usermod -a -G docker ubuntu

sudo usermod -a -G root ubuntu

sudo systemctl daemon-reload

sudo systemctl restart docker

sudo reboot

sudo usermod -a -G docker ubuntu && sudo usermod -a -G root ubuntu && sudo systemctl daemon-reload && sudo systemctl restart docker && sudo reboot

5. Change the IP of the machines in cluster.yml file as per the machines you have created .

6. Open powershell and run rke.exe up in each folder to create two clusters.

$ENV:KUBECONFIG = "C:\SUREDELETELATER\K8SB10\Jobs-CronJobs\KUBECONFIG-CLUSTER-1\kube\_config\_cluster.yml"

kubectl cluster-info

$ENV:KUBECONFIG = "C:\SUREDELETELATER\K8SB10\Jobs-CronJobs\KUBECONFIG-CLUSTER-2\kube\_config\_cluster.yml"

kubectl cluster-info

Deploying multiple K8S Clusters using AWS Cloudshell:

1. Login to AWS Cloudshell
2. Create a folder called KOPS
3. Download kops & kubectl & kubectx to it and give execution permissions.
4. Execute the following command to create prod.k8s.local cluster.

./kops create cluster --name=prod.k8s.local \

--state=s3://sreek8s --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=1 --node-size=t3.medium --master-size=t3.medium \

--master-volume-size 10 --node-volume-size 10 --yes

1. Kubeconfig will be found in below location.

KUBECONFIG=/home/cloudshell-user/.kube/prod\_config ./kubectl get no

1. Rename the config to prod\_config
2. Execute the following command to create dev.k8s.local cluster.

./kops create cluster --name=dev.k8s.local \

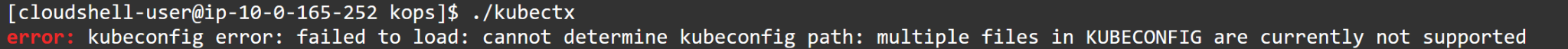
--state=s3://sreek8s --zones=us-east-1a,us-east-1b,us-east-1c \

--node-count=1 --node-size=t3.medium --master-size=t3.medium \

--master-volume-size 10 --node-volume-size 10 --yes

KUBECONFIG=/home/cloudshell-user/.kube/dev\_config ./kubectl get no

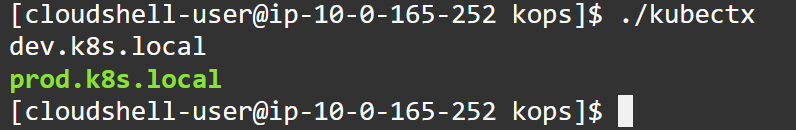
1. Rename the config to dev\_config
2. Download kubectx and extract it.
3. But kubectx wont work and give below error when used multiple files.



1. Now we need to merge both the kubeconfig files in to one.

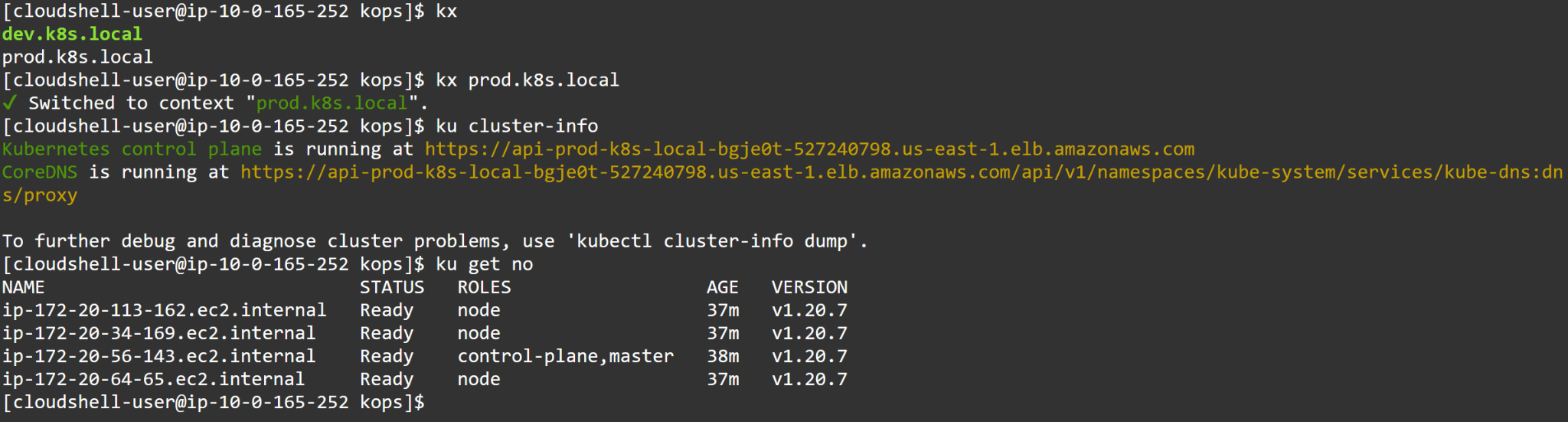
KUBECONFIG=/home/cloudshell-user/.kube/prod\_config:/home/cloudshell-user/.kube/dev\_config ./kubectl config view --merge --flatten > out.txt

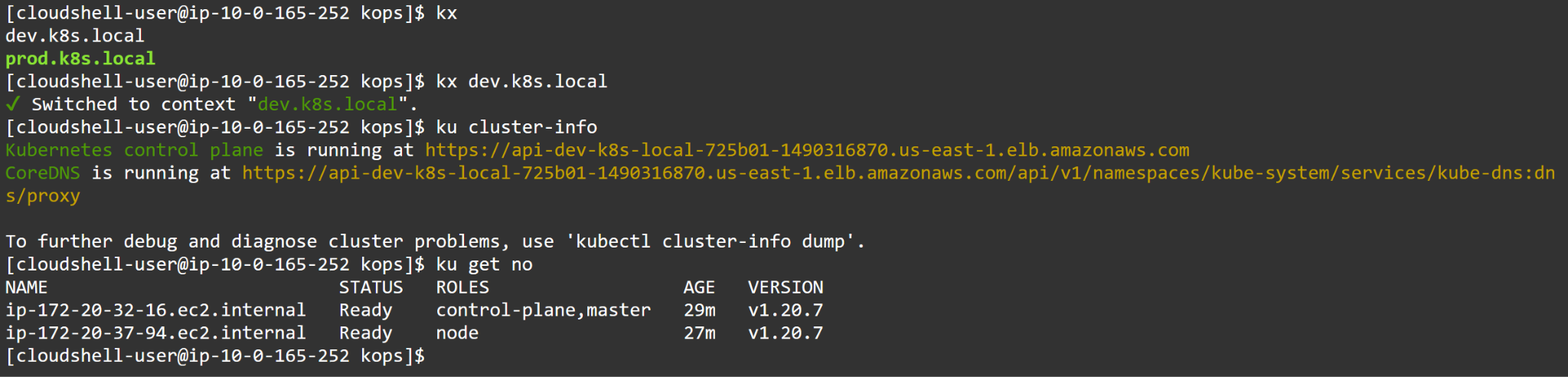
export KUBECONFIG=/home/cloudshell-user/kops/out.txt



alias ku=./kubectl

alias kx=./kubectx





kops upgrade cluster --state=s3://telugugcp.xyz

kops upgrade cluster --state=s3://telugugcp.xyz --yes

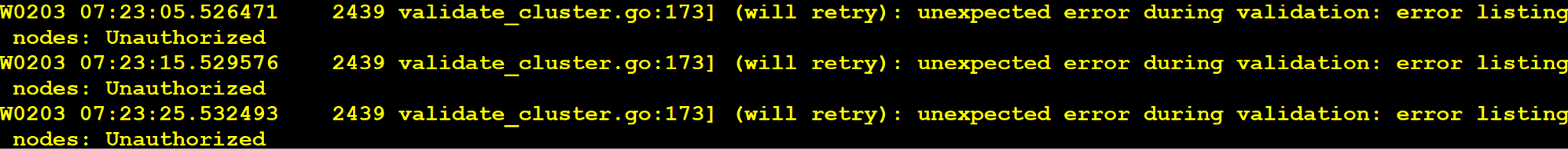
kops update cluster --name telugugcp.xyz --yes *[sync state to Kops state bucket]*

kops update cluster --name telugugcp.xyz --yes --admin=87600h

kops rolling-update cluster --name telugugcp.xyz --yes *[start updating the nodes]*

kops rolling-update cluster --master-interval=10m --node interval=10m --yes -v 10

You might experience errors as shown below.



The above error can be resolved by following command:

kops export kubecfg --admin

