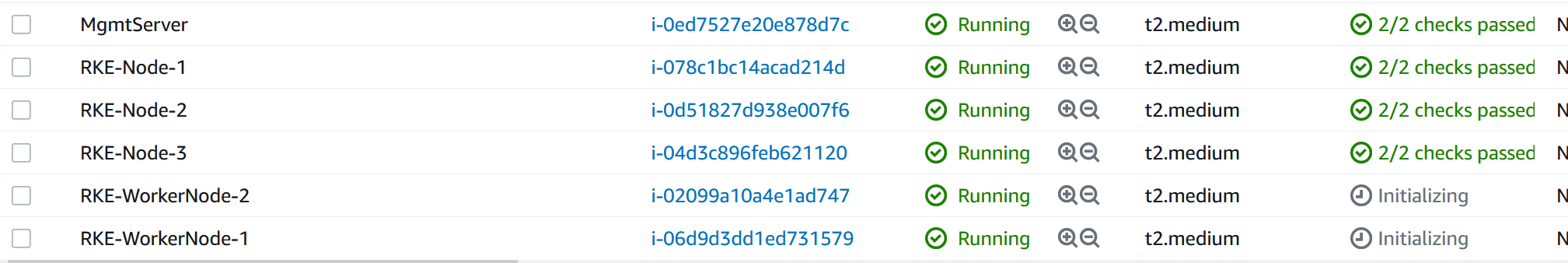
Deploying Highly Available Kubernetes Cluster With Rancher RKE:

1. Deploy 6 EC2 Ubuntu t2.medium instances and install docker using user data.

#!/bin/bash

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



We are going to use one machine as Mgmt Server for running RKE, 3 for Master Nodes and 2 for worker nodes.

In the Mgmt Node, download the RKE binary to /usr/local/bin

<https://github.com/rancher/rke/releases>

1. On all the Master and Worker nodes, enabled password authentication for root user and also add your public key in the .ssh/authorized\_keys
2. Check the SSH from Mgmt server to all Master and Worker nodes.
3. Create a directory naed as /myrkecluster and create a file called rke-cluster.yml. Paste the following in the YAML File.

nodes:

- address: 54.211.52.195

internal\_address: 172.16.1.5

user: root

role: [controlplane,etcd,worker]

hostname\_override: rke-node-1

- address: 3.94.82.218

internal\_address: 172.16.1.99

user: root

role: [controlplane,worker,etcd]

hostname\_override: rke-node-2

- address: 54.211.83.26

internal\_address: 172.16.1.128

user: root

role: [controlplane,worker,etcd]

hostname\_override: rke-node-3

- address: 3.86.243.77

internal\_address: 172.16.1.153

user: root

role: [worker]

hostname\_override: rke-workernode-1

- address: 3.84.51.83

internal\_address: 172.16.1.171

user: root

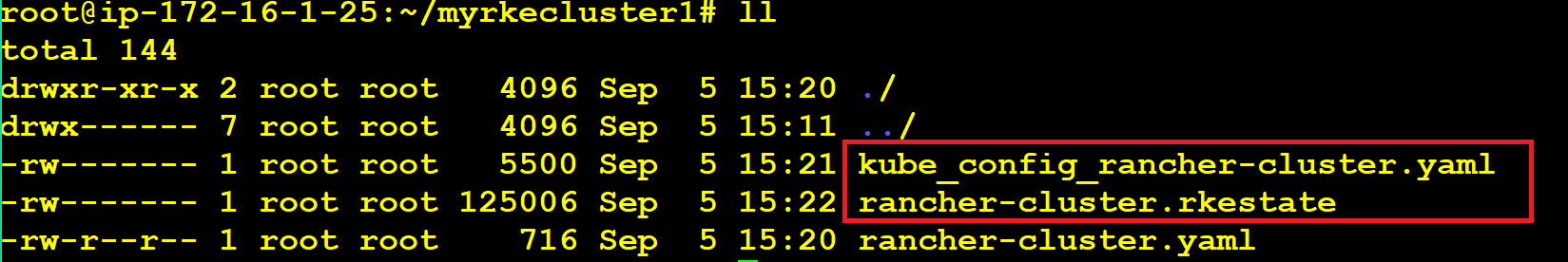
role: [worker]

hostname\_override: rke-workernode-2

1. Run the following commands to create the Kubernetes cluster.

rke up --config rancher-cluster.yaml

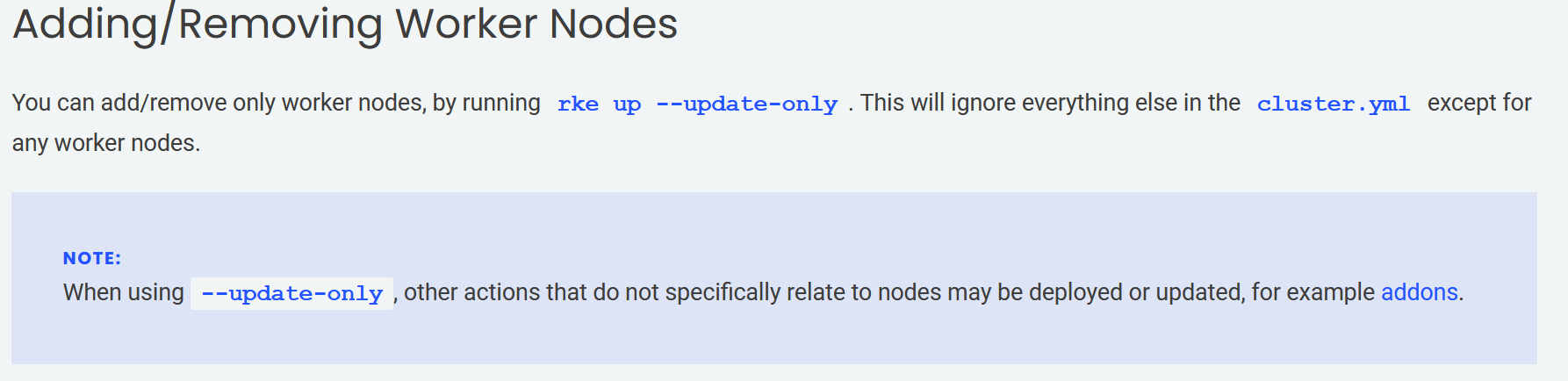
1. The RKE will create two new files as shown below.



One is a kubeconfig file and other is rke-state file which contain all the details of the cluster deployed.

Open Powershell and give the following command to kube config file:

$ENV:KUBECONFIG = "C:\SUREDELETELATER\K8SB10\RKE\kube\_config\_cluster.yml"



–Backup RKE Cluster–

rke etcd snapshot-save \

--config cluster.yml \

--name cluster-backup \

--s3 \

--access-key AKIA2QEFLENWI7WIV5FI \

--secret-key 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 \

--bucket-name k8sb10bucket \

--folder rkebackup \

--s3-endpoint s3.amazonaws.com

kubectl create deploy mytestdeploy01 --image=sreeharshav/testcontainer:v1 --replicas 10

kubectl expose deploy mytestdeploy01 --port=8000 --target-port=80 --type=NodePort

kubectl get svc

while true; do curl -sL http://ec2-54-211-52-195.compute-1.amazonaws.com:30542/ | grep -i 'IP A'; sleep 1; done

Create a new backup with apps are running:

rke etcd snapshot-save \

--config cluster.yml \

--name cluster-backup-with-apps \

--s3 \

--access-key AKIA2QEFLENWI7WIV5FI \

--secret-key 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 \

--bucket-name k8sb10bucket \

--folder rkebackup \

--s3-endpoint s3.amazonaws.com

Delete the one of the master node with etcd role and also delete the deployment and service.

Create a new machine and add to the cluster.yml and comment the old node. DONT PERFORM rke up.

To Restore run the following:

rke etcd snapshot-restore ^

--config cluster.yml ^

--name cluster-backup-with-apps ^

--s3 ^

--access-key AKIA2QEFLENWI7WIV5FI ^

--secret-key 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 ^

--bucket-name k8sb10bucket ^

--folder rkebackup ^

--s3-endpoint s3.amazonaws.com

The above command will restore both apps and etcd on the new node.

