**September 5, 2021**

**Installing KOPS**

Prerequisite:

Kubectl

**Install Kubectl Binary on Linux**

Need Latest version of kubectl on Linux -ubunto OS

Must have kubctl installed

Version of kubectl must be compatible with cluster version, within one minor version difference

1. Add kops user

sudo adduser kops

sudo echo "kops ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/kops

sudo su – kops

=========================================================================

1. Install Kubectl

B.i [Install and Set Up kubectl on Linux | Kubernetes](https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/)

1. Update the apt package index and install packages needed to use the Kubernetes apt repository:

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

1. Download the Google Cloud public signing key:

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg <https://packages.cloud.google.com/apt/doc/apt-key.gpg>

1. Add the Kubernetes apt repository:

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

1. Update apt package index with the new repository and install kubectl:

sudo apt-get update

sudo apt-get install -y kubectl

kubectl version --client

==OR USE USER DATA FOR (B) ABOVE==

B.ii

USERDATA

**#!/bin/bash**

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

curl -LO [https://dl.k8s.io/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl.sha256](https://dl.k8s.io/$(curl%20-L%20-s%20https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256)

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install -y kubectl

kubectl version --client

============================================================================

1. **Install Python 3 or Latest version**

sudo apt update -y

sudo apt install unzip wget -y

sudo apt install unzip python -y

1. **Install AWS CLI2**

**curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"**

**unzip awscliv2.zip**

**sudo ./aws/install -i** */usr/local/aws-cli* **-b** */usr/local/bin*

**his**

**NOTE: Restart Terminal if aws cli not found**

**Make s3 Bucket**

aws s3 mb s3://class24taa.local

aws s3 ls

vi .bashrc

export NAME=mybucket333.local

export KOPS\_STATE\_STORE=s3://class24taa.local

source .bashrc

**Add**

**Encrypt s3 bucket w ssl certificate – Free Encryption Service**

[**https://letsencrypt.org/?utm\_source=thenewstack&utm\_medium=website&utm\_campaign=platform**](https://letsencrypt.org/?utm_source=thenewstack&utm_medium=website&utm_campaign=platform)

[**https://thenewstack.io/how-to-set-up-your-s3-bucket-with-https-in-an-hour/**](https://thenewstack.io/how-to-set-up-your-s3-bucket-with-https-in-an-hour/)

**AWS Configure**

[Solve - the Config Profile could not be Found AWS CLI Error | bobbyhadz](https://bobbyhadz.com/blog/aws-cli-config-profile-could-not-be-found)

**[root@ip-172-31-80-21 ~]# ls ~/.aws**

**config credentials**

**[root@ip-172-31-80-21 ~]# ls ~/.aws/credentials**

**/root/.aws/credentials**

**[root@ip-172-31-80-21 ~]# cat ~/.aws/credentials**

**[default]**

**aws\_access\_key\_id = AKIAWXGYWJO3HSGECLVZ**

**aws\_secret\_access\_key = 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**

**[terese-aws-class24]**

**aws\_access\_key\_id = AKIAWXGYWJO3HSGECLVZ**

**aws\_secret\_access\_key = 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**

**[root@ip-172-31-80-21 ~]# cat ~/.aws/config**

**[default]**

**region = us-east-1**

**[profile terese-aws-class24]**

**region = us-east-1**

**output = json**

**[root@ip-172-31-80-21 ~]#**

1. **Install KOPS**

Install wget software

sudo apt install wget -y

**a.) Installs kops software in linux**

curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

chmod +x kops

sudo mv kops /usr/local/bin/kops

**b) Define and Add Environmental Variables**

**vi .bashrc**

Refresh and Apply the Env Variables

source .bashrc

**c) kops commands**

kops create

kops create -f <cluster spec>

kops create cluster

kops update cluster

kops rolling-update cluster

kops get clusters

kops delete cluster

kops toolbox template

kops version

**F) Create an IAM role from AWS Console or CLI with below Policies.**

AmazonEC2FullAccess = Giving this ec2 instance, Full Access to create instances, eg, Master and Wkr Nodes.

AmazonS3FullAccess = key-value-store

IAMFullAccess =To be able to assign IAM roles

AmazonVPCFullAccess = To place our resources in a “vpc” in which our instances will be running.

**5a) Create a Role in AWS:**

Services 🡺IAM🡺Roles🡺Create role🡺Assign the 4 Policies🡺<Name of Role>

Role Name – Class24-admin

\*\*\*Make sure to not assign AdministratorAccess – Can do everything, including access to the billing dashboard.\*\*\*

**5b) Attach IAM Role to instance in which KOPS software was installed:**

aws s3 ls

Error Message bc Kos instance isn’t attached to the role granting access to querry aws resources, or to make API calls yet.

Select Master/KOPS Server 🡪 Actions 🡪 Security/Instance Settings 🡪Modify IAM Role 🡪 Select the role created in 5a) 🡪<select class24 admin Role 🡪Save

1. Create sshkeys before creating cluster. To connect to our master server.

ssh-keygen **#did not pass any username or password**

1. **Secret Key Create for KOPS Cluster**

kops create secret --name venzatech.k8s.local sshpublickey admin -i ~/.ssh/id\_rsa.pub

#Must run the above command b4 creating cluster

1. Create KOPS Cluster

kops create cluster --zones ca-central-1b --networking weave --master-size t2.micro --master -count 1 --node-size t2.medium --node-count=2 ${NAME\_OF\_CLUSTER}

1. kops update cluster venzatech.k8s.local –yes

kops validate cluster --wait 10m

1. Authorize Access to Cluster with

**NOTE:** Unauthorized Error? Run

kops export kubecfg –admin

[amazon web services - kOps 1.19 reports error "Unauthorized" when interfacing with AWS cluster - Stack Overflow](https://stackoverflow.com/questions/66341494/kops-1-19-reports-error-unauthorized-when-interfacing-with-aws-cluster)

| **Number** | **Permission Type** | **Symbol** |
| --- | --- | --- |
| 0 | No Permission | — |
| 1 | Execute | –x |
| 2 | Write | -w- |
| 3 | Execute + Write | -wx |
| 4 | Read | r– |
| 5 | Read + Execute | r-x |
| 6 | Read +Write | rw- |
| 7 | Read + Write +Execute | rwx |

Kops version

Version 1.21.1 (git-ffabc3bf682bfc25ffdce99e54005345df94b467)

Kubectl version

**DEPLOYMENTS**

K8S-ingress/deployments/springsvc.yml

kops@Kops:~$ git clone https://github.com/Venza-Tech/k8s-ingress

Cloning into 'k8s-ingress'...

remote: Enumerating objects: 71, done.

remote: Counting objects: 100% (71/71), done.

remote: Compressing objects: 100% (68/68), done.

remote: Total 71 (delta 25), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (71/71), 22.77 KiB | 1.52 MiB/s, done.

kops@Kops:~$ ls

aws awscliv2.zip k8s-ingress kubectl kubectl.sha256

kops@Kops:~$ cd k8s-ingress/

kops@Kops:~/k8s-ingress$ ls

README.md deployments kubernetes-ingress-master

kops@Kops:~/k8s-ingress$ cd deployments/

kops@Kops:~/k8s-ingress/deployments$ ls

common daemon-set deployment mongodb.yml service springsvc.yml

kops@Kops:~/k8s-ingress/deployments$ kubectl apply -f springsvc.yml

deployment.apps/usermgt created

service/springapp created

persistentvolumeclaim/mongodbpvc created

replicaset.apps/mongodbrs created

service/mongo created

configmap/mongo-configmap created

secret/mongo-db-password created

kops@Kops:~/k8s-ingress/deployments$