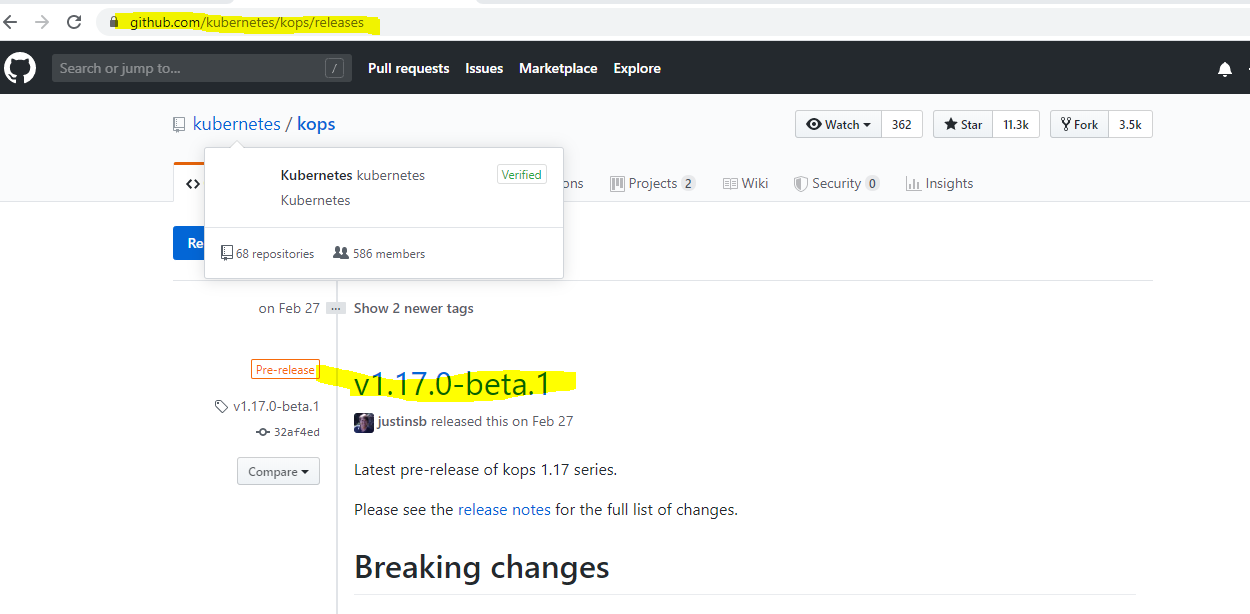
Topic-1:



Topic-2: Prepare AWS environment for Kops

The latest Kops release is below:



**Commands to install kops on linux machine:**

1. curl -LO 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
2. chmod +x kops-linux-amd64
3. sudo mv kops-linux-amd64 /usr/local/bin/kops

**Installing Dependencies(KubeCtl)**

1. curl -Lo kubectl https://storage.googleapis.com/kubernetes-release/release/$(curl -s <https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl>

2. chmod +x ./kubectl

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

**Download python PIP on linux:**

sudo apt-get install python-pip

If in case of error execute below:

Sudo apt-get update

**How to check python version:**

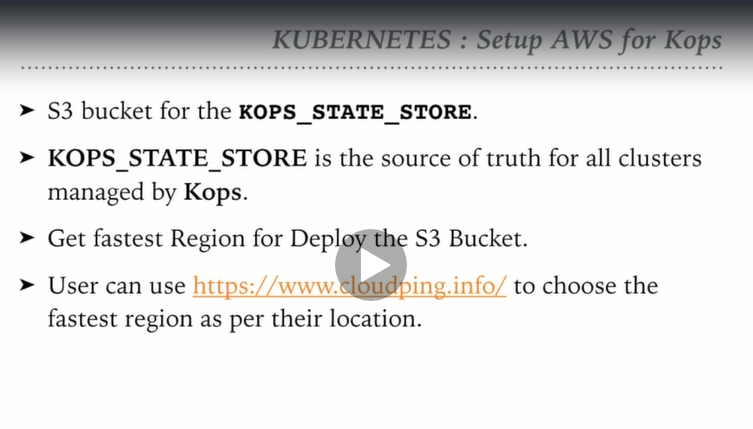
Python - -version

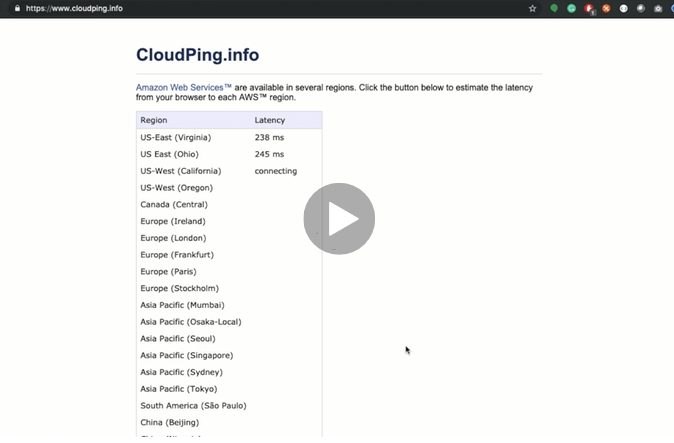
**Install AWS CLI:**

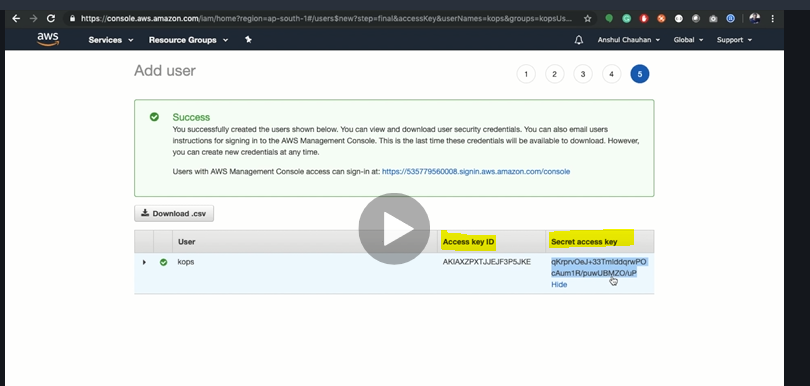
Pip install awscli

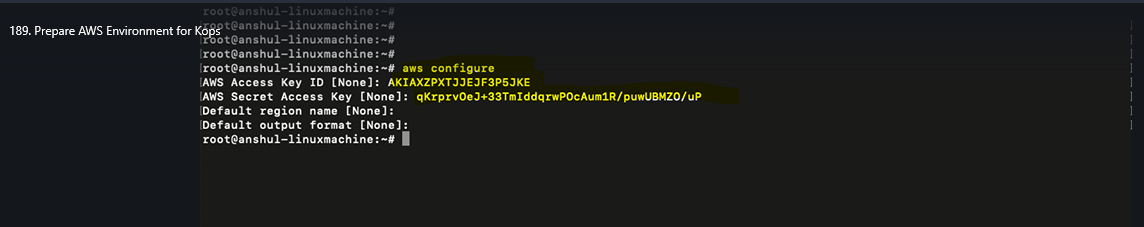
In case of error

sudo apt-get install awscli









Set-1:

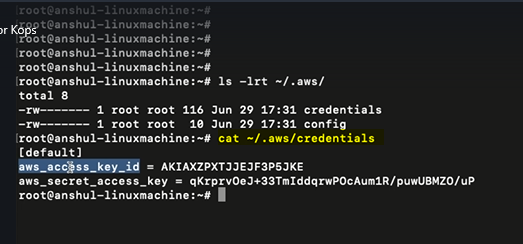
Access\_Id: AKIAR4MTZ3CSN6VWDFN6

Secret\_key: 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

Set-2:

AKIAR4MTZ3CSELFCM4AR

eGyMm/xY4kbcOjEaWHqudkU8TVey9nt6f2JVKaxf



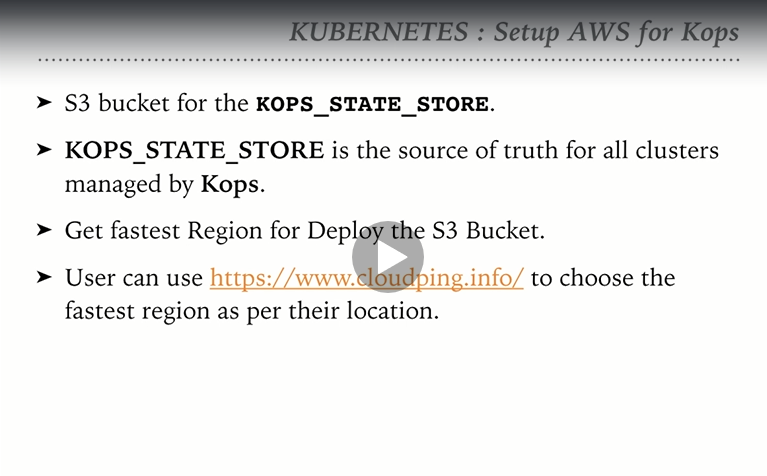
To verify aws is configured on linux machine:

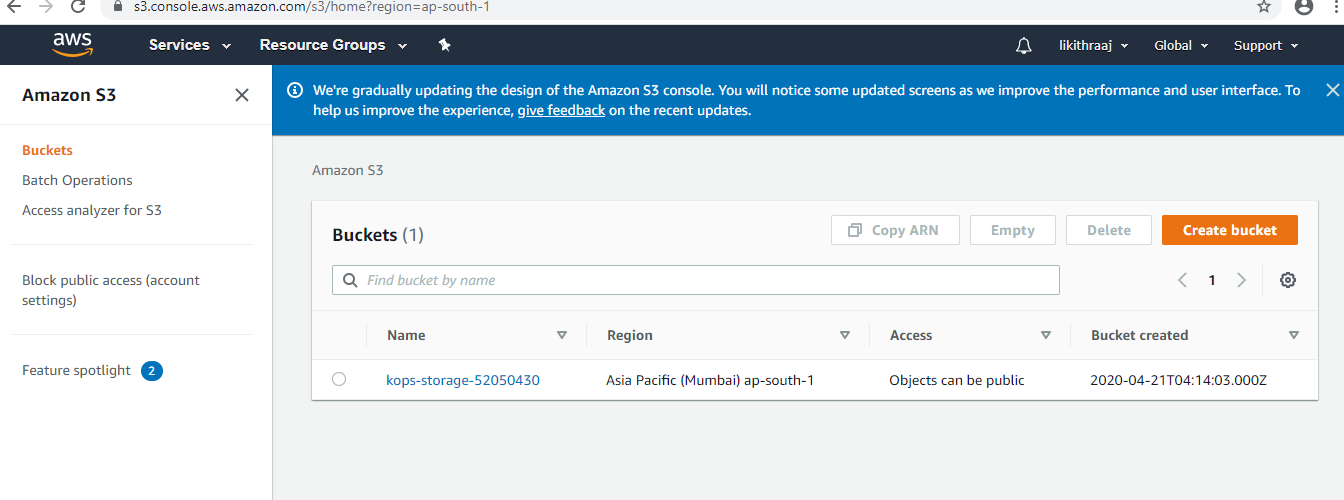
ls -lrt ~/.aws/

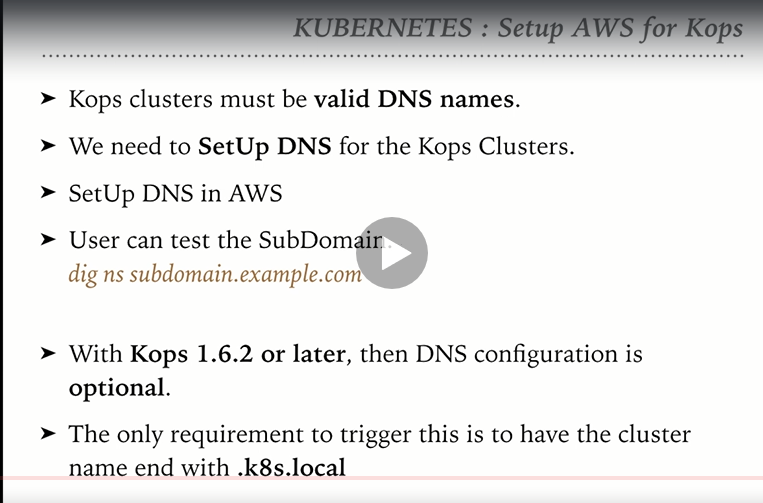
cat ~/.aws/credentials

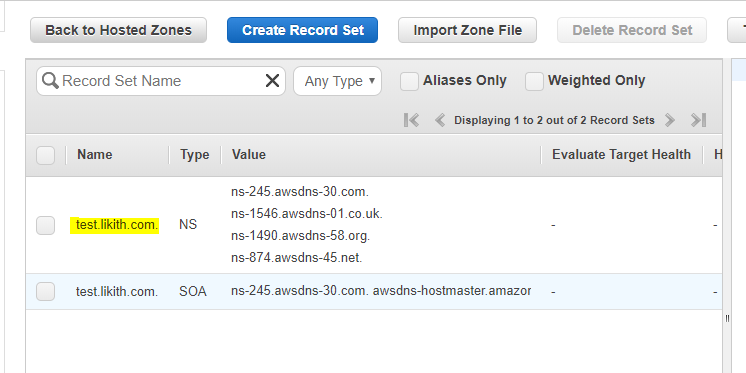
**Create a S3 bucket:**

**Simple Storage Service (S3)**

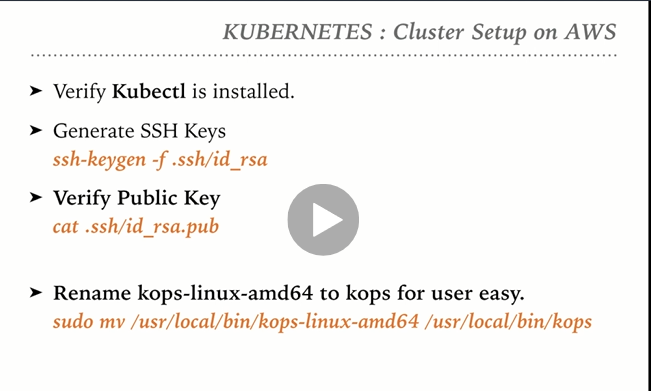


S3 bucket is created 





**Topic-3 : Kubernetes setup on AWS cloud**



Type kubectl to verify if it is installed

**To generate ssh key:**

ssh-keygen -f .ssh/id\_rsa

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDGSr2pQvuC4s9PEYsIqm/DkW4IODCDADk4wacjNsg5LclY/Tqh7dDH0YB21KC43PGkmJJ/TX/6Zd6mYwhhHlWcgeIWS+FaNCgnTtZKlWdaI1btv37Abxifb3Nn899psGPmcd+3Mxw9EE/X/8AXDTAAPxdOfi1Ve4ZFoAoTXeoXLmdiQNWORWENfbKx7cS3nUHFq0w5CN4sYdETQ/XWNMkoqSXZgewGc+Hbu4bGS/J4fuw3xdWbskkWbOwPnS9+uCMhw+0P2rFJgf4sk6j14An8AM2BJ4D/xs3LsWyLA8sC6j+y5Dvgit8bukydU8caAM9HenytOuOMHa1iUJNFxqOd ubuntu@ip-172-31-7-143

**To verify the ssh key:**

cat .ssh/id\_rsa.pub

If we are using higher version of kops then /usr/local/bin/kops-linux-amd64 doesn’t exist

To find the aws availability zones: Type ‘aws availability zones’ in google and the list will be rendered

<https://stelligent.com/2014/05/02/list-all-the-availability-zones/>

How to create a Kubernetes cluster:

kops create cluster --yes --state=s3://kops-storage-52050430 --zones=ap-south-1a,ap-southeast-1b,ap-southeast-2c --node-count=2 --node-size=t2.micro --master-size=t2.micro --name=test.k8s.local

How to validate cluster:

Kops validate cluster

How to check kops version

kops version

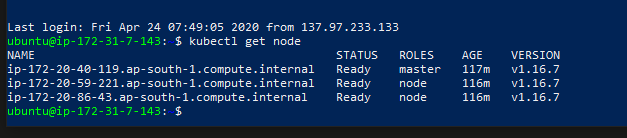
How to check cluster is ready:

Kops validate cluster - - state-s3://kops-storage-52050430

If there are any validation failed error then check if the EC2 instances are in running state then run the same command again.

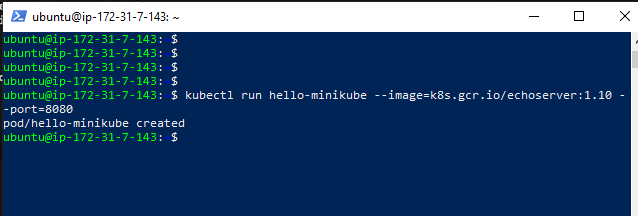
How to check cluster nodes status

Kubectl get node



Let’s create a Kubernetes Deployment using an existing image named echoserver, which is a simple HTTP server and expose it on port 8080 using --port.

kubectl run hello-minikube --image=k8s.gcr.io/echoserver:1.10 --port=8080



7. In order to access the hello-minikube service, we must first expose the deployment to an external IP via the command:

kubectl expose deployment hello-minikube --type=NodePort

**Topic-4: Build and push Docker custom image**

1. **To build an image from docker file:**

docker build -t <imagename>:<tag> <path>

docker build -t myimage:latest .

1. **To run the container**

Docker container run -d –-name <nameoftheimage> - p <hostport>:<containerport> <image>

Docker container run -d –name mynginx -p 8081: 80 nginx:alpine

1. **To tag an image**

Docker tag <imageid> <hub.dockerId>/<ImageName>

1. **To login**

Docker login –username=<hub.dockerid>

1. **To push the image to hub.docker**

Docker push <hub.dockerId>/<ImageName>

**Topic-5: Run first custom image on local Kubernetes**



**Topic-6**: **Run first custom image on AWS Kubernetes**

1. Make sure kops is installed along with it dependencies
2. If you want you can create your own domain otherwise you can continue with test.k8s.local
3. After a domain is created, there will be a listing in the Hosted zones
4. 
5. Create a cluster.
6. Validate if the cluster is up and running with the master and nodes
7. Create deployment
8. Describe deployment
9. Make the service available in the internet by creating load balancer
10. Verify the service is running
11. GO to Services>EC2 Instances>LoadBalancer
12. Copy the DNS name of the Loadbalancer and paste it in the browser. You should be able to see the service/application which was part of the docker Image.
13. Delete service
14. Delete deployment
15. Delete cluster