Create a Kubernetes Cluster on AWS

Installing Kubernetes on AWS

Amazon Web Services (AWS) recently introduced a managed Kubernetes service called [EKS](https://aws.amazon.com/eks/). Nevertheless, it’s still under preview mode. Therefore, at the moment Kubernetes can be installed on AWS as explained in the Kubernetes documentation either using [conjure-up](https://kubernetes.io/docs/getting-started-guides/ubuntu/), [Kubernetes Operations](https://github.com/kubernetes/kops) (kops), [CoreOS Tectonic](https://coreos.com/tectonic/) or [kube-aws](https://github.com/kubernetes-incubator/kube-aws). Out of those options I found kops extremely easier to use and its nicely designed for customizing the installation, executing upgrades and managing the Kubernetes clusters over time. In this article I will explain how to use Kubernetes Operations tool to install a Kubernetes Cluster on AWS in few minutes.

Steps to Follow

1. First we need an AWS account and access keys to start with. Login to your AWS console and generate access keys for your user by navigating to Users/Security credentials page.

2. Install AWS CLI by following its [official installation guide](https://docs.aws.amazon.com/cli/latest/userguide/installing.html):

# OSX using Homebrew  
brew install awscli# Linux  
pip install awscli --upgrade --user# awscli version: 1.6.5

3. Install kops by following its [official installation guide](https://github.com/kubernetes/kops#installing):

# OSX using Homebrew  
brew install kops# Linux  
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  
chmod +x kops-linux-amd64  
sudo mv kops-linux-amd64 /usr/local/bin/kops# kops version: 1.9.0

4. Create a new IAM user or use an existing IAM user and grant following permissions:

AmazonEC2FullAccess  
AmazonRoute53FullAccess  
AmazonS3FullAccess  
AmazonVPCFullAccess

5. Configure the AWS CLI by providing the Access Key, Secret Access Key and the AWS region that you want the Kubernetes cluster to be installed:

aws configureAWS Access Key ID [None]: AccessKeyValue  
AWS Secret Access Key [None]: SecretAccessKeyValue  
Default region name [None]: us-east-1  
Default output format [None]:

6. Create an AWS S3 bucket for kops to persist its state:

bucket\_name=imesh-kops-state-store  
aws s3api create-bucket \  
--bucket ${bucket\_name} \  
--region us-east-1

7. Enable versioning for the above S3 bucket:

aws s3api put-bucket-versioning --bucket ${bucket\_name} --versioning-configuration Status=Enabled

8. Provide a name for the Kubernetes cluster and set the S3 bucket URL in the following environment variables:

export KOPS\_CLUSTER\_NAME=imesh.k8s.local  
export KOPS\_STATE\_STORE=s3://${bucket\_name}

Add above code block can be added to the ~/.bash\_profile or ~/.profile file depending on the operating system to make them available on all terminal environments.

9. Create a Kubernetes cluster definition using kops by providing the required node count, node size, and AWS zones. The node size or rather the [EC2 instance type](https://aws.amazon.com/ec2/instance-types/) would need to be decided according to the workload that you are planning to run on the Kubernetes cluster:

kops create cluster \  
--node-count=2 \  
--node-size=t2.medium \  
--zones=us-east-1a \  
--name=${KOPS\_CLUSTER\_NAME}

If you are seeing any authentication issues, try to set the following environment variables to let kops directly read EC2 credentials without using the AWS CLI:

export AWS\_ACCESS\_KEY=AccessKeyValue  
export AWS\_SECRET\_KEY=SecretAccessKeyValue

If needed execute the kops create cluster help command to find additional parameters:

kops create cluster --help

10. Review the Kubernetes cluster definition by executing the below command:

kops edit cluster --name ${KOPS\_CLUSTER\_NAME}

11. Now, let’s create the Kubernetes cluster on AWS by executing kops update command:

kops update cluster --name ${KOPS\_CLUSTER\_NAME} --yes

12. Above command may take some time to create the required infrastructure resources on AWS. Execute the validate command to check its status and wait until the cluster becomes ready:

kops validate clusterUsing cluster from kubectl context: imesh.k8s.localValidating cluster imesh.k8s.localINSTANCE GROUPS  
NAME ROLE MACHINETYPE MIN MAX SUBNETS  
master-us-east-1a Master m3.medium 1 1 us-east-1a  
nodes Node m4.xlarge 2 2 us-east-1aNODE STATUS  
NAME ROLE READY  
ip-172-20-48-50.ec2.internal node True  
ip-172-20-50-191.ec2.internal node True  
ip-172-20-55-27.ec2.internal master TrueYour cluster imesh.k8s.local is ready

Once the above process completes, kops will configure the Kubernetes CLI (kubectl) with Kubernetes cluster API endpoint and user credentials.

13. Now, you may need to deploy the Kubernetes dashboard to access the cluster via its web based user interface:

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml>

14. Execute the below command to find the admin user’s password:

kops get secrets kube --type secret -oplaintext

15. Execute the below command to find the Kubernetes master hostname:

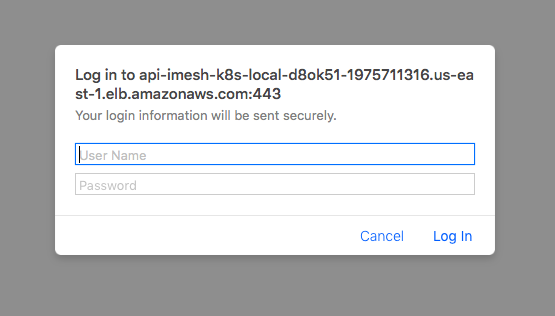
kubectl cluster-infoKubernetes master is running at [https://api-imesh-k8s-local-<dynamic-id>.us-east-1.elb.amazonaws.com](https://api-imesh-k8s-local-d8ok51-1975711316.us-east-1.elb.amazonaws.com/)  
KubeDNS is running at [https://api-imesh-k8s-local-<dynamic-id>.us-east-1.elb.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy](https://api-imesh-k8s-local-d8ok51-1975711316.us-east-1.elb.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy)

16. Access the Kubernetes dashboard using the following URL:

https://<kubernetes-master-hostname>/ui

Provide the username as admin and the password obtained above at the step 14 on the browser’s login page:

https://miro.medium.com/max/60/1*o9jn8mu6en8KPOUWLAtmZg.png?q=20

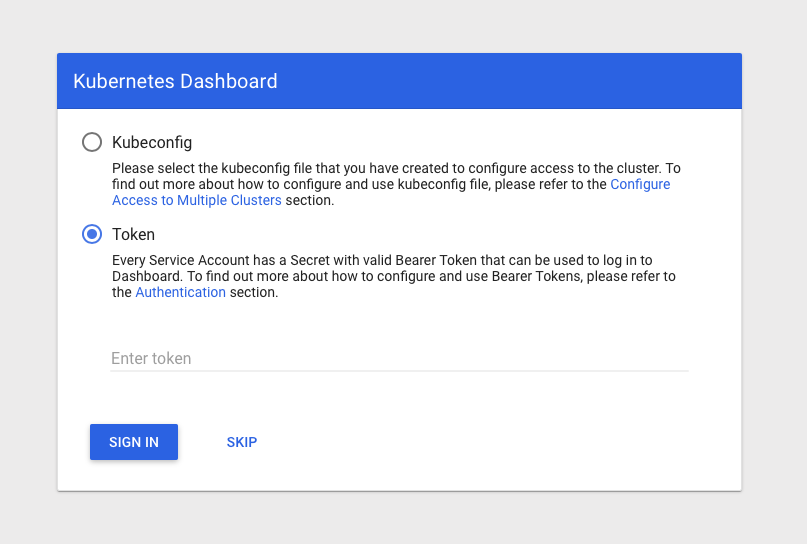


Execute the below command to find the admin service account token. Note the secret name used here is different from the previous one:

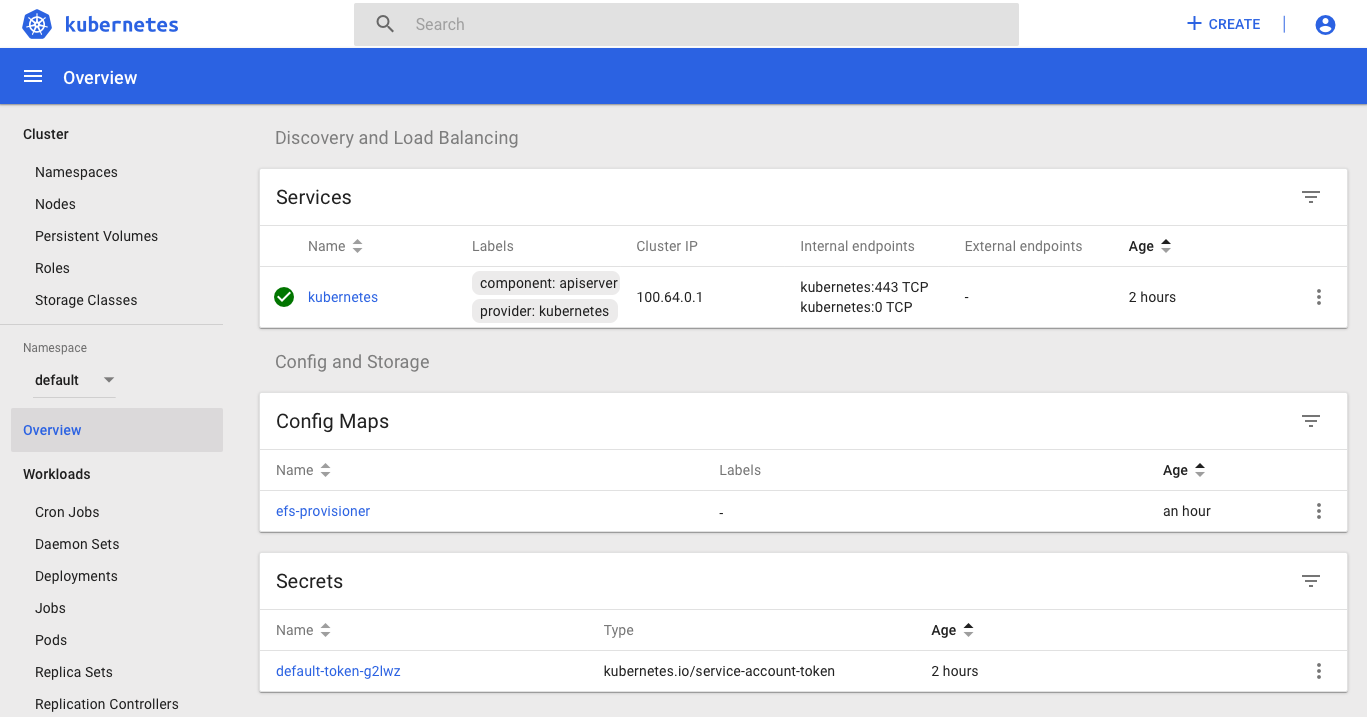
kops get secrets admin --type secret -oplaintext

Provide the above service account token on the service token request page:

https://miro.medium.com/max/60/1*cpWFVhTdANYLg9VFDJ05sg.png?q=20



https://miro.medium.com/max/60/1*e0kJQdAijcNd0amp-4iWog.png?q=20



https://miro.medium.com/max/60/1*ju6XcUxkoQuRuJYmllFspw.png?q=20

