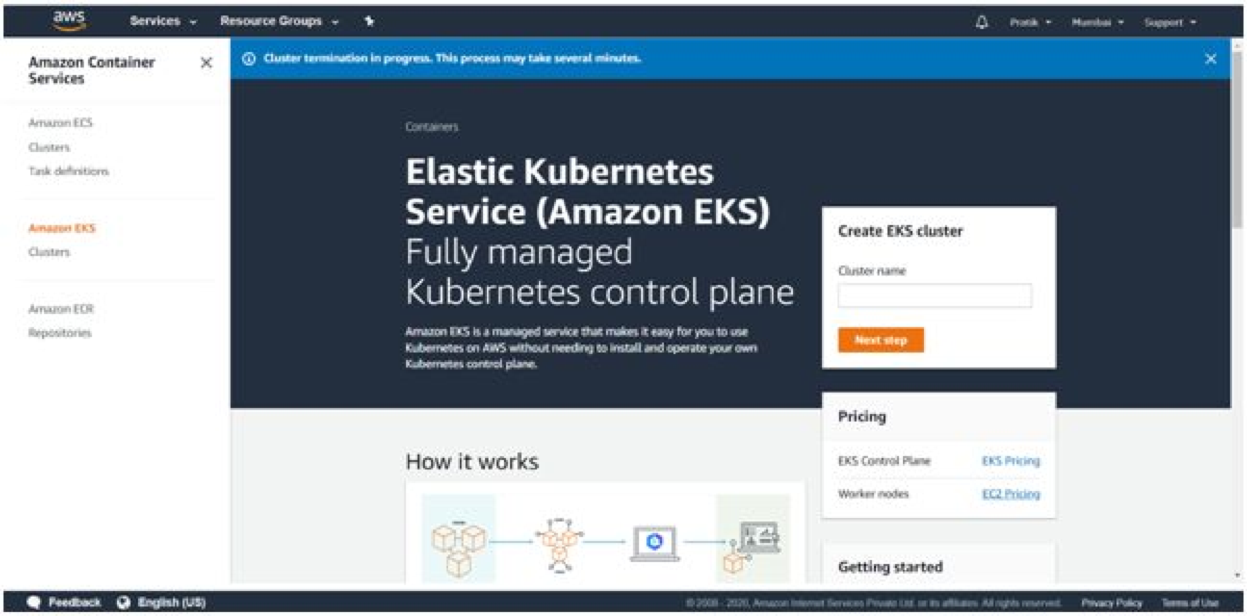
EKS TASK

Launching a cluster on eks using cluster file:

```
eksctl create cluster -f cluster1.yml
```

```
apiVersion: eksctl.io/v1alpha5 kind: ClusterConfig


metadata:
  name: mycluster
  region: ap-south-1


nodeGroups:
  - name: ng1
            desiredCapacity: 2
            instanceType: t2.micro
            ssh:
            publicKeyName: mykey11.pem
  - name: ng2
            desiredCapacity: 1
            instanceType: t2.small
            ssh:
            publicKeyName: mykey11.pem
  --cluster=mycluster'
```



Cluster termination in progress. This process may take several minutes.

Elastic Kubernetes Service (Amazon EKS)
Fully managed Kubernetes control plane

Amazon EKS is a managed service that makes it easy for you to use Kubernetes on AWS without needing to install and operate your own Kubernetes control plane.

Create EKS cluster

Cluster name

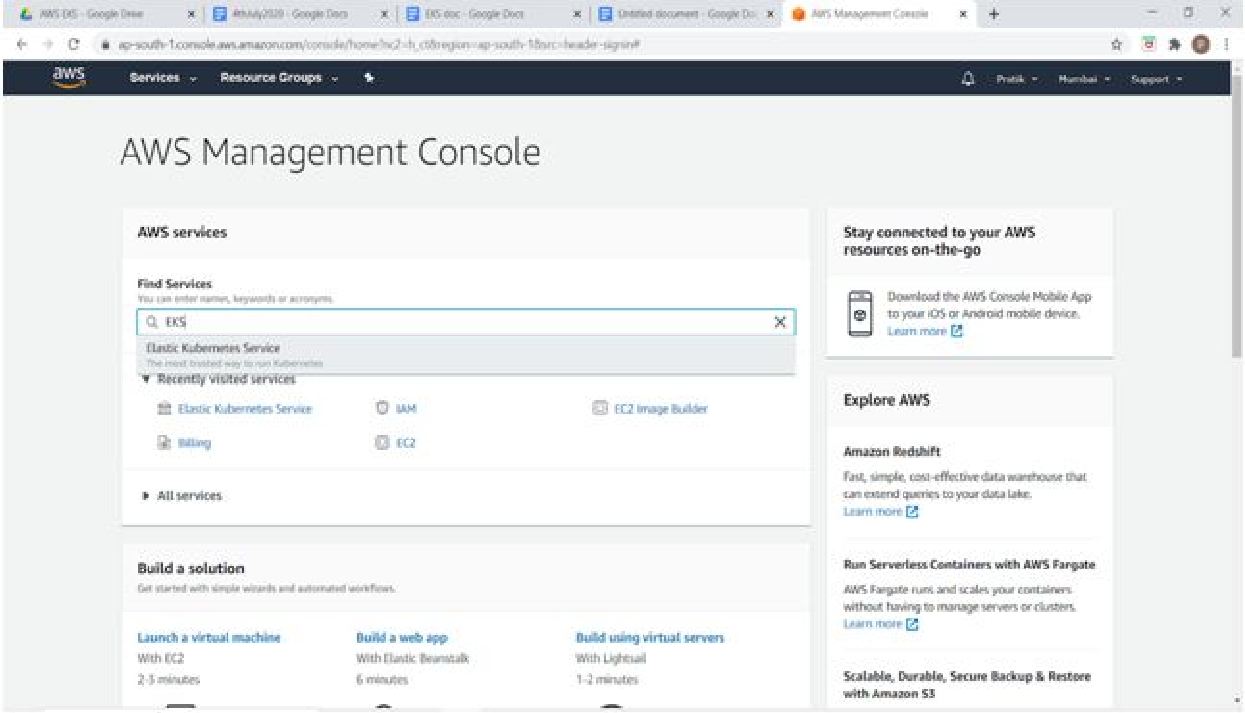
Pricing

EKS Control Plane	EKS Pricing
Worker nodes	EC2 Pricing

How it works

Getting started

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.



ap-south-1.console.aws.amazon.com/console/home?nc2=h_ct&region=ap-south-1&src=header-signin#

AWS Management Console

AWS services

Find Services
You can enter names, keywords or acronyms.

EKS

Elastic Kubernetes Service
The most trusted way to run Kubernetes

Build a solution
Get started with simple wizards and automated workflows.

Launch a virtual machine	Build a web app	Build using virtual servers
With EC2	With Elastic Beanstalk	With Lightsail
2-3 minutes	6 minutes	1-2 minutes

Stay connected to your AWS resources on-the-go

Download the AWS Console Mobile App to your iOS or Android mobile device.

Explore AWS

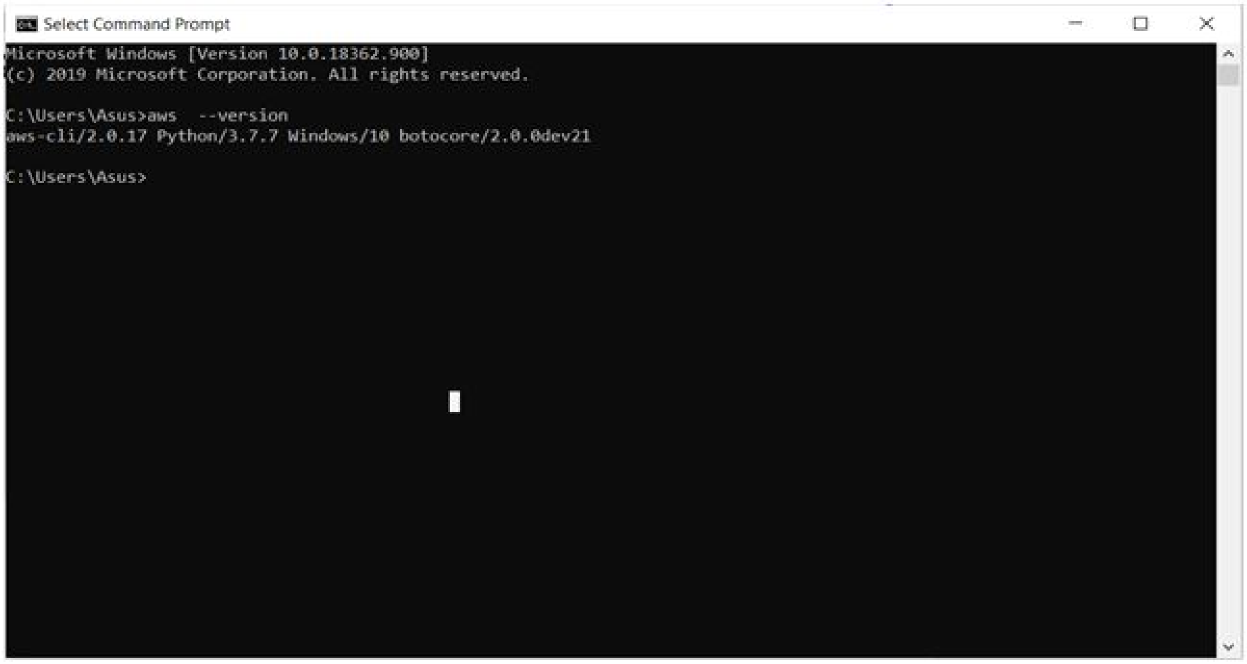
Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake.

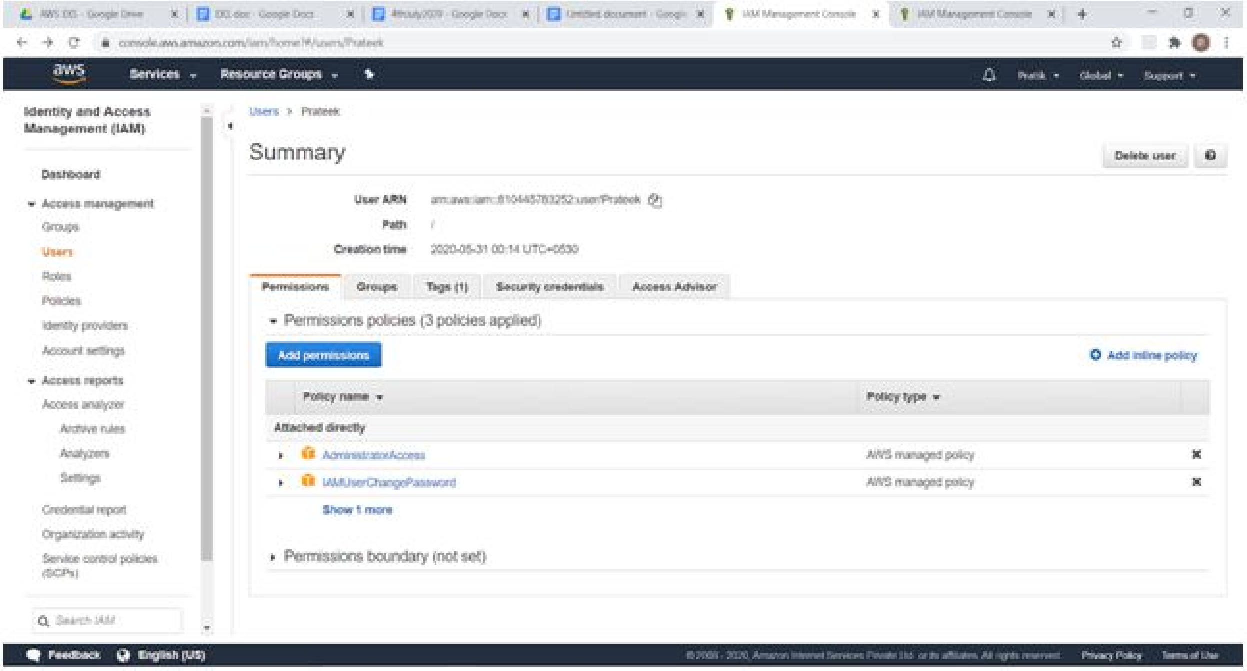
Run Serverless Containers with AWS Fargate

AWS Fargate runs and scales your containers without having to manage servers or clusters.

Scalable, Durable, Secure Backup & Restore with Amazon S3



```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>aws  --version
aws-cli/2.0.17 Python/3.7.7 Windows/10 botocore/2.0.0dev21

C:\Users\Asus>
```



Summary

User ARN arn:aws:iam::810445783252:user/Prateek

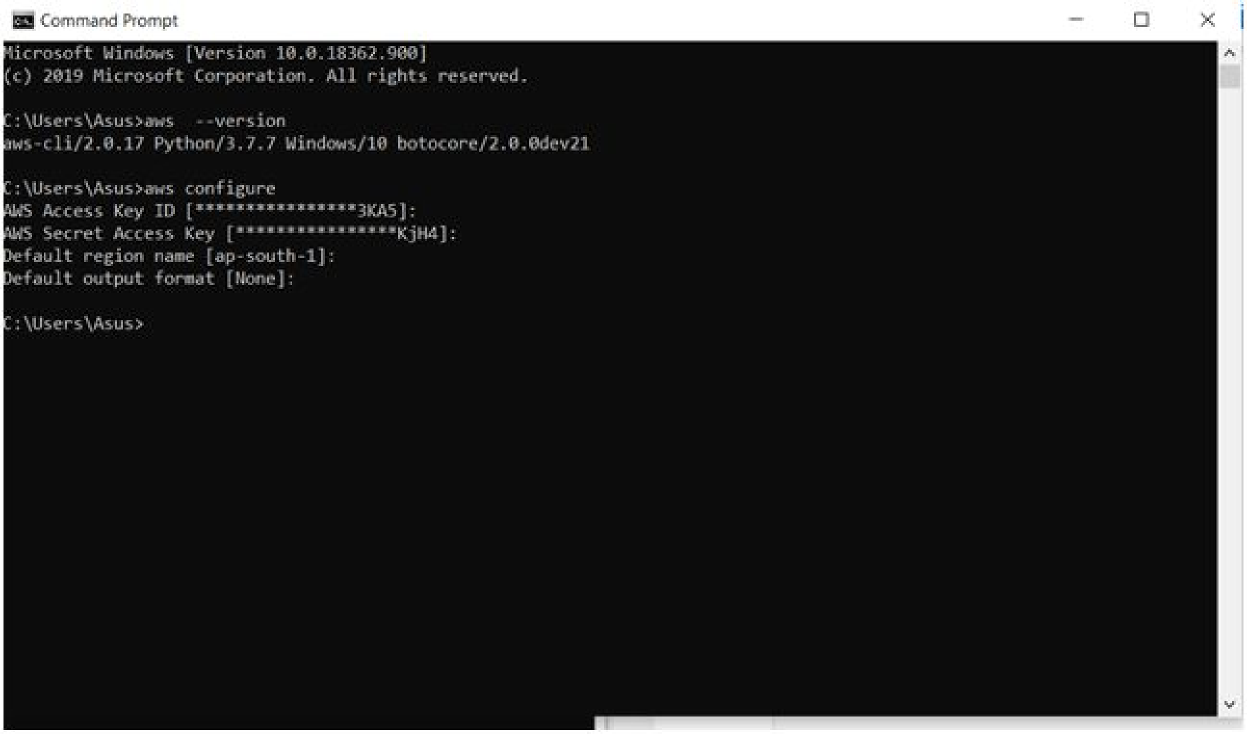
Path /

Creation time 2020-05-31 00:14 UTC+0530

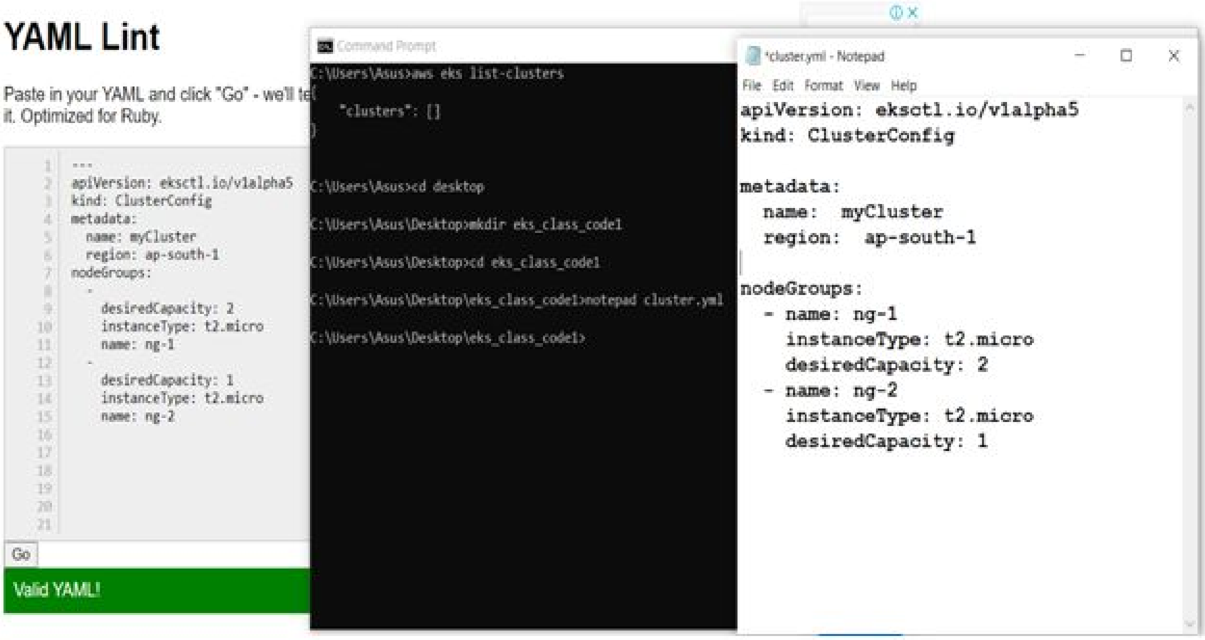
Permissions policies (3 policies applied)

Policy name	Policy type
Attached directly	
AdministratorAccess	AWS managed policy
IAMUserChangePassword	AWS managed policy

Permissions boundary (not set)



```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Asus>aws  --version
aws-cli/2.0.17 Python/3.7.7 Windows/10 botocore/2.0.0dev21

C:\Users\Asus>aws configure
AWS Access Key ID [****************3KA5]:
AWS Secret Access Key [****************KjH4]:
Default region name [ap-south-1]:
Default output format [None]:

C:\Users\Asus>
```



YAML Lint

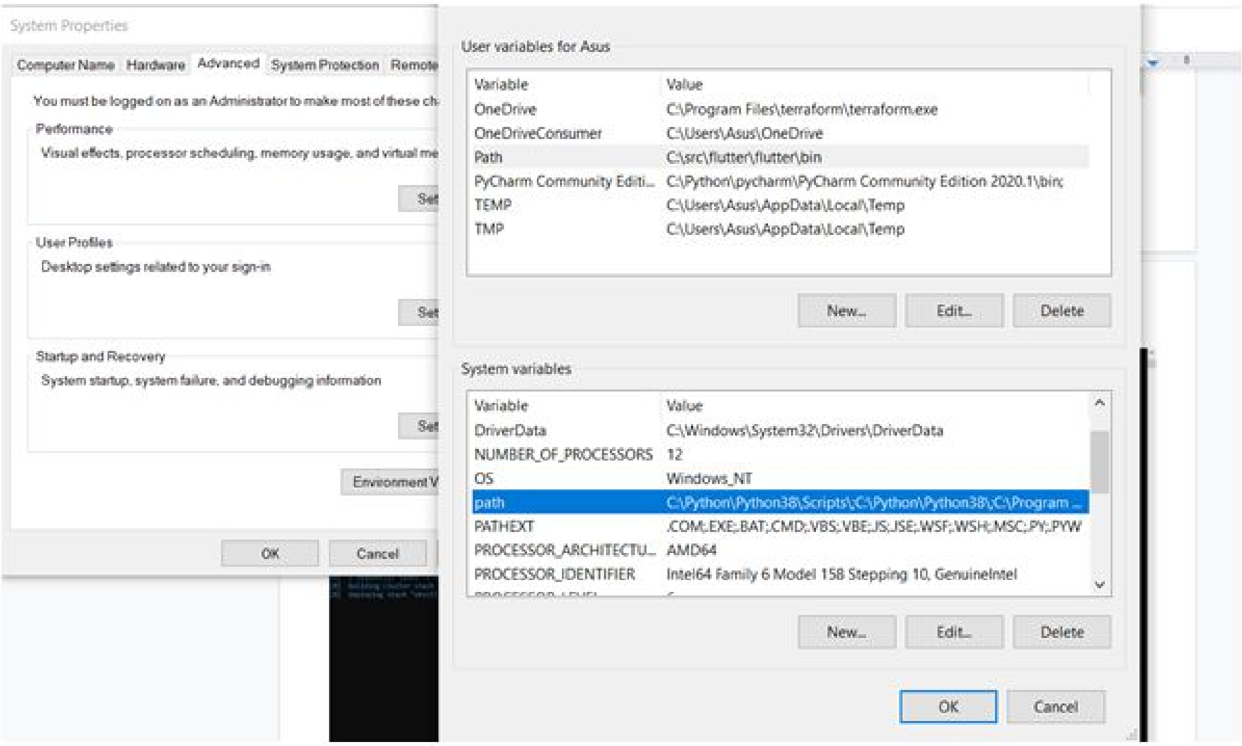
Paste in your YAML and click "Go" - we'll te
it. Optimized for Ruby.

```
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: myCluster
  region: ap-south-1
nodeGroups:
  -
    desiredCapacity: 2
    instanceType: t2.micro
    name: ng-1
  -
    desiredCapacity: 1
    instanceType: t2.micro
    name: ng-2
```

Valid YAML!

```
C:\Users\Asus>aws eks list-clusters
{
    "clusters": []
}


C:\Users\Asus>cd desktop

C:\Users\Asus\Desktop>mkdir eks_class_code1

C:\Users\Asus\Desktop>cd eks_class_code1

C:\Users\Asus\Desktop\eks_class_code1>notepad cluster.yml

C:\Users\Asus\Desktop\eks_class_code1>
```

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name:  myCluster
  region:  ap-south-1

nodeGroups:
  - name: ng-1
    instanceType: t2.micro
    desiredCapacity: 2
  - name: ng-2
    instanceType: t2.micro
    desiredCapacity: 1
```



System Properties

Computer Name | Hardware | Advanced | System Protection | Remote

You must be logged on as an Administrator to make most of these ch

Performance

Visual effects, processor scheduling, memory usage, and virtual me

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

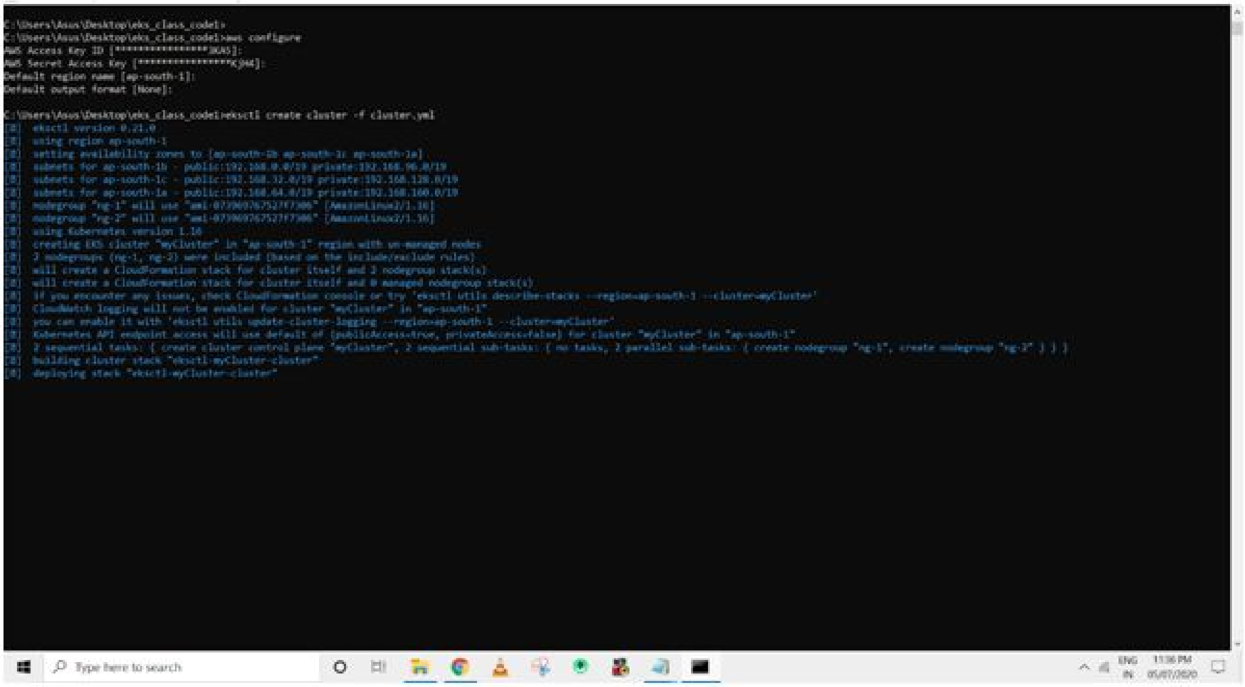
System startup, system failure, and debugging information

User variables for Asus

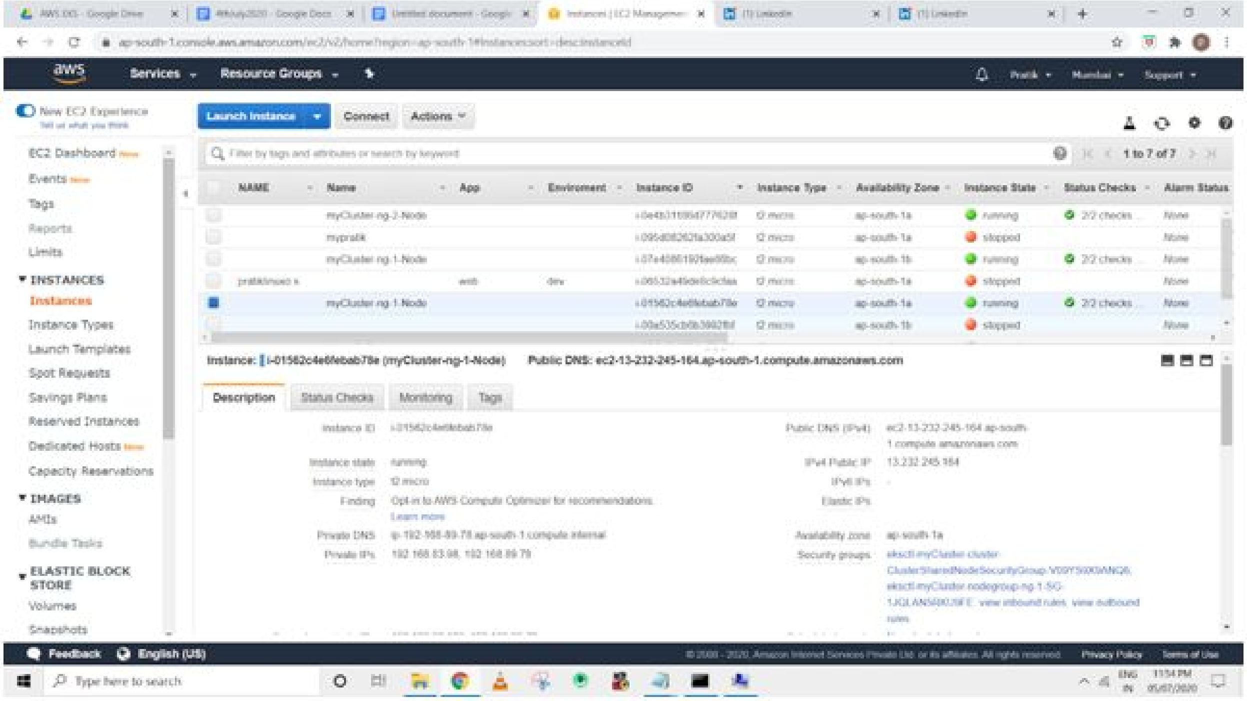
Variable	Value
OneDrive	C:\Program Files\terraform\terraform.exe
OneDriveConsumer	C:\Users\Asus\OneDrive
Path	C:\src\flutter\flutter\bin
PyCharm Community Editi...	C:\Python\pycharm\PyCharm Community Edition 2020.1\bin;
TEMP	C:\Users\Asus\AppData\Local\Temp
TMP	C:\Users\Asus\AppData\Local\Temp

System variables

Variable	Value
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
path	C:\Python\Python38\Scripts\;C:\Python\Python38\;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 158 Stepping 10, GenuineIntel



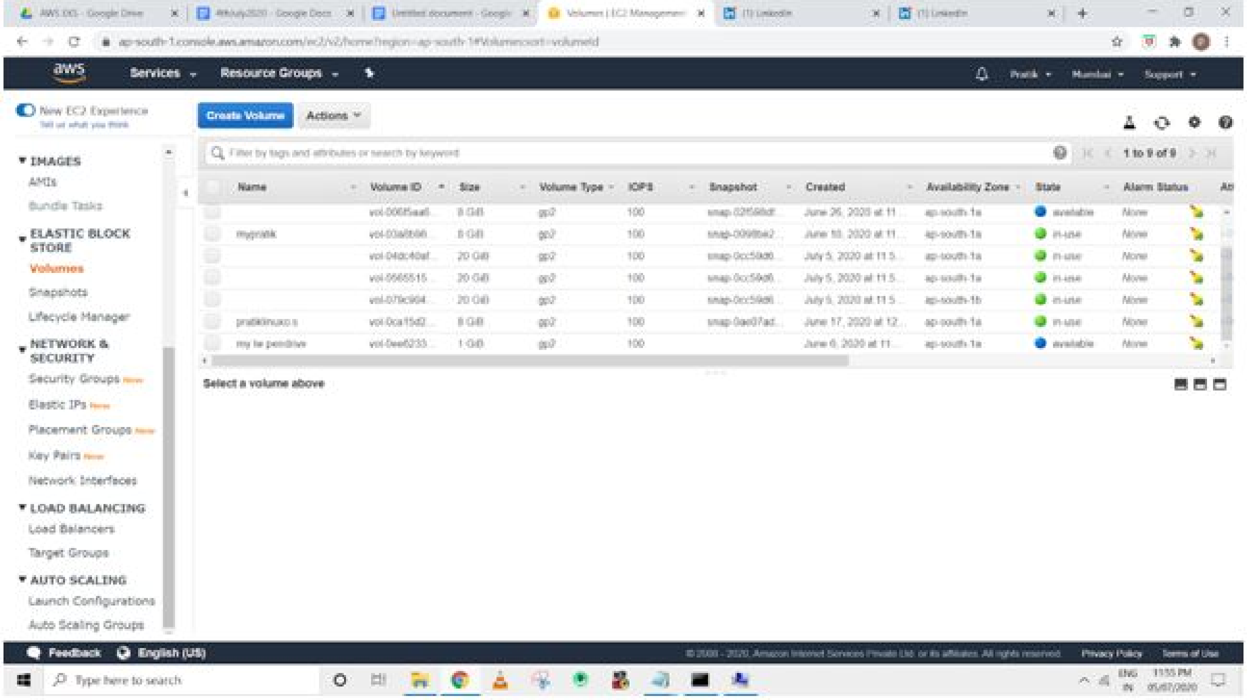
```
C:\Users\Asus\Desktop\eks_class_code1>
C:\Users\Asus\Desktop\eks_class_code1>aws configure
AWS Access Key ID [****************JKA5]:
AWS Secret Access Key [****************KjH4]:
Default region name [ap-south-1]:
Default output format [None]:

C:\Users\Asus\Desktop\eks_class_code1>eksctl create cluster -f cluster.yml
[ℹ]  eksctl version 0.21.0
[ℹ]  using region ap-south-1
[ℹ]  setting availability zones to [ap-south-1b ap-south-1c ap-south-1a]
[ℹ]  subnets for ap-south-1b - public:192.168.0.0/19 private:192.168.96.0/19
[ℹ]  subnets for ap-south-1c - public:192.168.32.0/19 private:192.168.128.0/19
[ℹ]  subnets for ap-south-1a - public:192.168.64.0/19 private:192.168.160.0/19
[ℹ]  nodegroup "ng-1" will use "ami-073969767527f7306" [AmazonLinux2/1.16]
[ℹ]  nodegroup "ng-2" will use "ami-073969767527f7306" [AmazonLinux2/1.16]
[ℹ]  using Kubernetes version 1.16
[ℹ]  creating EKS cluster "myCluster" in "ap-south-1" region with un-managed nodes
[ℹ]  2 nodegroups (ng-1, ng-2) were included (based on the include/exclude rules)
[ℹ]  will create a CloudFormation stack for cluster itself and 2 nodegroup stack(s)
[ℹ]  will create a CloudFormation stack for cluster itself and 0 managed nodegroup stack(s)
[ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=myCluster'
[ℹ]  CloudWatch logging will not be enabled for cluster "myCluster" in "ap-south-1"
[ℹ]  you can enable it with 'eksctl utils update-cluster-logging --region=ap-south-1 --cluster=myCluster'
[ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "myCluster" in "ap-south-1"
[ℹ]  2 sequential tasks: { create cluster control plane "myCluster", 2 sequential sub-tasks: { no tasks, 2 parallel sub-tasks: { create nodegroup "ng-1", create nodegroup "ng-2" } } }
[ℹ]  building cluster stack "eksctl-myCluster-cluster"
[ℹ]  deploying stack "eksctl-myCluster-cluster"
```



NAME	Name	App	Enviroment	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	myCluster-ng-2-Node			i-0e4b31f09d777628f	t2.micro	ap-south-1a	● running	✓ 2/2 checks ...	None
	mypratik			i-095d082621a300a5f	t2.micro	ap-south-1a	● stopped		None
	myCluster-ng-1-Node			i-07e4086192fae95bc	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None
pratiklinuxo.s		web	dev	i-06532a49de6c9cfaa	t2.micro	ap-south-1a	● stopped		None
	myCluster-ng-1-Node			i-01562c4e6febab78e	t2.micro	ap-south-1a	● running	✓ 2/2 checks ...	None
				i-00a535cb0b3992f0f	t2.micro	ap-south-1b	● stopped		None

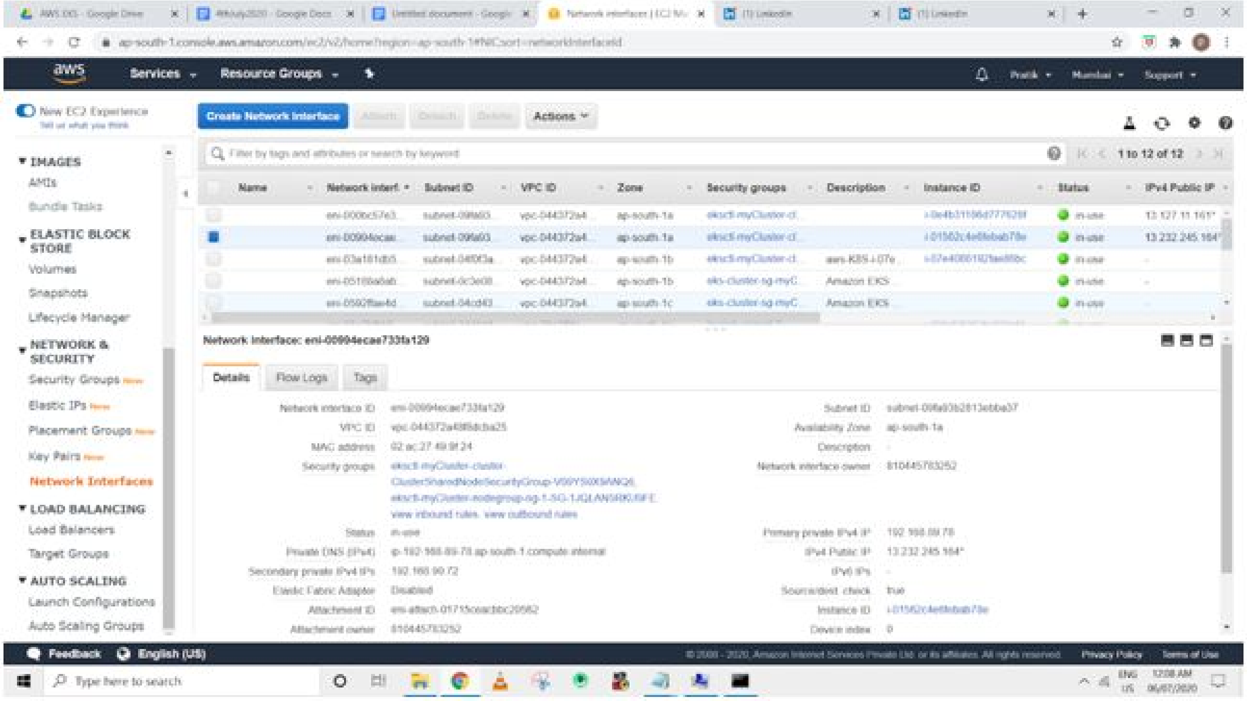
Instance: i-01562c4e6febab78e (myCluster-ng-1-Node) Public DNS: ec2-13-232-245-164.ap-south-1.compute.amazonaws.com

Instance ID	i-01562c4e6febab78e	Public DNS (IPv4)	ec2-13-232-245-164.ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	13.232.245.164
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-192-168-89-78.ap-south-1.compute.internal	Availability zone	ap-south-1a
Private IPs	192.168.83.98, 192.168.89.78	Security groups	eksctl-myCluster-cluster-ClusterSharedNodeSecurityGroup-V09Y5XKWNQ6, eksctl-myCluster-nodegroup-ng-1-SG-1JQLANSRKU9FE. view inbound rules. view outbound rules



Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm Status
	vol-0065aa6...	8 GiB	gp2	100	snap-02f598df...	June 26, 2020 at 11...	ap-south-1a	● available	None
mypratik	vol-03a8b66...	8 GiB	gp2	100	snap-0098be2...	June 10, 2020 at 11...	ap-south-1a	● in-use	None
	vol-04dc40af...	20 GiB	gp2	100	snap-0cc59d6...	July 5, 2020 at 11:5...	ap-south-1a	● in-use	None
	vol-0565515...	20 GiB	gp2	100	snap-0cc59d6...	July 5, 2020 at 11:5...	ap-south-1a	● in-use	None
	vol-079c904...	20 GiB	gp2	100	snap-0cc59d6...	July 5, 2020 at 11:5...	ap-south-1b	● in-use	None
pratiklinuxo.s	vol-0ca15d2...	8 GiB	gp2	100	snap-0ae07ad...	June 17, 2020 at 12...	ap-south-1a	● in-use	None
my hw pendrive	vol-0ee6233...	1 GiB	gp2	100		June 6, 2020 at 11...	ap-south-1a	● available	None

Select a volume above

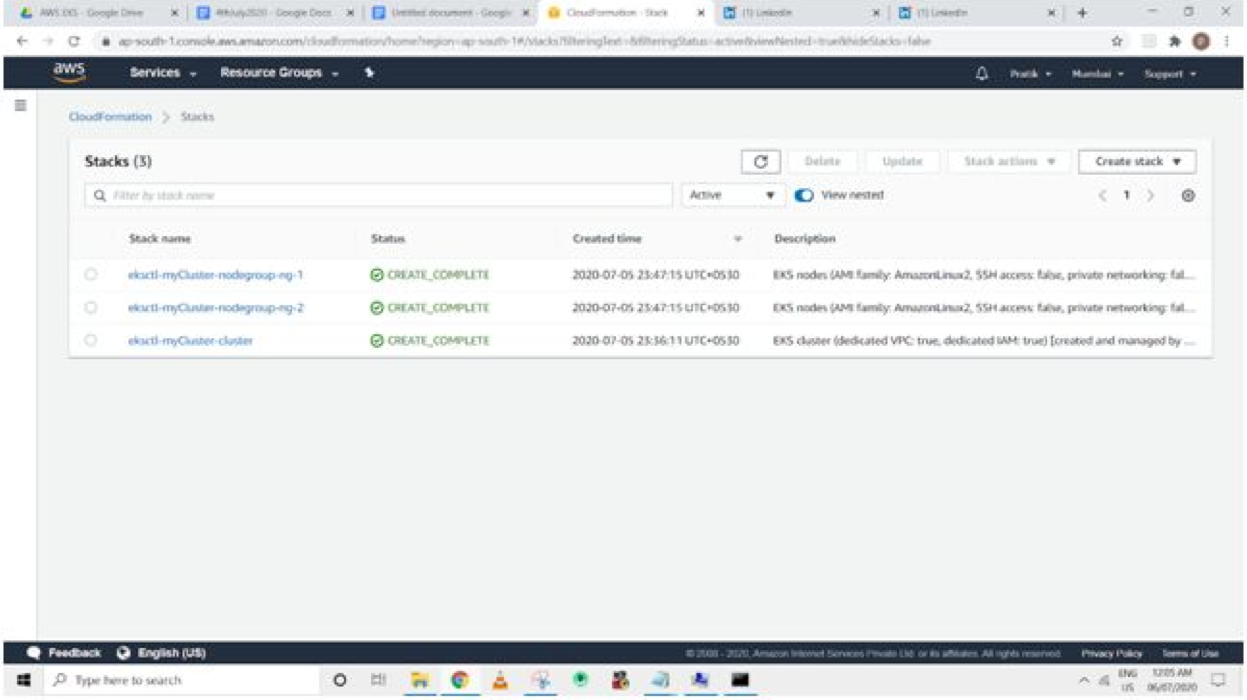


Name	Network interf	Subnet ID	VPC ID	Zone	Security groups	Description	Instance ID	Status	IPv4 Public IP
	eni-000bc57e3...	subnet-09fa03...	vpc-044372a4...	ap-south-1a	eksctl-myCluster-cl...		i-0e4b31106d777626f	in-use	13.127.11.161*
	eni-00994ecae...	subnet-09fa03...	vpc-044372a4...	ap-south-1a	eksctl-myCluster-cl...		i-01562c4e6febab78e	in-use	13.232.245.164*
	eni-03a181db5...	subnet-04f0f3a...	vpc-044372a4...	ap-south-1b	eksctl-myCluster-cl...	aws-K8S-i-07e...	i-07e4086192fae86bc	in-use	-
	eni-0518ba6ab...	subnet-0c3e08...	vpc-044372a4...	ap-south-1b	eks-cluster-sg-myC...	Amazon EKS ...		in-use	-
	eni-0592ffae4d...	subnet-04cd43...	vpc-044372a4...	ap-south-1c	eks-cluster-sg-myC...	Amazon EKS ...		in-use	-

Network interface: eni-00994ecae733fa129

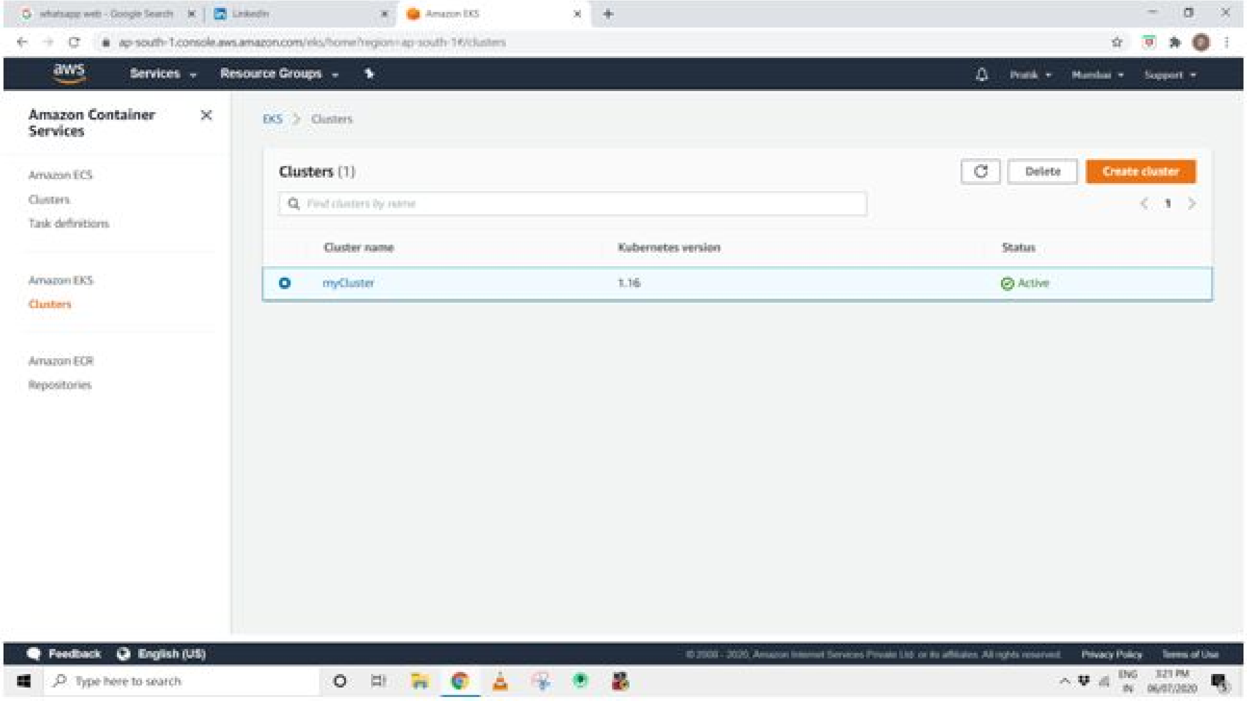
Details | Flow Logs | Tags

Network interface ID	eni-00994ecae733fa129	Subnet ID	subnet-09fa03b2813ebba37
VPC ID	vpc-044372a4f8fdcba25	Availability Zone	ap-south-1a
MAC address	02:ac:27:49:9f:24	Description	-
Security groups	eksctl-myCluster-cluster-ClusterSharedNodeSecurityGroup-V09YS0X9ANQ6, eksctl-myCluster-nodegroup-ng-1-SG-1JQLAN5RKU9FE view inbound rules, view outbound rules	Network interface owner	810445783252
Status	in-use	Primary private IPv4 IP	192.168.89.78
Private DNS (IPv4)	ip-192-168-89-78.ap-south-1.compute.internal	IPv4 Public IP	13.232.245.164*
Secondary private IPv4 IPs	192.168.90.72	IPv6 IPs	-
Elastic Fabric Adapter	Disabled	Source/dest. check	true
Attachment ID	eni-attach-01715ceacbbc20582	Instance ID	i-01562c4e6febab78e
Attachment owner	810445783252	Device index	0



Stacks (3)

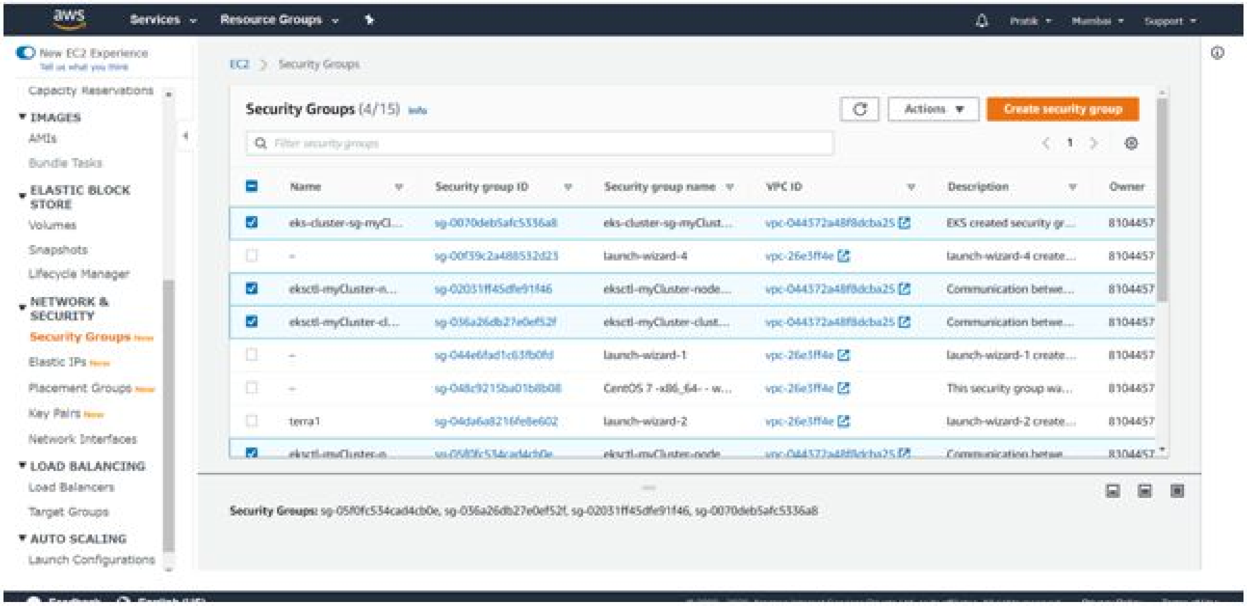
Stack name	Status	Created time	Description
eksctl-myCluster-nodegroup-ng-1	CREATE_COMPLETE	2020-07-05 23:47:15 UTC+0530	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: fal...
eksctl-myCluster-nodegroup-ng-2	CREATE_COMPLETE	2020-07-05 23:47:15 UTC+0530	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: fal...
eksctl-myCluster-cluster	CREATE_COMPLETE	2020-07-05 23:36:11 UTC+0530	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by ...



EKS > Clusters

Clusters (1)

	Cluster name	Kubernetes version	Status
●	myCluster	1.16	Active



EC2 > Security Groups

Security Groups (4/15)

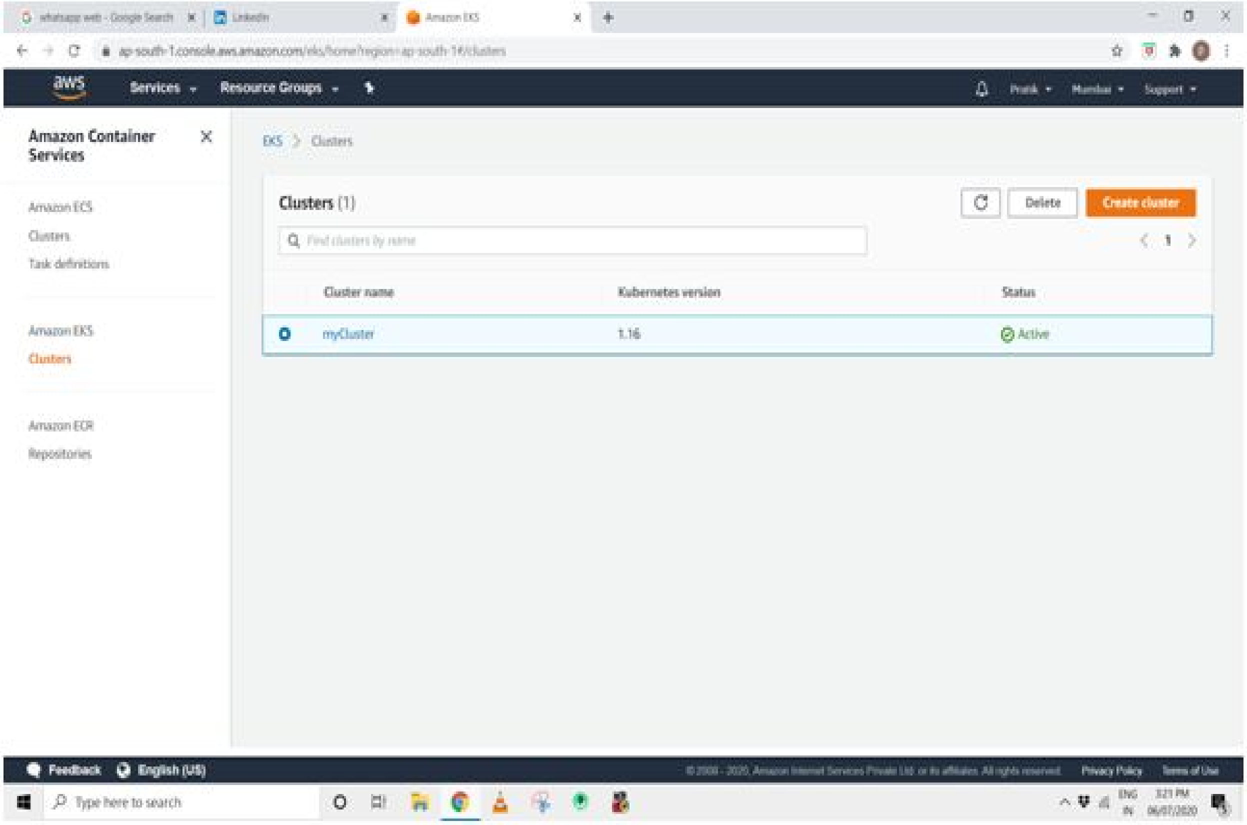
	Name	Security group ID	Security group name	VPC ID	Description	Owner
☑	eks-cluster-sg-myCl...	sg-0070deb5afc5336a8	eks-cluster-sg-myClust...	vpc-044372a48f8dcba25	EKS created security gr...	8104457
□	–	sg-00f39c2a488532d23	launch-wizard-4	vpc-26e3ff4e	launch-wizard-4 create...	8104457
☑	eksctl-myCluster-n...	sg-02031ff45dfe91f46	eksctl-myCluster-node...	vpc-044372a48f8dcba25	Communication betwe...	8104457
☑	eksctl-myCluster-cl...	sg-036a26db27e0ef52f	eksctl-myCluster-clust...	vpc-044372a48f8dcba25	Communication betwe...	8104457
□	–	sg-044e6fad1c63fb0fd	launch-wizard-1	vpc-26e3ff4e	launch-wizard-1 create...	8104457
□	–	sg-048c9215ba01b8b08	CentOS 7 -x86_64- - w...	vpc-26e3ff4e	This security group wa...	8104457
□	terra1	sg-04da6a8216fe8e602	launch-wizard-2	vpc-26e3ff4e	launch-wizard-2 create...	8104457
☑	eksctl-myCluster-n...	sg-05f0fc534cad4cb0e	eksctl-myCluster-node...	vpc-044372a48f8dcba25	Communication betwe...	8104457

Security Groups: sg-05f0fc534cad4cb0e, sg-036a26db27e0ef52f, sg-02031ff45dfe91f46, sg-0070deb5afc5336a8

launching a fargate using fargate.yml

```
eksctl create cluster -f fargate.yml
```

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: f-lwcluster
  region: ap-southeast-1

fargateProfiles:
  - name: fargate-default
            selectors:
            - namespace: kube-system
            - namespace: default
```



Cluster name	Kubernetes version	Status
myCluster	1.16	Active

Creating the jenkins deployment , a secret and pvc using customisation file

```
kubectl create -k
```

```
apiVersion: v1

kind: Service

metadata:

  name: my-jenkins

  labels:

            app: jenkinsapp

spec:

 ports:
  - port: 80
```

Creating the jenkins deployment , a secret and pvc using customisation file kubectl create -k .

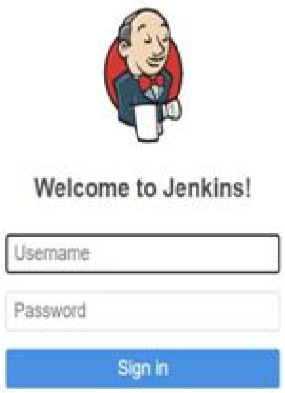
```
apiVersion: v1
kind: Service
metadata:
  name: my-jenkins
  labels:
        app: jenkinsapp
spec:
 ports:
  - port: 80


selector:
        app: jenkinsapp
  type: NodePort
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: my-jenkins
  labels:
        app: jenkinsapp spec:
  selector:
        matchLabels:
             app: jenkinsapp
  tier:frontend
  strategy:
             type: Recreate
  template:
             metadata:
     labels:
             app: jenkinsapp
     tier:frontend
       spec:
     containers:
             - image: jenkins/jenkins
     name:myjenkins


     env:
      - name: MYJENKINS_USER
valueFrom:
secretKeyRef:
             name: myjenkinssecret
          key: username
     - name: MYJENKINS_PASSWORD
valueFrom:
secretKeyRef:
             name: myjenkinssecret
          key:vpass
    ports:
     - containerPort: 80
```

```
name: jenkins-cont
volumeMounts:
    - name: myjenkins-persistent-storage
mountPath:/var/jenkins_home
volumes:
    - name: myjenkins-persistent-storage
persistentVolumeClaim
    -  claimName: efs-myjenkins
---
kind: StorageClass apiVersion: storage.k8s.io/v1
metadata:
  name:  aws-efs
  provisioner: jenkins/aws-efs
---
kind: PersistentVolumeClaim apiVersion: v1
metadata:
  name: efs-myjenkins
  annotations:
          volume.beta.kubernetes.io/storage-class: "aws-efs" spec:
  accessModes:
          - ReadWriteMany
  resources:
          requests:
          storage: 10Gi
---
apiVersion: v1 kind: Secret metadata:
  name: myjenkinssecret
```

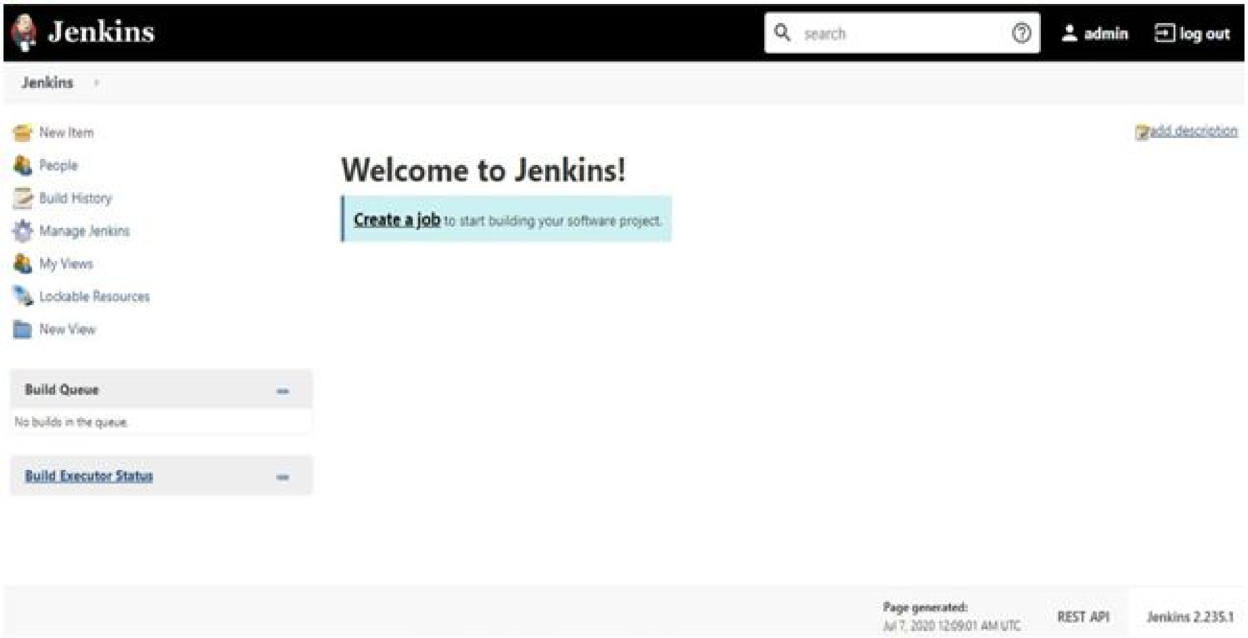
```
data:
  username: WYRat4KK
    vpass: gmKkaG0F
```

kustomisation file:

```
apiVersion: kustomize.config.k8s.io/v1beta1

kind: Kustomization
resources:
 - my-jenkins.yml
 - secret.yml
```



Welcome to Jenkins!



Welcome to Jenkins!

Create a job to start building your software project.

No builds in the queue.

Page generated: Jul 7, 2020 12:09:01 AM UTC

Ø Creating a efs setup and mounting the volume using yml file

- security Configuration
- Creation of storage class

1. Create provisioner

```
kind: Deployment

apiVersion: apps/v1

metadata:

  name: efs-provisioner

spec:

 selector:
```

```
 matchLabels:

  app: efs-provisioner

replicas: 1

strategy:

 type: Recreate

template:

 metadata:

  labels:

        app: efs-provisioner

 spec:

  containers:

        - name: efs-provisioner

        image: quay.io/external_storage/efs-provisioner:v0.1.0

env:

        - name: FILE_SYSTEM_ID

        value: fs-68bc37b9

        - name: AWS_REGION

        value: ap-southeast-1

        - name: PROVISIONER_NAME

        value: jenkins/aws-efs

        volumeMounts:

        - name: pv-volume

        mountPath: /persistentvolumes

  volumes:

        - name: pv-volume

        nfs:

        server: fs-87cb37c9.efs.ap-south-1.amazonaws.com
```

```
          path: /
```

create-rbac(security)

```
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: nfs-provisioner-role-binding
subjects:
  - kind: ServiceAccount
          name: default
          namespace: jenkins
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
```

Creating storage-class and mounting volume

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
 name: aws-efs
provisioner: jenkins/aws-efs
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
 name: jenkins-efs
 annotations:
  volume.beta.kubernetes.io/storage-class: "aws-efs"
spec:
 accessModes:
  - ReadWriteMany
 resources:
  requests:
   storage: 1Gi
```

Deploying jenkins

```
apiVersion: v1
kind: Service
metadata:
 name: myjenkinsvcv
 labels:
  app: jenkinsapp
spec:
 ports:
  - port: 8080
selector:
 app: jenkinsapp
 tier: frontened
type: LoadBalancer
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
 name: my-jenkins
 labels:
  app: jenkinsapp
spec:
 selector:
  matchLabels:
   app: jenkinsapp
   tier: frontened
 strategy:
  type: Recreate
 template:
  metadata:
   labels:
            app: jenkinsapp
            tier: frontened
spec:
 containers:
 - image: jenkins/jenkins
   name: myjenkins
env:
  - name: MYJENKINS_PASSWORD
            valueFrom:
            secretKeyRef:
            name: myjenkins-pass
            key: password
  ports:
  - containerPort: 8080
            name: mysql
            volumeMounts:
            - name: myjenkins-persistent-storage
            mountPath: /var/jenkins_home
   volumes:
   - name: myjenkins-persistent-storage
            persistentVolumeClaim:
             claimName: efs-myjenkins
```

USING HELM(client) AND TILLER(server):

https://github.com/helm/helm/releases

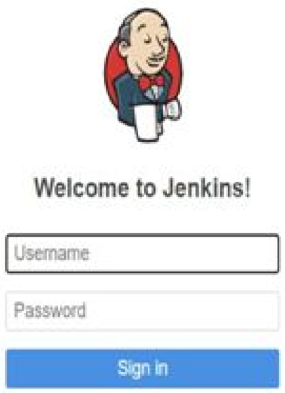
```
# helm init

# helm repo add stable https://kubernetes-charts.storage.googleapis.com/

# helm repo list

# helm repo update
```

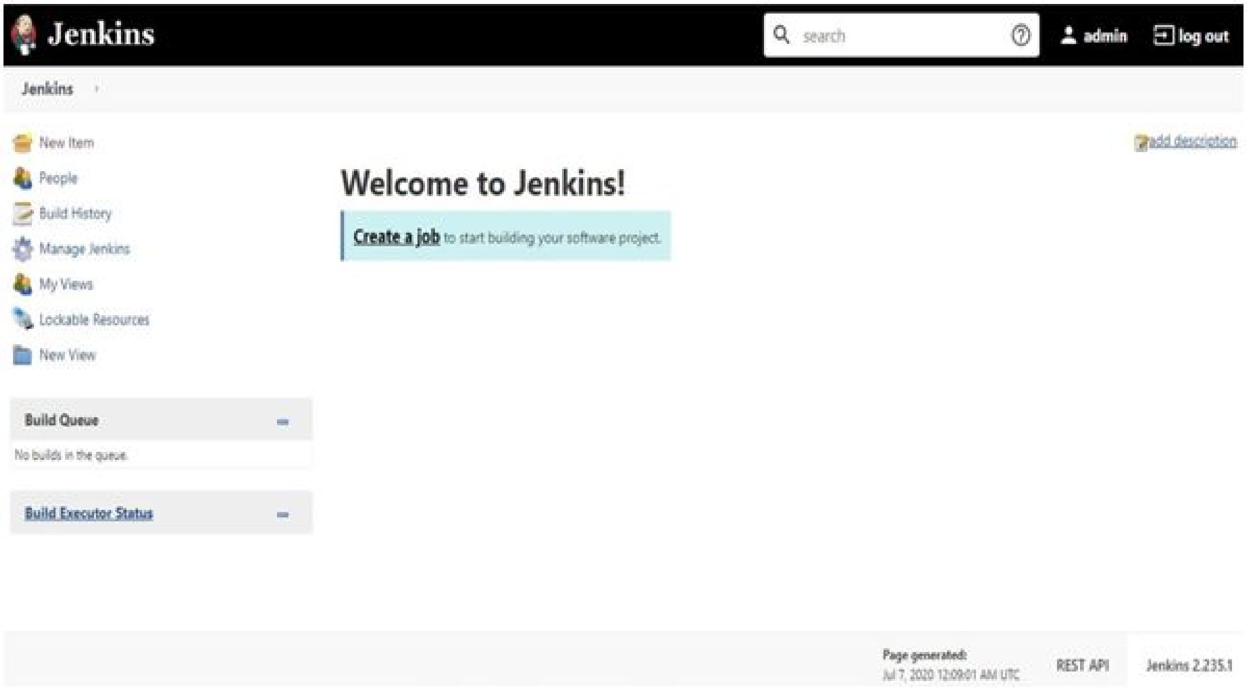
```
# kubectl -n kube-system create serviceaccount tiller
```

```
# kubectl create clusterrolebinding tiller --clusterrole cluster-admin

--serviceaccount=kube-system:tiller

# helm init --service-account tiller
```

```
# kubectl get pods --namespace kube-system
```

```
helm install stable/jenkins --version 2.1.2 --namespace jenkins --set

master.usePodSecurityContext=True --set master.adminPassword="redhat" --set

persistence.enabled=true --set persistence.storageClass=default --set
persistence.accessMode=ReadWriteOnce --set persistence.size=5Gi
```



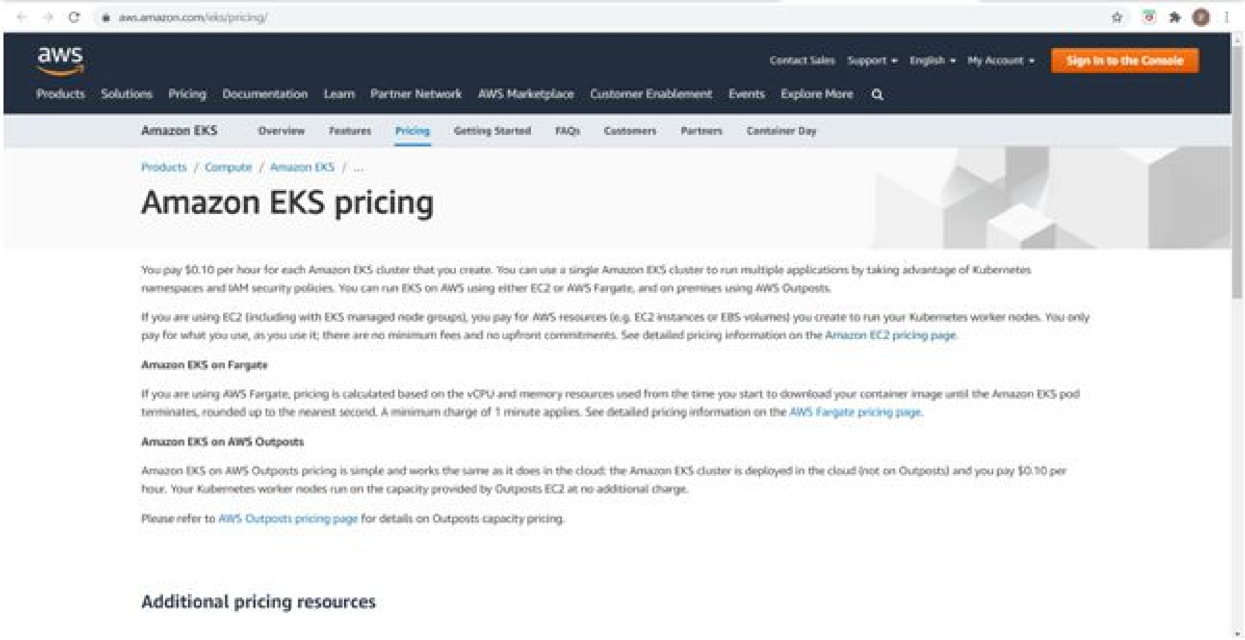


Welcome to Jenkins!

Create a job to start building your software project.

No builds in the queue.

Page generated: Jul 7, 2020 12:09:01 AM UTC



Amazon EKS pricing

You pay $0.10 per hour for each Amazon EKS cluster that you create. You can use a single Amazon EKS cluster to run multiple applications by taking advantage of Kubernetes namespaces and IAM security policies. You can run EKS on AWS using either EC2 or AWS Fargate, and on premises using AWS Outposts.

If you are using EC2 (including with EKS managed node groups), you pay for AWS resources (e.g. EC2 instances or EBS volumes) you create to run your Kubernetes worker nodes. You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments. See detailed pricing information on the Amazon EC2 pricing page.

Amazon EKS on Fargate

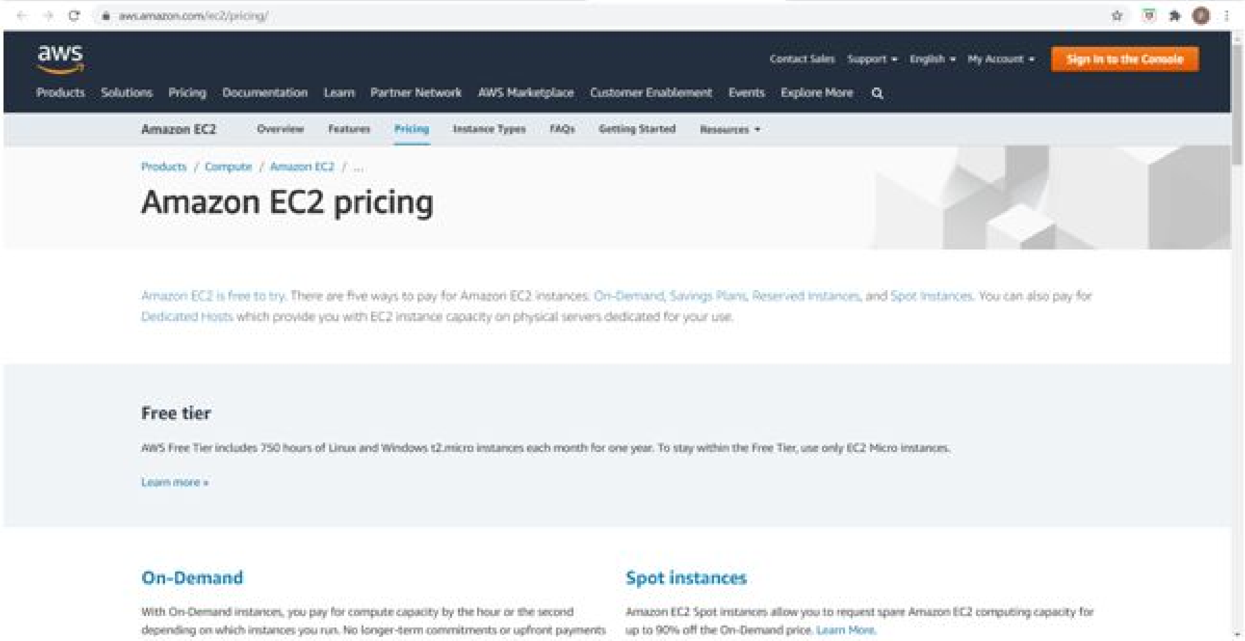
If you are using AWS Fargate, pricing is calculated based on the vCPU and memory resources used from the time you start to download your container image until the Amazon EKS pod terminates, rounded up to the nearest second. A minimum charge of 1 minute applies. See detailed pricing information on the AWS Fargate pricing page.

Amazon EKS on AWS Outposts

Amazon EKS on AWS Outposts pricing is simple and works the same as it does in the cloud: the Amazon EKS cluster is deployed in the cloud (not on Outposts) and you pay $0.10 per hour. Your Kubernetes worker nodes run on the capacity provided by Outposts EC2 at no additional charge.

Please refer to AWS Outposts pricing page for details on Outposts capacity pricing.

Additional pricing resources



Amazon EC2 pricing

Amazon EC2 is free to try. There are five ways to pay for Amazon EC2 instances: On-Demand, Savings Plans, Reserved Instances, and Spot Instances. You can also pay for Dedicated Hosts which provide you with EC2 instance capacity on physical servers dedicated for your use.

Free tier

AWS Free Tier includes 750 hours of Linux and Windows t2.micro instances each month for one year. To stay within the Free Tier, use only EC2 Micro instances.

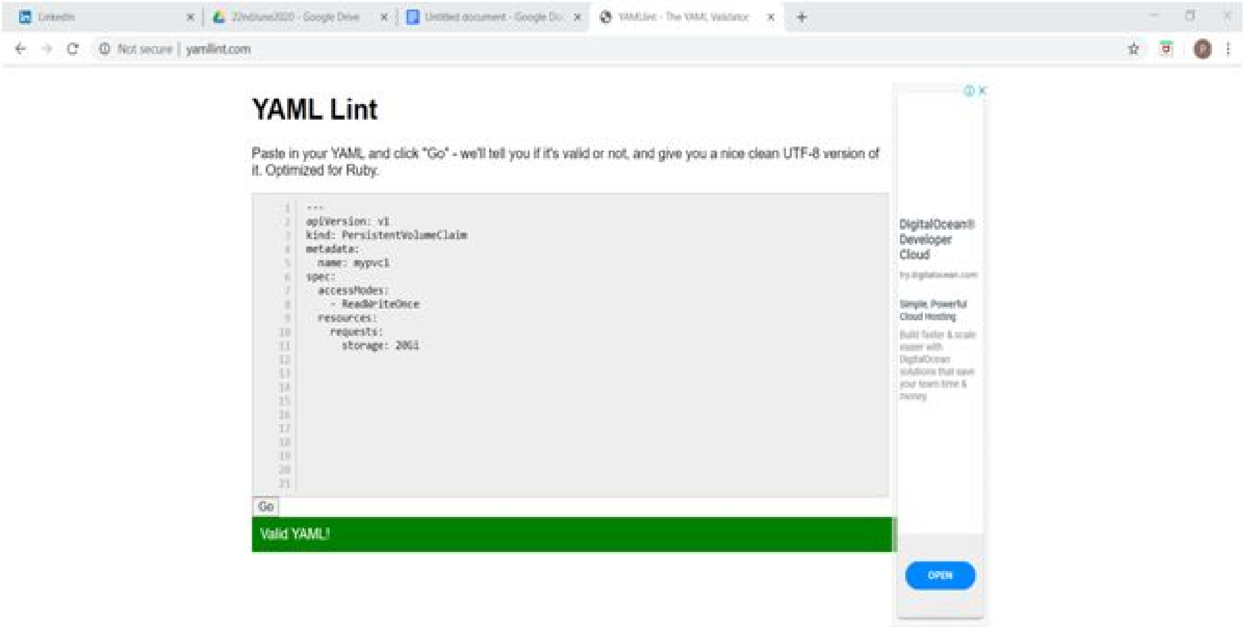
Learn more »

On-Demand

With On-Demand instances, you pay for compute capacity by the hour or the second depending on which instances you run. No longer-term commitments or upfront payments

Spot instances

Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price. Learn More.


LinkedIn
22ndJune2020 - Google Drive
Untitled document - Google Do
YAMLlint - The YAML Validator
Not secure | yamllint.com

YAML Lint

Paste in your YAML and click "Go" - we'll tell you if it's valid or not, and give you a nice clean UTF-8 version of it. Optimized for Ruby.

```
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc1
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
```

Valid YAML!