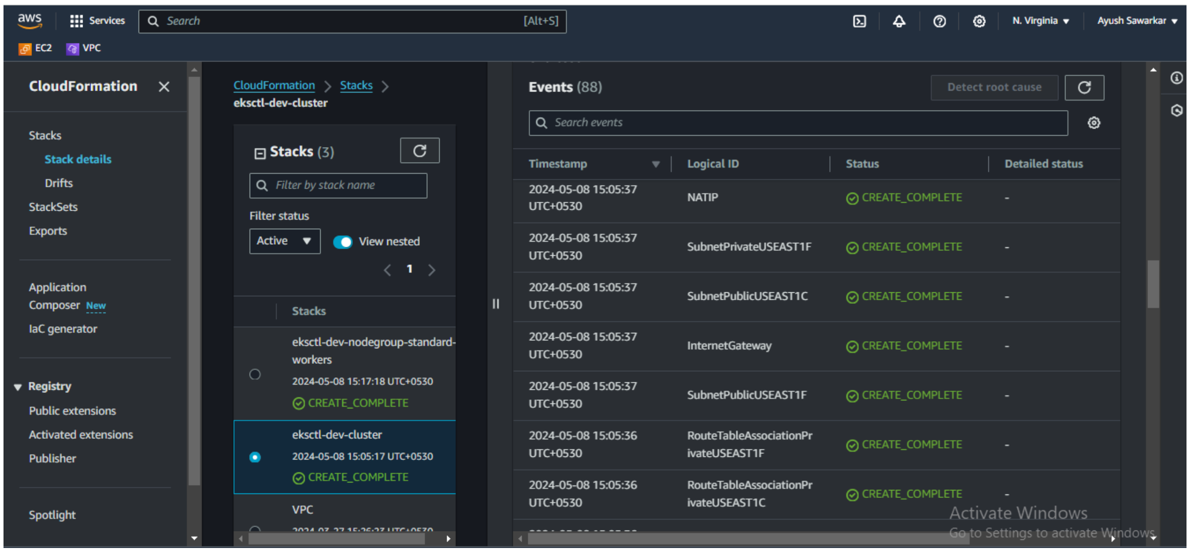
Table of Contents

K8s Assignment

Create an EKS cluster using Infrastructure as Code (IaC) i.e Cloud Formation



Stacks
eksctl-dev-nodegroup-standard-workers 2024-05-08 15:17:18 UTC+0530 CREATE_COMPLETE
eksctl-dev-cluster 2024-05-08 15:05:17 UTC+0530 CREATE_COMPLETE
VPC

Events (88)

Timestamp	Logical ID	Status	Detailed status
2024-05-08 15:05:37 UTC+0530	NATIP	CREATE_COMPLETE	-
2024-05-08 15:05:37 UTC+0530	SubnetPrivateUSEAST1F	CREATE_COMPLETE	-
2024-05-08 15:05:37 UTC+0530	SubnetPublicUSEAST1C	CREATE_COMPLETE	-
2024-05-08 15:05:37 UTC+0530	InternetGateway	CREATE_COMPLETE	-
2024-05-08 15:05:37 UTC+0530	SubnetPublicUSEAST1F	CREATE_COMPLETE	-
2024-05-08 15:05:36 UTC+0530	RouteTableAssociationPrivateUSEAST1F	CREATE_COMPLETE	-
2024-05-08 15:05:36 UTC+0530	RouteTableAssociationPrivateUSEAST1C	CREATE_COMPLETE	-

In this First CloudFormation of "eksctl-dev-cluster" I have developed these things

VPC (Virtual Private Cloud): A virtual network in the AWS cloud.

Internet Gateway: Enables communication between instances in your VPC and the internet.

NAT Gateway: Allows instances in private subnets to connect to the internet while preventing inbound connections initiated by the internet.

Route Tables: Defines the routes for network traffic leaving the subnets.

Subnets: Divisions of the VPC's IP address range in which you can launch AWS resources.

Security Groups: Act as a virtual firewall for controlling inbound and outbound traffic to AWS resources.

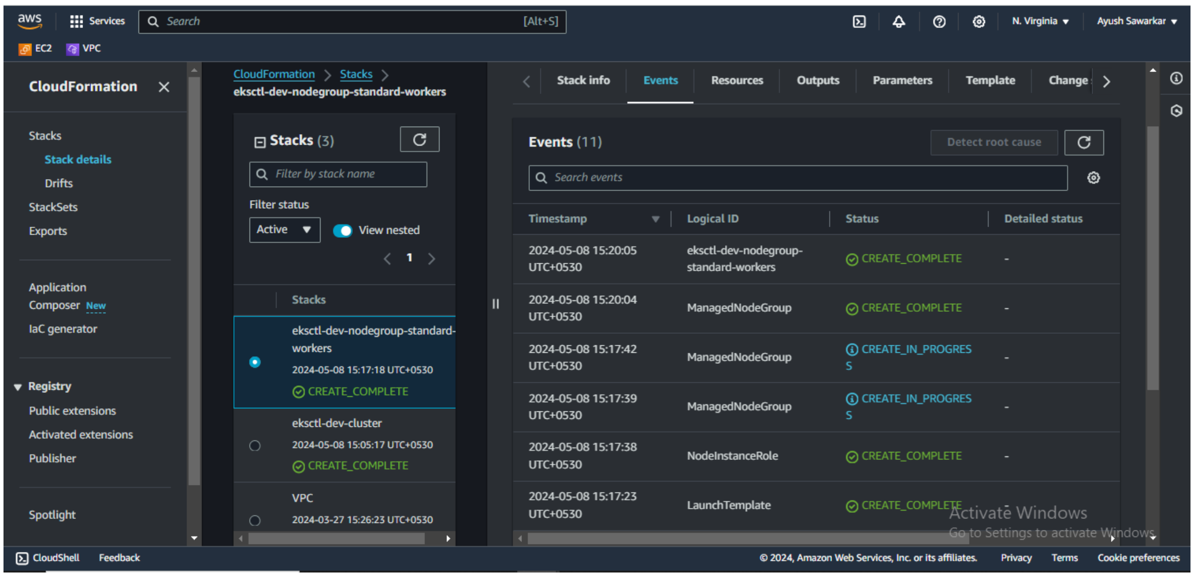
Service Role: A role used by AWS services to access other AWS resources.

Cluster: The EKS cluster itself, which is a managed Kubernetes service from AWS.

Ingress Rules: Configurations allowing inbound traffic to reach specific resources or services within the cluster.

Route Table Associations: Associates subnets with route tables, enabling the routing of traffic.

Cluster and nodes Deployment



CloudFormation > Stacks > eksctl-dev-nodegroup-standard-workers

Stacks (3)

Filter status: Active — View nested

Stacks
eksctl-dev-nodegroup-standard-workers 2024-05-08 15:17:18 UTC+0530 CREATE_COMPLETE
eksctl-dev-cluster 2024-05-08 15:05:17 UTC+0530 CREATE_COMPLETE
VPC 2024-03-27 15:26:23 UTC+0530

Stack info | Events | Resources | Outputs | Parameters | Template | Change

Events (11)

Timestamp	Logical ID	Status	Detailed status
2024-05-08 15:20:05 UTC+0530	eksctl-dev-nodegroup-standard-workers	CREATE_COMPLETE	-
2024-05-08 15:20:04 UTC+0530	ManagedNodeGroup	CREATE_COMPLETE	-
2024-05-08 15:17:42 UTC+0530	ManagedNodeGroup	CREATE_IN_PROGRESS	-
2024-05-08 15:17:39 UTC+0530	ManagedNodeGroup	CREATE_IN_PROGRESS	-
2024-05-08 15:17:38 UTC+0530	NodeInstanceRole	CREATE_COMPLETE	-
2024-05-08 15:17:23 UTC+0530	LaunchTemplate	CREATE_COMPLETE	

Activate Windows
Go to Settings to activate Windows.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

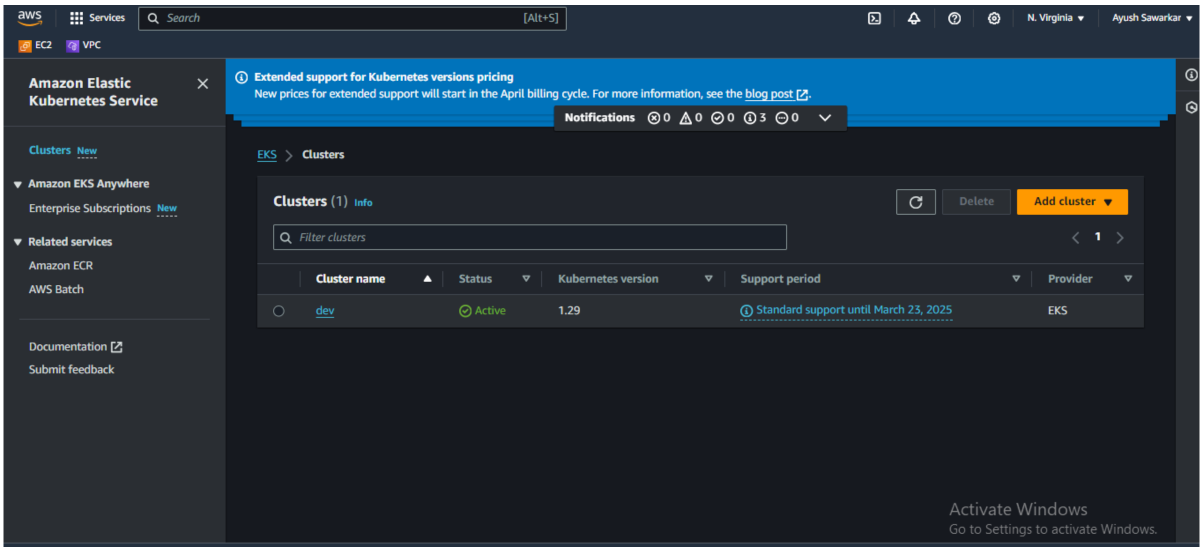
After That in another CloudFormation Template "eksctl-dev-nodegroup-standard-workers" I have created

LaunchTemplate: Defines an EC2 launch template used for launching instances in the managed node group. It specifies configurations such as block device mappings, metadata options, security group IDs, and tags.

ManagedNodeGroup: Specifies the configuration for the EKS managed node group. It includes settings like AMI type, instance types, launch template ID, node role, scaling configuration, subnets, and tags.

NodeInstanceRole: Defines an IAM role used by the EC2 instances in the node group. It specifies the assume role policy document, managed policy ARNs, path, and tags.

This is the EKS which has been Deployed.



Extended support for Kubernetes versions pricing
New prices for extended support will start in the April billing cycle. For more information, see the blog post.

EKS > Clusters

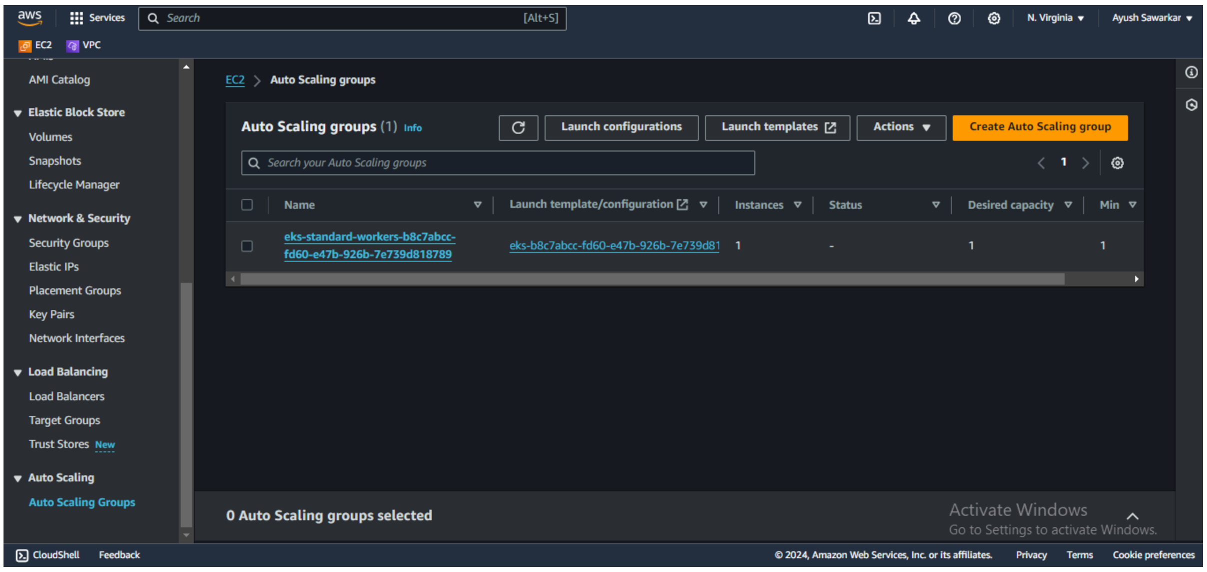
Clusters (1) Info — Delete — Add cluster

Cluster name	Status	Kubernetes version	Support period	Provider
dev	Active	1.29	Standard support until March 23, 2025	EKS

Activate Windows
Go to Settings to activate Windows.

Auto Scaling Group

In this only I have created the auto Scaling group for the instances will be scale out.

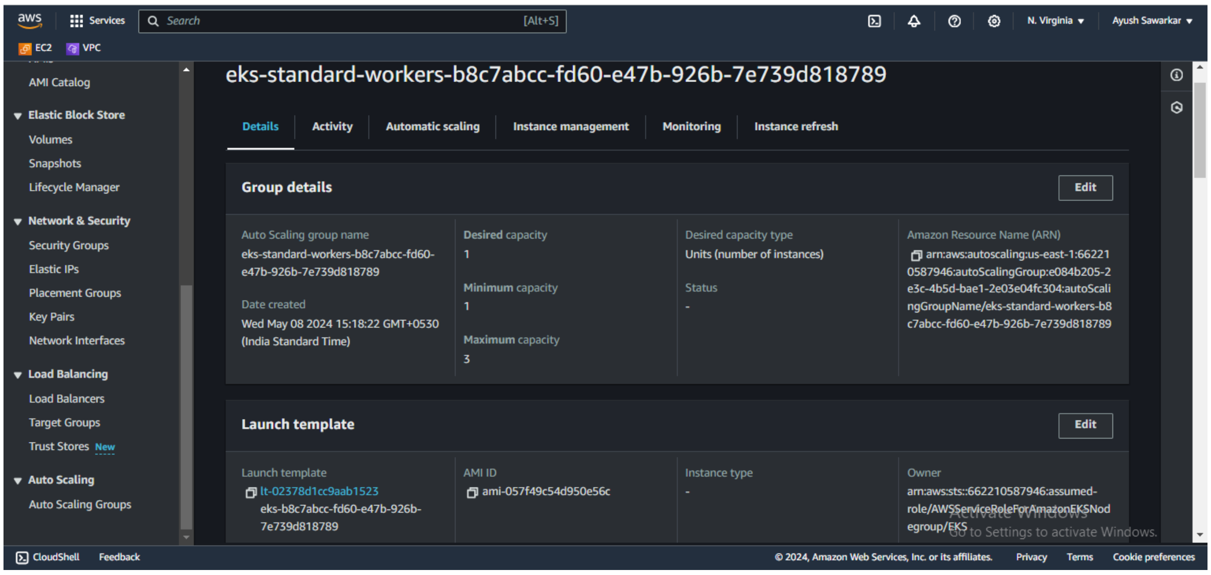


EC2 > Auto Scaling groups

Auto Scaling groups (1)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
eks-standard-workers-b8c7abcc-fd60-e47b-926b-7e739d818789	eks-b8c7abcc-fd60-e47b-926b-7e739d81	1	-	1	1

0 Auto Scaling groups selected



eks-standard-workers-b8c7abcc-fd60-e47b-926b-7e739d818789

Group details

Auto Scaling group name: eks-standard-workers-b8c7abcc-fd60-e47b-926b-7e739d818789

Date created: Wed May 08 2024 15:18:22 GMT+0530 (India Standard Time)

Desired capacity: 1

Minimum capacity: 1

Maximum capacity: 3

Desired capacity type: Units (number of instances)

Status: -

Amazon Resource Name (ARN): arn:aws:autoscaling:us-east-1:662210587946:autoScalingGroup:e084b205-2e3c-4b5d-bae1-2e03e04fc304:autoScalingGroupName/eks-standard-workers-b8c7abcc-fd60-e47b-926b-7e739d818789

Launch template

Launch template: lt-02378d1cc9aab1523 eks-b8c7abcc-fd60-e47b-926b-7e739d818789

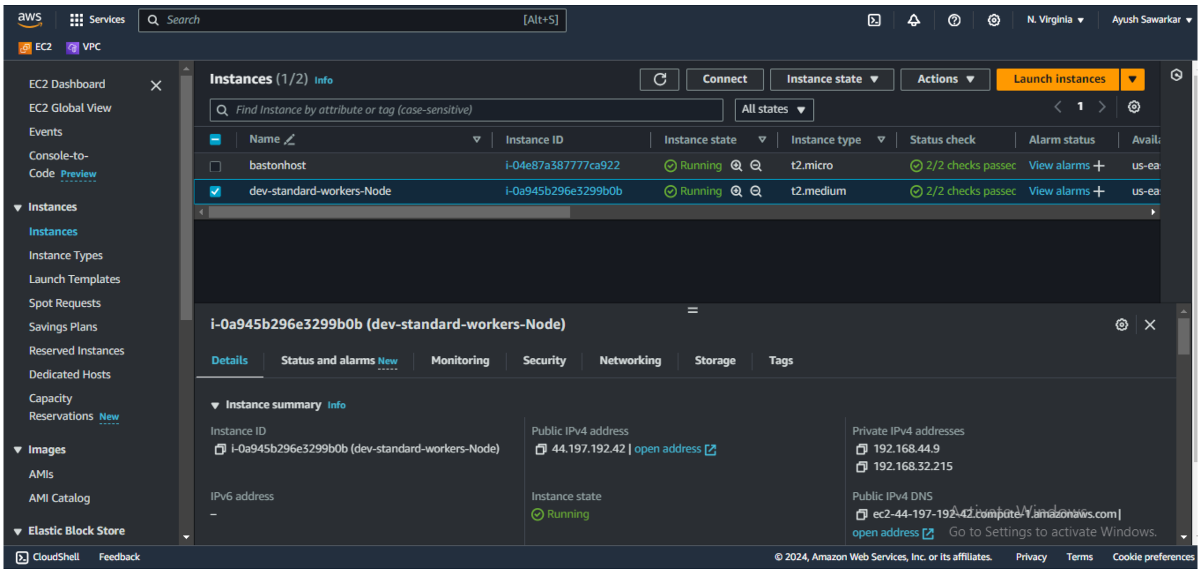
AMI ID: ami-057f49c54d950e56c

Instance type: -

Owner: arn:aws:sts::662210587946:assumed-role/AWSServiceRoleForAmazonEKSNodegroup/EKS

In the above Screenshot I have defined this thing

```
"NodegroupName": "standard-workers",
    "ScalingConfig": {
      "DesiredSize": 1,
      "MaxSize": 3,
      "MinSize": 1
    }
```



Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avail
bastonhost	i-04e87a387777ca922	Running	t2.micro	2/2 checks passec	View alarms	us-ea
dev-standard-workers-Node	i-0a945b296e3299b0b	Running	t2.medium	2/2 checks passec	View alarms	us-ea

i-0a945b296e3299b0b (dev-standard-workers-Node)

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID i-0a945b296e3299b0b (dev-standard-workers-Node)	Public IPv4 address 44.197.192.42 | open address	Private IPv4 addresses 192.168.44.9 192.168.32.215
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-44-197-192-42.compute-1.amazonaws.com | open address

Activate Windows
Go to Settings to activate Windows.

© 2024, Amazon Web Services, Inc. or its affiliates.

Where this server has been Spined up as the desired capacity is one.

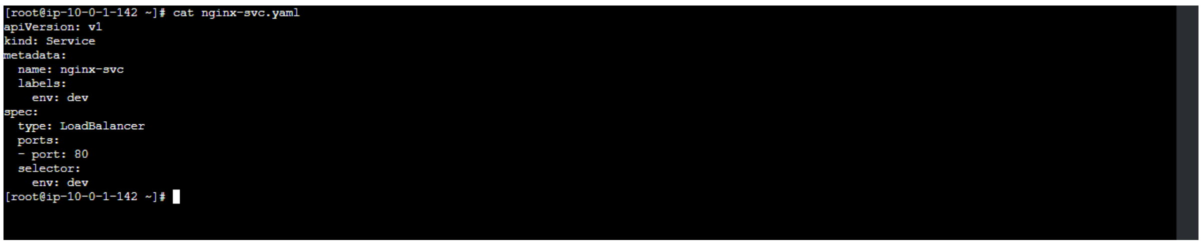
Using the below Command I have attached the Secured Cluster to the Baston Host server fom where I can control the Cluster I can Also be able to do the same thing using the Command Prompt.

aws eks update-kubeconfig --name my-cluster --region us-east-1

Creating application load balancer for EKS

Created nignx-svc.yaml

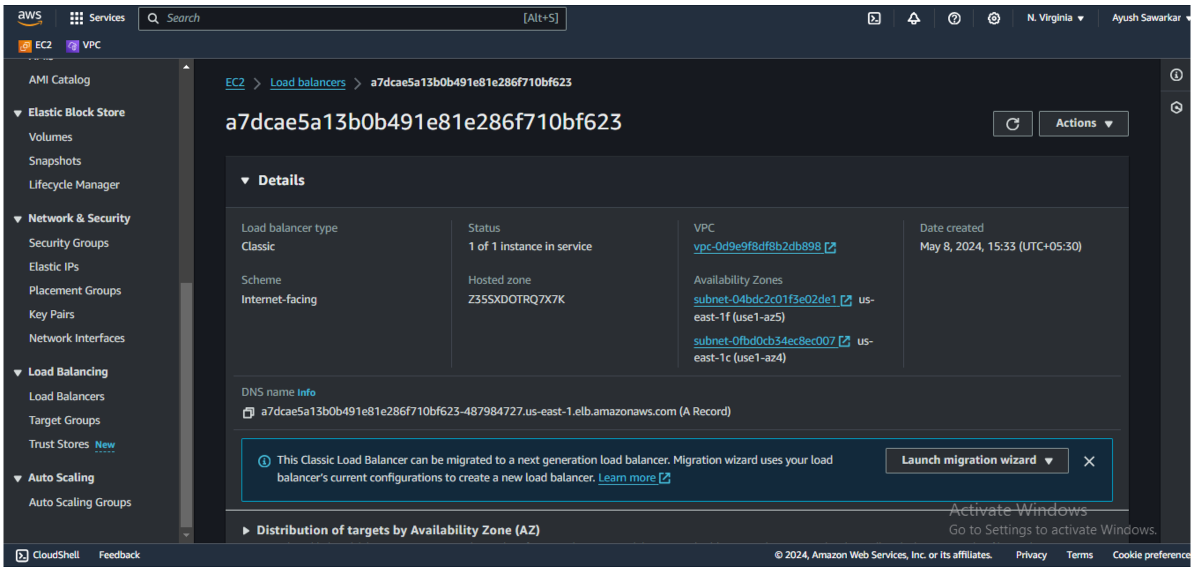
In this file I have created the application load balancer to balance the load on all server evenly for the dev environment which is cluster.



```
[root@ip-10-0-1-142 ~]# cat nginx-svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
  labels:
    env: dev
spec:
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    env: dev
[root@ip-10-0-1-142 ~]#
```

After that apply the File using

kubectl apply -f ./nginx-svc.yaml


a7dcae5a13b0b491e81e286f710bf623

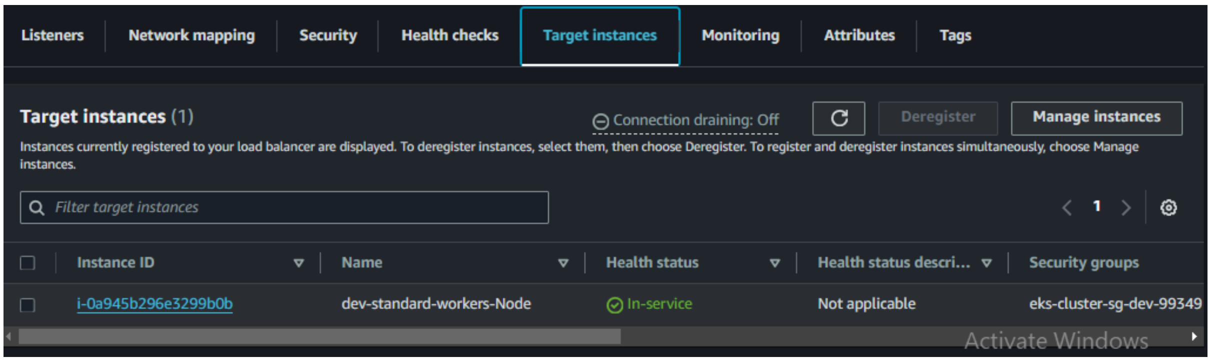
Details

Load balancer type: Classic
Status: 1 of 1 instance in service
VPC: vpc-0d9e9f8df8b2db898
Date created: May 8, 2024, 15:33 (UTC+05:30)
Scheme: Internet-facing
Hosted zone: Z35SXDOTRQ7X7K
Availability Zones: subnet-04bdc2c01f3e02de1 us-east-1f (use1-az5); subnet-0fbd0cb34ec8ec007 us-east-1c (use1-az4)

DNS name: a7dcae5a13b0b491e81e286f710bf623-487984727.us-east-1.elb.amazonaws.com (A Record)

This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. Learn more

Load Balancer with "a7dcae5a13b0b491e81e286f710bf623" has been created including all the instances with the cluster will be directly attach here.

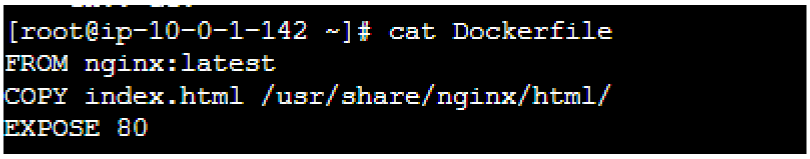

Target instances (1)

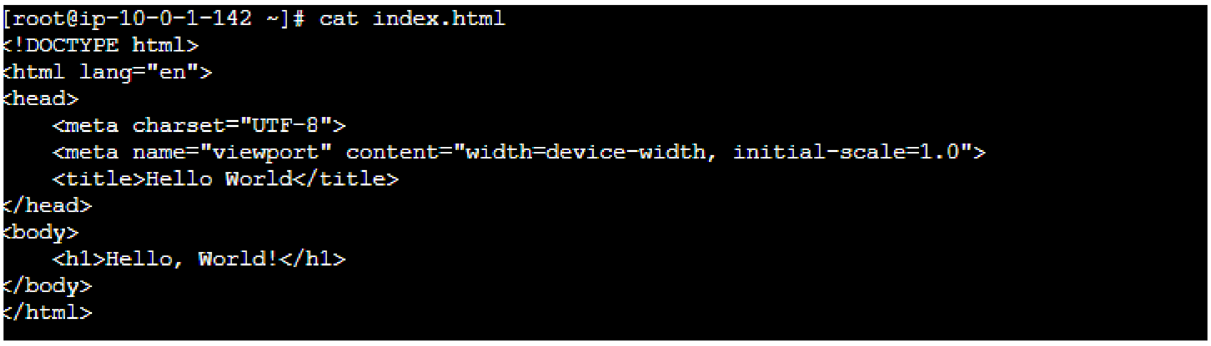
Instances currently registered to your load balancer are displayed. To deregister instances, select them, then choose Deregister. To register and deregister instances simultaneously, choose Manage instances.

Instance ID	Name	Health status	Health status descri...	Security groups
i-0a945b296e3299b0b	dev-standard-workers-Node	In-service	Not applicable	eks-cluster-sg-dev-99349

Created A Docker File

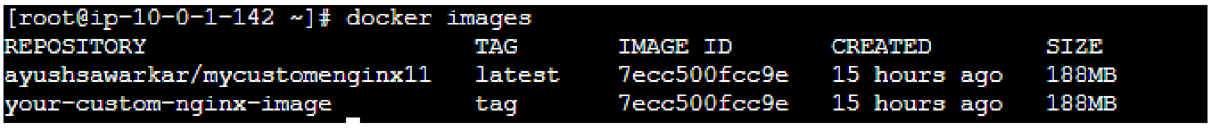
In this The latest image of nginx is been pull and inside the directory "/usr/share/nginx/html/" of the docker image my custom index.html file is been copied


```
[root@ip-10-0-1-142 ~]# cat Dockerfile
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```


```
[root@ip-10-0-1-142 ~]# cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

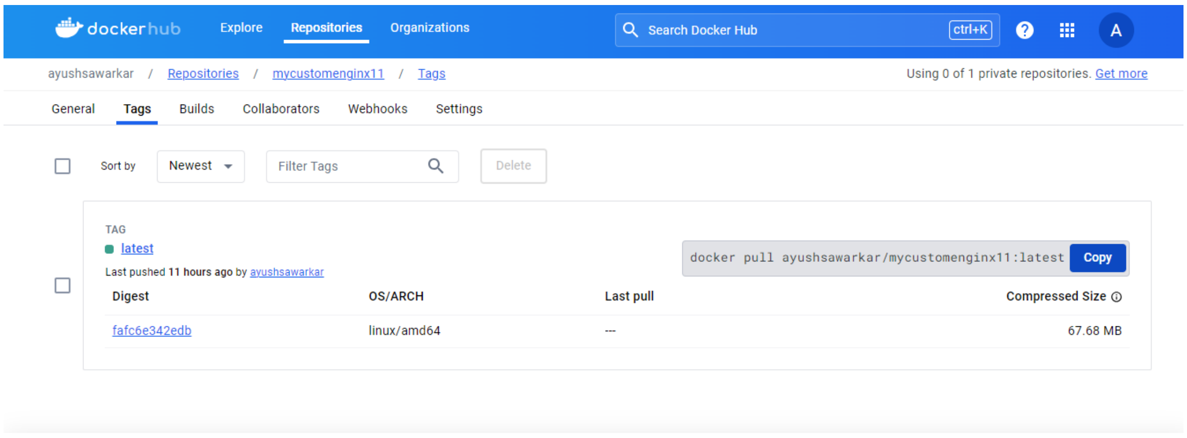
Tagging the image and pushing it to docker.hub

After creating image tagged that image with the new Docker Repository name



```
[root@ip-10-0-1-142 ~]# docker images
REPOSITORY                         TAG       IMAGE ID       CREATED          SIZE
ayushsawarkar/mycustomenginx11     latest    7ecc500fcc9e   15 hours ago     188MB
your-custom-nginx-image            tag       7ecc500fcc9e   15 hours ago     188MB
```

After creating Push that image to Docker.hub



ayushsawarkar / Repositories / mycustomenginx11 / Tags

Using 0 of 1 private repositories. Get more

General Tags Builds Collaborators Webhooks Settings

Sort by Newest Filter Tags Delete

TAG
latest
Last pushed **11 hours ago** by ayushsawarkar

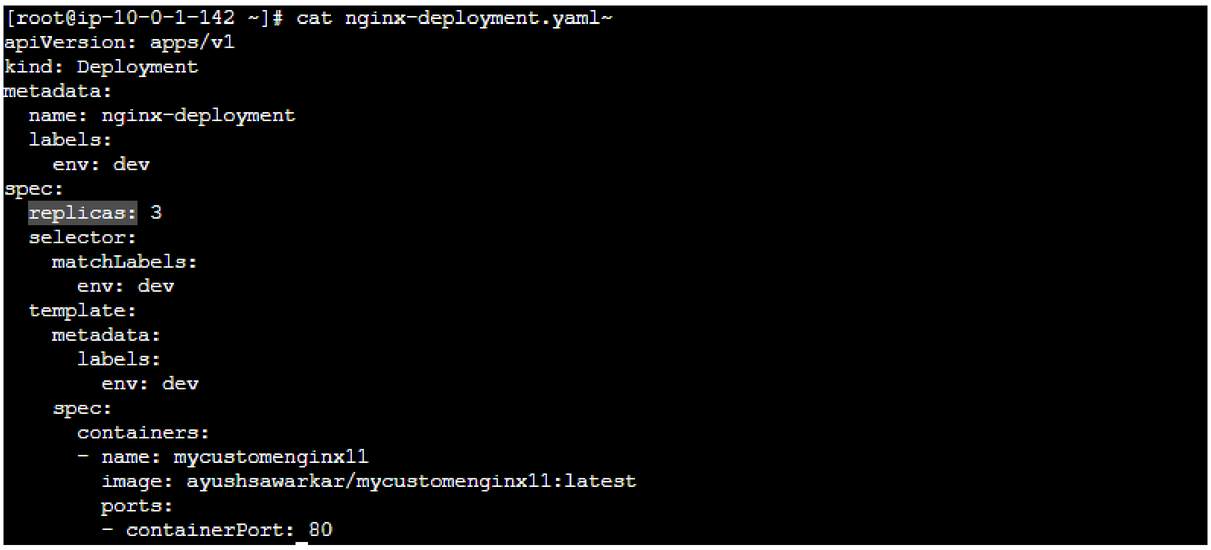
docker pull ayushsawarkar/mycustomenginx11:latest Copy

Digest	OS/ARCH	Last pull	Compressed Size
fafc6e342edb	linux/amd64	---	67.68 MB

Creating Pods

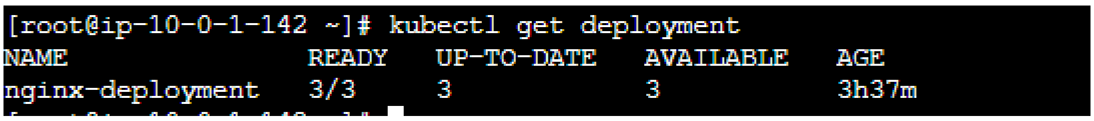
Created YAML File for the Deployment of the pods.

Where I have created a YAML File named nginx-deployment.yaml in which I'm pulling the docker image I have pushed early and giving spec of replica 3 which will deploy the 3 pods of that image.

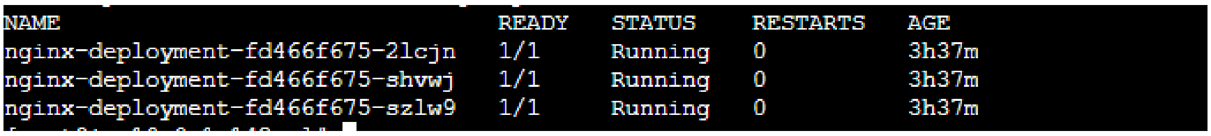


```
[root@ip-10-0-1-142 ~]# cat nginx-deployment.yaml~
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    env: dev
spec:
  replicas: 3
  selector:
    matchLabels:
      env: dev
  template:
    metadata:
      labels:
        env: dev
    spec:
      containers:
      - name: mycustomenginx11
        image: ayushsawarkar/mycustomenginx11:latest
        ports:
        - containerPort: 80
```

Checking the Deployment



```
[root@ip-10-0-1-142 ~]# kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           3h37m
```

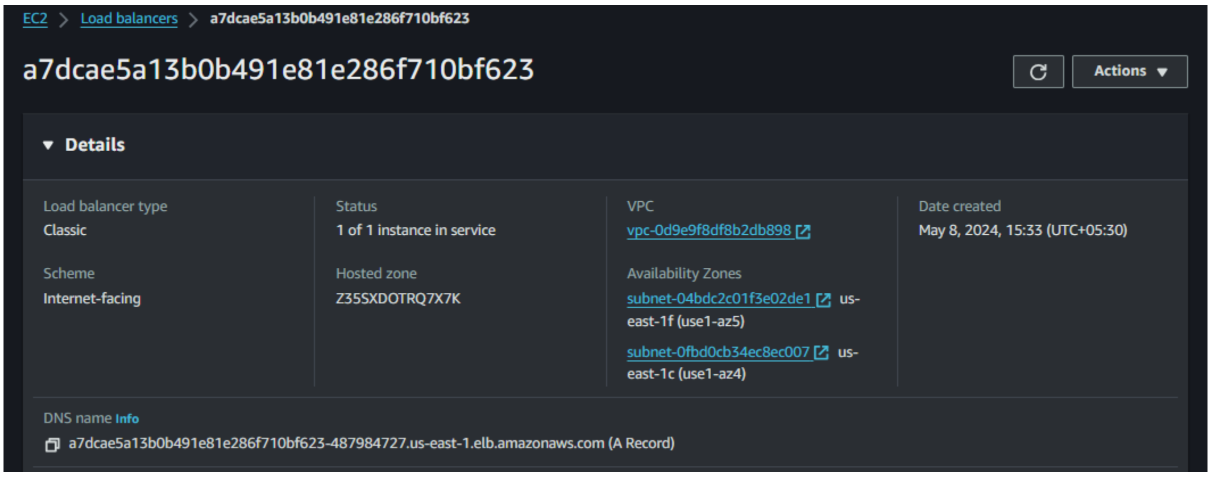


```
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-fd466f675-2lcjn   1/1     Running   0          3h37m
nginx-deployment-fd466f675-shvwj   1/1     Running   0          3h37m
nginx-deployment-fd466f675-szlw9   1/1     Running   0          3h37m
```

Pods are in running state.

Pods Testing

Hitting the DNS of load Balancer

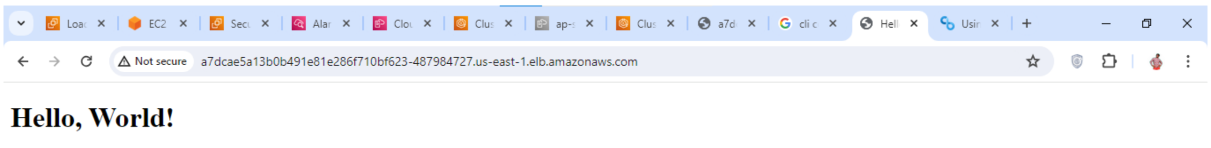


EC2 > Load balancers > a7dcae5a13b0b491e81e286f710bf623

a7dcae5a13b0b491e81e286f710bf623

Details

Load balancer type: Classic
Scheme: Internet-facing
Status: 1 of 1 instance in service
Hosted zone: Z35SXDOTRQ7X7K
VPC: vpc-0d9e9f8df8b2db898
Availability Zones: subnet-04bdc2c01f3e02de1 us-east-1f (use1-az5); subnet-0fbd0cb34ec8ec007 us-east-1c (use1-az4)
Date created: May 8, 2024, 15:33 (UTC+05:30)
DNS name Info: a7dcae5a13b0b491e81e286f710bf623-487984727.us-east-1.elb.amazonaws.com (A Record)



Not secure a7dcae5a13b0b491e81e286f710bf623-487984727.us-east-1.elb.amazonaws.com

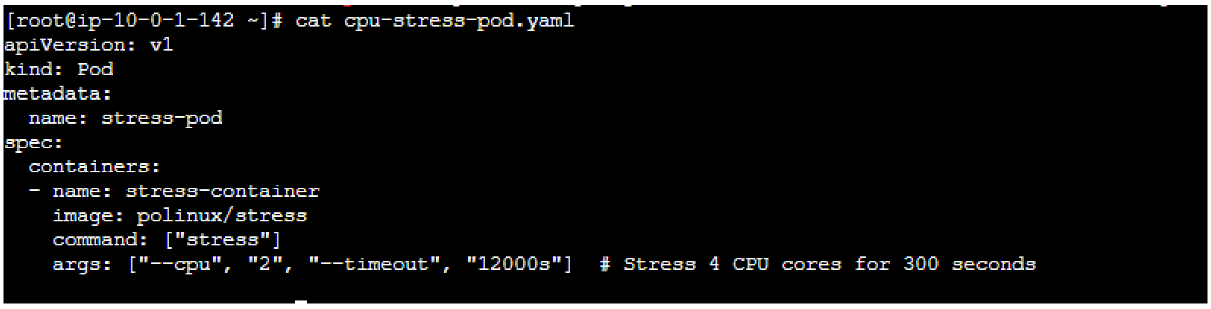
Hello, World!

Here The pods are running fine as the pods are created using the Docker hub.

Testing

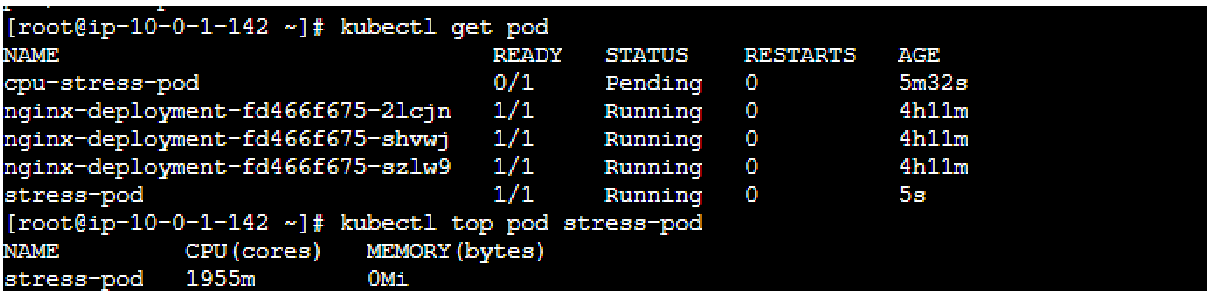
Add load On server

I have created a YAML File to deploy a Stress-Pod on the server which can increase the load on the instance.



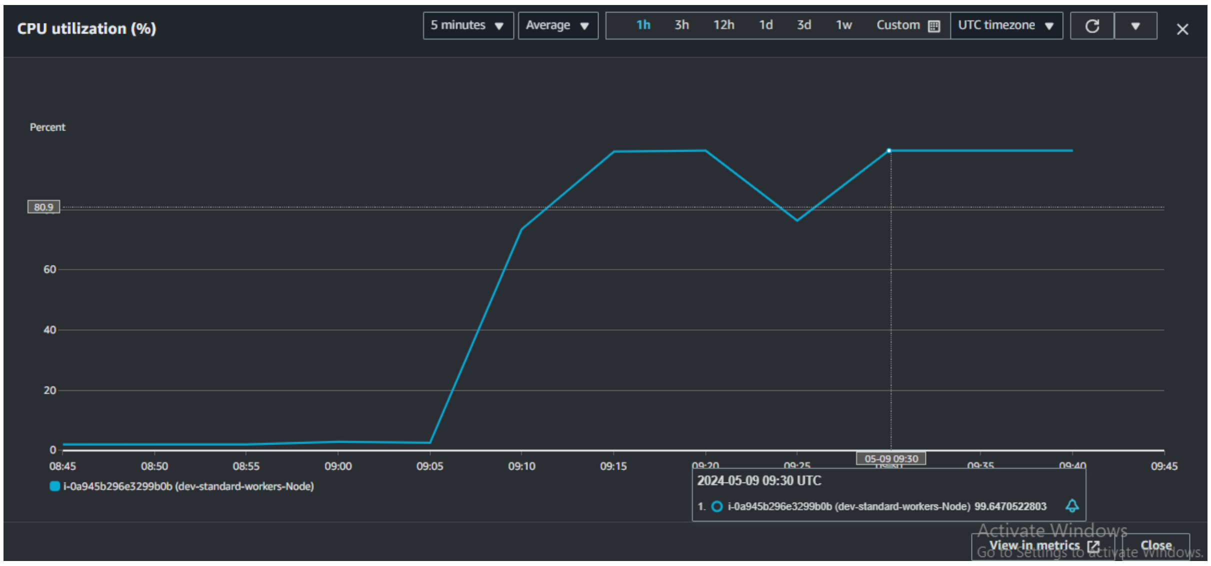
```
[root@ip-10-0-1-142 ~]# cat cpu-stress-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: stress-pod
spec:
  containers:
  - name: stress-container
    image: polinux/stress
    command: ["stress"]
    args: ["--cpu", "2", "--timeout", "12000s"]  # Stress 4 CPU cores for 300 seconds
```

Deployed the same thing and check the stress on the instance.



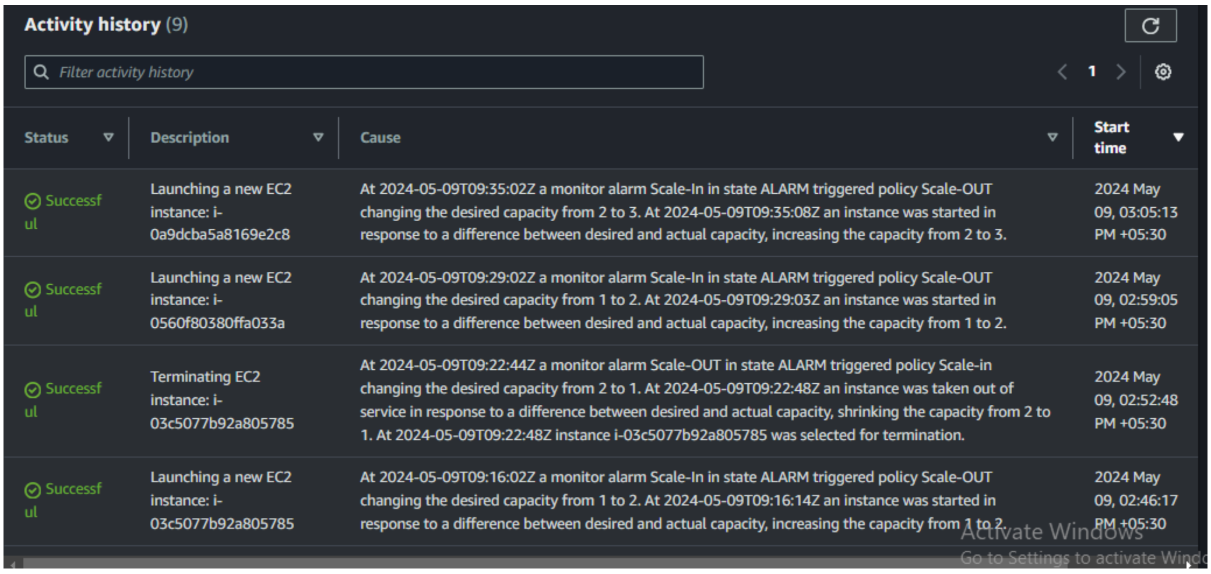
```
[root@ip-10-0-1-142 ~]# kubectl get pod
NAME                              READY   STATUS    RESTARTS   AGE
cpu-stress-pod                    0/1     Pending   0          5m32s
nginx-deployment-fd466f675-2lcjn  1/1     Running   0          4h11m
nginx-deployment-fd466f675-shvwj  1/1     Running   0          4h11m
nginx-deployment-fd466f675-szlw9  1/1     Running   0          4h11m
stress-pod                        1/1     Running   0          5s
[root@ip-10-0-1-142 ~]# kubectl top pod stress-pod
NAME         CPU(cores)   MEMORY(bytes)
stress-pod   1955m        0Mi
```

If Few minutes the serve CPU Utilisation has been increased



2024-05-09 09:30 UTC
1. i-0a945b296e3299b0b (dev-standard-workers-Node) 99.6470522803

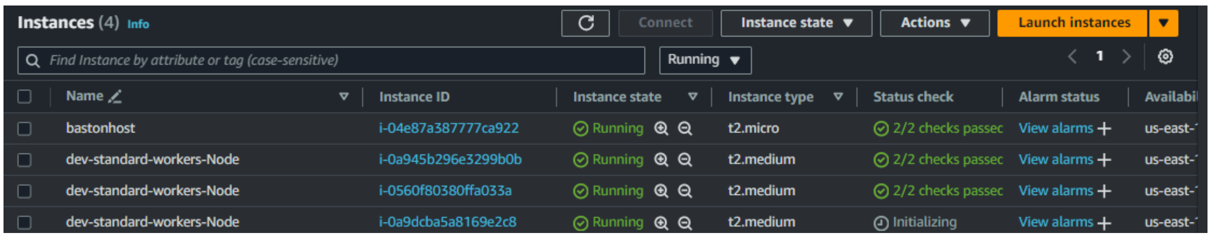
Here The auto scaling Has been triggered and it started to Spinning up new server to handle the load.



Activity history (9)

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-0a9dcba5a8169e2c8	At 2024-05-09T09:35:02Z a monitor alarm Scale-In in state ALARM triggered policy Scale-OUT changing the desired capacity from 2 to 3. At 2024-05-09T09:35:08Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.	2024 May 09, 03:05:13 PM +05:30
Successful	Launching a new EC2 instance: i-0560f80380ffa033a	At 2024-05-09T09:29:02Z a monitor alarm Scale-In in state ALARM triggered policy Scale-OUT changing the desired capacity from 1 to 2. At 2024-05-09T09:29:03Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2024 May 09, 02:59:05 PM +05:30
Successful	Terminating EC2 instance: i-03c5077b92a805785	At 2024-05-09T09:22:44Z a monitor alarm Scale-OUT in state ALARM triggered policy Scale-in changing the desired capacity from 2 to 1. At 2024-05-09T09:22:48Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2024-05-09T09:22:48Z instance i-03c5077b92a805785 was selected for termination.	2024 May 09, 02:52:48 PM +05:30
Successful	Launching a new EC2 instance: i-03c5077b92a805785	At 2024-05-09T09:16:02Z a monitor alarm Scale-In in state ALARM triggered policy Scale-OUT changing the desired capacity from 1 to 2. At 2024-05-09T09:16:14Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2024 May 09, 02:46:17 PM +05:30

2 Extra server has been Spined up.

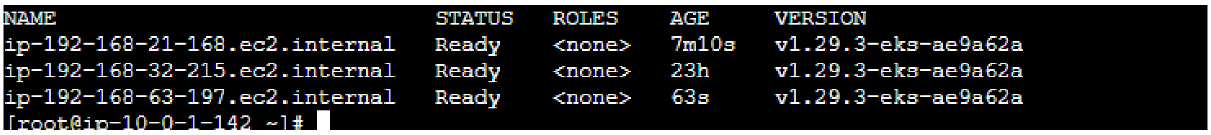


Instances (4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabi
bastonhost	i-04e87a387777ca922	Running	t2.micro	2/2 checks passec	View alarms	us-east-
dev-standard-workers-Node	i-0a945b296e3299b0b	Running	t2.medium	2/2 checks passec	View alarms	us-east-
dev-standard-workers-Node	i-0560f80380ffa033a	Running	t2.medium	2/2 checks passec	View alarms	us-east-
dev-standard-workers-Node	i-0a9dcba5a8169e2c8	Running	t2.medium	Initializing	View alarms	us-east-

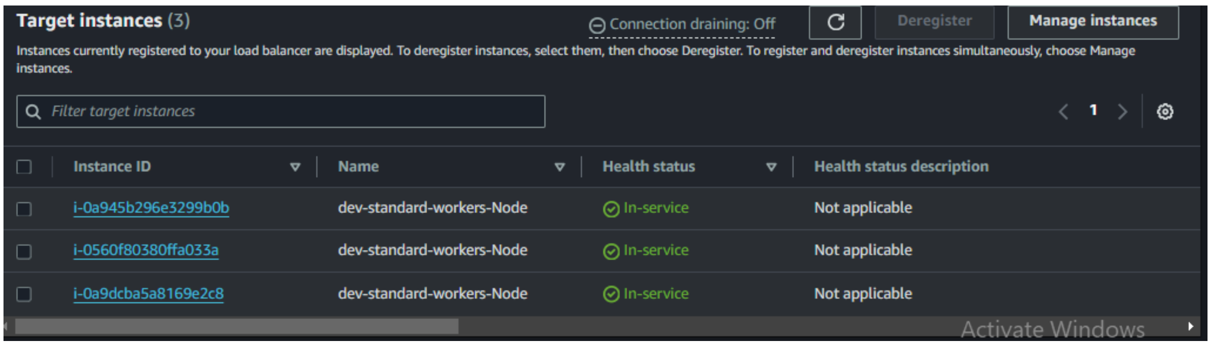
3 Nodes are Present in Cluster:

Here in the Cluster And node balancer the instances are staring showing as shown in the Screen shot 3 server are present in the EKS Cluster.



```
NAME                             STATUS   ROLES    AGE    VERSION
ip-192-168-21-168.ec2.internal   Ready    <none>   7m10s  v1.29.3-eks-ae9a62a
ip-192-168-32-215.ec2.internal   Ready    <none>   23h    v1.29.3-eks-ae9a62a
ip-192-168-63-197.ec2.internal   Ready    <none>   63s    v1.29.3-eks-ae9a62a
[root@ip-10-0-1-142 ~]#
```

Load Balancer also automatically attached the new Spinned Up servers.



Target instances (3)

Connection draining: Off

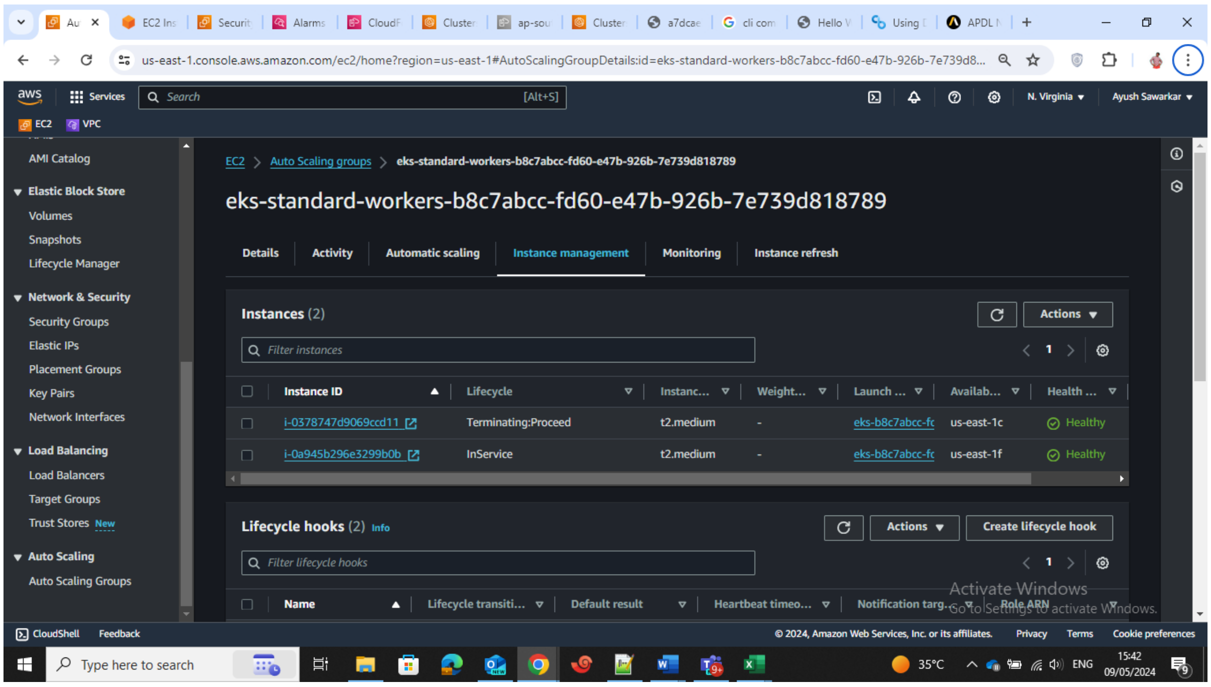
Instances currently registered to your load balancer are displayed. To deregister instances, select them, then choose Deregister. To register and deregister instances simultaneously, choose Manage instances.

Instance ID	Name	Health status	Health status description
i-0a945b296e3299b0b	dev-standard-workers-Node	In-service	Not applicable
i-0560f80380ffa033a	dev-standard-workers-Node	In-service	Not applicable
i-0a9dcba5a8169e2c8	dev-standard-workers-Node	In-service	Not applicable

From this We can say that the EKS has Evenly Distributed the pods which were running on the single instance after the new instances addition the EKS Has maintain them Evenly on every server.

Deleted Stress-Pod

After Deleting the pod which was increaing the CPU Utilization is less than it started to terminate the instances



EC2 > Auto Scaling groups > eks-standard-workers-b8c7abcc-fd60-e47b-926b-7e739d818789

eks-standard-workers-b8c7abcc-fd60-e47b-926b-7e739d818789

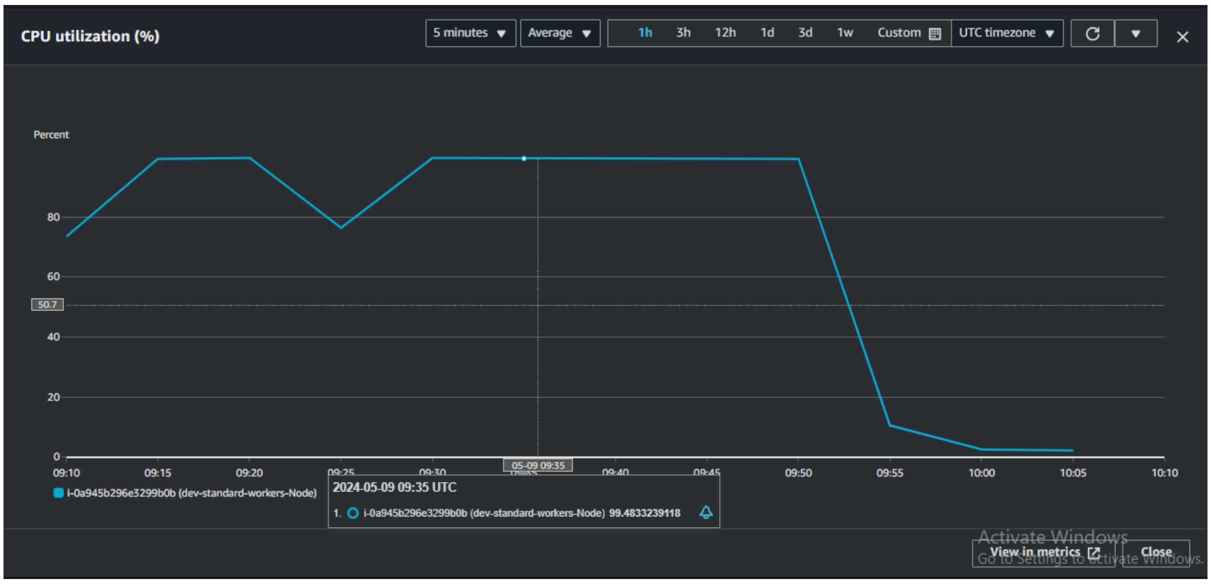
Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Instances (2)

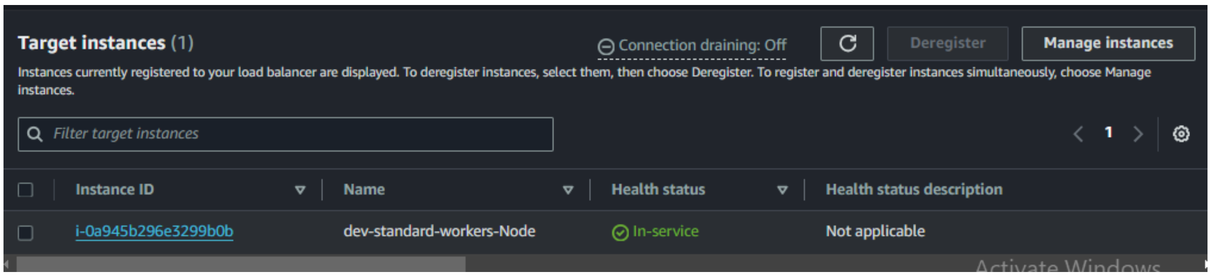
Instance ID	Lifecycle	Instanc...	Weight...	Launch ...	Availab...	Health ...
i-0378747d9069ccd11	Terminating:Proceed	t2.medium	-	eks-b8c7abcc-fc	us-east-1c	Healthy
i-0a945b296e3299b0b	InService	t2.medium	-	eks-b8c7abcc-fc	us-east-1f	Healthy

Lifecycle hooks (2) Info

Name	Lifecycle transiti...	Default result	Heartbeat timeo...	Notification targ...	Role ARN



Load Balancer is now targeting only one instance again.



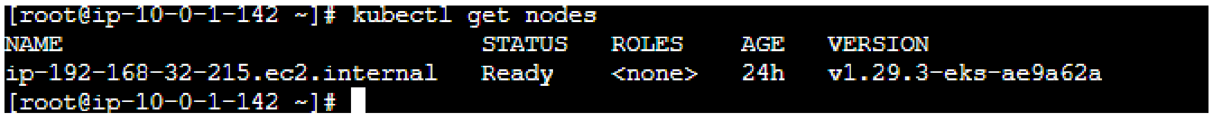
Target instances (1)

Connection draining: Off | Deregister | Manage instances

Instances currently registered to your load balancer are displayed. To deregister instances, select them, then choose Deregister. To register and deregister instances simultaneously, choose Manage instances.

Instance ID	Name	Health status	Health status description
i-0a945b296e3299b0b	dev-standard-workers-Node	In-service	Not applicable

Here we can see that the main server is only there and all other server which were spined up was terminated successfully.

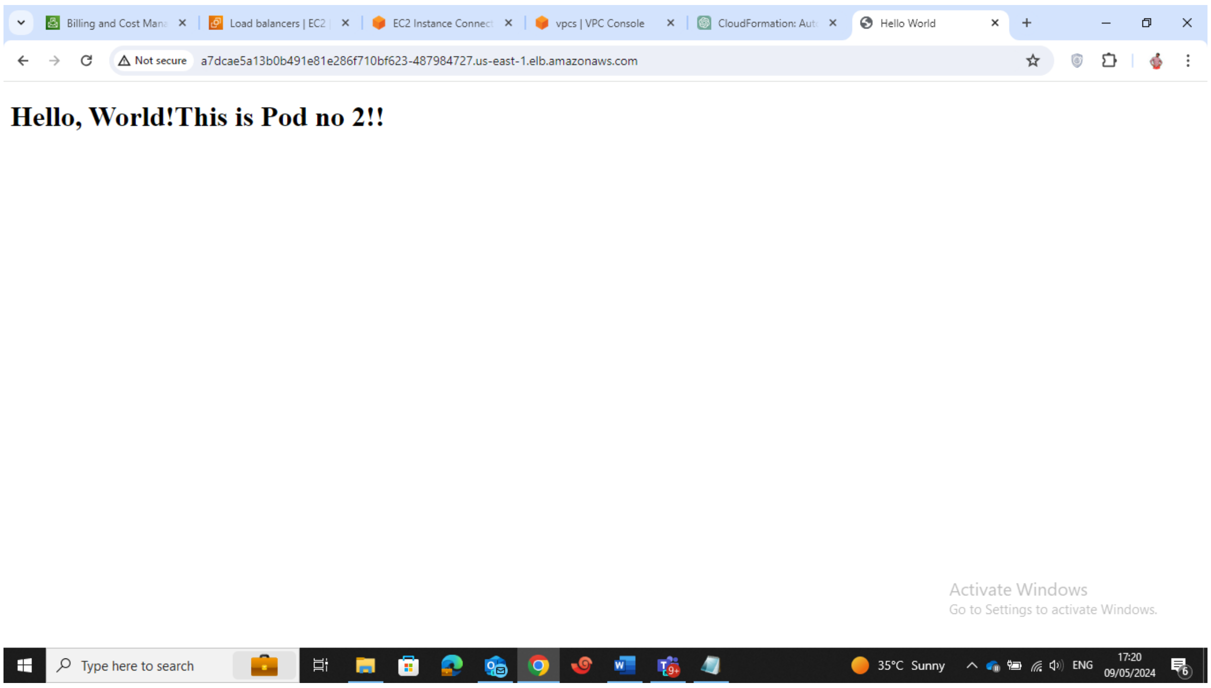


```
[root@ip-10-0-1-142 ~]# kubectl get nodes
NAME                             STATUS   ROLES    AGE   VERSION
ip-192-168-32-215.ec2.internal   Ready    <none>   24h   v1.29.3-eks-ae9a62a
[root@ip-10-0-1-142 ~]#
```

Testing all pods are working or

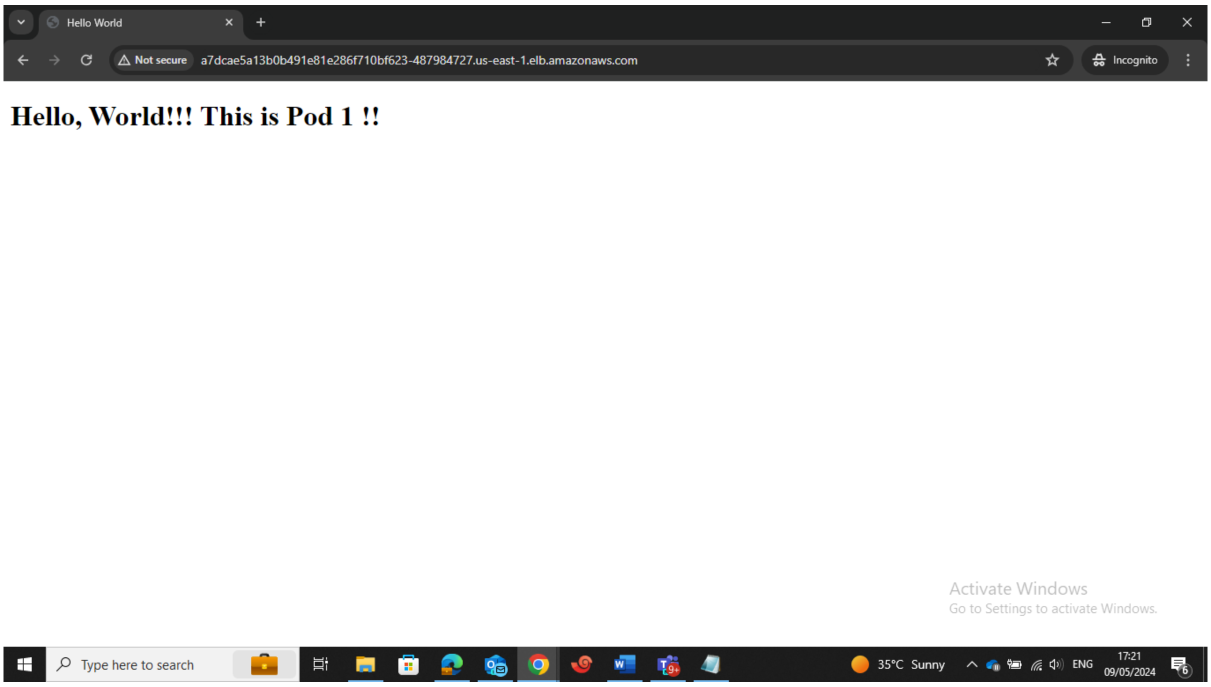
For this I have login to each pod and make changes into the index.html file at /usr/share/nginx/html/index.html this location.

And after hitting the Load Balancer DNS everything is working fine.



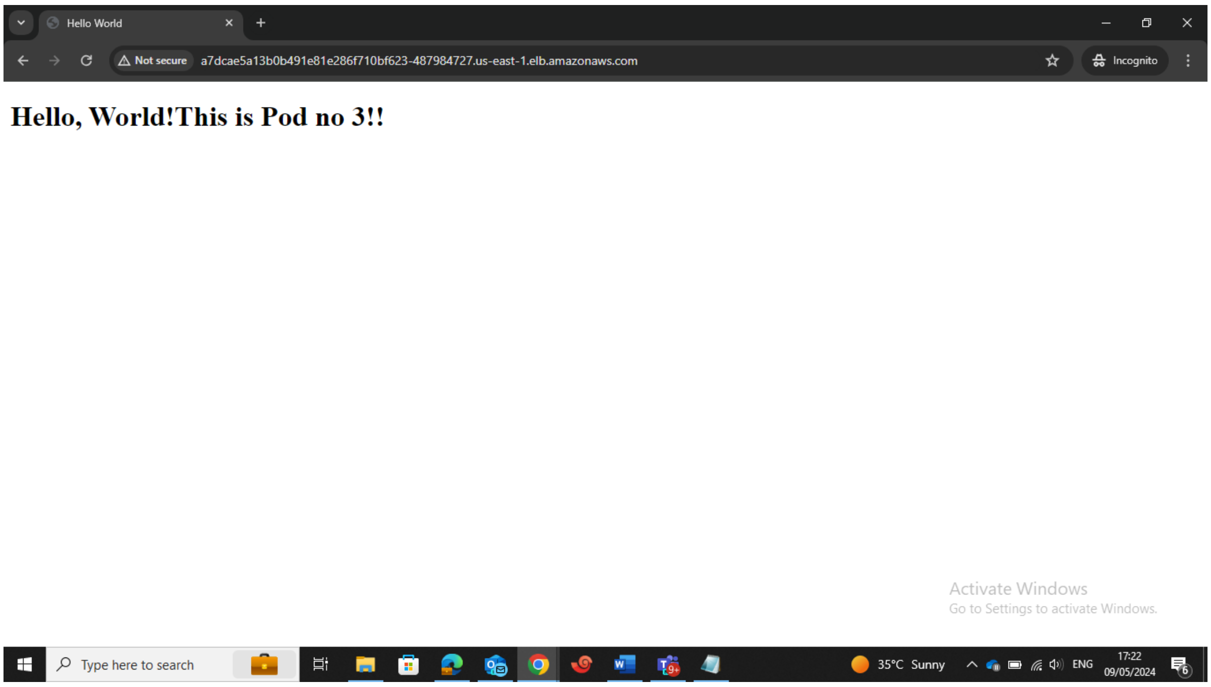
Not secure a7dcae5a13b0b491e81e286f710bf623-487984727.us-east-1.elb.amazonaws.com

Hello, World!This is Pod no 2!!



Not secure a7dcae5a13b0b491e81e286f710bf623-487984727.us-east-1.elb.amazonaws.com

Hello, World!!! This is Pod 1 !!


Hello World
Not secure
a7dcae5a13b0b491e81e286f710bf623-487984727.us-east-1.elb.amazonaws.com
Incognito

Hello, World!This is Pod no 3!!

Activate Windows
Go to Settings to activate Windows.
Type here to search
35°C Sunny
ENG
17:22
09/05/2024