INTRODUCTION TO AWS CODEDEPLOY AND CODE PIPILINE

CODE PIPELINE:

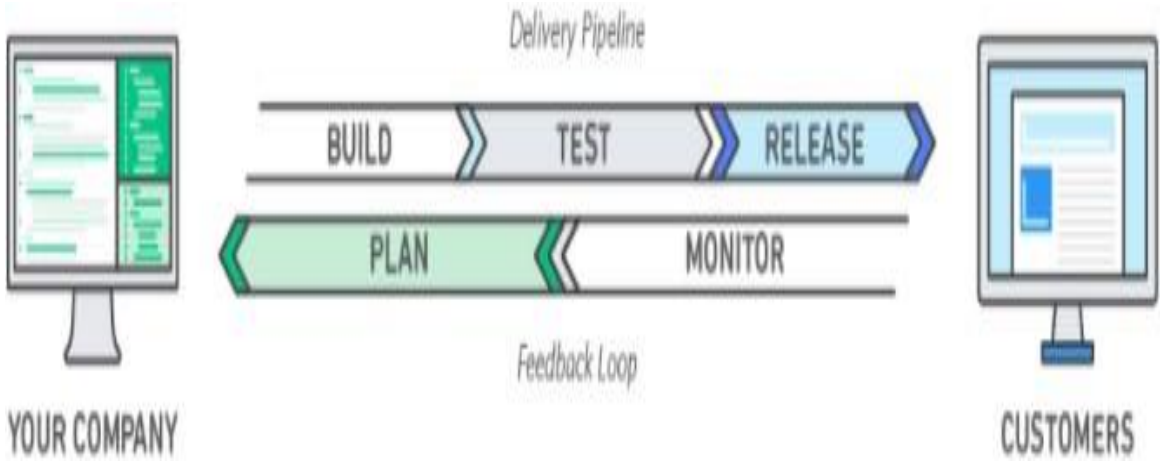
AWS CodePipeline is a continuous delivery service that enables you to model, visualize, and automate the steps required to release your software. AWS CodePipeline then builds, tests, and deploys your application according to the defined workflow every time there is a code change.

CODE DEPLOY:

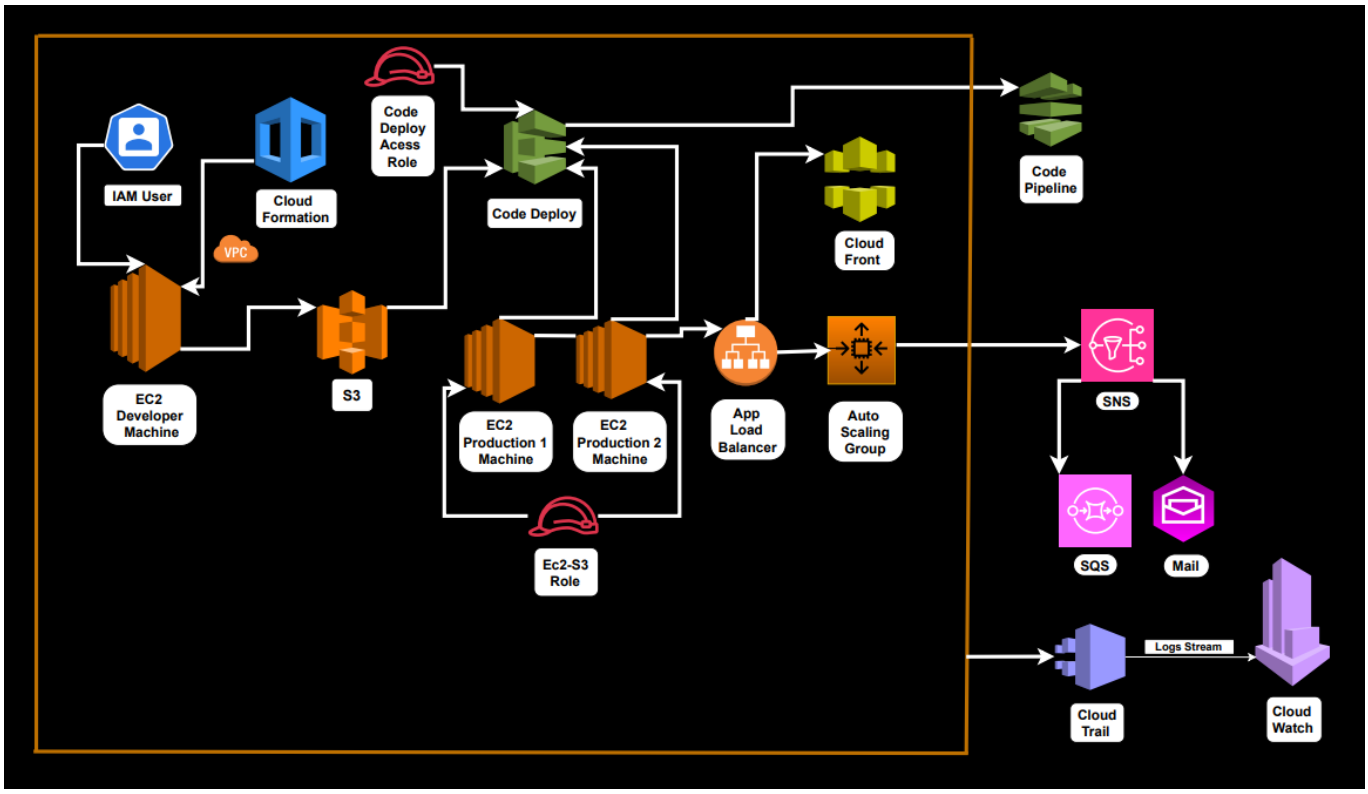
AWS CodeDeploy is a service that automates code deployments to Elastic Compute Cloud (EC2) and on-premises servers. Accelerating how fast a developer can release code allows him to release new features for an application faster and avoid deployment errors in complex applications.

OBJECTIVE TO THE AWS CODE DEPLOY AND CODE PIPELINE:

The objective of this project is to develop and deploy a dynamic web application using a microservice architecture, with a focus on front-end development utilizing HTML and CSS. The project aims path based routing using App Loadbalancer and CloudFront for Content Delivery Network (CDN) to deliver a website. The goal is to manage and scale the web application efficiently, ensuring high availability, security, performance optimization, and automated CI/CD with AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline.



CODE DEPLOY AND CODE PIPELINE PROJECT:



CODE DEPLOY AND CODE PIPELINE PROJECT STEPS:

CodeDeploy is a deployment service from AWS which can automate application deployments to Amazon EC2 instances, on-premises instances or Lambda functions. This does a onetime deployment, for scheduling of deployment you may have to use AWS CodePipeline also.

Application: A CodeDeploy application can be defined from AWS CodeDeploy web console.

Revision: Represents the code need to be deployed on EC2 instance

Appspec file: This contains the instruction to CodeDeploy, like copying of files, executing the scripts etc during the code deployment process. It is present in the root directory of unzipped code with

Deployment Group: Represent set of machines of Lambda function where code has to be deployed

Deployment: The process of deployment.

Setup in Brief:

I have used two EC2 instance of AMZ2 Linux. First one is the web server we will be configuring, also called CodeDeploy agent. Second and Third EC2 machine is supposed to use by developer where the codes are programmed. The names of the resources in the experiment are arbitrary and may name the resources your own

1. Create **IAM Roles** for EC2-S3 & CodeDeploy access.
2. Create **IAM user** account for Developer Machine.
3. Create **VPC** using with help of **Cloud Formation** service.
4. Create 3 **EC2** instances – 1 Developer Machine & 2 Production Machine.
5. Create **SQS** & **SNS** for Communication service.
6. Create **Cloud Trail** for Logs that streams on **Cloud Watch**.
7. Create **S3 Bucket** for store a website files.
8. Connect IAM user in Developer Machine **CLI.**
9. Install and prepare the Code Deploy agent on webserver in Production Machine.
10. Create the code from Developer Machine.
11. Create **Code Deploy** Application and Push the code to S3 bucket from Developer machine.
12. Create Deployment Group to include webserver.
13. Create Deployment to push the code to the webserver.
14. Create **App Load Balancer** for path based routing and **Auto-Scaling Group**.
15. Create **Cloud Front** with App LB.
16. Browse the Website.
17. Create **Code Pipeline** S3 bucket.
18. Re-edit the website in Developer Machine.
19. Copy website file to S3 bucket.
20. Refresh the website changes will show.

AWS SERVICES USED :

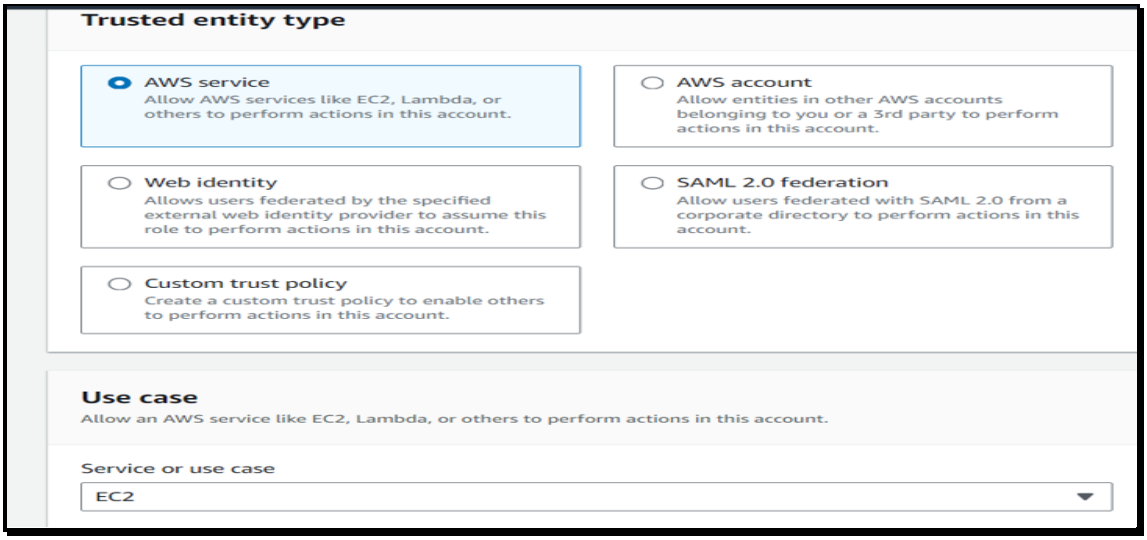
IAM, VPC, Cloud Formation, EC2, SQS, SNS, Cloud Trail, Cloud Watch, S3, Code Deploy, App Load Balancer & Auto Scaling Group, Cloud Front, Code Pipeline.

Step 1 : Create **IAM Roles** for EC2-S3 & Code Deploy access.

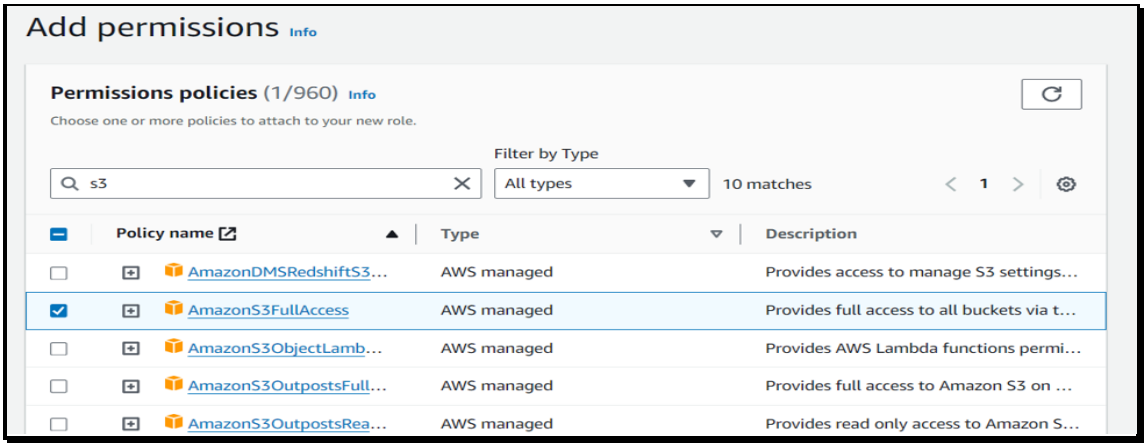
a) EC2 – S3 full access :

IAM Service

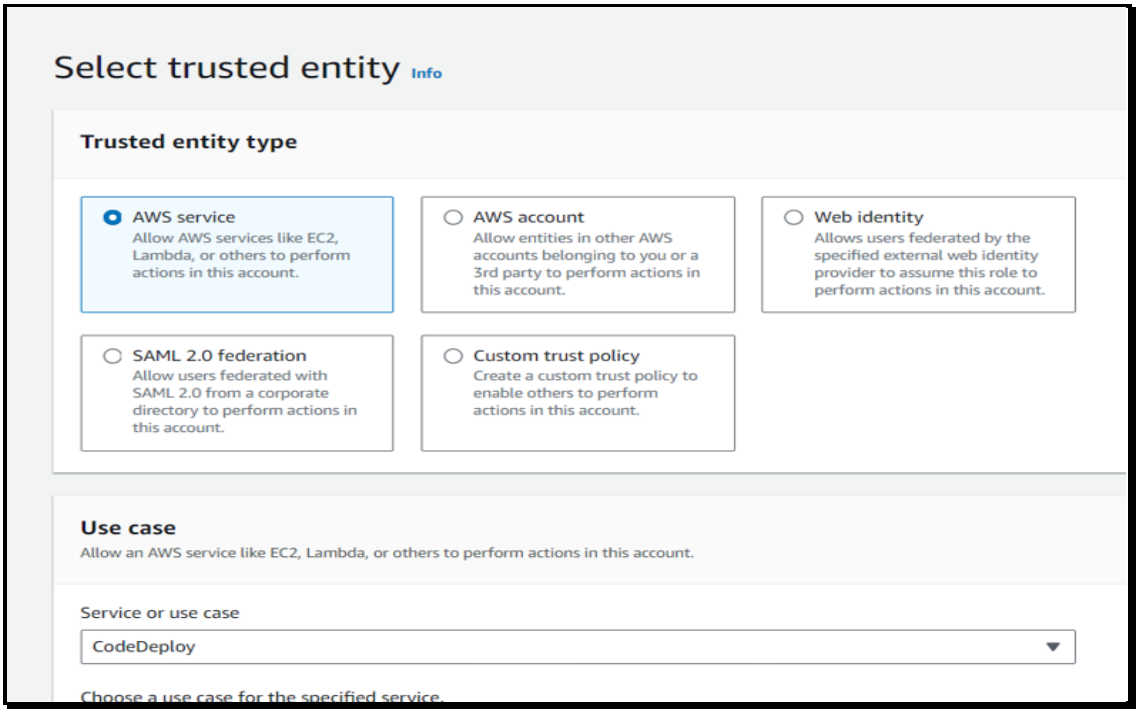
Roles: Select EC2 UseCase



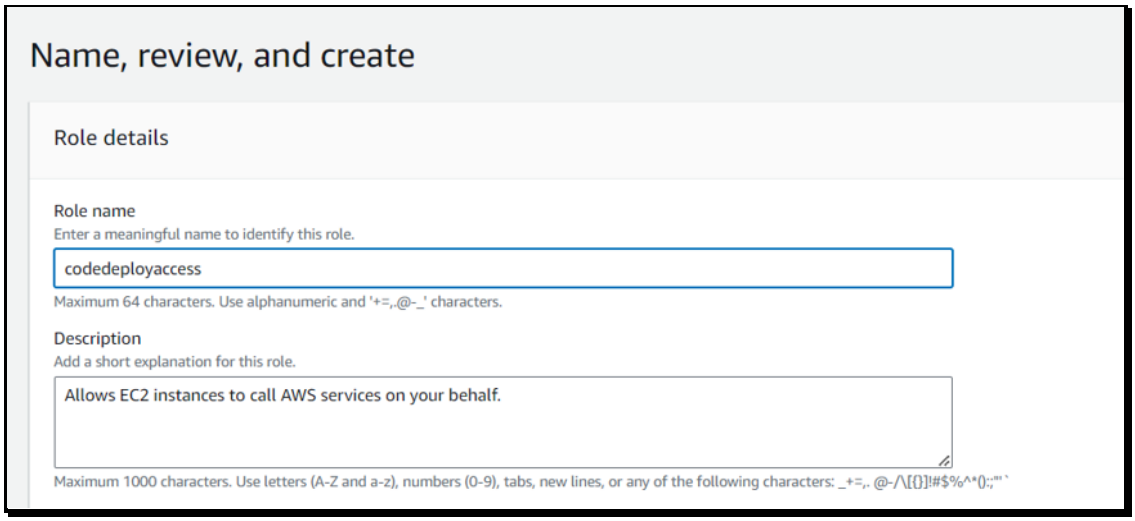
Then select S3 full access policy



b) Create an Code-Deploy role:
CodeDeploy UseCase

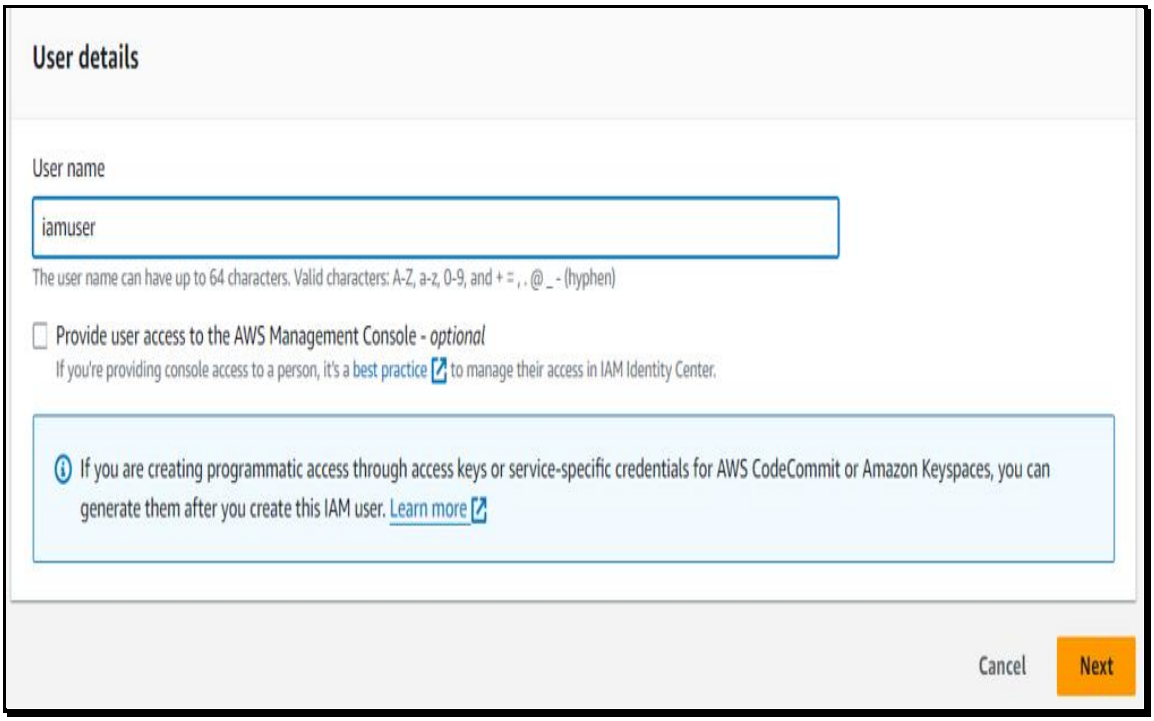


Attach a permission policy

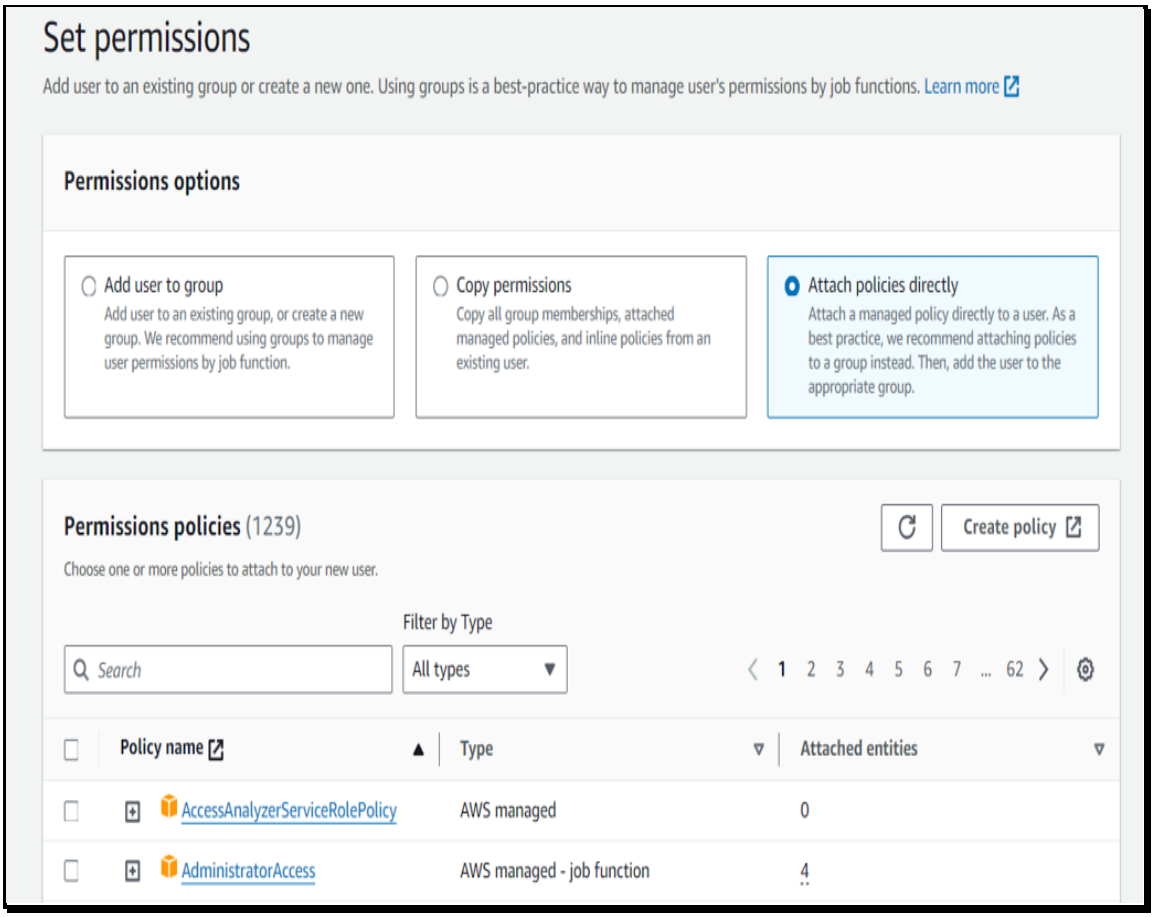


Step 2: Create **IAM user** account for Developer Machine

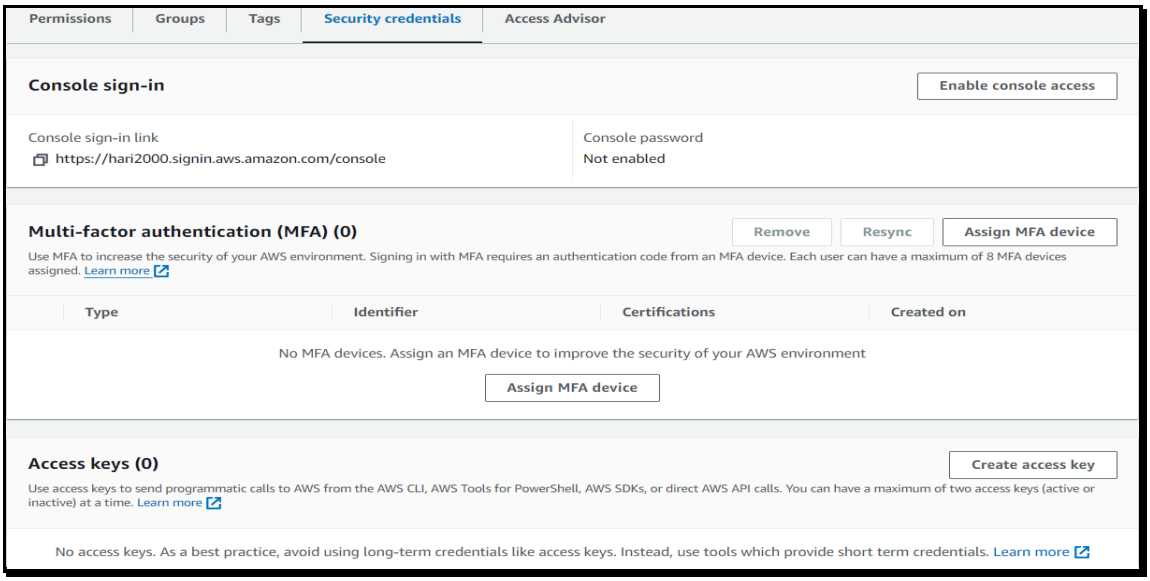
Goto users and add user

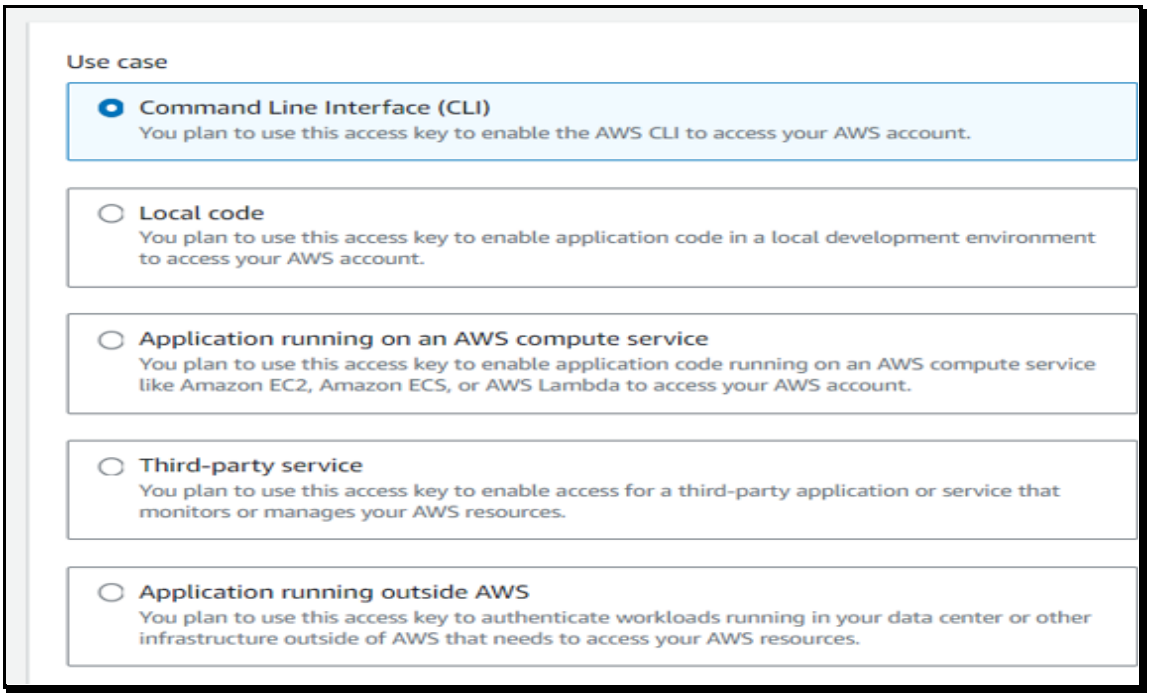


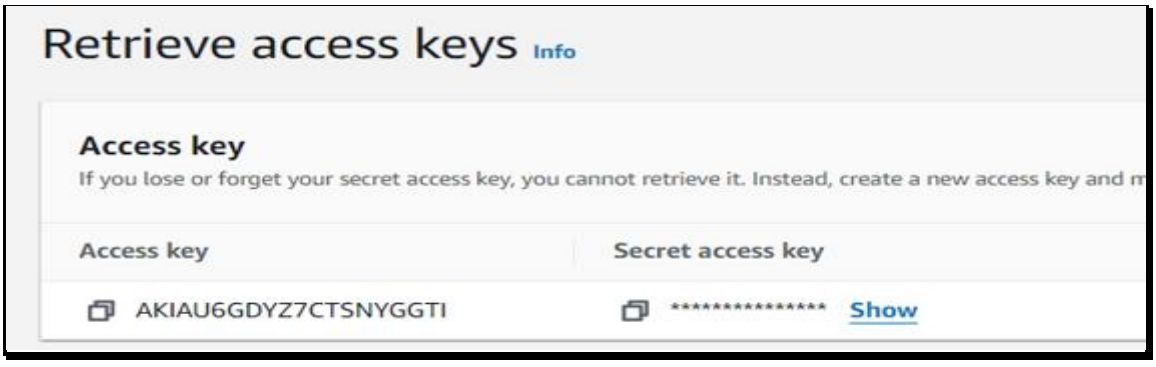
Add permission



Create Access key

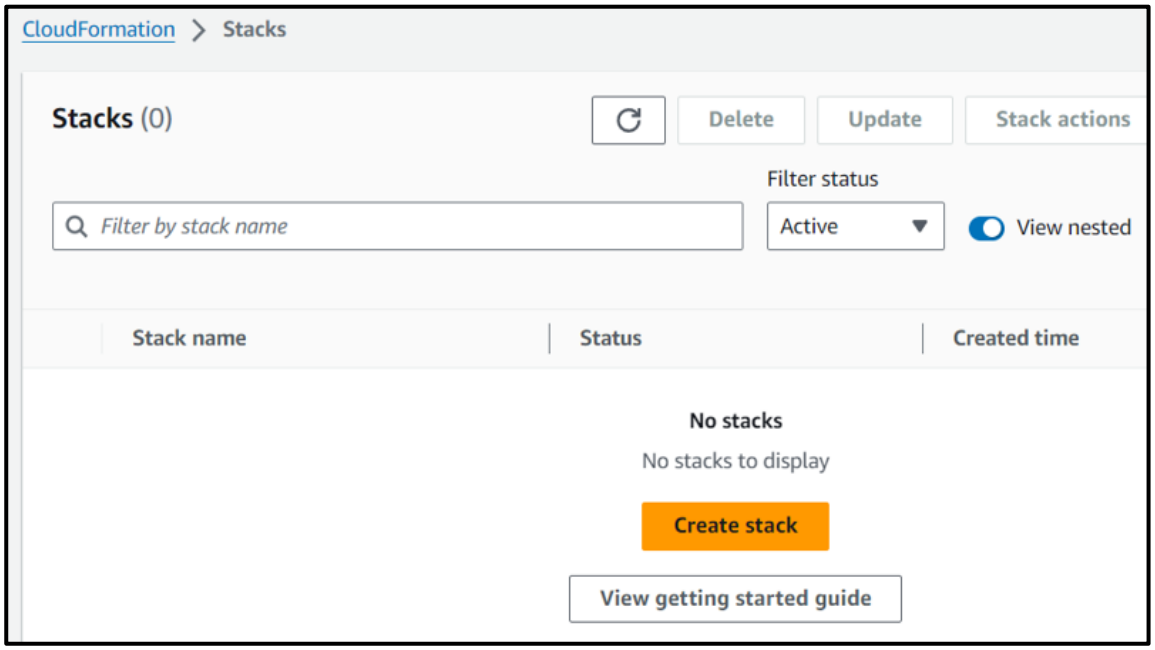




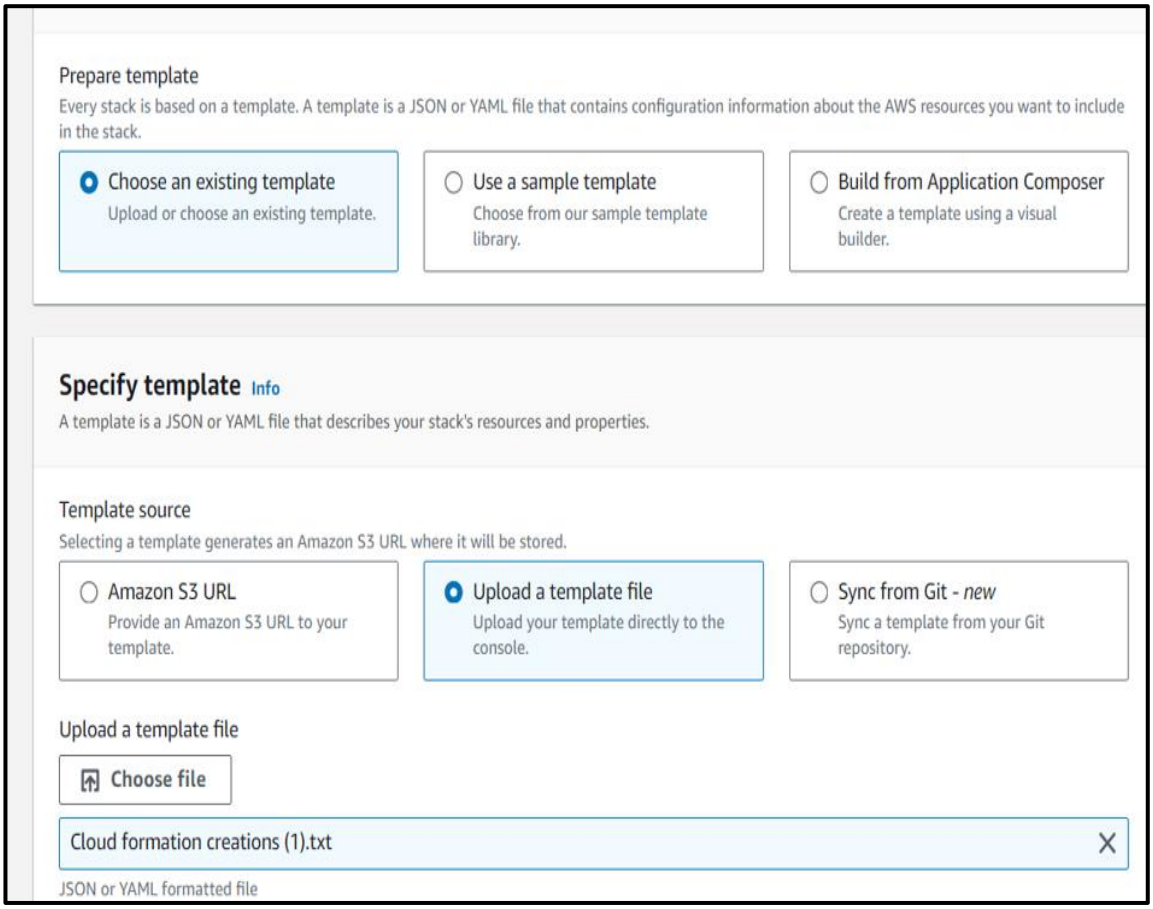


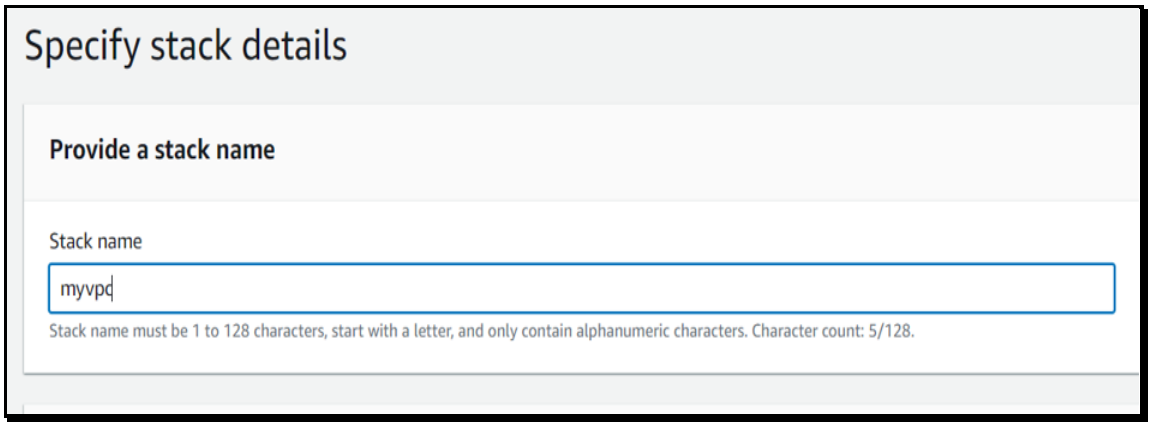
Step 3 : Create **VPC** using with help of **Cloud Formation** service.

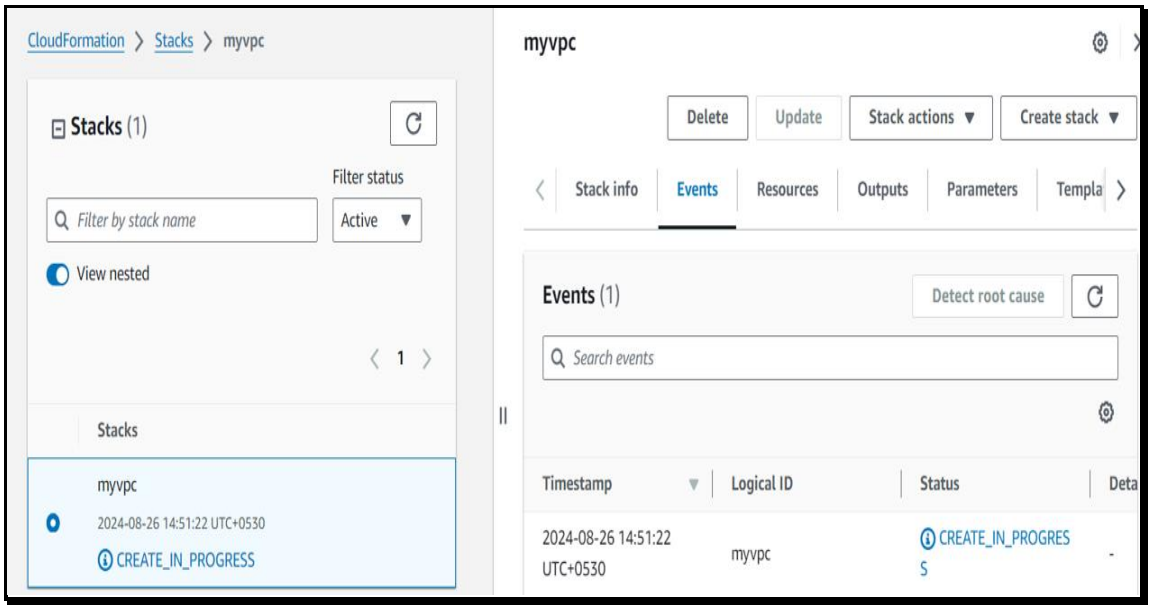
Create a Stack



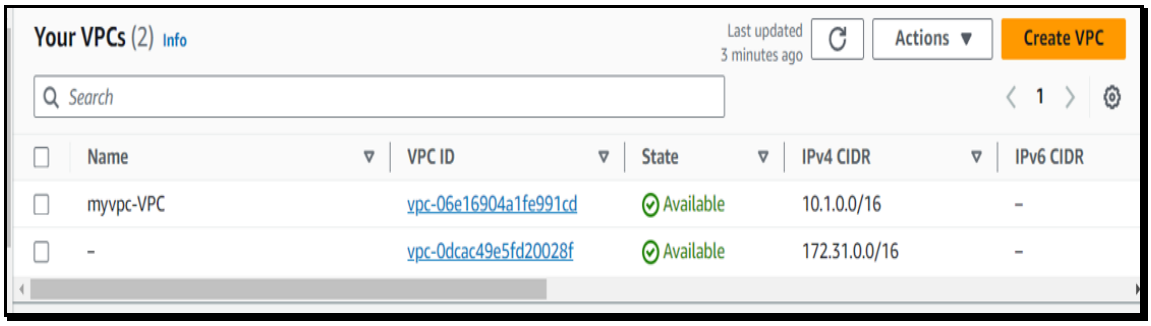
Upload a Yaml Code for creating VPC

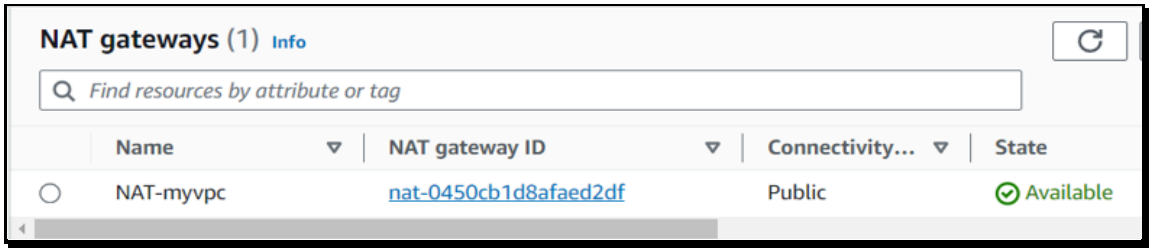






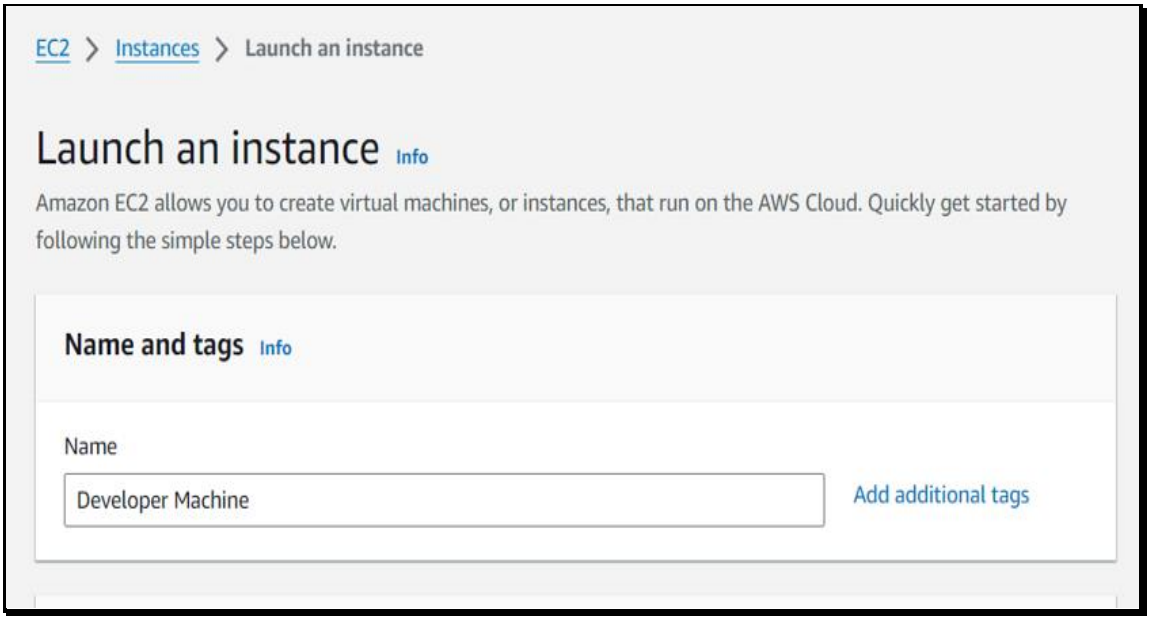
VPC was created

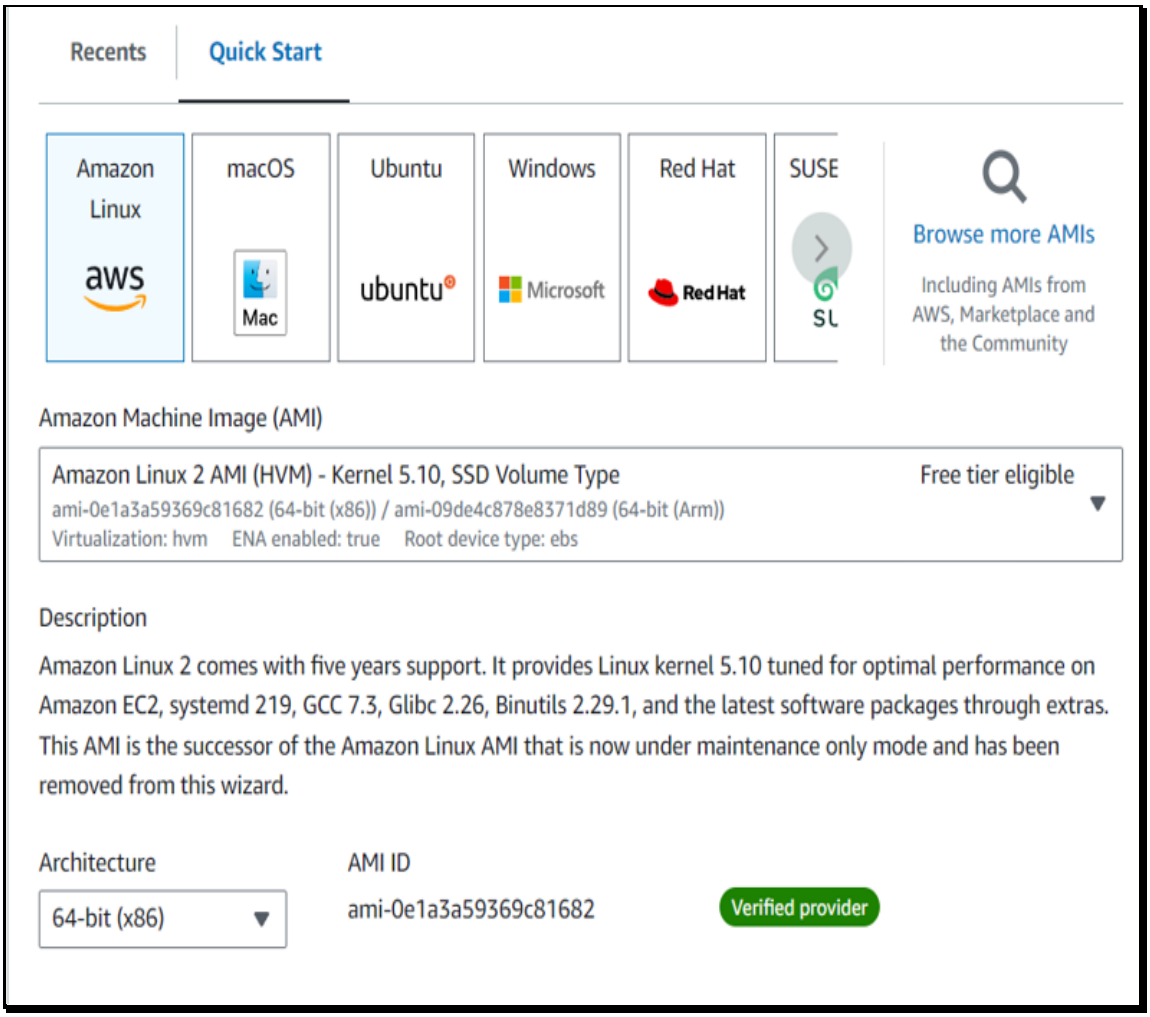


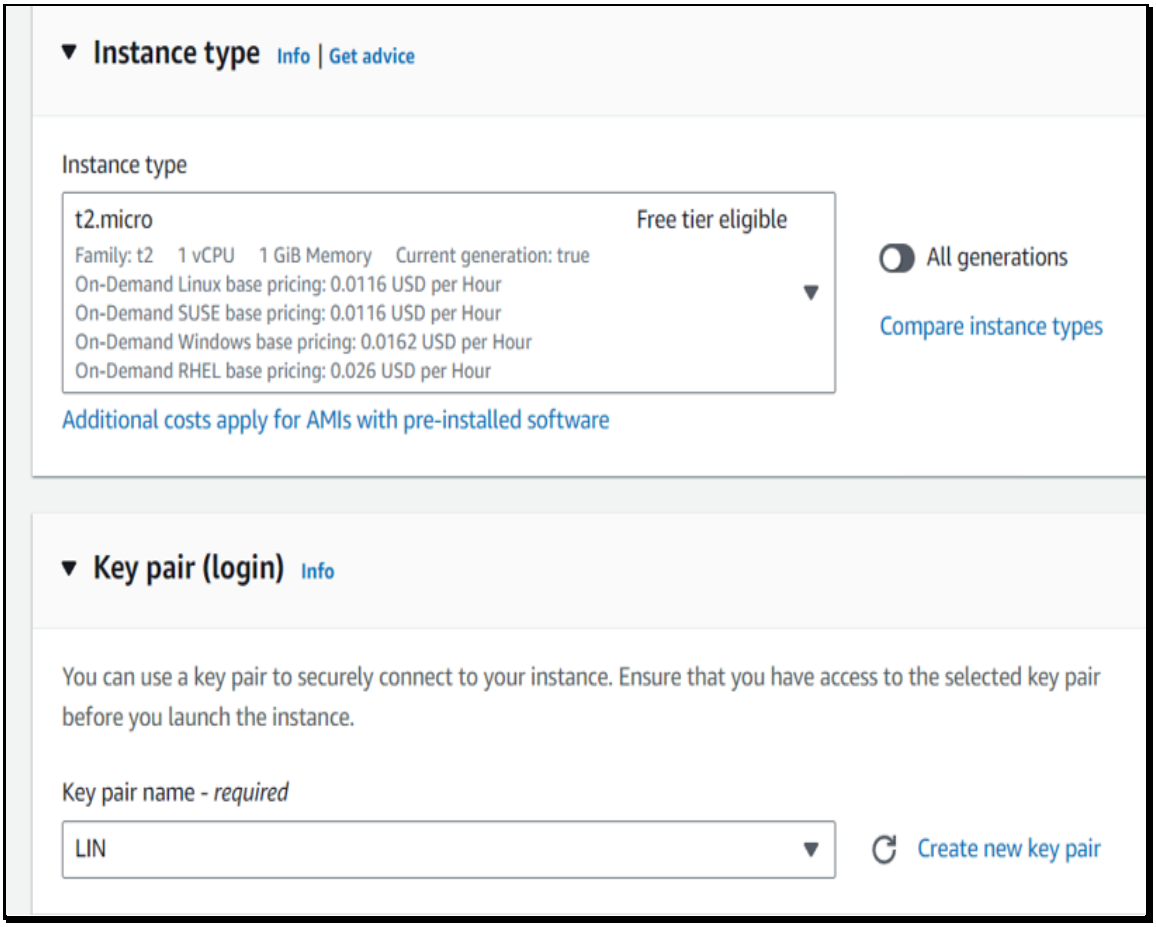


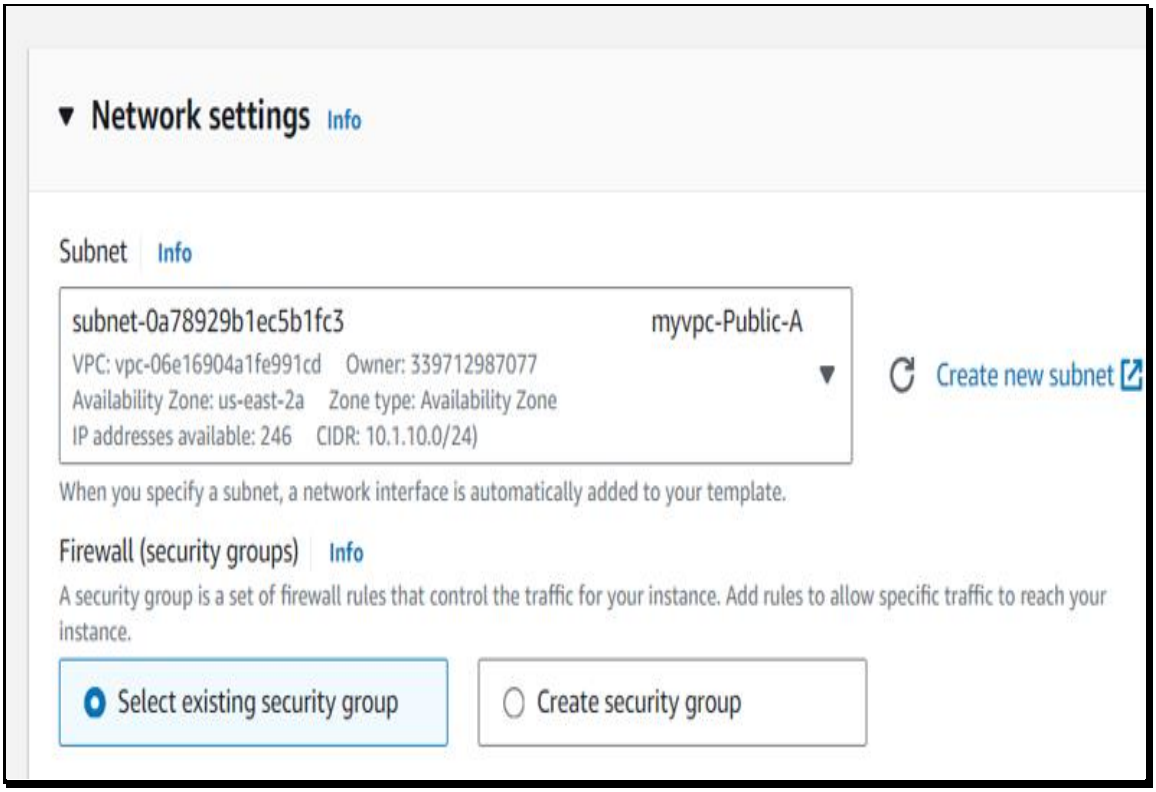
Step 4 : Create 3 **EC2** instances – 1 Developer Machine & 2 Production Machine.

a) Create a 1 instance for Developer Machine

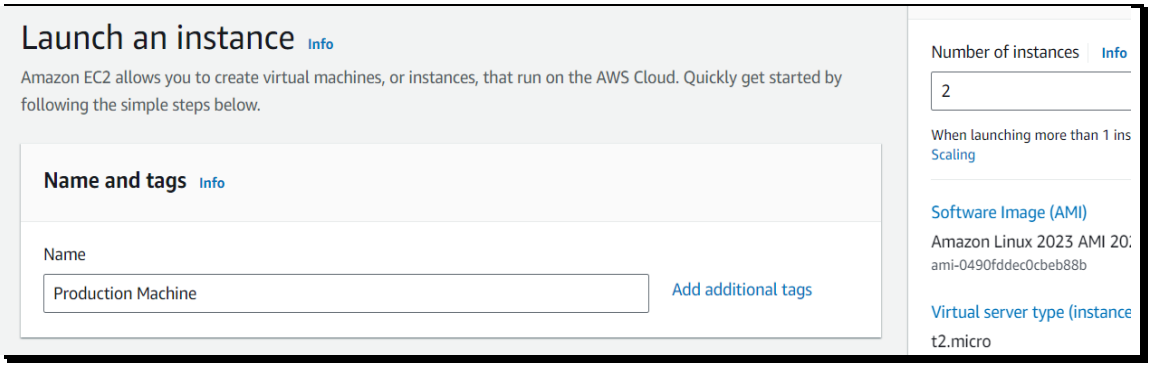


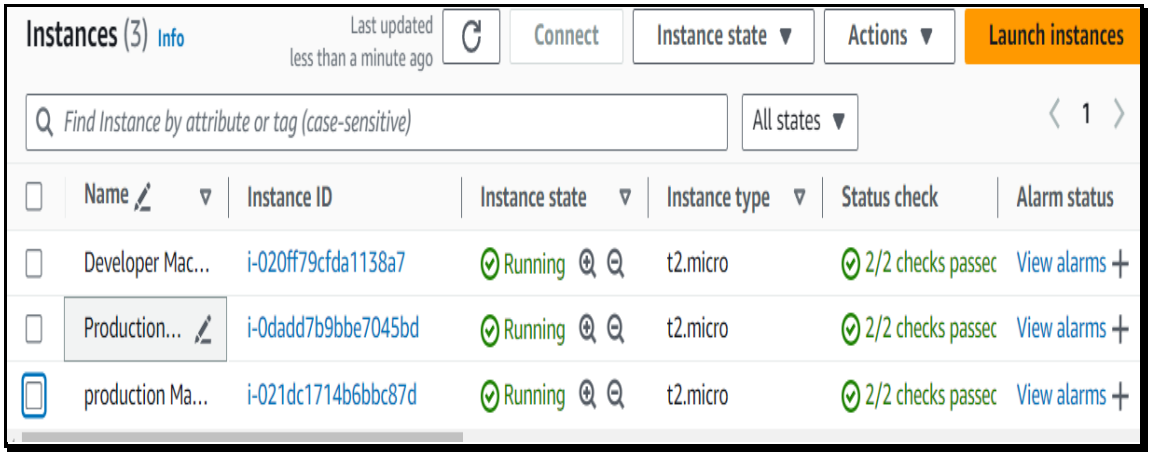


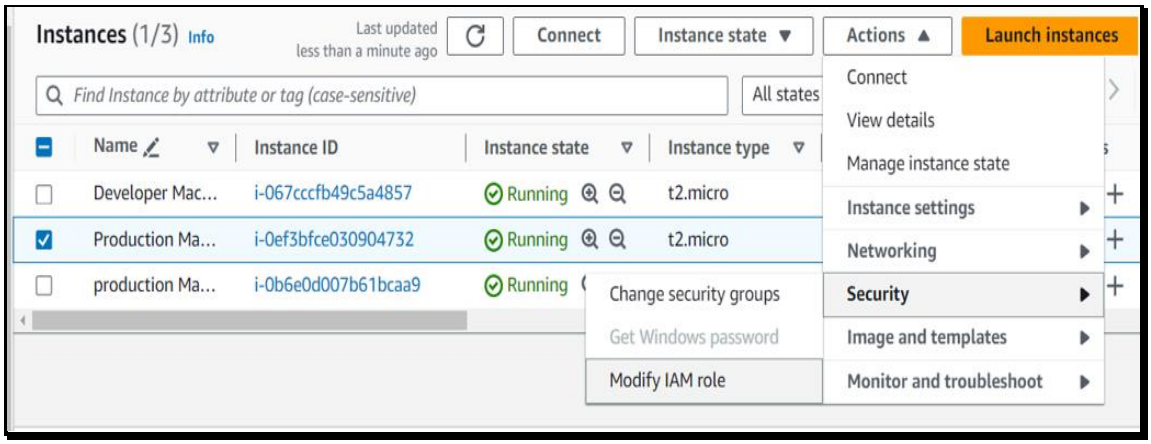

Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
LIN
Create new key pair

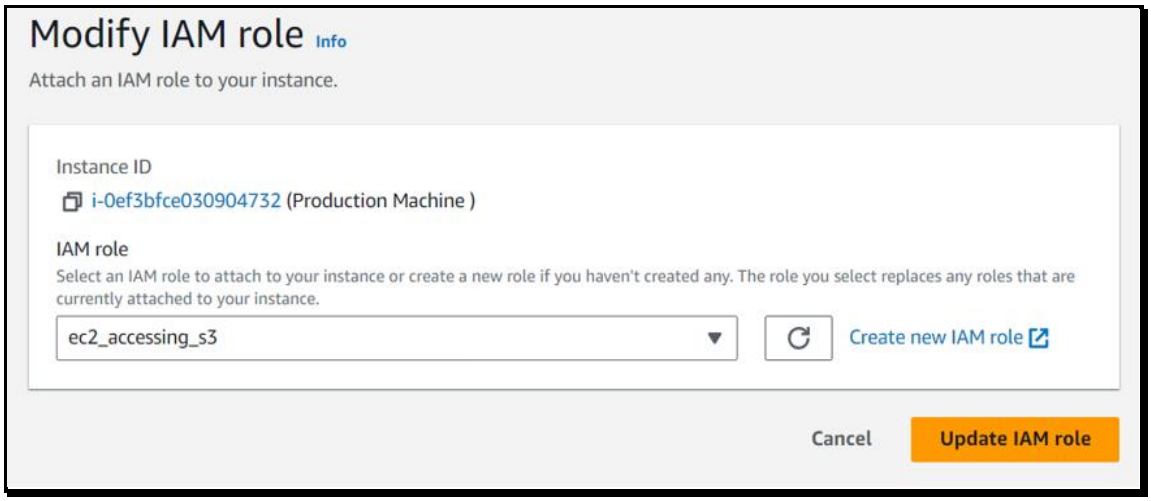

Network settings Info
Subnet Info
subnet-0a78929b1ec5b1fc3
myvpc-Public-A
VPC: vpc-06e16904a1fe991cd Owner: 339712987077
Availability Zone: us-east-2a Zone type: Availability Zone
IP addresses available: 246 CIDR: 10.1.10.0/24)
Create new subnet
When you specify a subnet, a network interface is automatically added to your template.
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Select existing security group
Create security group

b) Create a 2 instance for Production Machine



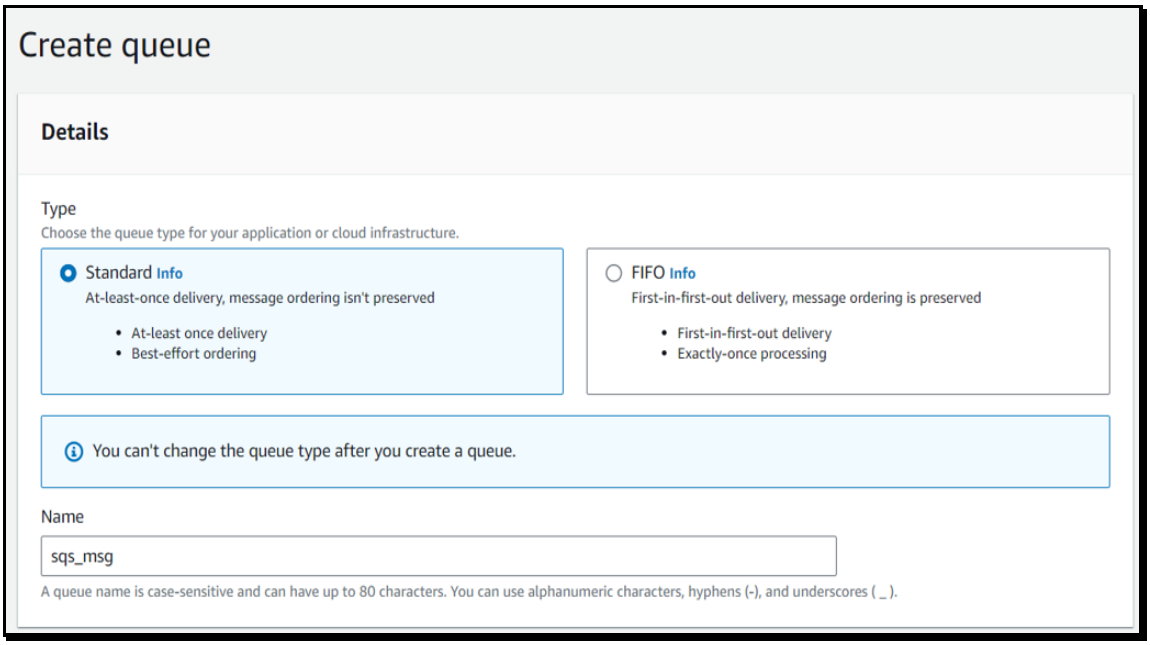


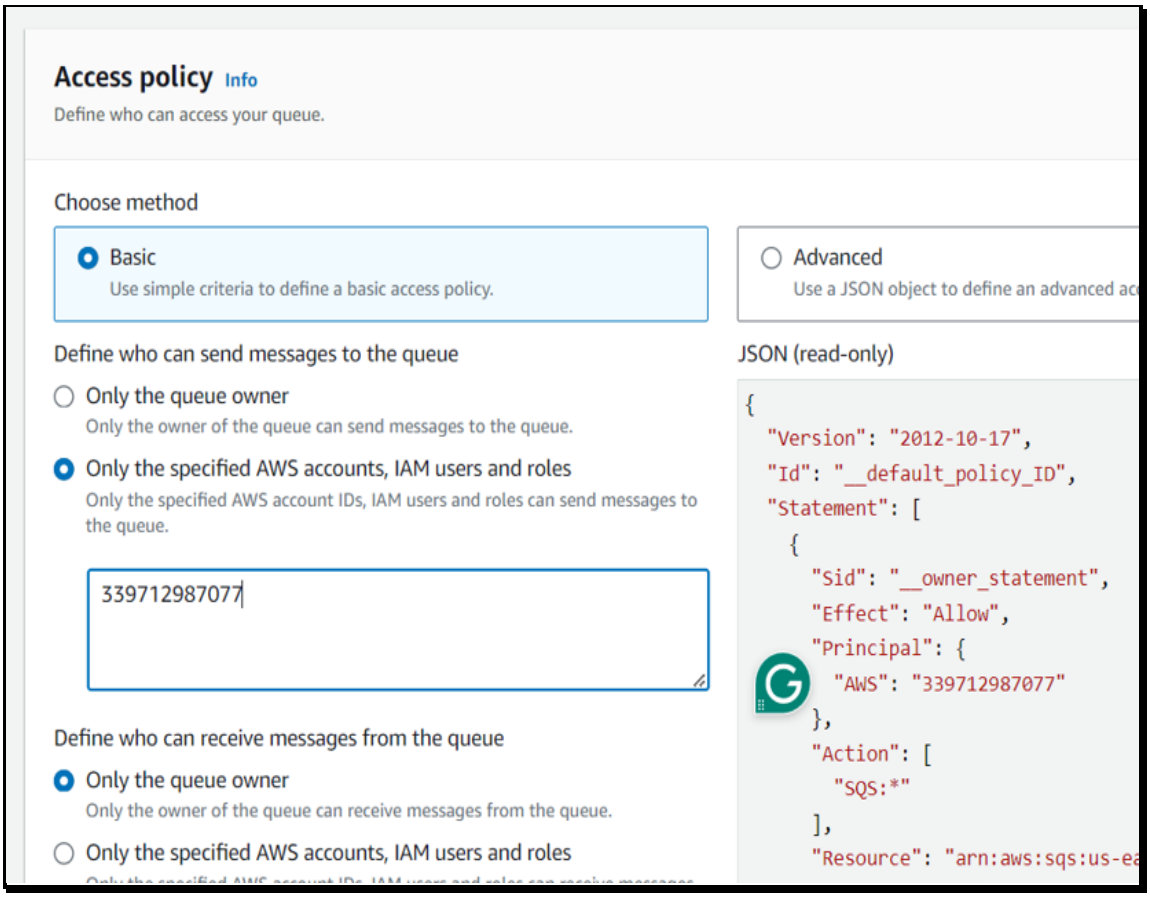




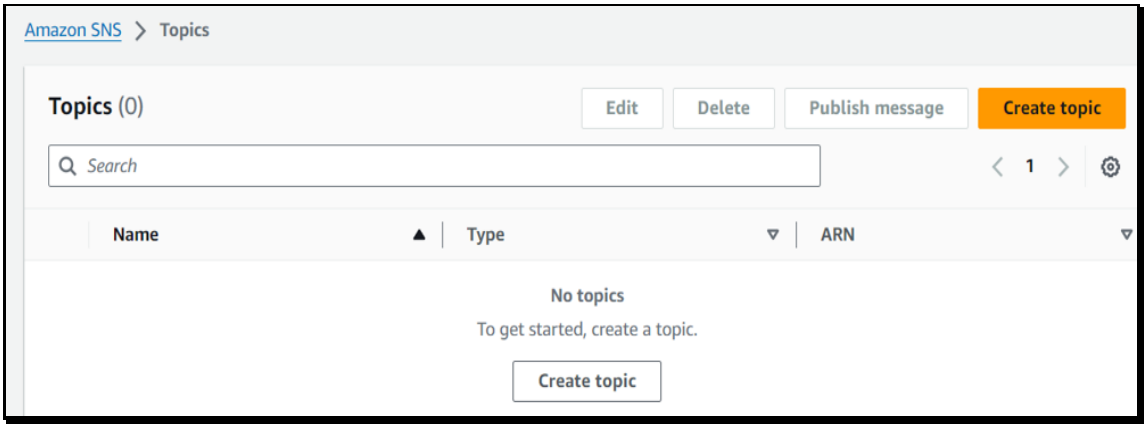
Step 5 : Create **SQS** & **SNS** for Communication service.

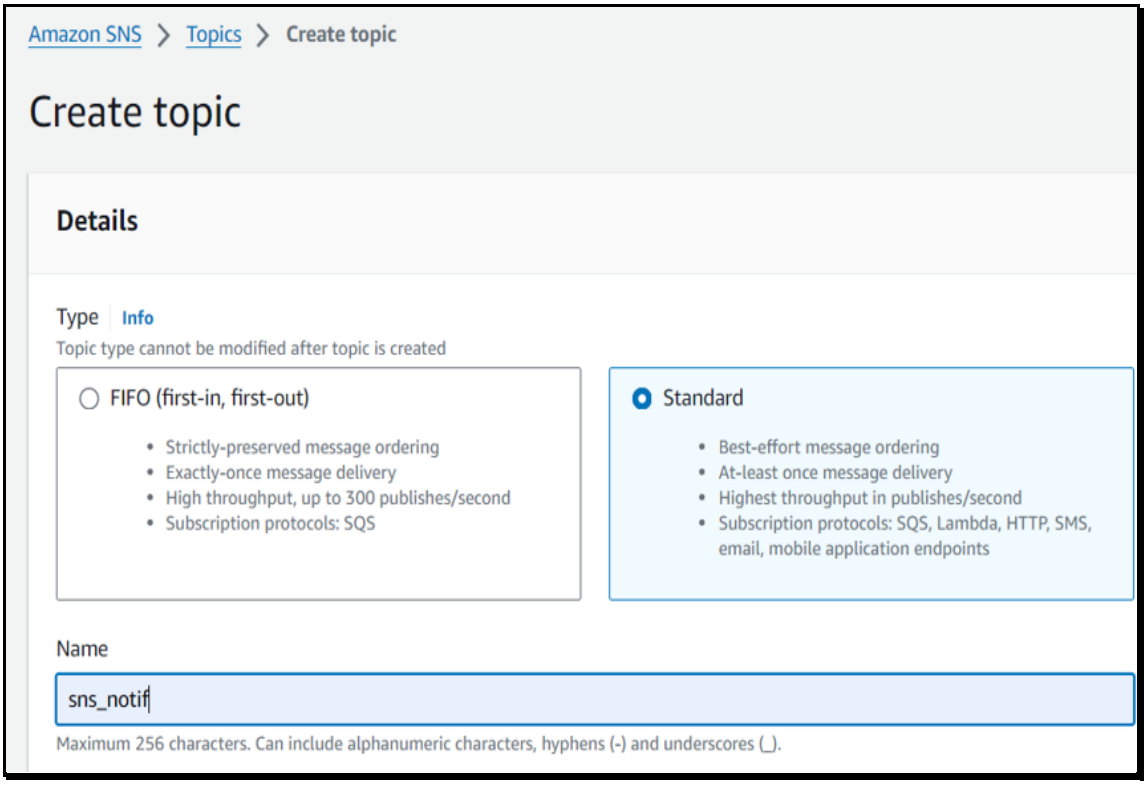
a) Create a SQS

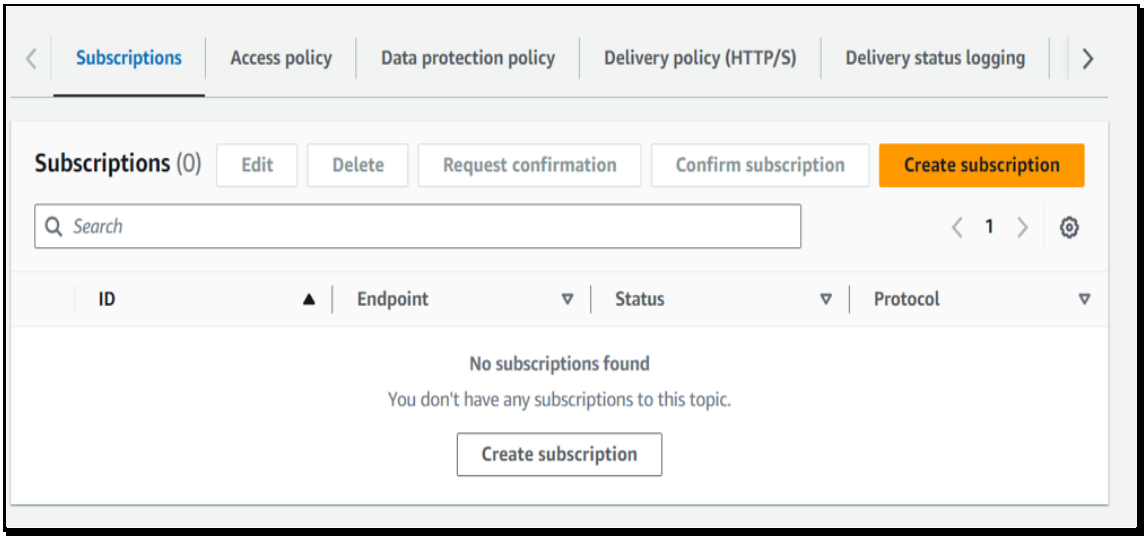


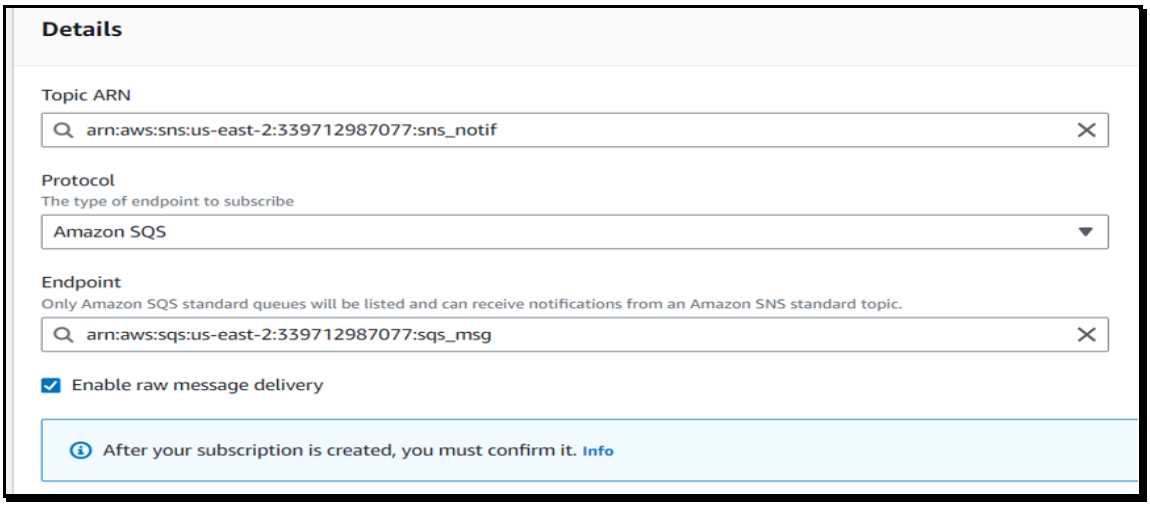


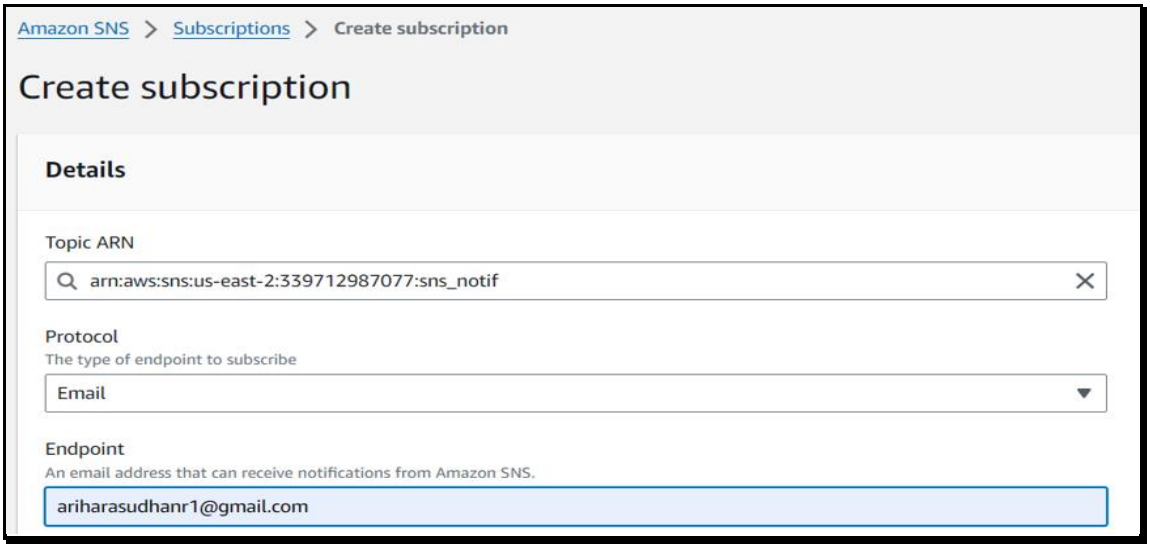
b) Create a SNS

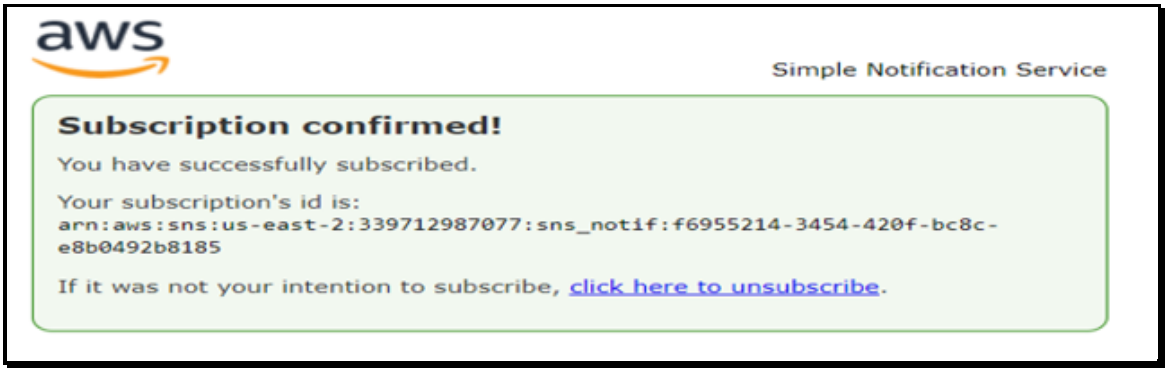


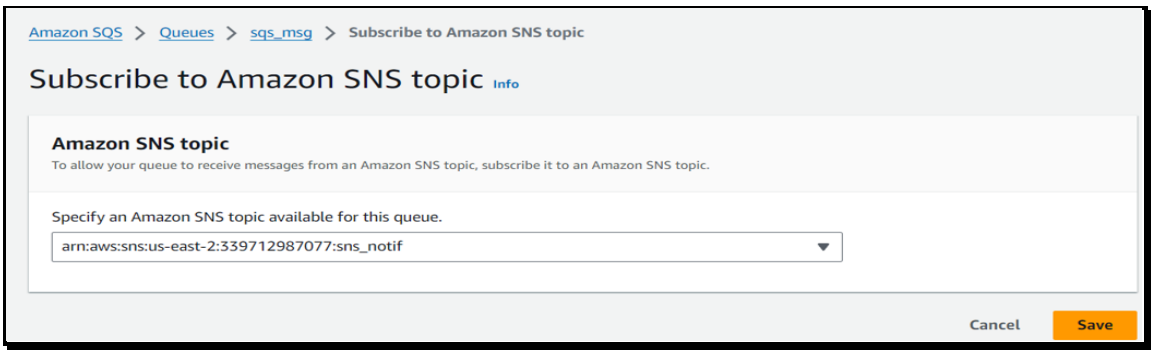




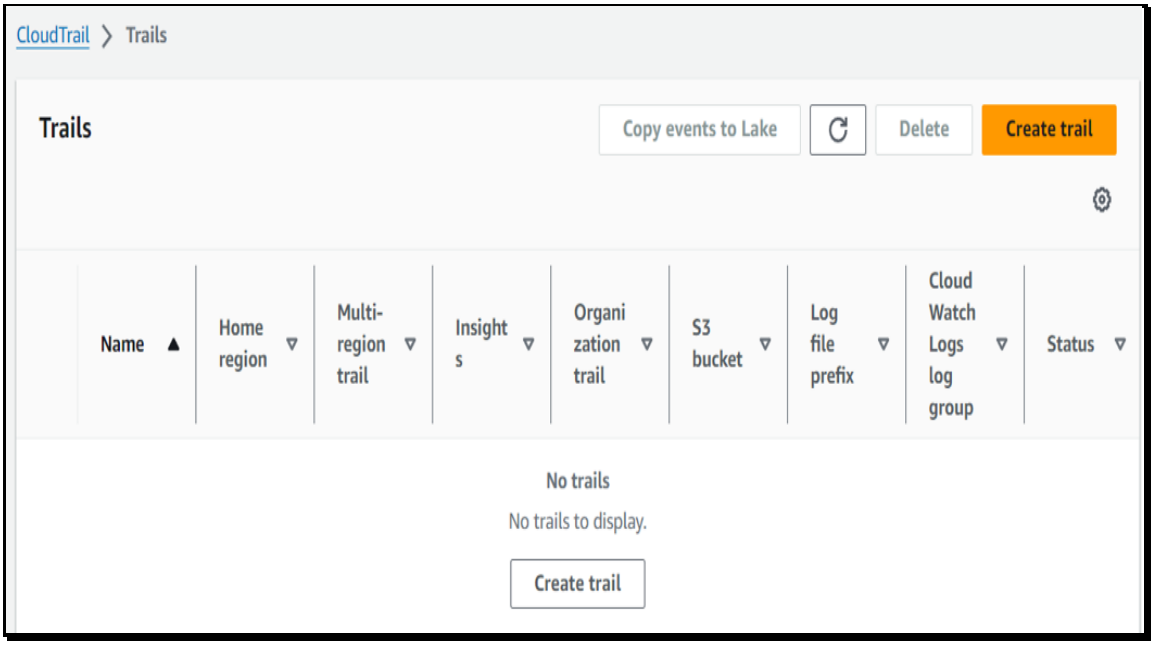

Details
Topic ARN
arn:aws:sns:us-east-2:339712987077:sns_notif
Protocol
The type of endpoint to subscribe
Amazon SQS
Endpoint
Only Amazon SQS standard queues will be listed and can receive notifications from an Amazon SNS standard topic.
arn:aws:sqs:us-east-2:339712987077:sqs_msg
Enable raw message delivery
After your subscription is created, you must confirm it. Info

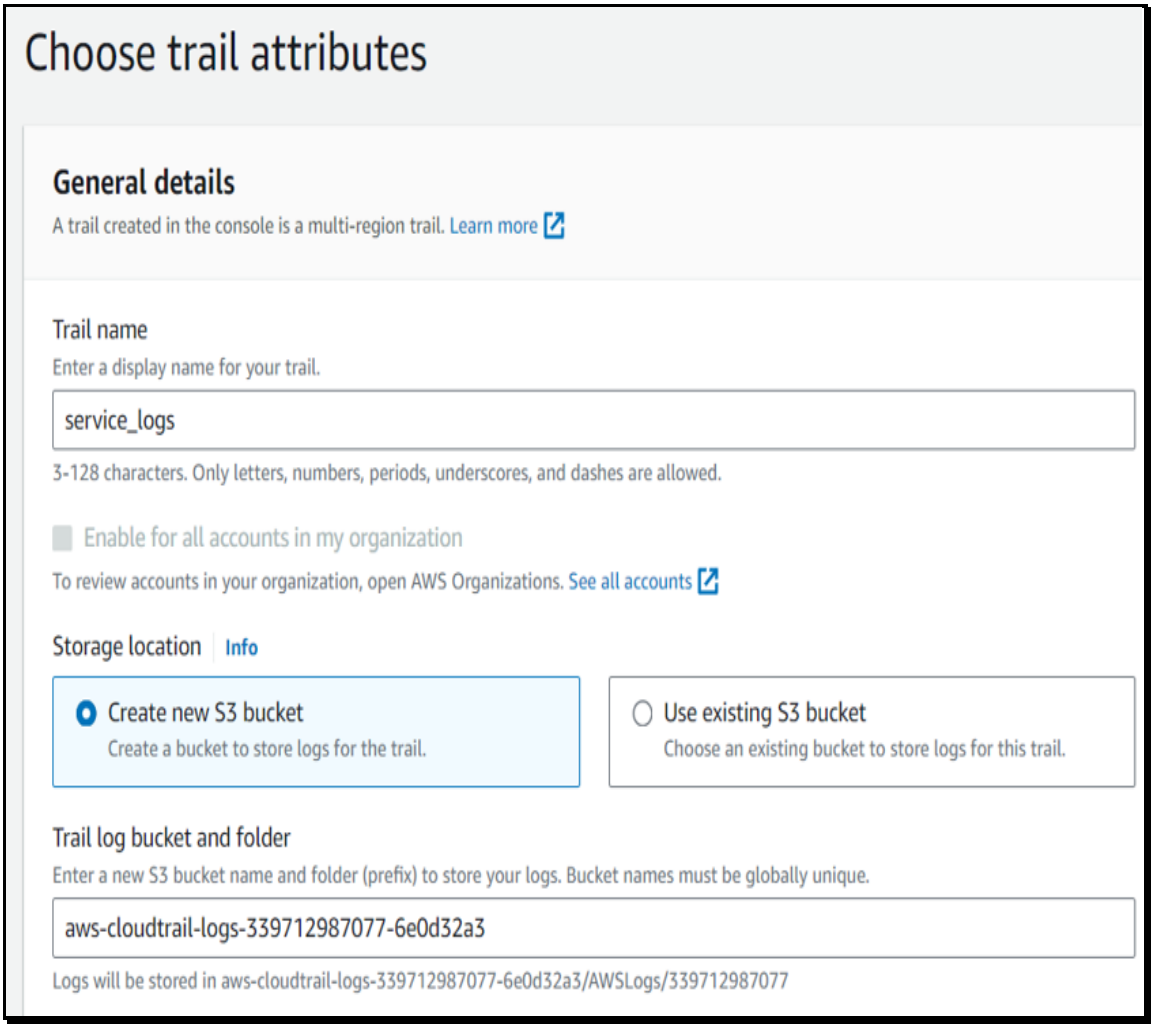

Amazon SNS > Subscriptions > Create subscription
Create subscription
Details
Topic ARN
arn:aws:sns:us-east-2:339712987077:sns_notif
Protocol
The type of endpoint to subscribe
Email
Endpoint
An email address that can receive notifications from Amazon SNS.
ariharasudhanr1@gmail.com

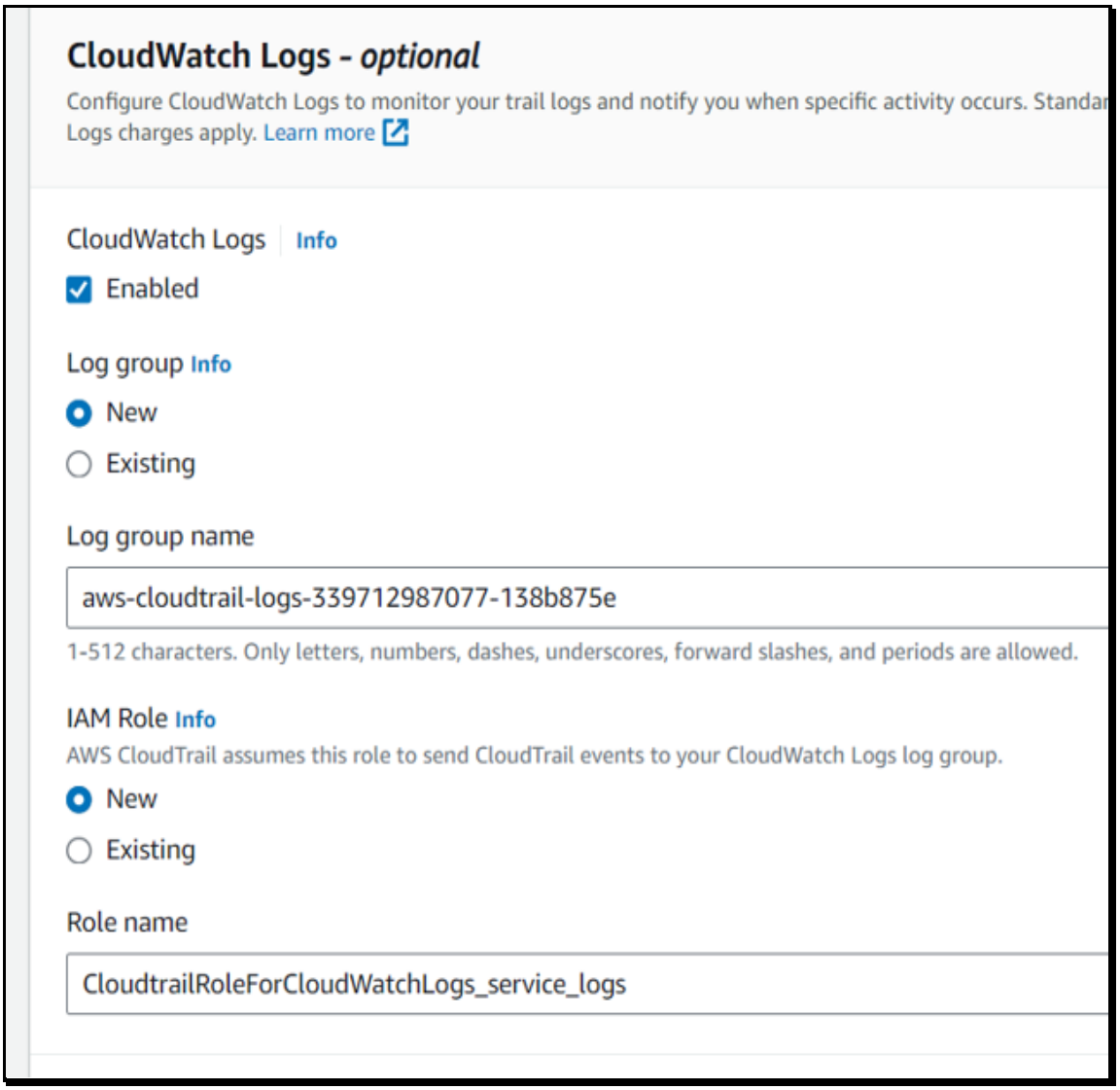

aws
Simple Notification Service
Subscription confirmed!
You have successfully subscribed.
Your subscription's id is:
arn:aws:sns:us-east-2:339712987077:sns_notif:f6955214-3454-420f-bc8c-e8b0492b8185
If it was not your intention to subscribe, click here to unsubscribe.

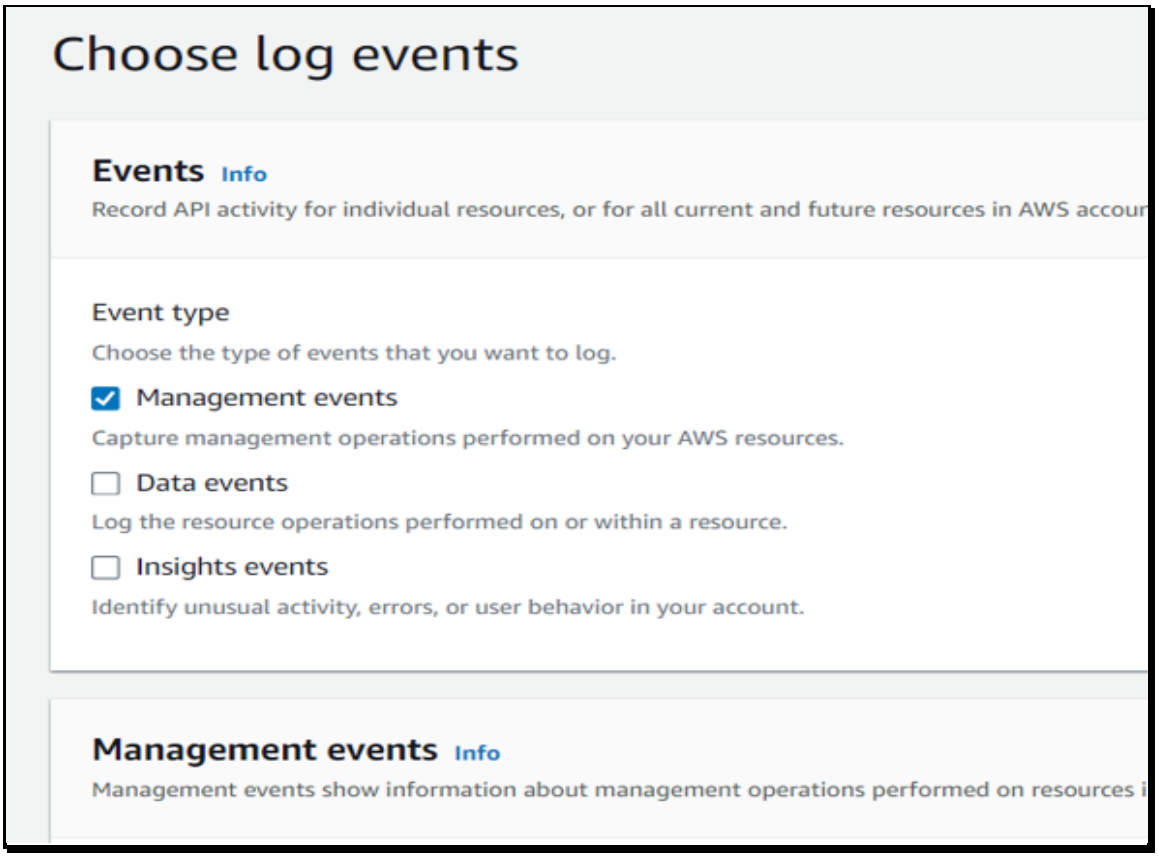

Amazon SQS > Queues > sqs_msg > Subscribe to Amazon SNS topic
Subscribe to Amazon SNS topic Info
Amazon SNS topic
To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.
Specify an Amazon SNS topic available for this queue.
arn:aws:sns:us-east-2:339712987077:sns_notif
Cancel
Save

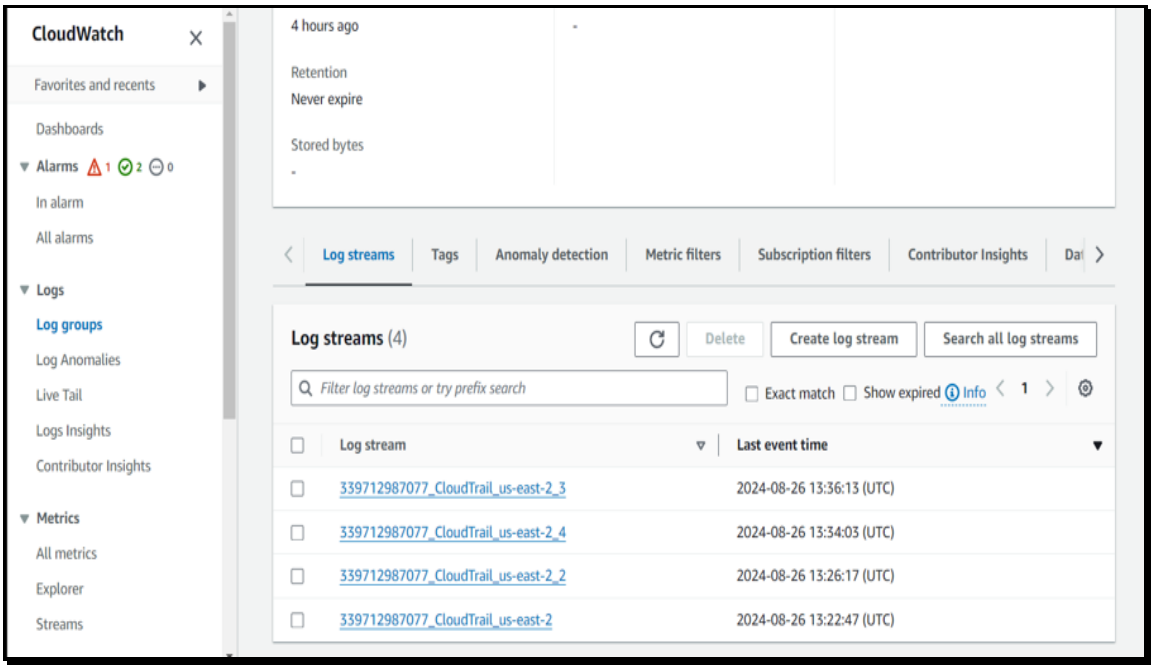
Step 6 : Create **Cloud Trail** for Logs that streams on **Cloud Watch**.

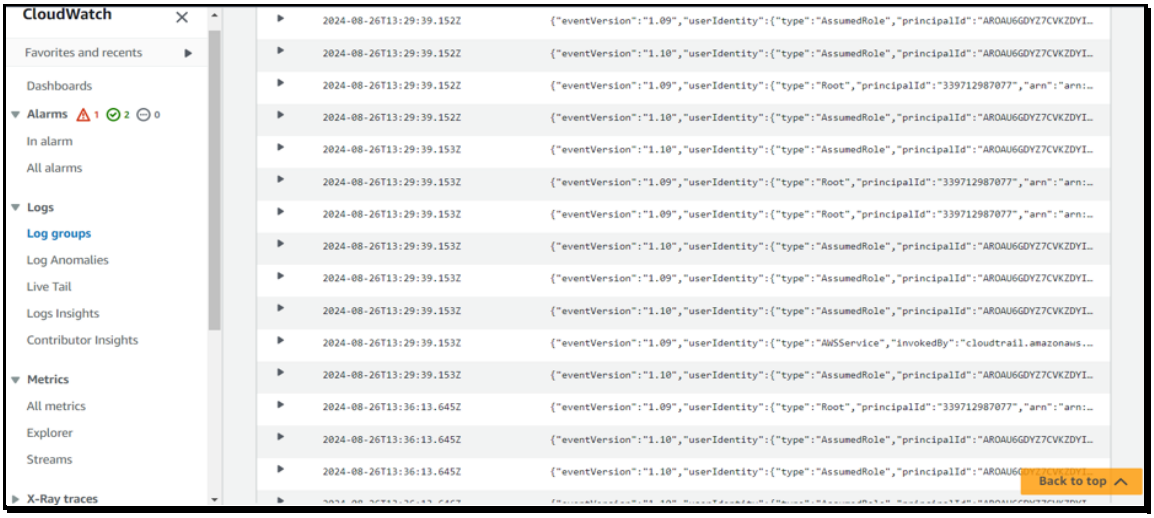


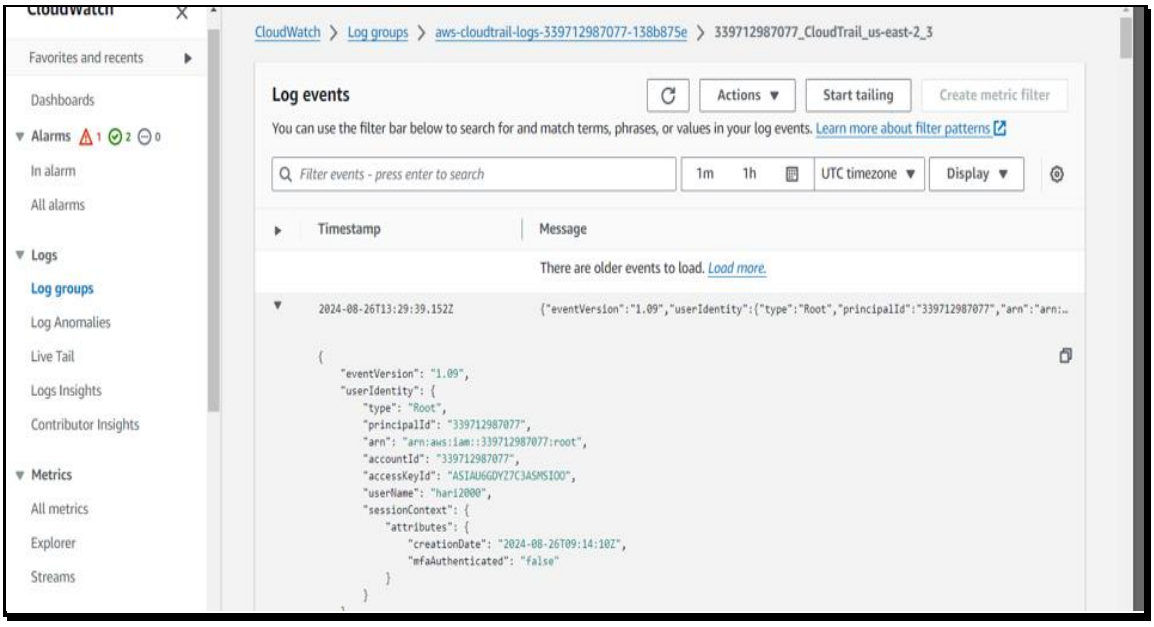



CloudWatch Logs - optional
Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standar
Logs charges apply. Learn more
CloudWatch Logs Info
Enabled
Log group Info
New
Existing
Log group name
aws-cloudtrail-logs-339712987077-138b875e
1-512 characters. Only letters, numbers, dashes, underscores, forward slashes, and periods are allowed.
IAM Role Info
AWS CloudTrail assumes this role to send CloudTrail events to your CloudWatch Logs log group.
New
Existing
Role name
CloudtrailRoleForCloudWatchLogs_service_logs

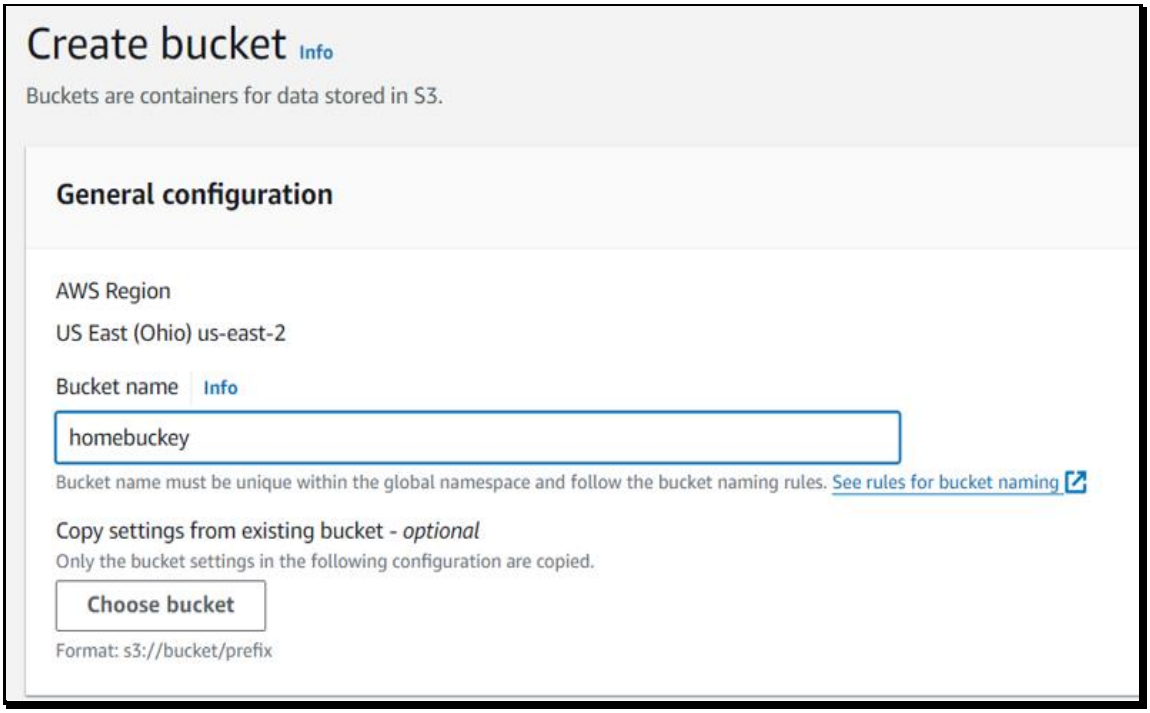

Choose log events
Events Info
Record API activity for individual resources, or for all current and future resources in AWS accour
Event type
Choose the type of events that you want to log.
Management events
Capture management operations performed on your AWS resources.
Data events
Log the resource operations performed on or within a resource.
Insights events
Identify unusual activity, errors, or user behavior in your account.
Management events Info
Management events show information about management operations performed on resources i

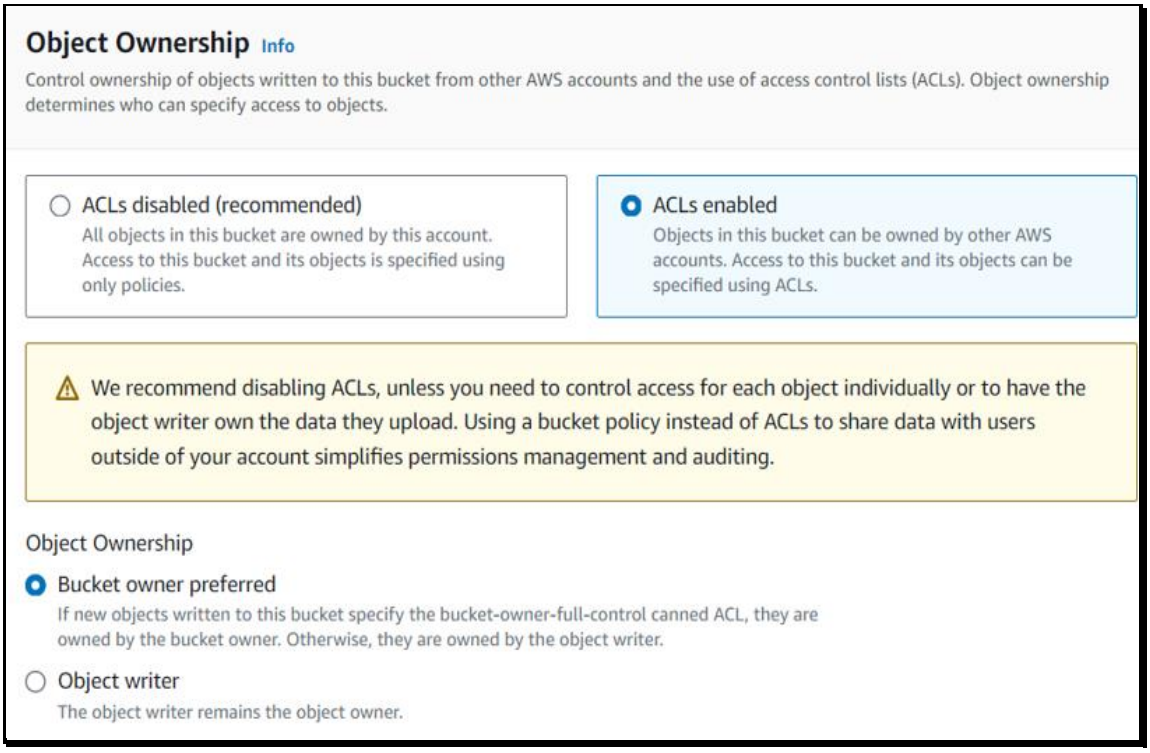

CloudWatch
Favorites and recents
Dashboards
Alarms 1 2 0
In alarm
All alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
All metrics
Explorer
Streams
4 hours ago
Retention
Never expire
Stored bytes
Log streams
Tags
Anomaly detection
Metric filters
Subscription filters
Contributor Insights
Log streams (4)
Delete
Create log stream
Search all log streams
Filter log streams or try prefix search
Exact match
Show expired
Info
Log stream
Last event time
339712987077_CloudTrail_us-east-2_3
2024-08-26 13:36:13 (UTC)
339712987077_CloudTrail_us-east-2_4
2024-08-26 13:34:03 (UTC)
339712987077_CloudTrail_us-east-2_2
2024-08-26 13:26:17 (UTC)
339712987077_CloudTrail_us-east-2
2024-08-26 13:22:47 (UTC)

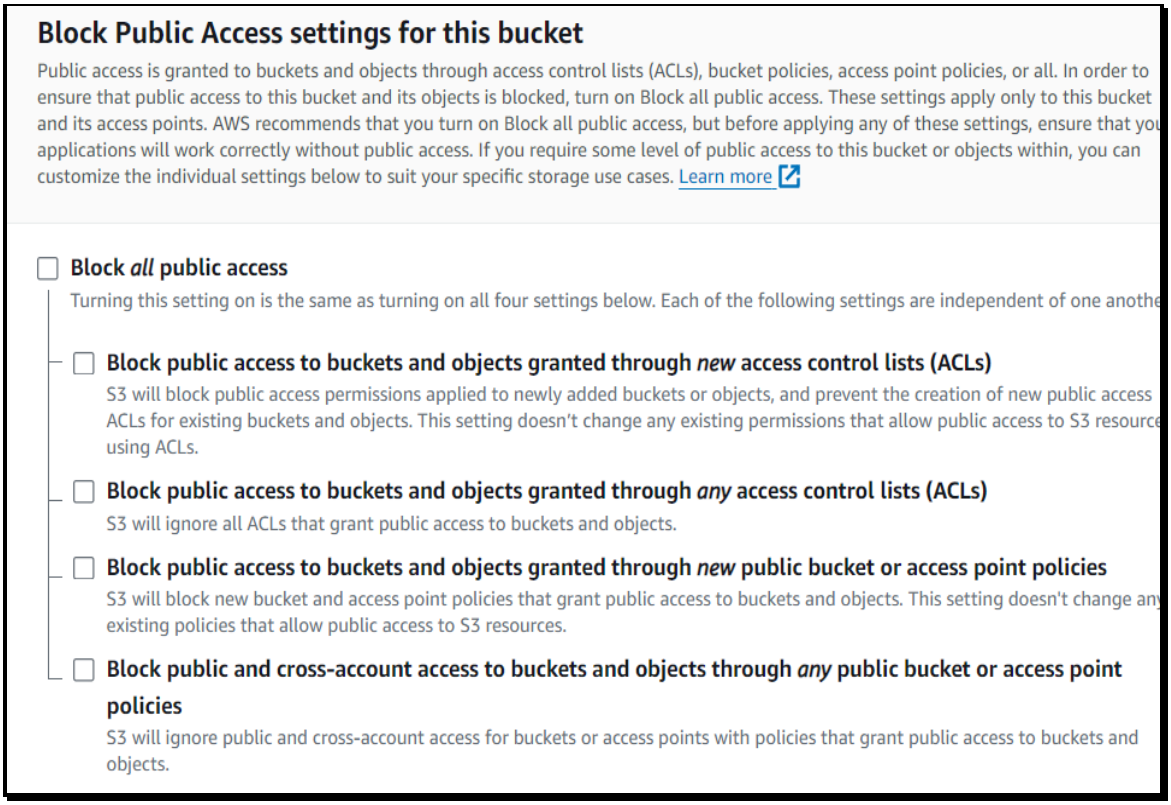

CloudWatch
Favorites and recents
Dashboards
Alarms 1 2 0
In alarm
All alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
All metrics
Explorer
Streams
X-Ray traces
Back to top

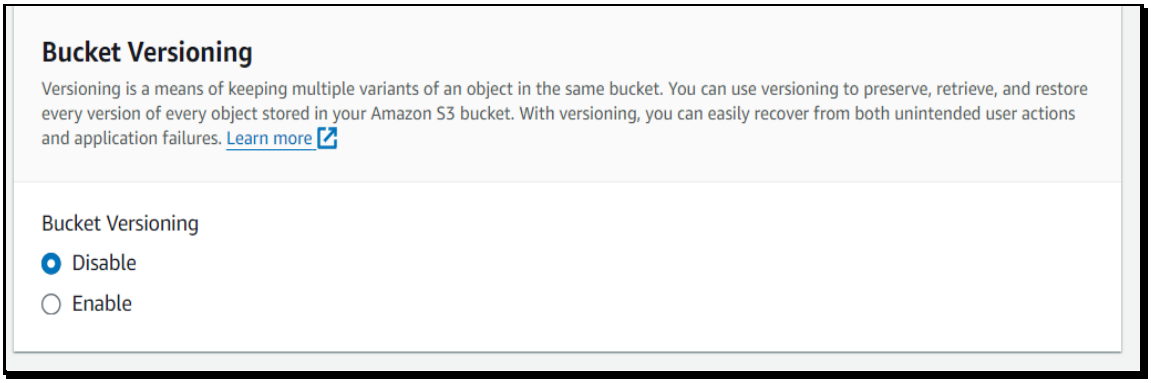

CloudWatch
Favorites and recents
Dashboards
Alarms 1 2 0
In alarm
All alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
All metrics
Explorer
Streams
CloudWatch > Log groups > aws-cloudtrail-logs-339712987077-138b875e > 339712987077_CloudTrail_us-east-2_3
Log events
Actions
Start tailing
Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Filter events - press enter to search
1m
1h
UTC timezone
Display
Timestamp
Message
There are older events to load. Load more.
2024-08-26T13:29:39.152Z
{"eventVersion":"1.09","userIdentity":{"type":"Root","principalId":"339712987077","arn":"arn:...
{
"eventVersion": "1.09",
"userIdentity": {
"type": "Root",
"principalId": "339712987077",
"arn": "arn:aws:iam::339712987077:root",
"accountId": "339712987077",
"accessKeyId": "ASIAU6GDYZ7C3ASMSIOO",
"userName": "hari2000",
"sessionContext": {
"attributes": {
"creationDate": "2024-08-26T09:14:10Z",
"mfaAuthenticated": "false"
}
}

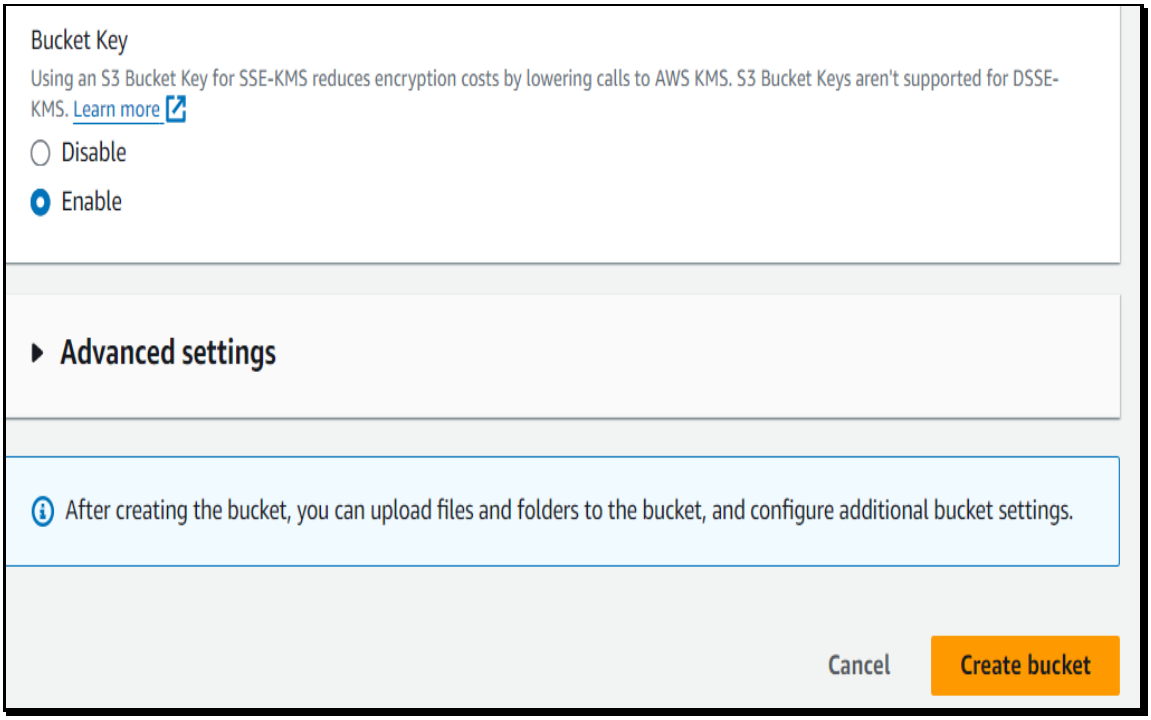
Step 7 : Create **S3 Bucket** for store a website files.





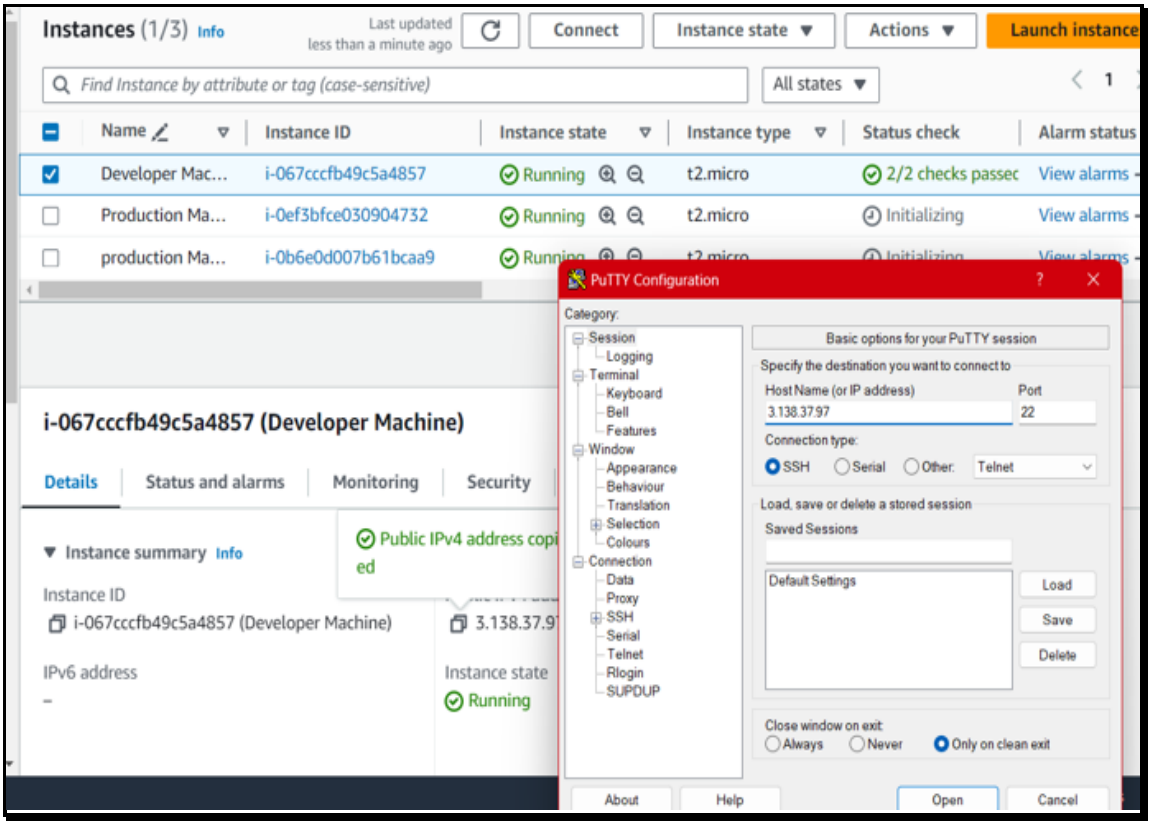

Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that you applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.


Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable

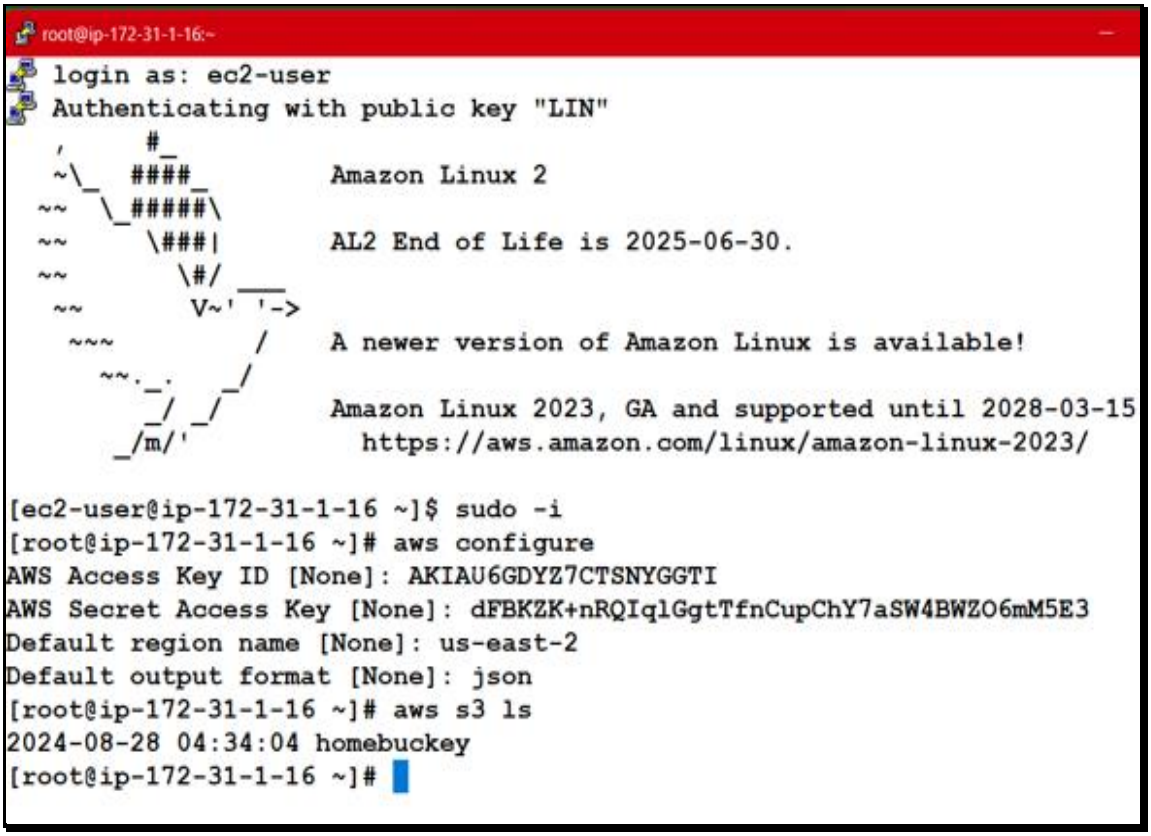

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

Step 8 : Connect IAM user in Developer Machine **CLI.**

Put a Developer Machine IP in Putty

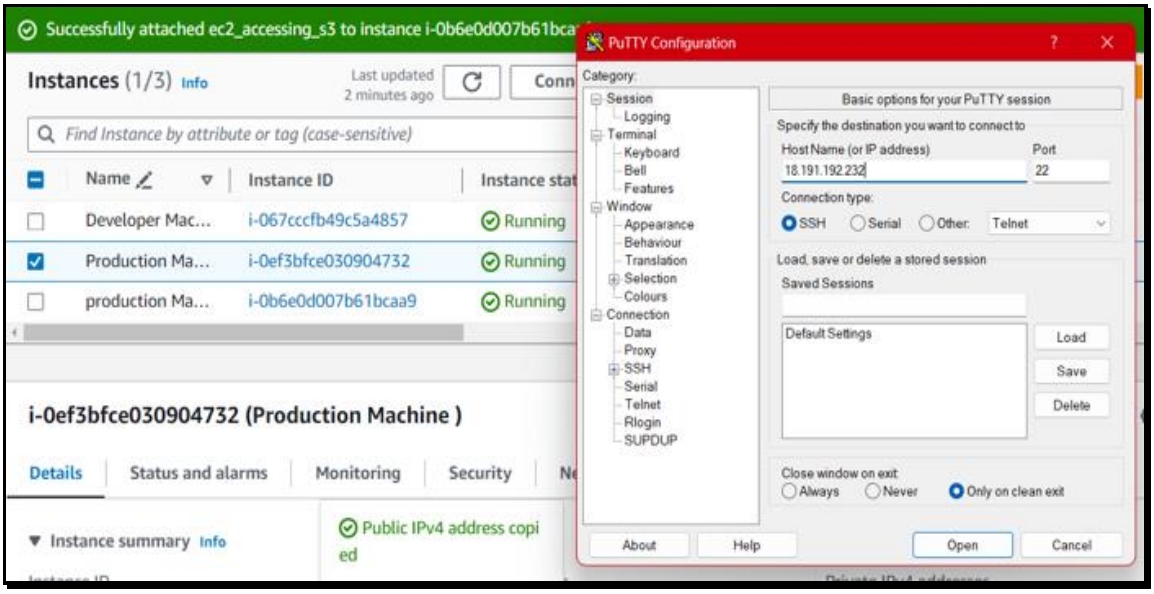


Connect IAM user in Developer Machine CLI

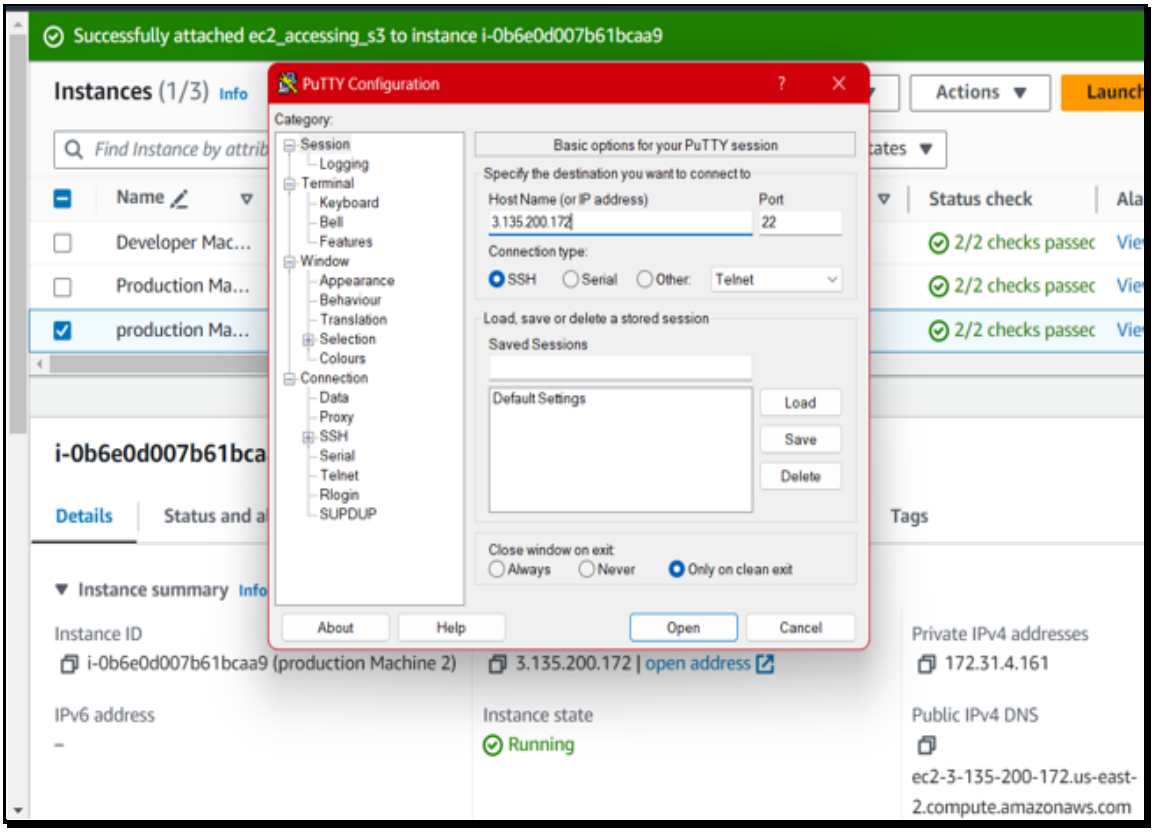


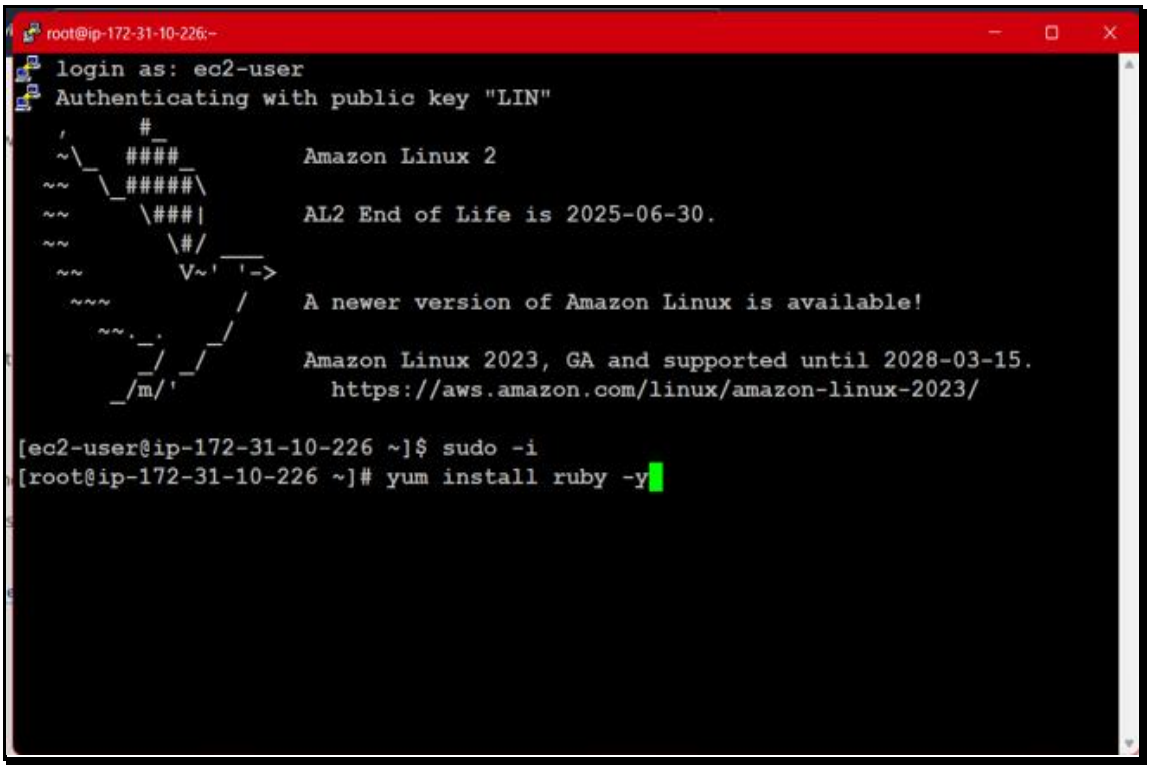
Step 9 : Install and prepare the Code Deploy agent on webserver in 2 Production Machines.

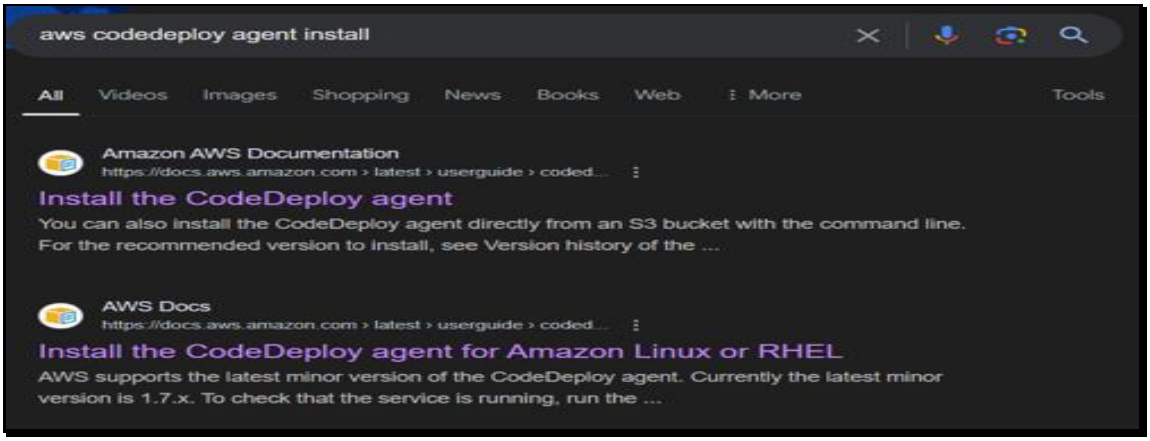
a) Put a Production Machine 1 IP in Putty



b) Put a Production Machine 2 IP in Putty



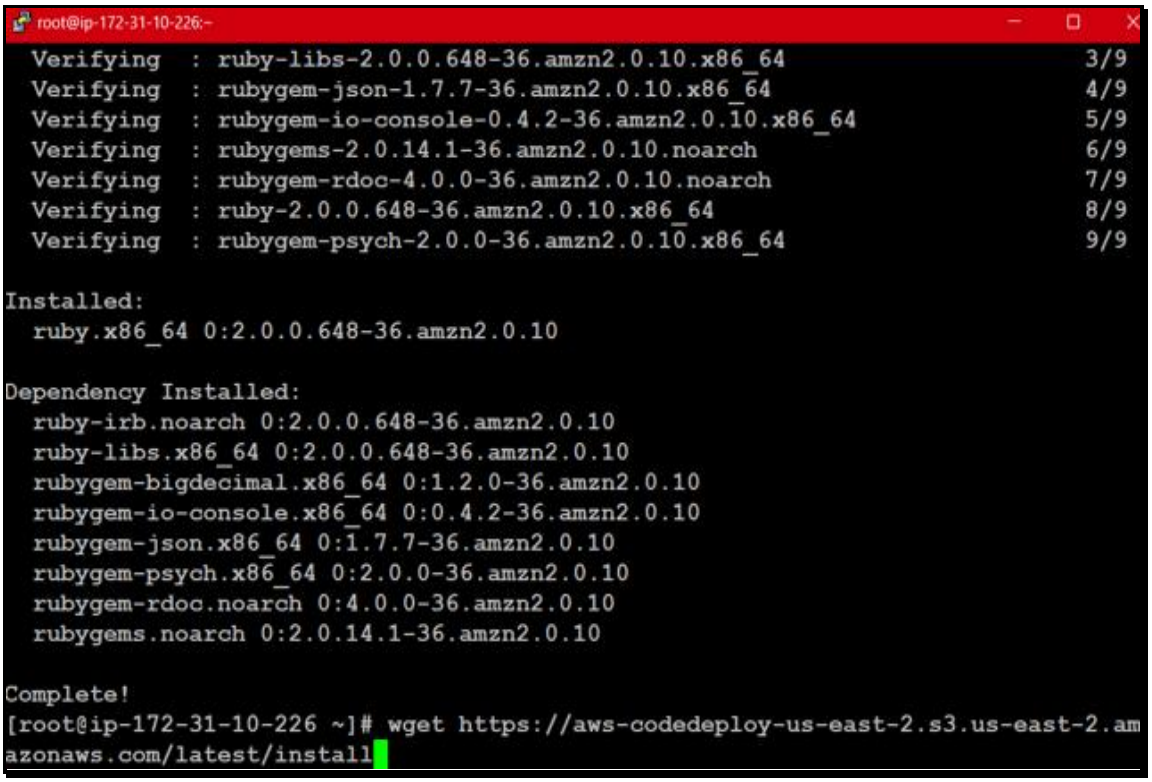


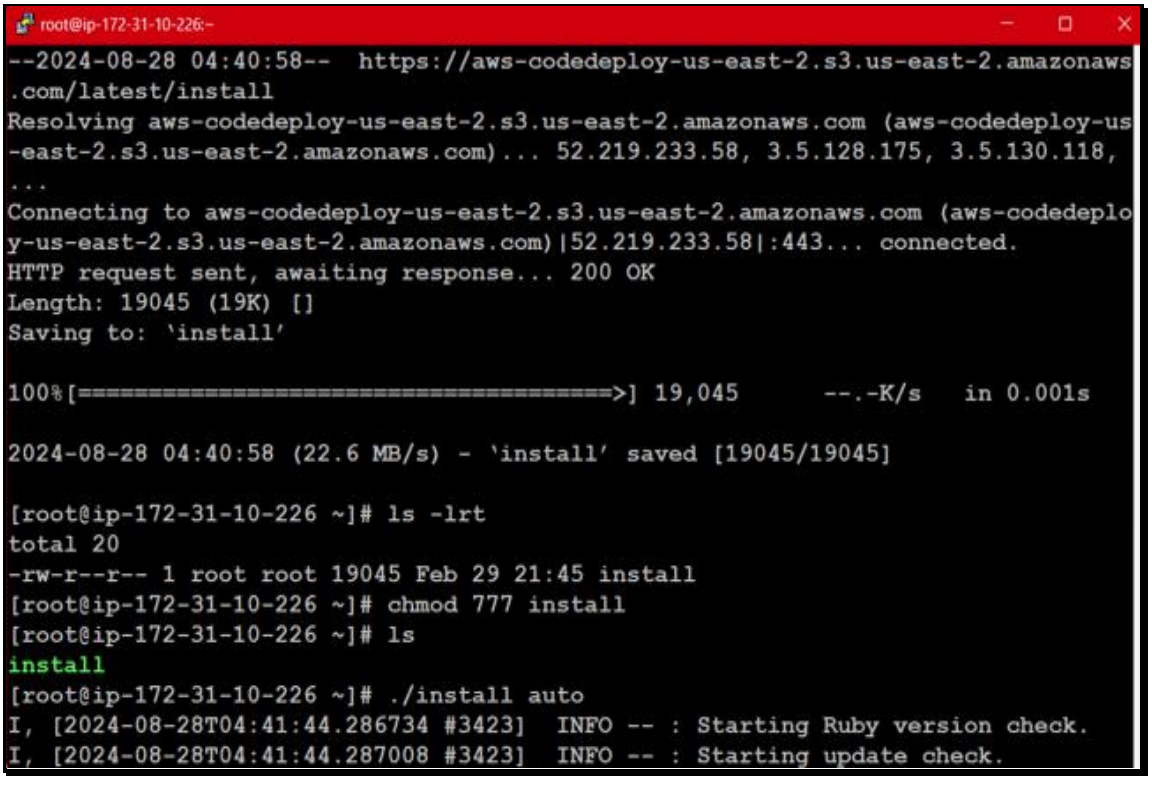


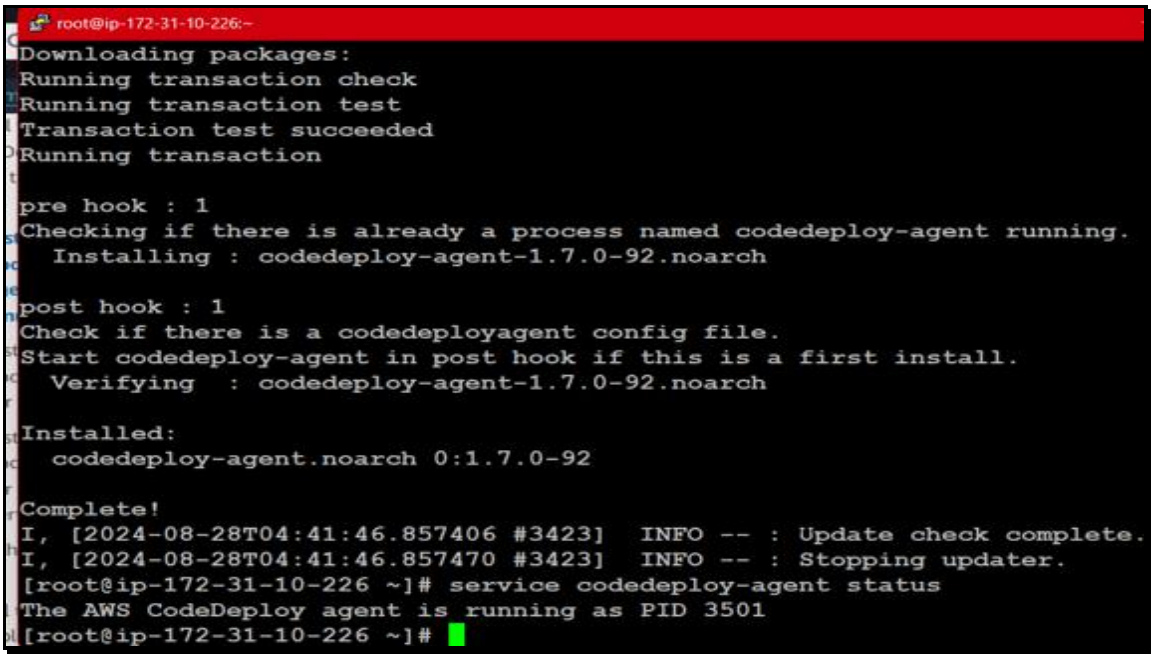
bucket-name is the name of the Amazon S3 bucket that contains the CodeDeploy Resource Kit files for your region, and *region-identifier* is the identifier for your region.

For example:

`https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install`


root@ip-172-31-10-226:~
Verifying : ruby-libs-2.0.0.648-36.amzn2.0.10.x86_64 3/9
Verifying : rubygem-json-1.7.7-36.amzn2.0.10.x86_64 4/9
Verifying : rubygem-io-console-0.4.2-36.amzn2.0.10.x86_64 5/9
Verifying : rubygems-2.0.14.1-36.amzn2.0.10.noarch 6/9
Verifying : rubygem-rdoc-4.0.0-36.amzn2.0.10.noarch 7/9
Verifying : ruby-2.0.0.648-36.amzn2.0.10.x86_64 8/9
Verifying : rubygem-psych-2.0.0-36.amzn2.0.10.x86_64 9/9
Installed:
ruby.x86_64 0:2.0.0.648-36.amzn2.0.10
Dependency Installed:
ruby-irb.noarch 0:2.0.0.648-36.amzn2.0.10
ruby-libs.x86_64 0:2.0.0.648-36.amzn2.0.10
rubygem-bigdecimal.x86_64 0:1.2.0-36.amzn2.0.10
rubygem-io-console.x86_64 0:0.4.2-36.amzn2.0.10
rubygem-json.x86_64 0:1.7.7-36.amzn2.0.10
rubygem-psych.x86_64 0:2.0.0-36.amzn2.0.10
rubygem-rdoc.noarch 0:4.0.0-36.amzn2.0.10
rubygems.noarch 0:2.0.14.1-36.amzn2.0.10
Complete!
[root@ip-172-31-10-226 ~]# wget https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install


root@ip-172-31-10-226:~
--2024-08-28 04:40:58-- https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install
Resolving aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com (aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com)... 52.219.233.58, 3.5.128.175, 3.5.130.118,
...
Connecting to aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com (aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com)|52.219.233.58|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19045 (19K) []
Saving to: 'install'
100%[=====>] 19,045 --.-K/s in 0.001s
2024-08-28 04:40:58 (22.6 MB/s) - 'install' saved [19045/19045]
[root@ip-172-31-10-226 ~]# ls -lrt
total 20
-rw-r--r-- 1 root root 19045 Feb 29 21:45 install
[root@ip-172-31-10-226 ~]# chmod 777 install
[root@ip-172-31-10-226 ~]# ls
install
[root@ip-172-31-10-226 ~]# ./install auto
I, [2024-08-28T04:41:44.286734 #3423] INFO -- : Starting Ruby version check.
I, [2024-08-28T04:41:44.287008 #3423] INFO -- : Starting update check.


root@ip-172-31-10-226:~
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
pre hook : 1
Checking if there is already a process named codedeploy-agent running.
Installing : codedeploy-agent-1.7.0-92.noarch
post hook : 1
Check if there is a codedeployagent config file.
Start codedeploy-agent in post hook if this is a first install.
Verifying : codedeploy-agent-1.7.0-92.noarch
Installed:
codedeploy-agent.noarch 0:1.7.0-92
Complete!
I, [2024-08-28T04:41:46.857406 #3423] INFO -- : Update check complete.
I, [2024-08-28T04:41:46.857470 #3423] INFO -- : Stopping updater.
[root@ip-172-31-10-226 ~]# service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 3501
[root@ip-172-31-10-226 ~]#


root@ip-172-31-4-161:/var/www/html/register
Verifying : codedeploy-agent-1.7.0-92.noarch
Installed:
codedeploy-agent.noarch 0:1.7.0-92
Complete!
I, [2024-08-28T04:45:01.429682 #3425] INFO -- : Update check complete.
I, [2024-08-28T04:45:01.429741 #3425] INFO -- : Stopping updater.
[root@ip-172-31-4-161 ~]# service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 3503

Step 10 : Create the code from Developer Machine.

```
root@ip-172-31-1-16:~/deploy_dir/sampleapp
login as: ec2-user
Authenticating with public key "LIN"
       #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-1-16 ~]$ sudo -i
[root@ip-172-31-1-16 ~]# aws configure
AWS Access Key ID [None]: AKIAU6GDYZ7CTSNYGGTI
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]: json
[root@ip-172-31-1-16 ~]# aws s3 ls
2024-08-28 04:34:04 homebuckey
[root@ip-172-31-1-16 ~]# mkdir deploy_dir
[root@ip-172-31-1-16 ~]# cd deploy_dir/
[root@ip-172-31-1-16 deploy_dir]# mkdir sampleapp
[root@ip-172-31-1-16 deploy_dir]# cd sampleapp/
[root@ip-172-31-1-16 sampleapp]# pwd
/root/deploy_dir/sampleapp
[root@ip-172-31-1-16 sampleapp]# vi index.html
```

```
<div class="footer">
    <center>
    <h1>Contact</h1>
    <h2>+91 9876543210 | btreesystems@gmail.com</h2>
    <p >
        <a class="links" href="https://www.youtube.com/@btreesystems">Youtube</a
> |
        <a class="links" href="https://www.instagram.com/btreesystems/?hl=en">In
stagram</a> |
        <a  class="links" href="https://www.linkedin.com/company/btreesystems/?o
riginalSubdomain=in">Linkedin</a>
    </p>

    </center>


</div>



    </body>
</html
```

```
       #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-1-16 ~]$ sudo -i
[root@ip-172-31-1-16 ~]# aws configure
AWS Access Key ID [None]: AKIAU6GDYZ7CTSNYGGTI
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]: json
[root@ip-172-31-1-16 ~]# aws s3 ls
2024-08-28 04:34:04 homebuckey
[root@ip-172-31-1-16 ~]# mkdir deploy_dir
[root@ip-172-31-1-16 ~]# cd deploy_dir/
[root@ip-172-31-1-16 deploy_dir]# mkdir sampleapp
[root@ip-172-31-1-16 deploy_dir]# cd sampleapp/
[root@ip-172-31-1-16 sampleapp]# pwd
/root/deploy_dir/sampleapp
[root@ip-172-31-1-16 sampleapp]# vi index.html
[root@ip-172-31-1-16 sampleapp]# vi appspec.html
```

```
version: 0.0
os: linux
files:
  - source: /index.html
    destination: /var/www/html
hooks:
  BeforeInstall:
  - location: scripts/httpd_install.sh
    timeout: 300
    runas: root
  - location: scripts/httpd_start.sh
    timeout: 300
    runas: root
ApplicationStop:
  - location: scripts/httpd_stop.sh
    timeout: 300
    runas: root

~
```

```
[root@ip-10-1-10-106 sampleapp]# vi index.html
[root@ip-10-1-10-106 sampleapp]# vi appspec.yaml
[root@ip-10-1-10-106 sampleapp]# mkdir scripts
[root@ip-10-1-10-106 sampleapp]# cd scripts/
[root@ip-10-1-10-106 scripts]#
```

```
[root@ip-10-1-10-106 sampleapp]# cd scripts/
[root@ip-10-1-10-106 scripts]# touch httpd_install.sh
[root@ip-10-1-10-106 scripts]# touch httpd_start.sh
[root@ip-10-1-10-106 scripts]# touch httpd_stop.sh
[root@ip-10-1-10-106 scripts]# ls
httpd_install.sh  httpd_start.sh  httpd_stop.sh
[root@ip-10-1-10-106 scripts]# vi httpd_install.sh
```

```
#!/bin/bash
yum install -y httpd
~
~
~
~
~
~
~
~
```

```
[root@ip-10-1-10-106 sampleapp]# cd scripts/
[root@ip-10-1-10-106 scripts]# touch httpd_install.sh
[root@ip-10-1-10-106 scripts]# touch httpd_start.sh
[root@ip-10-1-10-106 scripts]# touch httpd_stop.sh
[root@ip-10-1-10-106 scripts]# ls
httpd_install.sh  httpd_start.sh  httpd_stop.sh
[root@ip-10-1-10-106 scripts]# vi httpd_install.sh
[root@ip-10-1-10-106 scripts]# vi httpd_start.sh
```

```
#!/bin/bash
systemctl start httpd
systemctl enable httpd
~
~
~
~
~
~
~
```

```
[root@ip-10-1-10-106 sampleapp]# cd scripts/
[root@ip-10-1-10-106 scripts]# touch httpd_install.sh
[root@ip-10-1-10-106 scripts]# touch httpd_start.sh
[root@ip-10-1-10-106 scripts]# touch httpd_stop.sh
[root@ip-10-1-10-106 scripts]# ls
httpd_install.sh  httpd_start.sh  httpd_stop.sh
[root@ip-10-1-10-106 scripts]# vi httpd_install.sh
[root@ip-10-1-10-106 scripts]# vi httpd_start.sh
[root@ip-10-1-10-106 scripts]# vi httpd_stop.sh
```

```
#!/bin/bash
systemctl stop httpd
systemctl disable httpd
~
~
~
~
~
~
~
~
~
```

```
[root@ip-10-1-10-106 sampleapp]# aws deploy create-application --application-name sampleapp
{
    "applicationId": "9b40548c-9923-4890-b97c-0aea87d1d03c"
```

```
[root@ip-10-1-10-106 scripts]# mkdir deploy
[root@ip-10-1-10-106 scripts]# cd deploy
[root@ip-10-1-10-106 deploy]# mkdir sampleapp
[root@ip-10-1-10-106 deploy]# ls
sampleapp
[root@ip-10-1-10-106 deploy]# rmdir sampleapp
[root@ip-10-1-10-106 deploy]# ls
[root@ip-10-1-10-106 deploy]# mkdir sample
[root@ip-10-1-10-106 deploy]# cd sample
[root@ip-10-1-10-106 sample]# pwd
/root/deploy_dir/sampleapp/scripts/deploy/sample
[root@ip-10-1-10-106 sample]# vi index.html
```

root@ip-10-1-10-106:~/deploy_dir/sampleapp/scripts/deploy/sample

```
    border-inline-color: black;
}

.tabs{
    margin-left: 90px;
}
.naming{
    border-radius: 10px;
    background-color: black;
    color: white;
}
</style>
</head>
<body>
    <div class="register">
    <center>
        <h1>BTREE SYSTEM</h1>
        <h4>Training | Career Building | Freshers Guide</h4>
        <p>Empowering young students to level up their skills through training in software development technologies
<200d> </p>
        <hr>
    </center>
    <center>
        <h2>Registration Form</h2>

        <form class =tabs>
            <table class="table">

                <tr>
                    <td >Name:</td>
                    <td><input class="names"type="text" placeholder="Enter your name"></td>
                </tr>

-- INSERT --                                                                    35,25         22%
```

```
[root@ip-10-1-10-106 sample]# vi appspec.yaml
[root@ip-10-1-10-106 sample]# aws deploy push --application-name sample --s3-location s3://homebuckey/sample.zip

/root/deploy_dir/sampleapp/scripts/deploy/sample/appspec.yml was not found
[root@ip-10-1-10-106 sample]# mv appspec.yaml appspec.yml
```

root@ip-172-31-1-16:~/deploy_dir/sampleapp/scripts/deploy/sample

```
version: 0.0
os: linux
files:
 - source: /index.html
   destination: /var/www/html/register
hooks:
 BeforeInstall:
 - location: scripts/httpd_install.sh
   timeout: 300
   runas: root
 - location: scripts/httpd_start.sh
   timeout: 300
   runas: root
ApplicationStop:
 - location: scripts/httpd_stop.sh
   timeout: 300
   runas: root
```

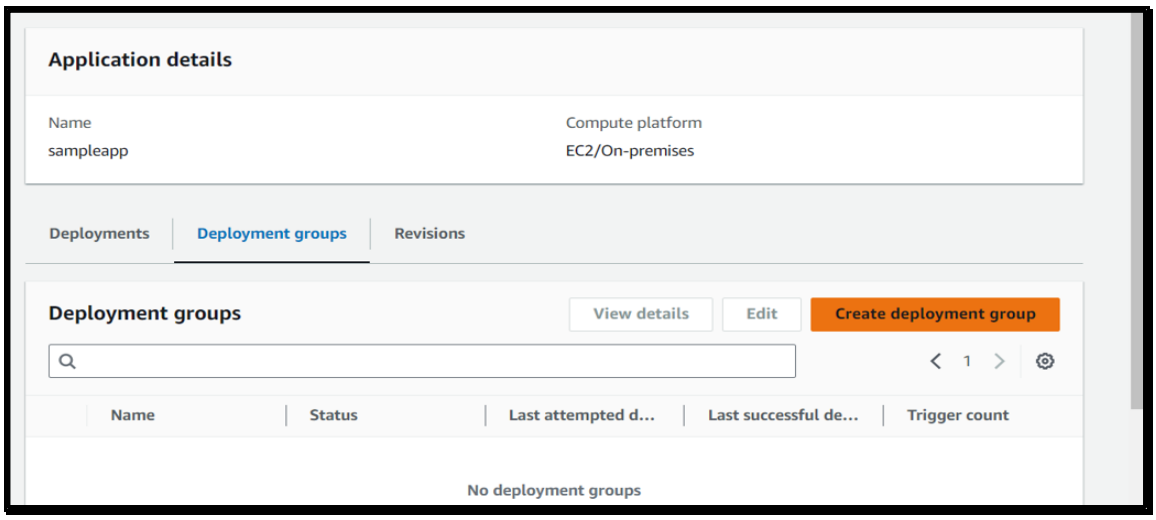
```
[root@ip-10-1-10-106 sample]# mkdir scripts
[root@ip-10-1-10-106 sample]# cd scripts/
[root@ip-10-1-10-106 scripts]# vi httpd_install.sh
[root@ip-10-1-10-106 scripts]# vi httpd_start.sh
[root@ip-10-1-10-106 scripts]# vi httpd_stop.sh
[root@ip-10-1-10-106 scripts]# cd ..
```

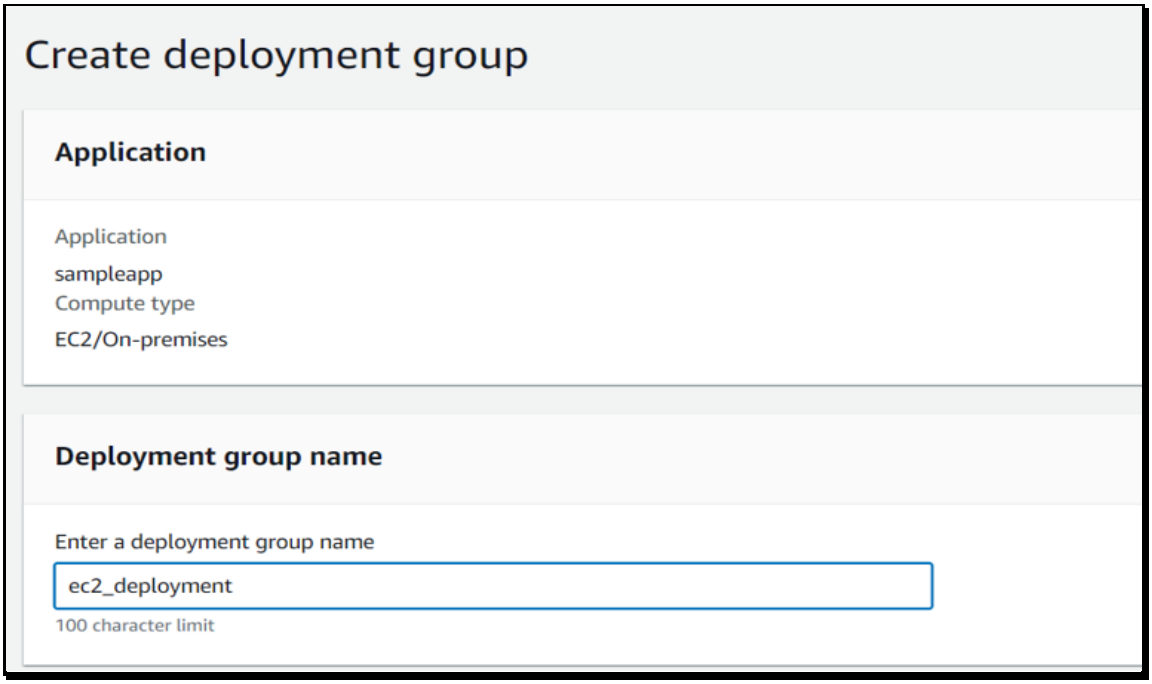
Step 11 : Create **Code Deploy** Application and Push the code to S3 bucket from Developer machine.

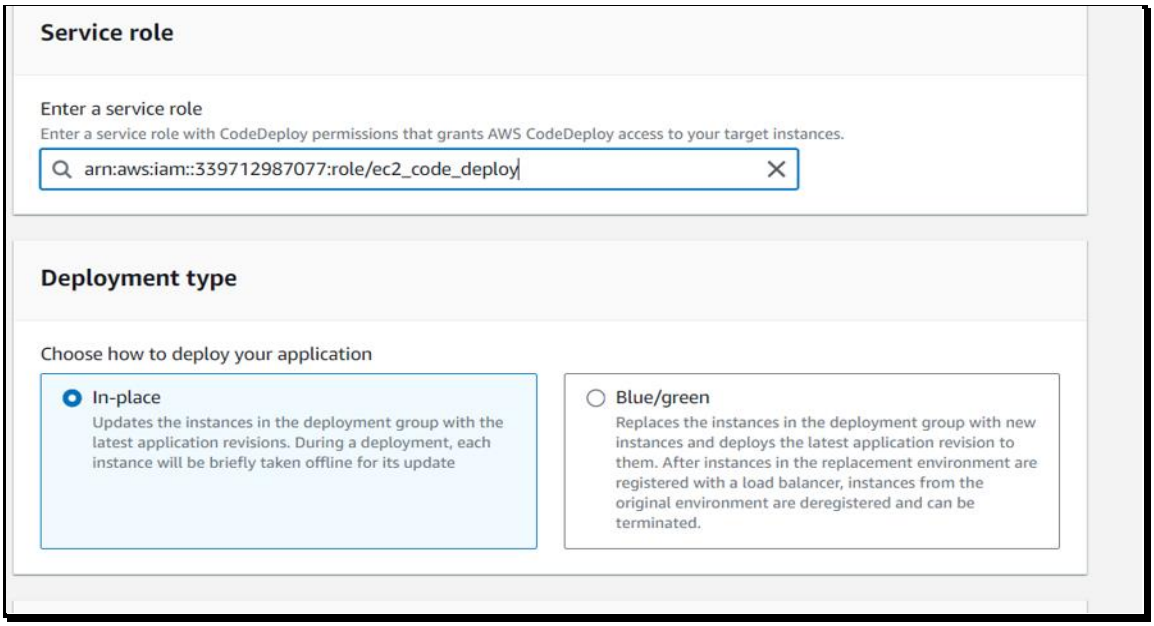
```
[root@ip-10-1-10-106 sampleapp]# mv appspec.yaml appspec.yml
[root@ip-10-1-10-106 sampleapp]# ls
appspec.yml  index.html  scripts
[root@ip-10-1-10-106 sampleapp]# aws deploy push --application-name sampleapp --s3-location s3://homebuckey/sampleapp
zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sampleapp --s3-location bucket=homebuckey,key=sampleapp.zip,bundleTyp
=zip,eTag=cdb94f395d5d3cd1c2d9d8c79ae73a3b --deployment-group-name <deployment-group-name> --deployment-config-name <
eployment-config-name> --description <description>
[root@ip-10-1-10-106 sampleapp]#
```

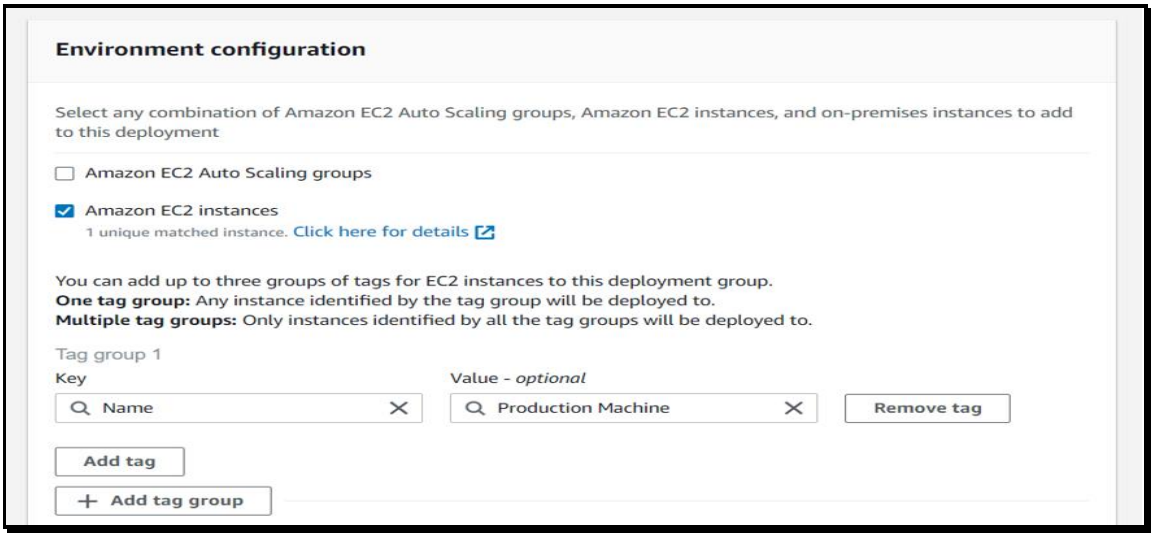
```
[root@ip-10-1-10-106 sample]# aws deploy push --application-name sample --s3-location s3://homebuckey/sample.zip

/root/deploy_dir/sampleapp/scripts/deploy/sample/appspec.yml was not found
[root@ip-10-1-10-106 sample]# mv appspec.yaml appspec.yml
[root@ip-10-1-10-106 sample]# aws deploy push --application-name sample --s3-location s3://homebuckey/sample.zip
To deploy with this revision, run:
aws deploy create-deployment --application-name sample --s3-location bucket=homebuckey,key=sample.zip,bundleType=zip,
Tag=a9684a8c21ae4ba26c340cbe0dcbe8e4,version=0TwI81aVxBSgXtv148ZiCQQS02aKbAAF --deployment-group-name <deployment-grou
p-name> --deployment-config-name <deployment-config-name> --description <description>
[root@ip-10-1-10-106 sample]#
```

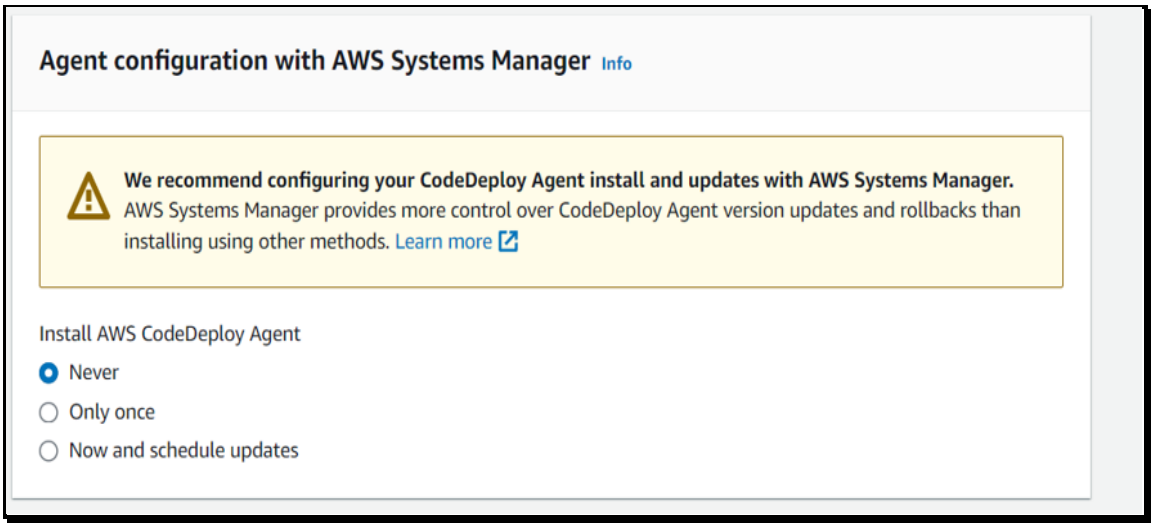
Step 12 : Create Deployment Group to include webserver.

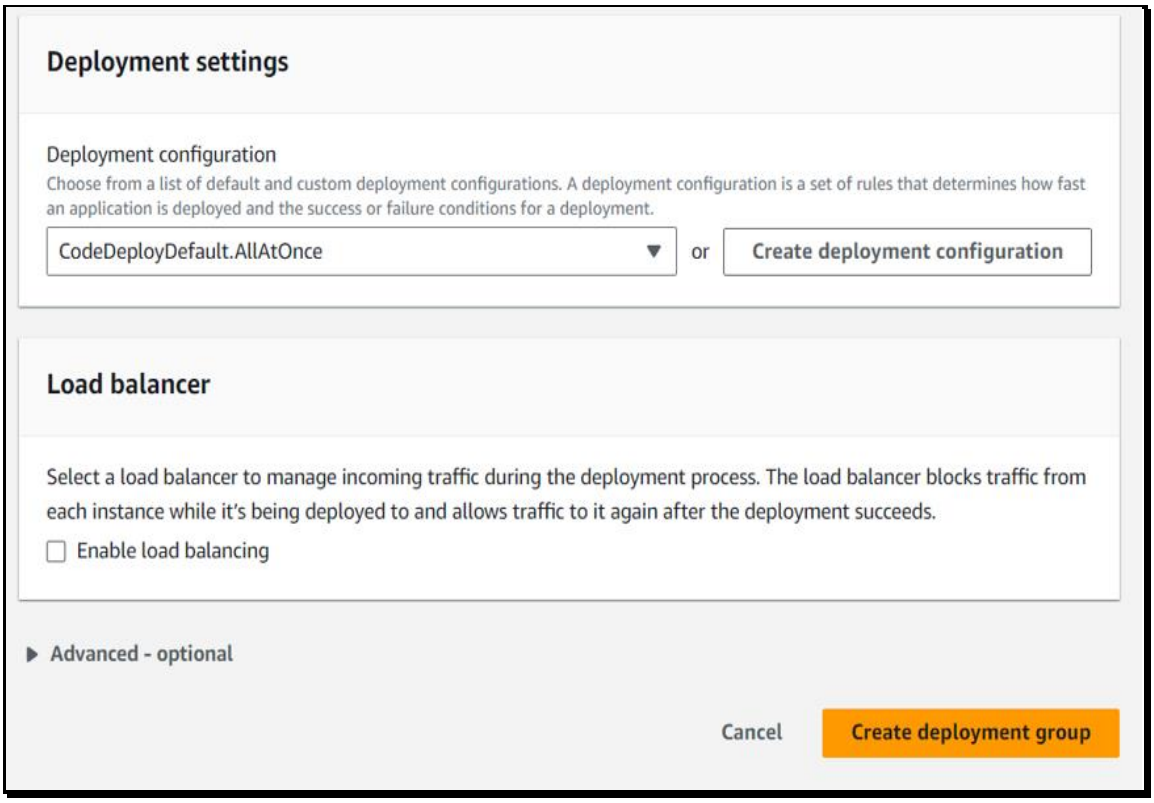







Environment configuration
Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment
Amazon EC2 Auto Scaling groups
Amazon EC2 instances
1 unique matched instance. Click here for details
You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.
Tag group 1
Key
Name
Value - optional
Production Machine
Remove tag
Add tag
Add tag group


Agent configuration with AWS Systems Manager Info
We recommend configuring your CodeDeploy Agent install and updates with AWS Systems Manager.
AWS Systems Manager provides more control over CodeDeploy Agent version updates and rollbacks than installing using other methods. Learn more
Install AWS CodeDeploy Agent
Never
Only once
Now and schedule updates


Deployment settings
Deployment configuration
Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.
CodeDeployDefault.AllAtOnce
or
Create deployment configuration
Load balancer
Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.
Enable load balancing
Advanced - optional
Cancel
Create deployment group

Application

Application

sampleapp

Compute type

EC2/On-premises

Deployment group name

Enter a deployment group name

regis

100 character limit

Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

☐ Amazon EC2 Auto Scaling groups

☑ Amazon EC2 instances

1 unique matched instance. Click here for details

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

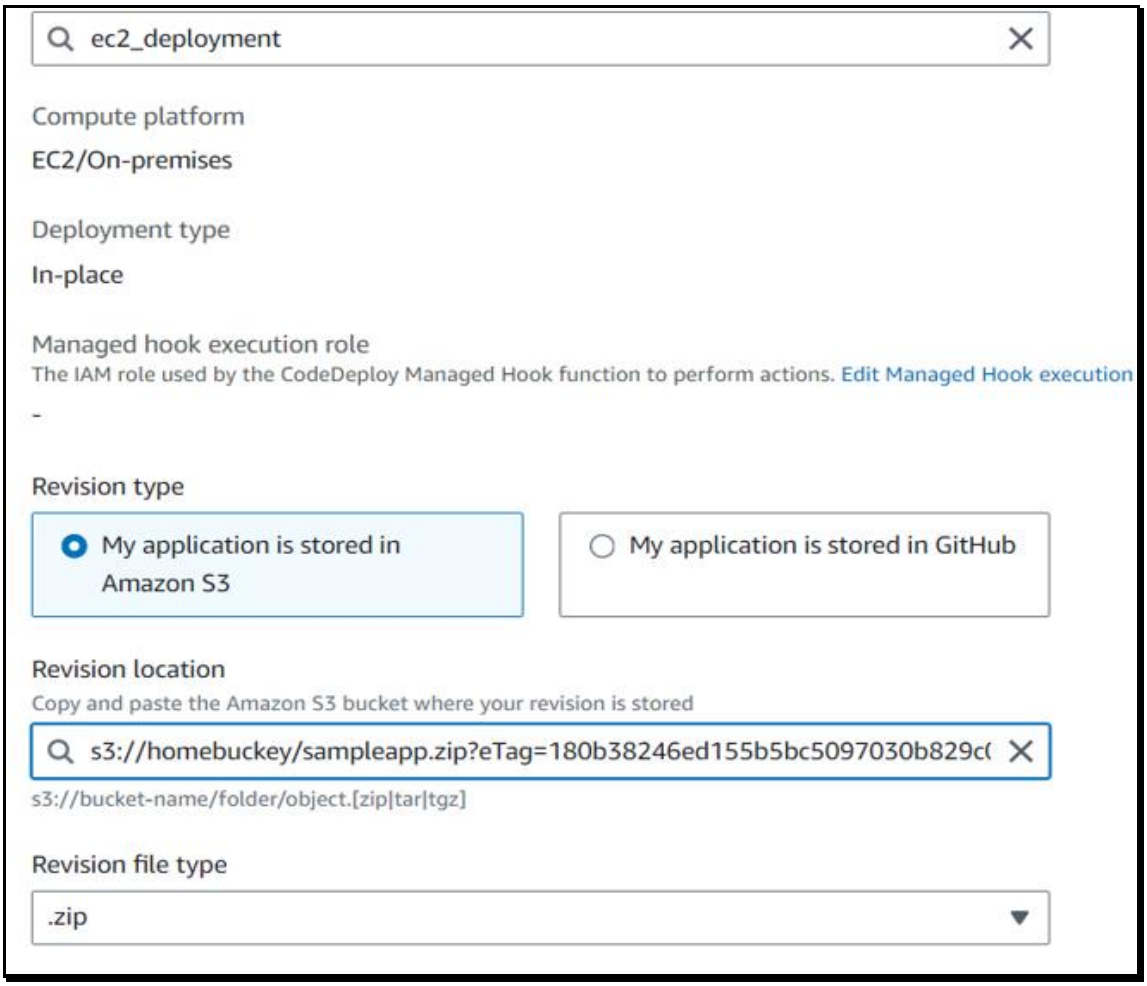
Tag group 1

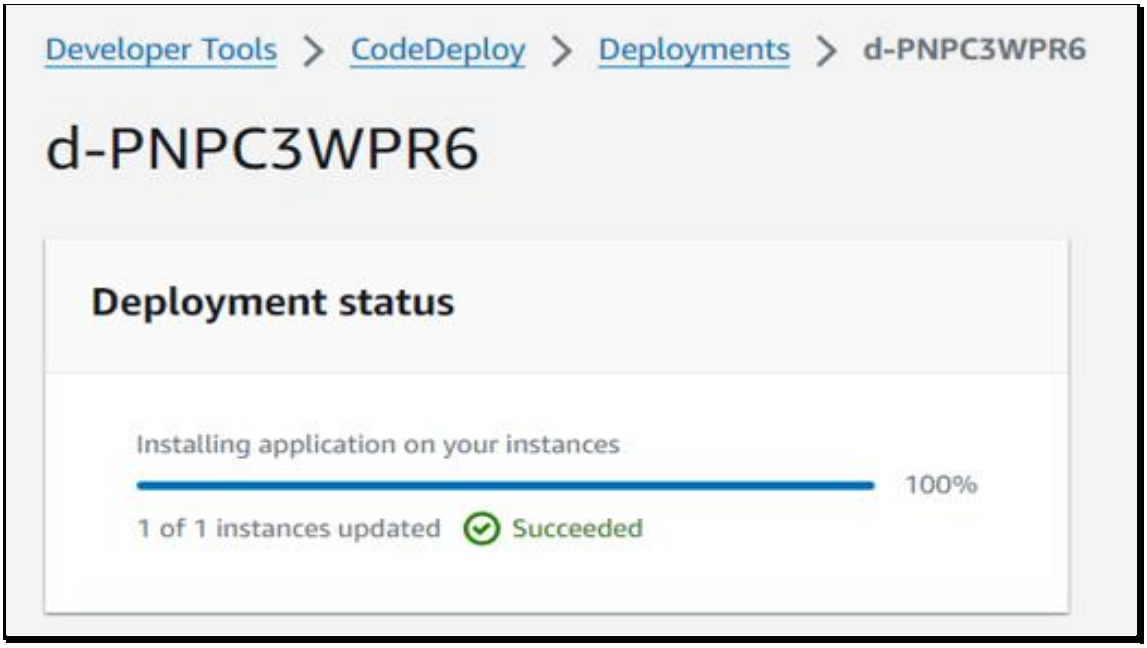
Key	Value - *optional*	
Name	production Machine 2	Remove tag

Add tag

+ Add tag group

Step 13 : Create Deployment to push the code to the webserver.





Deployment type

In-place

Managed hook execution role

The IAM role used by the CodeDeploy Managed Hook function to perform actions. Edit Managed Hook execution role.

-

Revision type

(•) My application is stored in Amazon S3

( ) My application is stored in GitHub

Revision location

Copy and paste the Amazon S3 bucket where your revision is stored

s3://homebuckey/sample.zip?eTag=ec8e3f3cfd2fa3f9ee520fb95638fa7b

s3://bucket-name/folder/object.[zip|tar|tgz]

Revision file type

.zip

Application details

Name	Compute platform
sampleapp	EC2/On-premises

Deployments | Deployment groups | Revisions

Deployment groups

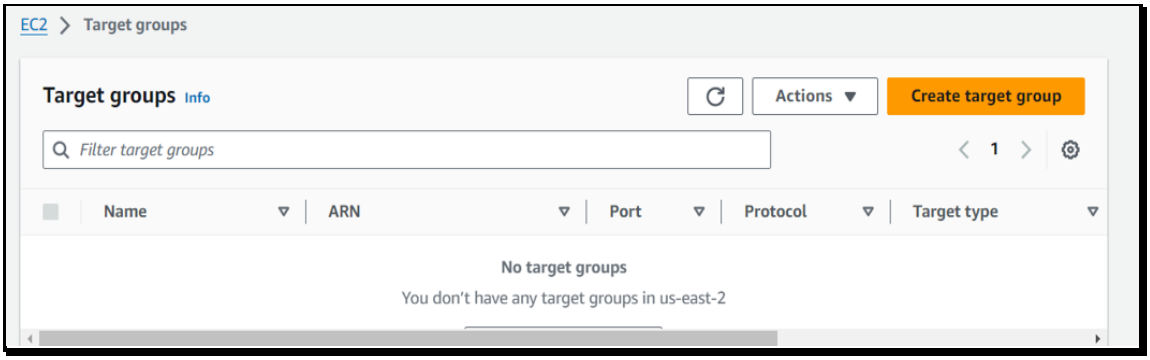
View details | Edit | Create deployment group

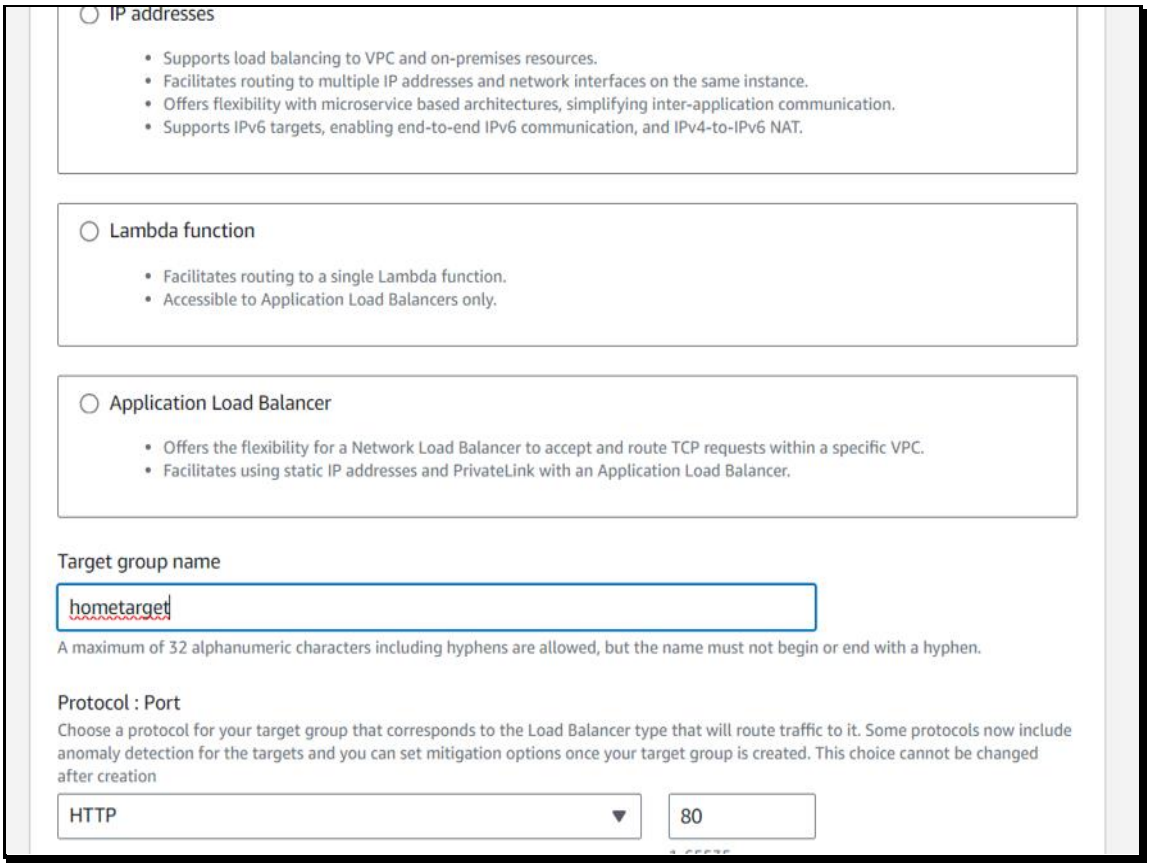
1

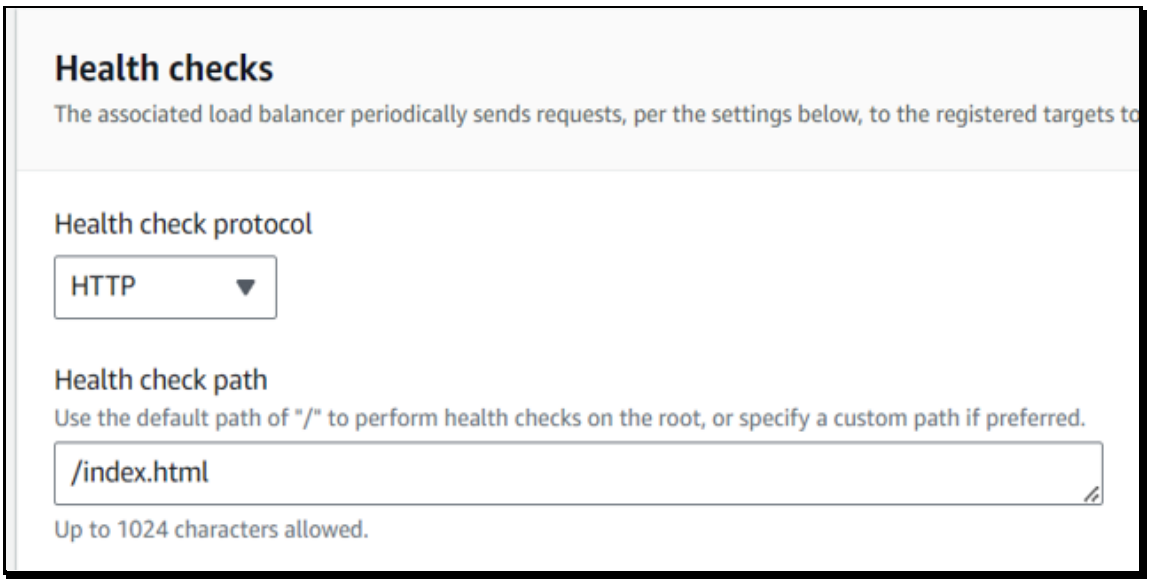
Name	Status	Last attempted d...	Last successful de...	Trigger count
regis	Succeeded	Aug 28, 2024 2:18...	Aug 28, 2024 2:18...	0
ec2_deployment	Succeeded	Aug 28, 2024 12:1...	Aug 28, 2024 12:1...	0

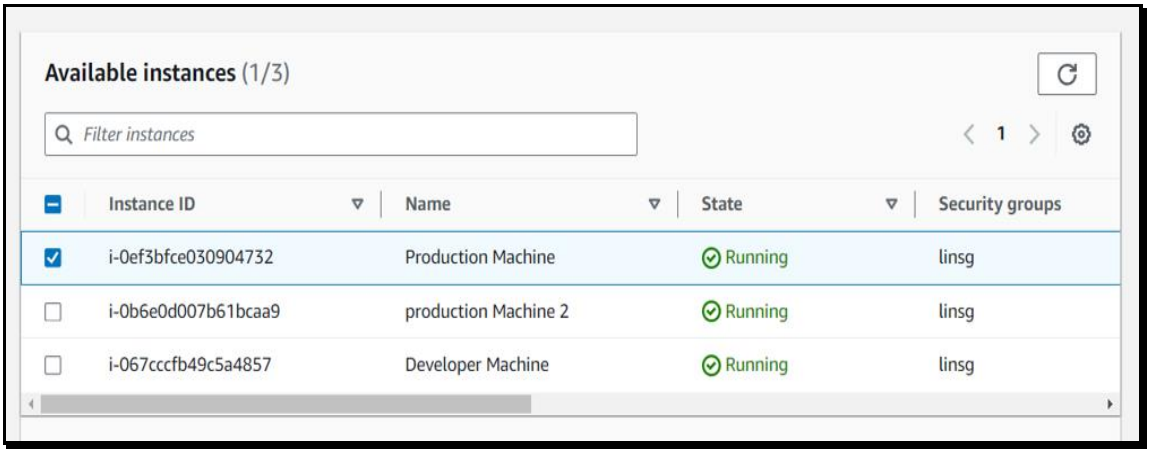
Step 14 : Create **App Load Balancer** for path based routing and **Auto-Scaling Group**.

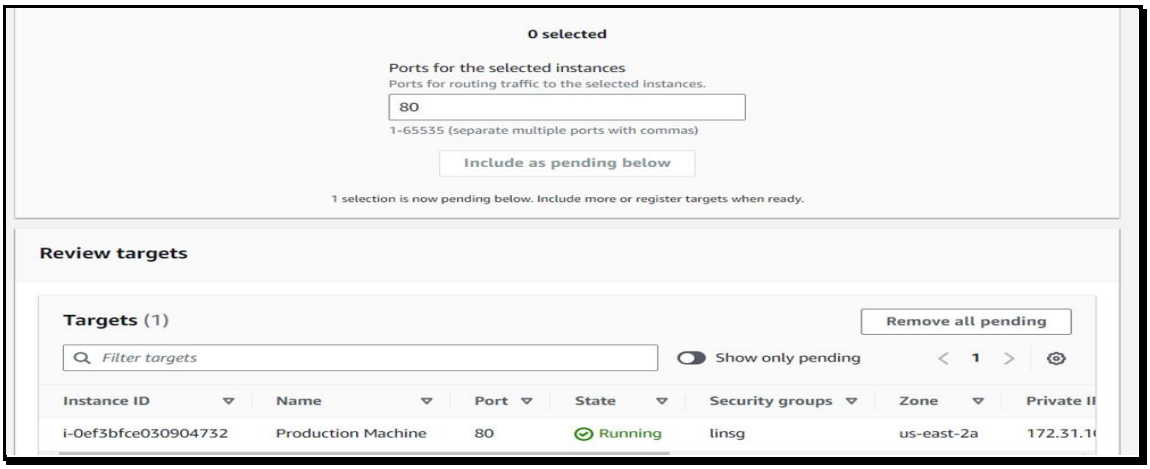
Create a 2 Target Group

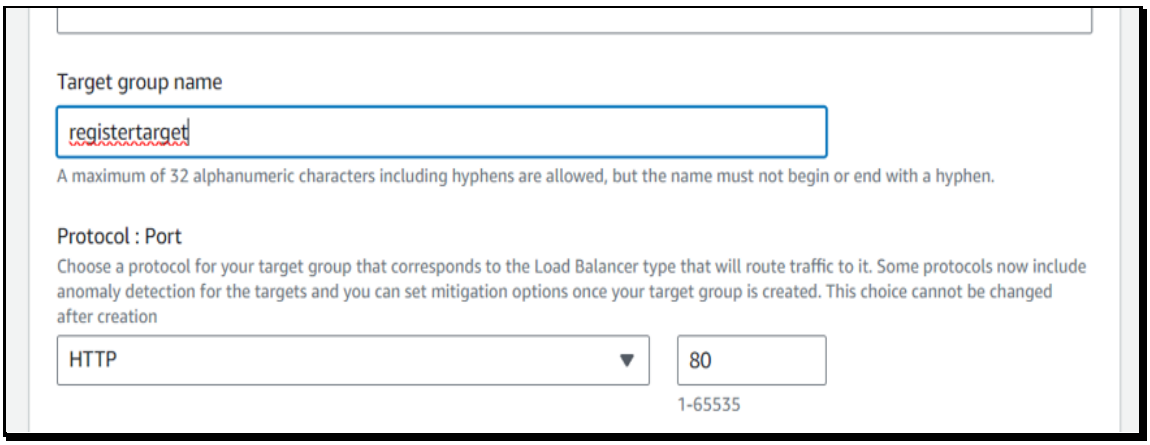




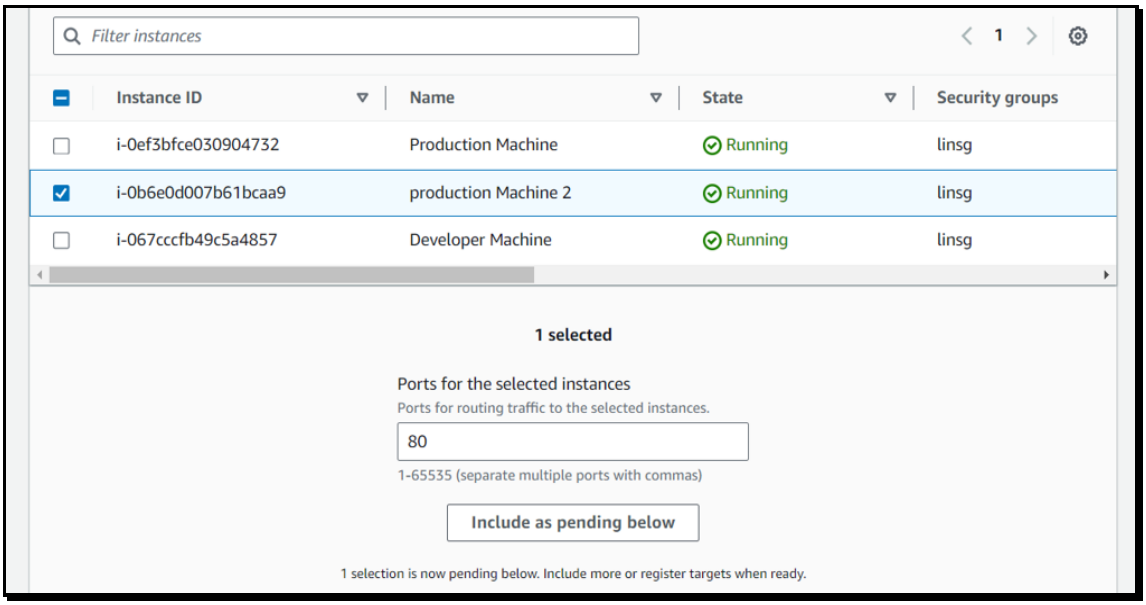


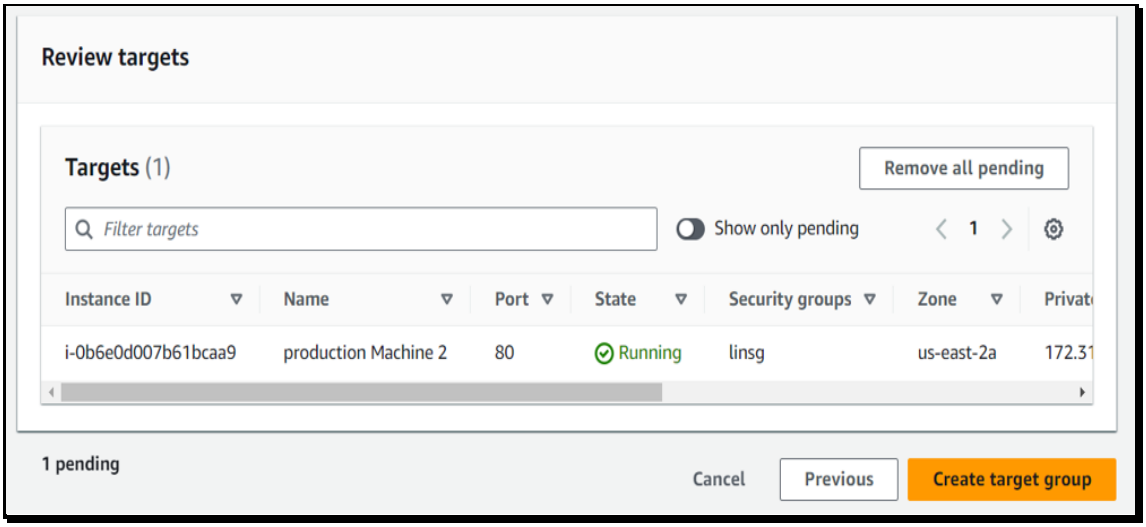

Available instances (1/3)
Filter instances
1
Instance ID
Name
State
Security groups
i-0ef3bfce030904732
Production Machine
Running
linsg
i-0b6e0d007b61bcaa9
production Machine 2
Running
linsg
i-067cccfb49c5a4857
Developer Machine
Running
linsg

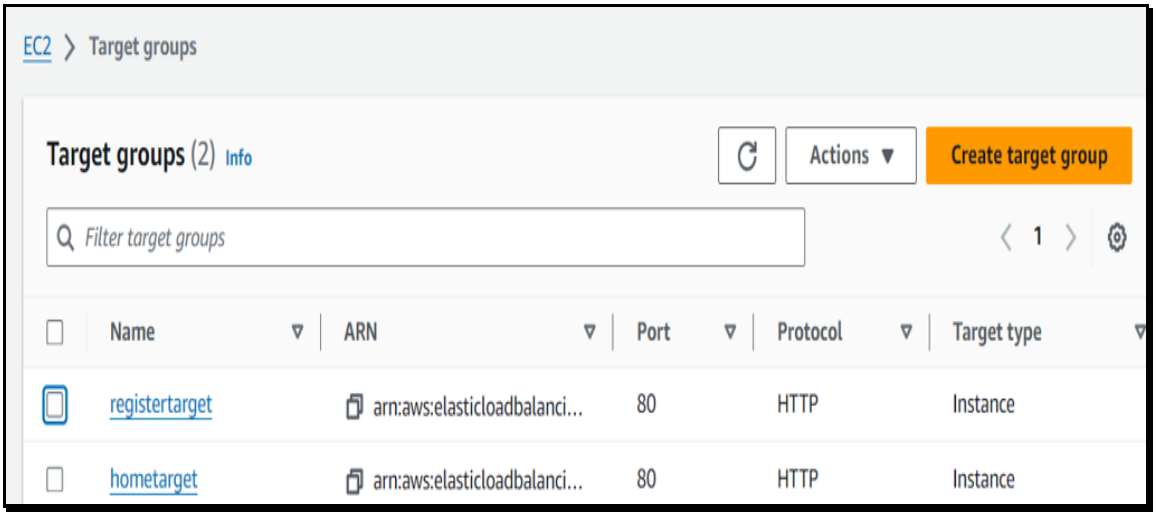

0 selected
Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)
Include as pending below
1 selection is now pending below. Include more or register targets when ready.
Review targets
Targets (1)
Remove all pending
Filter targets
Show only pending
1
Instance ID
Name
Port
State
Security groups
Zone
Private II
i-0ef3bfce030904732
Production Machine
80
Running
linsg
us-east-2a
172.31.1


Target group name
registertarget
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.
Protocol : Port
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation
HTTP
80
1-65535

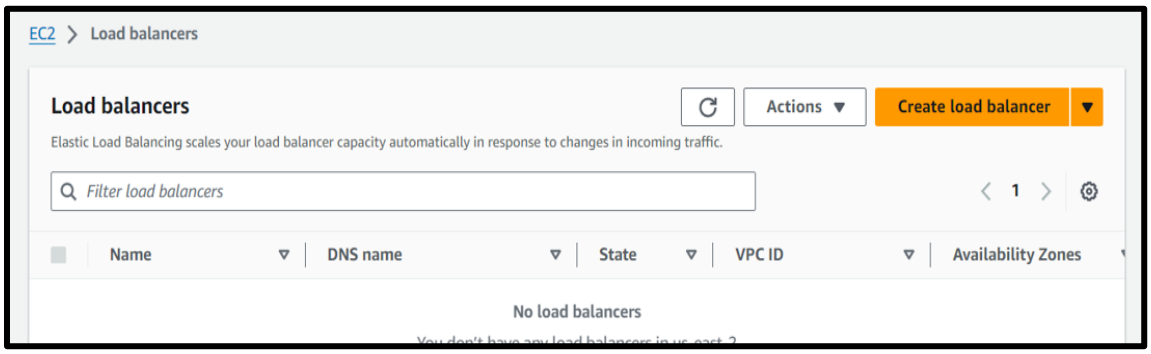

Health check protocol
HTTP
Health check path
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.
/register/index.html
Up to 1024 characters allowed.
Advanced health check settings

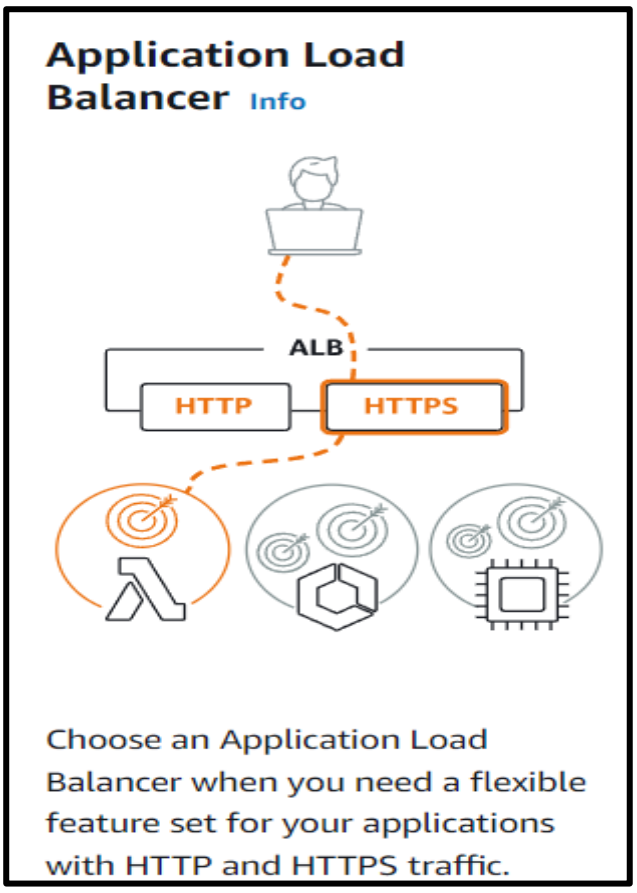

Filter instances
1
Instance ID
Name
State
Security groups
i-0ef3bfce030904732
Production Machine
Running
linsg
i-0b6e0d007b61bcaa9
production Machine 2
Running
linsg
i-067cccfb49c5a4857
Developer Machine
Running
linsg
1 selected
Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)
Include as pending below
1 selection is now pending below. Include more or register targets when ready.

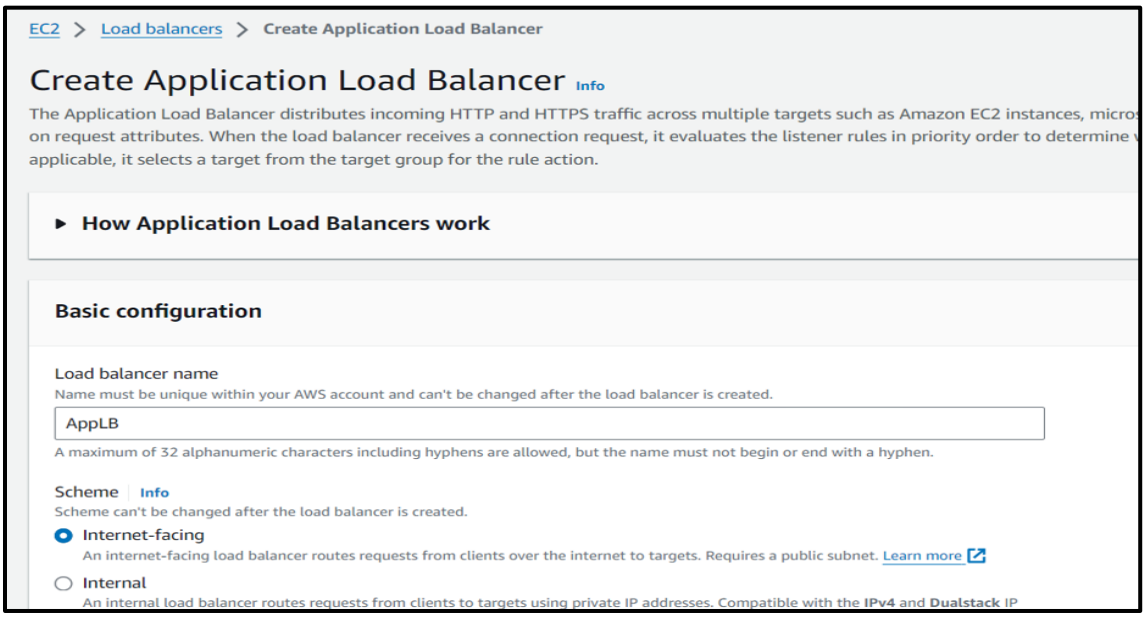

Review targets
Targets (1)
Remove all pending
Filter targets
Show only pending
1
Instance ID
Name
Port
State
Security groups
Zone
Privat
i-0b6e0d007b61bcaa9
production Machine 2
80
Running
linsg
us-east-2a
172.31
1 pending
Cancel
Previous
Create target group

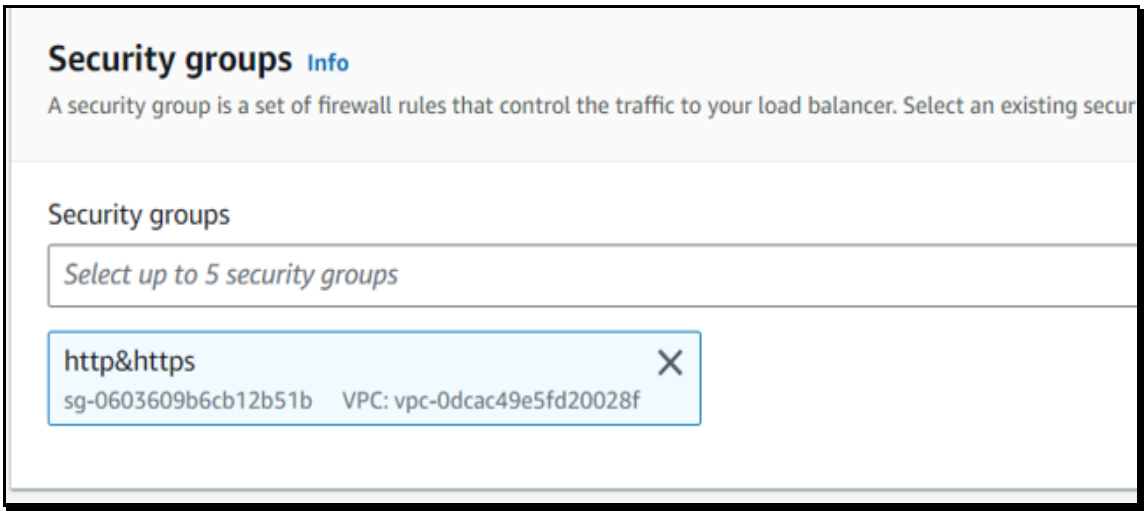

EC2
Target groups
Target groups (2) Info
Actions
Create target group
Filter target groups
1
Name
ARN
Port
Protocol
Target type
registertarget
arn:aws:elasticloadbalanci...
80
HTTP
Instance
hometarget
arn:aws:elasticloadbalanci...
80
HTTP
Instance

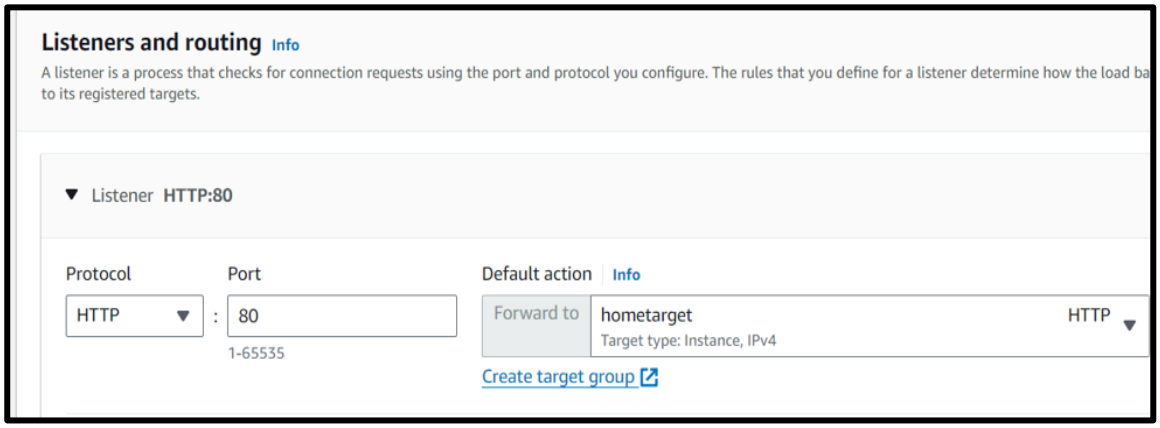
Create a APP Load Balancer for path based routing

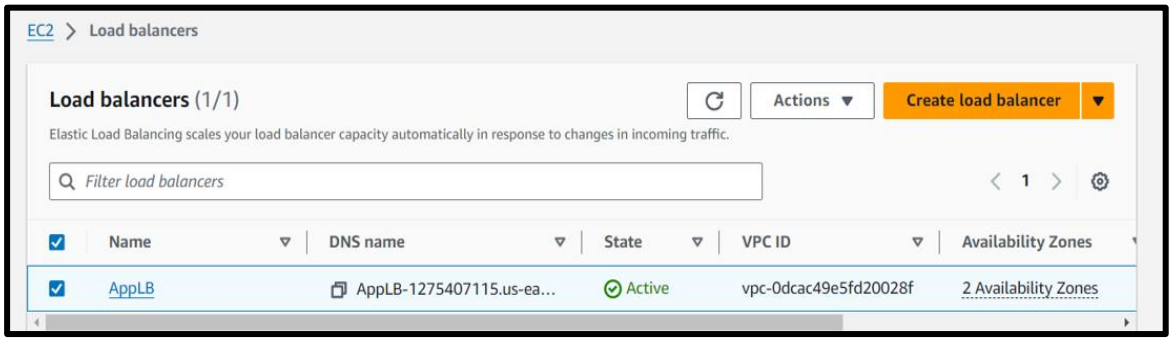


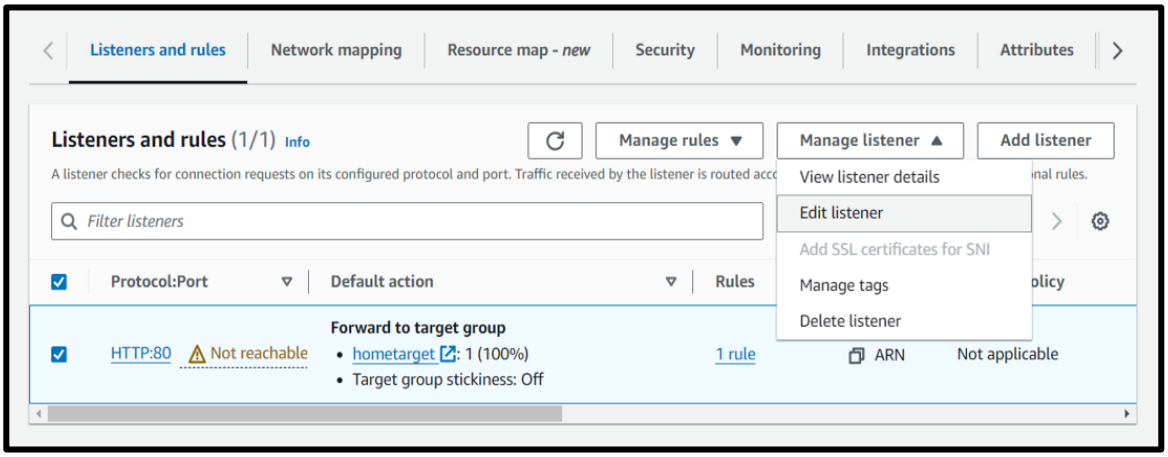


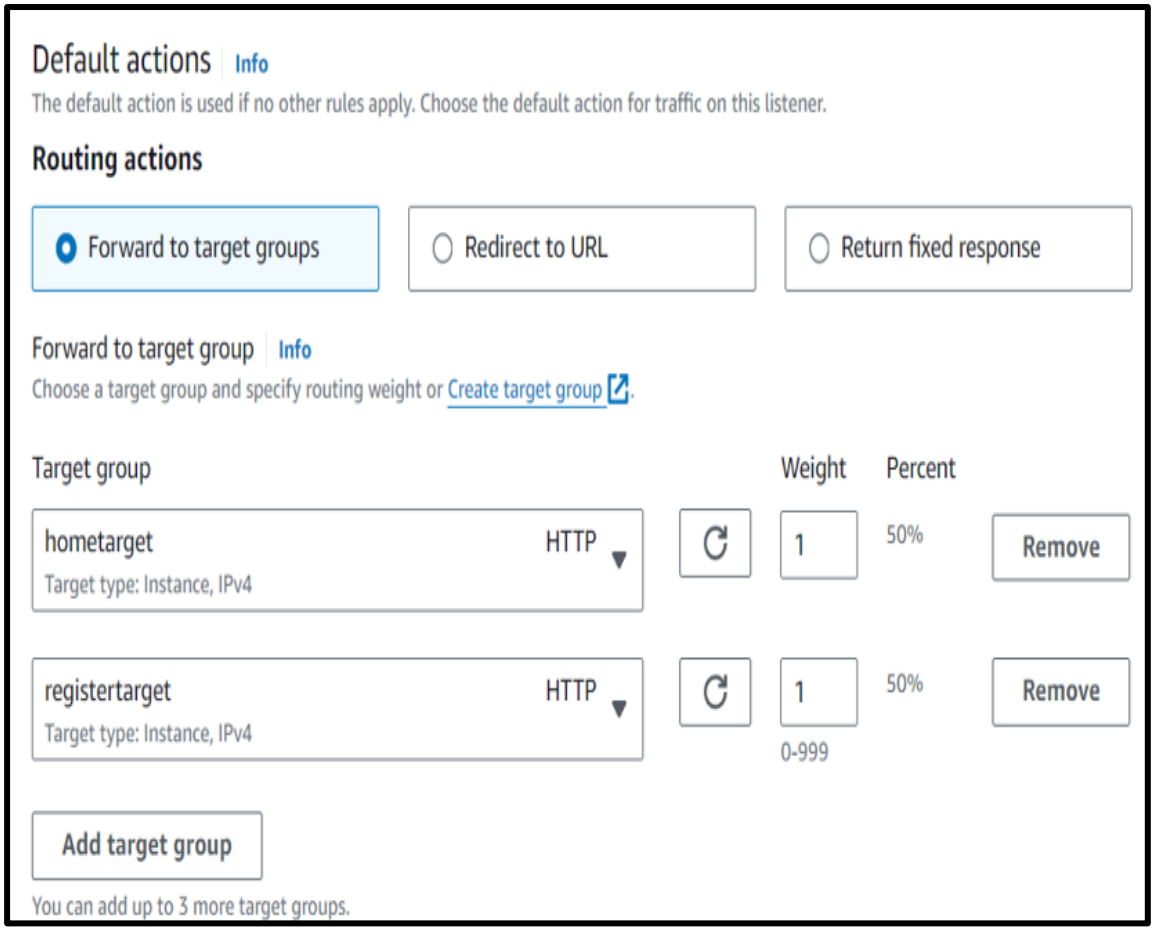


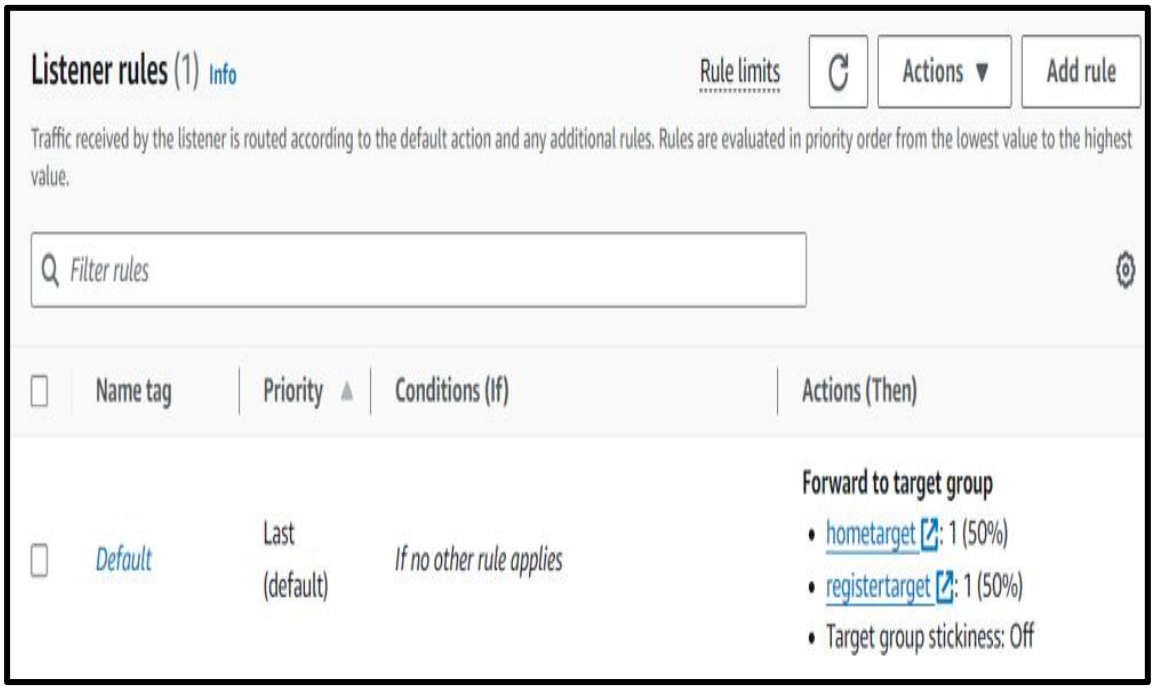

Security groups Info
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing secur
Security groups
Select up to 5 security groups
http&https
sg-0603609b6cb12b51b VPC: vpc-0dcac49e5fd20028f


Listeners and routing Info
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load ba
to its registered targets.
Listener HTTP:80
Protocol
Port
Default action Info
HTTP
80
1-65535
Forward to
hometarget
HTTP
Target type: Instance, IPv4
Create target group

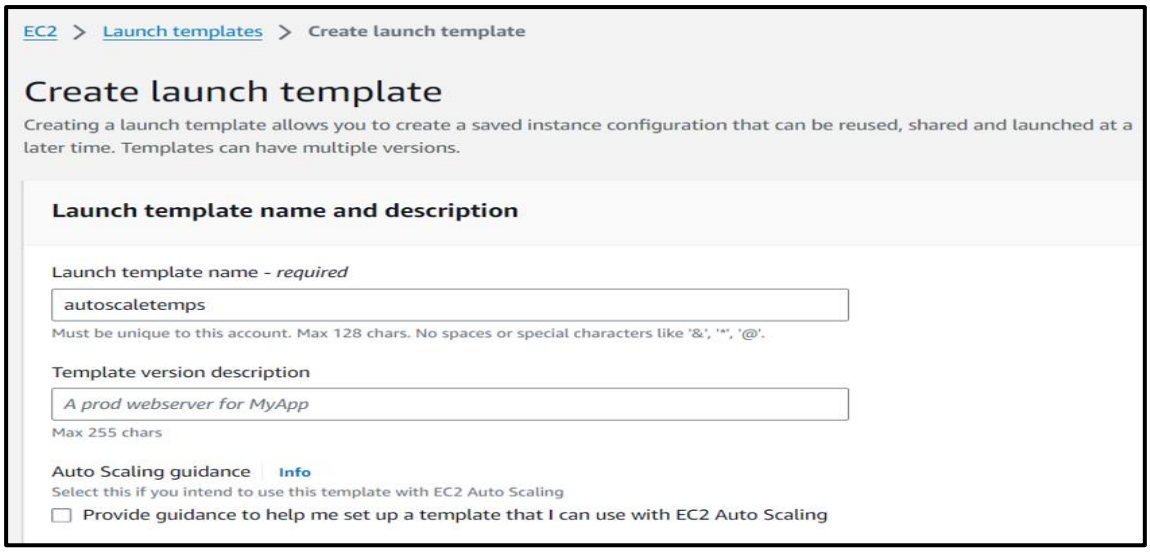

EC2 > Load balancers
Load balancers (1/1)
Actions
Create load balancer
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Filter load balancers
1
Name
DNS name
State
VPC ID
Availability Zones
AppLB
AppLB-1275407115.us-ea...
Active
vpc-0dcac49e5fd20028f
2 Availability Zones

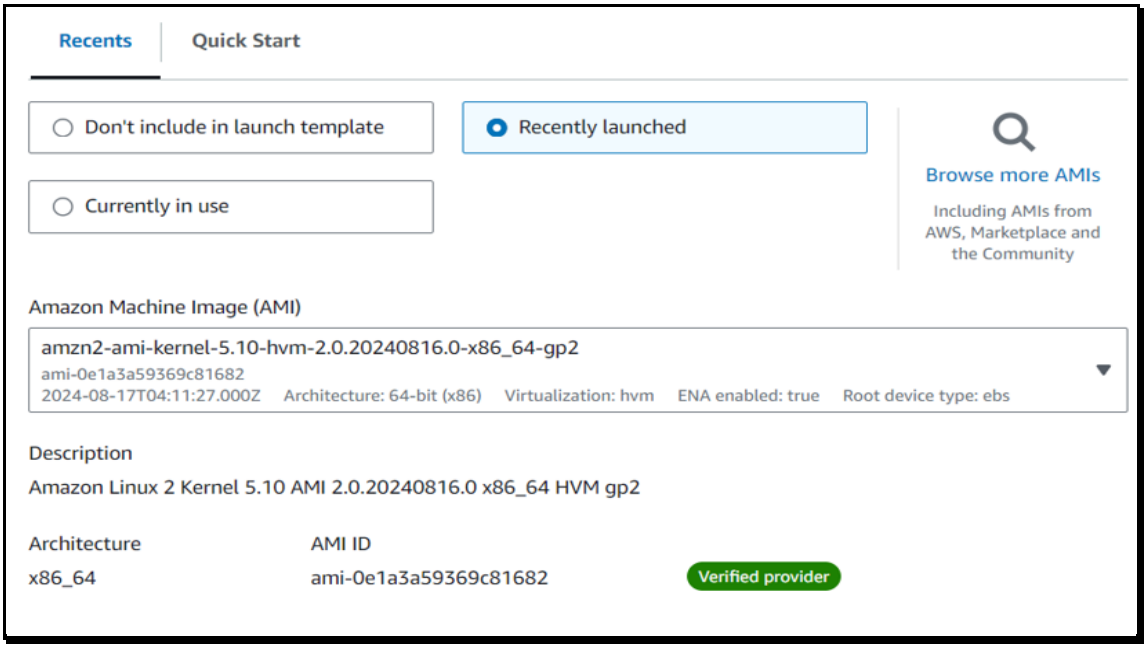

Listeners and rules
Network mapping
Resource map - new
Security
Monitoring
Integrations
Attributes
Listeners and rules (1/1) Info
Manage rules
Manage listener
Add listener
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed acc
View listener details
Edit listener
Add SSL certificates for SNI
Manage tags
Delete listener
Filter listeners
Protocol:Port
Default action
Rules
HTTP:80
Not reachable
Forward to target group
hometarget: 1 (100%)
Target group stickiness: Off
1 rule
ARN
Not applicable

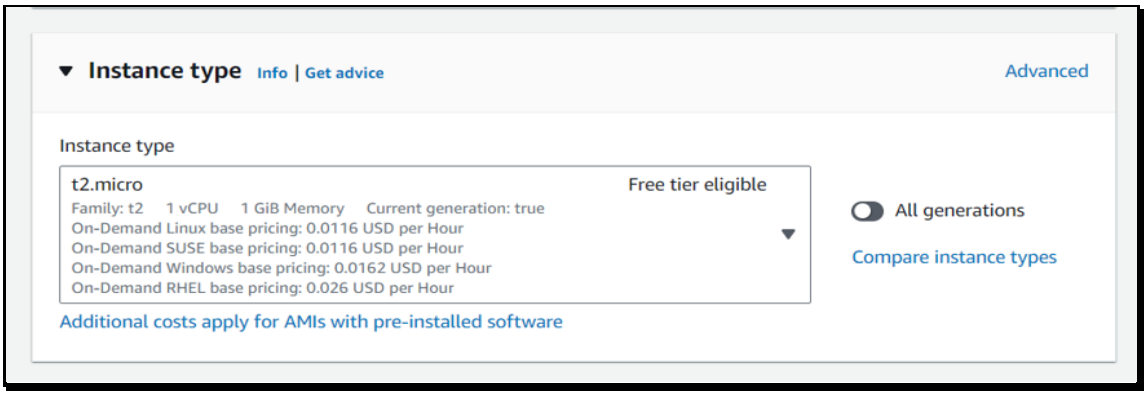

Default actions Info
The default action is used if no other rules apply. Choose the default action for traffic on this listener.
Routing actions
Forward to target groups
Redirect to URL
Return fixed response
Forward to target group Info
Choose a target group and specify routing weight or Create target group.
Target group
Weight
Percent
hometarget
HTTP
Target type: Instance, IPv4
1
50%
Remove
registertarget
HTTP
Target type: Instance, IPv4
1
50%
Remove
0-999
Add target group
You can add up to 3 more target groups.

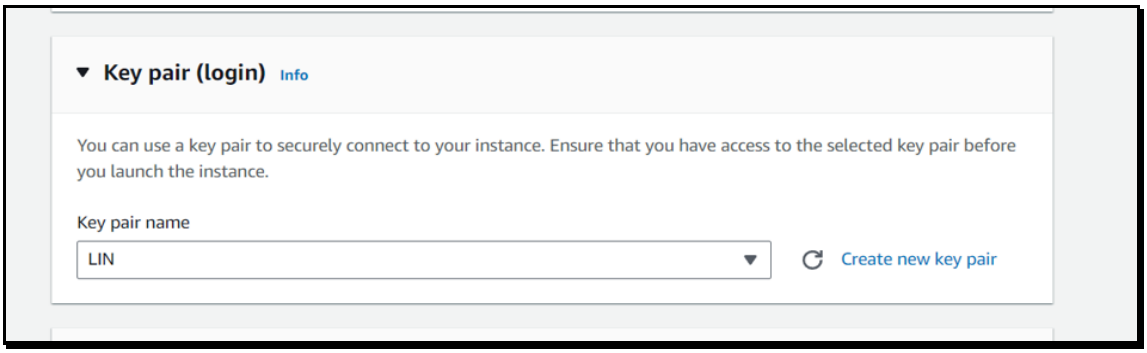

Listener rules (1) Info
Rule limits
Actions
Add rule
Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.
Filter rules
Name tag
Priority
Conditions (If)
Actions (Then)
Default
Last (default)
If no other rule applies
Forward to target group
hometarget: 1 (50%)
registertarget: 1 (50%)
Target group stickiness: Off

Create a Template

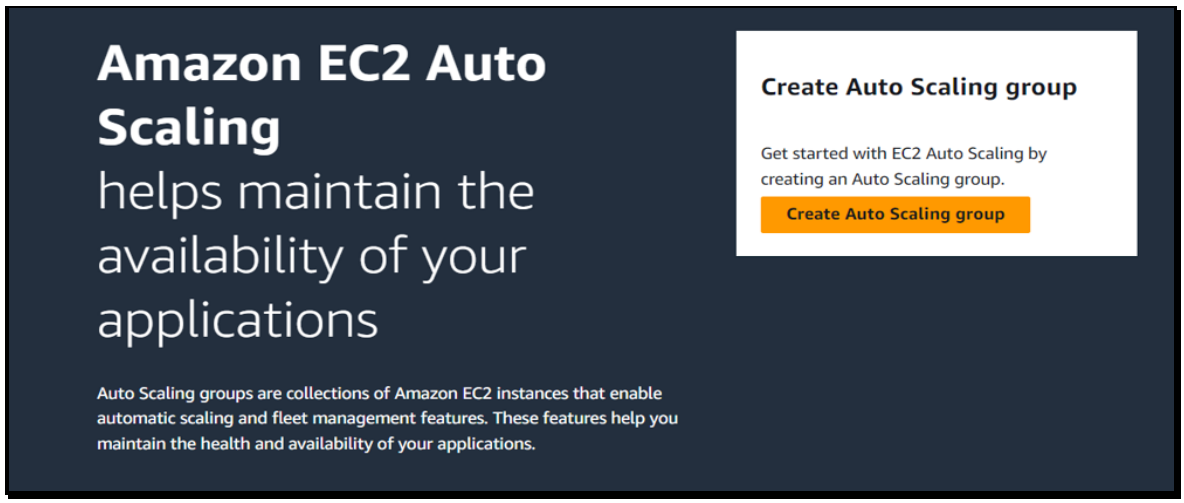


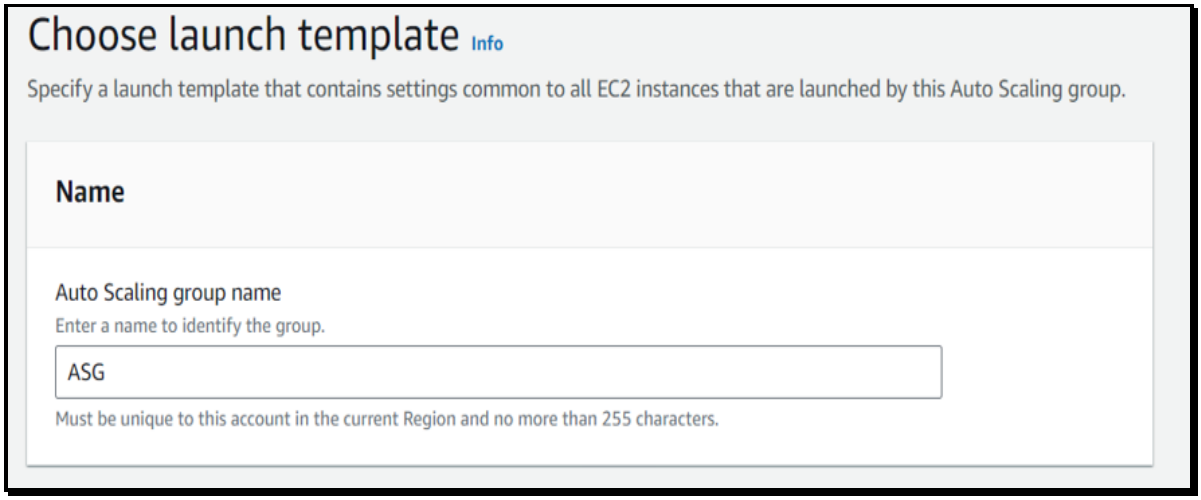


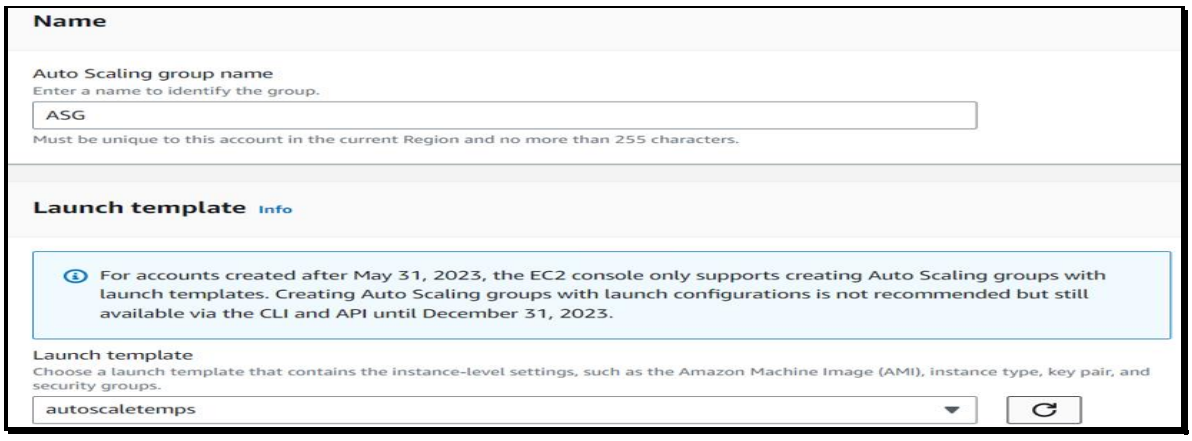




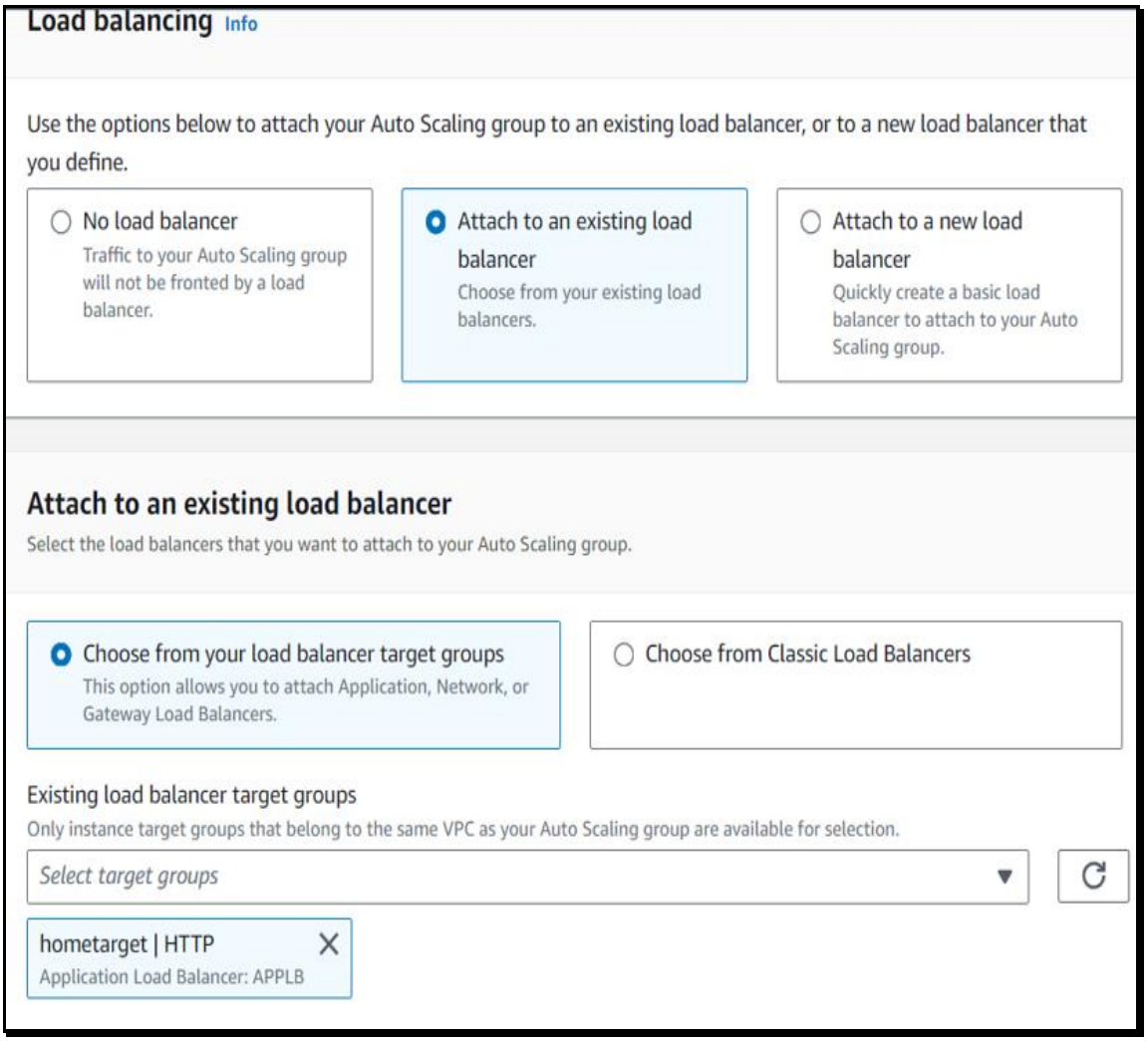
Create a Auto scaling Group

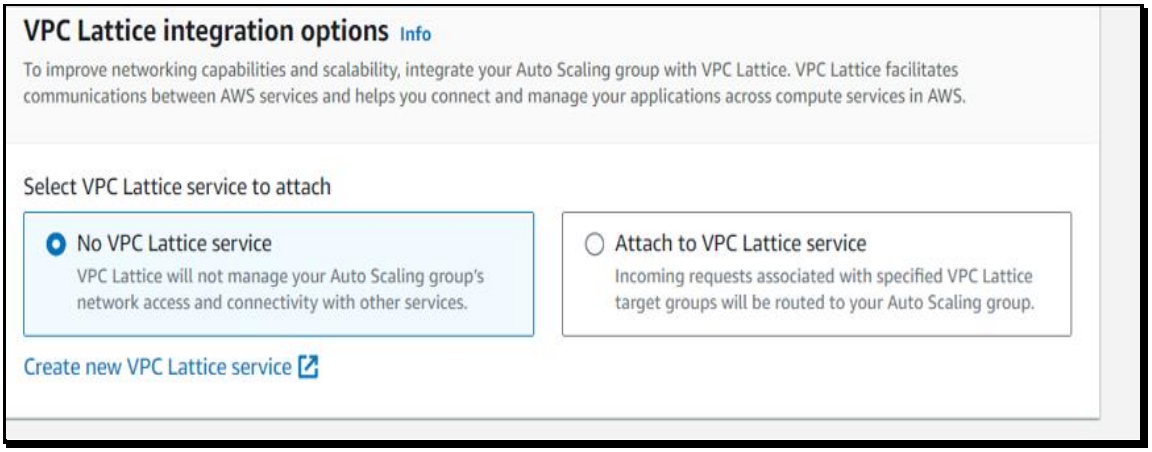


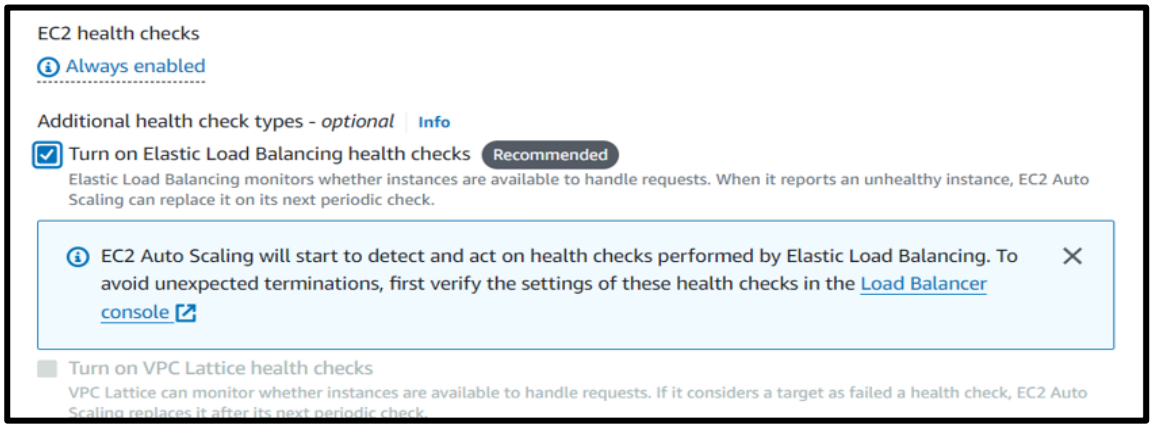


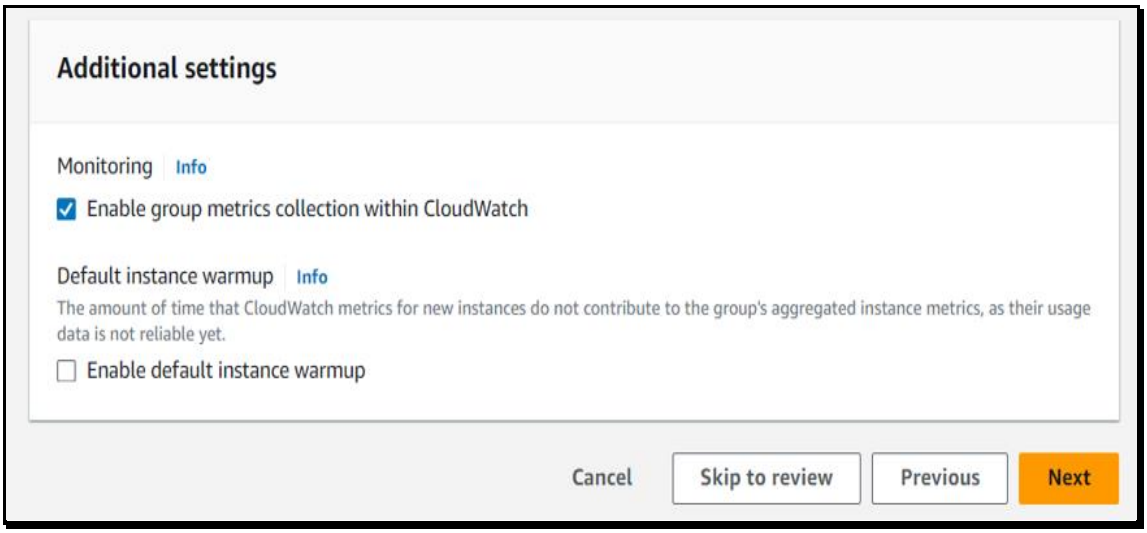


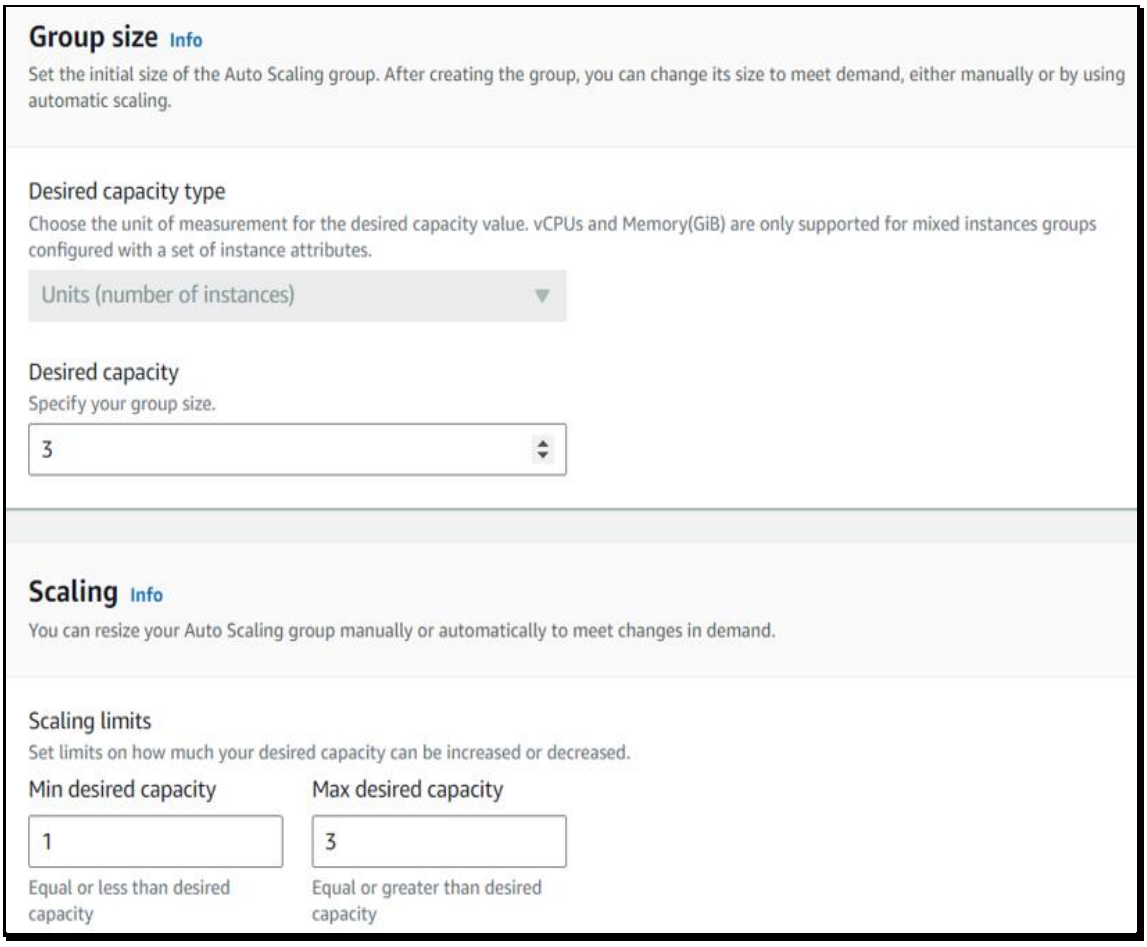
Choose a Hometarget Group in APP Load Balancer

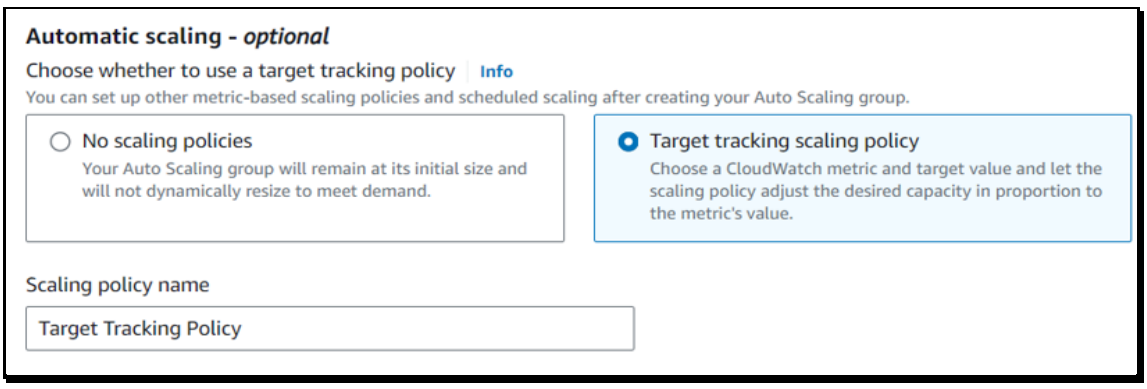




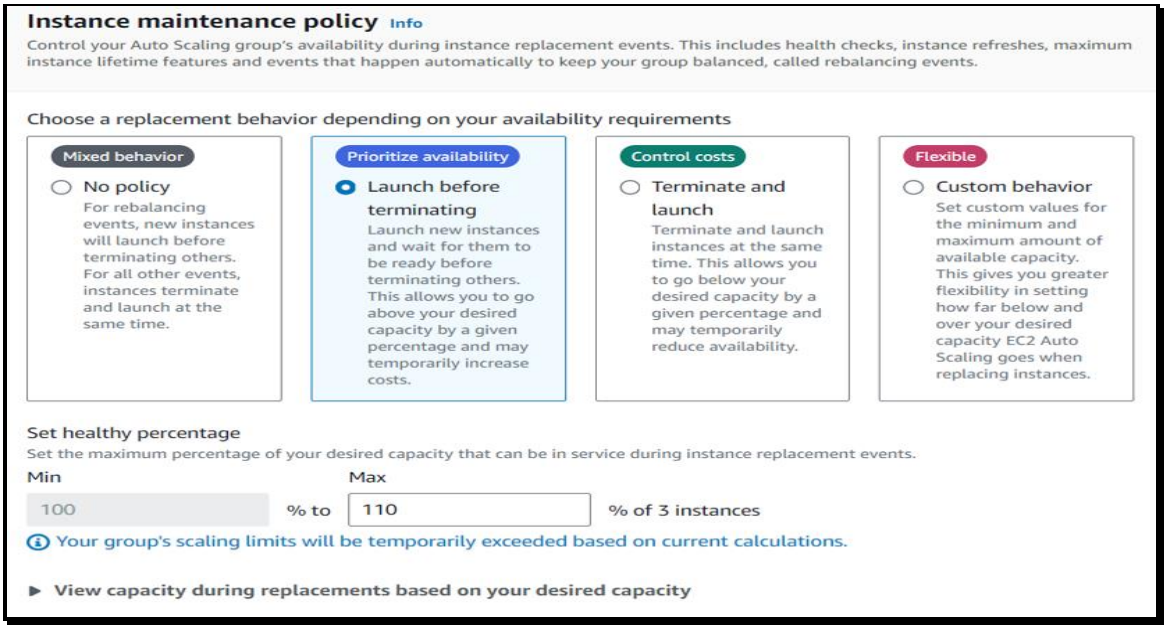


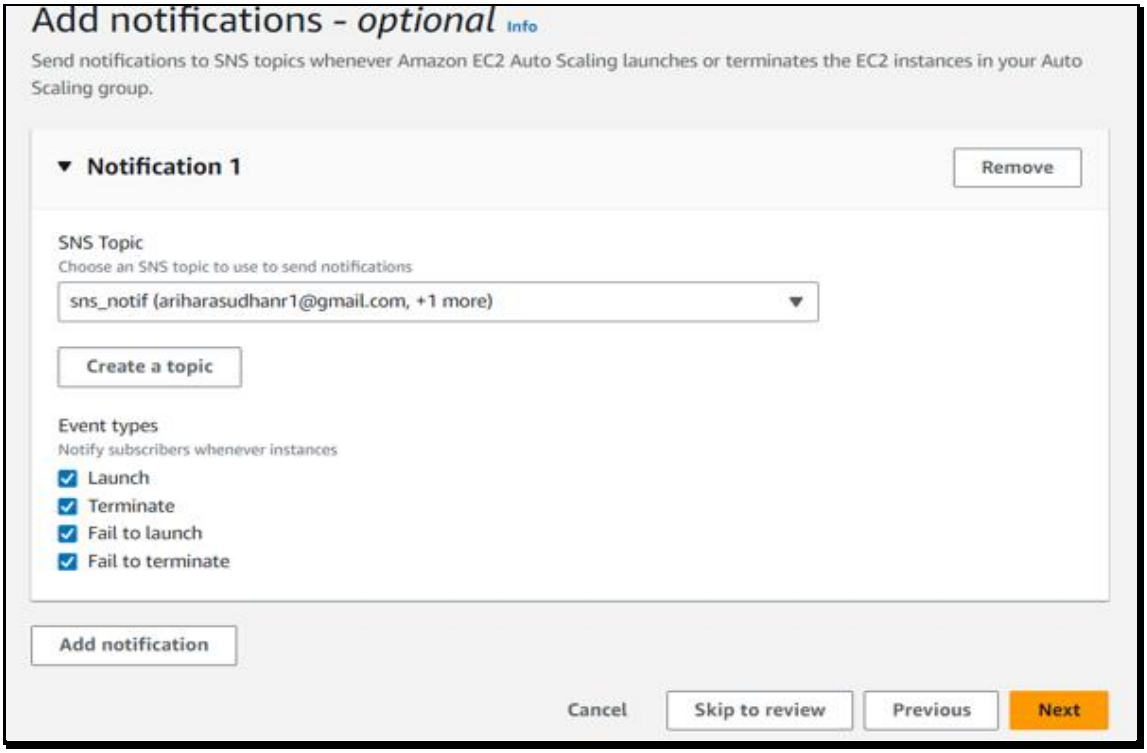

Additional settings
Monitoring Info
Enable group metrics collection within CloudWatch
Default instance warmup Info
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.
Enable default instance warmup
Cancel
Skip to review
Previous
Next

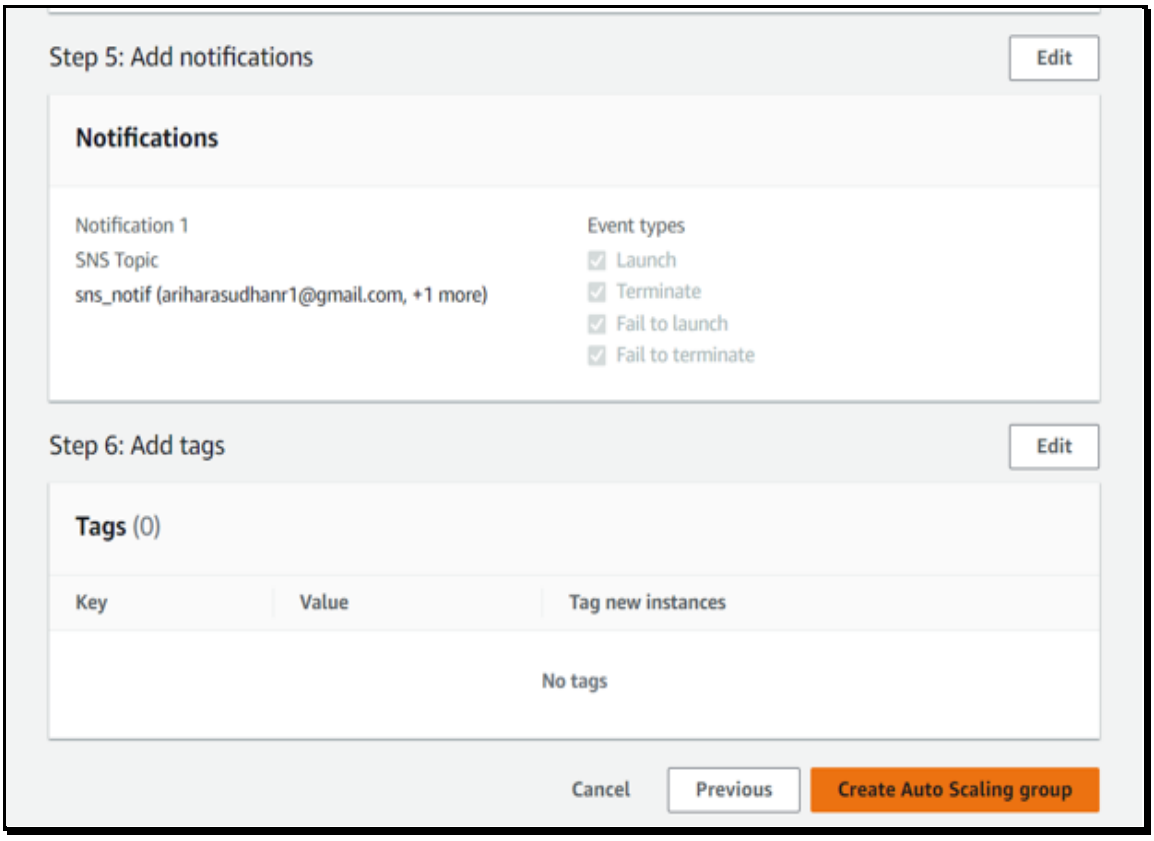

Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.
Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)
Desired capacity
Specify your group size.
3
Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.
Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
Max desired capacity
1
3
Equal or less than desired capacity
Equal or greater than desired capacity

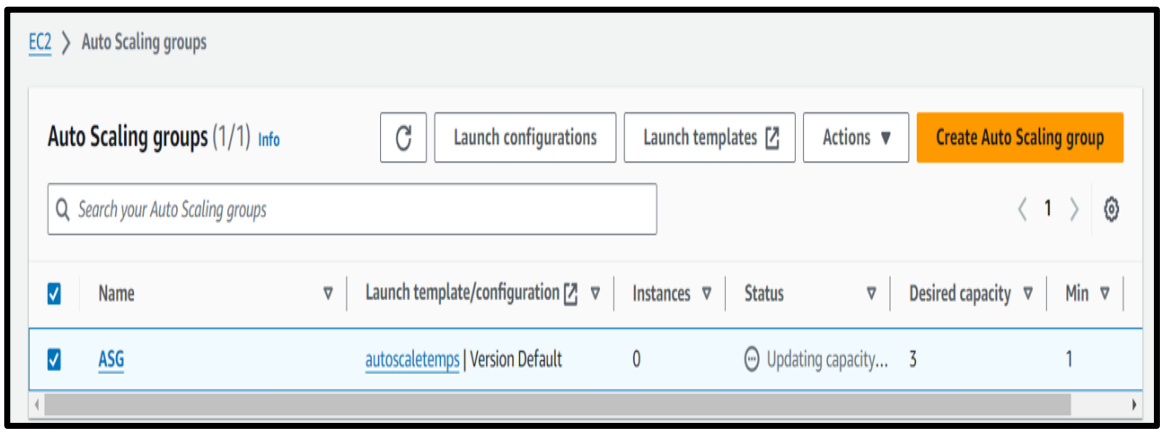

Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.
No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.
Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.
Scaling policy name
Target Tracking Policy

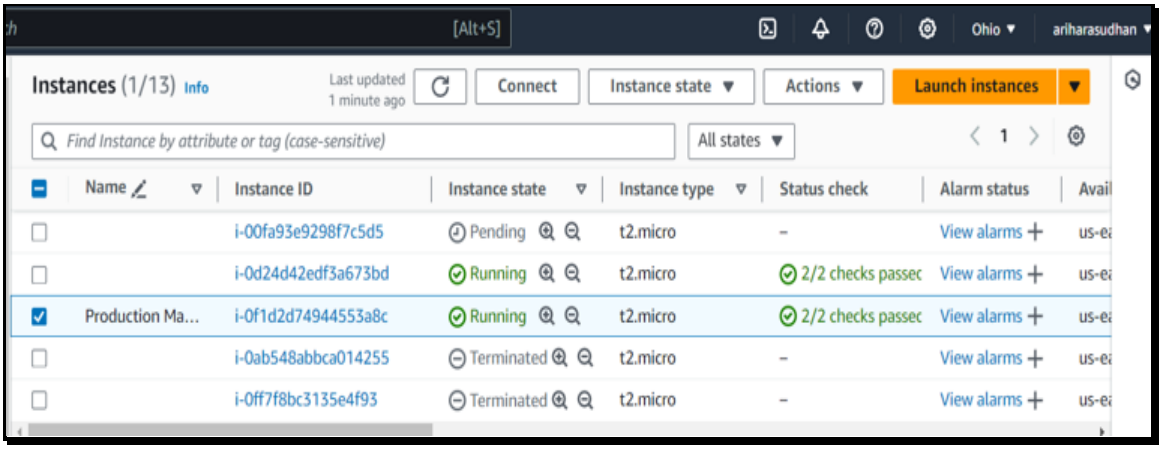

Metric type Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 m
better scaling performance.
Average CPU utilization
Target value
80
Instance warmup Info
300
seconds
Disable scale in to create only a scale-out policy

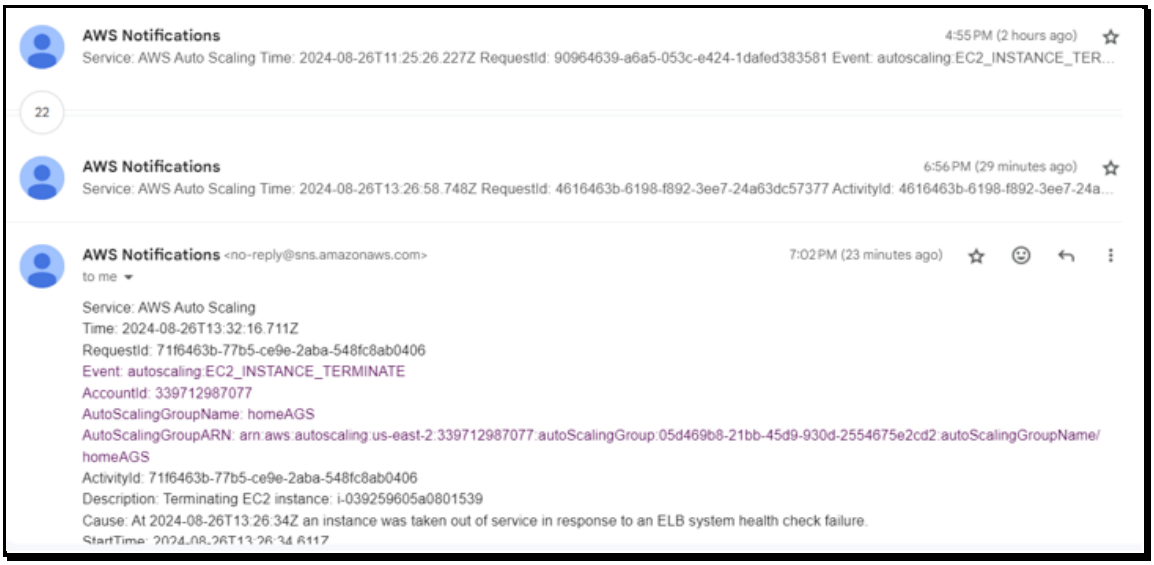

Instance maintenance policy Info
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.
Choose a replacement behavior depending on your availability requirements
Mixed behavior
No policy
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.
Prioritize availability
Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.
Control costs
Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.
Flexible
Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.
Set healthy percentage
Set the maximum percentage of your desired capacity that can be in service during instance replacement events.
Min
Max
100
% to
110
% of 3 instances
Your group's scaling limits will be temporarily exceeded based on current calculations.
View capacity during replacements based on your desired capacity

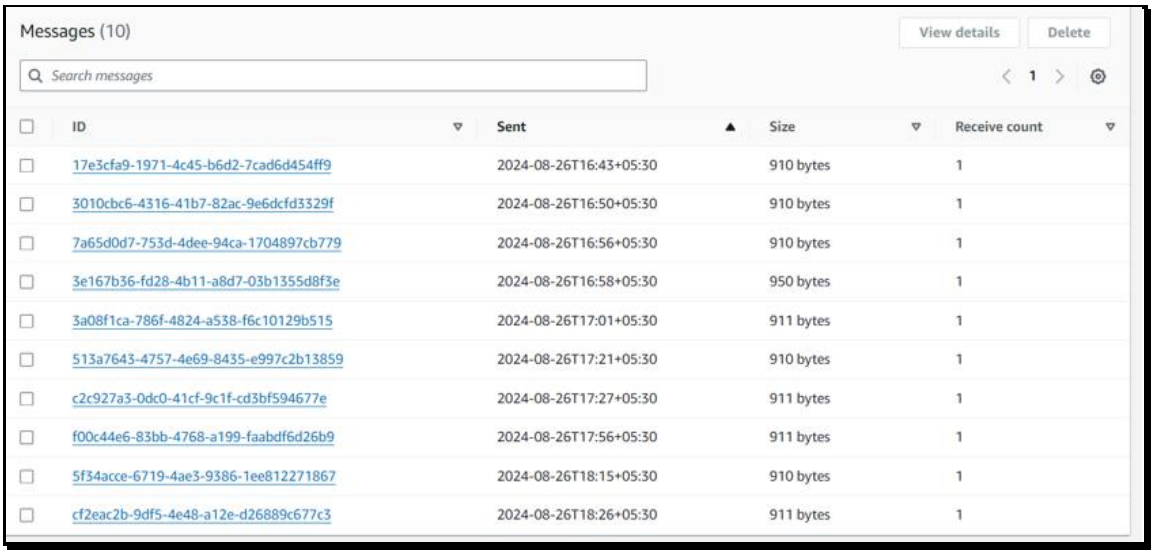

Add notifications - optional Info
Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.
Notification 1
Remove
SNS Topic
Choose an SNS topic to use to send notifications
sns_notif (ariharasudhanr1@gmail.com, +1 more)
Create a topic
Event types
Notify subscribers whenever instances
Launch
Terminate
Fail to launch
Fail to terminate
Add notification
Cancel
Skip to review
Previous
Next

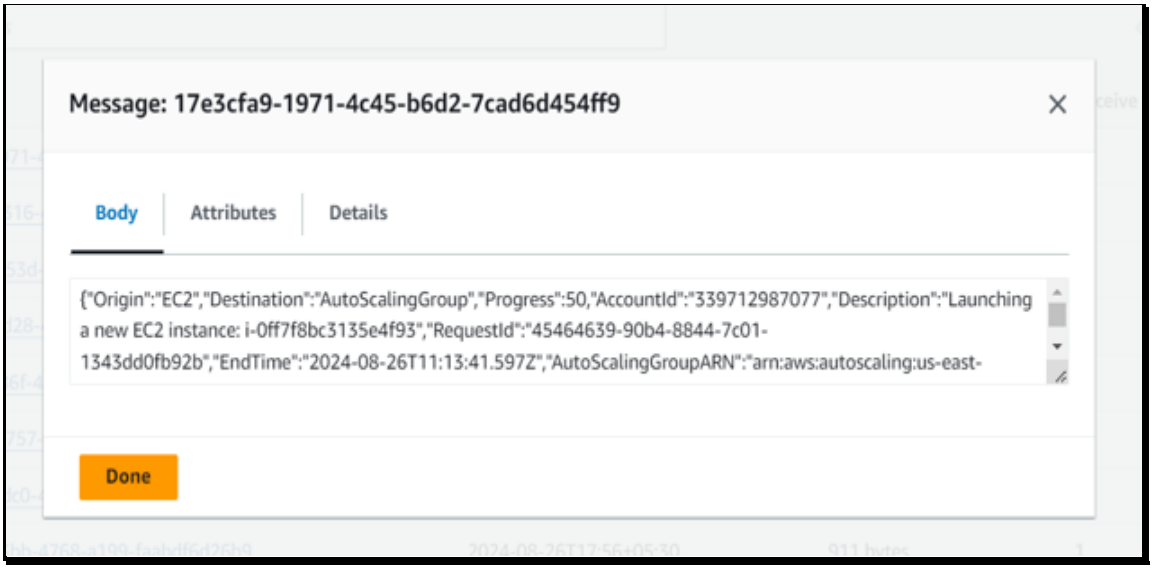

Step 5: Add notifications
Edit
Notifications
Notification 1
SNS Topic
sns_notif (ariharasudhanr1@gmail.com, +1 more)
Event types
Launch
Terminate
Fail to launch
Fail to terminate
Step 6: Add tags
Edit
Tags (0)
Key
Value
Tag new instances
No tags
Cancel
Previous
Create Auto Scaling group


EC2 > Auto Scaling groups
Auto Scaling groups (1/1) Info
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
1
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
ASG
autoscaletemps | Version Default
0
Updating capacity...
3
1

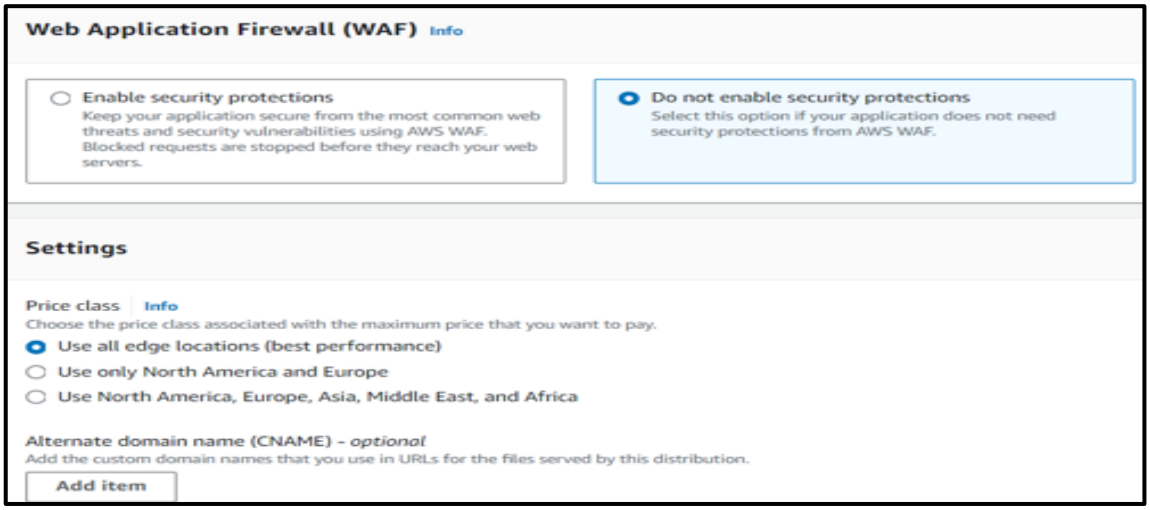

[Alt+S]
Ohio
ariharasudhan
Instances (1/13) Info
Last updated 1 minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
i-00fa93e9298f7c5d5
Pending
t2.micro
View alarms
i-0d24d42edf3a673bd
Running
t2.micro
2/2 checks passec
View alarms
Production Ma...
i-0f1d2d74944553a8c
Running
t2.micro
2/2 checks passec
View alarms
i-0ab548abbca014255
Terminated
t2.micro
View alarms
i-0ff7f8bc3135e4f93
Terminated
t2.micro
View alarms


AWS Notifications
4:55 PM (2 hours ago)
Service: AWS Auto Scaling Time: 2024-08-26T11:25:26.227Z RequestId: 90964639-a6a5-053c-e424-1dafed383581 Event: autoscaling:EC2_INSTANCE_TER...
22
AWS Notifications
6:56 PM (29 minutes ago)
Service: AWS Auto Scaling Time: 2024-08-26T13:26:58.748Z RequestId: 4616463b-6198-f892-3ee7-24a63dc57377 ActivityId: 4616463b-6198-f892-3ee7-24a...
AWS Notifications <no-reply@sns.amazonaws.com>
7:02 PM (23 minutes ago)
to me
Service: AWS Auto Scaling
Time: 2024-08-26T13:32:16.711Z
RequestId: 71f6463b-77b5-ce9e-2aba-548fc8ab0406
Event: autoscaling:EC2_INSTANCE_TERMINATE
AccountId: 339712987077
AutoScalingGroupName: homeAGS
AutoScalingGroupARN: arn:aws:autoscaling:us-east-2:339712987077:autoScalingGroup:05d469b8-21bb-45d9-930d-2554675e2cd2:autoScalingGroupName/homeAGS
ActivityId: 71f6463b-77b5-ce9e-2aba-548fc8ab0406
Description: Terminating EC2 instance: i-039259605a0801539
Cause: At 2024-08-26T13:26:34Z an instance was taken out of service in response to an ELB system health check failure.
StartTime: 2024-08-26T13:26:34.611Z

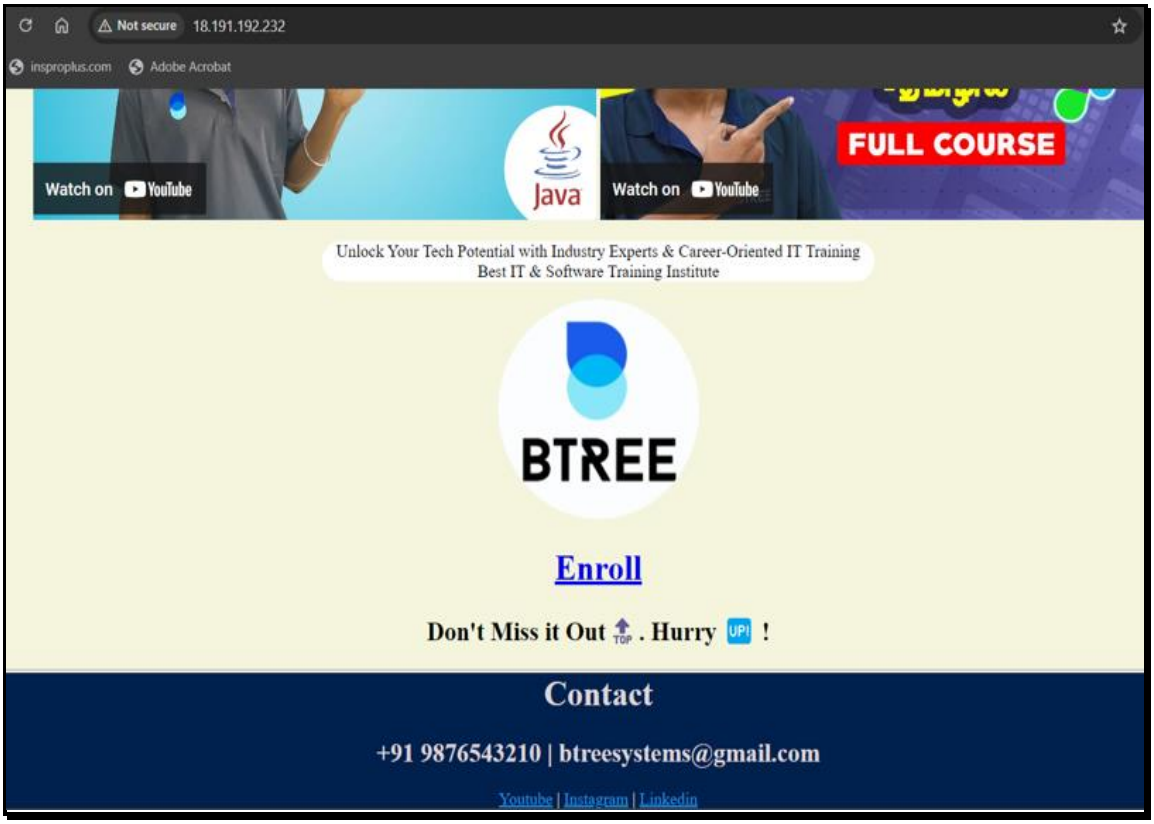

Messages (10)
View details
Delete
Search messages
1
ID
Sent
Size
Receive count
17e3cfa9-1971-4c45-b6d2-7cad6d454ff9 2024-08-26T16:43+05:30 910 bytes 1
3010cbc6-4316-41b7-82ac-9e6dcfd3329f 2024-08-26T16:50+05:30 910 bytes 1
7a65d0d7-753d-4dee-94ca-1704897cb779 2024-08-26T16:56+05:30 910 bytes 1
3e167b36-fd28-4b11-a8d7-03b1355d8f3e 2024-08-26T16:58+05:30 950 bytes 1
3a08f1ca-786f-4824-a538-f6c10129b515 2024-08-26T17:01+05:30 911 bytes 1
513a7643-4757-4e69-8435-e997c2b13859 2024-08-26T17:21+05:30 910 bytes 1
c2c927a3-0dc0-41cf-9c1f-cd3bf594677e 2024-08-26T17:27+05:30 911 bytes 1
f00c44e6-83bb-4768-a199-faabdf6d26b9 2024-08-26T17:56+05:30 911 bytes 1
5f34acce-6719-4ae3-9386-1ee812271867 2024-08-26T18:15+05:30 910 bytes 1
cf2eac2b-9df5-4e48-a12e-d26889c677c3 2024-08-26T18:26+05:30 911 bytes 1

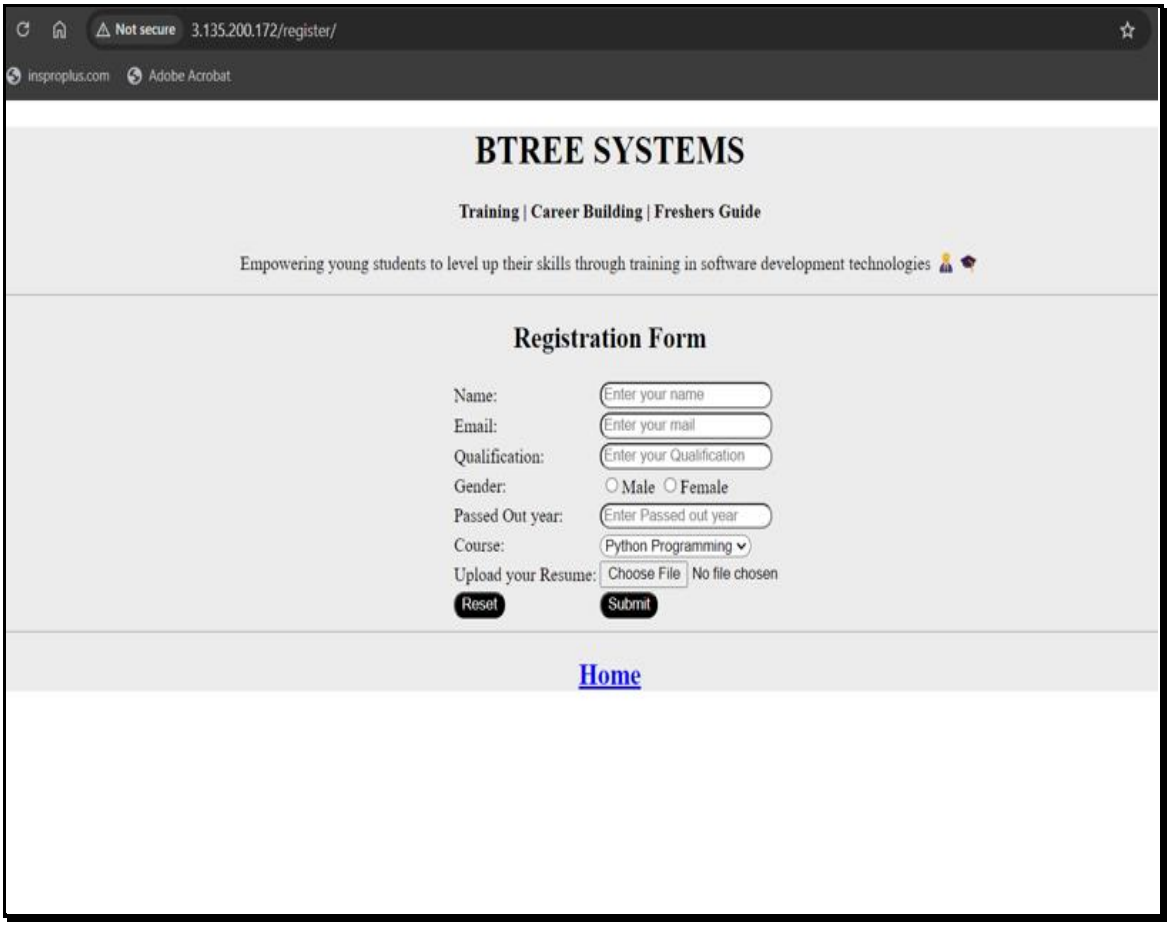

Message: 17e3cfa9-1971-4c45-b6d2-7cad6d454ff9
Body
Attributes
Details
{"Origin":"EC2","Destination":"AutoScalingGroup","Progress":50,"AccountId":"339712987077","Description":"Launching a new EC2 instance: i-0ff7f8bc3135e4f93","RequestId":"45464639-90b4-8844-7c01-1343dd0fb92b","EndTime":"2024-08-26T11:13:41.597Z","AutoScalingGroupARN":"arn:aws:autoscaling:us-east-
Done

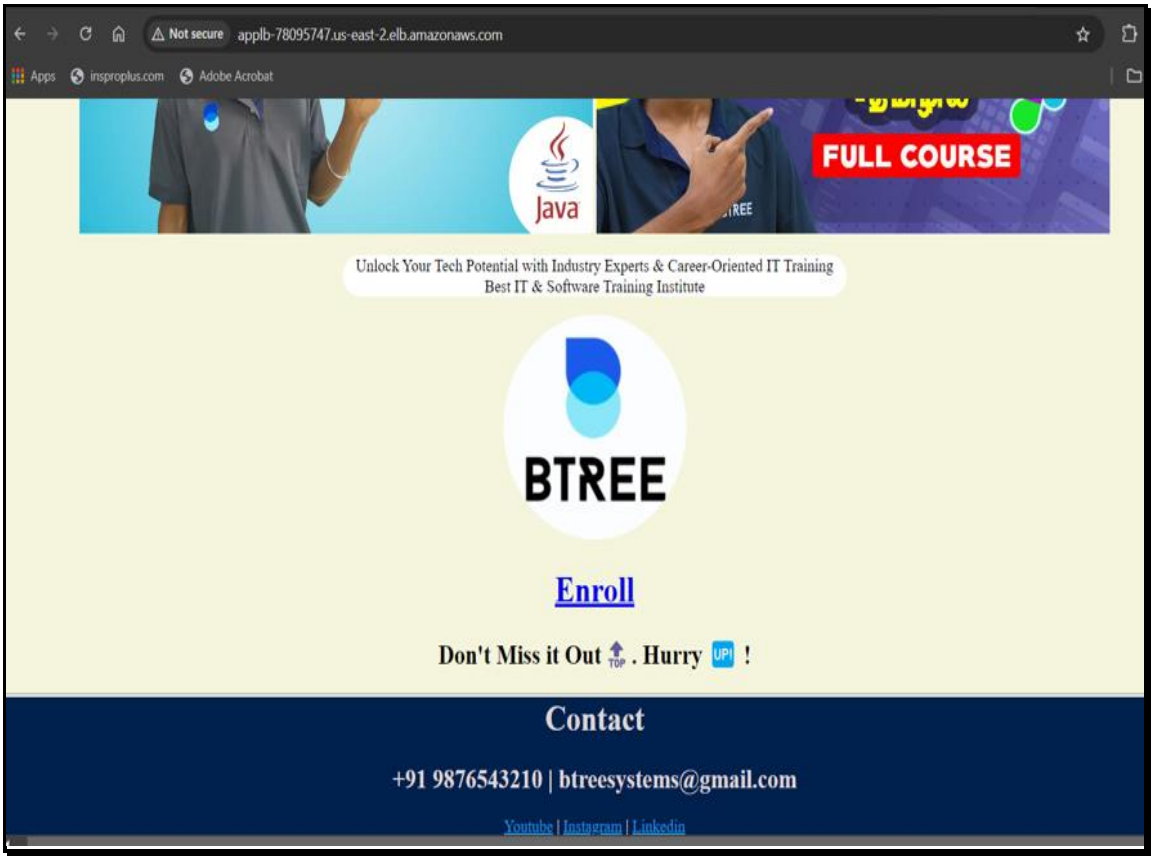
Step 15 : Create **Cloud Front** with App LB.

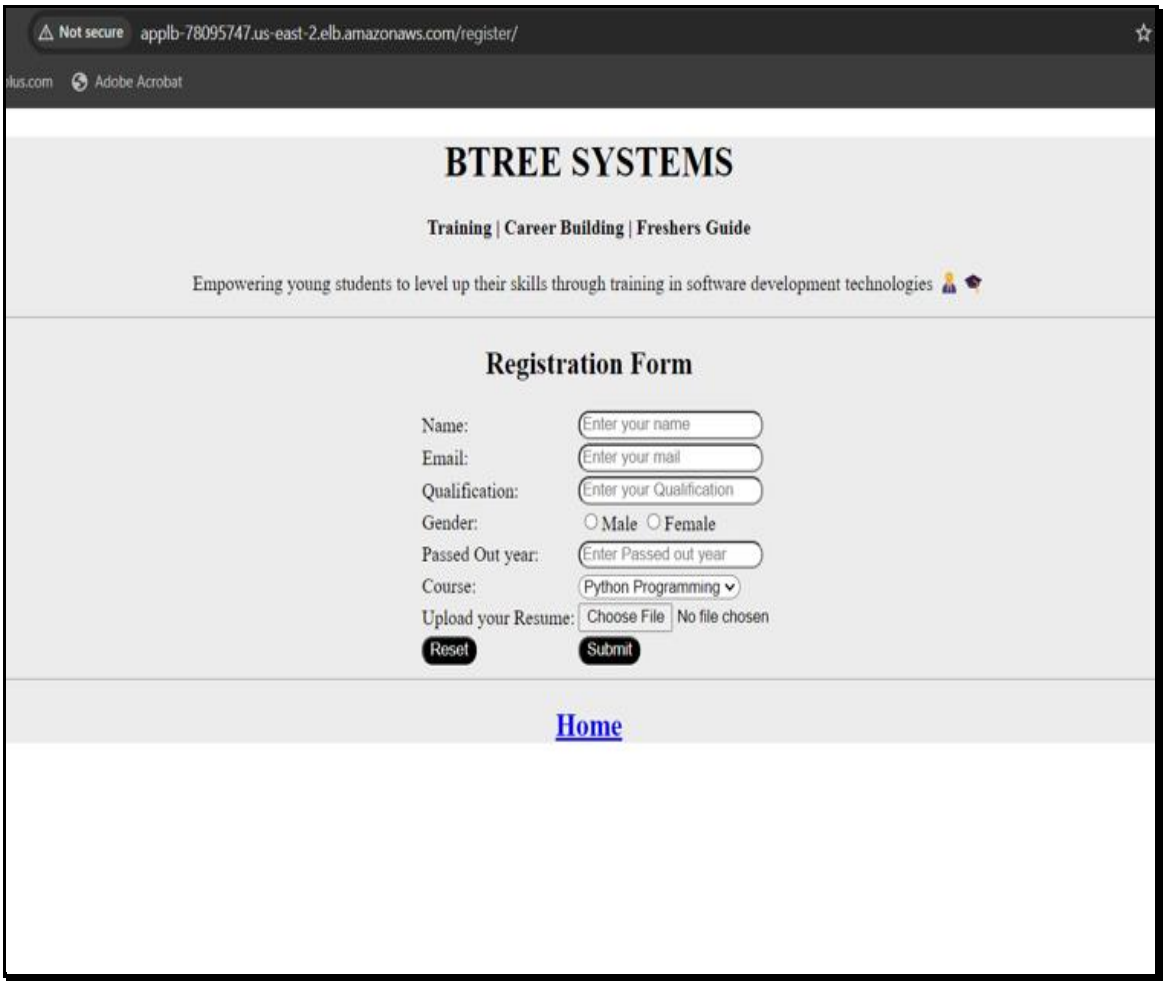


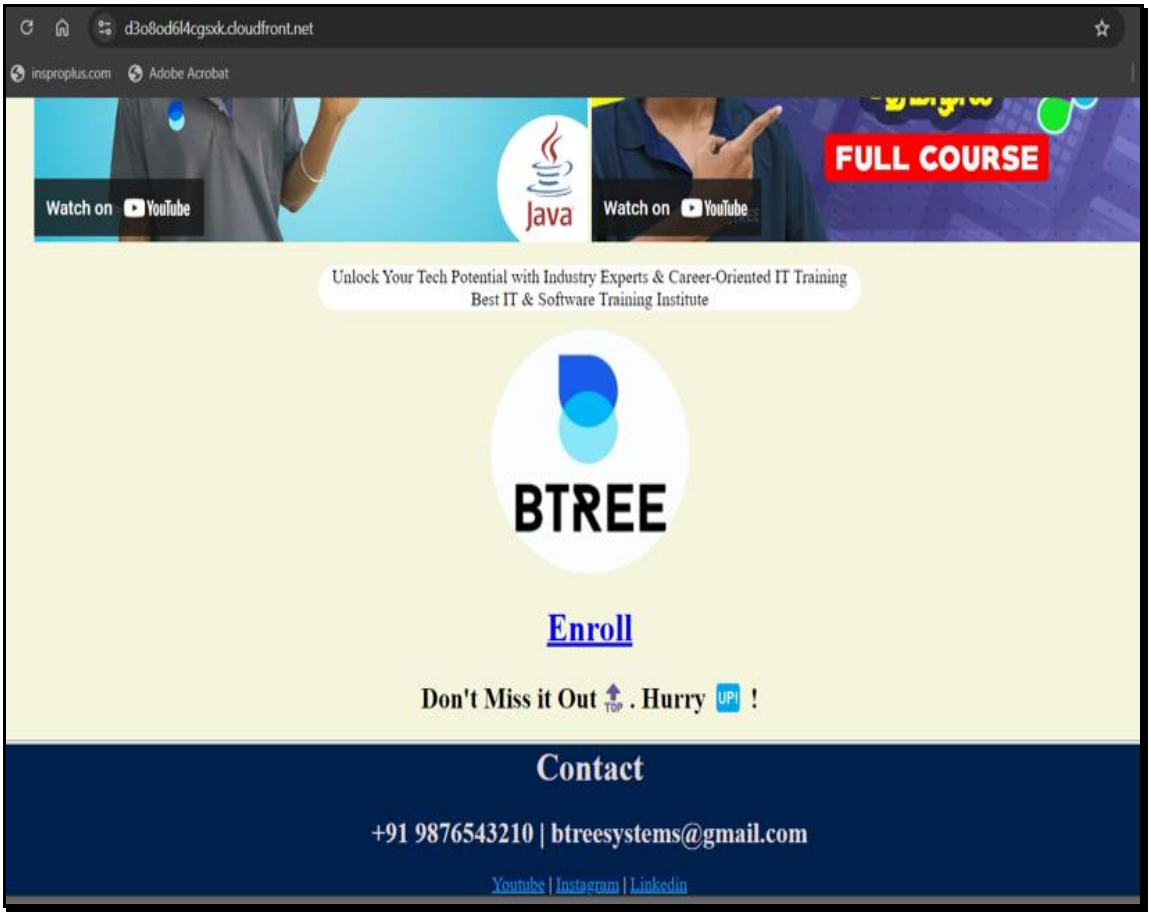
Step 16 : Browse the Website.

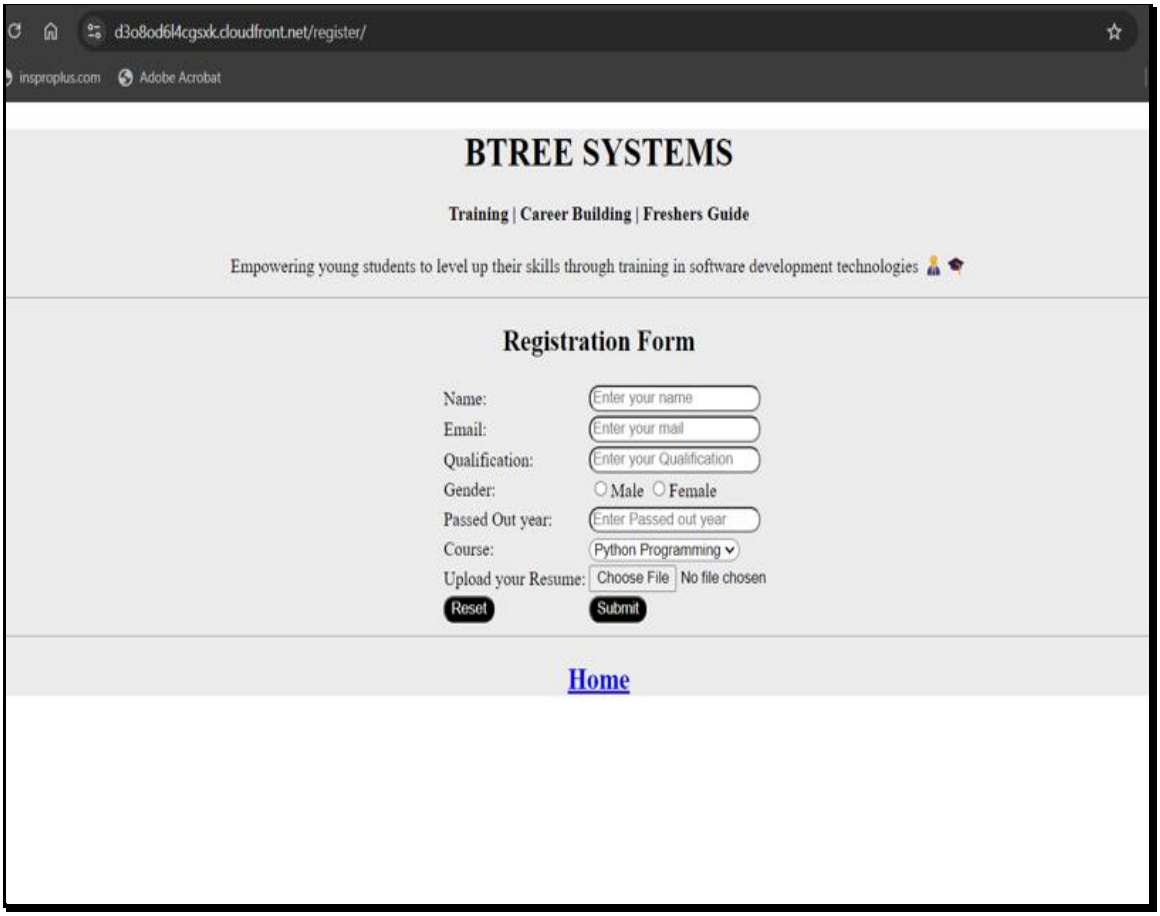





Not secure
applb-78095747.us-east-2.elb.amazonaws.com
Apps
insproplus.com
Adobe Acrobat
Java
FULL COURSE
Unlock Your Tech Potential with Industry Experts & Career-Oriented IT Training
Best IT & Software Training Institute
BTREE
Enroll
Don't Miss it Out . Hurry !
Contact
+91 9876543210 | btreesystems@gmail.com
Youtube | Instagram | Linkedin

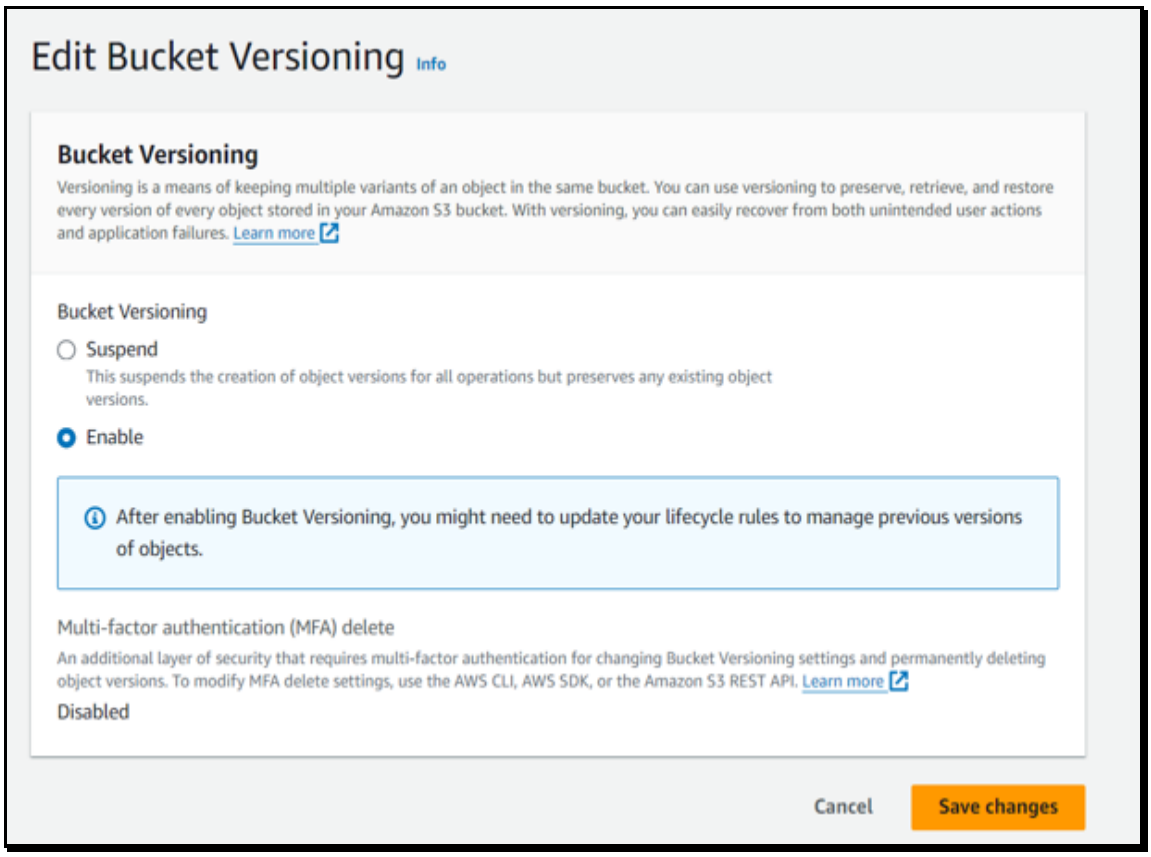

Not secure
applb-78095747.us-east-2.elb.amazonaws.com/register/
Adobe Acrobat
BTREE SYSTEMS
Training | Career Building | Freshers Guide
Empowering young students to level up their skills through training in software development technologies
Registration Form
Name:
Enter your name
Email:
Enter your mail
Qualification:
Enter your Qualification
Gender:
Male
Female
Passed Out year:
Enter Passed out year
Course:
Python Programming
Upload your Resume:
Choose File
No file chosen
Reset
Submit
Home


d3o8od6l4cgsxk.cloudfront.net
insproplus.com
Adobe Acrobat
Watch on YouTube
Java
Watch on YouTube
FULL COURSE
Unlock Your Tech Potential with Industry Experts & Career-Oriented IT Training
Best IT & Software Training Institute
BTREE
Enroll
Don't Miss it Out . Hurry !
Contact
+91 9876543210 | btreesystems@gmail.com
Youtube | Instagram | Linkedin

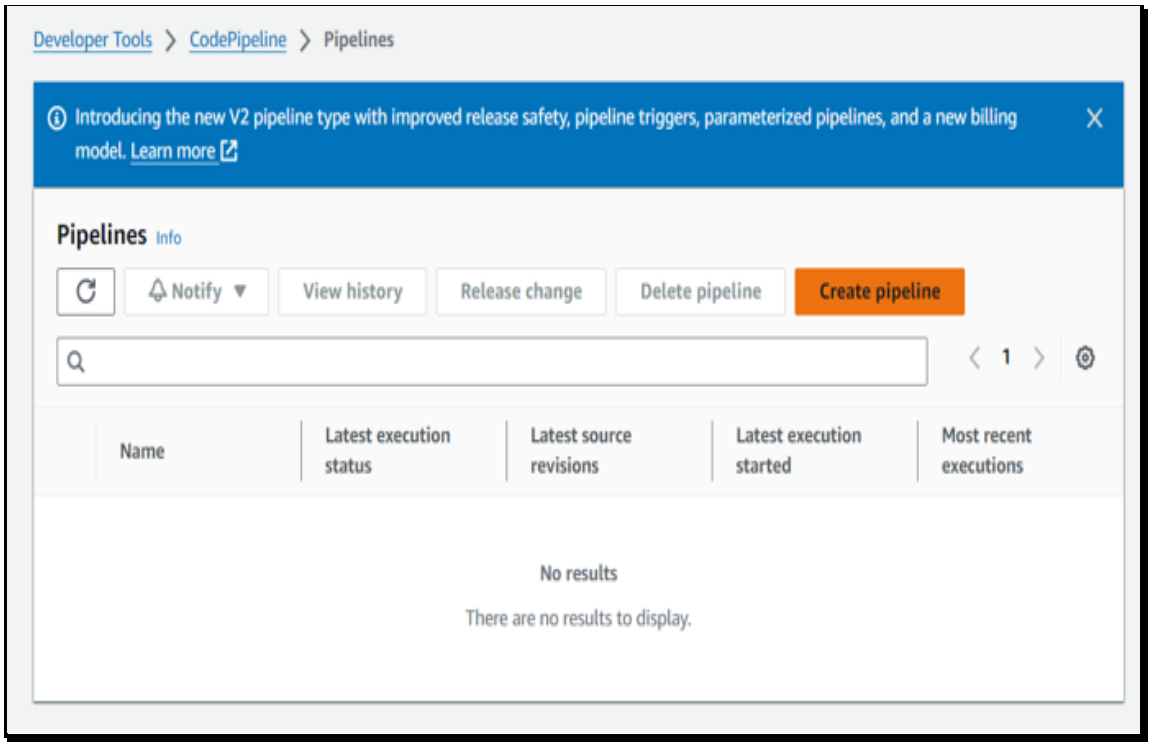

d3o8od6l4cgsxk.cloudfront.net/register/
insproplus.com
Adobe Acrobat
BTREE SYSTEMS
Training | Career Building | Freshers Guide
Empowering young students to level up their skills through training in software development technologies
Registration Form
Name:
Enter your name
Email:
Enter your mail
Qualification:
Enter your Qualification
Gender:
Male
Female
Passed Out year:
Enter Passed out year
Course:
Python Programming
Upload your Resume:
Choose File
No file chosen
Reset
Submit
Home

Step 17 : Create **Code Pipeline** S3 bucket.

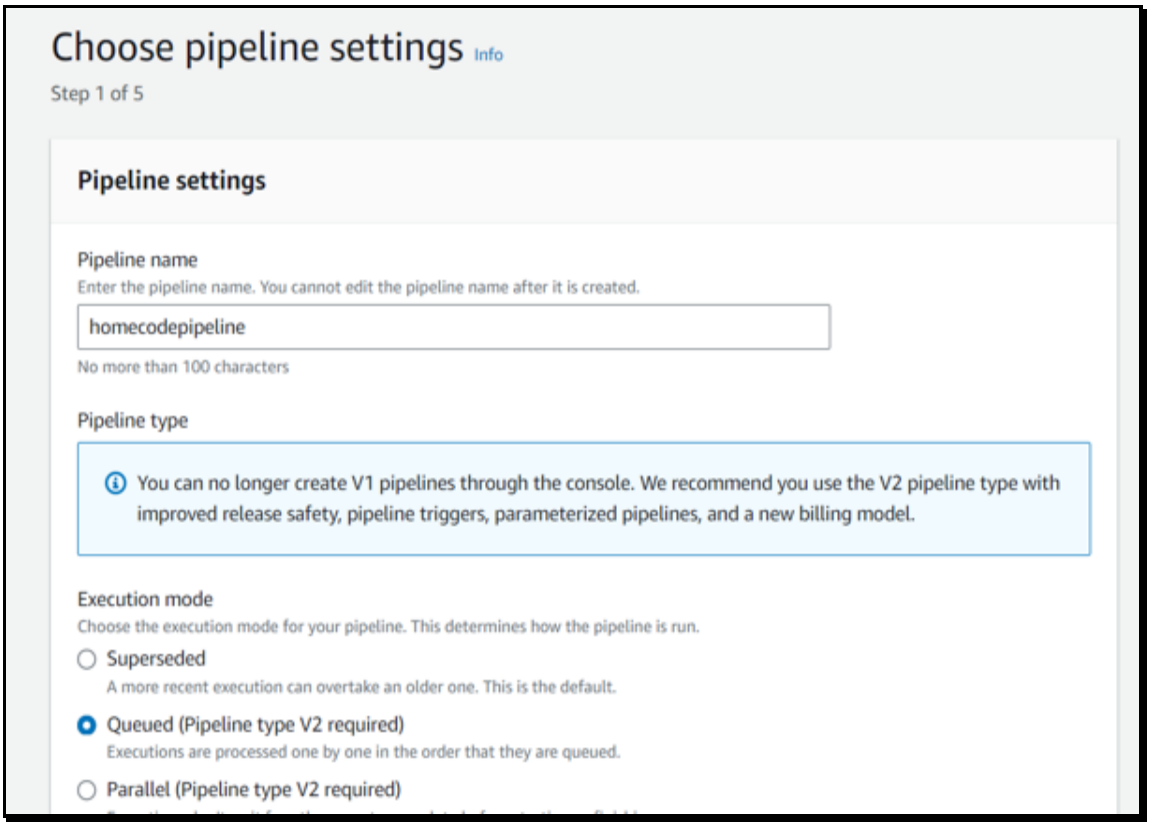
Enable Bucket Versioning

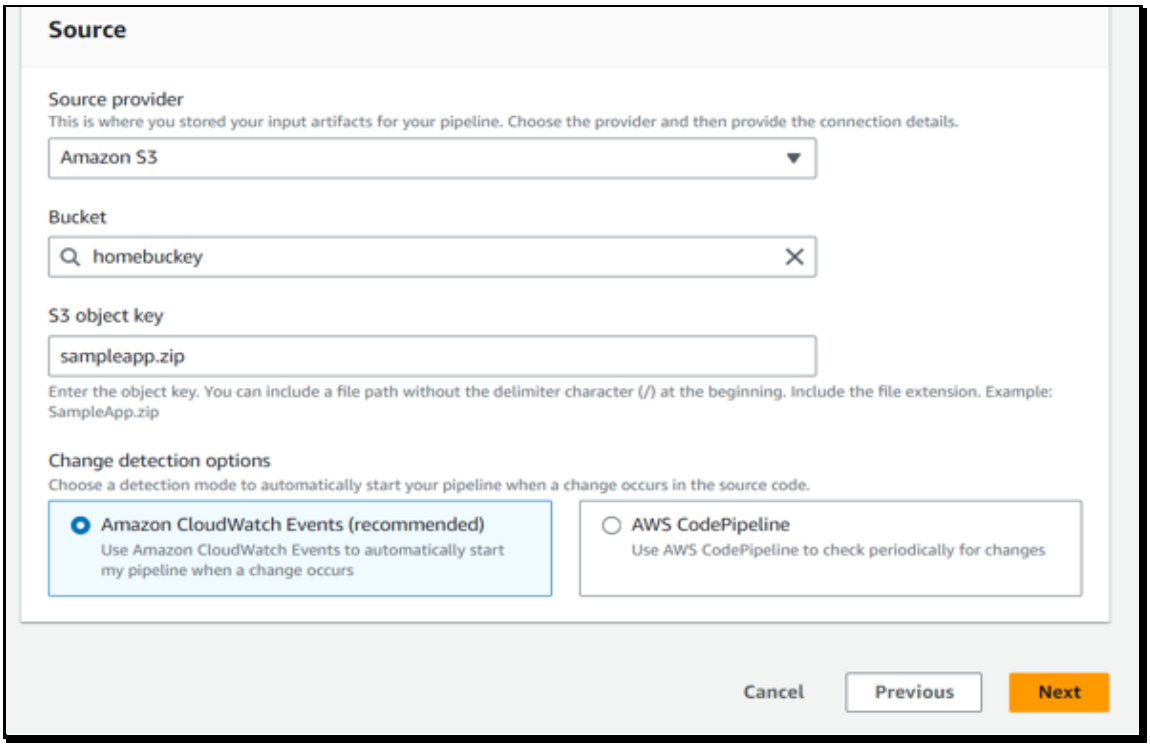


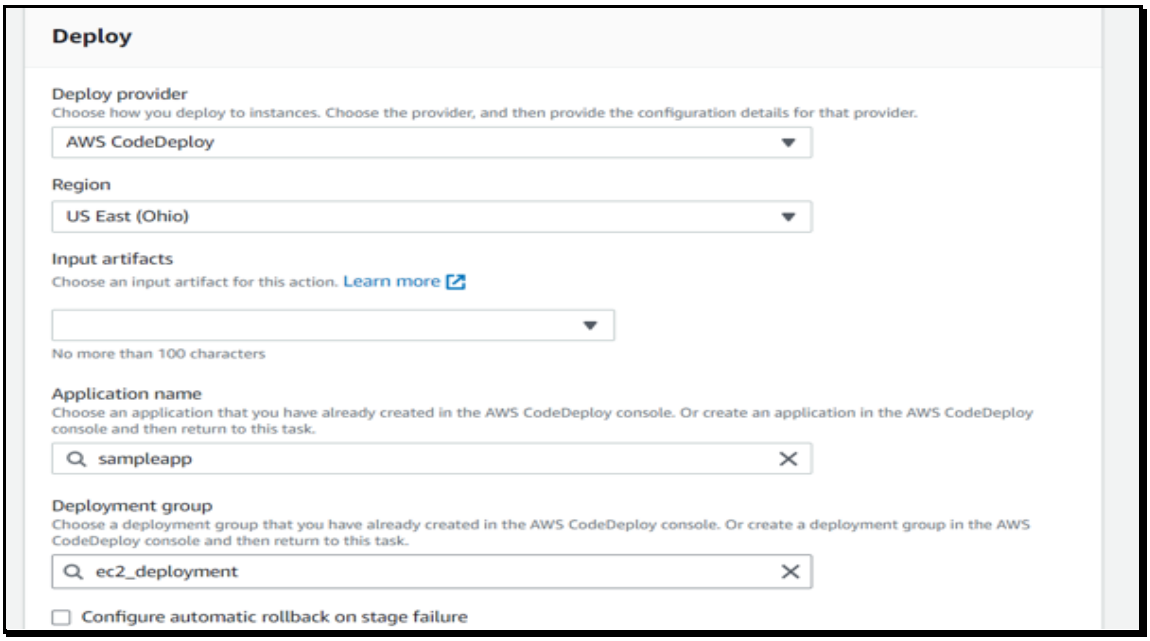
Create Pipeline

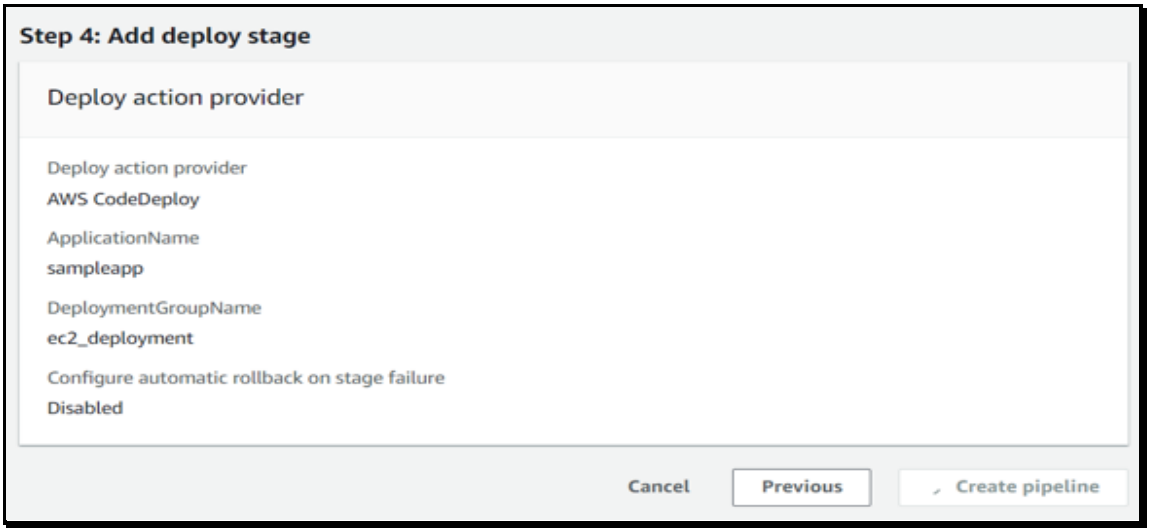


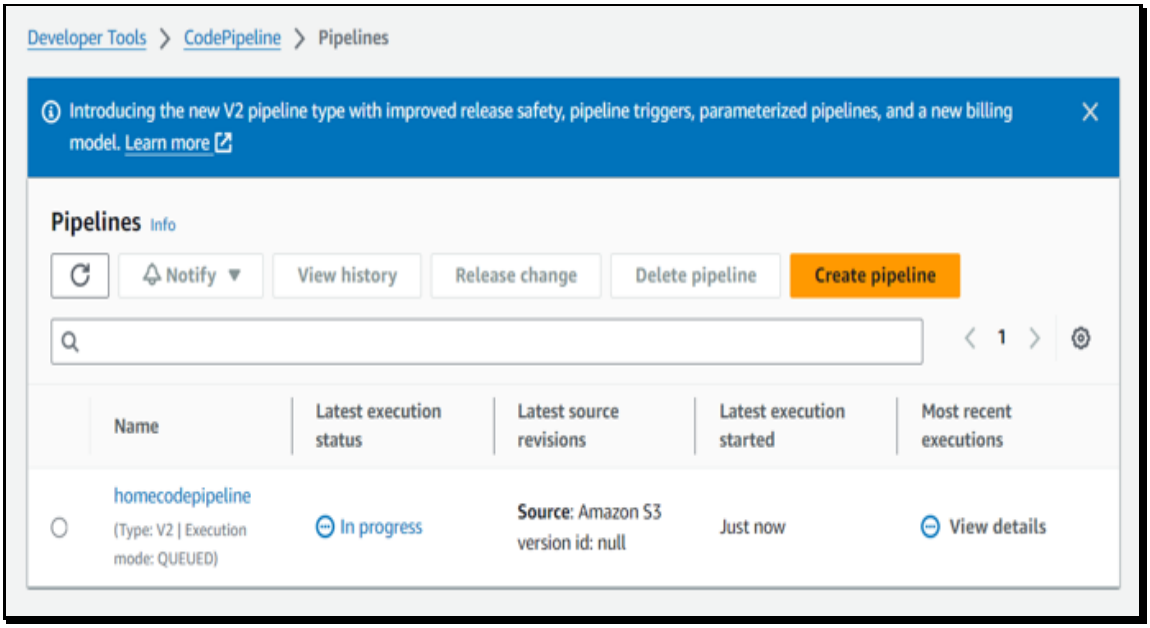
Create 1st Code Pipeline



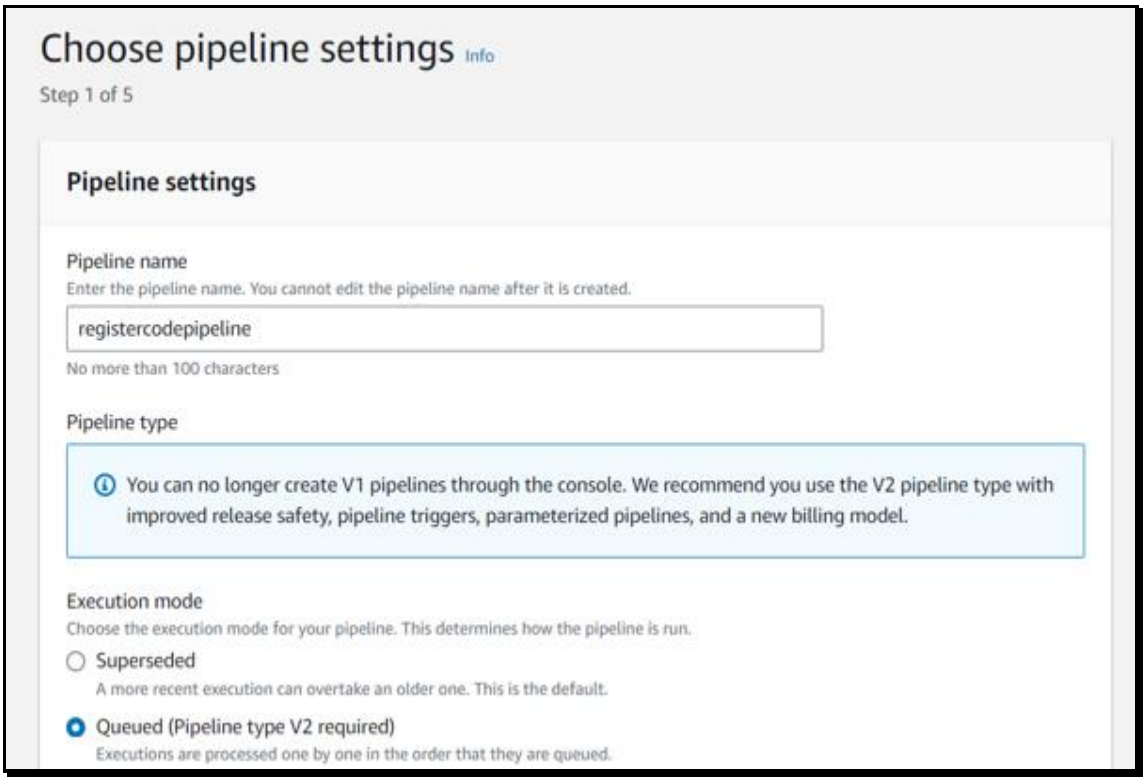


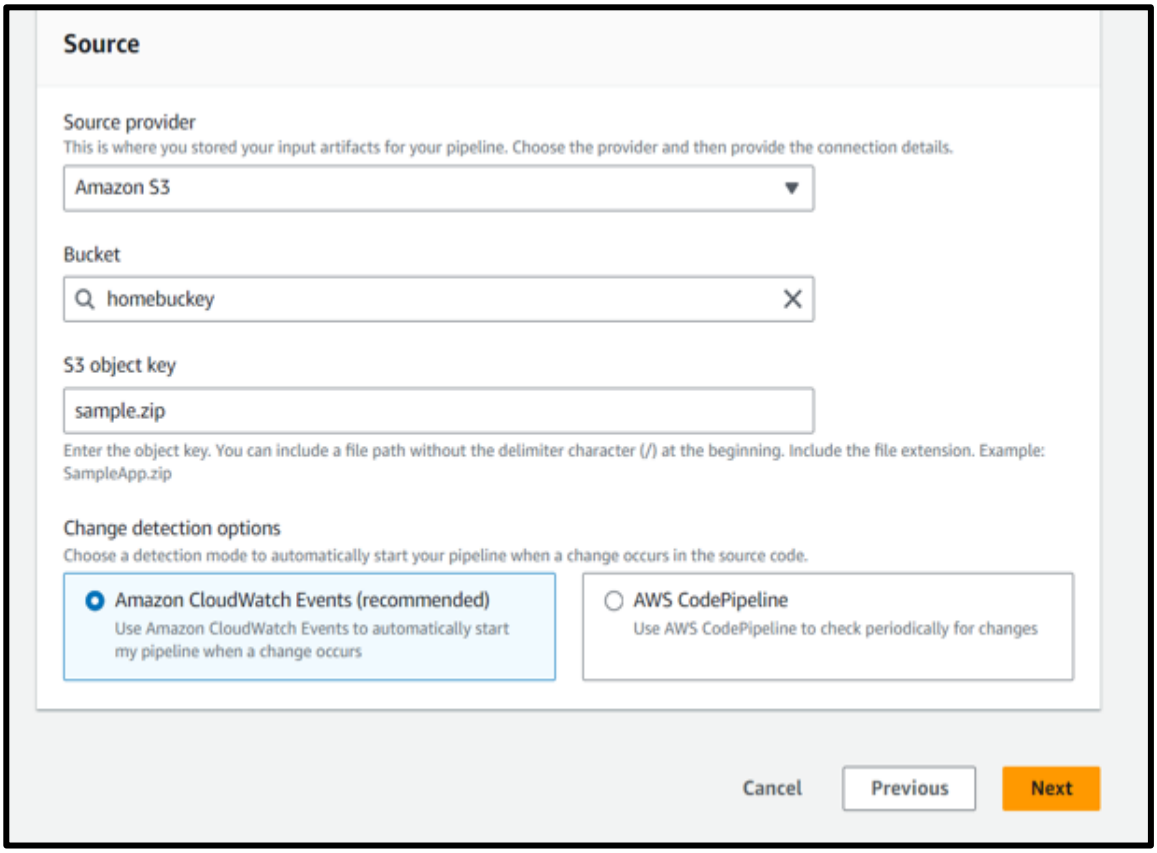

Deploy
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy
Region
US East (Ohio)
Input artifacts
Choose an input artifact for this action. Learn more
No more than 100 characters
Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.
sampleapp
Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.
ec2_deployment
Configure automatic rollback on stage failure

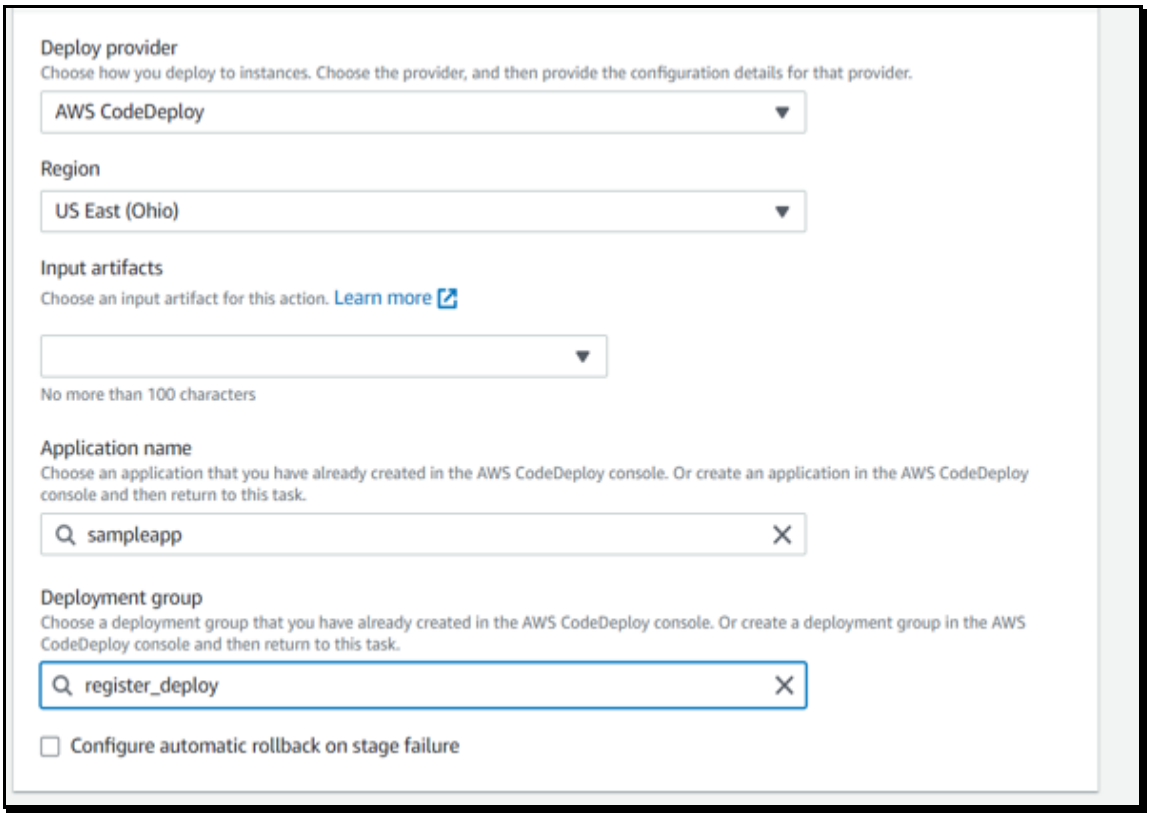

Step 4: Add deploy stage
Deploy action provider
Deploy action provider
AWS CodeDeploy
ApplicationName
sampleapp
DeploymentGroupName
ec2_deployment
Configure automatic rollback on stage failure
Disabled
Cancel
Previous
Create pipeline

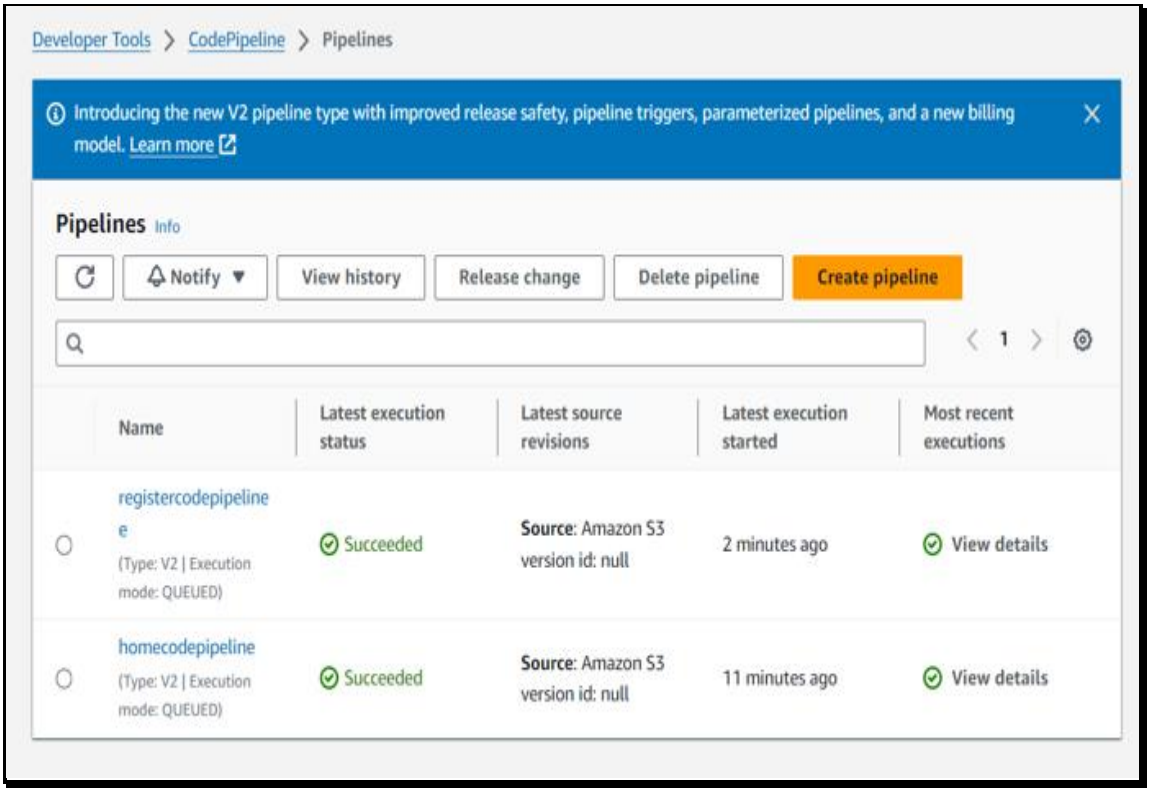

Developer Tools
CodePipeline
Pipelines
Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more
Pipelines Info
Notify
View history
Release change
Delete pipeline
Create pipeline
Name
Latest execution status
Latest source revisions
Latest execution started
Most recent executions
homecodepipeline
(Type: V2 | Execution mode: QUEUED)
In progress
Source: Amazon S3 version id: null
Just now
View details

Create 2nd Code Pipeline






Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS CodeDeploy
Region
US East (Ohio)
Input artifacts
Choose an input artifact for this action. Learn more
No more than 100 characters
Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.
sampleapp
Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.
register_deploy
Configure automatic rollback on stage failure


Developer Tools > CodePipeline > Pipelines
Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more
Pipelines Info
Notify
View history
Release change
Delete pipeline
Create pipeline
1
Name
Latest execution status
Latest source revisions
Latest execution started
Most recent executions
registercodepipeline
(Type: V2 | Execution mode: QUEUED)
Succeeded
Source: Amazon S3 version id: null
2 minutes ago
View details
homecodepipeline
(Type: V2 | Execution mode: QUEUED)
Succeeded
Source: Amazon S3 version id: null
11 minutes ago
View details

Step 18 : Re-edit the website in Developer Machine.

```
[root@ip-172-31-1-16 sample]# cd ..
[root@ip-172-31-1-16 deploy]# cd ..
[root@ip-172-31-1-16 scripts]# cd ..
[root@ip-172-31-1-16 sampleapp]# vi index.html
[root@ip-172-31-1-16 sampleapp]# zip -r ../sampleapp.zip .
  adding: scripts/ (stored 0%)
  adding: scripts/httpd_install.sh (stored 0%)
  adding: scripts/httpd_start.sh (deflated 21%)
  adding: scripts/httpd_stop.sh (deflated 21%)
  adding: scripts/deploy/ (stored 0%)
  adding: scripts/deploy/sample/ (stored 0%)
  adding: scripts/deploy/sample/index.html (deflated 68%)
  adding: scripts/deploy/sample/scripts/ (stored 0%)
  adding: scripts/deploy/sample/scripts/httpd_install.sh (stored 0%)
  adding: scripts/deploy/sample/scripts/httpd_start.sh (deflated 21%)
  adding: scripts/deploy/sample/scripts/httpd-stop.sh (deflated 21%)
  adding: scripts/deploy/sample/appspec.yml (deflated 52%)
  adding: appspec.yml (deflated 52%)
  adding: index.html (deflated 60%)
[root@ip-172-31-1-16 sampleapp]# cd ..
[root@ip-172-31-1-16 deploy_dir]# ls
sampleapp  sampleapp.zip
```

root@ip-172-31-1-16:~/deploy_dir/sampleapp

```
            <img class="logo "src="https://media.licdn.com/dms/image/v2/D560BAQHq7aqaQNG5pg/company-logo_200_200/compa
ny-logo_200_200/0/1705737075022/btreesystems_logo?e=2147483647&v=beta&t=vRQ2FPg4yQuCx29rsnlUptNjumGSsjkOXWbSWq5hXj4" w
idth="200" height="170"> <br>
            <h1>

            <a  href="register.html"> Enroll</a>
            </h1>
            <h2 class="hurry">Don't Miss it Out . Hurry  !</h2>

        </center>


        <hr>


<div class="footer">
    <center>
    <h1>Contact</h1>
    <h2>+91 0123456789 | btreesystems@gmail.com</h2>
    <p >
        <a class="links" href="https://www.youtube.com/@btreesystems">Youtube</a> |
        <a class="links" href="https://www.instagram.com/btreesystems/?hl=en">Instagram</a> |
        <a  class="links" href="https://www.linkedin.com/company/btreesystems/?originalSubdomain=in">Linkedin</a>
    </p>

    </center>


</div>



    </body>
</html>
-- INSERT --                                                                147,23        Bot
```

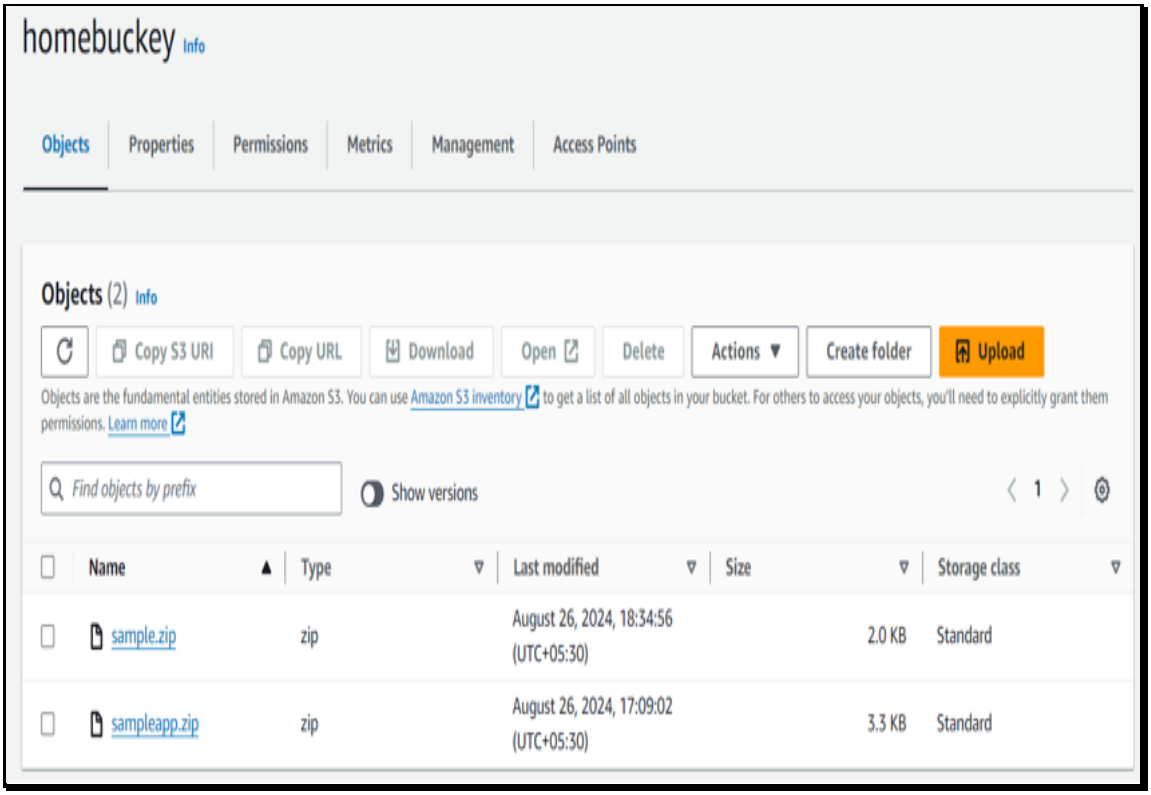
```
[root@ip-172-31-1-16 deploy_dir]# aws s3 cp sampleapp.zip s3://homebuckey
upload: ./sampleapp.zip to s3://homebuckey/sampleapp.zip
[root@ip-172-31-1-16 deploy_dir]# cd ./sample/
-bash: cd: ./sample/: No such file or directory
[root@ip-172-31-1-16 deploy_dir]# cd ./sample
-bash: cd: ./sample: No such file or directory
[root@ip-172-31-1-16 deploy_dir]# cd sampleapp
[root@ip-172-31-1-16 sampleapp]# pwd
/root/deploy_dir/sampleapp
[root@ip-172-31-1-16 sampleapp]# ls
appspec.yml  index.html  scripts
[root@ip-172-31-1-16 sampleapp]# cd /scripts/deploy/sample
-bash: cd: /scripts/deploy/sample: No such file or directory
[root@ip-172-31-1-16 sampleapp]# cd /scripts
-bash: cd: /scripts: No such file or directory
[root@ip-172-31-1-16 sampleapp]# cd scripts
[root@ip-172-31-1-16 scripts]# cd deploy/sample
[root@ip-172-31-1-16 sample]# vi index.html
```

root@ip-172-31-1-16:~/deploy_dir/sampleapp/scripts/deploy/sample

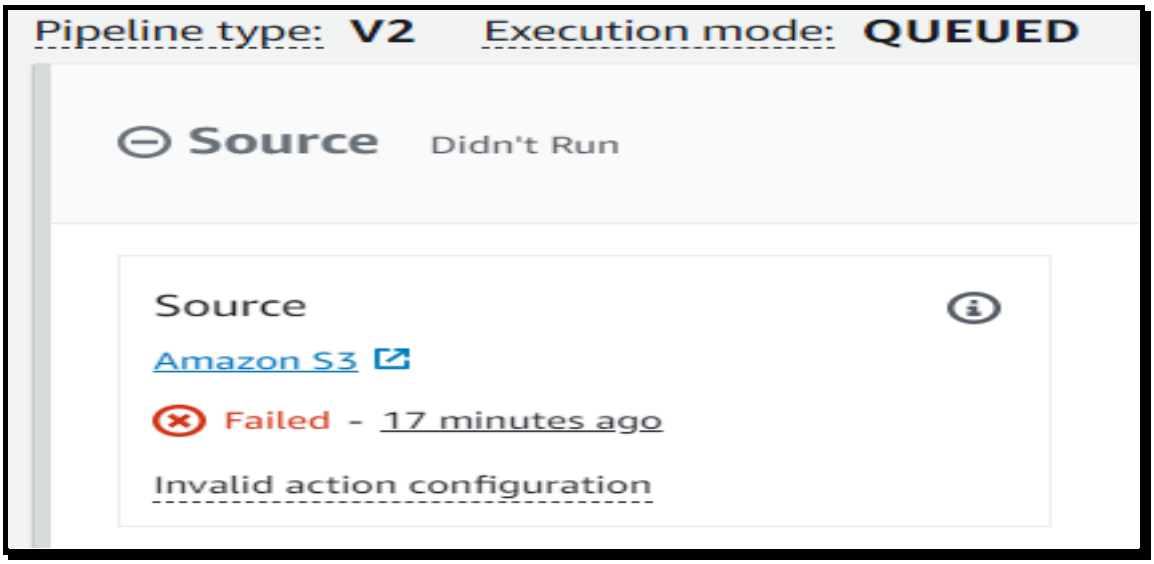
```
}

.file{

    border-inline-color: black;
}

.tabs{
    margin-left: 90px;
}
.naming{
    border-radius: 10px;
    background-color: black;
    color: white;
}
</style>
</head>
<body>
    <div class="register">
    <center>
        <h1>BTREE SYSTEMS FORM</h1>
        <h4>Training | Career Building | Freshers Guide</h4>
        <p>Empowering young students to level up their skills through training in software development technologies
<200d> </p>
        <hr>
    </center>
    <center>
        <h2>Registration Form</h2>

        <form class =tabs>
            <table class="table">

                    <tr>
                        <td >Name:</td>
-- INSERT --                                                                    35,31         18%
```

Step 19 : Copy website file to S3 bucket.

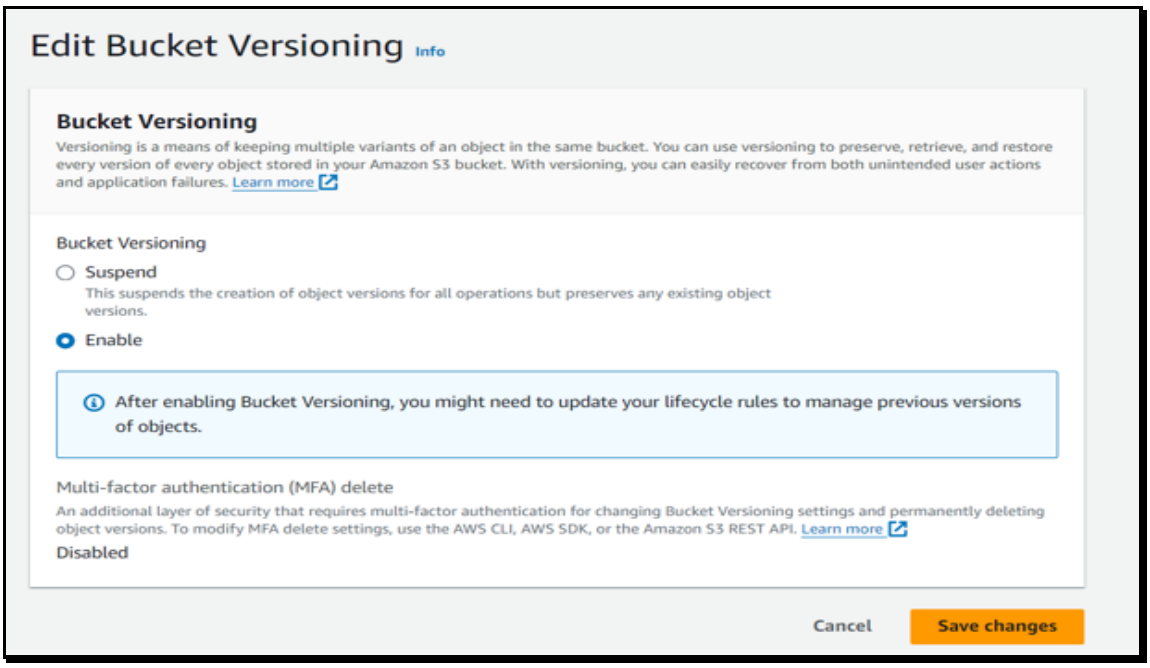
```
[root@ip-172-31-1-16 sampleapp]# zip -r ../sampleapp.zip .
  adding: scripts/ (stored 0%)
  adding: scripts/httpd_install.sh (stored 0%)
  adding: scripts/httpd_start.sh (deflated 21%)
  adding: scripts/httpd_stop.sh (deflated 21%)
  adding: scripts/deploy/ (stored 0%)
  adding: scripts/deploy/sample/ (stored 0%)
  adding: scripts/deploy/sample/index.html (deflated 68%)
  adding: scripts/deploy/sample/scripts/ (stored 0%)
  adding: scripts/deploy/sample/scripts/httpd_install.sh (stored 0%)
  adding: scripts/deploy/sample/scripts/httpd_start.sh (deflated 21%)
  adding: scripts/deploy/sample/scripts/httpd-stop.sh (deflated 21%)
  adding: scripts/deploy/sample/appspec.yml (deflated 52%)
  adding: appspec.yml (deflated 52%)
  adding: index.html (deflated 60%)
[root@ip-172-31-1-16 sampleapp]# cd ..
[root@ip-172-31-1-16 deploy_dir]# ls
sampleapp  sampleapp.zip
[root@ip-172-31-1-16 deploy_dir]# aws s3 cp sampleapp.zip s3://homebuckey
upload: ./sampleapp.zip to s3://homebuckey/sampleapp.zip
[root@ip-172-31-1-16 deploy dir]#
```



CHALLENGES FACED :



TOUBLESHOOT :

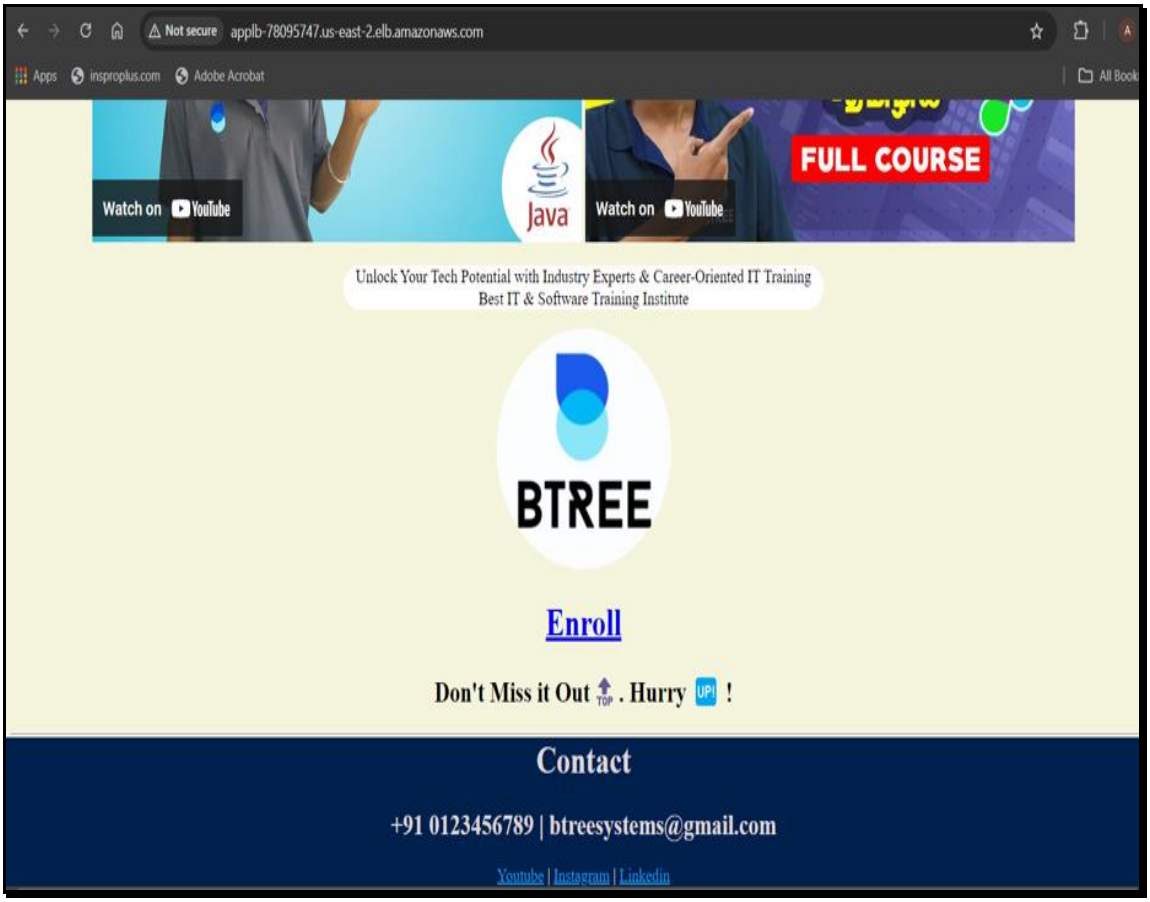


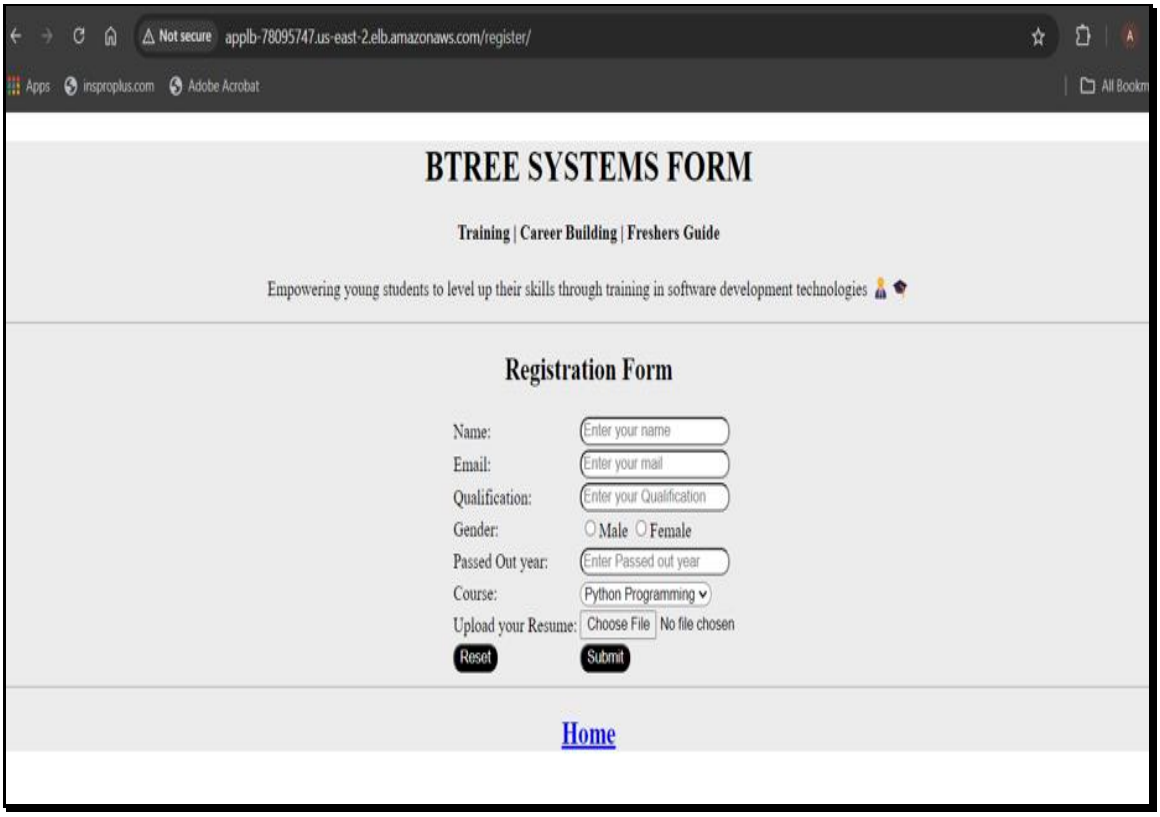
LESSON LEARNED :

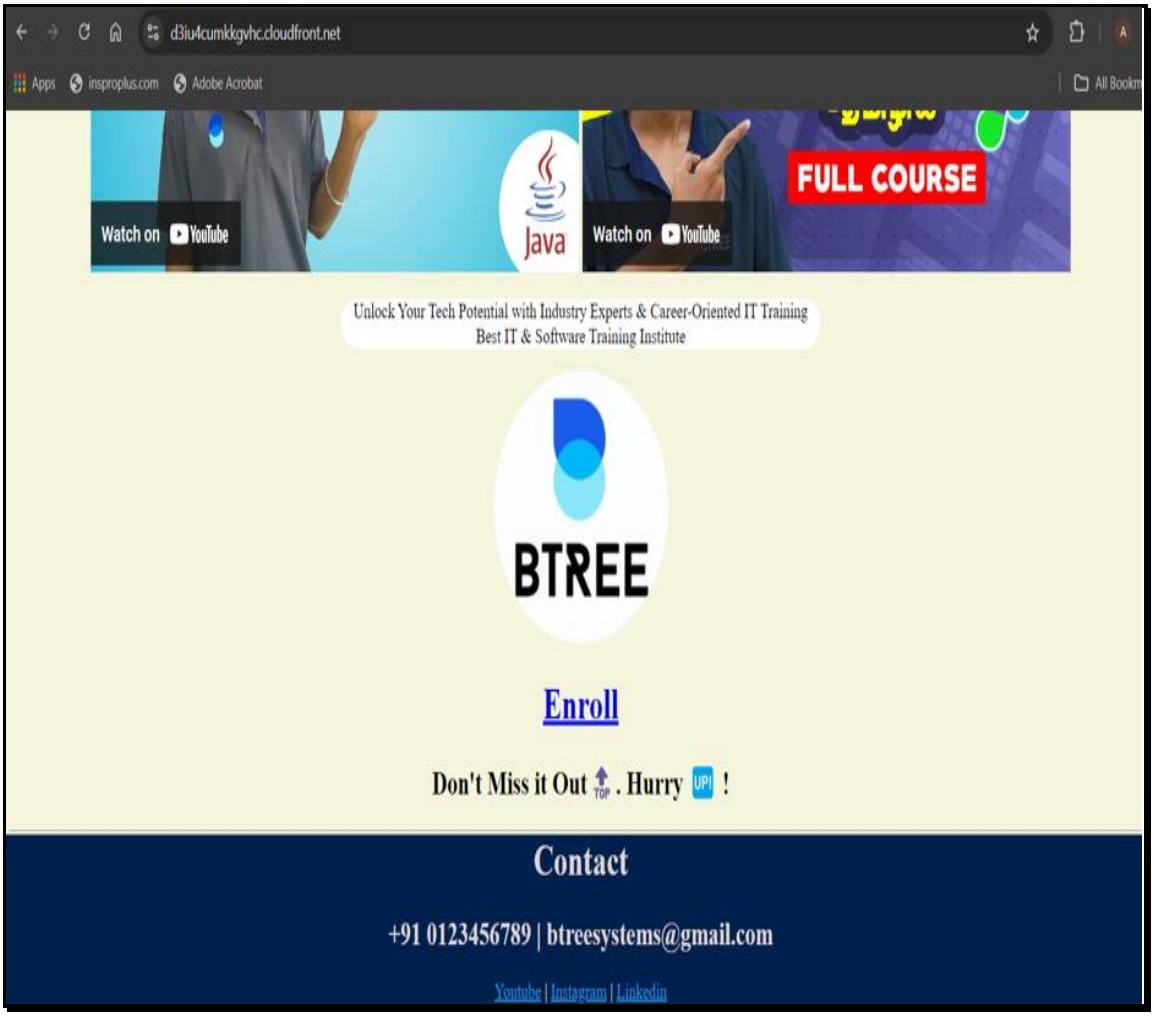
- Hands – on experience with all Amazon web services.
- Good understanding of networking concepts including VPN, VPC, Route table, subnet, and internet gateways.
- Acquried proficiency in how website works from Developer to End user.

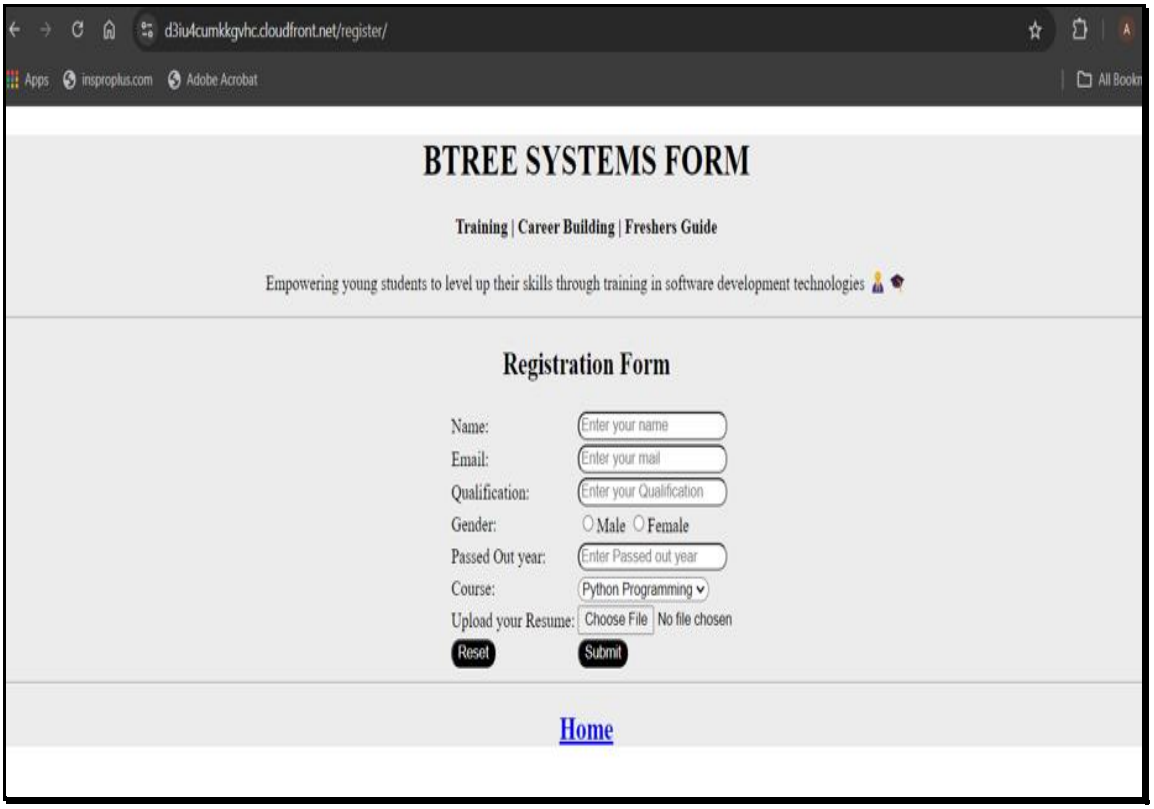
OUTPUT :

Step 20 : Refresh the website changes will show.






d3iu4cumkkgvhc.cloudfront.net
Apps
insproplus.com
Adobe Acrobat
Watch on YouTube
Java
FULL COURSE
Unlock Your Tech Potential with Industry Experts & Career-Oriented IT Training
Best IT & Software Training Institute
BTREE
Enroll
Don't Miss it Out . Hurry !
Contact
+91 0123456789 | btreesystems@gmail.com
Youtube | Instagram | Linkedin


d3iu4cumkkgvhc.cloudfront.net/register/
Apps
insproplus.com
Adobe Acrobat
BTREE SYSTEMS FORM
Training | Career Building | Freshers Guide
Empowering young students to level up their skills through training in software development technologies
Registration Form
Name:
Enter your name
Email:
Enter your mail
Qualification:
Enter your Qualification
Gender:
Male
Female
Passed Out year:
Enter Passed out year
Course:
Python Programming
Upload your Resume:
Choose File
No file chosen
Reset
Submit
Home