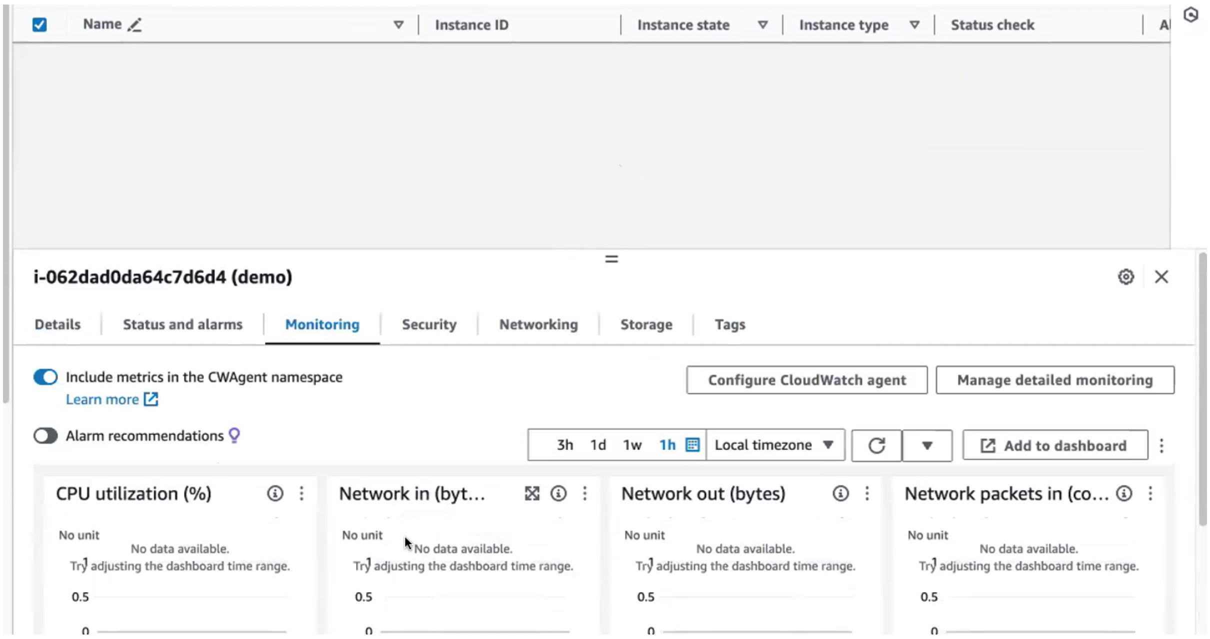
# AWS elasticbeanstalk and cloudwatch 5/7/24(Friday) addi sess

## Cloudwatch:

- **To monitor the ec2 memory ,we create custom matrix.**
- **Before that create ec2 instance**
- **This is default matrix**



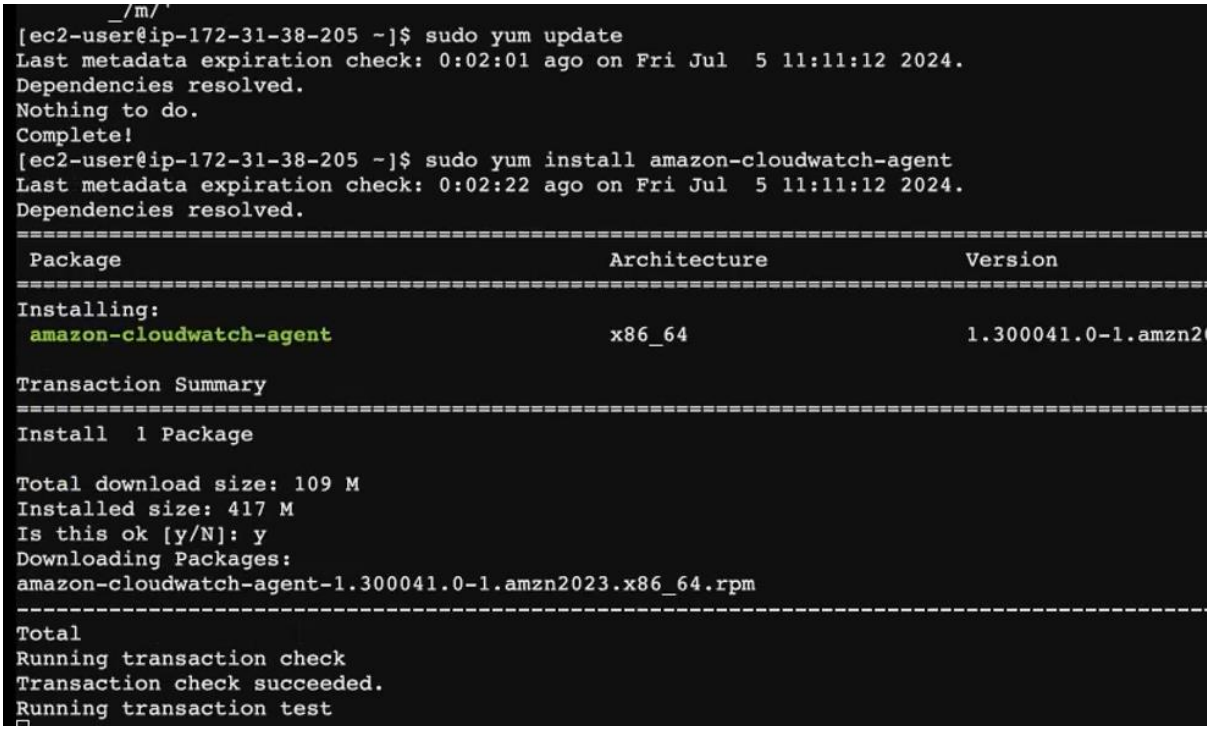
i-062dad0da64c7d6d4 (demo)

No data available.
Try adjusting the dashboard time range.

## Creating custom matrix:

**before creating your custom metrics, we want our cloud watch to monitor this Ec. 2 using the custom metric. For that we have to install a tool. We will be installing cloud watch agent. We call that as Cloud Watch agent. So with the help of Cloud Watch agent, we can create custom metrics to monitor your Ec2, memory.**

**Connect with ec2.give the command to install amazon Cloud Watch agent**

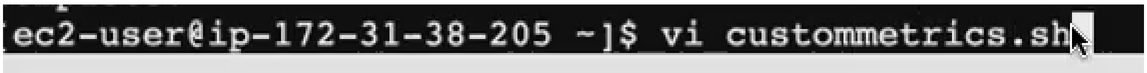


```
       _/m/'
[ec2-user@ip-172-31-38-205 ~]$ sudo yum update
Last metadata expiration check: 0:02:01 ago on Fri Jul  5 11:11:12 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-38-205 ~]$ sudo yum install amazon-cloudwatch-agent
Last metadata expiration check: 0:02:22 ago on Fri Jul  5 11:11:12 2024.
Dependencies resolved.
==========================================================================================
 Package                                   Architecture               Version
==========================================================================================
Installing:
 amazon-cloudwatch-agent                   x86_64                     1.300041.0-1.amzn2

Transaction Summary
==========================================================================================
Install  1 Package

Total download size: 109 M
Installed size: 417 M
Is this ok [y/N]: y
Downloading Packages:
amazon-cloudwatch-agent-1.300041.0-1.amzn2023.x86_64.rpm
------------------------------------------------------------------------------------------
Total
Running transaction check
Transaction check succeeded.
Running transaction test
```

**Link for install a.c.w other than linux**

[https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-commandline-fleet.html](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-commandline-fleet.html)

After installing the cloud watch, we have to create the metrics. Okay, so you have to create the metrics and that metrics will be created in your Cloud watch agent dashboard. So what I do, I will create a **metric. We we can write. You can create this using python /shell scripting or any scripting language you can use.**

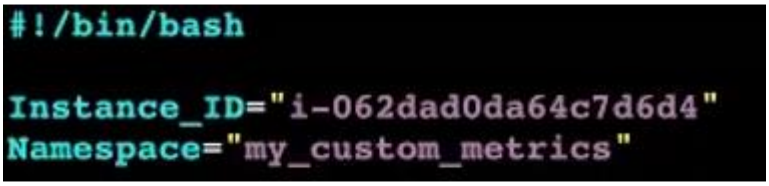


```
[ec2-user@ip-172-31-38-205 ~]$ vi custommetrics.sh
```

#bin/bash

Instance-id= copy and paste

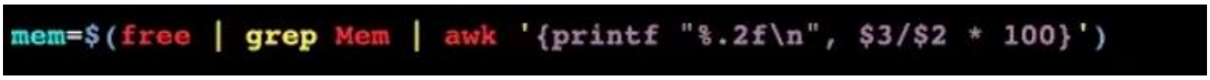
Namespace=anything we give



```
#!/bin/bash

Instance_ID="i-062dad0da64c7d6d4"
Namespace="my_custom_metrics"
```

what this metric should do, it has to. We have to. We have to monitor the memory utilization. So I will create a variable called mem.So this variable what we are doing using the free command, this free command will display you the system memory information.

I'm only filtering the line containing my memory information that does my RAM information. Okay? So next, I'm further filtering out using my awk command. So this this command what it does, it will calculate the percentage of used memory. Okay, it will calculate the percentage of used memory, and I am formatting it to 2 decimal places 0.2f/n



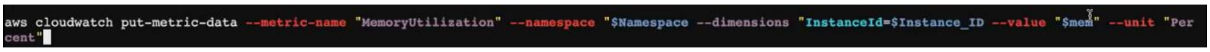
```
mem=$(free | grep Mem | awk '{printf "%.2f\n", $3/$2 * 100}')
```

when this script is executed, my metric will be created. Okay. So when this script is executed, this memory variable will be created, and it will be holding this value. But what I want to, I want to create a metric.

I want to create a metric in my cloud watch dashboard. So it's like creating a resource. You're creating a resource in the cloud watch. We are creating a custom metric for that I use aws command. I use aws cli. You have aws, cloud watch, so aws cloud, watch, command will help you to create a metric in your cloud watch dashboard.

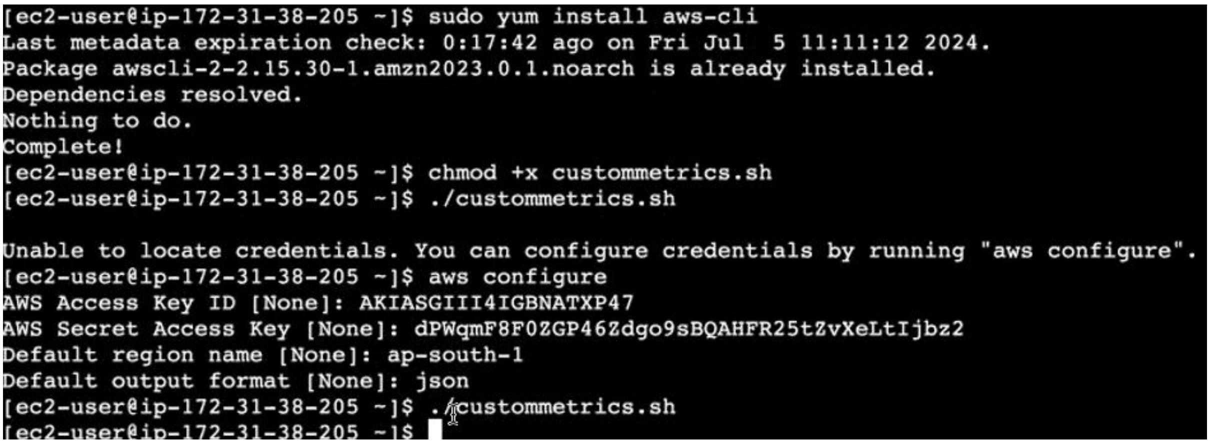
going to create a metric in the cloud watch dashboard. So how do I create a metric using this keyword metric data I'm going to publish, or I'm going to create a custom metric. give any name for my metric

Aw cli command to create your metric. So while creating a metric, you give the val name of your metric, and in which namespace your metric should be created. and what is the instance that it has to monitor the resource, what it has to monitor? We put it under dimensions and value what it has to what value it has to return. Okay, so what it has to monitor, or what this metric has to perform, it has to perform the memory utilization in my Ec. 2 mission.



```
aws cloudwatch put-metric-data --metric-name "MemoryUtilization" --namespace "$Namespace --dimensions "InstanceId=$Instance_ID --value "$mem" --unit "Per
cent"
```

So before executing this script in this script, what I have done, I have used. Aws cli. So 1st we will install  Aws Cli.

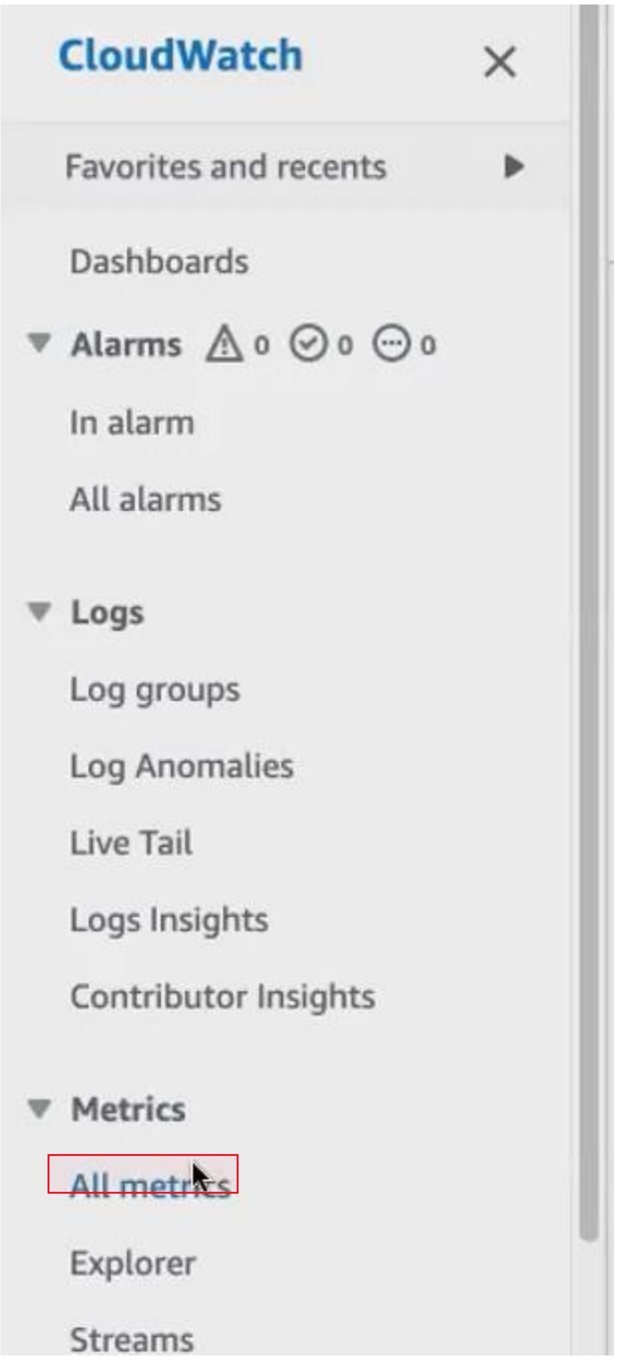


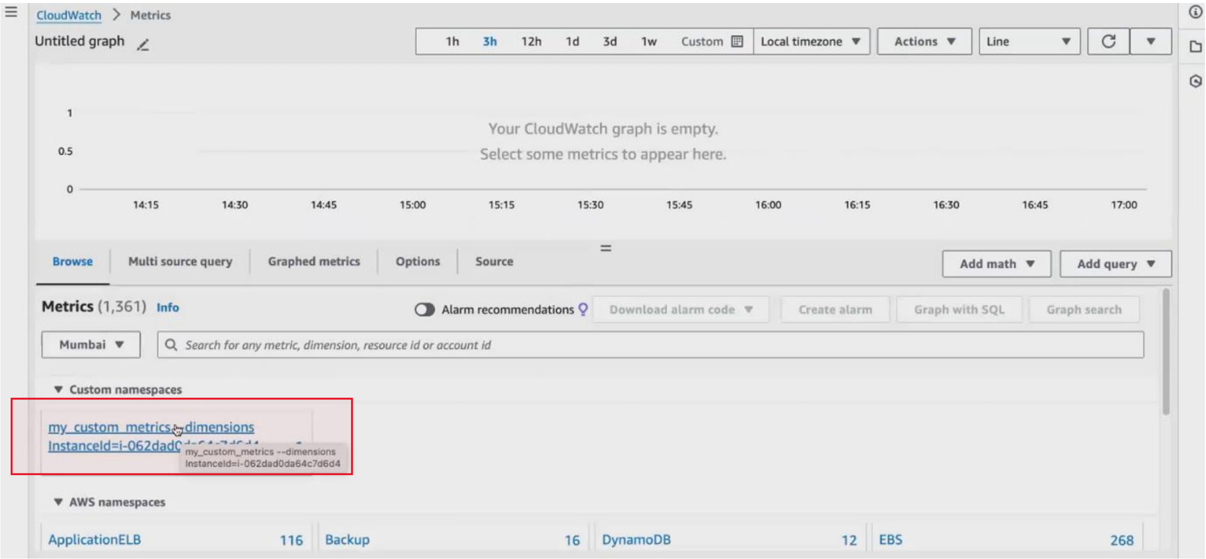
```
[ec2-user@ip-172-31-38-205 ~]$ sudo yum install aws-cli
Last metadata expiration check: 0:17:42 ago on Fri Jul  5 11:11:12 2024.
Package awscli-2-2.15.30-1.amzn2023.0.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-38-205 ~]$ chmod +x custommetrics.sh
[ec2-user@ip-172-31-38-205 ~]$ ./custommetrics.sh

Unable to locate credentials. You can configure credentials by running "aws configure".
[ec2-user@ip-172-31-38-205 ~]$ aws configure
AWS Access Key ID [None]: AKIASGIII4IGBNATXP47
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
[ec2-user@ip-172-31-38-205 ~]$ ./custommetrics.sh
[ec2-user@ip-172-31-38-205 ~]$
```

Access key and matrix access key was generated only by IAM users. Then execute the script.

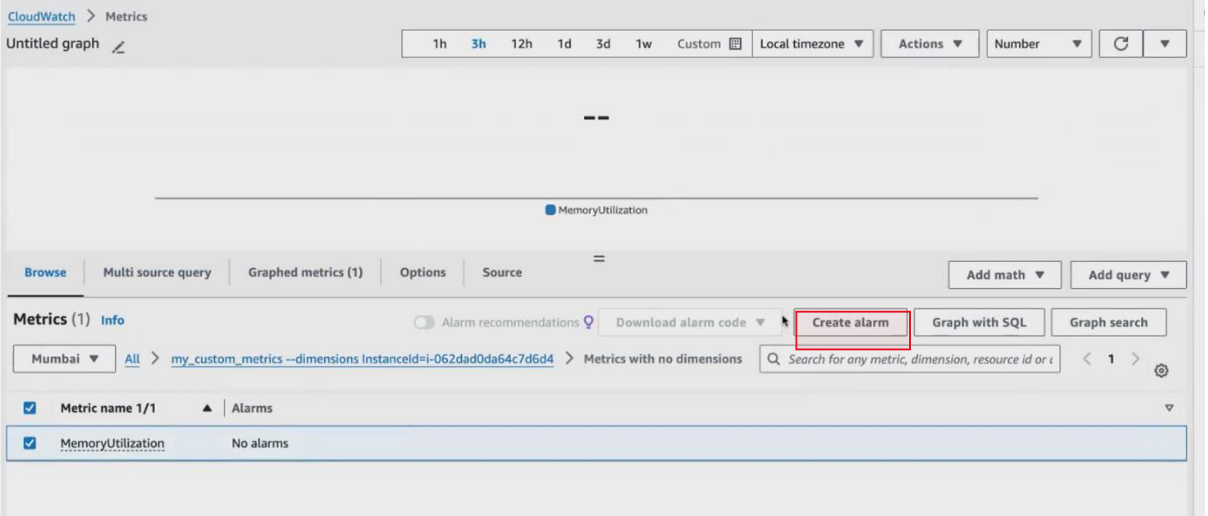
<u>How to check custom matrix created or not</u>

Click cloud watch


CloudWatch
Favorites and recents
Dashboards
Alarms 0 0 0
In alarm
All alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
All metrics
Explorer
Streams

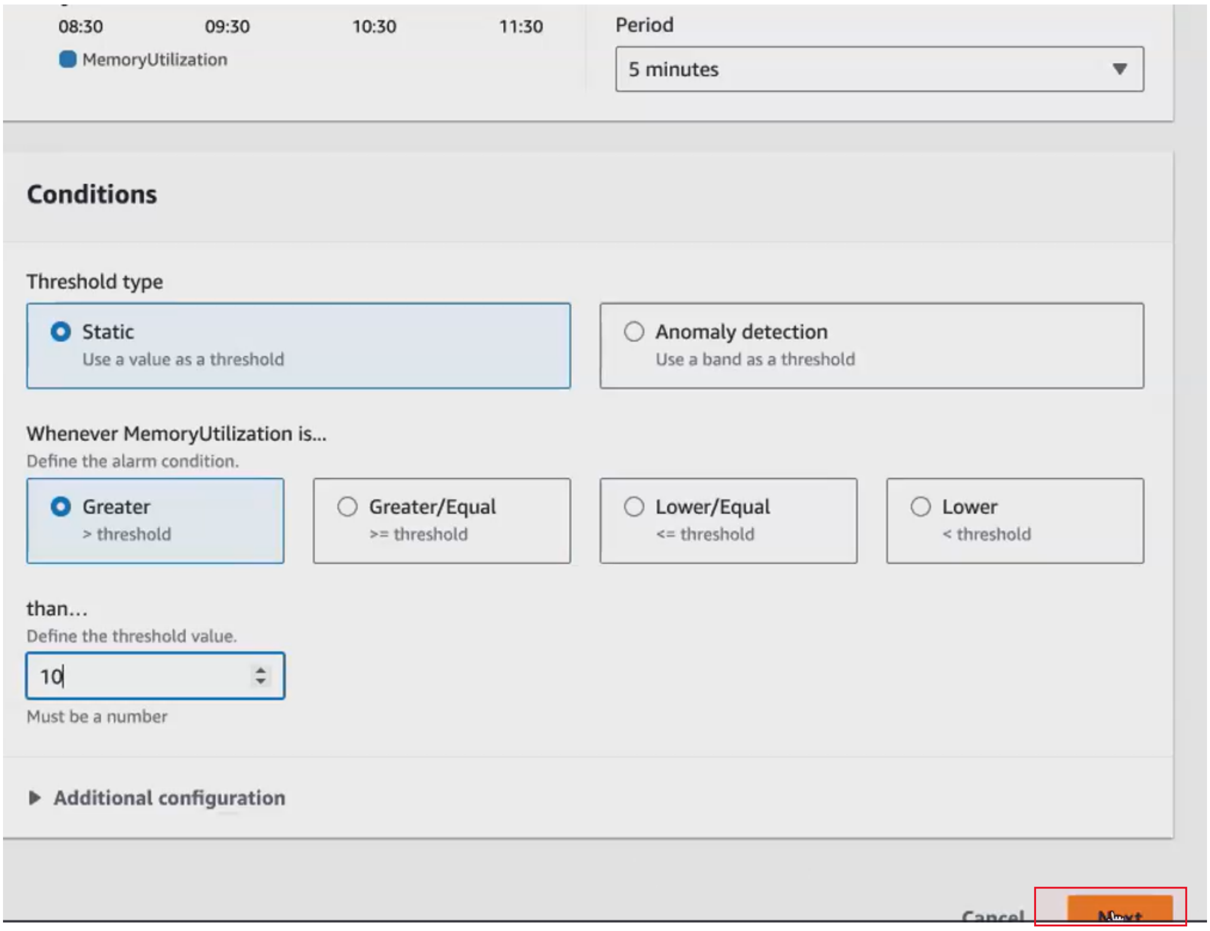


**Click on the matrix,see the memory utilization. Click create alarm [whenever attain threshold value for our memory create alarm]**



| ☑ Metric name 1/1 | Alarms |
|---|---|
| ☑ MemoryUtilization | No alarms |

**Give threshold value as 10. threshold value, it will send you a notification**



Period

5 minutes ▼

## Conditions

Threshold type

● Static
Use a value as a threshold

○ Anomaly detection
Use a band as a threshold

Whenever MemoryUtilization is...
Define the alarm condition.

● Greater
> threshold

○ Greater/Equal
>= threshold

○ Lower/Equal
<= threshold

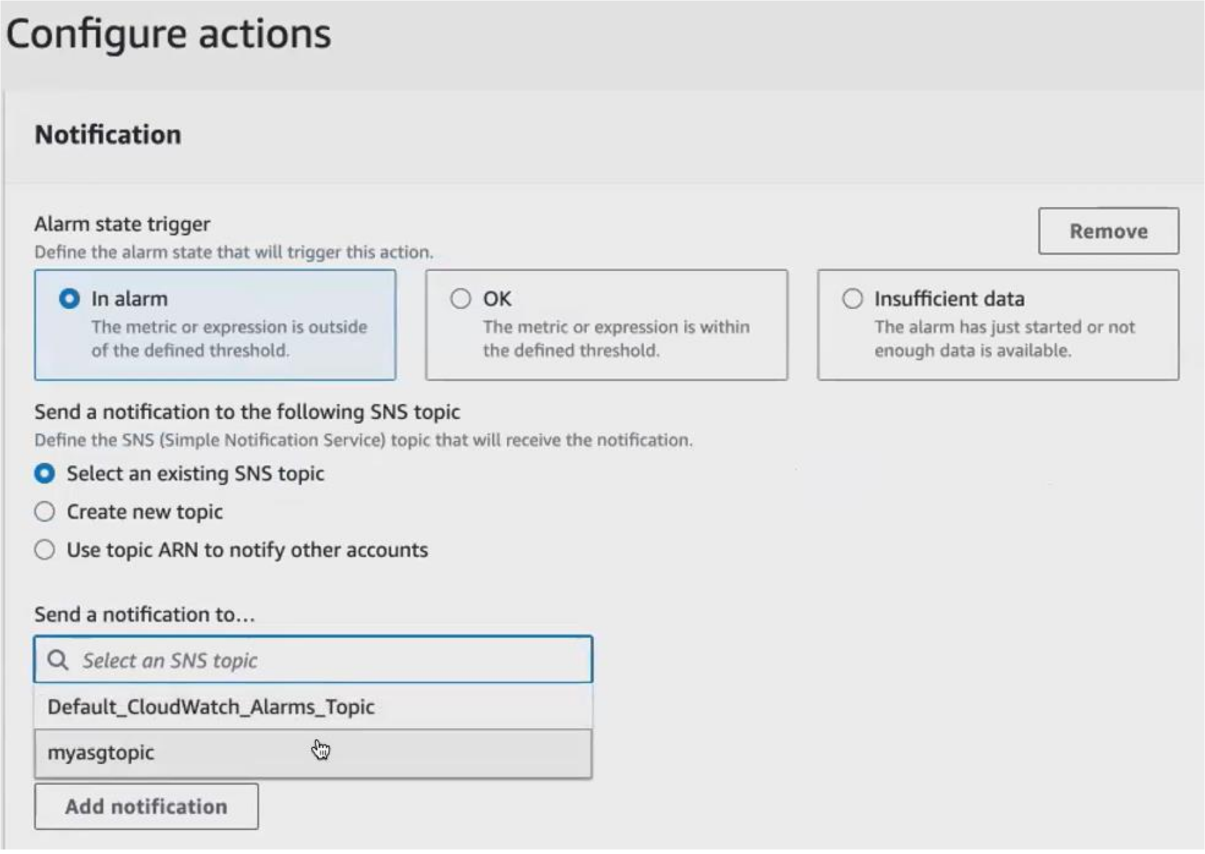
○ Lower
< threshold

than...
Define the threshold value.

10

Must be a number

▶ Additional configuration



Configure actions

Notification

Alarm state trigger

Define the alarm state that will trigger this action.

● In alarm
The metric or expression is outside of the defined threshold.

○ OK
The metric or expression is within the defined threshold.

○ Insufficient data
The alarm has just started or not enough data is available.

Send a notification to the following SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification.

● Select an existing SNS topic

○ Create new topic

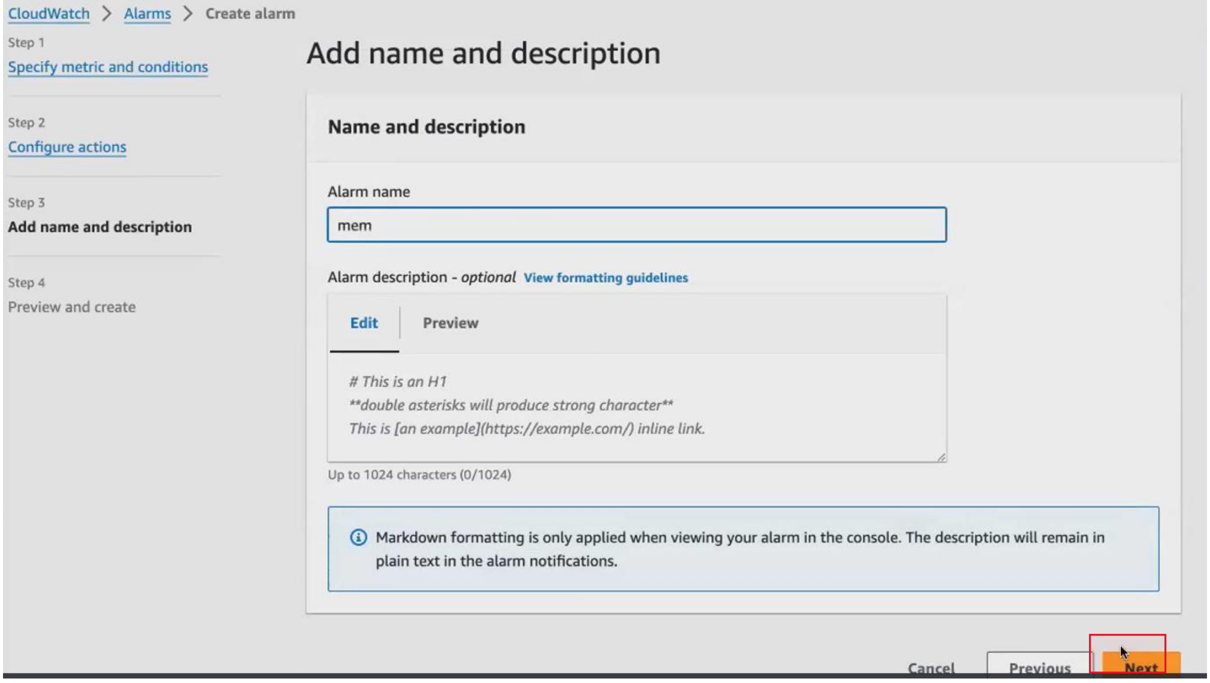
○ Use topic ARN to notify other accounts

Send a notification to...

Select an SNS topic

Default_CloudWatch_Alarms_Topic

myasgtopic



Add name and description

Name and description

Alarm name

mem

Alarm description - *optional* View formatting guidelines

*# This is an H1*
***double asterisks will produce strong character***
*This is [an example](https://example.com/) inline link.*

Up to 1024 characters (0/1024)

ⓘ Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

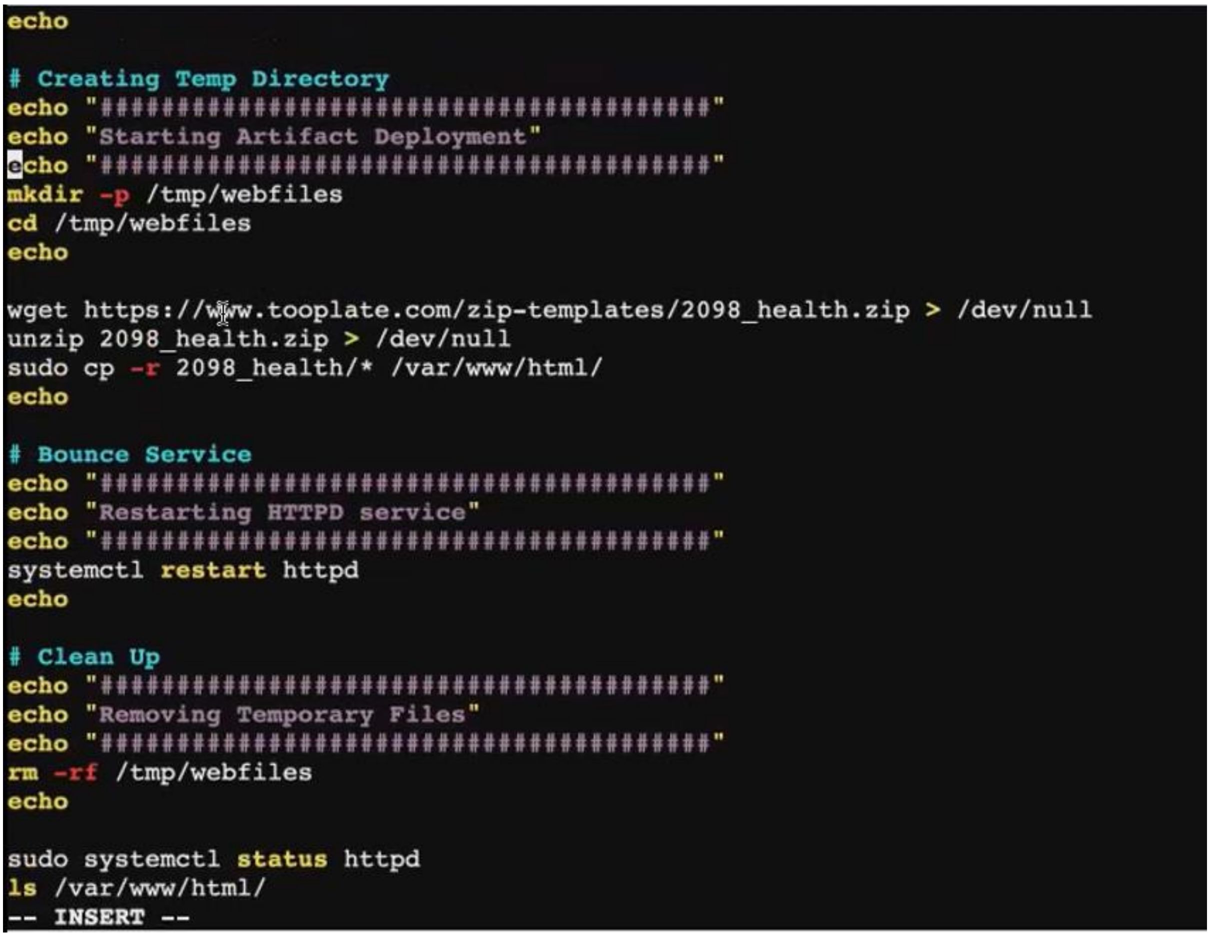
**Alarm has been created**

You can see the metric that we have created, and you can see the namespace under the namespace. You will have the metrics. So under your namespace, you can have multiple metrics. So now I have created only one metric. Same way. You can create multiple metrics. So you can see under cloud watch under using aws. Cloud watch put metrics. I've created only one metric. So in the same way, I can create multiple metrics to monitor the memory itself.

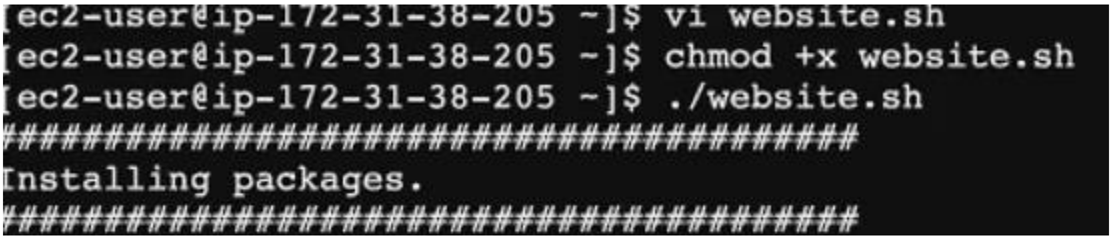
you can have a namespace under a particular namespace. You can have multiple metrics. So here, if you see here, under your custom metric.so under these are, you can see in the bottom. You have Aws, namespaces which are default namespaces. Okay, so you can have under Ec. 2. This is the namespace, Ec. 2 is the namespace. These are default namespaces. So for the custom namespace for the custom metric. We have created the namespace, and we have given this as the name. So under your namespace, you can have multiple metrics.

**you can monitor your ec2 the same way. You can also monitor your application:**

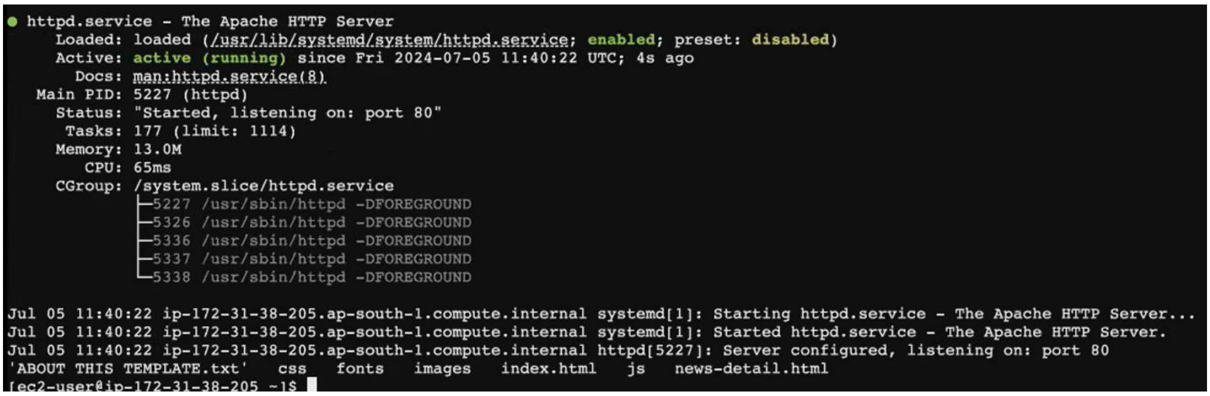
**we will create a small application in this ec2.** So before monitoring, using your cloud watch what we do, we will create an application. We will launch a simple Apache application. Let me run the script to launch the application. You can use if you have any old script that I've given shared with you, you can use the same script. Also think for launching the website we used. We used a script. Right? We can use the same script here.I think, in the load balancer document, you can find it load balancer or the user data document. You can find it because in the user data will be using shell script only to launch the website. So we can use the same script here. Let me also copy the same script, and I will just put it here said, because to monitor the application, you need to have an application running.



```
echo

# Creating Temp Directory
echo "########################################"
echo "Starting Artifact Deployment"
echo "########################################"
mkdir -p /tmp/webfiles
cd /tmp/webfiles
echo

wget https://www.tooplate.com/zip-templates/2098_health.zip > /dev/null
unzip 2098_health.zip > /dev/null
sudo cp -r 2098_health/* /var/www/html/
echo

# Bounce Service
echo "########################################"
echo "Restarting HTTPD service"
echo "########################################"
systemctl restart httpd
echo

# Clean Up
echo "########################################"
echo "Removing Temporary Files"
echo "########################################"
rm -rf /tmp/webfiles
echo

sudo systemctl status httpd
ls /var/www/html/
-- INSERT --
```



```
[ec2-user@ip-172-31-38-205 ~]$ vi website.sh
[ec2-user@ip-172-31-38-205 ~]$ chmod +x website.sh
[ec2-user@ip-172-31-38-205 ~]$ ./website.sh
########################################
Installing packages.
########################################
```



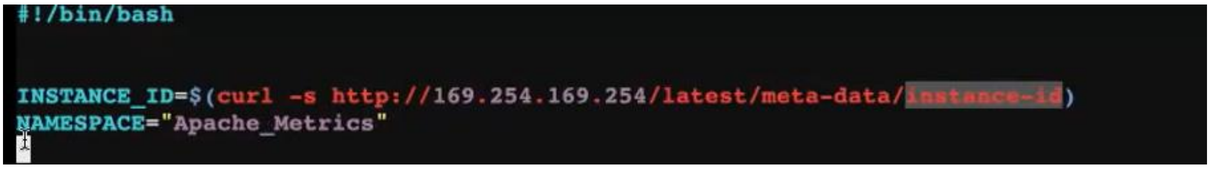
```
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-07-05 11:40:22 UTC; 4s ago
       Docs: man:httpd.service(8)
   Main PID: 5227 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 1114)
     Memory: 13.0M
        CPU: 65ms
     CGroup: /system.slice/httpd.service
             ├─5227 /usr/sbin/httpd -DFOREGROUND
             ├─5326 /usr/sbin/httpd -DFOREGROUND
             ├─5336 /usr/sbin/httpd -DFOREGROUND
             ├─5337 /usr/sbin/httpd -DFOREGROUND
             └─5338 /usr/sbin/httpd -DFOREGROUND

Jul 05 11:40:22 ip-172-31-38-205.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Jul 05 11:40:22 ip-172-31-38-205.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Jul 05 11:40:22 ip-172-31-38-205.ap-south-1.compute.internal httpd[5227]: Server configured, listening on: port 80
'ABOUT THIS TEMPLATE.txt'   css   fonts   images   index.html   js   news-detail.html
[ec2-user@ip-172-31-38-205 ~]$
```

## Webserver is running

**Now you can see Apache is running. Now the Apache we have installed Apache, and the application is running inside my Ec2. Now, what we do even create a metric here. So before creating your custom  metrics make sure to install Cloud Watch Agent. Since we have already installed Cloud watch Agent for checking the memory. I'm not installing it, but whenever you're creating a custom metric, make sure you install the Cloud Watch agent, and whichever ec2 you are monitoring it.**

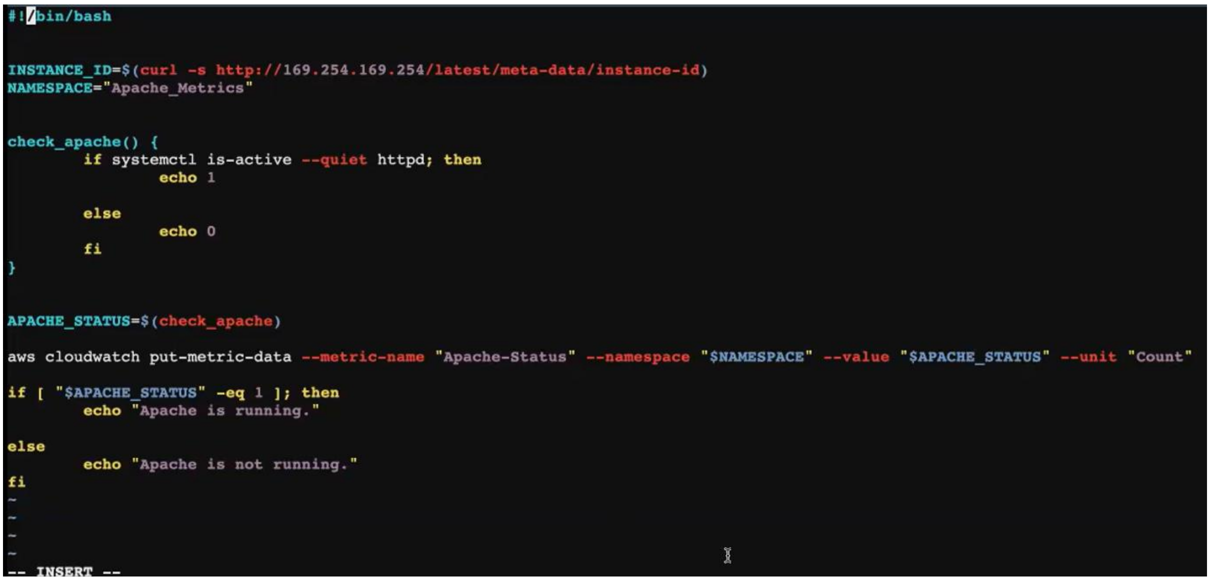
**now I will create a shell script to create another custom metric. This is to monitor the application. So, using this metric, we saw how to monitor the infrastructure, using your CPU, your system, how to monitor your Ec2. Now we will see how whether this is to monitor your application that is running inside the Ec2.**

**I have used. Instance id here you can hard code your Id. Okay, you can hard code your instance, Id or you. You are not sure about the instance Id, because in real time we will be launching our instances through terraform. his command you will be using  http 169.254 give complete detail about ec2.**

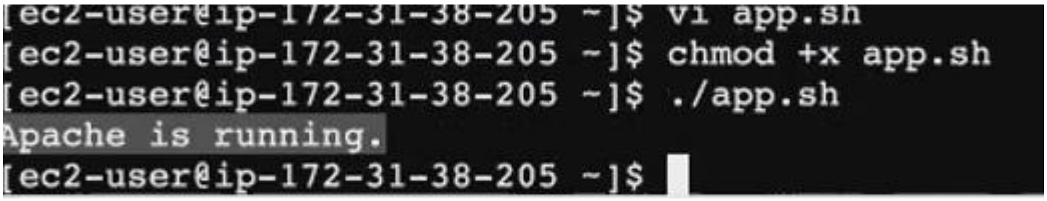


```
#!/bin/bash


INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
NAMESPACE="Apache_Metrics"
```

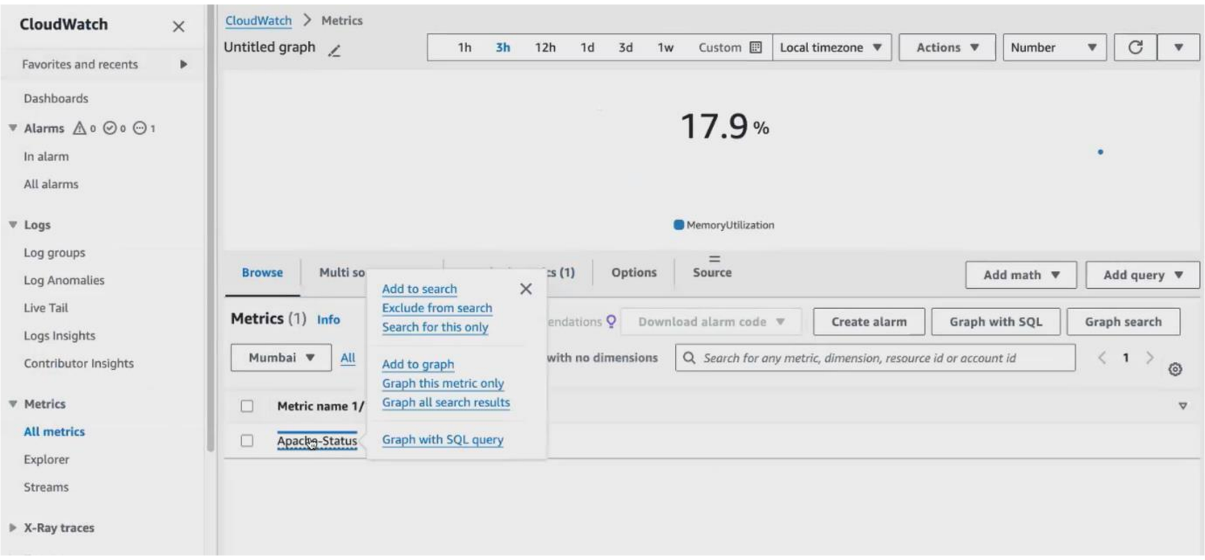
**I'm creating function. This function name can be anything you can give any name for your function.Any name you can give, but at the end you have to give a open parenthesis and close parentheses. Since I'm using shell script you. Just this is out in the shell script. We create a function. So what this function should do, this function will check if my Apache application is running or not.**



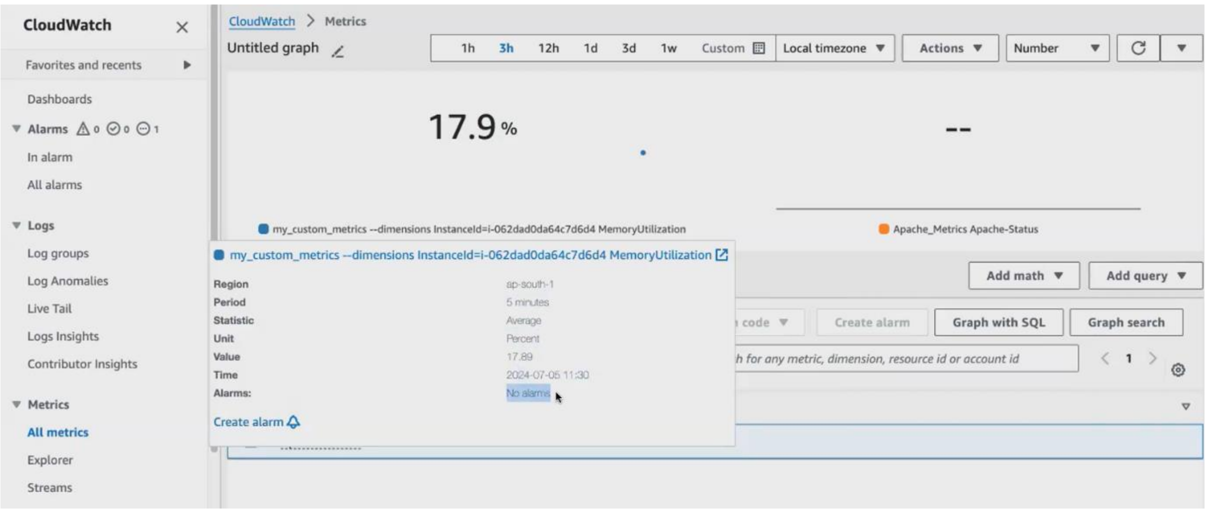
```
#!/bin/bash

INSTANCE_ID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)
NAMESPACE="Apache_Metrics"

check_apache() {
        if systemctl is-active --quiet httpd; then
                echo 1

        else
                echo 0
        fi
}

APACHE_STATUS=$(check_apache)

aws cloudwatch put-metric-data --metric-name "Apache-Status" --namespace "$NAMESPACE" --value "$APACHE_STATUS" --unit "Count"

if [ "$APACHE_STATUS" -eq 1 ]; then
        echo "Apache is running."

else
        echo "Apache is not running."
fi
~
~
~
~
-- INSERT --
```



```
[ec2-user@ip-172-31-38-205 ~]$ vi app.sh
[ec2-user@ip-172-31-38-205 ~]$ chmod +x app.sh
[ec2-user@ip-172-31-38-205 ~]$ ./app.sh
Apache is running.
[ec2-user@ip-172-31-38-205 ~]$ 
```



## Apache custom matrix created



my_custom_metrics --dimensions InstanceId=i-062dad0da64c7d6d4 MemoryUtilization

| | |
|---|---|
| Region | ap-south-1 |
| Period | 5 minutes |
| Statistic | Average |
| Unit | Percent |
| Value | 17.89 |
| Time | 2024-07-05 11:30 |
| Alarms: | No alarms |

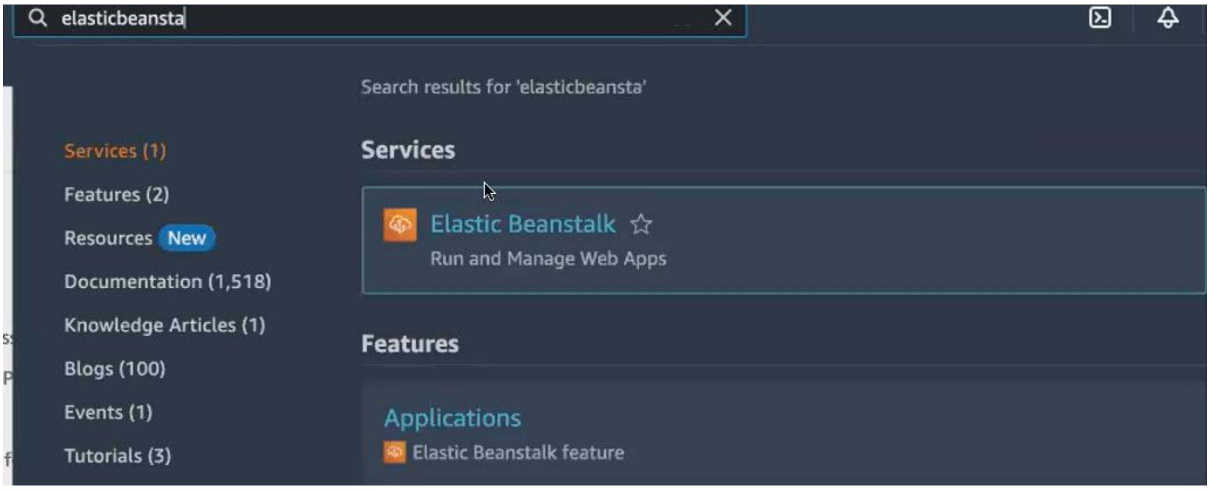
Create alarm

https://docs.google.com/document/d/1PaEBggi56c6DFcw9MVqt0gN5kDOr0Z8z0XLuT_CmAtU/edit

**link for cloud watch scrpting**

**if you want to monitor your Ec2, you can write a custom metric, but mostly we. We don't use the custom metrics. Rather, we will use the default. Metrics itself. Aws has got lot of default metrics. We will make use of those metrics itself in case if you want to create your own metric.**

## Elasticbeanstalk:



**it is mostly used by your developers rather than the devops engineer or the operations team. So your developers what they do, they develop the application. But they are not aware of the deployment part. So like the Devops engineer will do the deployment or the operations team will be doing the deployment. Okay? So here, what they can do, your developer. They themselves can deploy the application if they want to deploy it in the Aws environment, they can. They themselves can deploy it in the Aws environment using elastic beanstalk. Okay? So what elastic beanstalk does it will give you a end to end application management platform.**

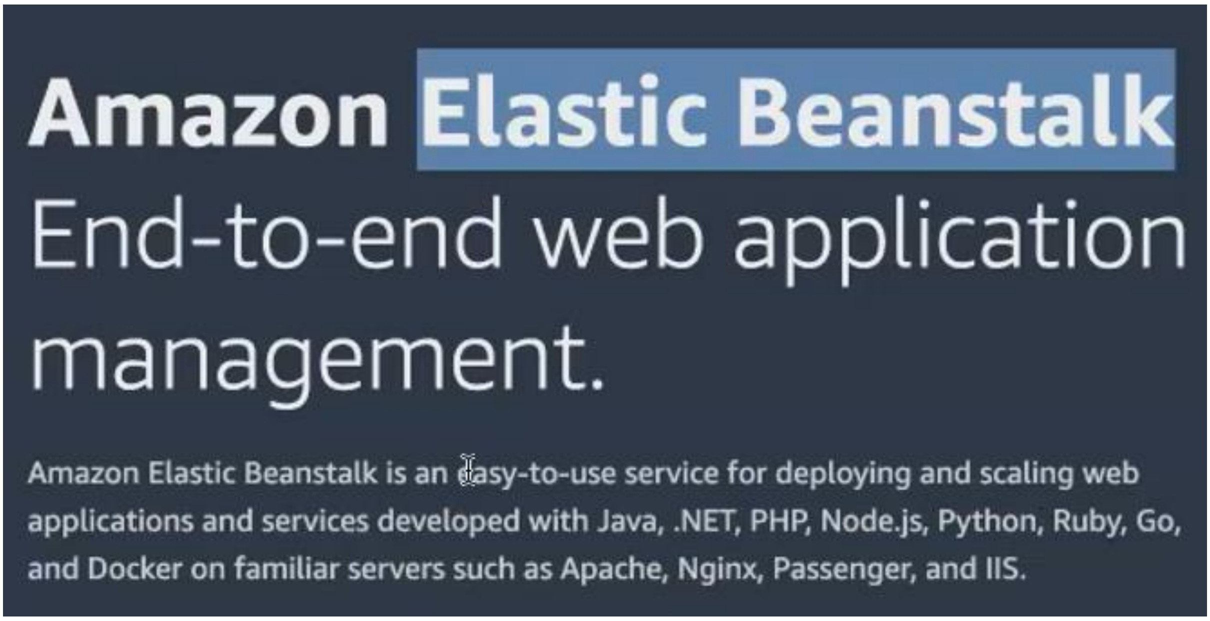
infrastructure as service platform [ec2,s3]

platform as a service [ **Elasticbeanstalk]**

platform as a service means **it also gives you the infrastructure along with the infrastructure, It provides us the platform also.**

**Example: Now, I've launched a Apache application. Okay? So what I did, I created an Ec2 . inside the Ec. 2 I installed WGet command to install Apache 2 then Unzip? So I install all those tools before launching my application.**

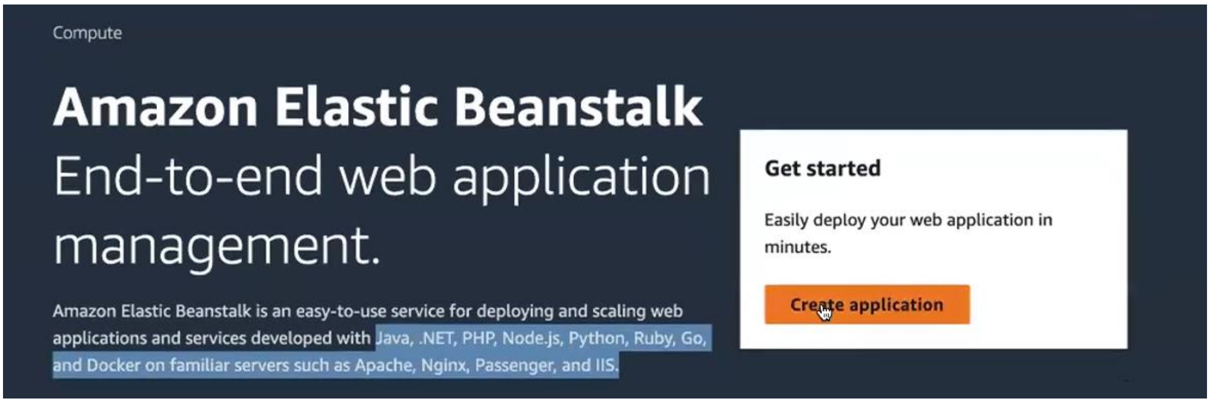
**myself set up the environment to launch the application. But if you're using elastic being stuck. You don't have to worry about setting up the environment. So you get the infrastructure along with the infrastructure. You will also get the platform. You just need to have the application. Okay? So what we do, we will completely configure everything in the console.**



Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.



Compute

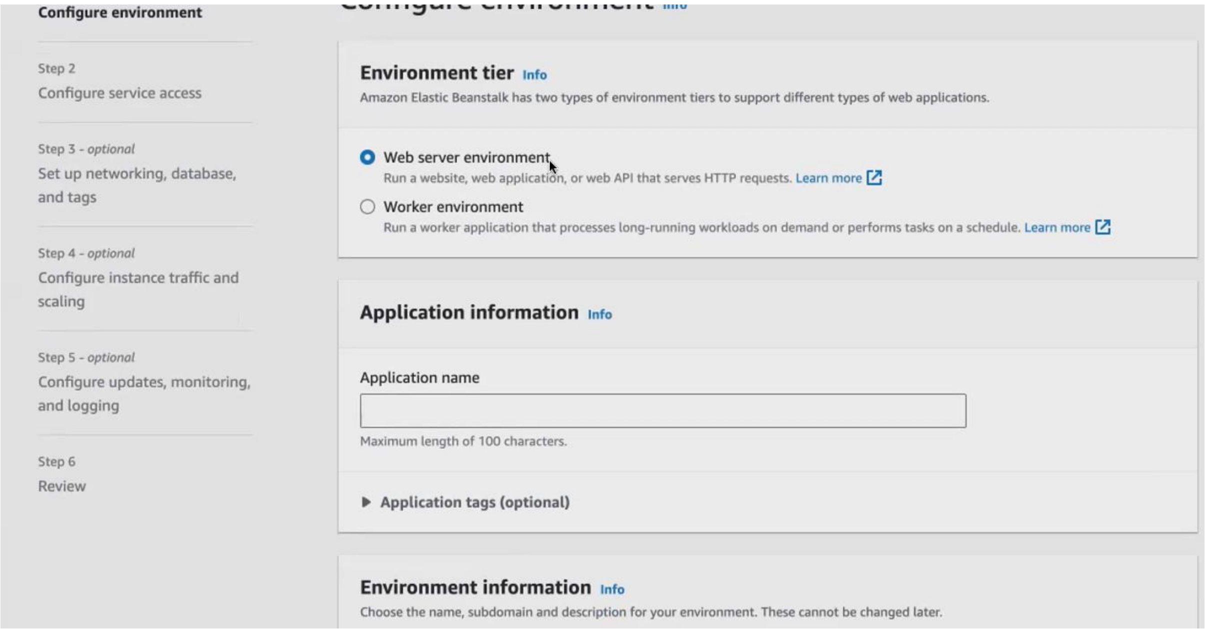
Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.



Configure environment

Step 2
Configure service access

Step 3 - *optional*
Set up networking, database, and tags

Step 4 - *optional*
Configure instance traffic and scaling

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

● Web server environment
Run a website, web application, or web API that serves HTTP requests. Learn more

○ Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more

Application information Info

Application name

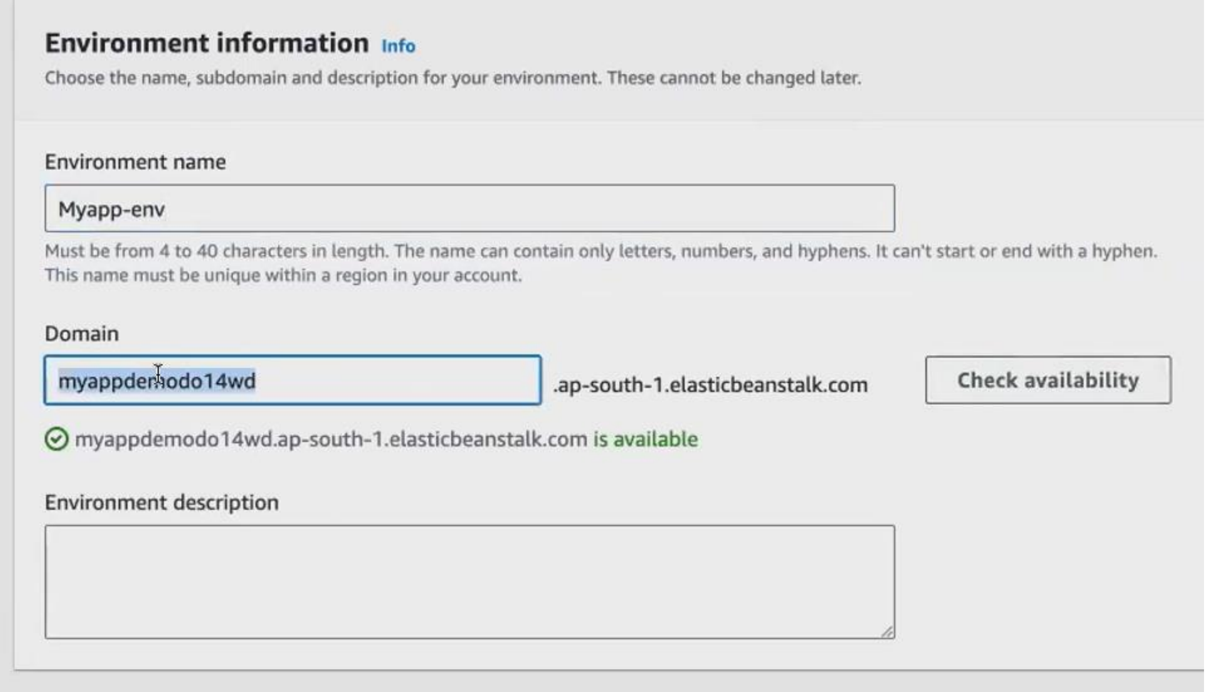
Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Either you want a web server environment or a worker environment .so we we mostly go with if you're using a web application, you go with web server, environment worker environment is for the batch processing. If you're running any heavy workload or batch processing, you go with worker environment. But mostly we go with only web based applications. So you use web server environment. Okay? So this web server environment can use application can run the application which is accessible by Http and Https request. Okay, so you give your application name your wish.



**Environment information** Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Myapp-env

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain

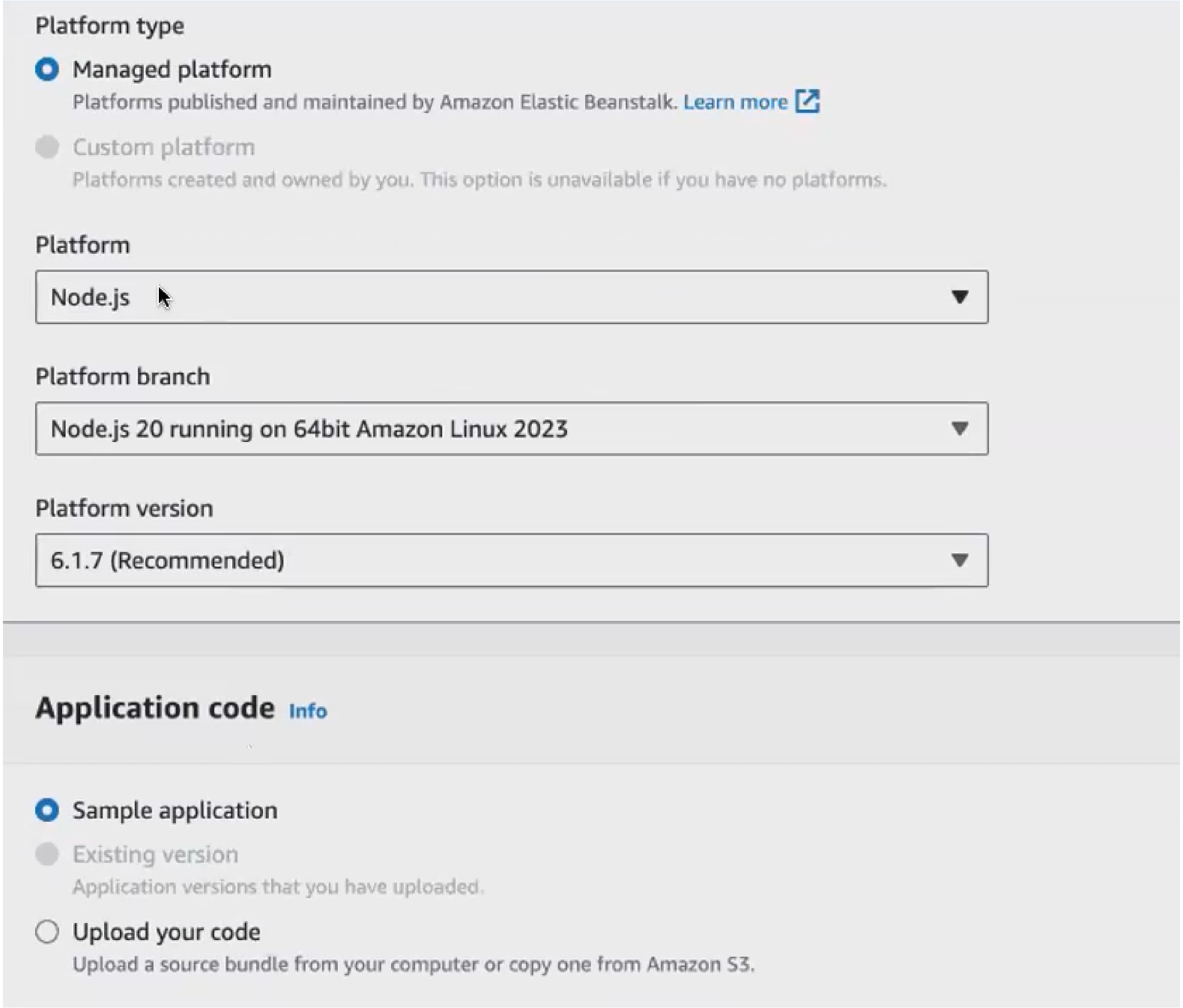
myappdemodo14wd .ap-south-1.elasticbeanstalk.com

myappdemodo14wd.ap-south-1.elasticbeanstalk.com is available

Environment description

After give domain name check availability .then only shows green colour.

you have to select the platform type. As I told you, your elastic beanstalk gives you the platform to run your application. Now, what the developer will have developer will have only the application. so my developer can have a Java application. Node Js application or a python application. So based on his application, you have to select the platform. So if I'm using a node Js application. I have to just select the node Js platform and the platform branch I can select which branch he is using, so my developer will know whether he's using Nodejs 2018, or 17, so he can pick the version accordingly, and and the platform version



Platform type

● Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. Learn more

○ Custom platform

Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Node.js ▼

Platform branch

Node.js 20 running on 64bit Amazon Linux 2023 ▼

Platform version

6.1.7 (Recommended) ▼

Application code Info

● Sample application

○ Existing version

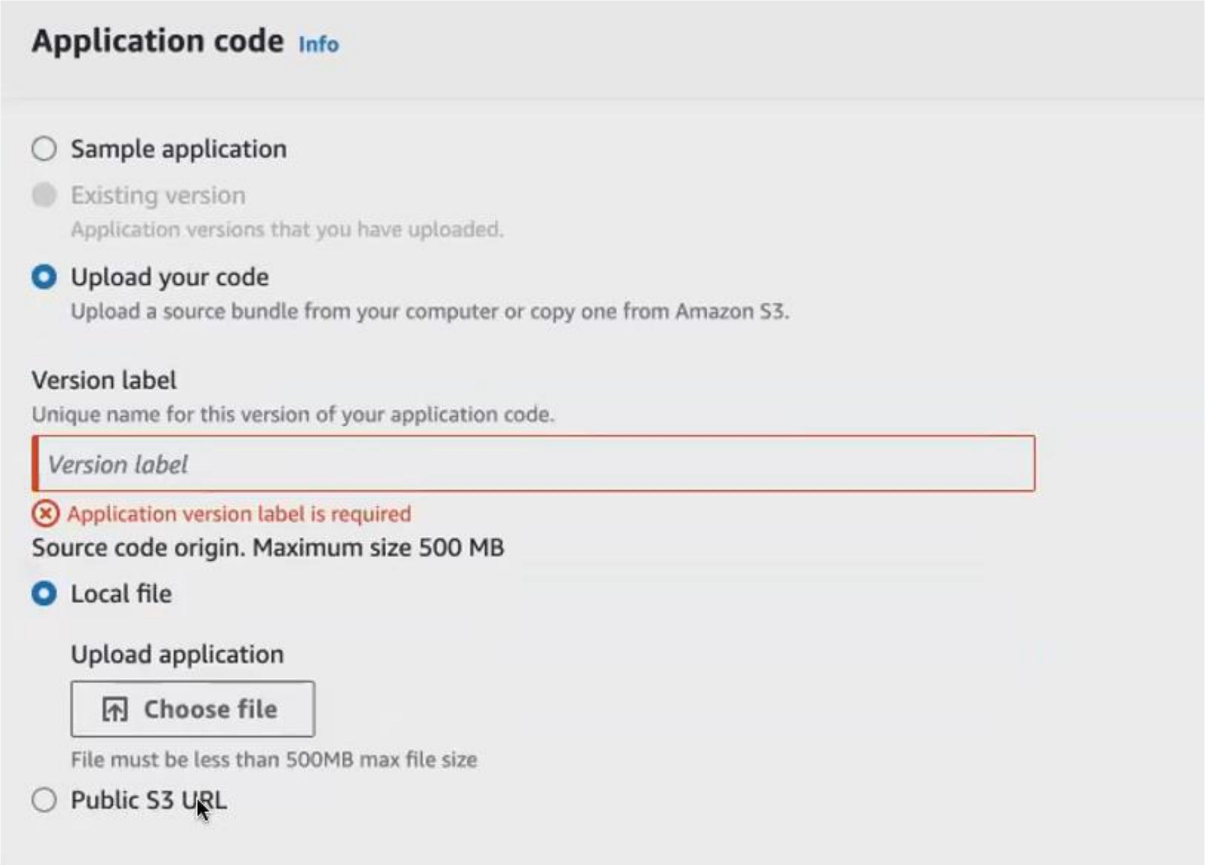
Application versions that you have uploaded.

○ Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

**it will launch an ec2 inside the Ec2 , it will install a Nodejs application.**

**if I give you a Nodejs application, what you will do first, you have to set up the environment, you have to install Nodejs, and then you have to run the application in that Ec 2.** So same thing, your elastic bean is doing it, for you just have to select the platform. You don't have to worry about configuring the platform in the Ec. 2. You just select the platform name and the version next you can. If you have the application. If I have the Nojs application, I can just upload it here. Okay, I can just upload my Node js application here.



Application code Info

○ Sample application

● Existing version
Application versions that you have uploaded.

● Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Version label
Unique name for this version of your application code.

Version label

ⓧ Application version label is required

Source code origin. Maximum size 500 MB

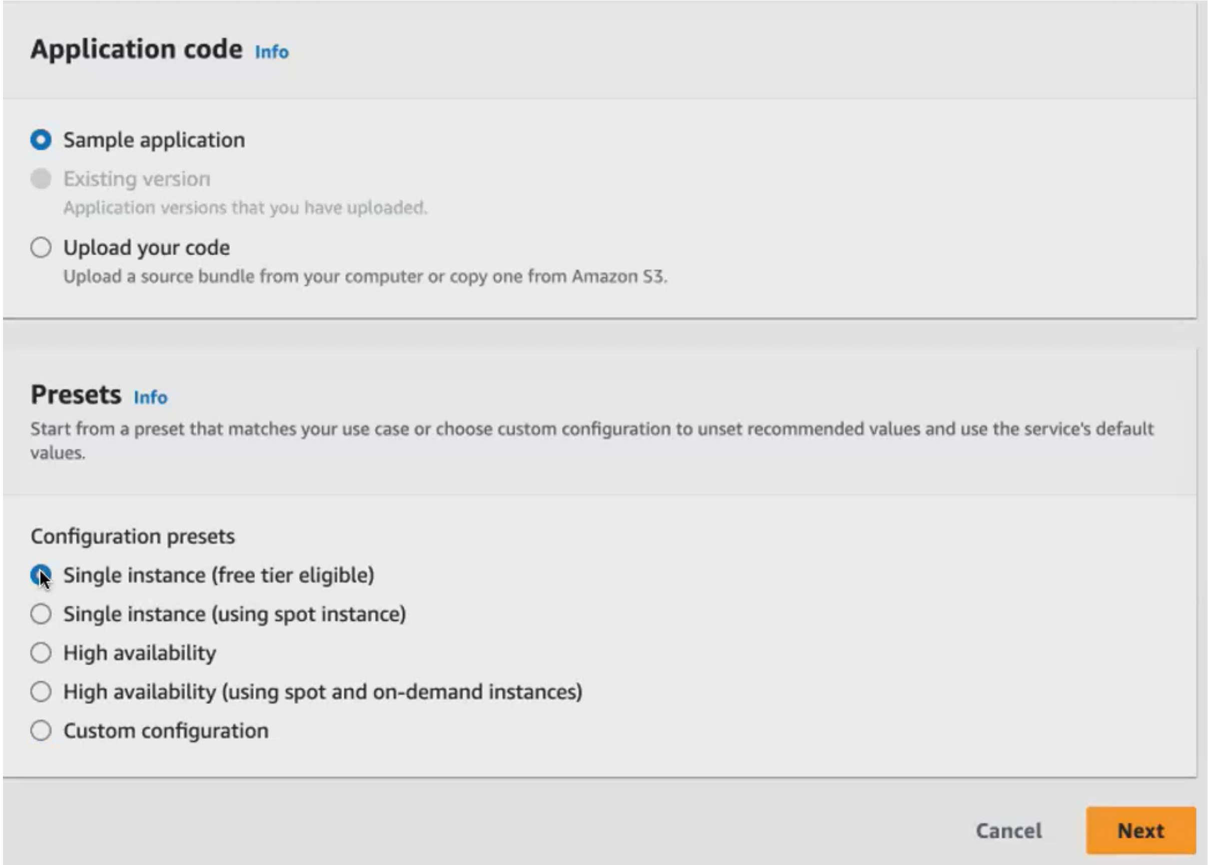
● Local file

Upload application

File must be less than 500MB max file size

○ Public S3 URL

**If we have code inside local machine just upload it or give sample application. Now we give sample application.**



Application code

● Sample application

○ Existing version

Application versions that you have uploaded.

○ Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

Presets

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

● Single instance (free tier eligible)

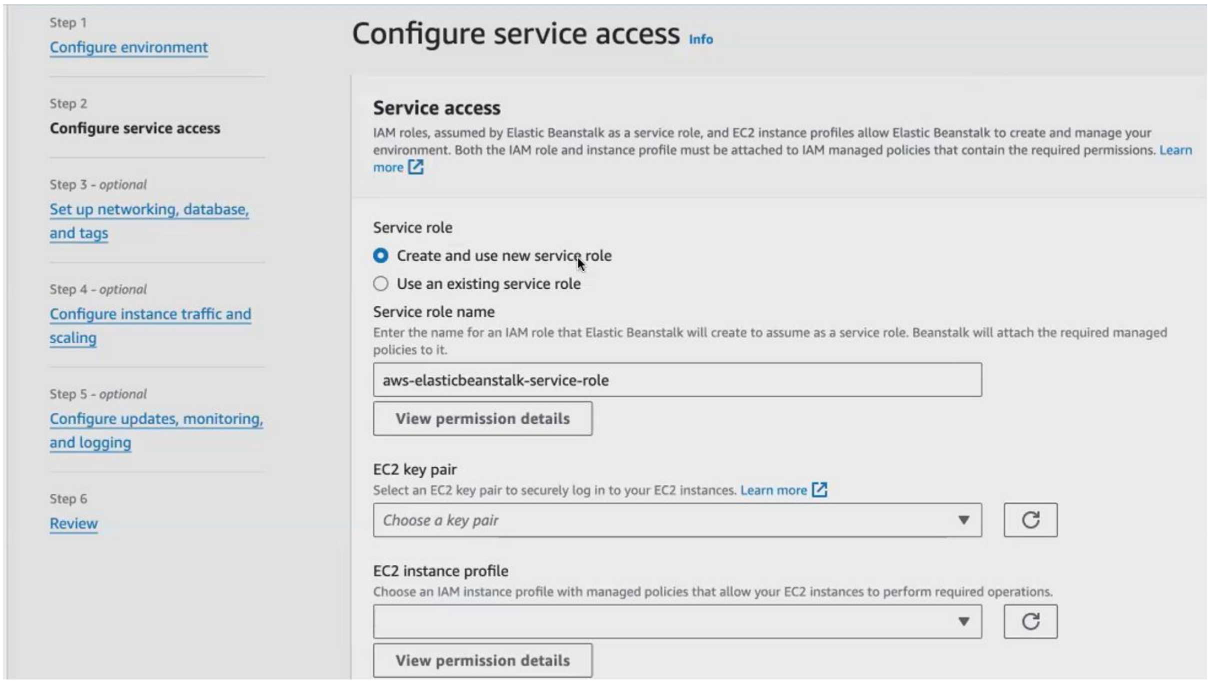
○ Single instance (using spot instance)

○ High availability

○ High availability (using spot and on-demand instances)

○ Custom configuration

**your elastic Beanstalk will launch an Ec2. So your elastic beanstalk is a different service, and your Ec2 is a different service, my elastic beanstalk wants to talk with my Ec 2. We have** to create a role**. We have to create a service role. So only with the help of service role. Your elastic beanstalk can connect with your easy to ec2.**



Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Set up networking, database, and tags

Step 4 - optional
Configure instance traffic and scaling

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Configure service access Info

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more

Service role

● Create and use new service role

○ Use an existing service role

Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

Choose a key pair

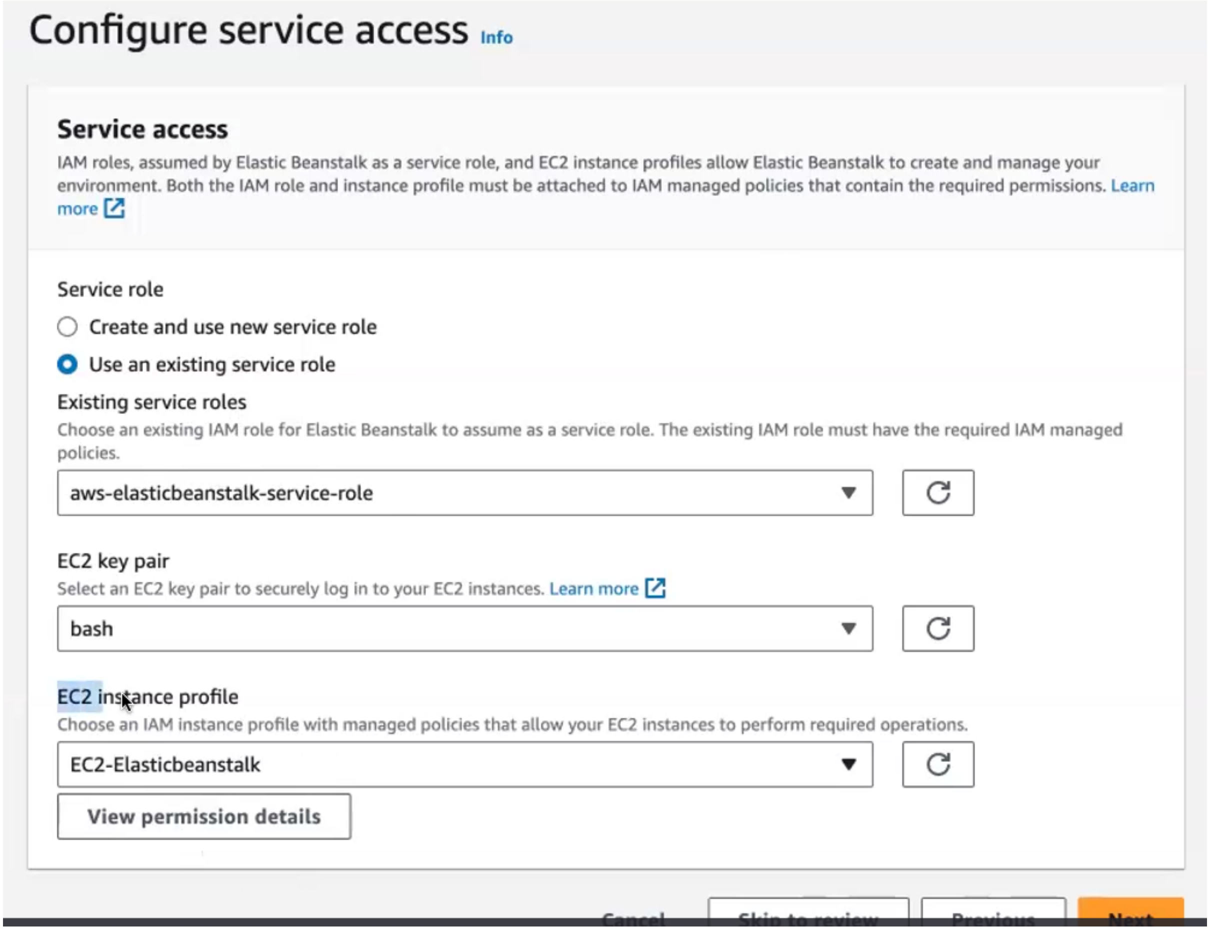
EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

**1st time give create new service role in IAM user.**

**Here yasmin give already created role**

**here in the elastic beanstalk, I am creating 2 IAM role, 1 1st IAM role is for the elastic beanstalk that you get it by default from the elastic beanstalk itself. You don't have to create it manually, wherein the second IAM role is for the Ec. 2. Okay? So your Ec 2 should set the elastic bean stocks application deployment. So for that, I go to my IAM go and click on roles. So we will create a role aws services.**



## Configure service access Info

### Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more

Service role

○ Create and use new service role

● Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

bash

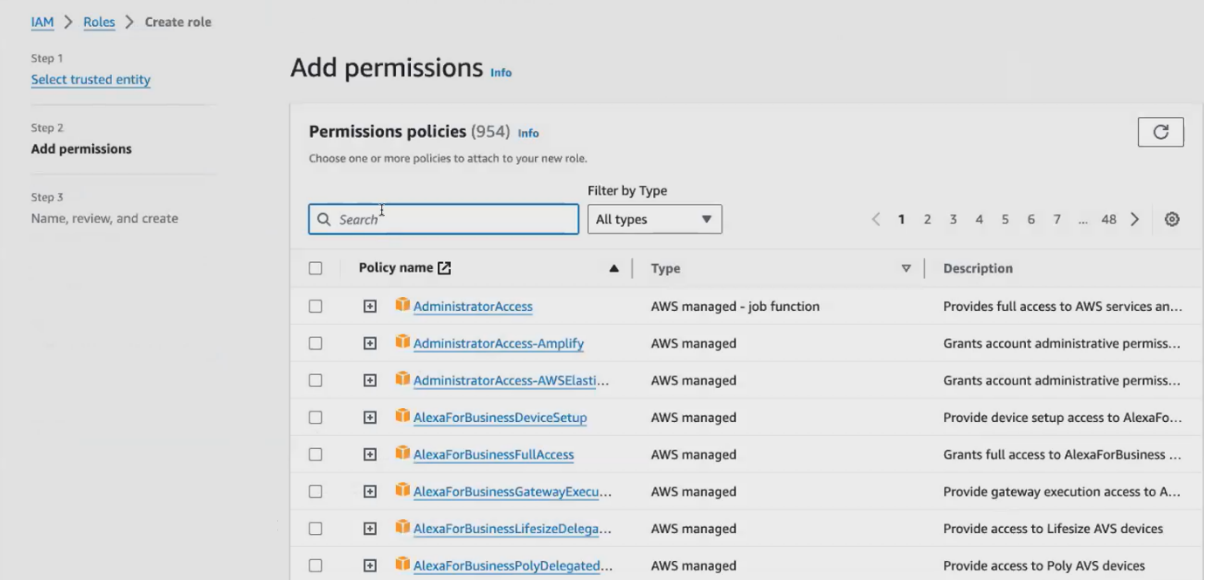
EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

EC2-Elasticbeanstalk

**have to give click on the view permission details. So I have to give elastic bean stalk webtier, elastic bean stalk worktier , , elastic bean stalk multi-tier container docker, okay, these 3 permissions I have to attach for this iam rule.**

**Goto iam user → roles→create roles**



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

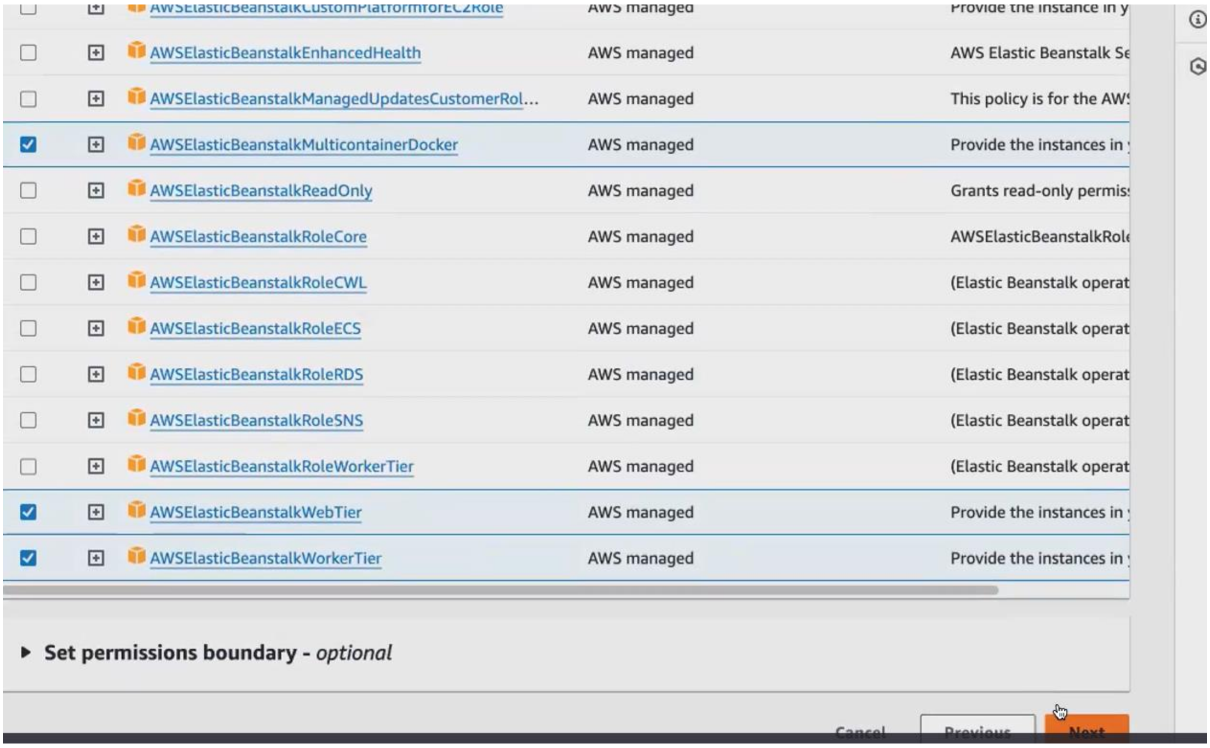
## Add permissions Info

**Permissions policies** (954) Info

Choose one or more policies to attach to your new role.

Filter by Type: All types

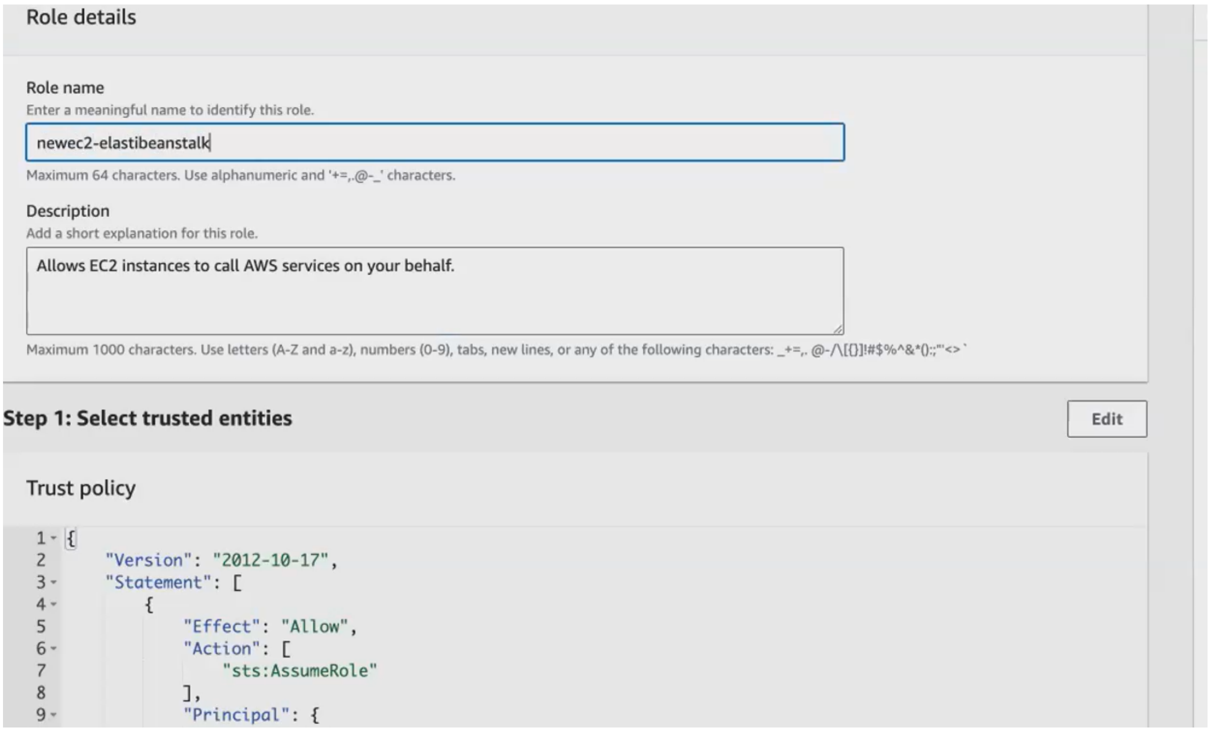
| □ | Policy name | Type | Description |
|---|---|---|---|
| □ | AdministratorAccess | AWS managed - job function | Provides full access to AWS services an... |
| □ | AdministratorAccess-Amplify | AWS managed | Grants account administrative permiss... |
| □ | AdministratorAccess-AWSElasti... | AWS managed | Grants account administrative permiss... |
| □ | AlexaForBusinessDeviceSetup | AWS managed | Provide device setup access to AlexaFo... |
| □ | AlexaForBusinessFullAccess | AWS managed | Grants full access to AlexaForBusiness ... |
| □ | AlexaForBusinessGatewayExecu... | AWS managed | Provide gateway execution access to A... |
| □ | AlexaForBusinessLifesizeDelega... | AWS managed | Provide access to Lifesize AVS devices |
| □ | AlexaForBusinessPolyDelegated... | AWS managed | Provide access to Poly AVS devices |



| | | | | |
|---|---|---|---|---|
| □ | ⊞ | AWSElasticBeanstalkCustomPlatformforEC2Role | AWS managed | Provide the instance in y |
| □ | ⊞ | AWSElasticBeanstalkEnhancedHealth | AWS managed | AWS Elastic Beanstalk Se |
| □ | ⊞ | AWSElasticBeanstalkManagedUpdatesCustomerRol... | AWS managed | This policy is for the AWS |
| ☑ | ⊞ | AWSElasticBeanstalkMulticontainerDocker | AWS managed | Provide the instances in |
| □ | ⊞ | AWSElasticBeanstalkReadOnly | AWS managed | Grants read-only permis |
| □ | ⊞ | AWSElasticBeanstalkRoleCore | AWS managed | AWSElasticBeanstalkRol |
| □ | ⊞ | AWSElasticBeanstalkRoleCWL | AWS managed | (Elastic Beanstalk operat |
| □ | ⊞ | AWSElasticBeanstalkRoleECS | AWS managed | (Elastic Beanstalk operat |
| □ | ⊞ | AWSElasticBeanstalkRoleRDS | AWS managed | (Elastic Beanstalk operat |
| □ | ⊞ | AWSElasticBeanstalkRoleSNS | AWS managed | (Elastic Beanstalk operat |
| □ | ⊞ | AWSElasticBeanstalkRoleWorkerTier | AWS managed | (Elastic Beanstalk operat |
| ☑ | ⊞ | AWSElasticBeanstalkWebTier | AWS managed | Provide the instances in |
| ☑ | ⊞ | AWSElasticBeanstalkWorkerTier | AWS managed | Provide the instances in |

▸ **Set permissions boundary -** *optional*

## Click next.give role name



Role details

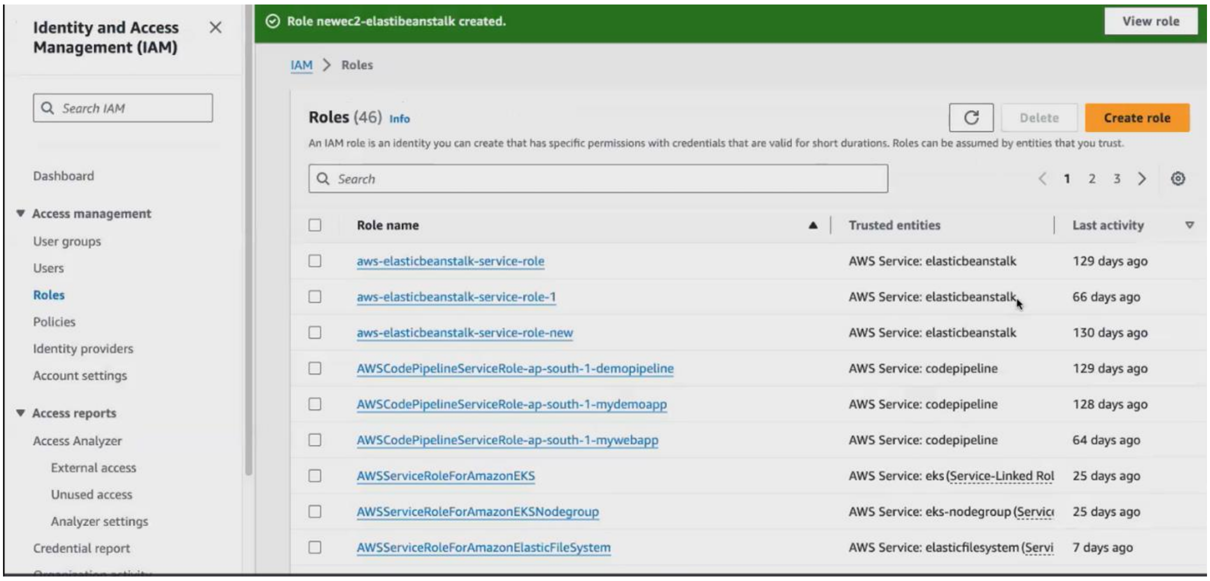
Role name
Enter a meaningful name to identify this role.
newec2-elastibeanstalk
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^&*();:"'<>`

**Step 1: Select trusted entities**

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
```



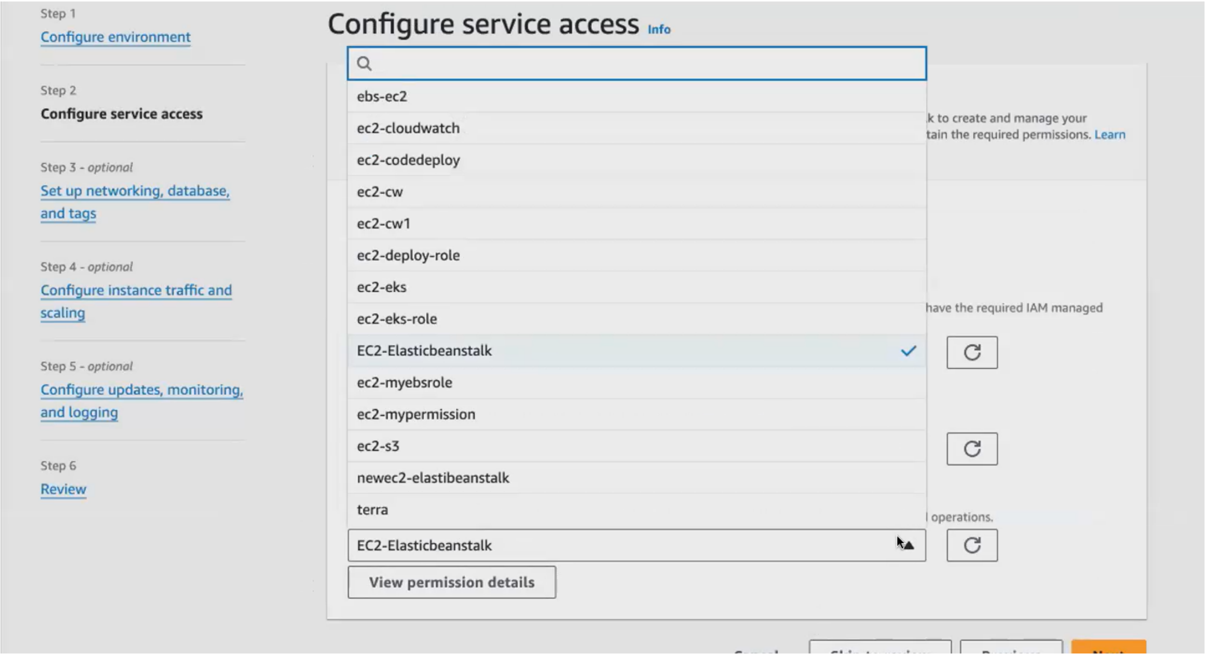
Role newec2-elastibeanstalk created.

Roles (46) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

| Role name | Trusted entities | Last activity |
|---|---|---|
| aws-elasticbeanstalk-service-role | AWS Service: elasticbeanstalk | 129 days ago |
| aws-elasticbeanstalk-service-role-1 | AWS Service: elasticbeanstalk | 66 days ago |
| aws-elasticbeanstalk-service-role-new | AWS Service: elasticbeanstalk | 130 days ago |
| AWSCodePipelineServiceRole-ap-south-1-demopipeline | AWS Service: codepipeline | 129 days ago |
| AWSCodePipelineServiceRole-ap-south-1-mydemoapp | AWS Service: codepipeline | 128 days ago |
| AWSCodePipelineServiceRole-ap-south-1-mywebapp | AWS Service: codepipeline | 64 days ago |
| AWSServiceRoleForAmazonEKS | AWS Service: eks (Service-Linked Rol | 25 days ago |
| AWSServiceRoleForAmazonEKSNodegroup | AWS Service: eks-nodegroup (Servic | 25 days ago |
| AWSServiceRoleForAmazonElasticFileSystem | AWS Service: elasticfilesystem (Servi | 7 days ago |

## Role created.use this role in e.b.s



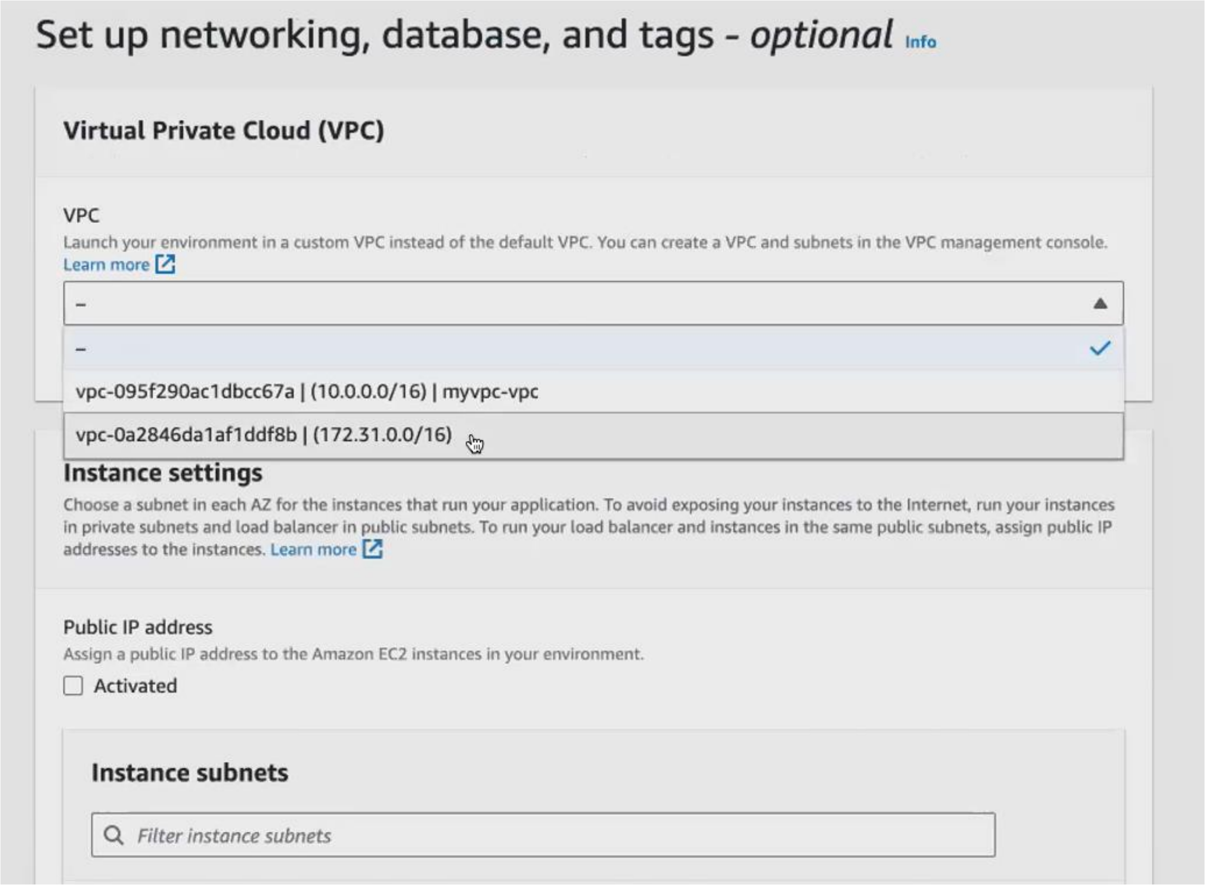
Configure service access Info

ebs-ec2
ec2-cloudwatch
ec2-codedeploy
ec2-cw
ec2-cw1
ec2-deploy-role
ec2-eks
ec2-eks-role
EC2-Elasticbeanstalk ✓
ec2-myebsrole
ec2-mypermission
ec2-s3
newec2-elastibeanstalk
terra

EC2-Elasticbeanstalk

k to create and manage your
tain the required permissions. Learn
have the required IAM managed
operations.

### Select default vpc



## Set up networking, database, and tags - *optional* Info

**Virtual Private Cloud (VPC)**

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. Learn more

–
– ✓
vpc-095f290ac1dbcc67a | (10.0.0.0/16) | myvpc-vpc
vpc-0a2846da1af1ddf8b | (172.31.0.0/16)

**Instance settings**

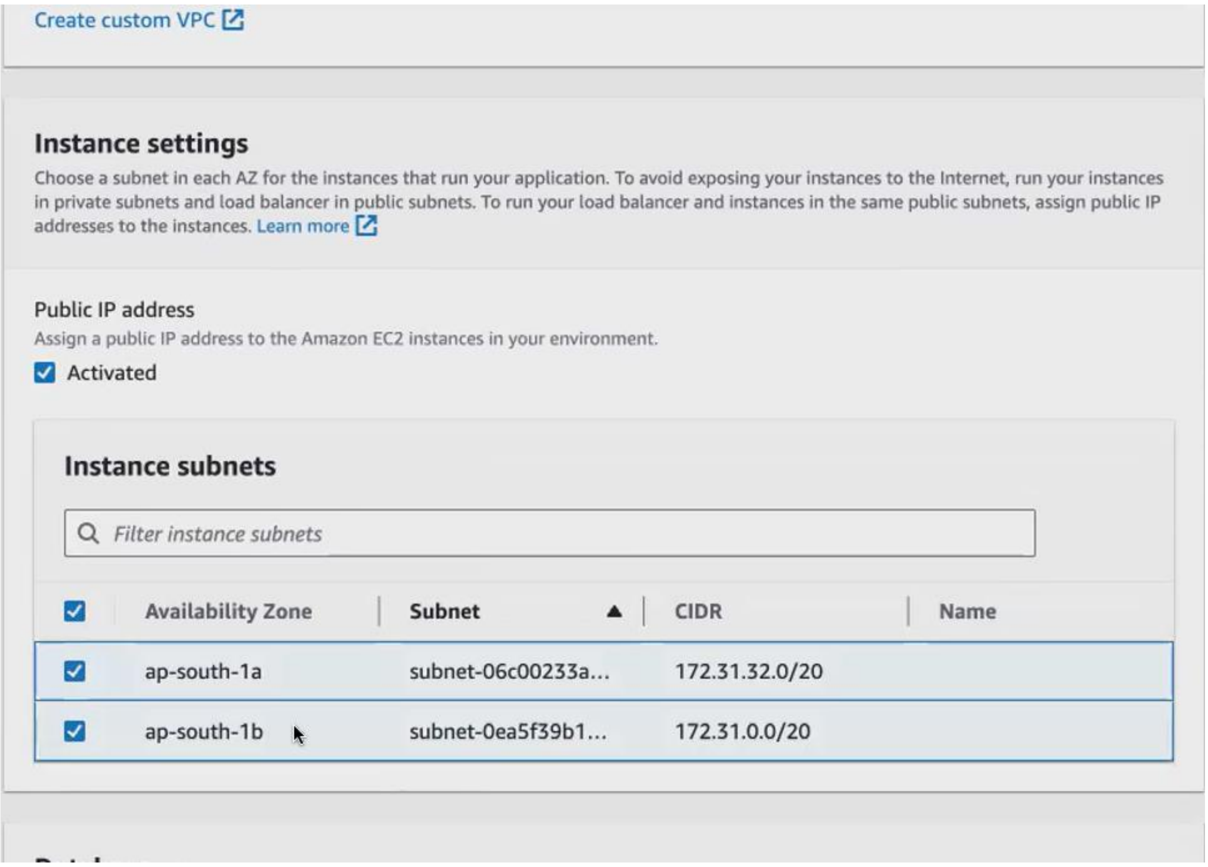
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. Learn more

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

□ Activated

**Instance subnets**



**Instance settings**

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. Learn more

Public IP address

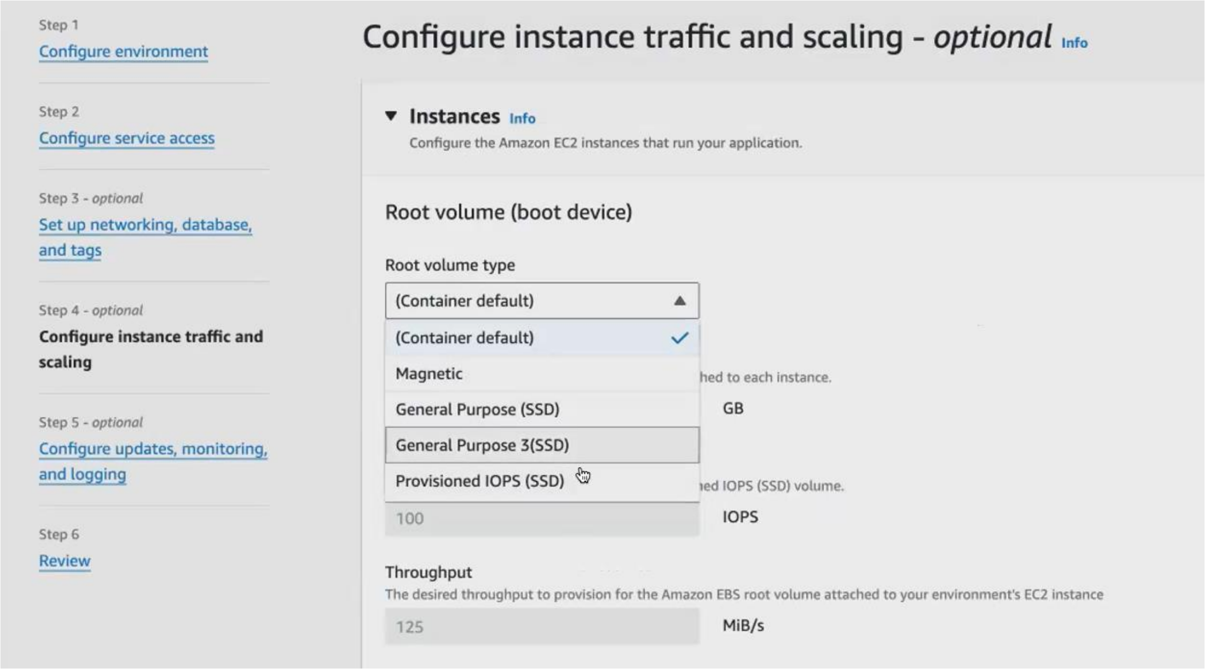
Assign a public IP address to the Amazon EC2 instances in your environment.

☑ Activated

**Instance subnets**

| ☑ | Availability Zone | Subnet ▲ | CIDR | Name |
|---|---|---|---|---|
| ☑ | ap-south-1a | subnet-06c00233a... | 172.31.32.0/20 | |
| ☑ | ap-south-1b | subnet-0ea5f39b1... | 172.31.0.0/20 | |

### Select default volume



Step 1
Configure environment

Step 2
Configure service access

Step 3 - *optional*
Set up networking, database, and tags

Step 4 - *optional*
**Configure instance traffic and scaling**

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Configure instance traffic and scaling - *optional* Info

**Instances** Info
Configure the Amazon EC2 instances that run your application.

Root volume (boot device)

Root volume type

(Container default) ▲

- (Container default) ✓
- Magnetic
- General Purpose (SSD)
- General Purpose 3(SSD)
- Provisioned IOPS (SSD)

hed to each instance.

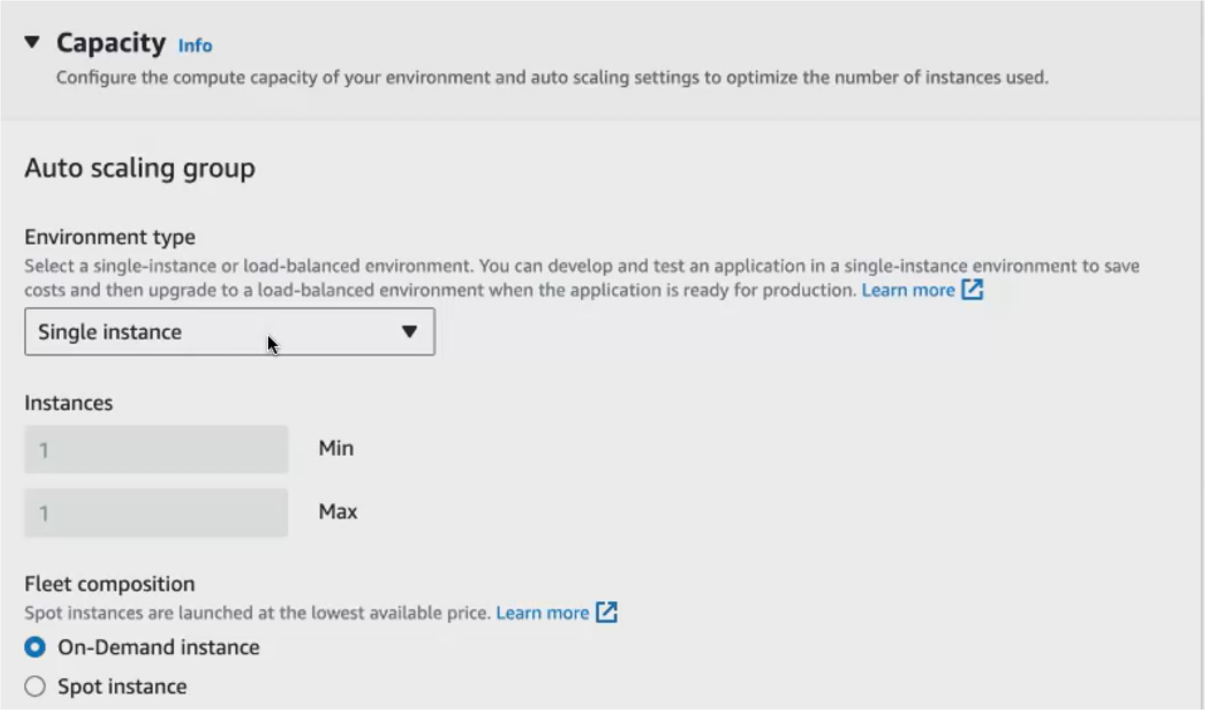
GB

ıed IOPS (SSD) volume.

100 IOPS

Throughput
The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

125 MiB/s



**Capacity** Info
Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Environment type
Select a single-instance or load-balanced environment. You can develop and test an application in a single-instance environment to save costs and then upgrade to a load-balanced environment when the application is ready for production. Learn more

Single instance ▼

Instances

1 Min

1 Max

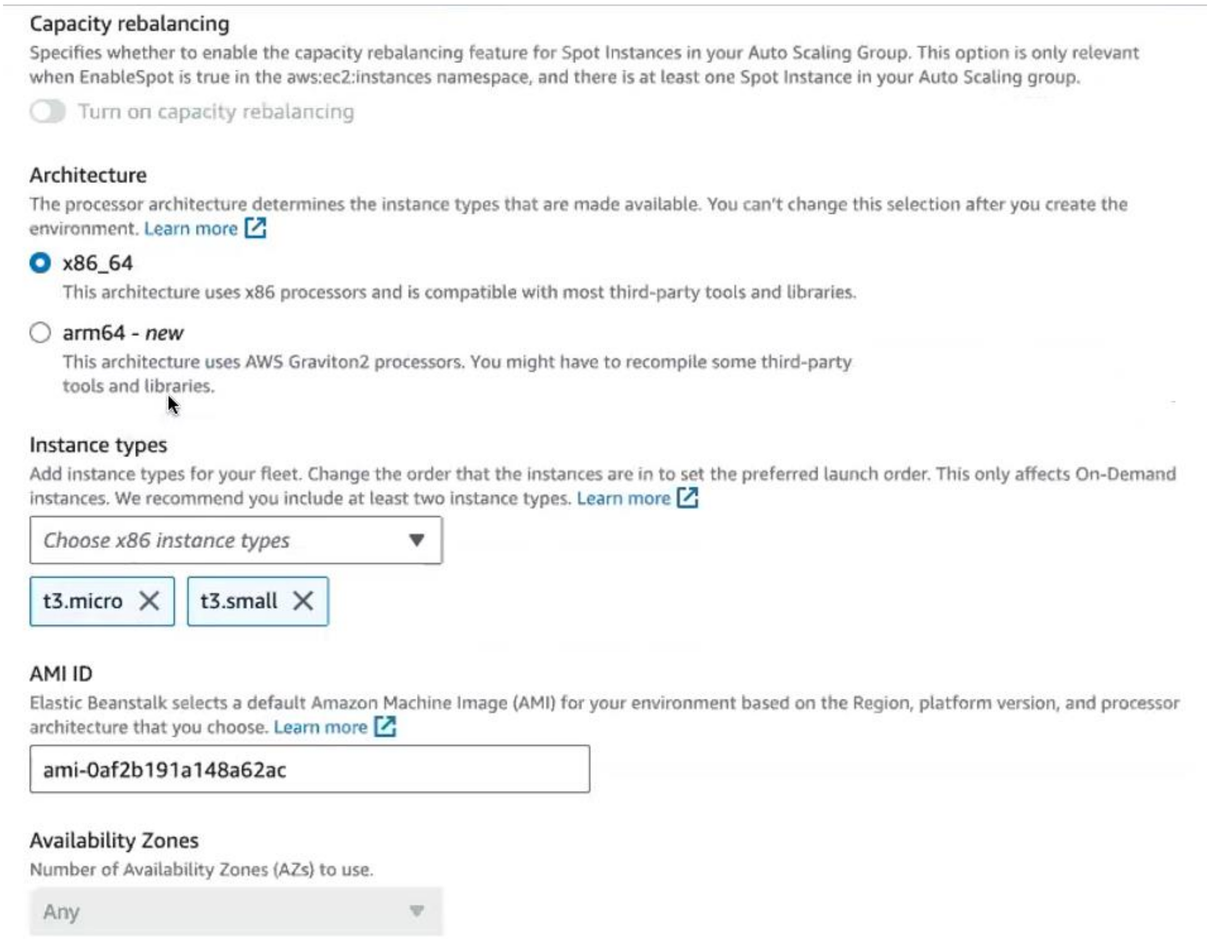
Fleet composition
Spot instances are launched at the lowest available price. Learn more

● On-Demand instance

○ Spot instance

Your Elastic building stock will create an Ec 2 for you. So it will create in this architecture so it can either launch a t 3 micro or a t 3 small. Any one instance type will be launched with this ami id

Click next



Capacity rebalancing

Specifies whether to enable the capacity rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when EnableSpot is true in the aws:ec2:instances namespace, and there is at least one Spot Instance in your Auto Scaling group.

○ Turn on capacity rebalancing

Architecture

The processor architecture determines the instance types that are made available. You can't change this selection after you create the environment. Learn more

● x86_64

This architecture uses x86 processors and is compatible with most third-party tools and libraries.

○ arm64 - *new*

This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

Instance types

Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. Learn more

*Choose x86 instance types* ▼

t3.micro ✕ t3.small ✕

AMI ID

Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. Learn more

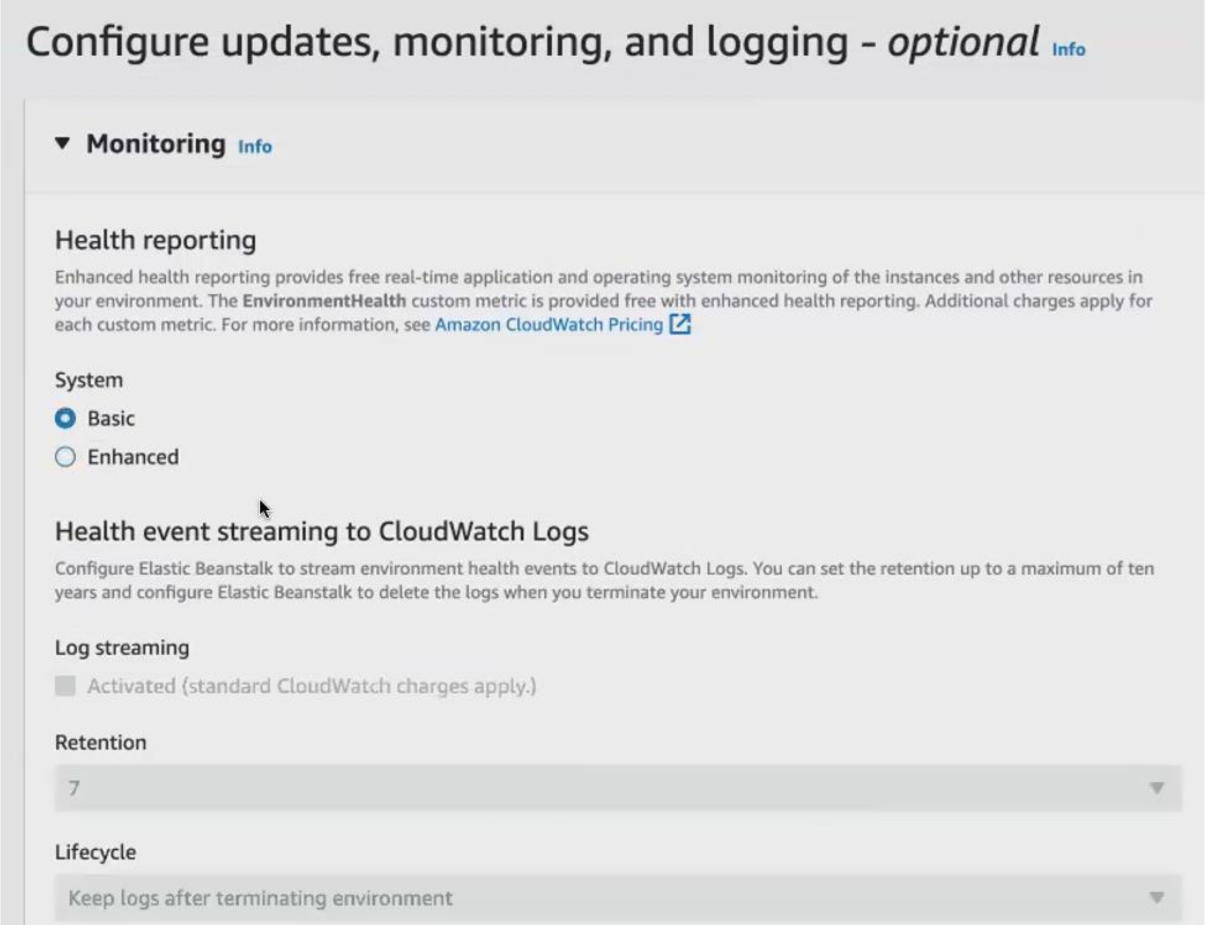
ami-0af2b191a148a62ac

Availability Zones

Number of Availability Zones (AZs) to use.

Any ▼

**I don't want to monitor it. Let me go with basics itself.**



## Configure updates, monitoring, and logging - *optional* Info

▼ **Monitoring** Info

### Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see Amazon CloudWatch Pricing

System

● Basic

○ Enhanced

### Health event streaming to CloudWatch Logs

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming

□ Activated (standard CloudWatch charges apply.)

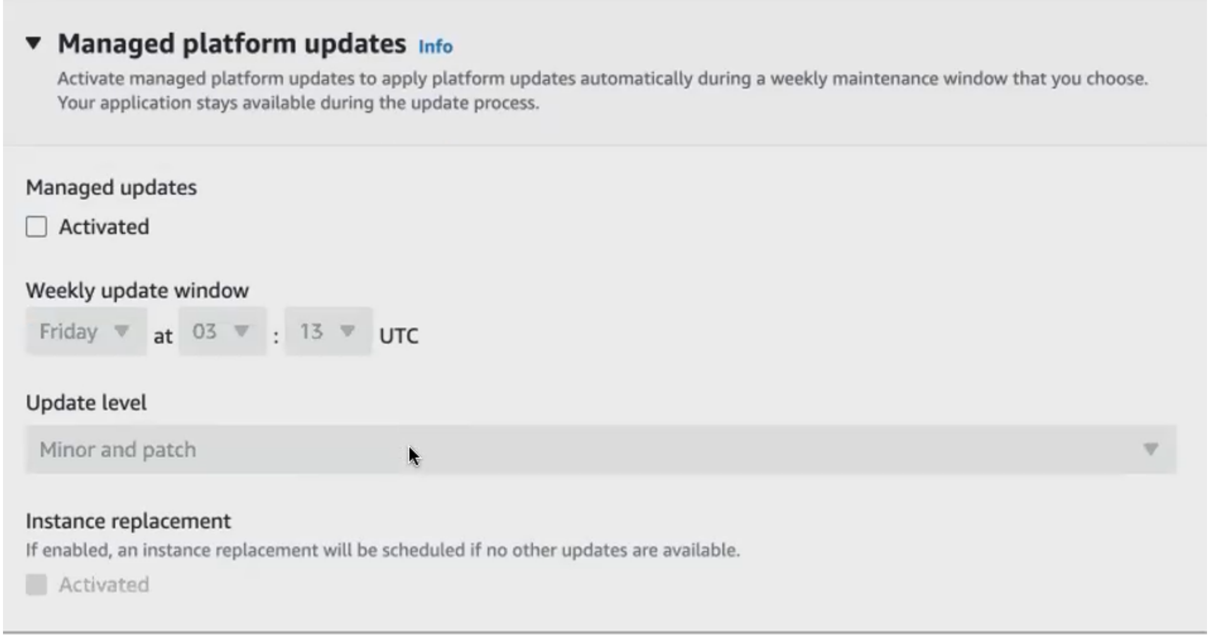
Retention

7

Lifecycle

Keep logs after terminating environment

**Upadate the app give this.**



▼ **Managed platform updates** Info

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates

□ Activated

Weekly update window

Friday at 03 : 13 UTC

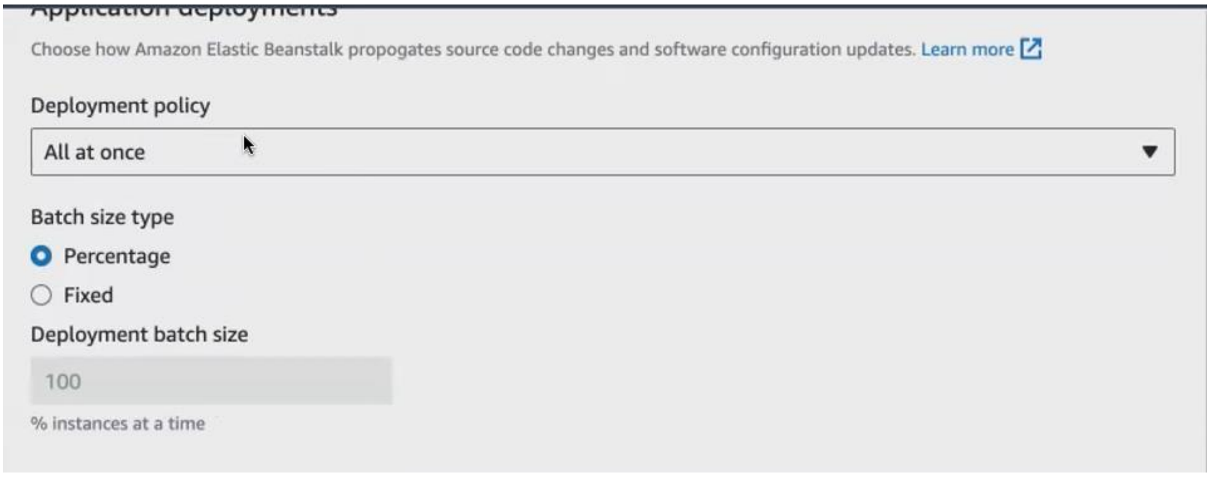
Update level

Minor and patch

Instance replacement

If enabled, an instance replacement will be scheduled if no other updates are available.

□ Activated



Application deployments

Choose how Amazon Elastic Beanstalk propogates source code changes and software configuration updates. Learn more

Deployment policy

All at once ▼

Batch size type

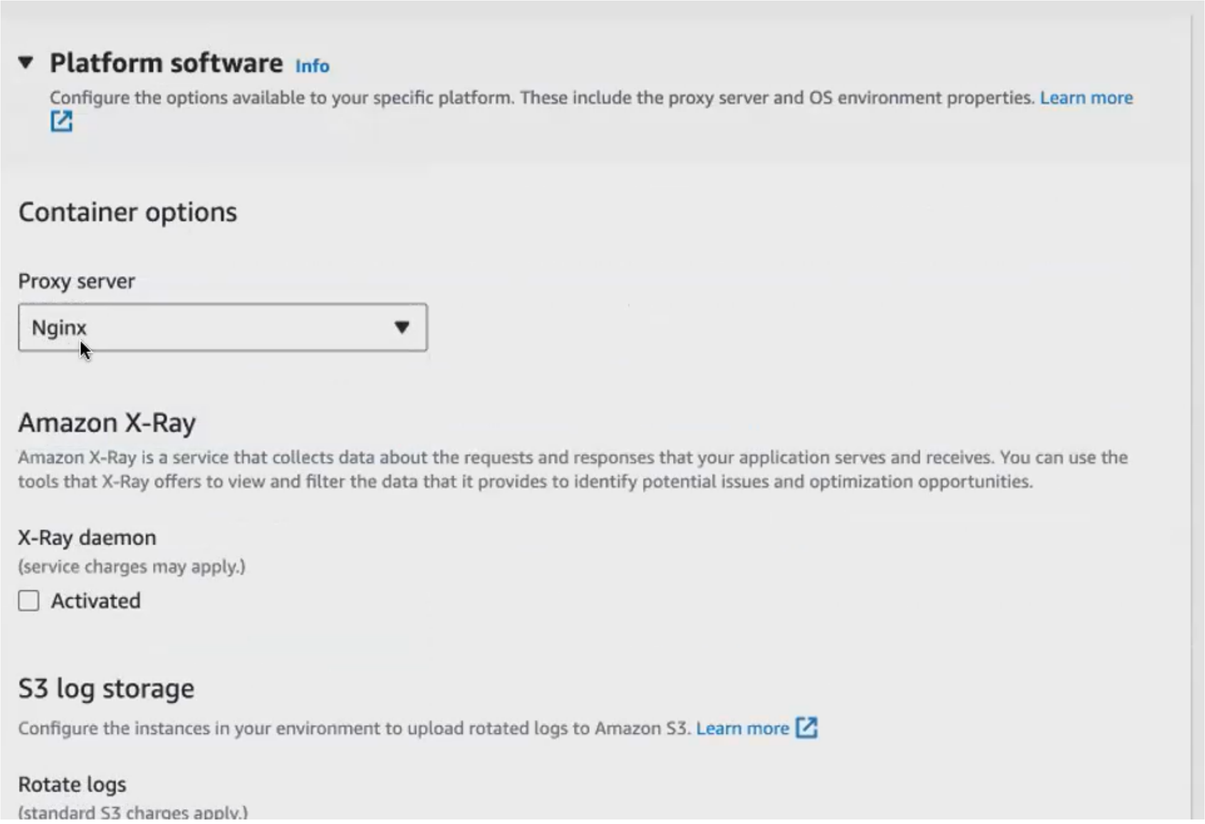
● Percentage

○ Fixed

Deployment batch size

100

% instances at a time



▼ Platform software Info

Configure the options available to your specific platform. These include the proxy server and OS environment properties. Learn more

Container options

Proxy server

Nginx ▼

Amazon X-Ray

Amazon X-Ray is a service that collects data about the requests and responses that your application serves and receives. You can use the tools that X-Ray offers to view and filter the data that it provides to identify potential issues and optimization opportunities.

X-Ray daemon

(service charges may apply.)

□ Activated

S3 log storage

Configure the instances in your environment to upload rotated logs to Amazon S3. Learn more

Rotate logs

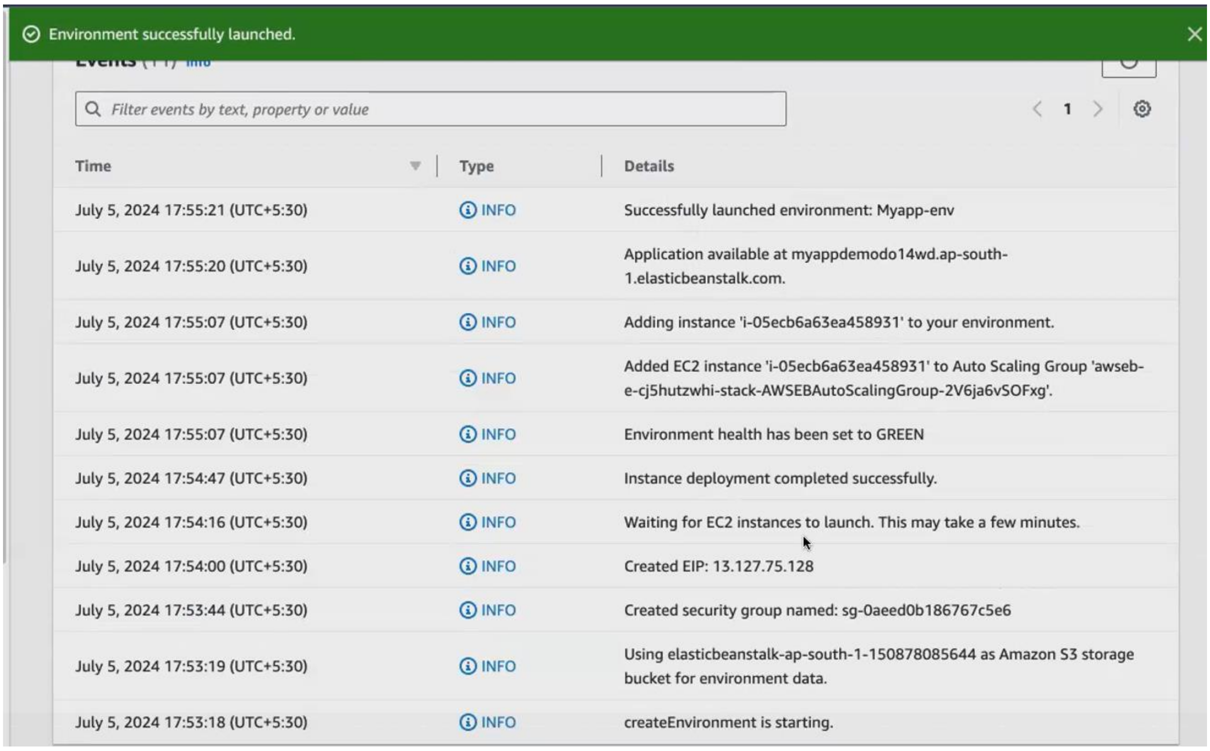
(standard S3 charges apply.)

**I'm using Nodejs application, the proxy server, I think, for the Nodejs application. Your web server is going to be nginx. Okay to access your node. Js application in the browser.**

**Click next.**

## elasticbeanstalk will do

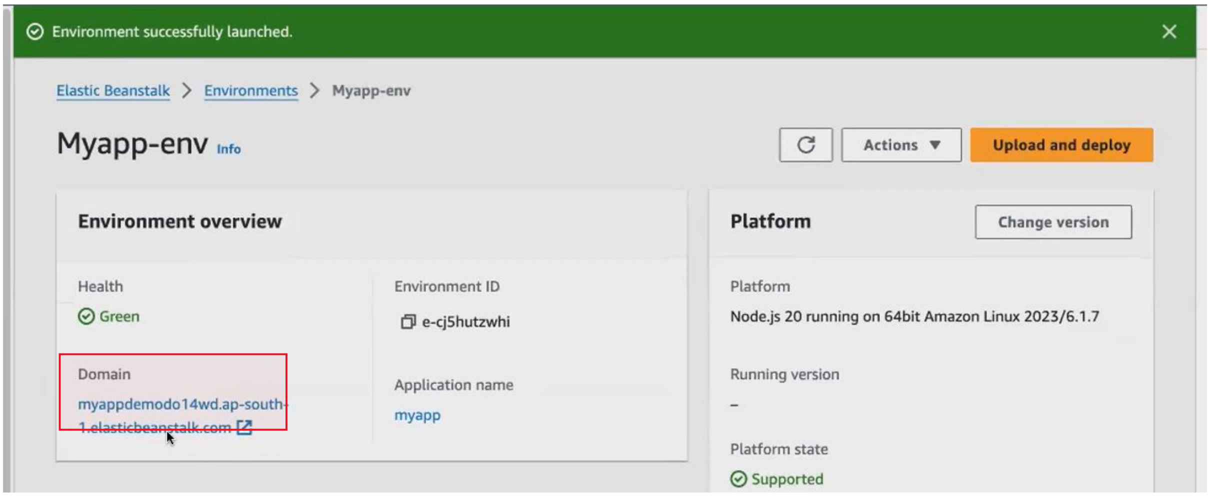
- ✓ It will launch an Ec2
- ✓ inside the Ec2 ,It will launch the platform. That is this platform. It will launch.
- ✓ It will install this platform after install this platform.
- ✓ It will upload your application into the plan into your Ec2.
- ✓ It will create auto scaling. It will create, load balancer. It will create cloud watch  Everything will be launched by your elasticbeanstalk



Environment successfully launched.

| Time | Type | Details |
|---|---|---|
| July 5, 2024 17:55:21 (UTC+5:30) | INFO | Successfully launched environment: Myapp-env |
| July 5, 2024 17:55:20 (UTC+5:30) | INFO | Application available at myappdemodo14wd.ap-south-1.elasticbeanstalk.com. |
| July 5, 2024 17:55:07 (UTC+5:30) | INFO | Adding instance 'i-05ecb6a63ea458931' to your environment. |
| July 5, 2024 17:55:07 (UTC+5:30) | INFO | Added EC2 instance 'i-05ecb6a63ea458931' to Auto Scaling Group 'awseb-e-cj5hutzwhi-stack-AWSEBAutoScalingGroup-2V6ja6vSOFxg'. |
| July 5, 2024 17:55:07 (UTC+5:30) | INFO | Environment health has been set to GREEN |
| July 5, 2024 17:54:47 (UTC+5:30) | INFO | Instance deployment completed successfully. |
| July 5, 2024 17:54:16 (UTC+5:30) | INFO | Waiting for EC2 instances to launch. This may take a few minutes. |
| July 5, 2024 17:54:00 (UTC+5:30) | INFO | Created EIP: 13.127.75.128 |
| July 5, 2024 17:53:44 (UTC+5:30) | INFO | Created security group named: sg-0aeed0b186767c5e6 |
| July 5, 2024 17:53:19 (UTC+5:30) | INFO | Using elasticbeanstalk-ap-south-1-150878085644 as Amazon S3 storage bucket for environment data. |
| July 5, 2024 17:53:18 (UTC+5:30) | INFO | createEnvironment is starting. |

## How to access the application:



Environment successfully launched.

Elastic Beanstalk > Environments > Myapp-env

Myapp-env Info

Environment overview

Health: Green

Environment ID: e-cj5hutzwhi

Domain: myappdemodo14wd.ap-south-1.elasticbeanstalk.com

Application name: myapp

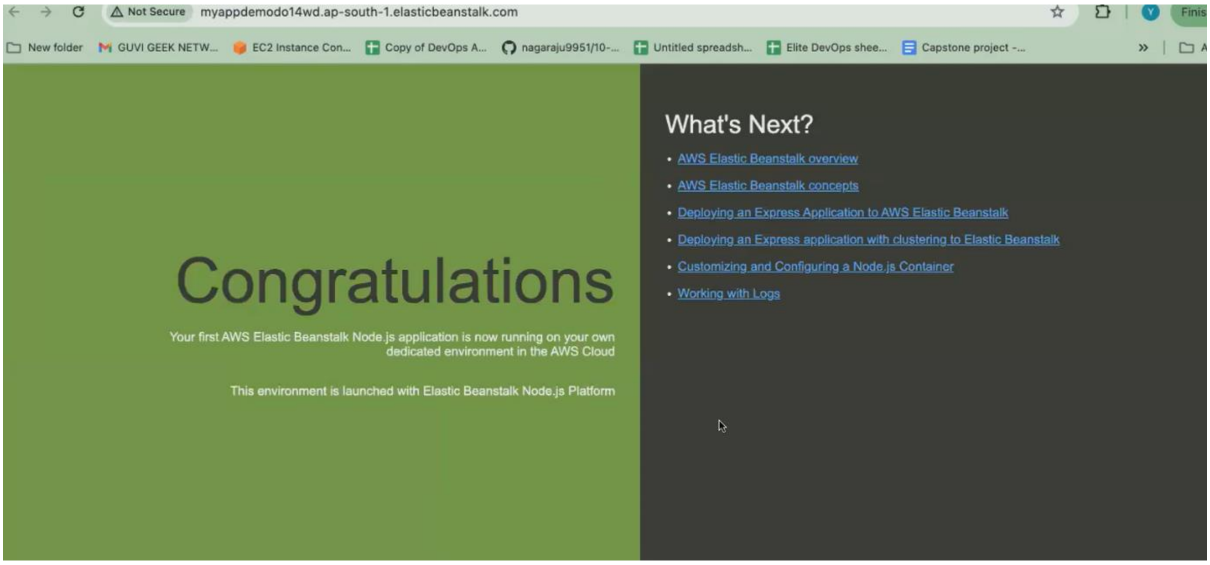
Platform

Platform: Node.js 20 running on 64bit Amazon Linux 2023/6.1.7

Running version: –

Platform state: Supported

Click the domain name  getinto the browser and open



Not Secure myappdemodo14wd.ap-south-1.elasticbeanstalk.com

Congratulations

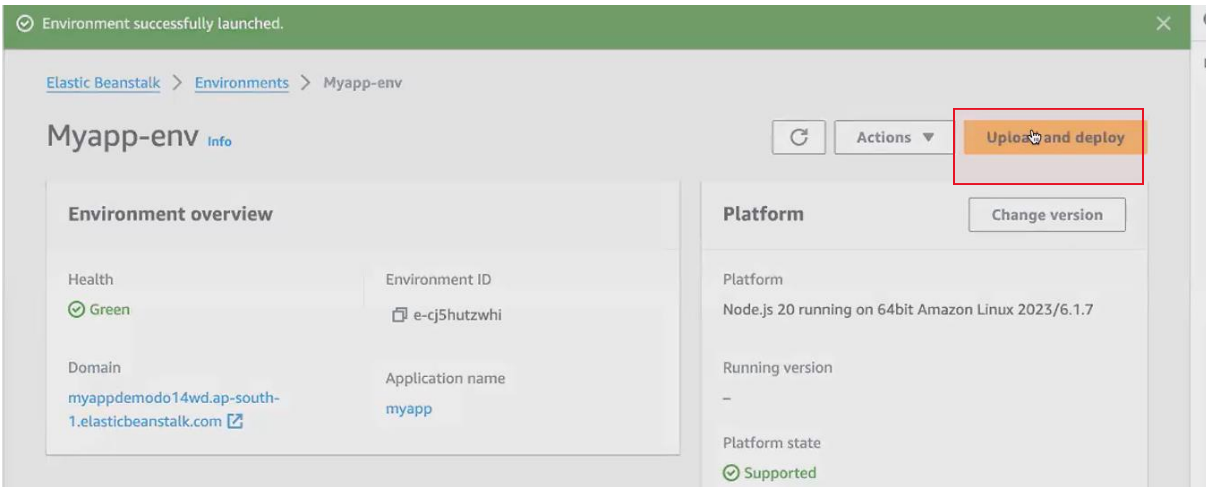
Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploying an Express Application to AWS Elastic Beanstalk
- Deploying an Express application with clustering to Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

You have launched the platform. If any chance, by any chance. If you have the application later, you can also click on upload and deploy. So you can just choose the file, choose our application, and you can deploy, you can click on the deploy. Your application will be deployed in the platform, in the environment.



Environment successfully launched.

Elastic Beanstalk > Environments > Myapp-env

Myapp-env Info

Environment overview

Health: Green

Environment ID: e-cj5hutzwhi

Domain: myappdemodo14wd.ap-south-1.elasticbeanstalk.com

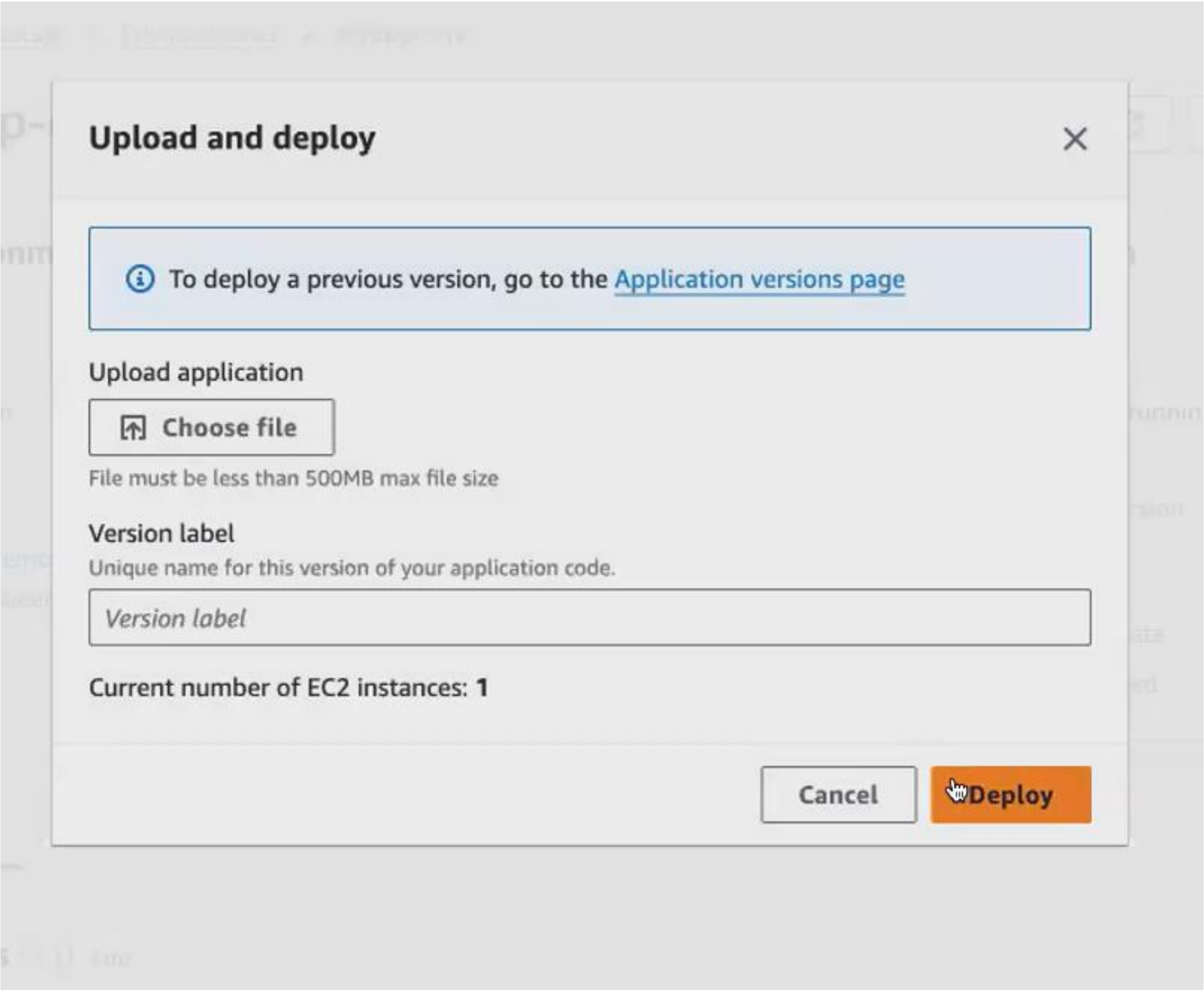
Application name: myapp

Platform

Platform: Node.js 20 running on 64bit Amazon Linux 2023/6.1.7

Running version: –

Platform state: Supported



Upload and deploy

To deploy a previous version, go to the Application versions page

Upload application

File must be less than 500MB max file size

Version label

Unique name for this version of your application code.

Version label

Current number of EC2 instances: 1