Configuration Manual

MSc Research Project
MSc in Cloud Computing

Suraj Beragu
Student ID: x21117951

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	SURAJ BERAGU		
Student ID:	X21117951		
Programme:	MSc in Cloud Computing	**Year:**	2022-23
Module:	Research Project		
Lecturer:	Sean Heeney		
Submission Due Date:	15th December 2022		
Project Title:	Effective use of Cloud Computing and Machine Learning Technologies for Smart Healthcare Applications		
Word Count:	881	**Page Count:**	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Suraj Beragu

Date: 15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Suraj Beragu

Student ID: x21117951

The Configuration Manual is divided into the following sections –

- Configuring the Python-Django Application with ML Model
- Deploying the application on AWS Cloud With CI/CD pipeline

1 Configuring the Python-Django Application

Step 1 – Install the prerequisites onto your local machine

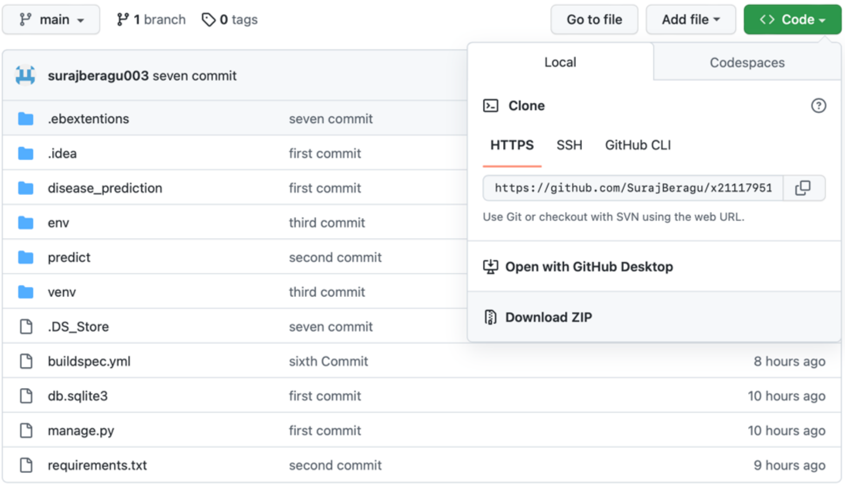
- Python 3.7 or Later
- Pip
- Virtualenv

Step 2 – Create a virtual Python environment and install Django

- Create a virtual environment named env
- Activate the virtual environment.
- Use pip to install Django.

Step 3 – Create a Django Project

- The Django application can be downloaded from the git hub repository



surajberagu003 seven commit

.ebextentions	seven commit	
.idea	first commit	
disease_prediction	first commit	
env	third commit	
predict	second commit	
venv	third commit	
.DS_Store	seven commit	
buildspec.yml	sixth Commit	8 hours ago
db.sqlite3	first commit	10 hours ago
manage.py	first commit	10 hours ago
requirements.txt	second commit	9 hours ago

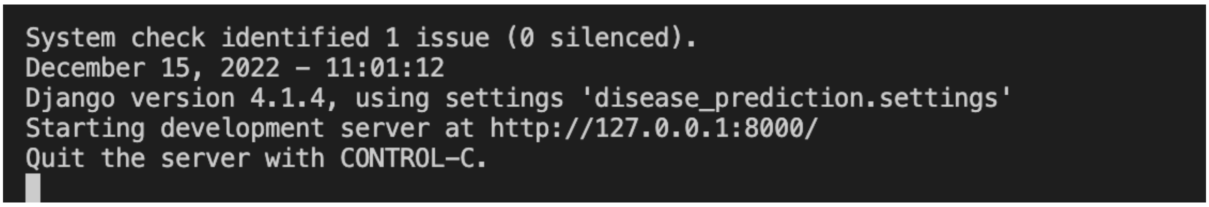
```
https://github.com/SurajBeragu/x21117951
```

Use Git or checkout with SVN using the web URL.

- Activate the virtual environment
- Install requirements.txt with following commands

  $ pip install -r requirements.txt
- Run the Django application on your local machine with the following command

  $ python manage.py runserver 8080



```
System check identified 1 issue (0 silenced).
December 15, 2022 - 11:01:12
Django version 4.1.4, using settings 'disease_prediction.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

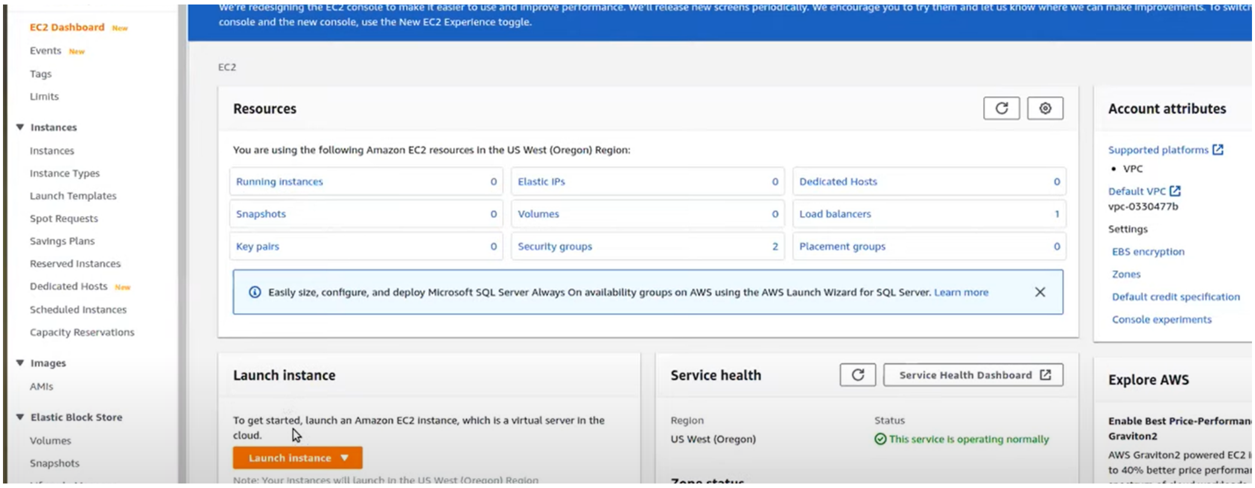
- Examine the server log to view the response to your request. Press Ctrl+C to shutdown the web server and return to your virtual environment.

- Once the application is running successfully stop the server and create a new file called .ebextentions and place the following contents in the file

```
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: myproject.wsgi:application
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /static: static
```

2 Deploying the Django application on AWS Cloud

- Go to ec2 instance and launch.



We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch console and the new console, use the New EC2 Experience toggle.

You are using the following Amazon EC2 resources in the US West (Oregon) Region:

Running instances	0	Elastic IPs	0	Dedicated Hosts	0
Snapshots	0	Volumes	0	Load balancers	1
Key pairs	0	Security groups	2	Placement groups	0

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

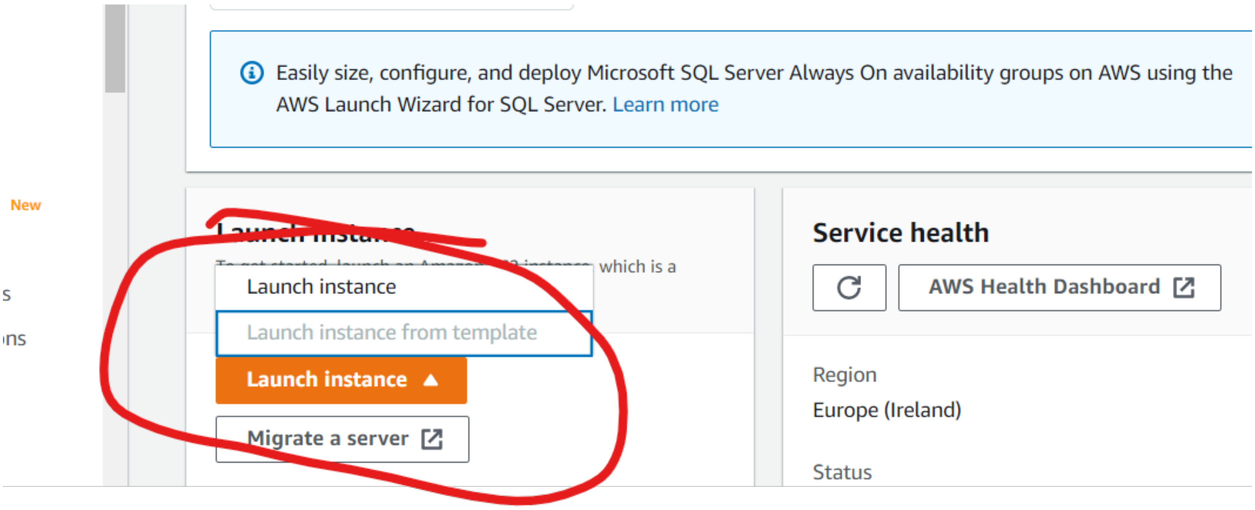
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US West (Oregon) Region

Region: US West (Oregon)

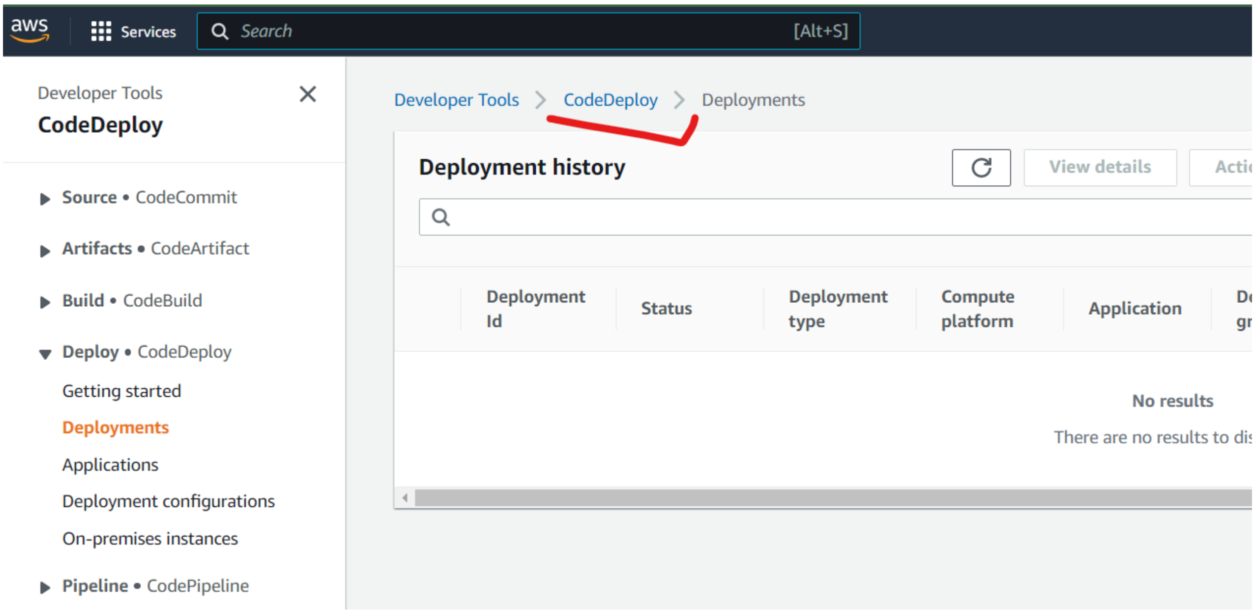
Status: This service is operating normally

Default VPC: vpc-0330477b



Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Region: Europe (Ireland)

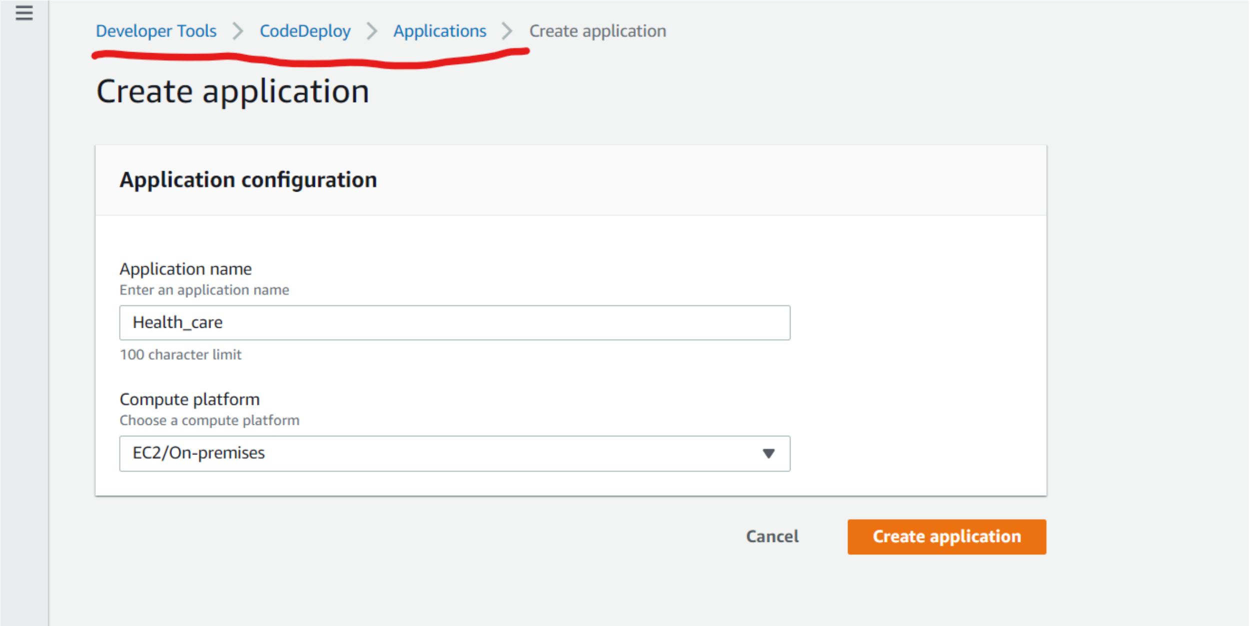


Developer Tools > CodeDeploy > Deployments

Deployment Id	Status	Deployment type	Compute platform	Application	De gr

No results

There are no results to dis



Developer Tools > CodeDeploy > Applications > Create application

Create application

Application configuration

Application name
Enter an application name

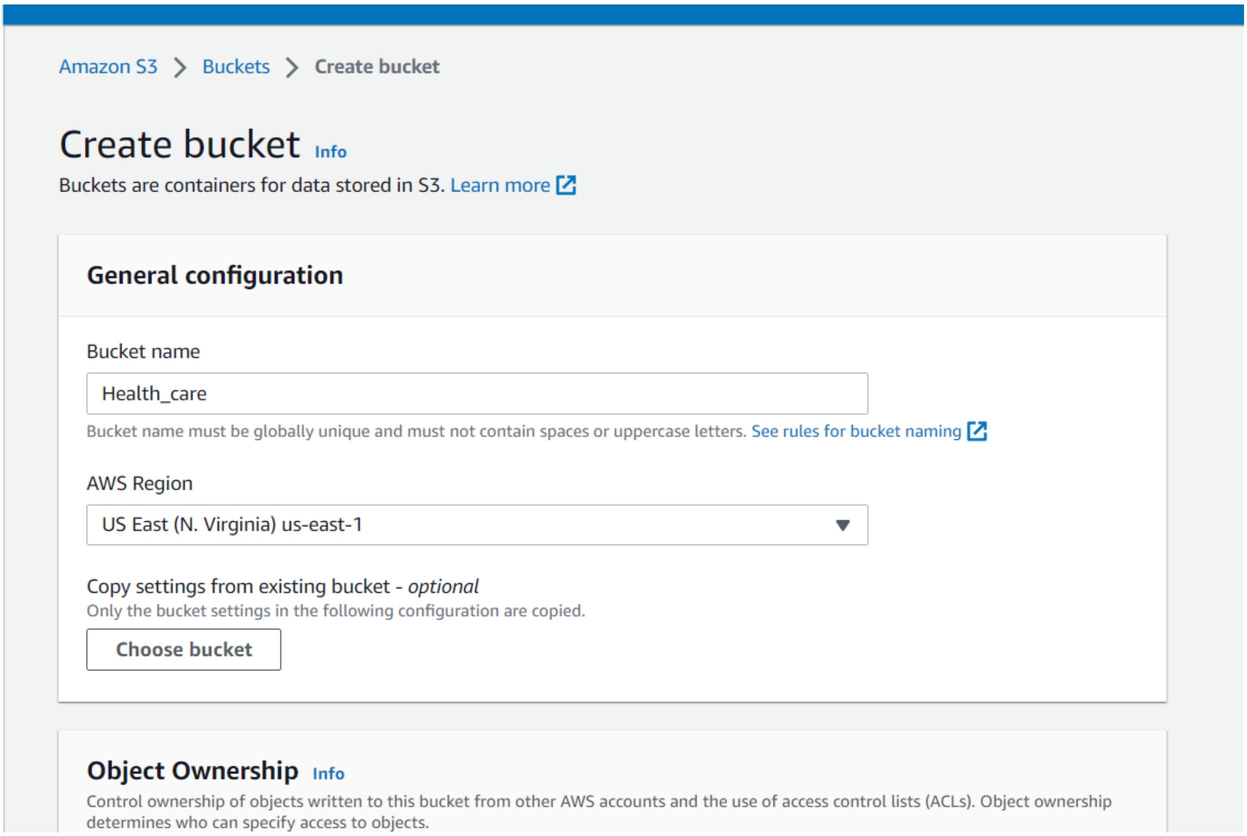
Health_care

100 character limit

Compute platform
Choose a compute platform

EC2/On-premises

- Create a s3 bucket



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

Health_care

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

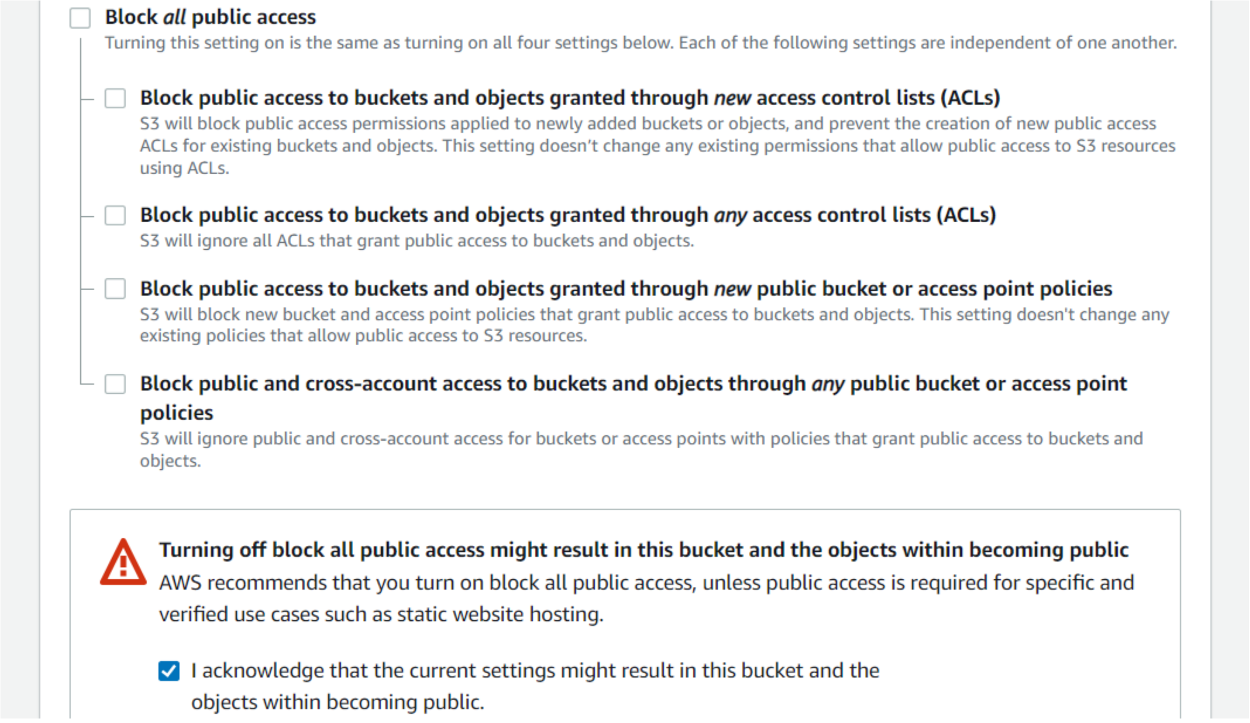
AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.



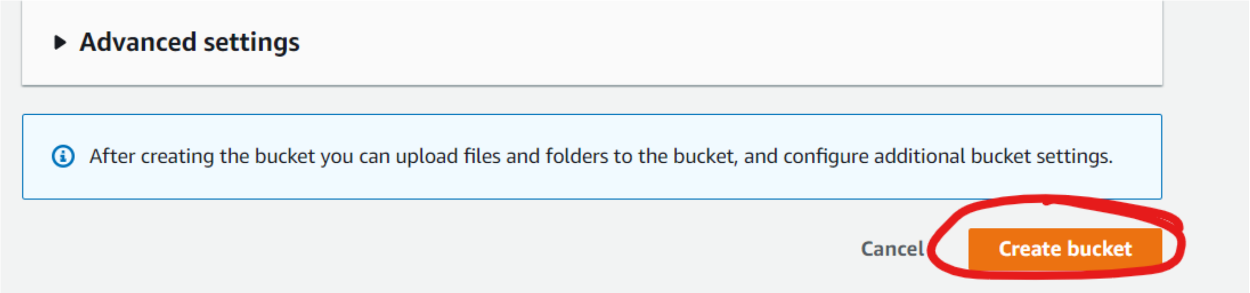
□ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ **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.

⚠ **Turning off block all public access might result in this bucket and the objects within becoming public**
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☑ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

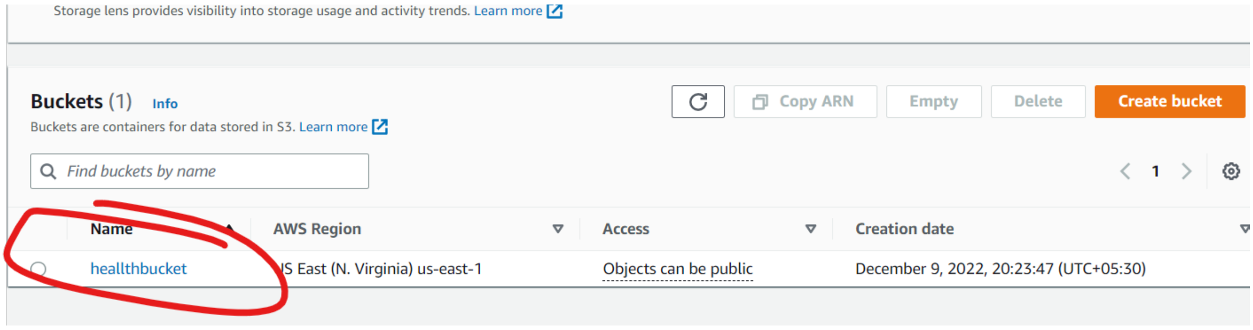
- Then create bucket button



▸ **Advanced settings**

ⓘ After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Bucket is created:

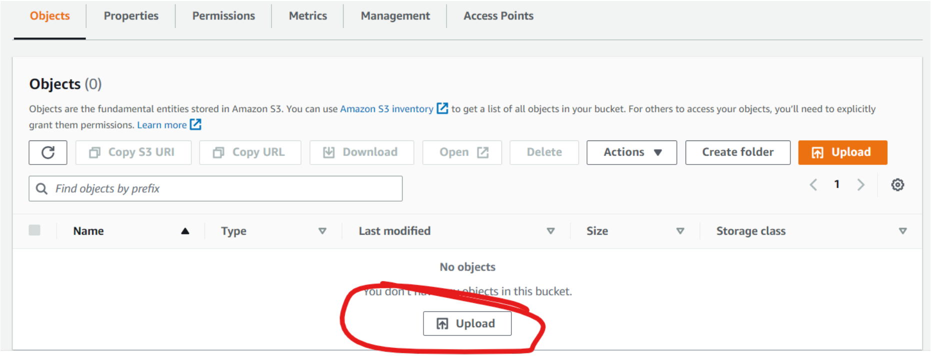


Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info
Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	healthbucket	JS East (N. Virginia) us-east-1	Objects can be public	December 9, 2022, 20:23:47 (UTC+05:30)

Click to upload files



Objects (0)

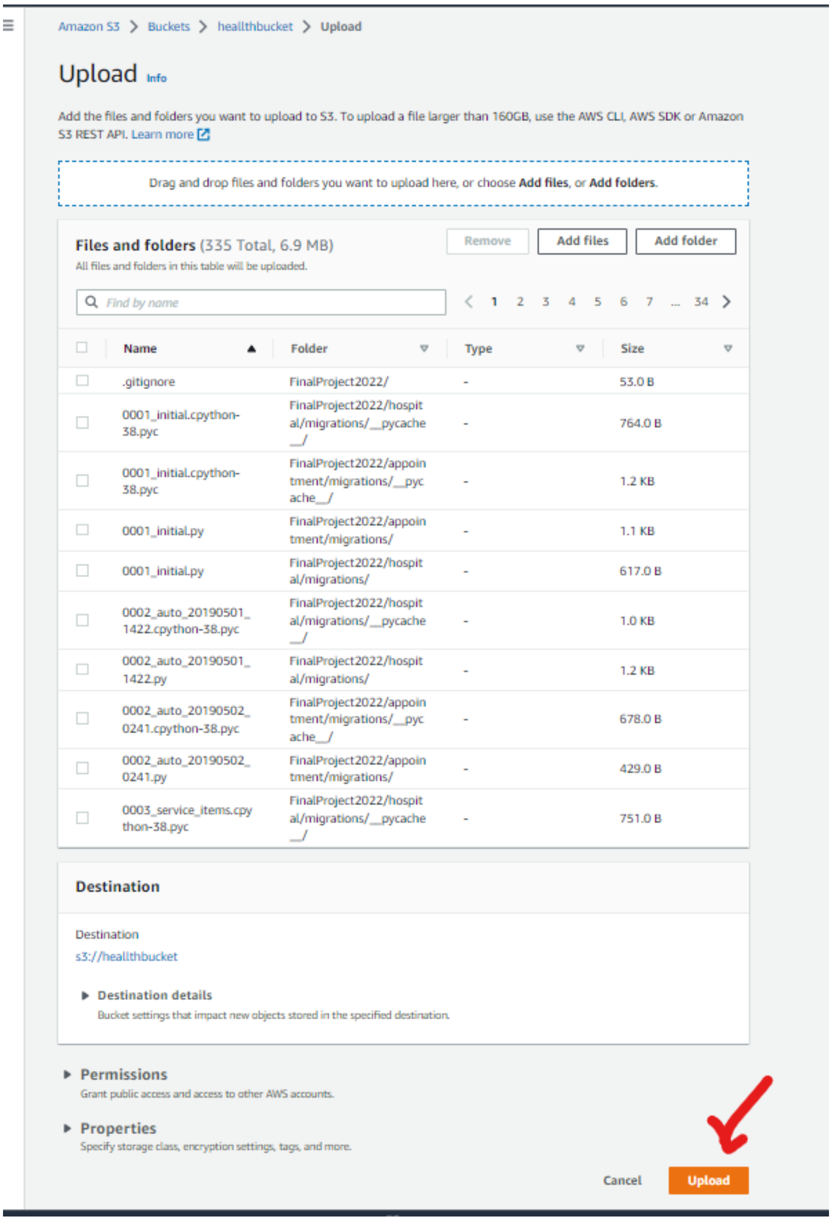
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

- Now the file are ready to upload



Upload

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (335 Total, 6.9 MB)

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
.gitignore	FinalProject2022/	-	53.0 B
0001_initial.cpython-38.pyc	FinalProject2022/hospital/migrations/__pycache__/	-	764.0 B
0001_initial.cpython-38.pyc	FinalProject2022/appointment/migrations/__pycache__/	-	1.2 KB
0001_initial.py	FinalProject2022/appointment/migrations/	-	1.1 KB
0001_initial.py	FinalProject2022/hospital/migrations/	-	617.0 B
0002_auto_20190501_1422.cpython-38.pyc	FinalProject2022/hospital/migrations/__pycache__/	-	1.0 KB
0002_auto_20190501_1422.py	FinalProject2022/hospital/migrations/	-	1.2 KB
0002_auto_20190502_0241.cpython-38.pyc	FinalProject2022/appointment/migrations/__pycache__/	-	678.0 B
0002_auto_20190502_0241.py	FinalProject2022/appointment/migrations/	-	429.0 B
0003_service_items.cpython-38.pyc	FinalProject2022/hospital/migrations/__pycache__/	-	751.0 B

Destination

Destination

s3://heallthbucket

▶ Destination details

Bucket settings that impact new objects stored in the specified destination.

▶ Permissions

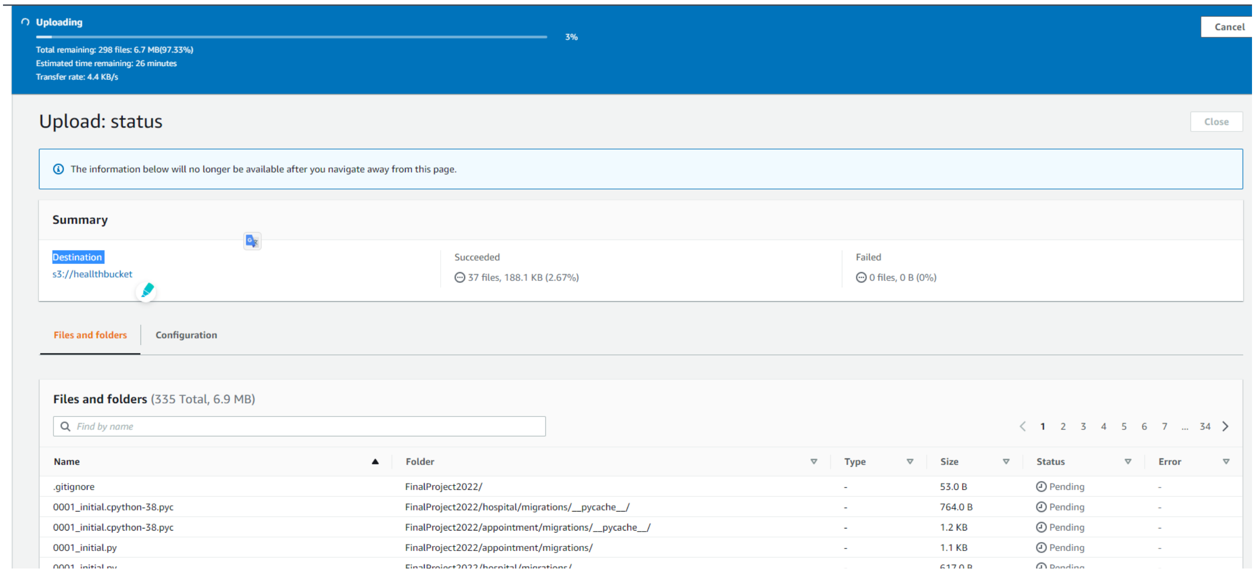
Grant public access and access to other AWS accounts.

▶ Properties

Specify storage class, encryption settings, tags, and more.

Destination

s3://heallthbucket



Uploading

3%

Total remaining: 298 files: 6.7 MB(97.33%)
Estimated time remaining: 26 minutes
Transfer rate: 4.4 KB/s

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination
s3://healthbucket

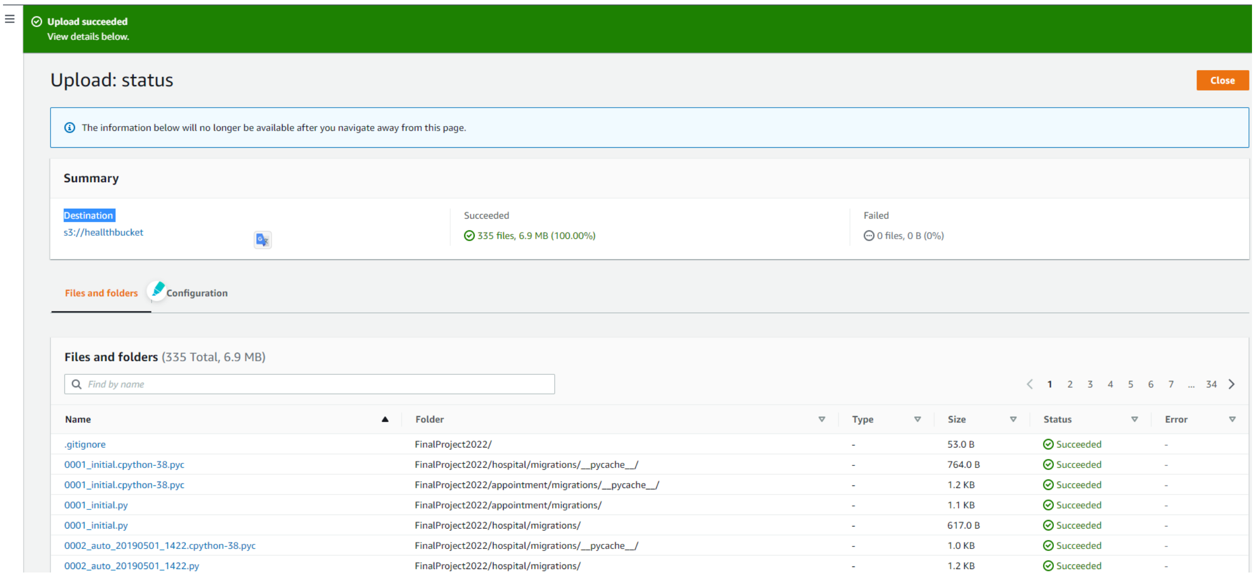
Succeeded
37 files, 188.1 KB (2.67%)

Failed
0 files, 0 B (0%)

Files and folders (335 Total, 6.9 MB)

Name	Folder	Type	Size	Status	Error
.gitignore	FinalProject2022/	-	53.0 B	Pending	-
0001_initial.cpython-38.pyc	FinalProject2022/hospital/migrations/__pycache__/	-	764.0 B	Pending	-
0001_initial.cpython-38.pyc	FinalProject2022/appointment/migrations/__pycache__/	-	1.2 KB	Pending	-
0001_initial.py	FinalProject2022/appointment/migrations/	-	1.1 KB	Pending	-
0001_initial.py	FinalProject2022/hospital/migrations/	-	617.0 B	Pending	-

- File is successfully upload



Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination
s3://healthbucket

Succeeded
335 files, 6.9 MB (100.00%)

Failed
0 files, 0 B (0%)

Files and folders (335 Total, 6.9 MB)

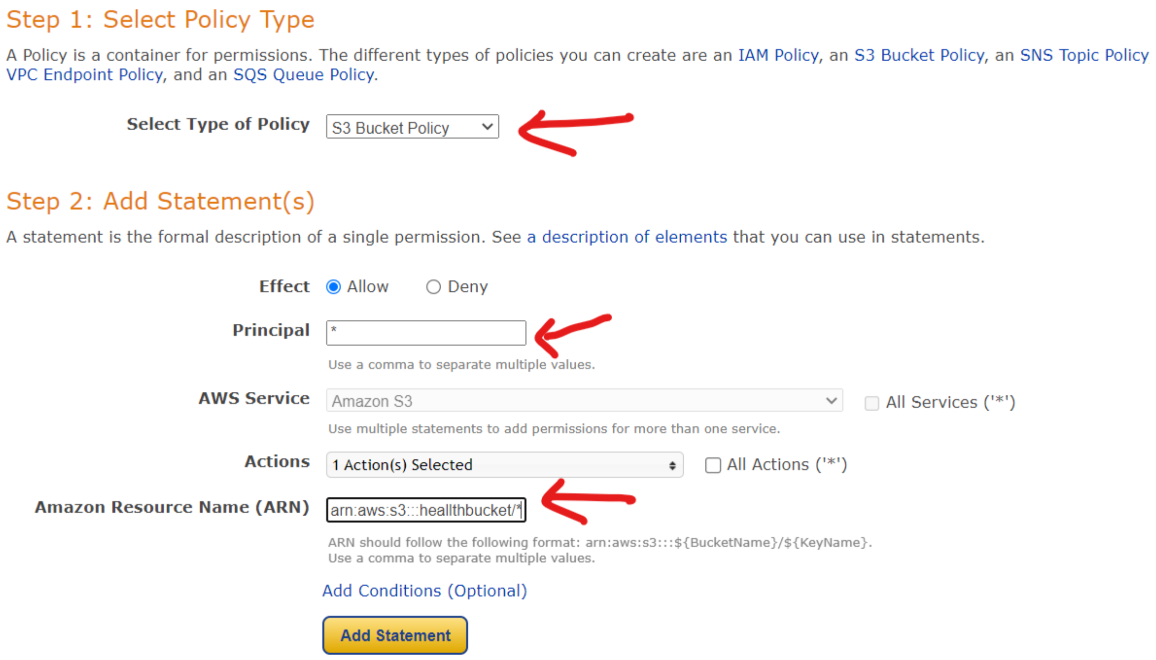
Name	Folder	Type	Size	Status	Error
.gitignore	FinalProject2022/	-	53.0 B	Succeeded	-
0001_initial.cpython-38.pyc	FinalProject2022/hospital/migrations/__pycache__/	-	764.0 B	Succeeded	-
0001_initial.cpython-38.pyc	FinalProject2022/appointment/migrations/__pycache__/	-	1.2 KB	Succeeded	-
0001_initial.py	FinalProject2022/appointment/migrations/	-	1.1 KB	Succeeded	-
0001_initial.py	FinalProject2022/hospital/migrations/	-	617.0 B	Succeeded	-
0002_auto_20190501_1422.cpython-38.pyc	FinalProject2022/hospital/migrations/__pycache__/	-	1.0 KB	Succeeded	-
0002_auto_20190501_1422.py	FinalProject2022/hospital/migrations/	-	1.2 KB	Succeeded	-

- **Copy the ARN Id**

arn:aws:s3::: healthbucket

the click the policy generator

here is the policy setup



Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect ● Allow ○ Deny

Principal *

Use a comma to separate multiple values.

AWS Service Amazon S3 □ All Services ('*')

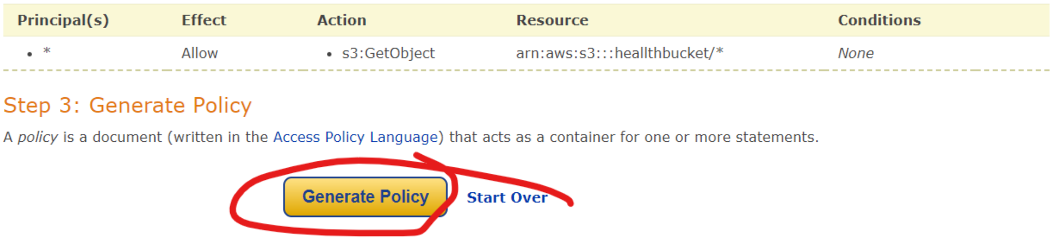
Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected □ All Actions ('*')

Amazon Resource Name (ARN) arn:aws:s3:::heallthbucket/*

ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.

- Then click the generate policy button:

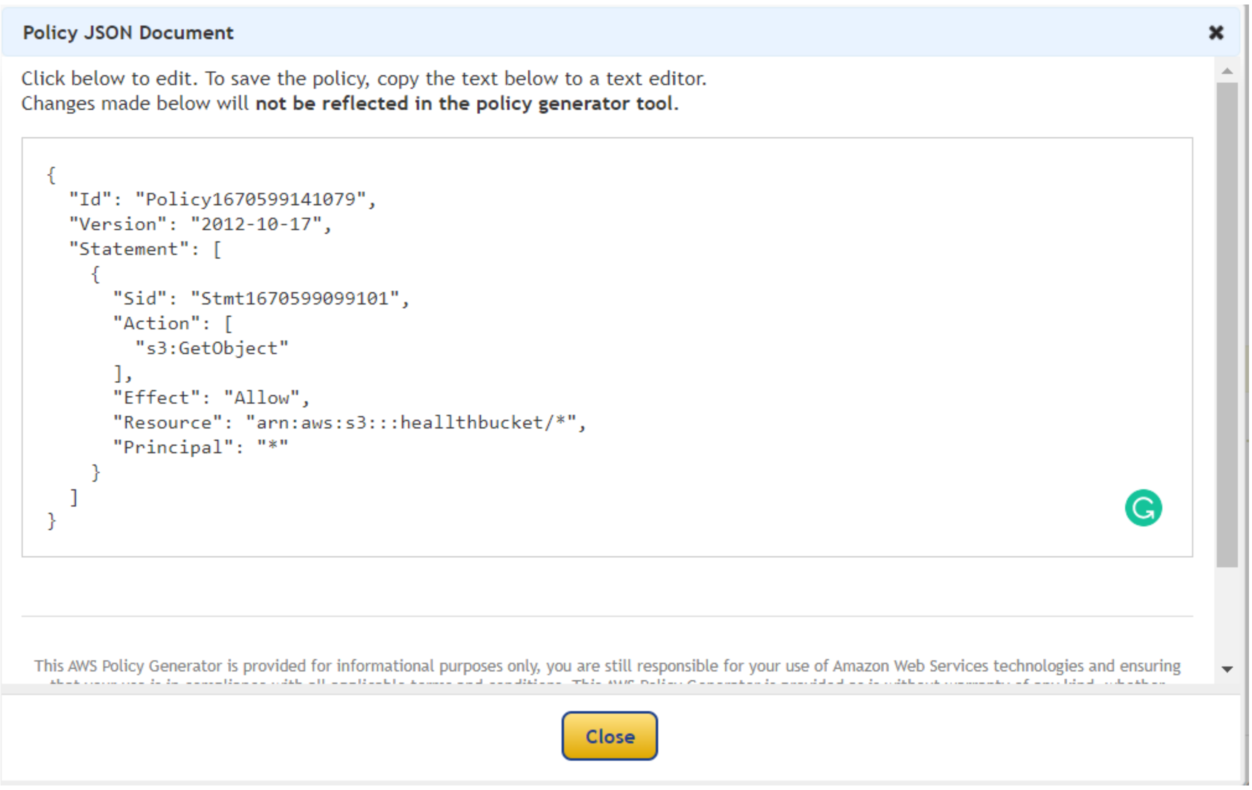


Principal(s)	Effect	Action	Resource	Conditions
• *	Allow	• s3:GetObject	arn:aws:s3:::heallthbucket/*	*None*

Step 3: Generate Policy

A *policy* is a document (written in the Access Policy Language) that acts as a container for one or more statements.

- The policy will generated like below figure:



Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool.**

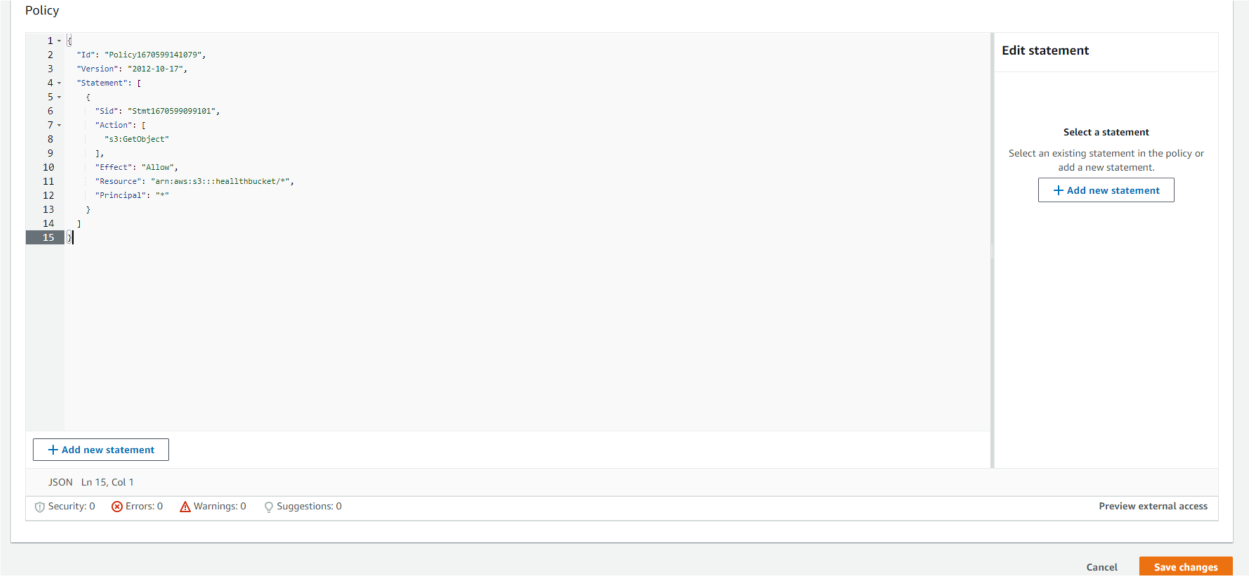
```
{
  "Id": "Policy1670599141079",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1670599099101",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::healthbucket/*",
      "Principal": "*"
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring

- Json format.

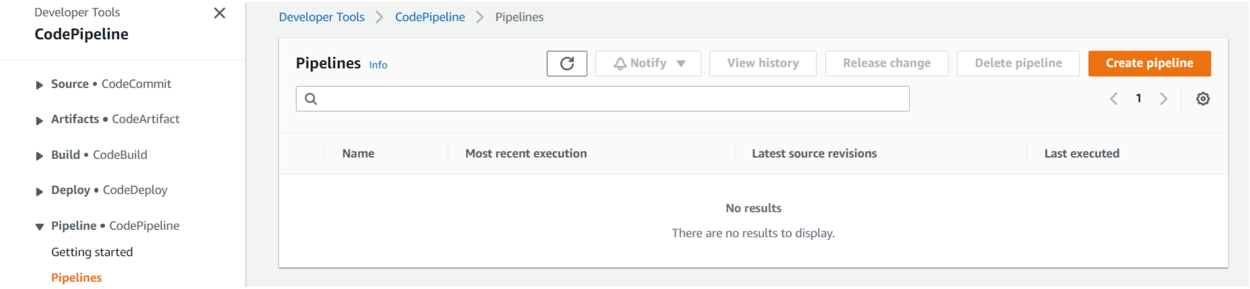
```
{
  "Id": "Policy1670599141079",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1670599099101",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::healthbucket/*",
      "Principal": "*"
    }
  ]
}
```

- Copy the policy script into the bucket policy then click the save button.



```
{
  "Id": "Policy1670599141079",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1670599099101",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::heallthbucket/*",
      "Principal": "*"
    }
  ]
}
```

- Now create a pipeline

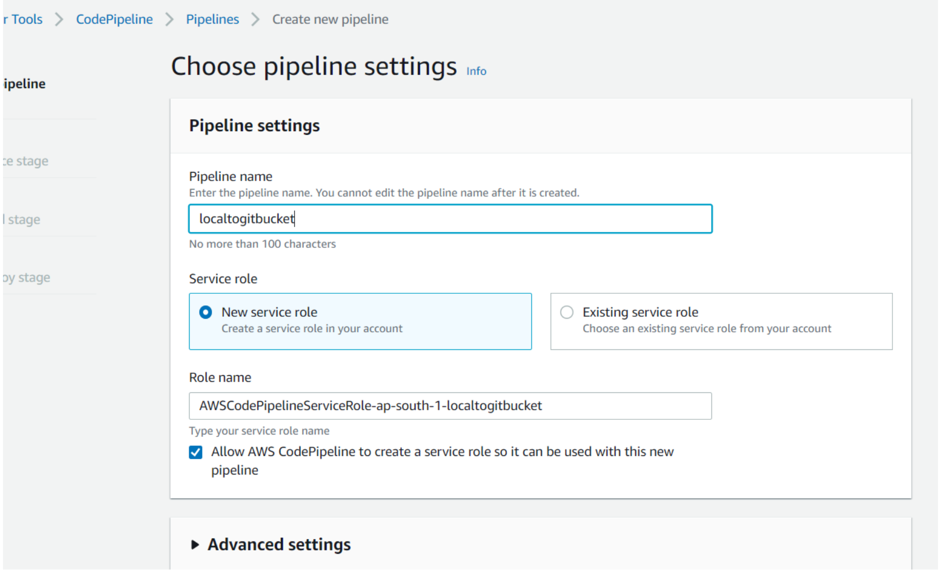


Name	Most recent execution	Latest source revisions	Last executed

No results

There are no results to display.

- Choosing the pipeline name


r Tools > CodePipeline > Pipelines > Create new pipeline
ipeline
e stage
l stage
y stage
Choose pipeline settings Info
Advanced settings

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
localtogitbucket
No more than 100 characters

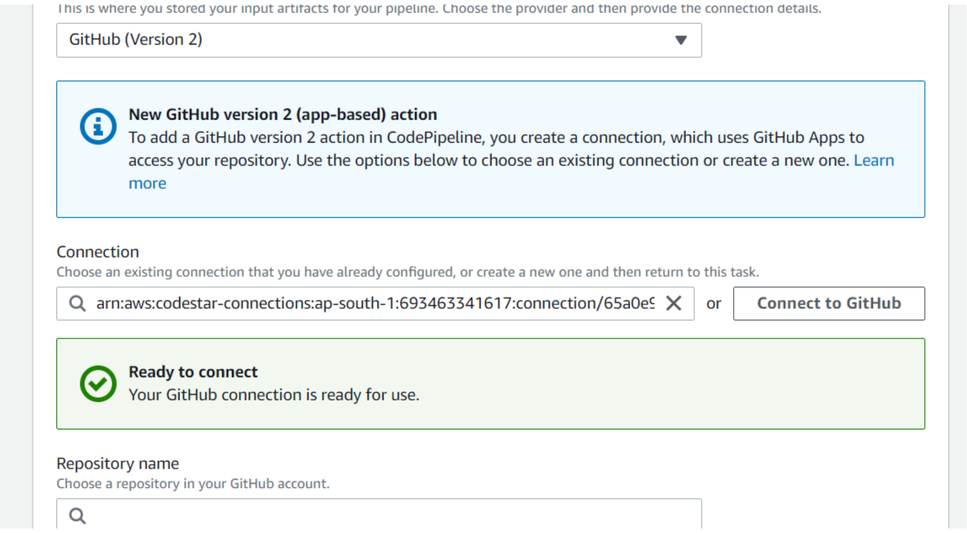
Service role
● New service role
Create a service role in your account
○ Existing service role
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-ap-south-1-localtogitbucket
Type your service role name

☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

- Connect to GitHub

- The GitHub repository is connected now



This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2) ▼

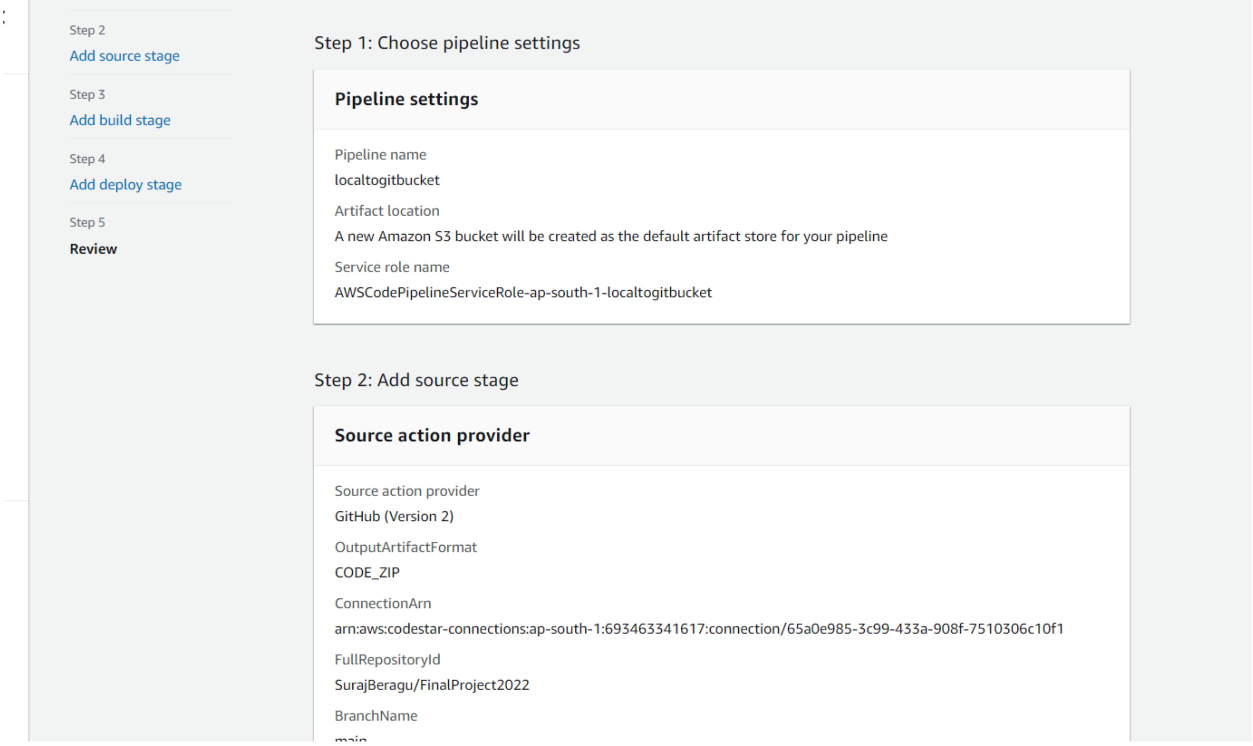
New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. Learn more

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

arn:aws:codestar-connections:ap-south-1:693463341617:connection/65a0e9 ✕ or **Connect to GitHub**

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.



Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
localtogitbucket

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-ap-south-1-localtogitbucket

Step 2: Add source stage

Source action provider

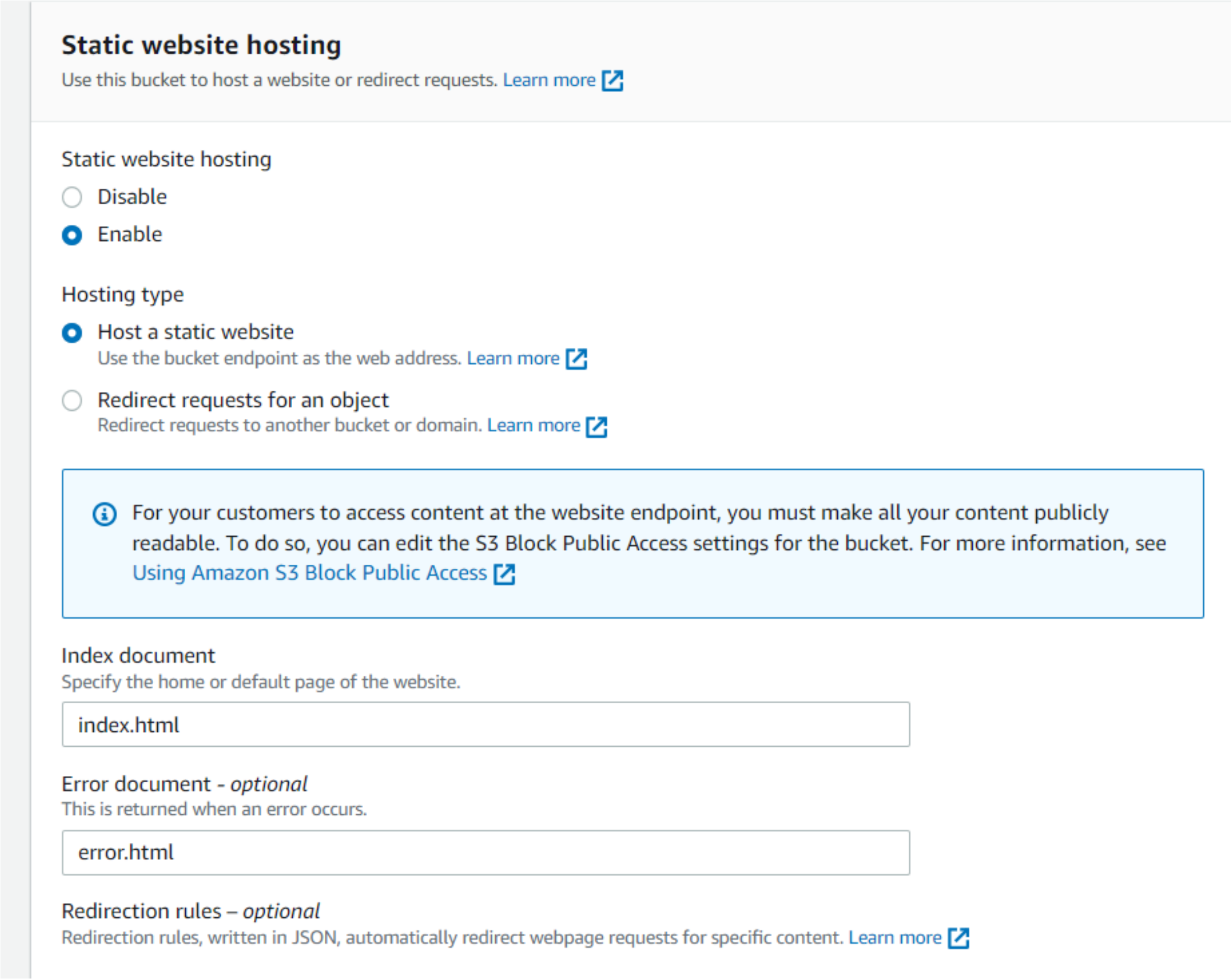
Source action provider
GitHub (Version 2)

OutputArtifactFormat
CODE_ZIP

ConnectionArn
arn:aws:codestar-connections:ap-south-1:693463341617:connection/65a0e985-3c99-433a-908f-7510306c10f1

FullRepositoryId
SurajBeragu/FinalProject2022

BranchName
main



Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting

○ Disable

● Enable

Hosting type

● Host a static website

Use the bucket endpoint as the web address. Learn more

○ Redirect requests for an object

Redirect requests to another bucket or domain. Learn more

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

Index document

Specify the home or default page of the website.

index.html

Error document - *optional*

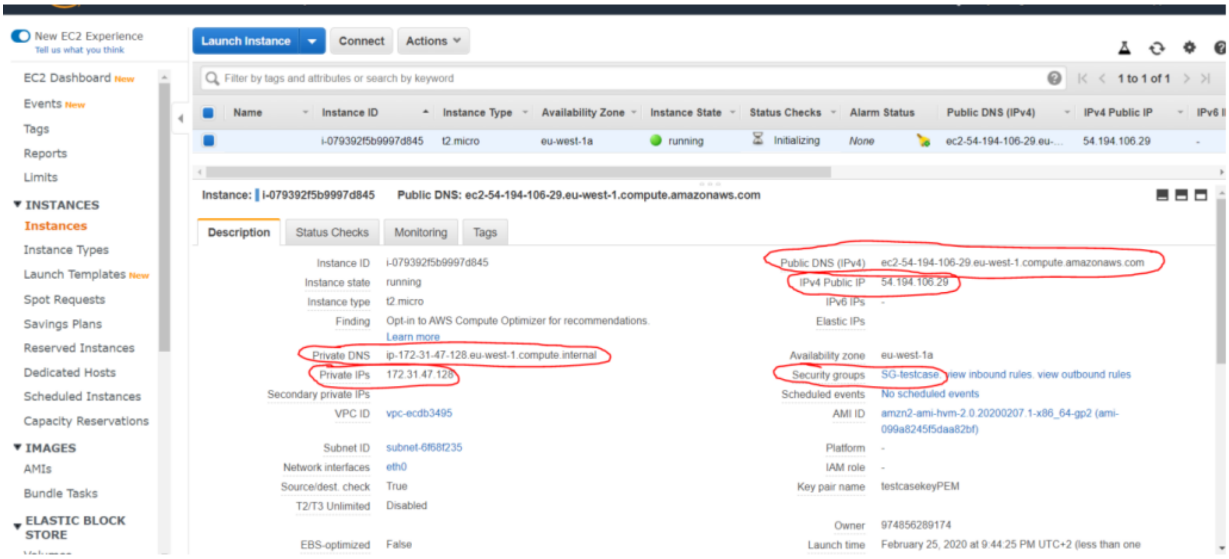
This is returned when an error occurs.

error.html

Redirection rules – *optional*

Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more

- By default, S3 bucket settings do not contain a website hosting option; a bucket home page and error page directory to be included. The most important function of an S3 bucket is to store data, therefore in the future, Jenkins can be used to automate the uploading of data into the bucket. Due to the absence of a Jenkins task, this task had to be completed manually by dragging and dropping files into the upload box. To ensure security, performance, and cost management, it is advised to configure CloudFront Service in tandem with an S3 bucket to distribute and secure information. CloudFront plays a crucial role in delivering data to consumers, encrypting connections with a custom SSL certificate, and protecting against DDoS attacks by default using AWS Shield Standard. Integrating S3 service into its design is vital.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 I
	i-079392f5b9997d845	t2.micro	eu-west-1a	running	Initializing	None	ec2-54-194-106-29.eu-...	54.194.106.29	-

Instance: i-079392f5b9997d845 Public DNS: ec2-54-194-106-29.eu-west-1.compute.amazonaws.com

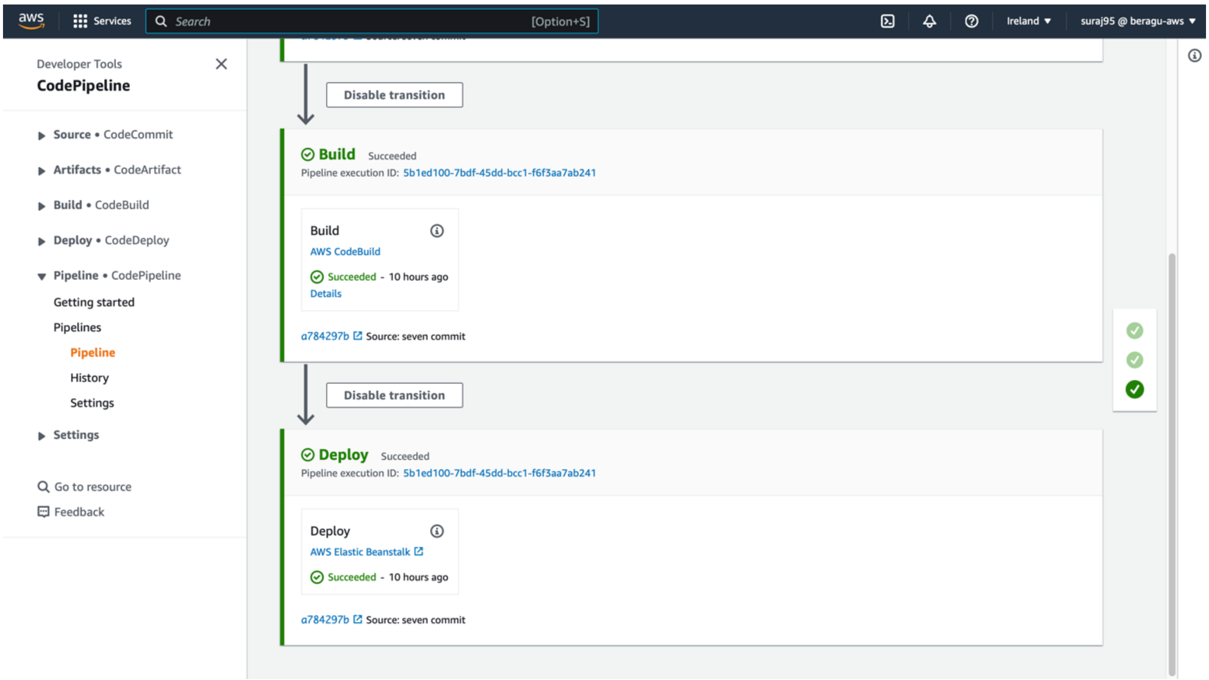
Instance ID	i-079392f5b9997d845	Public DNS (IPv4)	ec2-54-194-106-29.eu-west-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	54.194.106.29
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-172-31-47-128.eu-west-1.compute.internal	Availability zone	eu-west-1a
Private IPs	172.31.47.128	Security groups	SG-testcase. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-ecdb3495	AMI ID	amzn2-ami-hvm-2.0.20200207.1-x86_64-gp2 (ami-099a8245f5daa82bf)
Subnet ID	subnet-6f68f235	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	testcasekeyPEM
T2/T3 Unlimited	Disabled		
		Owner	974856289174
EBS-optimized	False	Launch time	February 25, 2020 at 9:44:25 PM UTC+2 (less than one

To connect to the instance, the best option is using Command Prompt for its speed and ease of control. First, the user should run "sudo yum update" in Command Prompt to apply all AWS updates. Then, they should register to the Microsoft repository and install the necessary packages with the following commands.

- From your local system connect to Git repository with the following commands –

```
echo "# Sample" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/SurajBeragu/Sample.git
git push -u origin main
```

- Once the Code is pushed from the local machine it will reside in the git repository and automatically trigger the Code pipeline in AWS cloud, once the pipeline is successfully completed the application will be deployed on the elastic beanstalk.



✓ Build Succeeded

Pipeline execution ID: 5b1ed100-7bdf-45dd-bcc1-f6f3aa7ab241

Build
AWS CodeBuild
✓ Succeeded - 10 hours ago
Details

a784297b Source: seven commit

✓ Deploy Succeeded

Pipeline execution ID: 5b1ed100-7bdf-45dd-bcc1-f6f3aa7ab241

Deploy
AWS Elastic Beanstalk
✓ Succeeded - 10 hours ago

a784297b Source: seven commit