Don Bosco Institute Of Technology,

(DBIT), Mumbai

Department Of Information Technology

LAB JOURNAL

On

ITL504 : Advance DevOps Lab

By

42 Tushar Padhy

Academic Year : Nov, 2022

INDEX

Sr no	Exp No	Name of the experiment	Date
1.	1	To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration	22/07/2022
2.	2	To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.	31/07/2022
3	5	To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.	20/08/2022
4	6	To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform.	27/09/2022
5	7	To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.	03/09/2022
6	8	Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application	10/09/2022
7	9	To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine	17/09/2022
8	10	To perform Port, Service monitoring,	24/09/2022

		Windows/Linux server monitoring using Nagios	
9	11	To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs	01/10/2022
10	12	To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.	08/10/2022

Name: Tushar Padhy

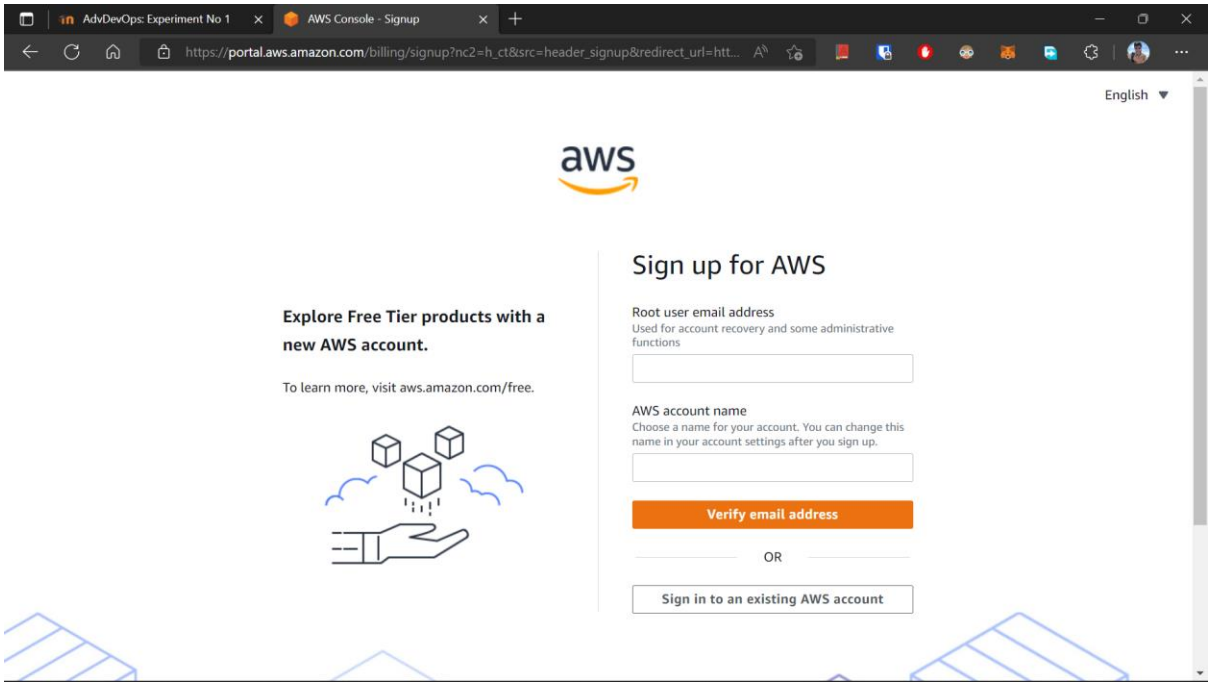
TE-IT SEM V

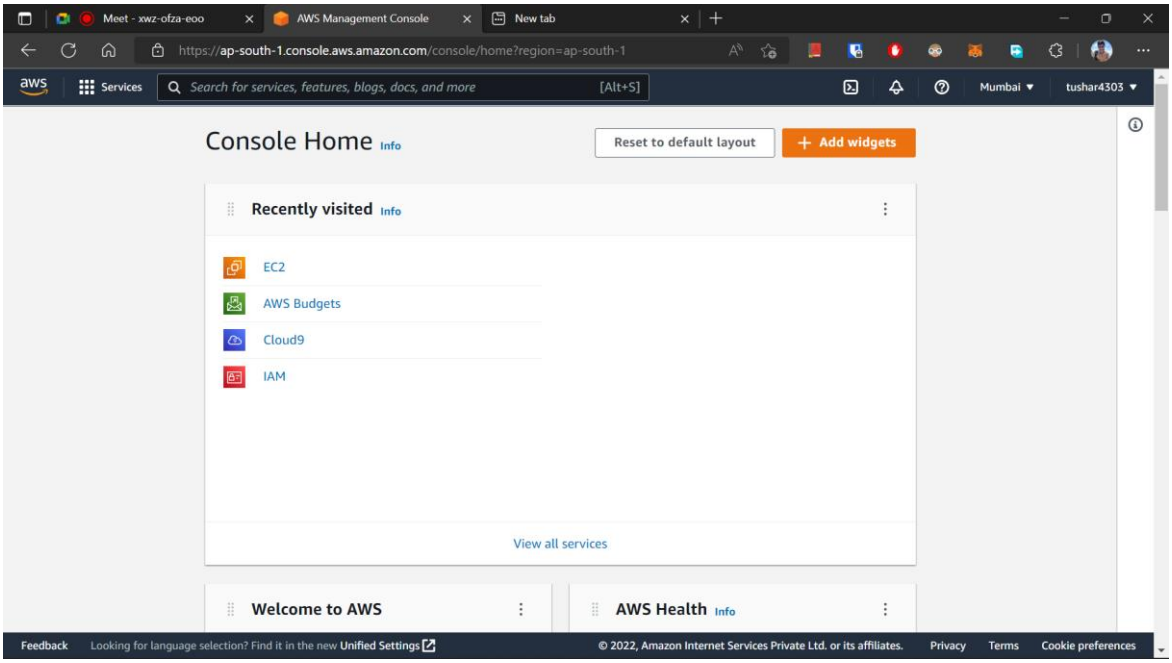
Subject: Advanced DevOps Lab

Experiment No 1:

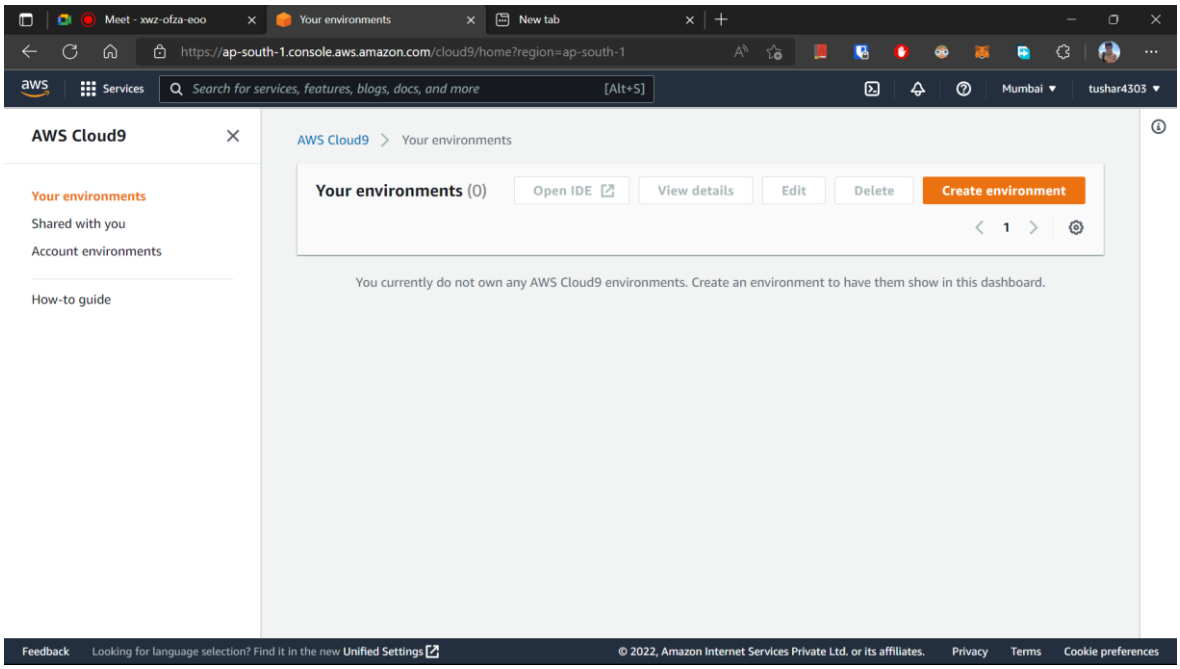
Aim: To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Step 1: Go to Amazon AWS's website and create an account.

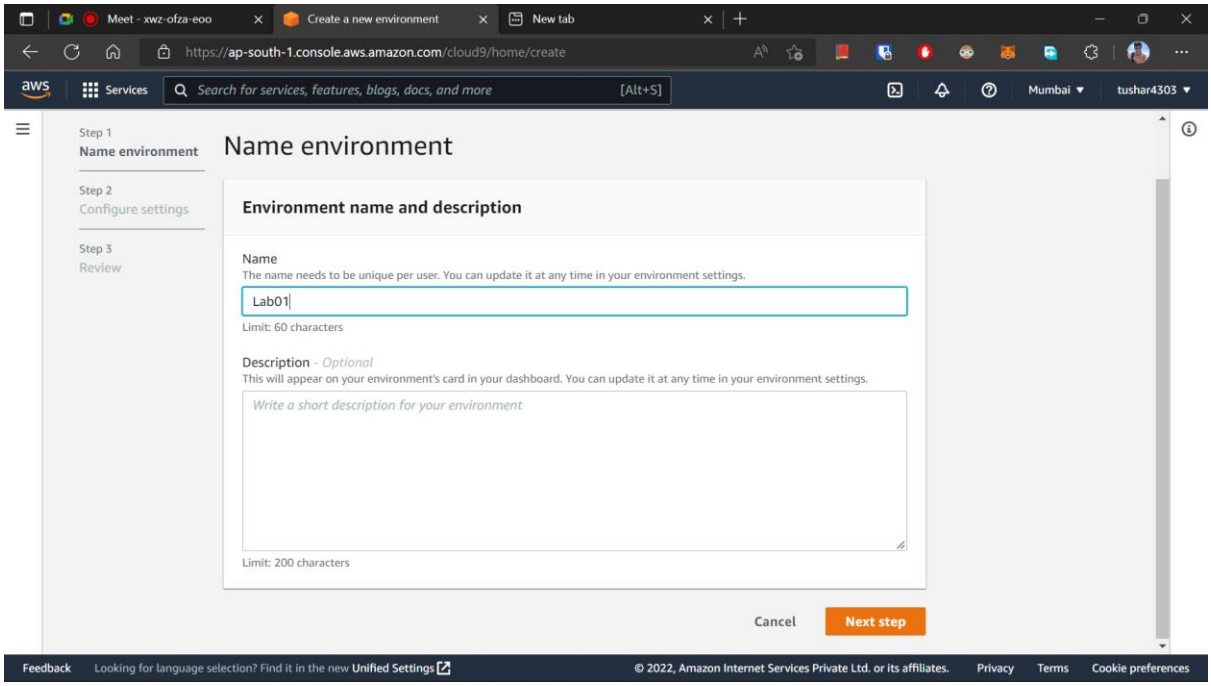




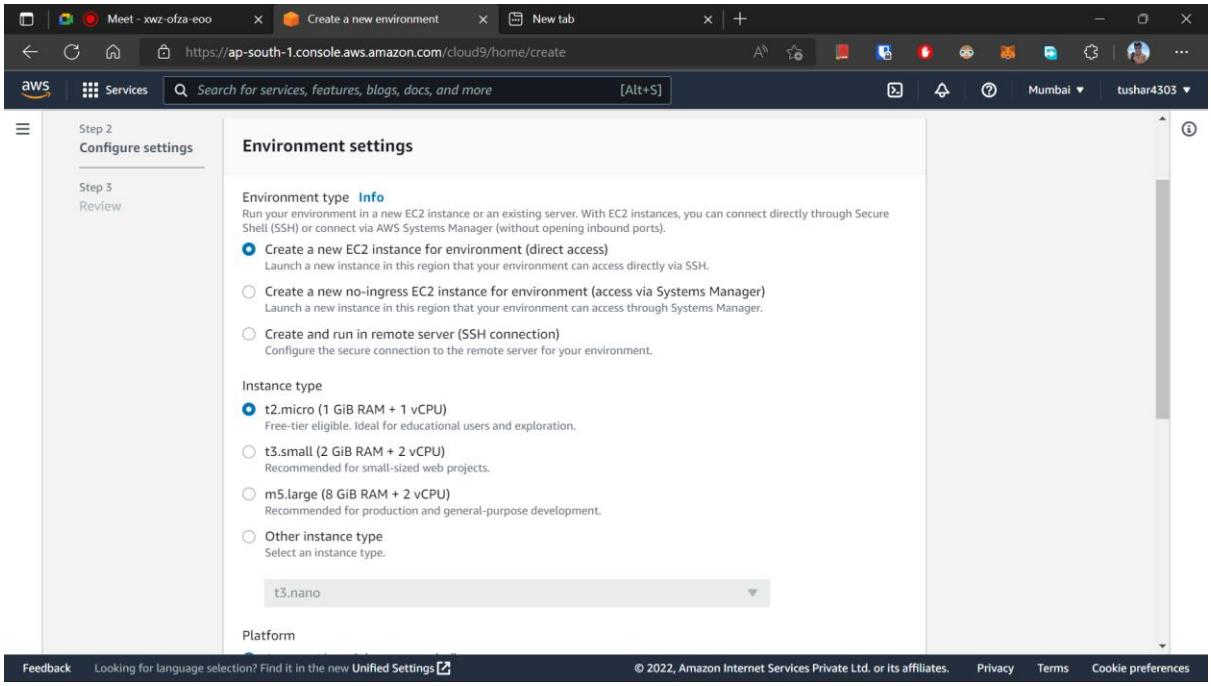
Step 2: Search for AWS Cloud9 using the search bar present on top, to create an environment for running the ide.



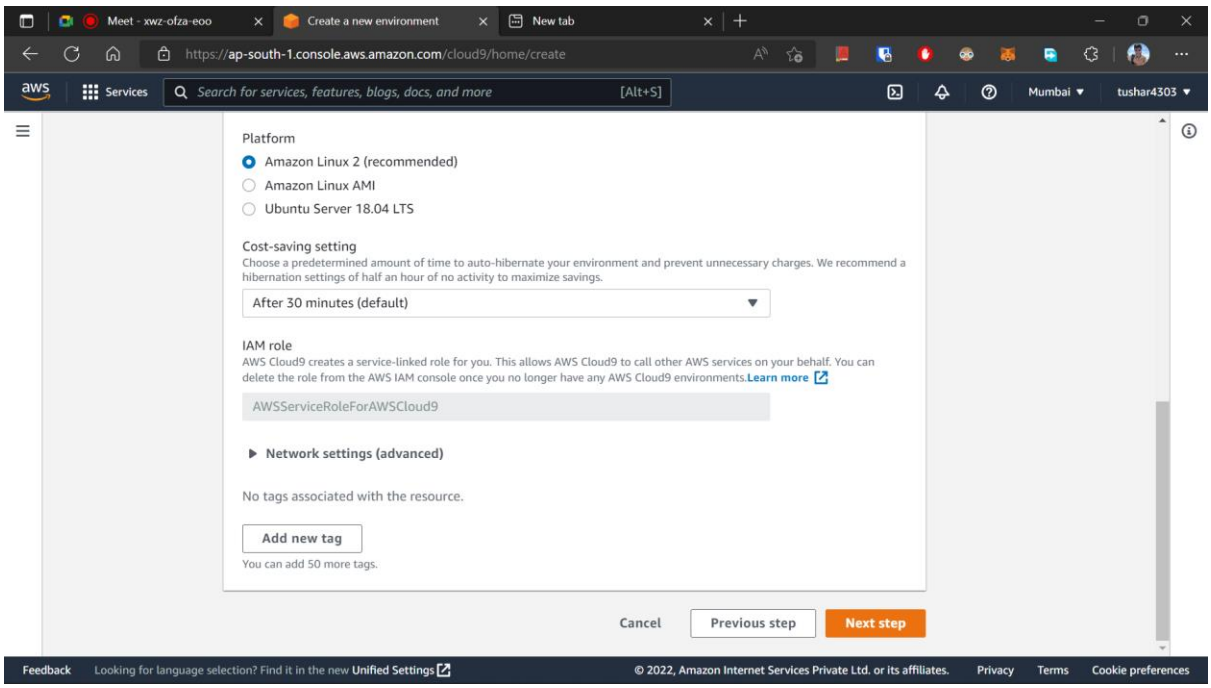
Step 3: Create an environment on which our ide would run.

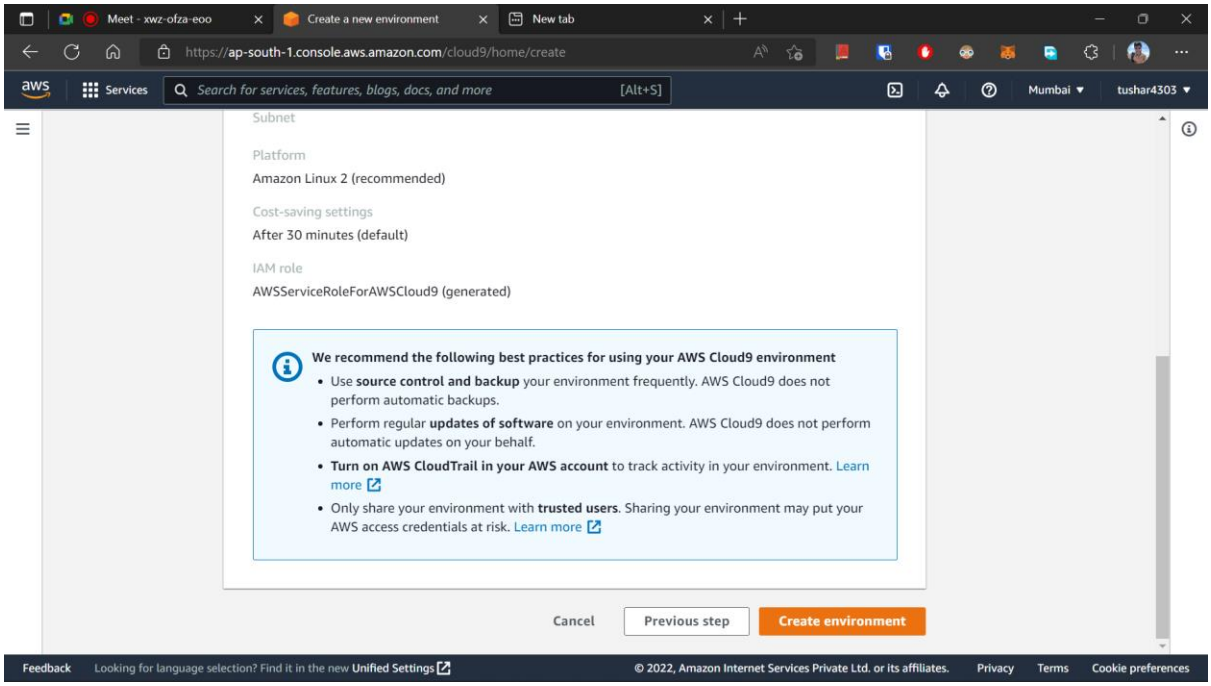


Step 4: Configure the environment according to your requirements.

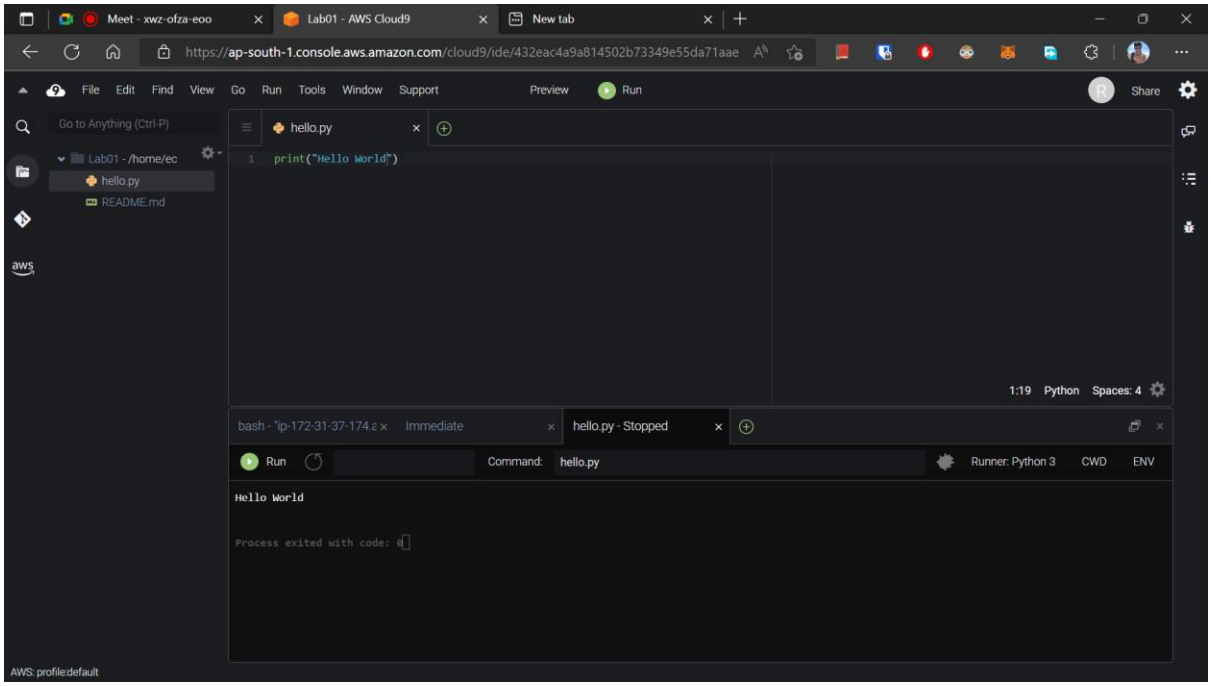


Choose the Operating system and if needed here we can also set a predefined time after which the environment hibernated.

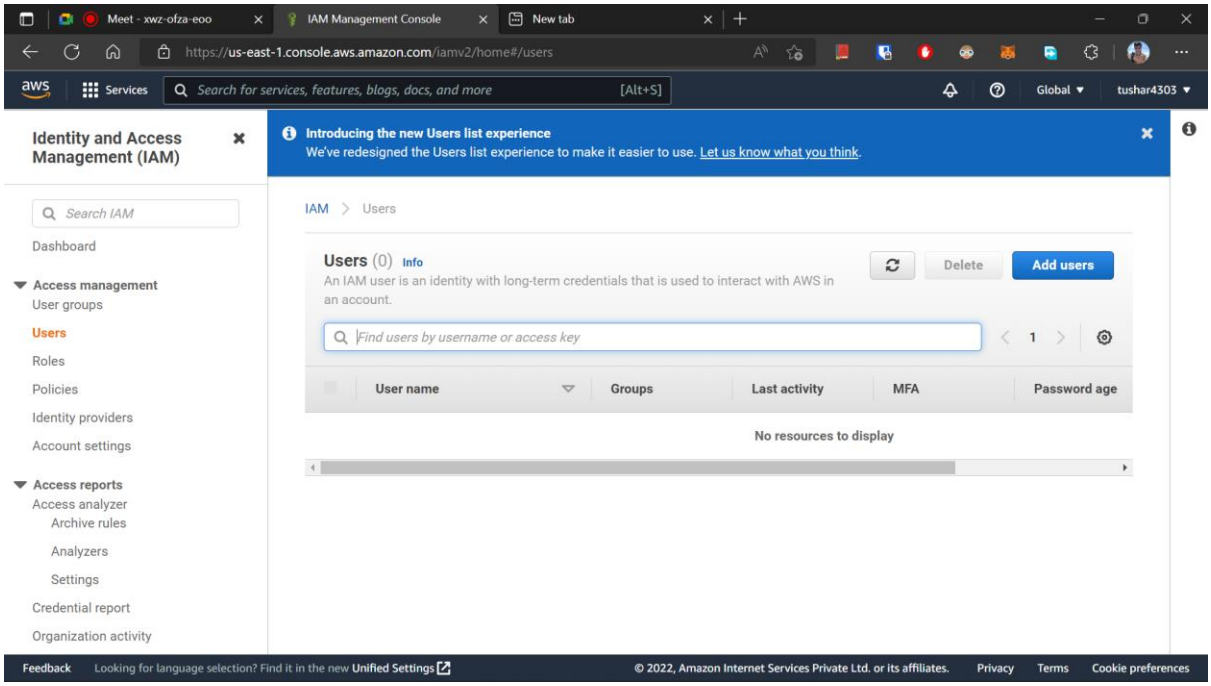




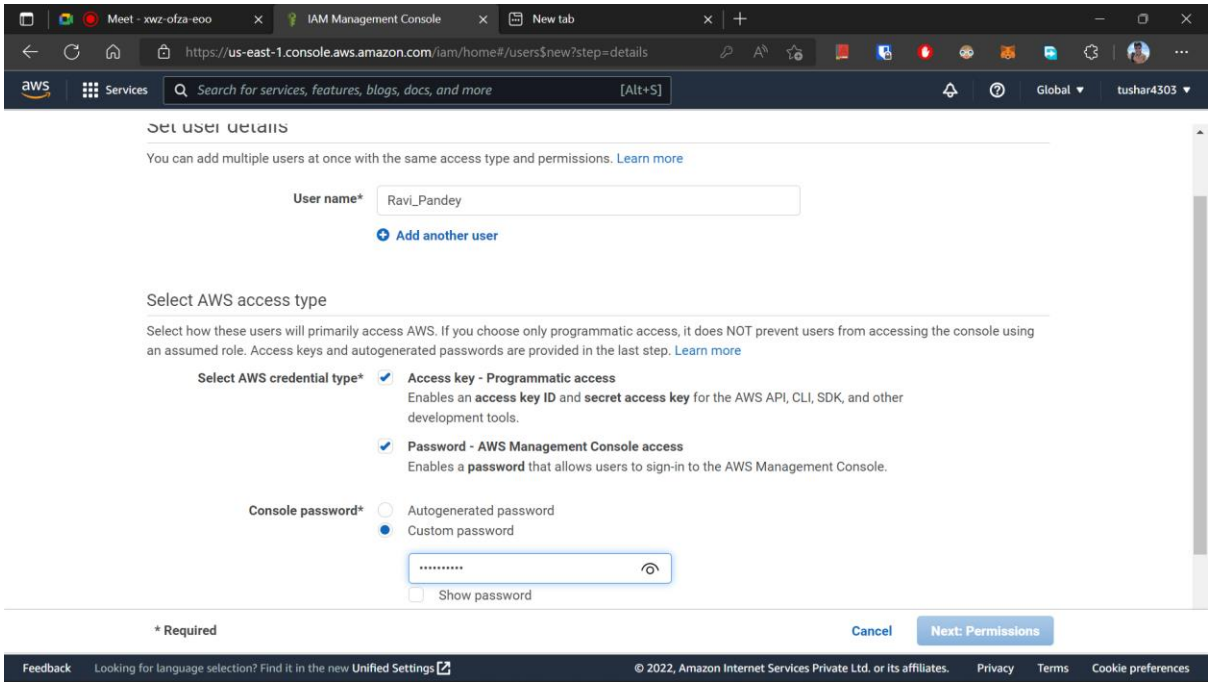
Step 5: Create the source code file.



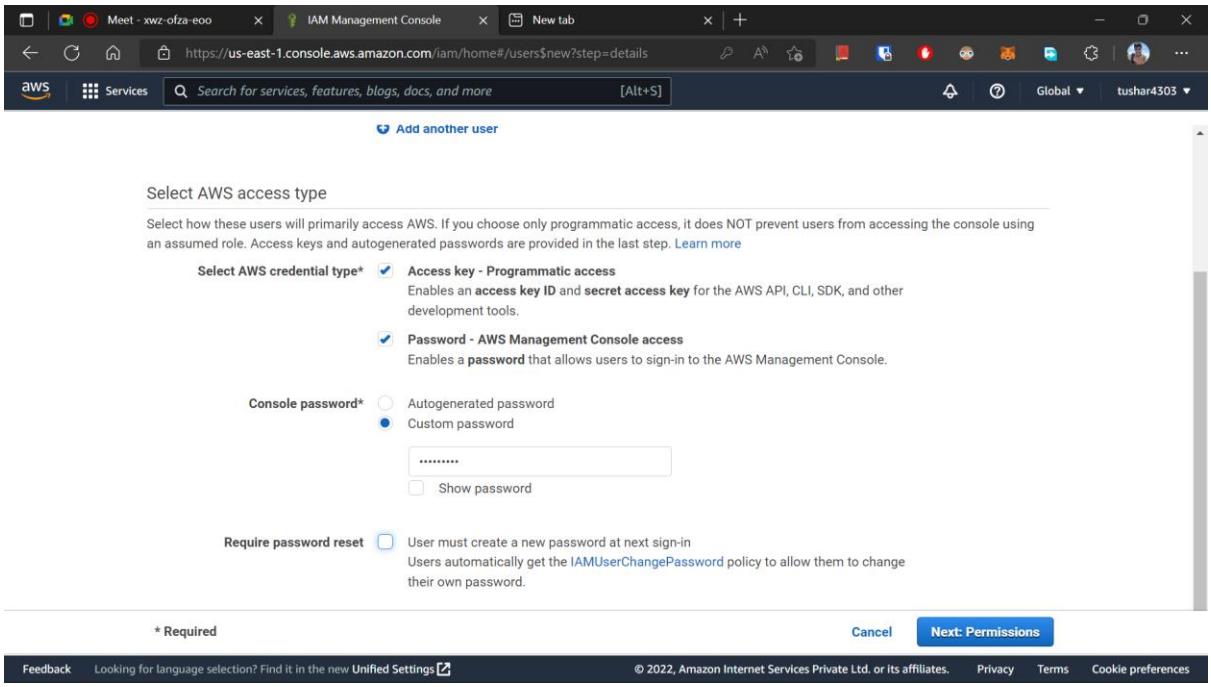
Step 6: Add a new IAM user using Identity Access Management console.



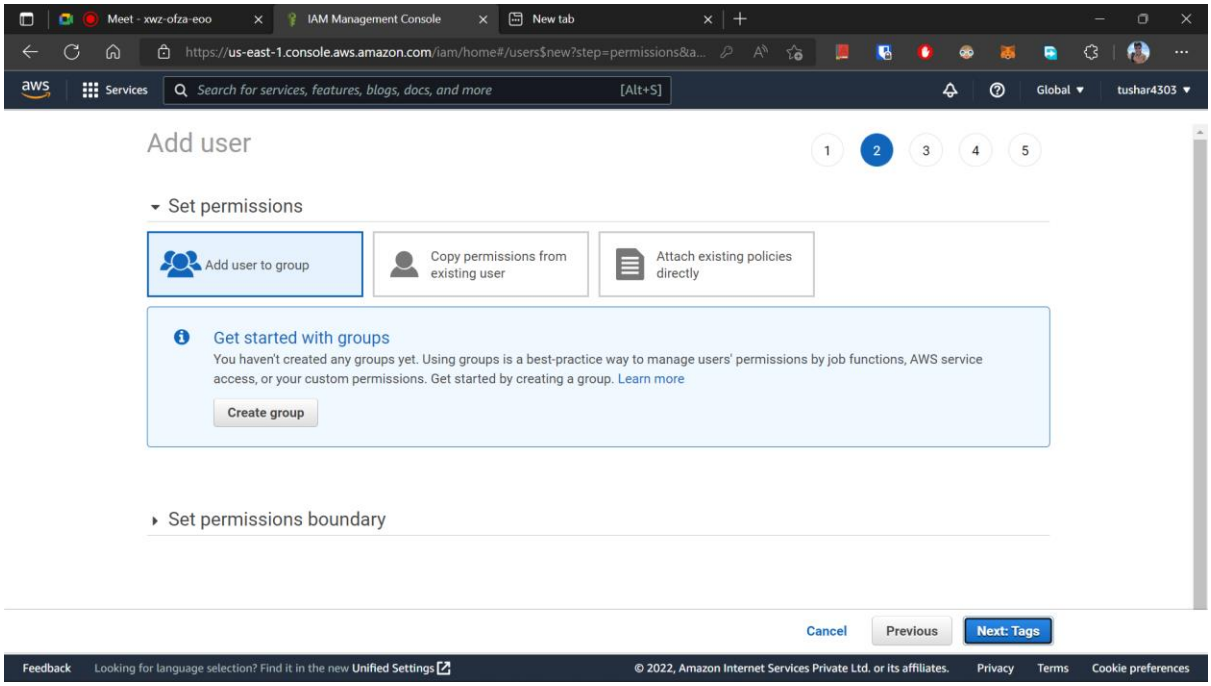
Step 7: Create credentials for the new user (Autogenerated / custom)

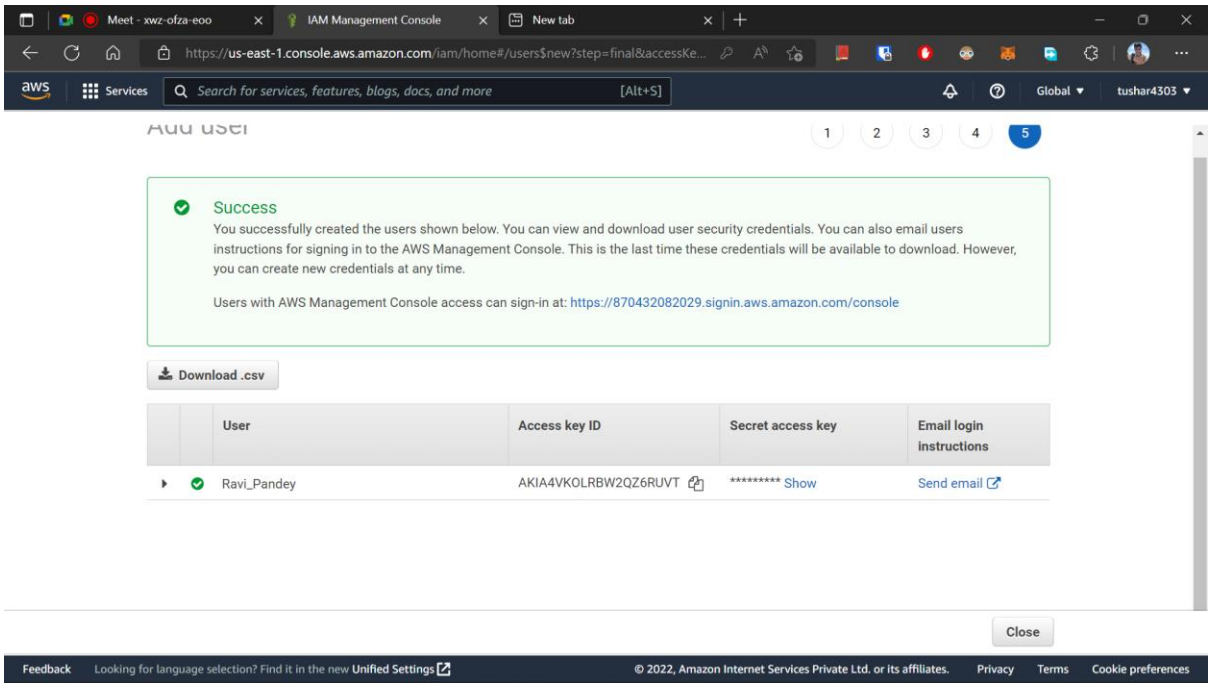


Optional step: Enable Require password reset flag if you want the user to reset his password on the next signup.

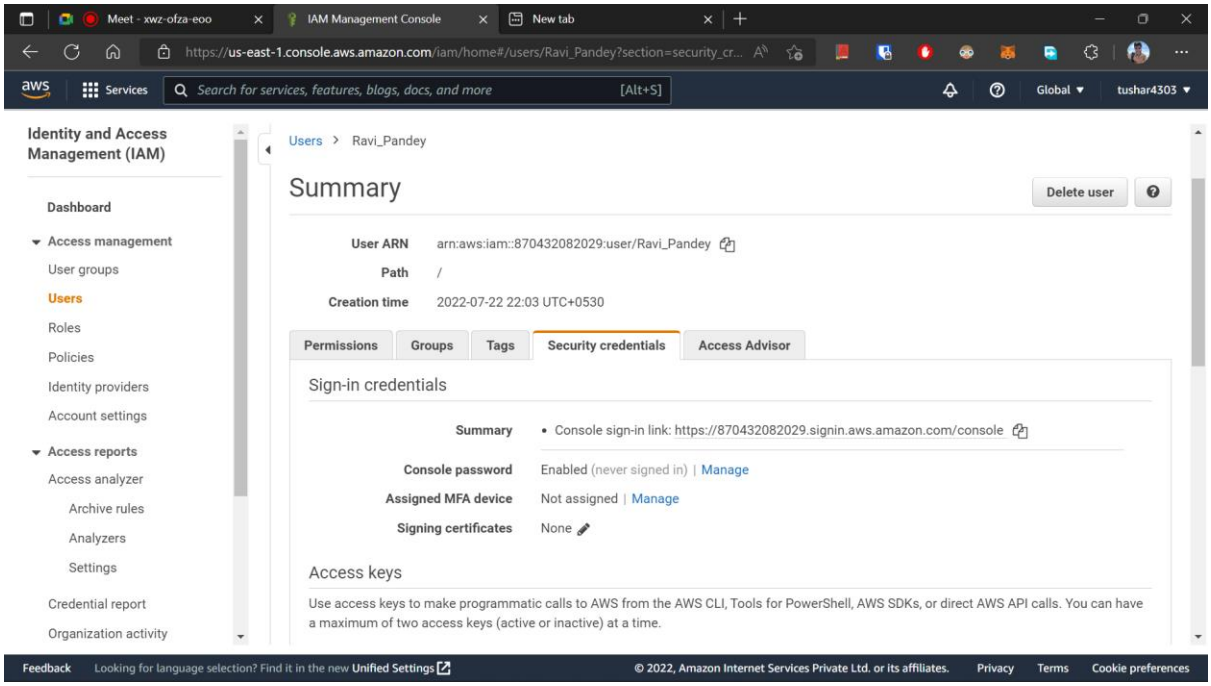


Step 8: Assign all the necessary permissions to the newly created user.

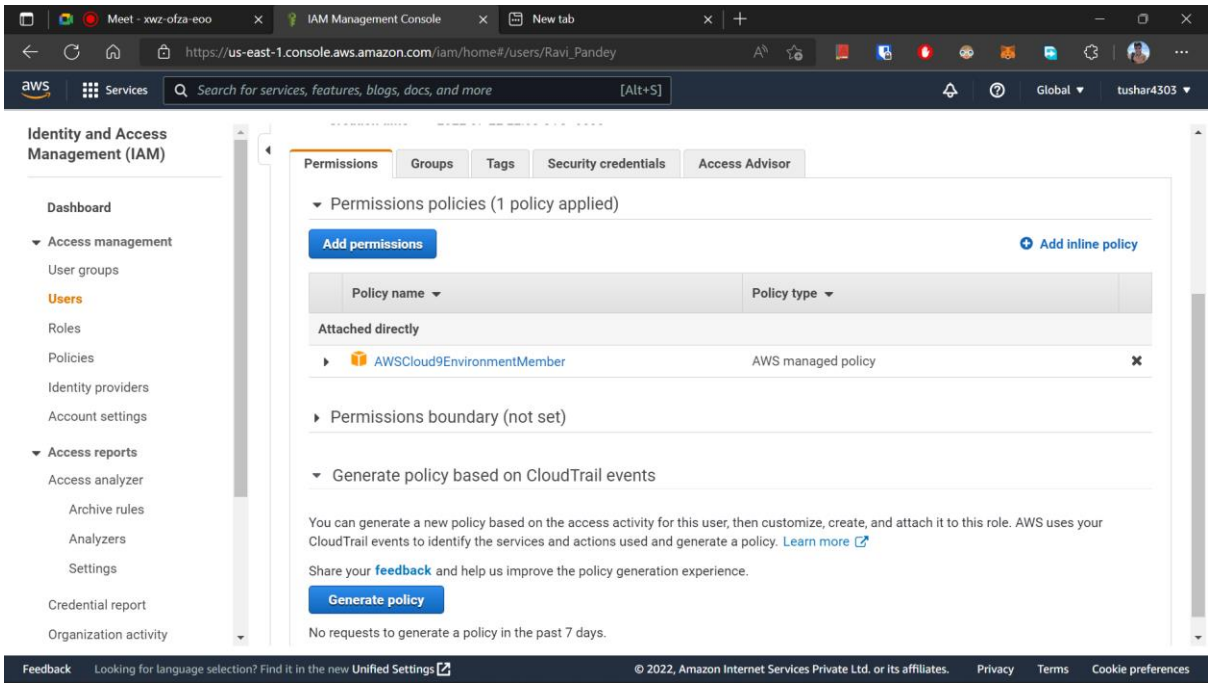


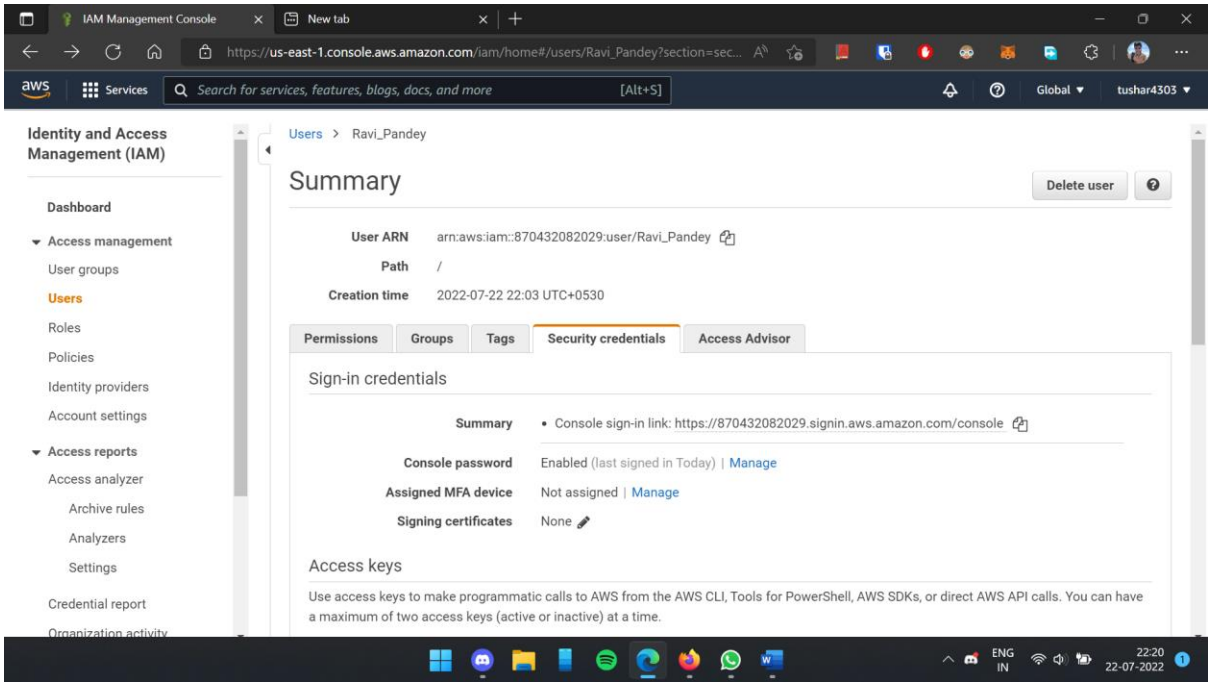


Step 9: Share these credentials with the concerned user/member.

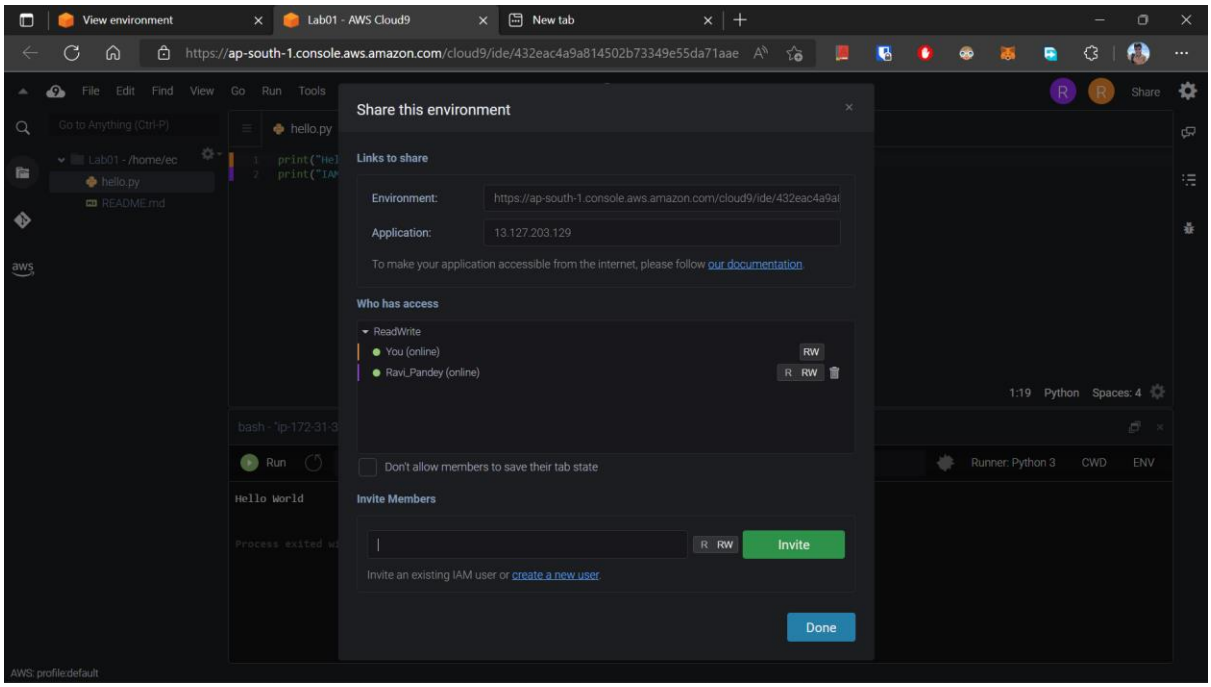


Step 10: Make sure that the user has permission to access the IDE environment.

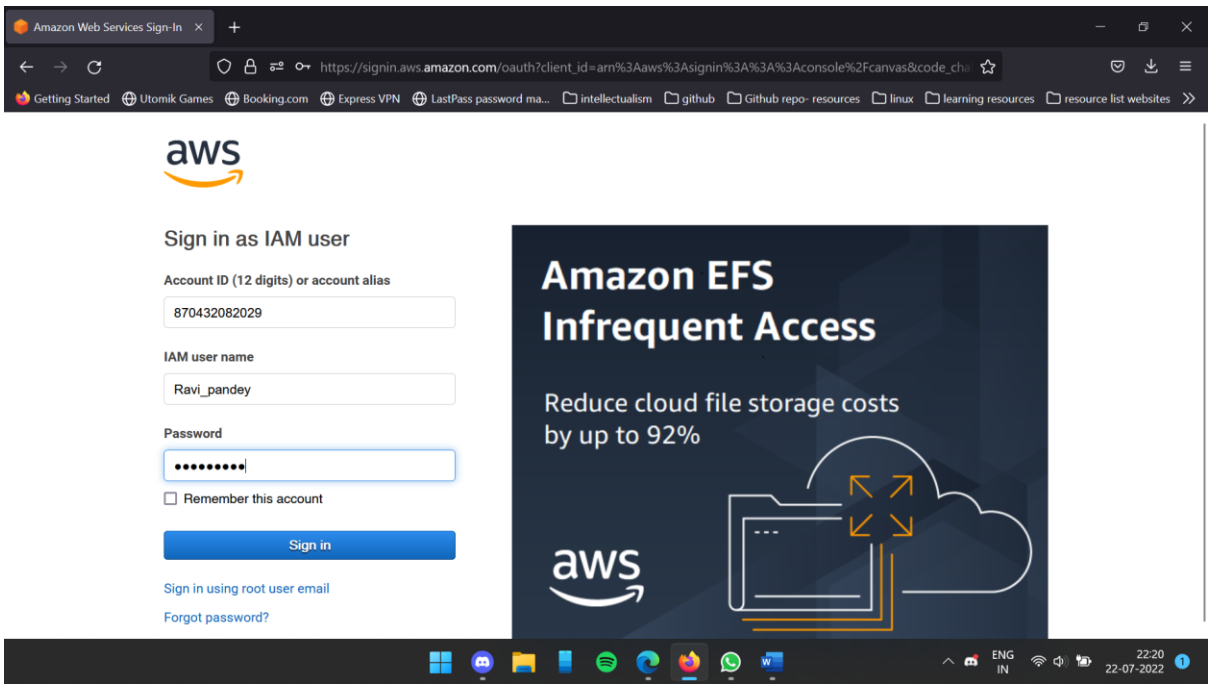




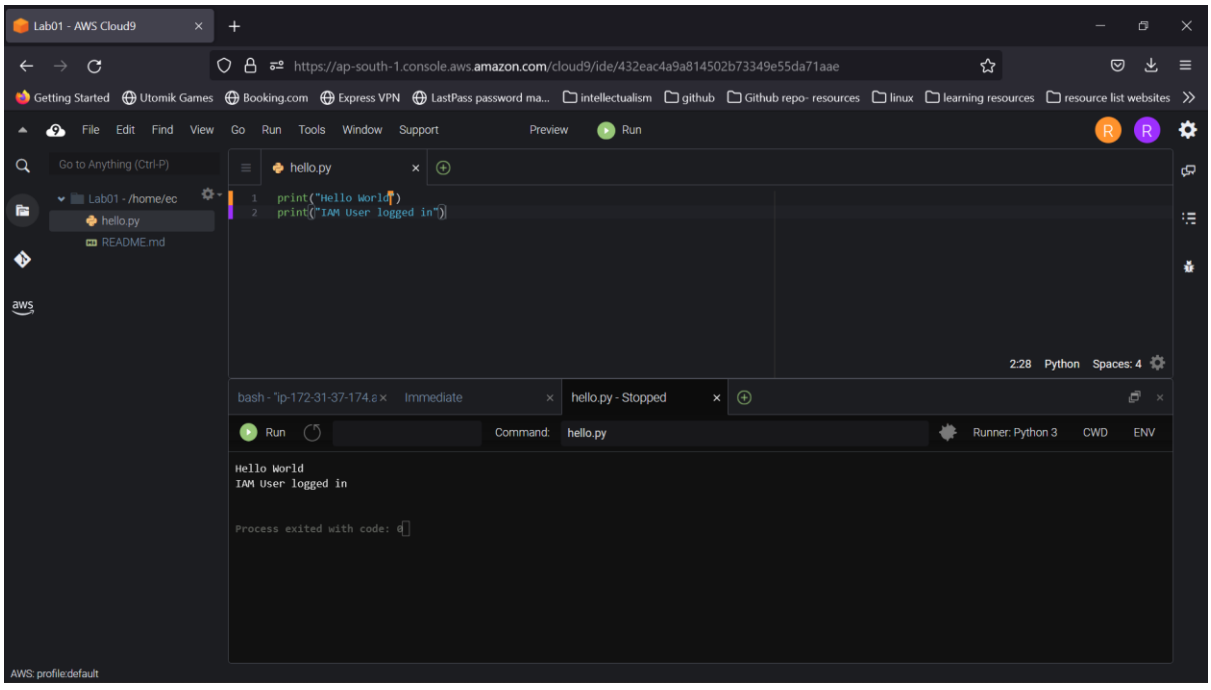
Step 11: Invite the newly created user to this IDE environment and share the link.

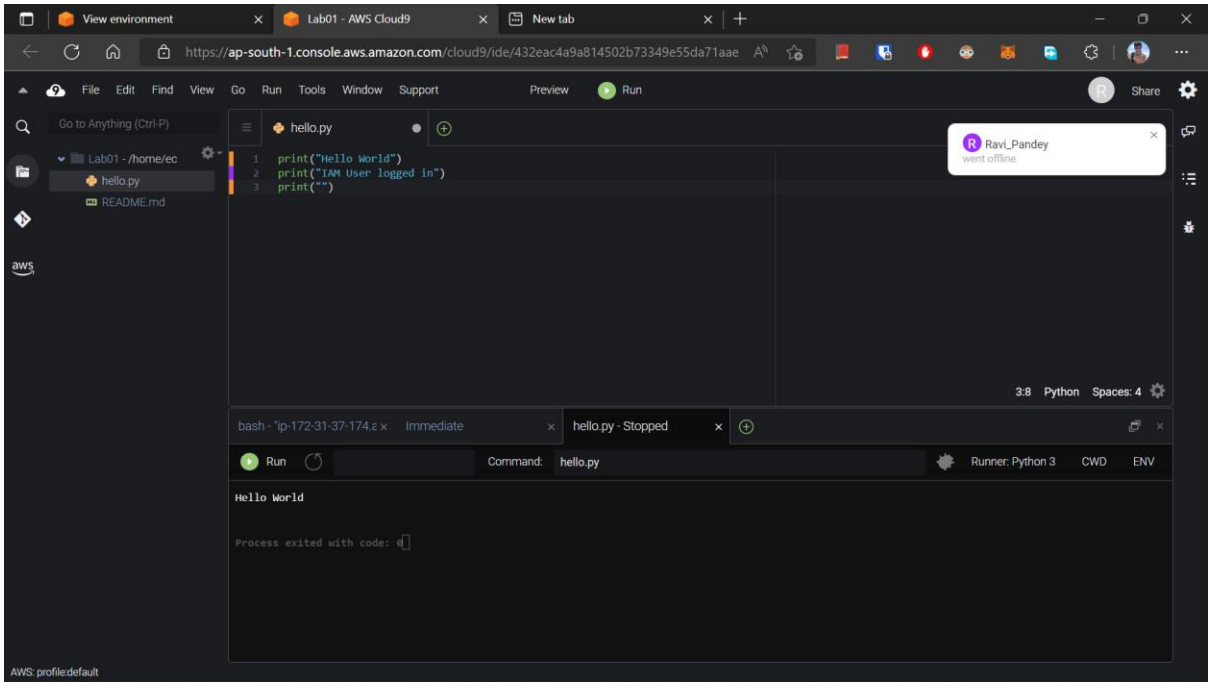


Step 12: Login from the IAM account and access the ide environment using the environment link.



As we can see here the two User can now simultaneously collaborate on the same source code, on the same IDE.




View environment
Lab01 - AWS Cloud9
New tab
https://ap-south-1.console.aws.amazon.com/cloud9/ide/432eac4a9a814502b73349e55da71aae
File Edit Find View Go Run Tools Window Support
Preview
Run
Share
Go to Anything (Ctrl-P)
Lab01 - /home/ec
hello.py
README.md
print("Hello World")
print("IAM User logged in")
print("")
Ravi_Pandey
went offline
3:8 Python Spaces: 4
bash - "ip-172-31-37-174.e
Immediate
hello.py - Stopped
Run
Command: hello.py
Runner: Python 3
CWD
ENV
Hello World
Process exited with code:
AWS: profile:default

ADVANCED DEVOPS LAB

Name: Tushar Padhy

Roll no: 42

Experiment No: 02

Date Of Submission: 7th August,

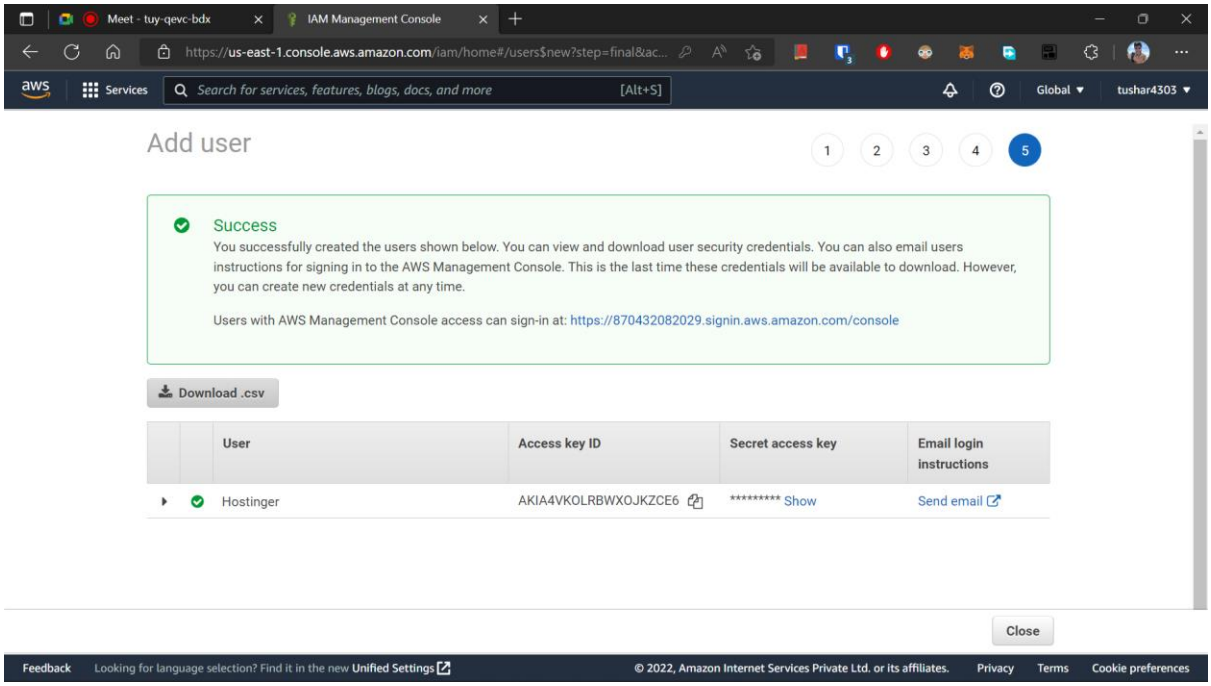
2022

Aim:

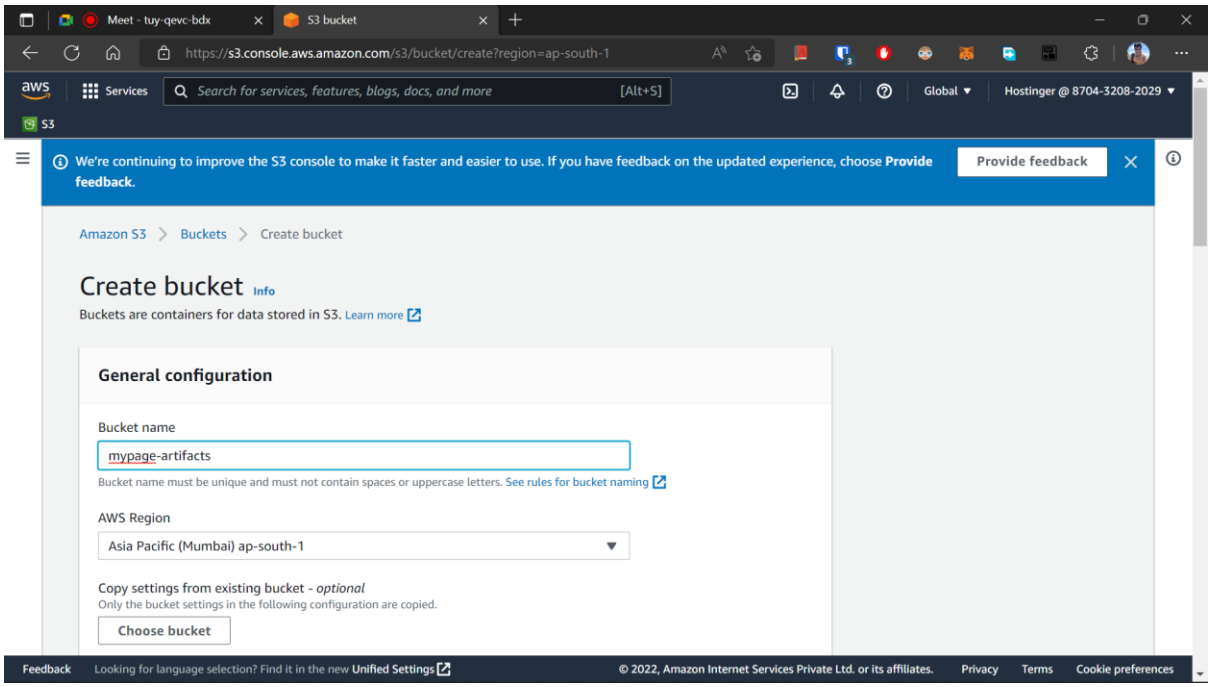
To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Output:

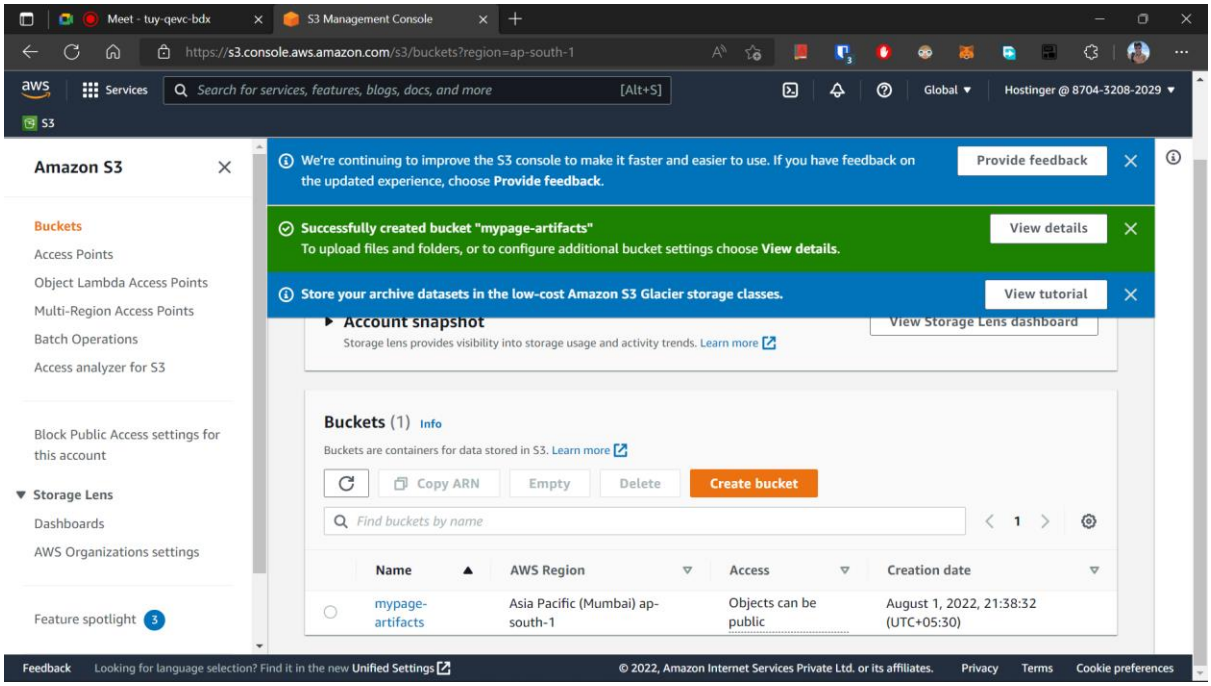
1. Create a IAM user



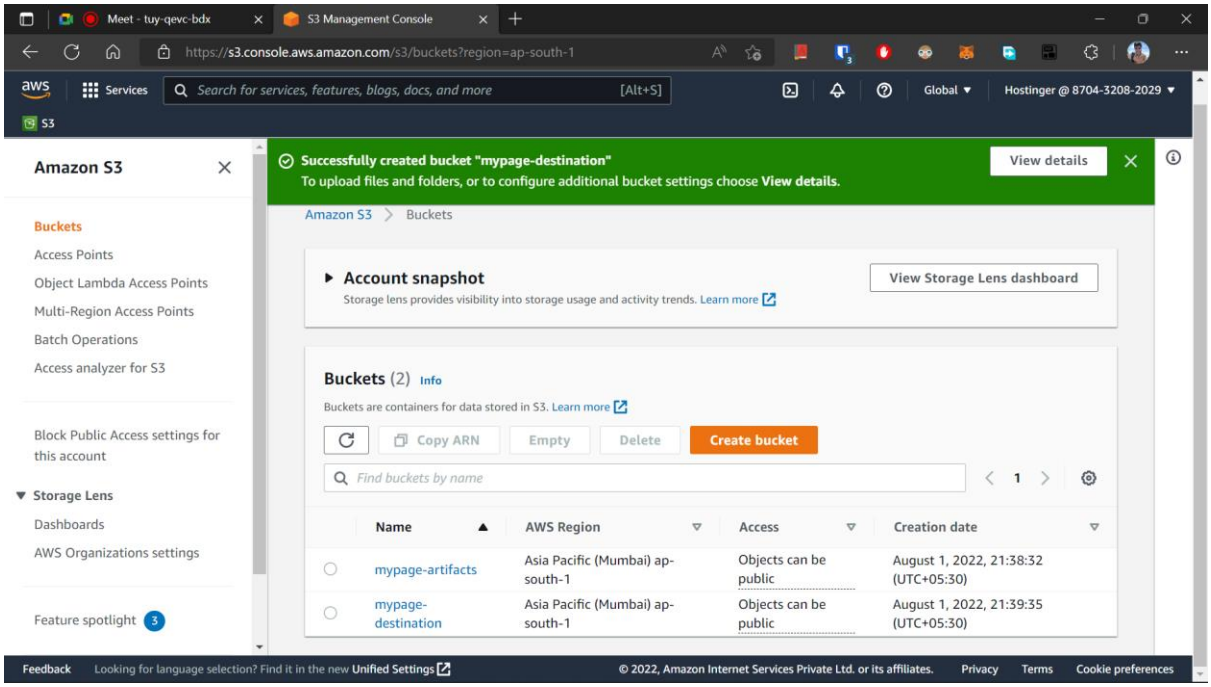
2. Create bucket artifact to store the artifacts

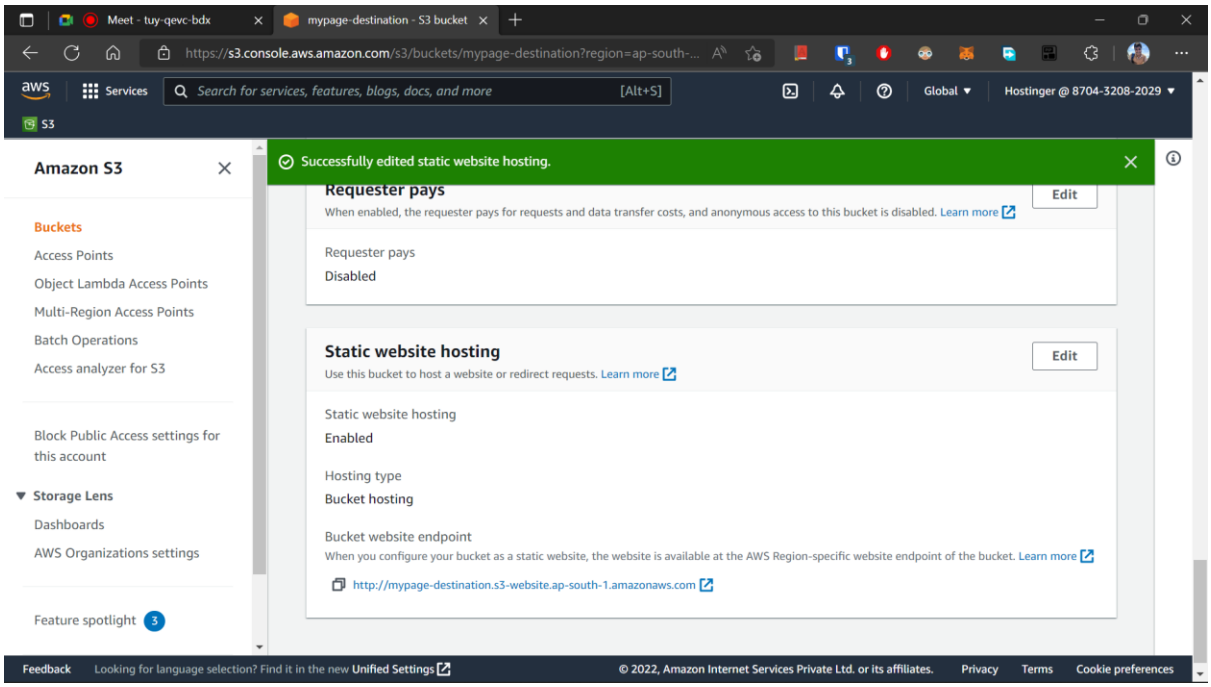


3. Create another bucket for destination where the files for the static website which will be hosted are stored. Enable the versioning too.

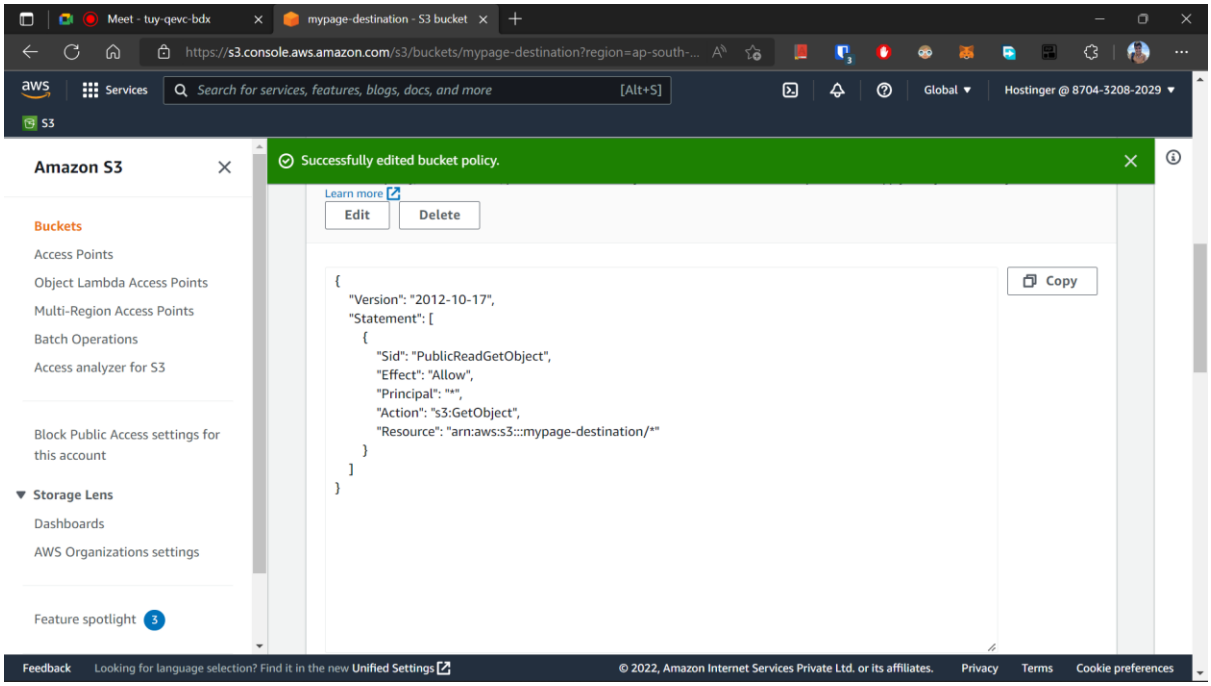


4. Enable the destination bucket hosting.

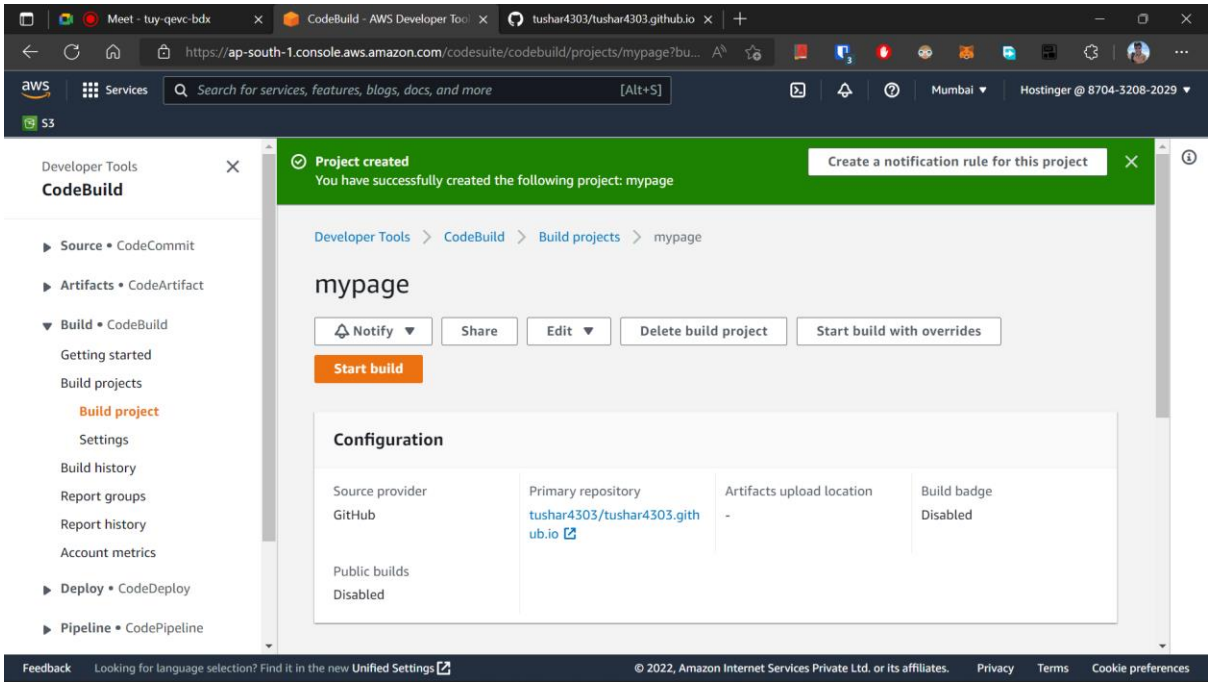




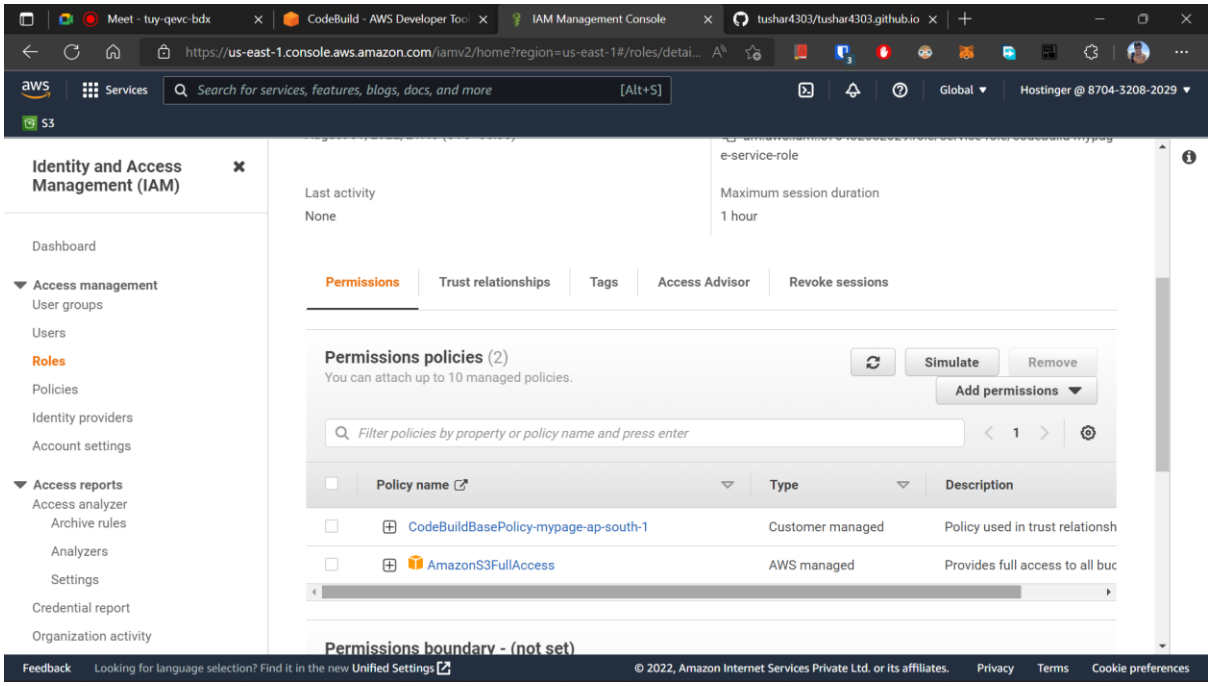
5. Add the bucket policy under destination bucket



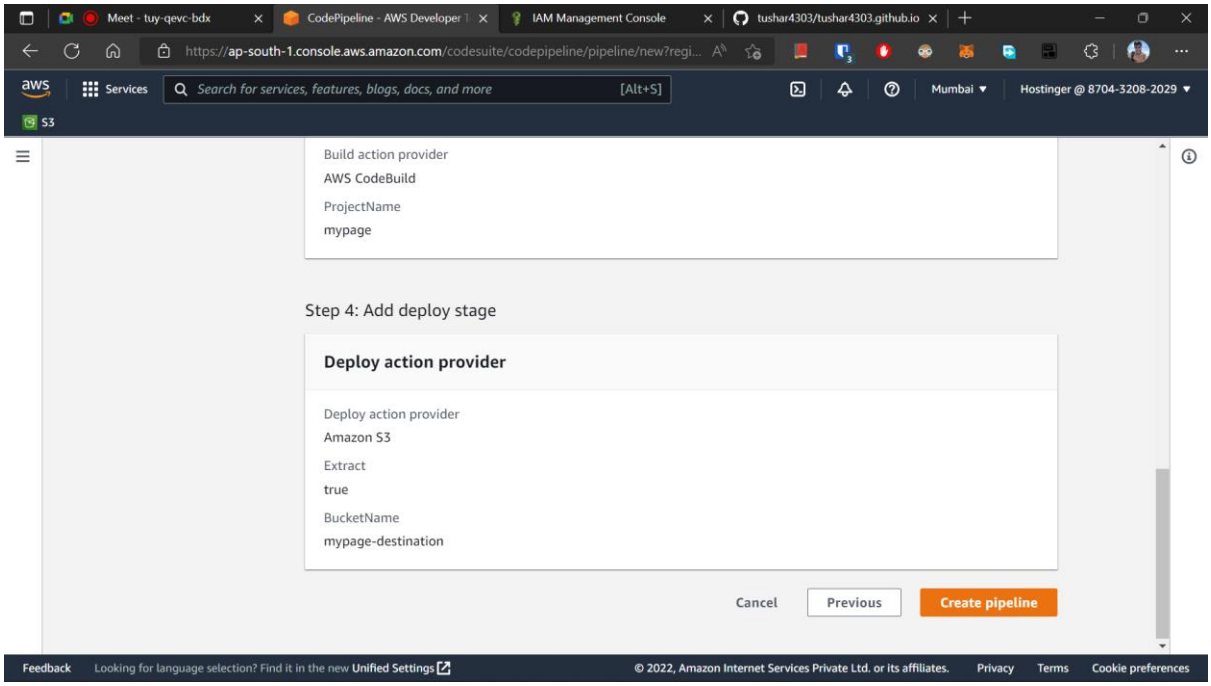
6. Create a build project under code build



7. Under the project, inside build details attach the s3 full access policy.

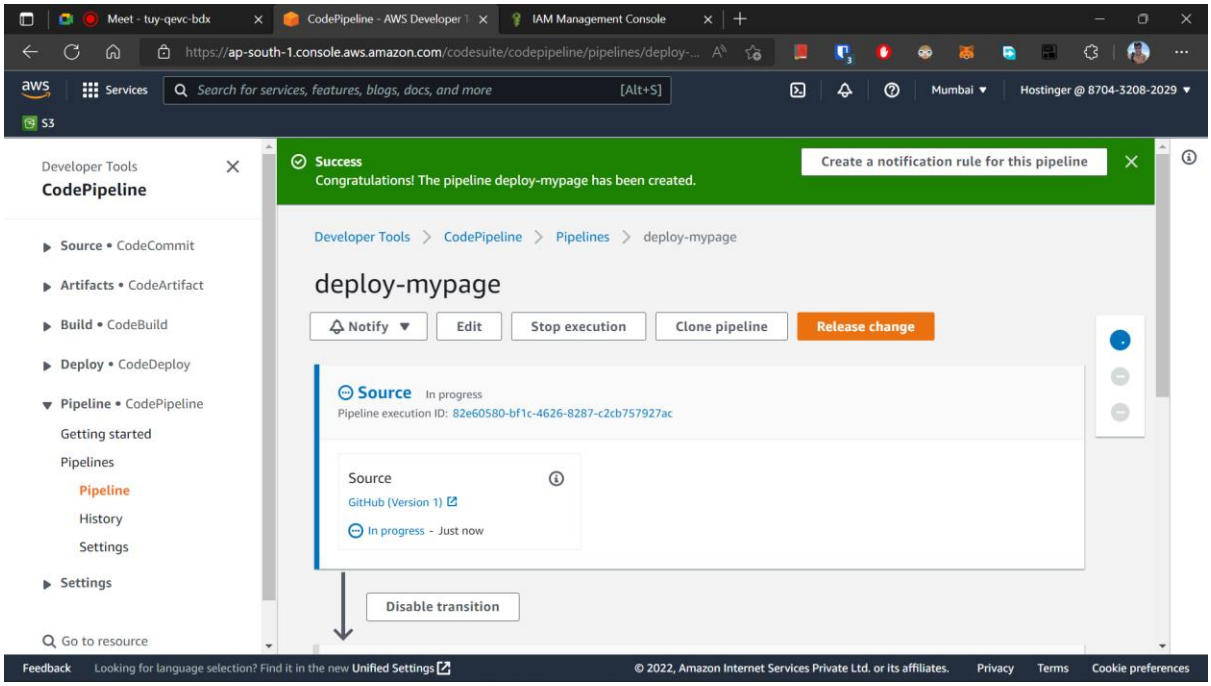


8. Now, create a pipeline and enter all the required details. (Important - don't give any access to CannedAcl)

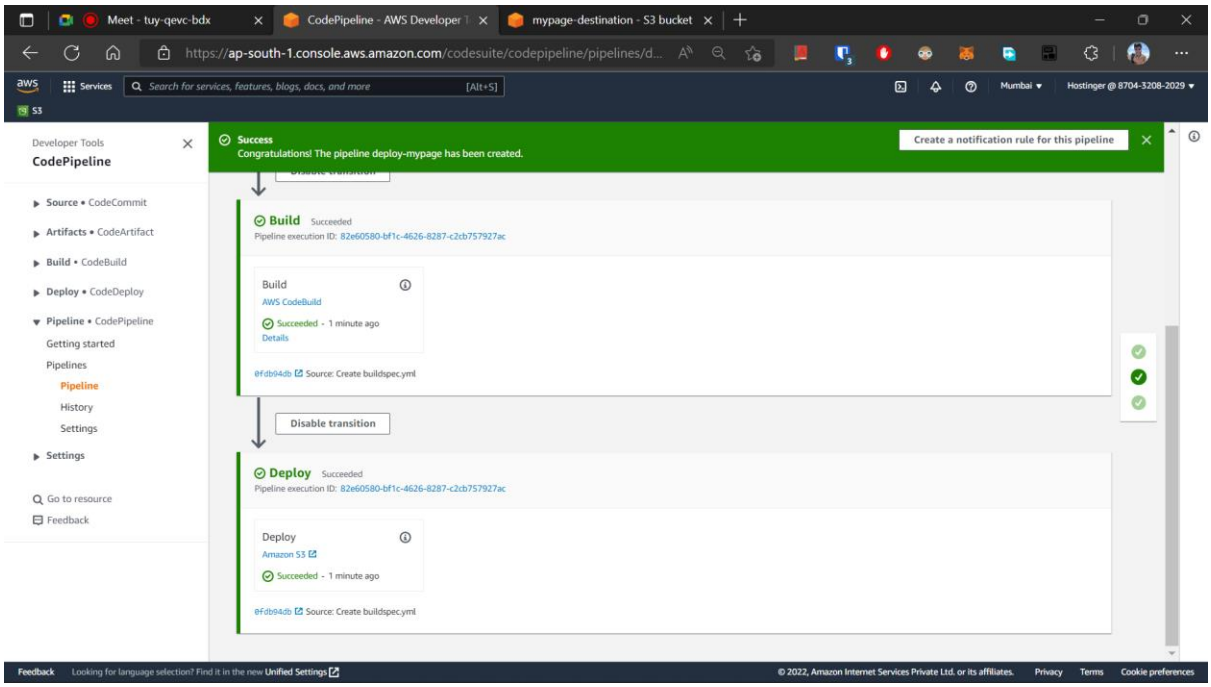


9. Now the build will start with three steps.

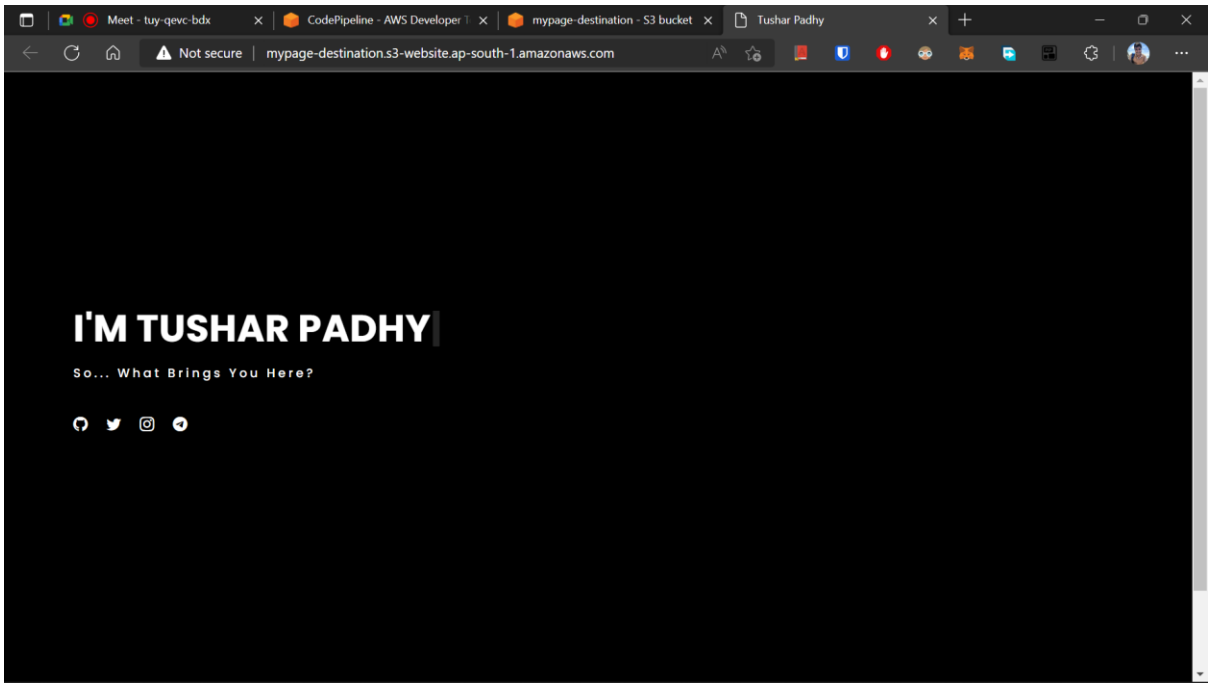
Step 1: Source



Step 2: Build
Step 3 : Deploy



10. And so a static website is hosted!



ADVANCE DEVOPS LAB

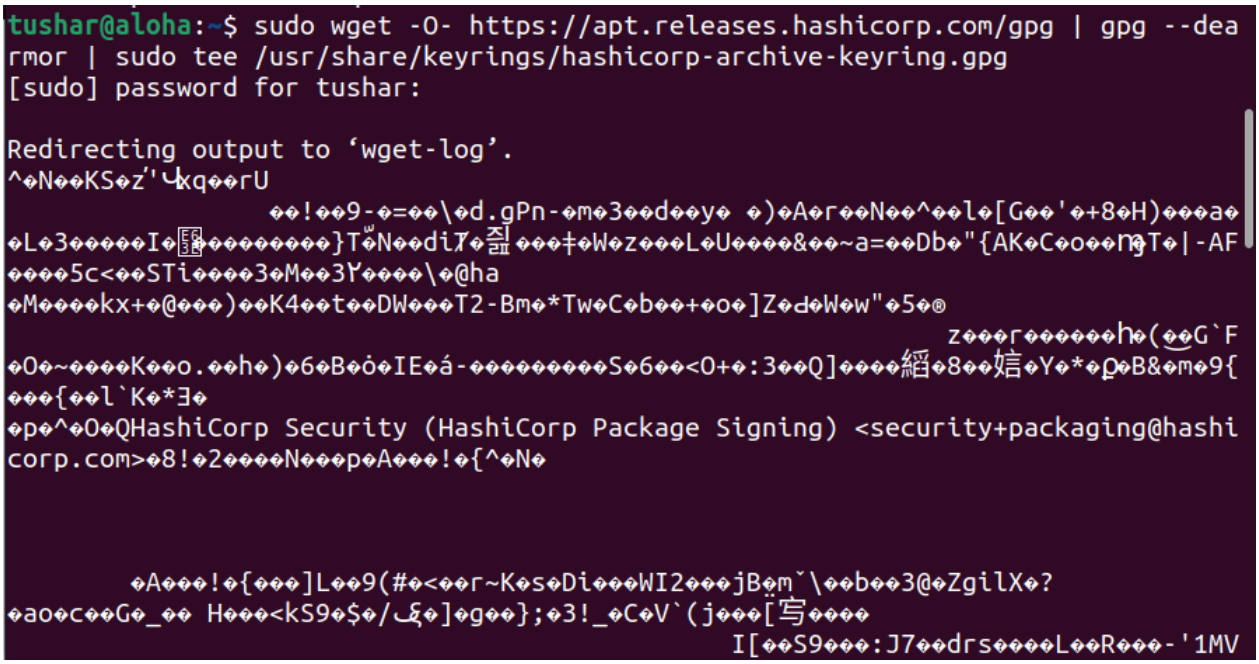
Name : Tushar Padhy
Roll No: 42
Exp No - 05

Aim:

To understand terraform lifecycle, core concepts/terminologies and install it on a Linux Machine

Output:

STEP 1 - Install the HashiCorp GPG key



STEP 2 - Download the package information from HashiCorp and Install Terraform from the new repository.

```
tushar@aloha:~$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyri
ng.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /
etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.re
leases.hashicorp.com jammy main
tushar@aloha:~$ sudo apt update && sudo apt install terraform
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Get:2 https://apt.releases.hashicorp.com jammy InRelease [11.1 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 https://packages.microsoft.com/repos/edge stable InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:8 https://apt.releases.hashicorp.com jammy/main amd64 Packages [62.9 kB]
Get:9 https://apt.releases.hashicorp.com jammy/main i386 Packages [15.3 kB]
Fetched 89.3 kB in 2s (43.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
248 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
```

STEP 3 - Verify the version

```
Unpacking terraform (1.2.7) ...
Setting up terraform (1.2.7) ...
tushar@aloha:~$ terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
```

ADVANCE DEVOPS LAB

Name : Tushar Padhy
Roll No: 42
Exp No - 06

Aim:

To Build, change, and destroy AWS / GCP /Microsoft Azure/ DigitalOcean infrastructure Using Terraform

Output:

Building Infrastructure

STEP 1 - Installing AWS CLI

```
tushar@aloha:~$ sudo apt  install awscli
[sudo] password for tushar:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docutils-common groff gsfonts imagemagick imagemagick-6-common
  imagemagick-6.q16 libde265-0 libfftw3-double3 libheif1 libilmbase25
  libjxr-tools libjxr0 liblqr-1-0 libmagickcore-6.q16-6
  libmagickcore-6.q16-6-extra libmagickwand-6.q16-6 libnetpbm10 libopenexr25
  netpbm psutils python3-botocore python3-docutils python3-jmespath
  python3-pyasn1 python3-pygments python3-roman python3-rsa python3-s3transfer
Suggested packages:
  imagemagick-doc autotrace enscript ffmpeg gimp gnuplot grads graphviz hp2xx
  html2ps libwmf-bin mplayer povray radiance texlive-base-bin transfig
  ufraw-batch libfftw3-bin libfftw3-dev inkscape docutils-doc
  fonts-linuxlibertine | ttf-linux-libertine texlive-lang-french
  texlive-latex-base texlive-latex-recommended python-pygments-doc
```

STEP 2 - Verification of AWS-CLI

```
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Setting up awscli (1.22.34-1) ...
tushar@aloha:~$ aws --version
aws-cli/1.22.34 Python/3.10.4 Linux/5.15.0-33-generic botocore/1.23.34
tushar@aloha:~$
```

STEP 3- Write Configuration

```
tushar@aloha:~$ mkdir Terraform-AWS
tushar@aloha:~$ cd Terraform-AWS
tushar@aloha:~/Terraform-AWS$ touch main.tf
tushar@aloha:~/Terraform-AWS$
```

```
tushar@aloha:~/Terraform-AWS$ cat main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.16"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region = "ap-south-1"
}

resource "aws_instance" "app_server" {
  ami           = "ami-830c94e3"
  instance_type = "t2.micro"

  tags = {
    Name = "ExampleAppServerInstance"
  }
}
```

STEP 4 - Initialize the directory.

```
tushar@aloha:~/Terraform-AWS$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 4.16"...
- Installing hashicorp/aws v4.28.0...
- Installed hashicorp/aws v4.28.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
tushar@aloha:~/Terraform-AWS$
```

STEP 5- Format and validate the configuration

```
tushar@aloha:~/Terraform-AWS$ terraform fmt
main.tf
tushar@aloha:~/Terraform-AWS$ terraform validate
Success! The configuration is valid.

tushar@aloha:~/Terraform-AWS$
```

STEP 6- Create Infrastructure

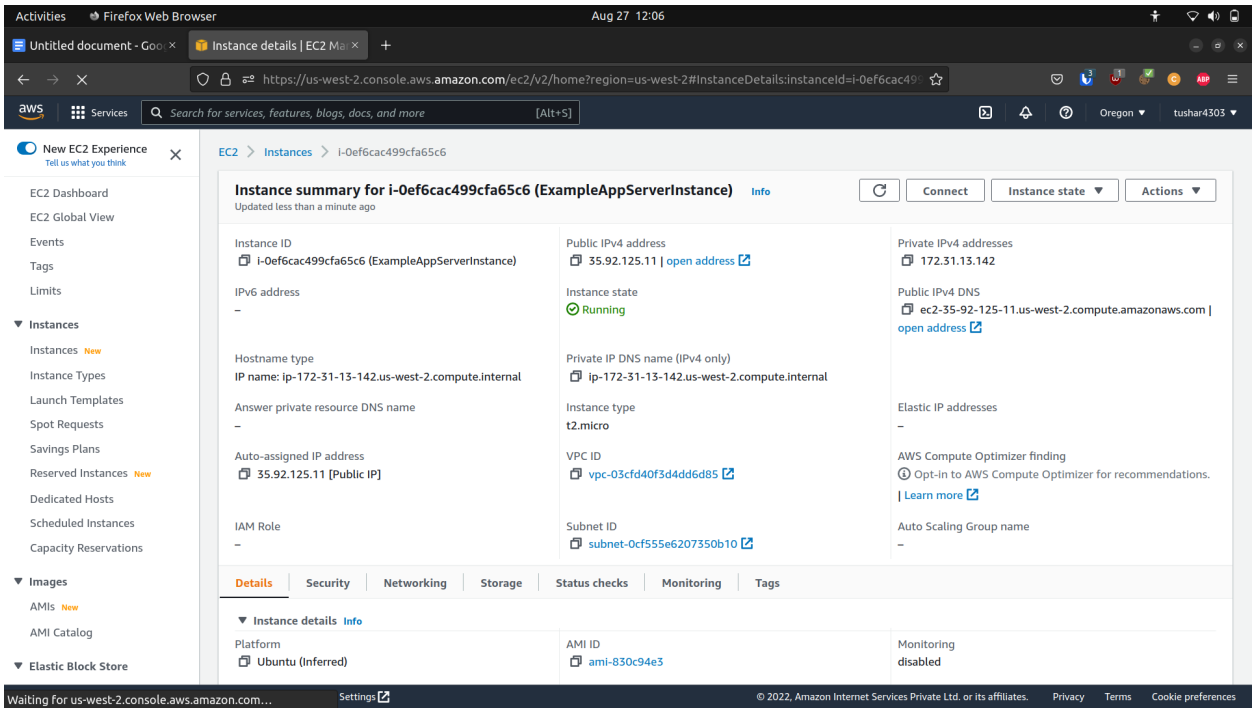
```
Default output format [None]:
tushar@aloha:~/Terraform-AWS$ terraform apply

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.app_server will be created
  + resource "aws_instance" "app_server" {
      + ami                                  = "ami-830c94e3"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

```
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.app_server: Creating...
aws_instance.app_server: Still creating... [10s elapsed]
aws_instance.app_server: Still creating... [20s elapsed]
aws_instance.app_server: Still creating... [30s elapsed]
aws_instance.app_server: Still creating... [40s elapsed]
aws_instance.app_server: Creation complete after 47s [id=i-0ef6cac499cfa65c6]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
tushar@aloha:~/Terraform-AWS$
```

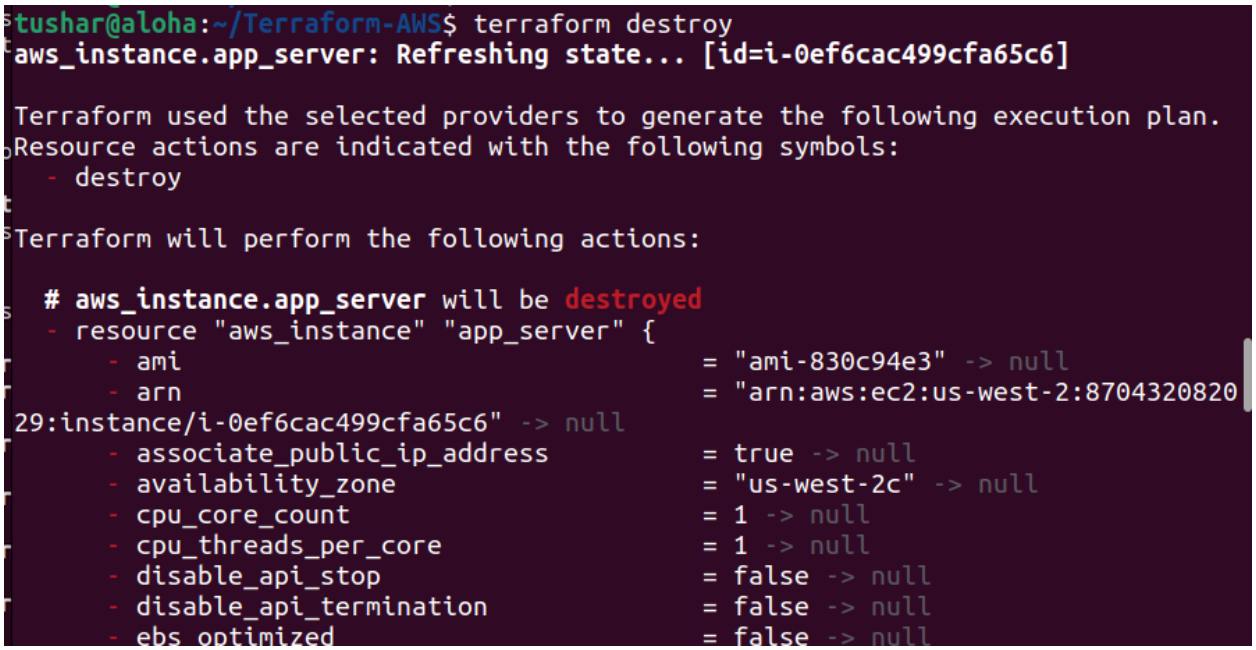
STEP 7- Inspect state

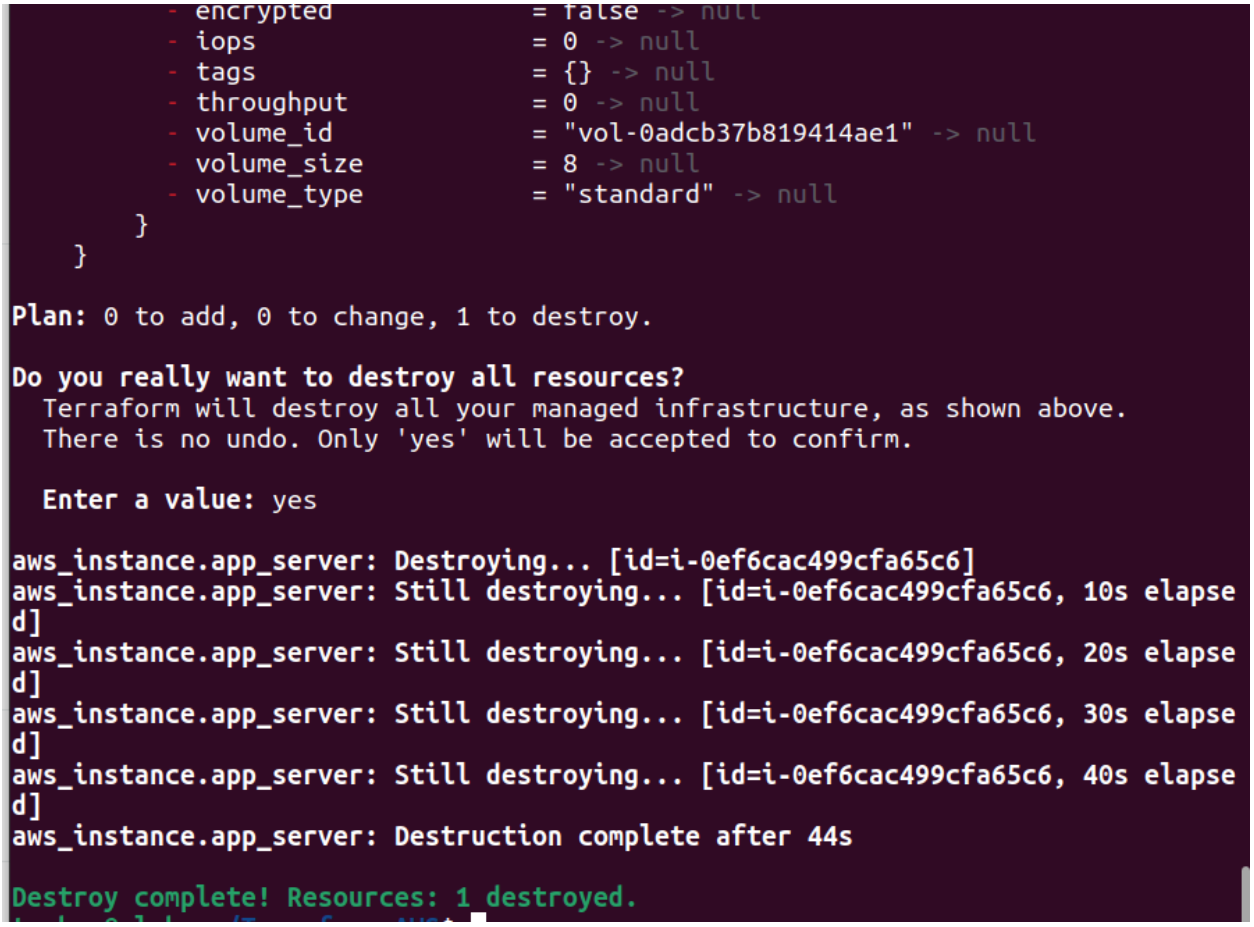
```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
tushar@aloha:~/Terraform-AWS$ terraform show
# aws_instance.app_server:
resource "aws_instance" "app_server" {
    ami                                  = "ami-830c94e3"
    arn                                  = "arn:aws:ec2:us-west-2:870432082029:
nstance/i-0ef6cac499cfa65c6"
    associate_public_ip_address          = true
    availability_zone                    = "us-west-2c"
    cpu_core_count                       = 1
    cpu_threads_per_core                 = 1
    disable_api_stop                     = false
    disable_api_termination              = false
    ebs_optimized                        = false
    get_password_data                    = false
    hibernation                          = false
    id                                   = "i-0ef6cac499cfa65c6"
    instance_initiated_shutdown_behavior = "stop"
    instance_state                       = "running"
    instance_type                        = "t2.micro"
    ipv6_address_count                   = 0
    ipv6_addresses                       = []
    monitoring                           = false
}
tushar@aloha:~/Terraform-AWS$ terraform state list
aws_instance.app_server
tushar@aloha:~/Terraform-AWS$
```

STEP 8- Checking instance of AWS



STEP 9- Terraform destroy



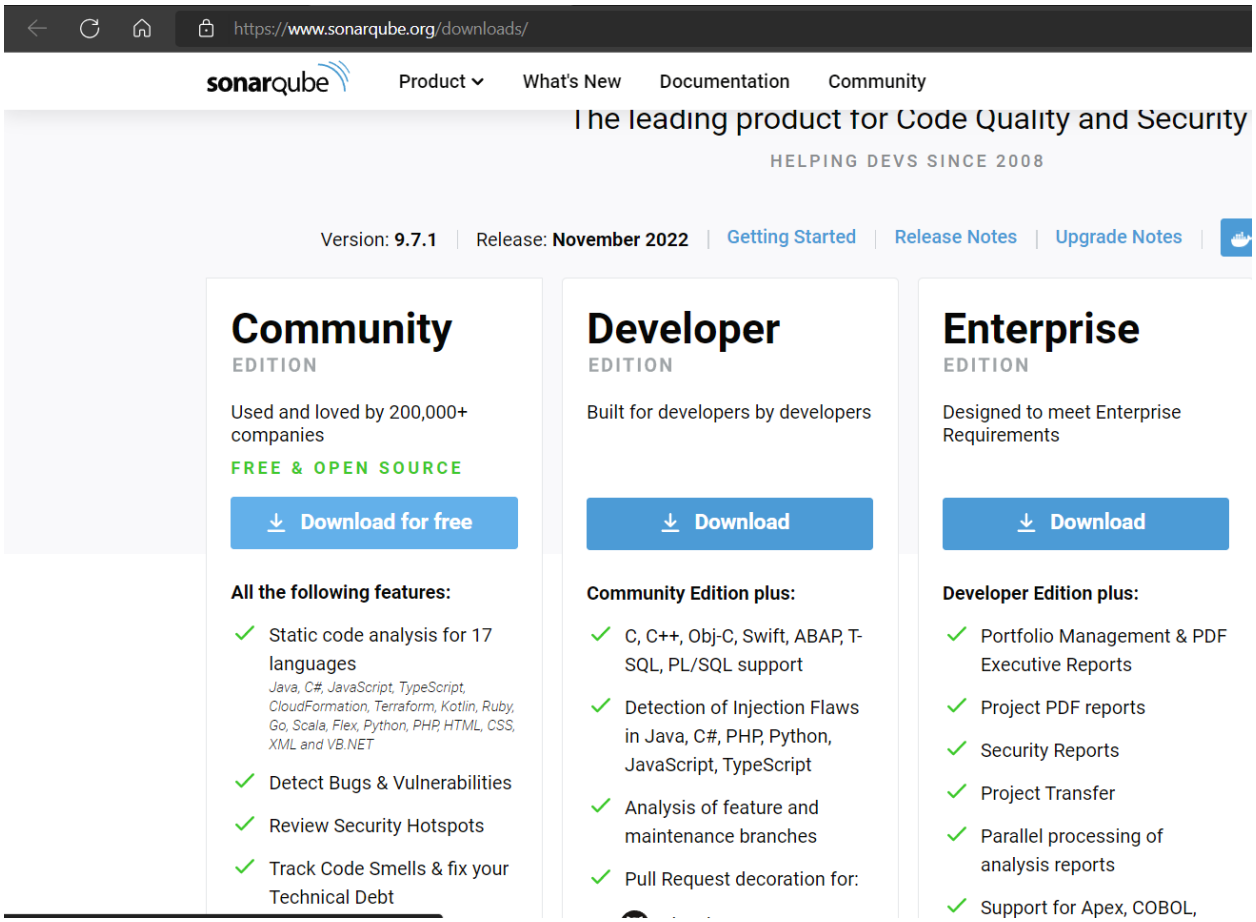

- encrypted = false -> null
- iops = 0 -> null
- tags = {} -> null
- throughput = 0 -> null
- volume_id = "vol-0adcb37b819414ae1" -> null
- volume_size = 8 -> null
- volume_type = "standard" -> null
}
}
Plan: 0 to add, 0 to change, 1 to destroy.
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
Enter a value: yes
aws_instance.app_server: Destroying... [id=i-0ef6cac499cfa65c6]
aws_instance.app_server: Still destroying... [id=i-0ef6cac499cfa65c6, 10s elapsed]
aws_instance.app_server: Still destroying... [id=i-0ef6cac499cfa65c6, 20s elapsed]
aws_instance.app_server: Still destroying... [id=i-0ef6cac499cfa65c6, 30s elapsed]
aws_instance.app_server: Still destroying... [id=i-0ef6cac499cfa65c6, 40s elapsed]
aws_instance.app_server: Destruction complete after 44s
Destroy complete! Resources: 1 destroyed.

ADVANCED DEVOPS LAB

Name : Tushar Padhy Roll no : 42
Experiment No : 08

Aim: Perform static analysis using sonarqube and show the analysis

1. Download the SonarQube Community Edition zip file.



2. Download and install Java 11 on your system.

https://adoptium.net/temurin/releases/?version=11

TEMURIN by ADOPTIUM

The latest releases recommended for use in production are listed below, and are regularly updated and supported by the Adoptium community. Migration help, container images and package installation guides are available in the documentation section. You can read the Release Notes for each version thanks to our friends at Foojay.io!

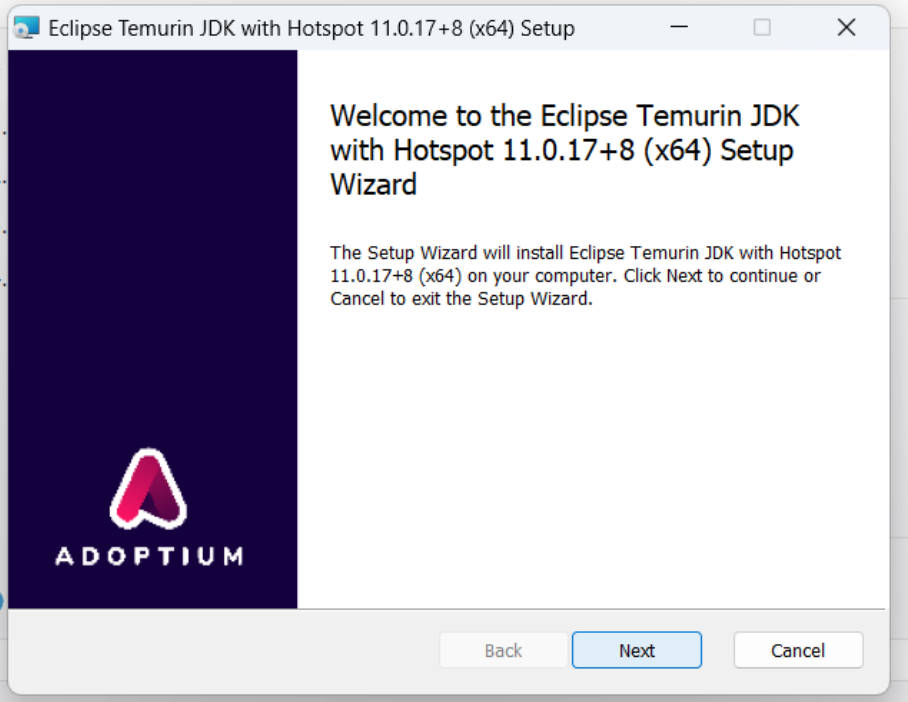
Use the drop-down boxes below to filter the list of current releases.

Operating System	Architecture	Package Type	Version
Windows	x64	JDK	11

jdk-11.0.17+8 Temurin October 25, 2022	Windows	x64	JDK - 174 MB Checksum .msi JDK - 197 MB Checksum .zip

Previous releases are available in the Temurin archive.

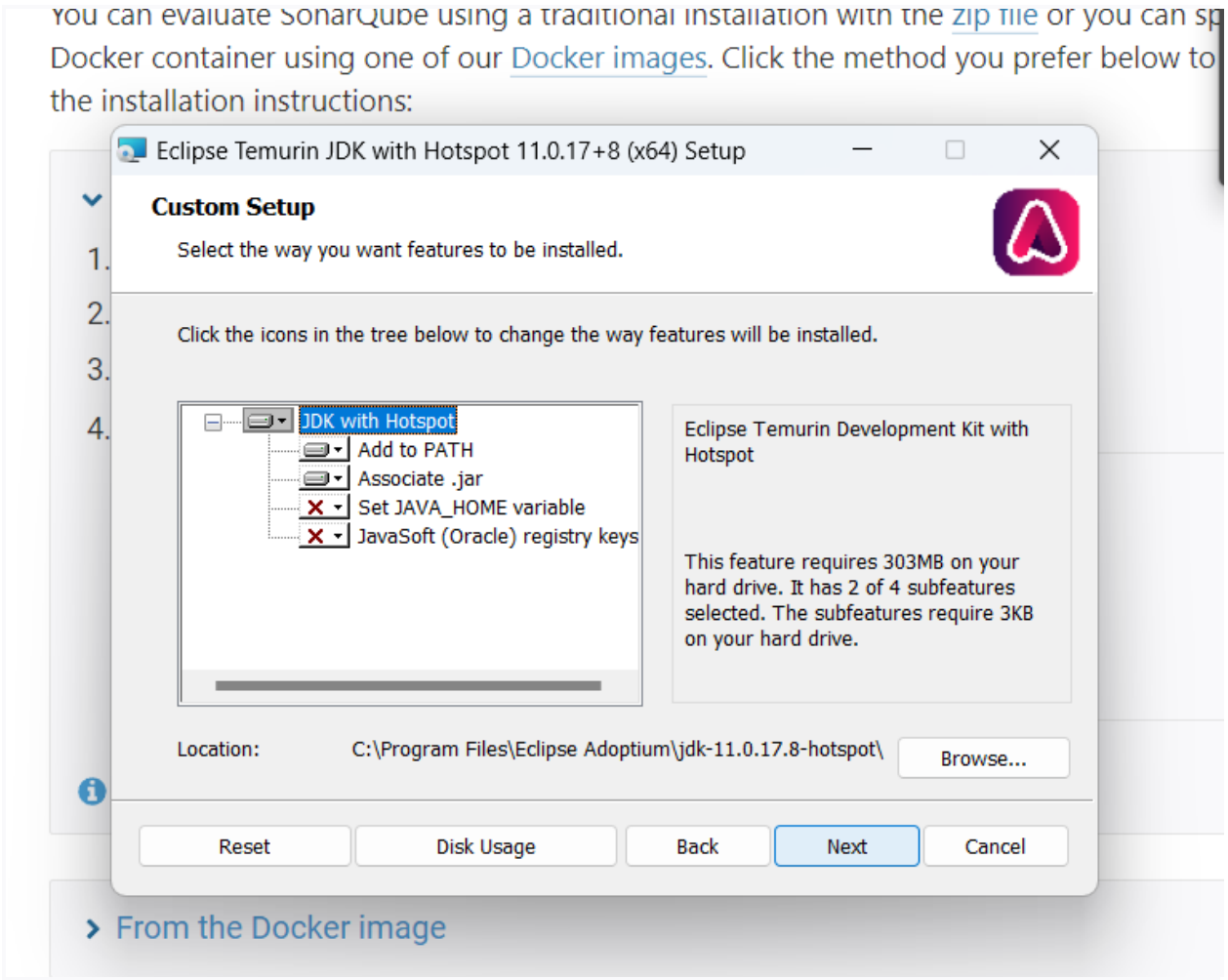
You can evaluate SonarQube using a traditional installation with the zip file or you can sp
Docker container using one of our Docker images. Click the method you prefer below to
the installation instructions:



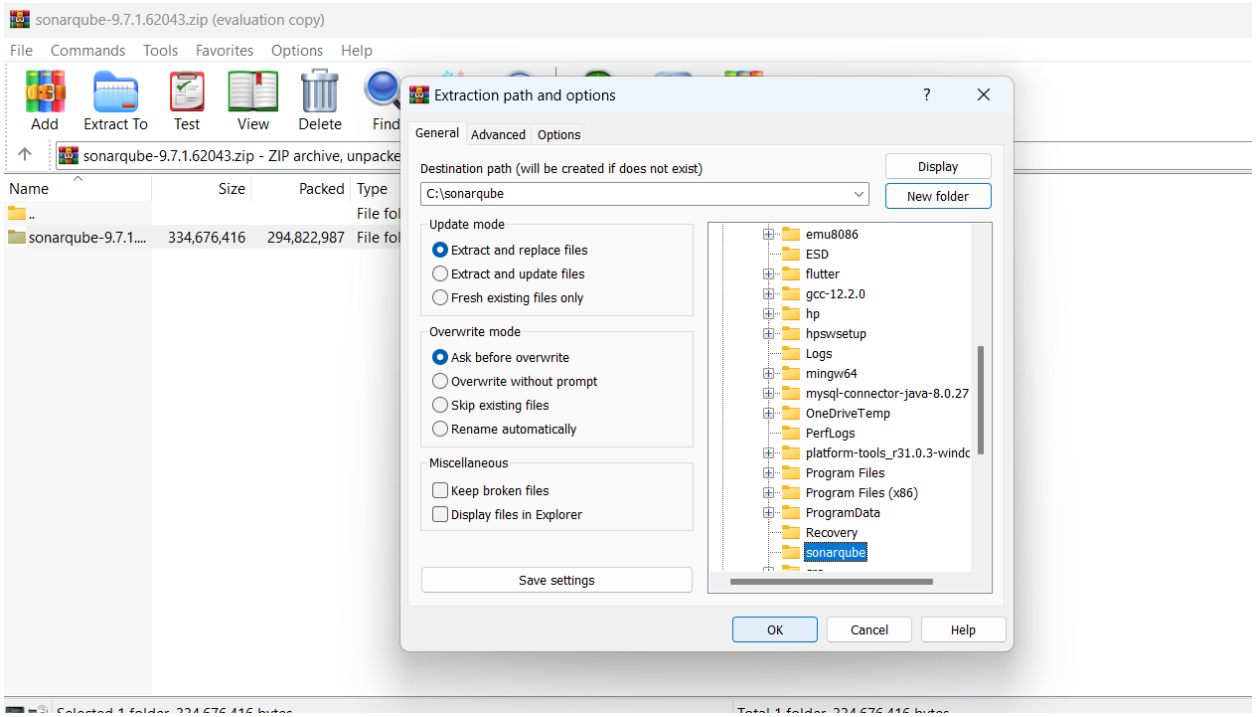
1.
2.
3.
4.

From the Docker image

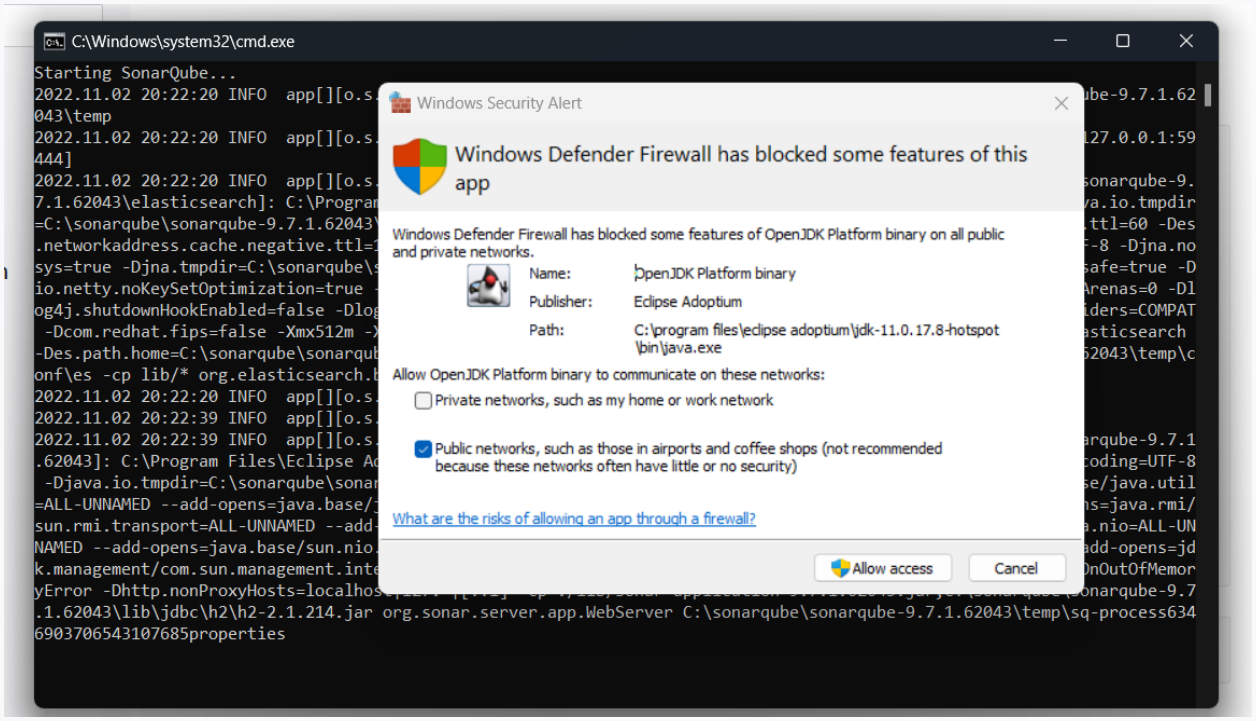
Once your instance is up and running, Log in to http://localhost:9000 using System Admini



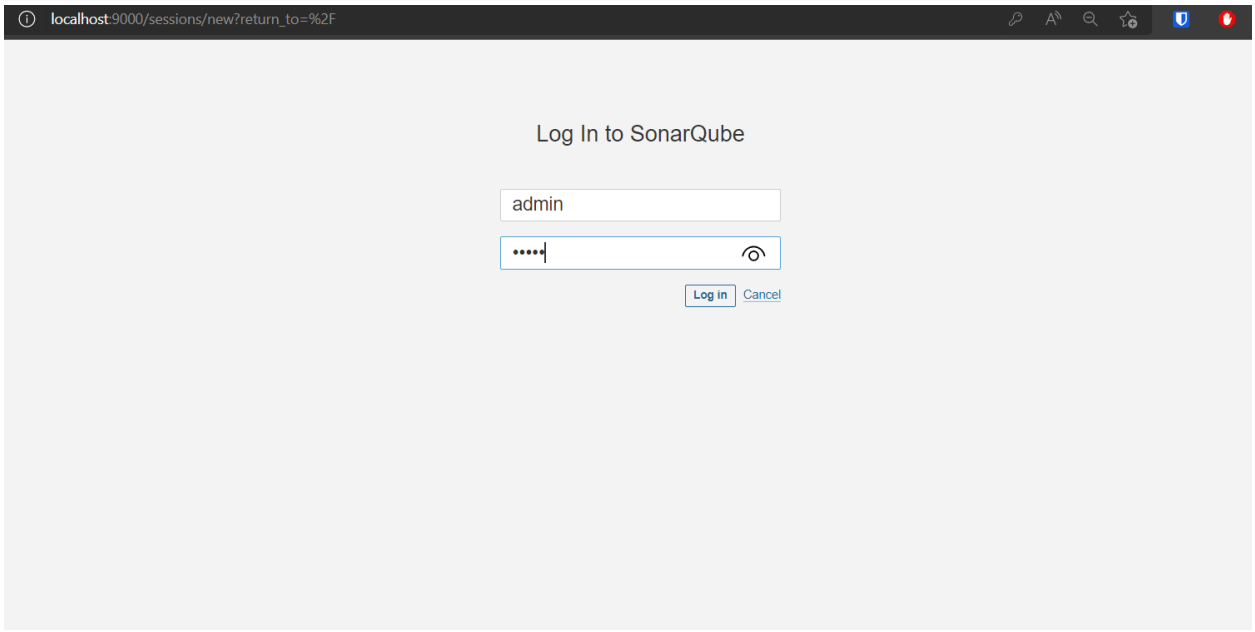
3. As a **non-`root` user**, unzip it, let's say in *C:\sonarqube* or */opt/sonarqube*.



4. As a **non-`root` user**, start the SonarQube Server:

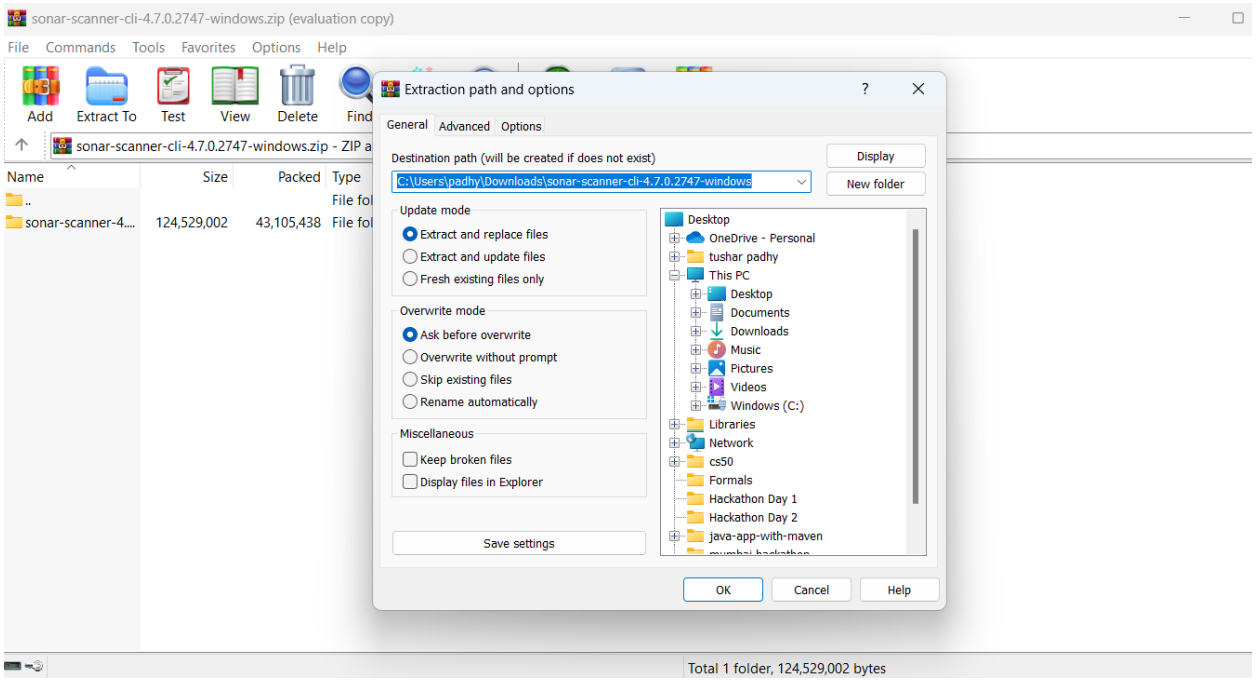


5. Once your instance is up and running, Log in to http://localhost:9000 using System Administrator credentials:

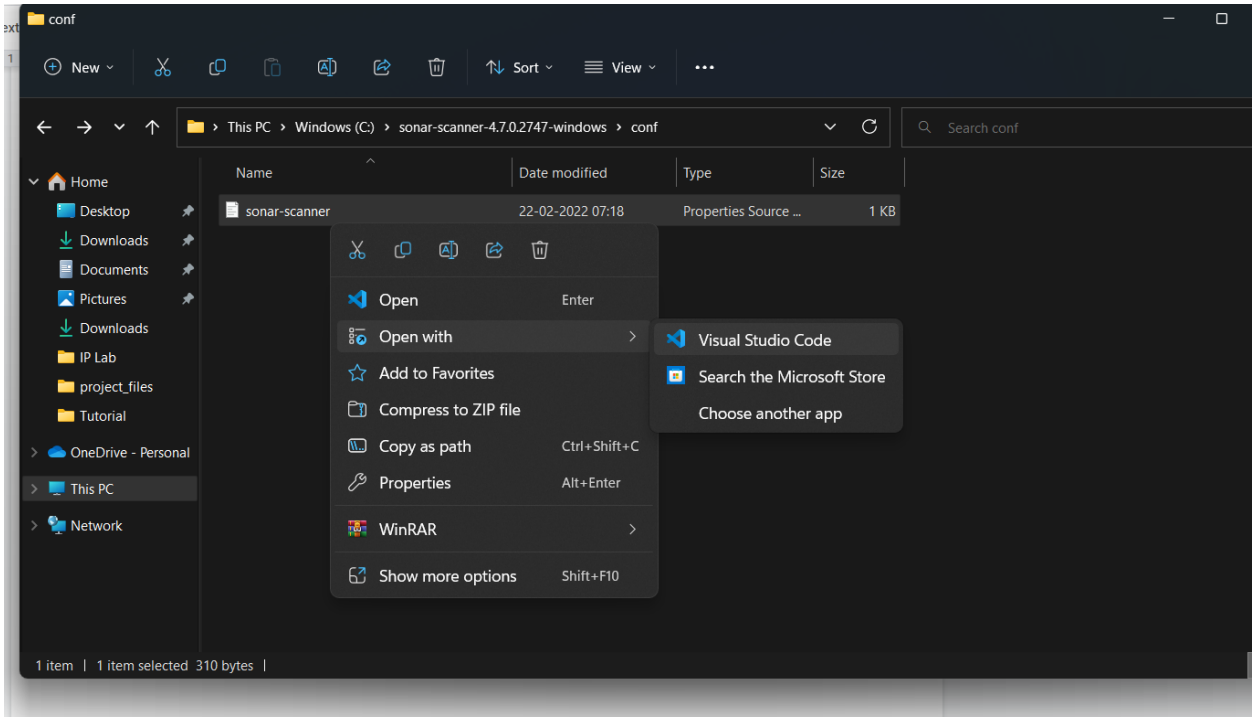


6. Install sonar scanner from the official link and extract it to the same folder where sonarqube has been installed

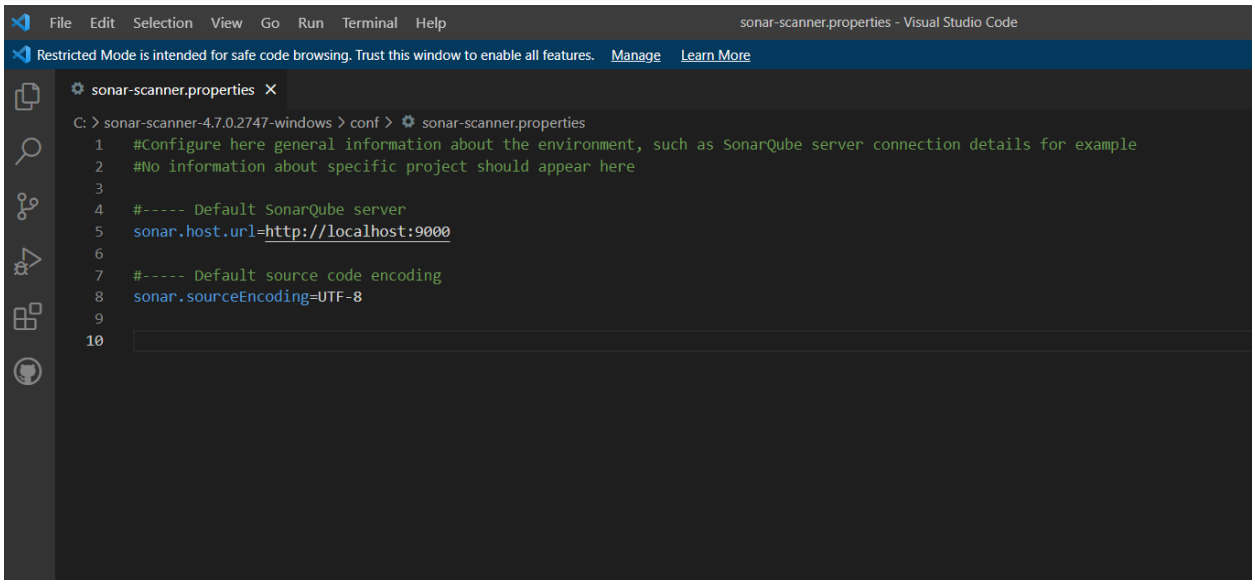
https://docs.sonarqube.org/latest/analysis/scan/sonarscanner



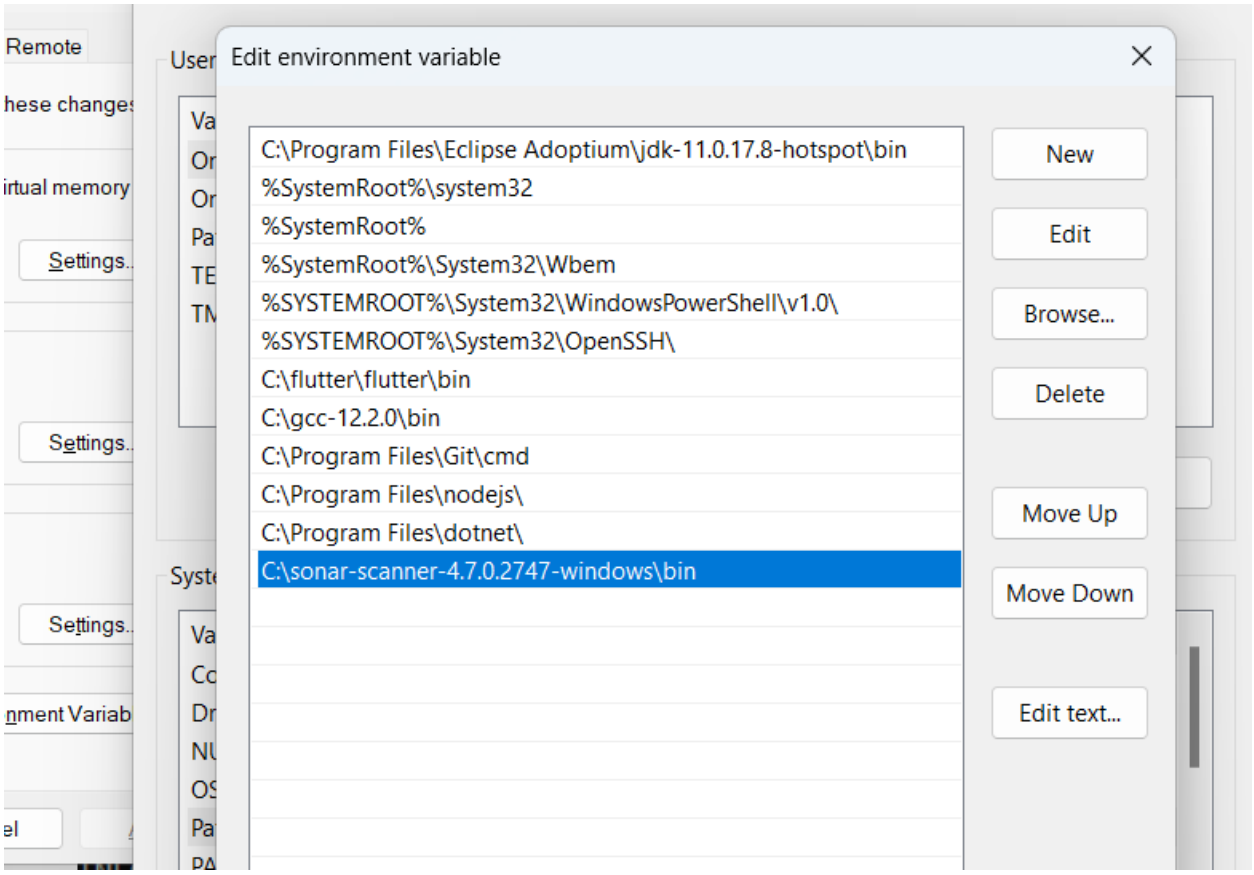
7. Go into the sonar-scanner directory and open the sonar-scanner.properties file



8. Update the global settings to point to your SonarQube server by editing



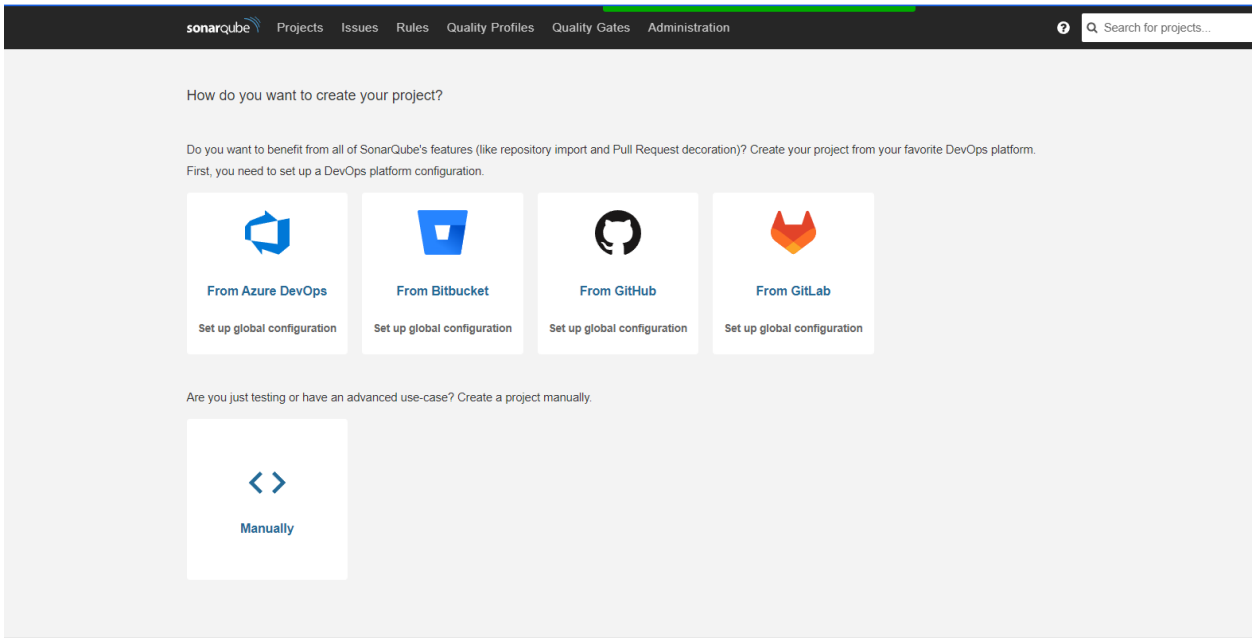
9. Add the bin path of sonarscanner into the environment variables of your system



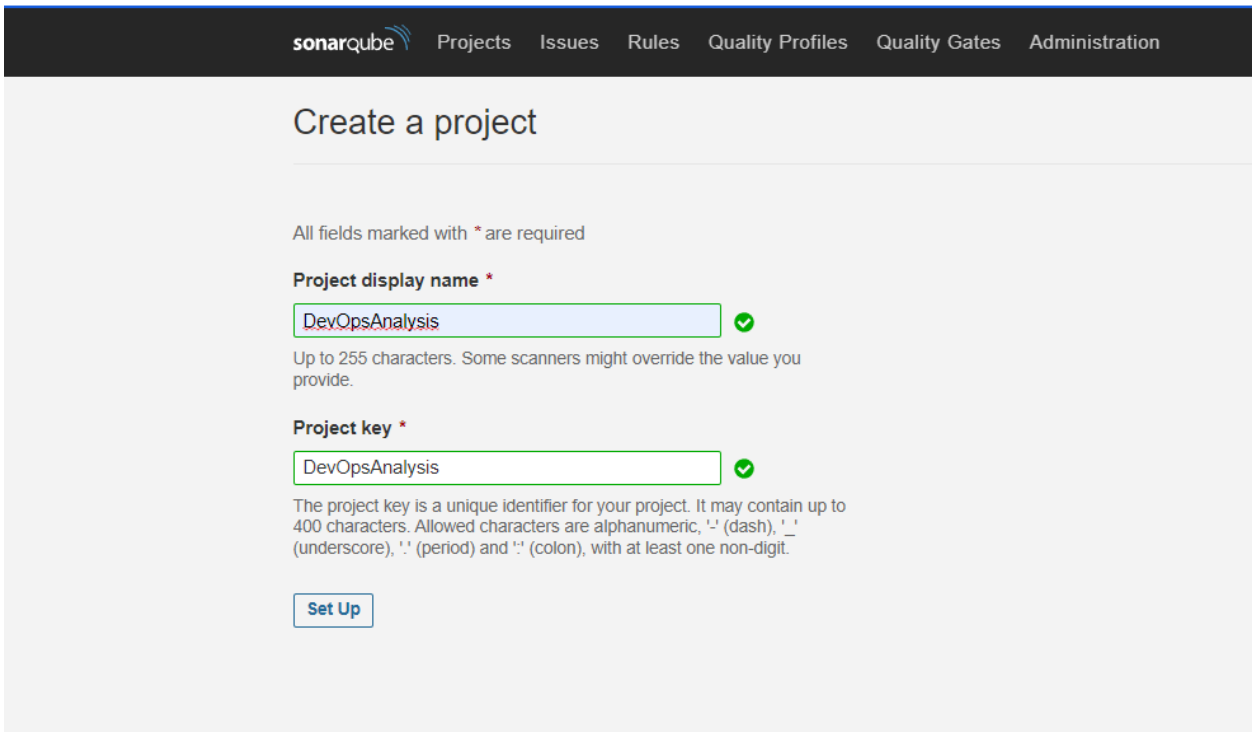
10. Verify your installation by opening a new shell and executing the command `sonar-scanner -h`

```
C:\Users\padhy>sonar-scanner -h
INFO:
INFO: usage: sonar-scanner [options]
INFO:
INFO: Options:
INFO:  -D,--define <arg>     Define property
INFO:  -h,--help             Display help information
INFO:  -v,--version          Display version information
INFO:  -X,--debug            Produce execution debug output

C:\Users\padhy>
```

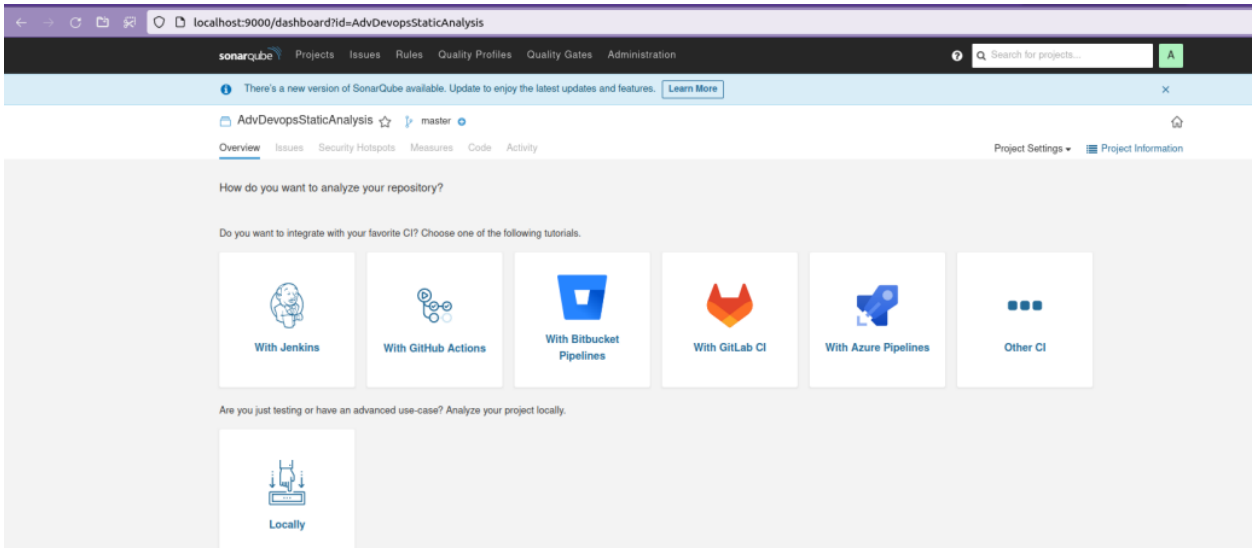
11. Now that our sonar-scanner is installed, we will create a project in our sonarqube that is logged in through http://localhost:9000/projects/create



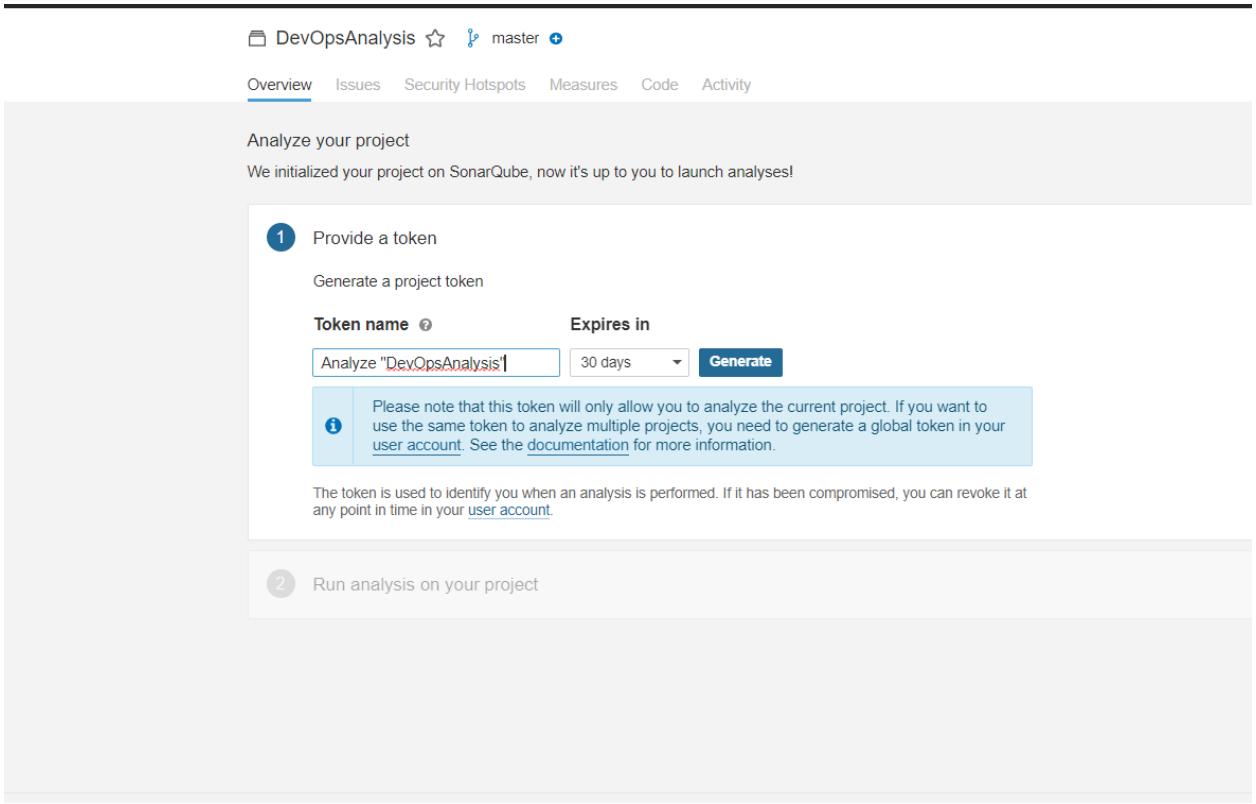
12. Click on manually and create the project by giving in the values for the asked fields



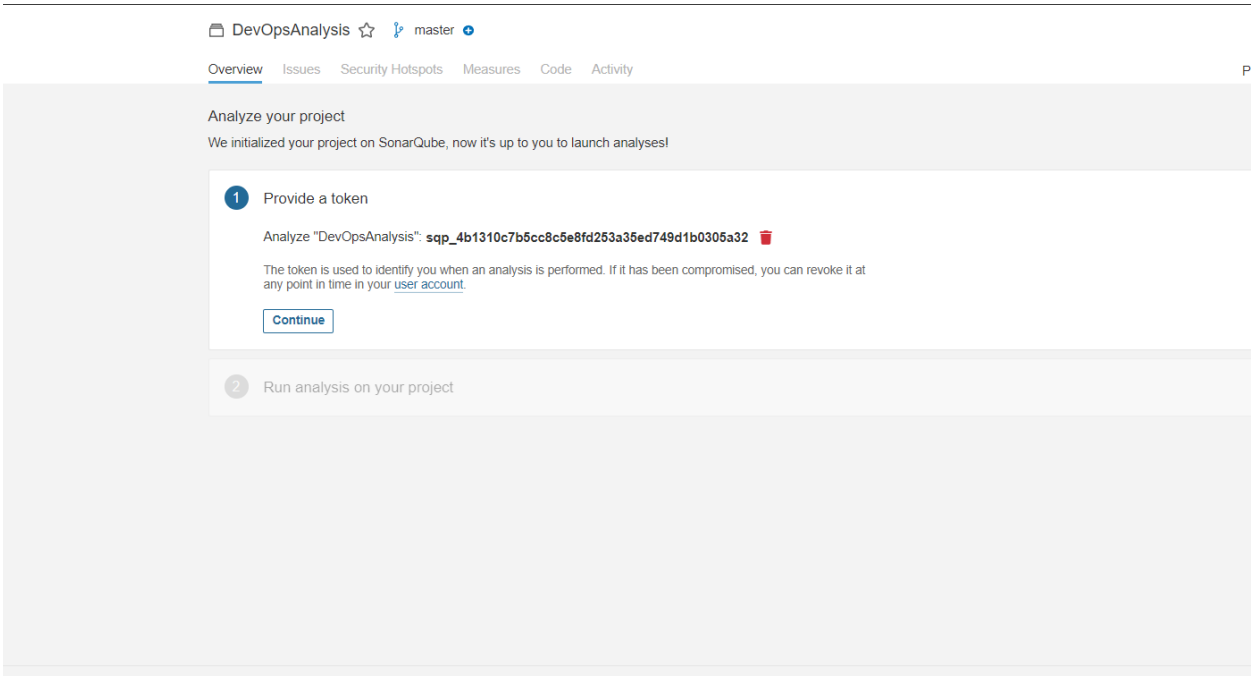
13. It will look something like this after clicking set up, now in order to analyze the code, click on locally



14. Click on generate a token

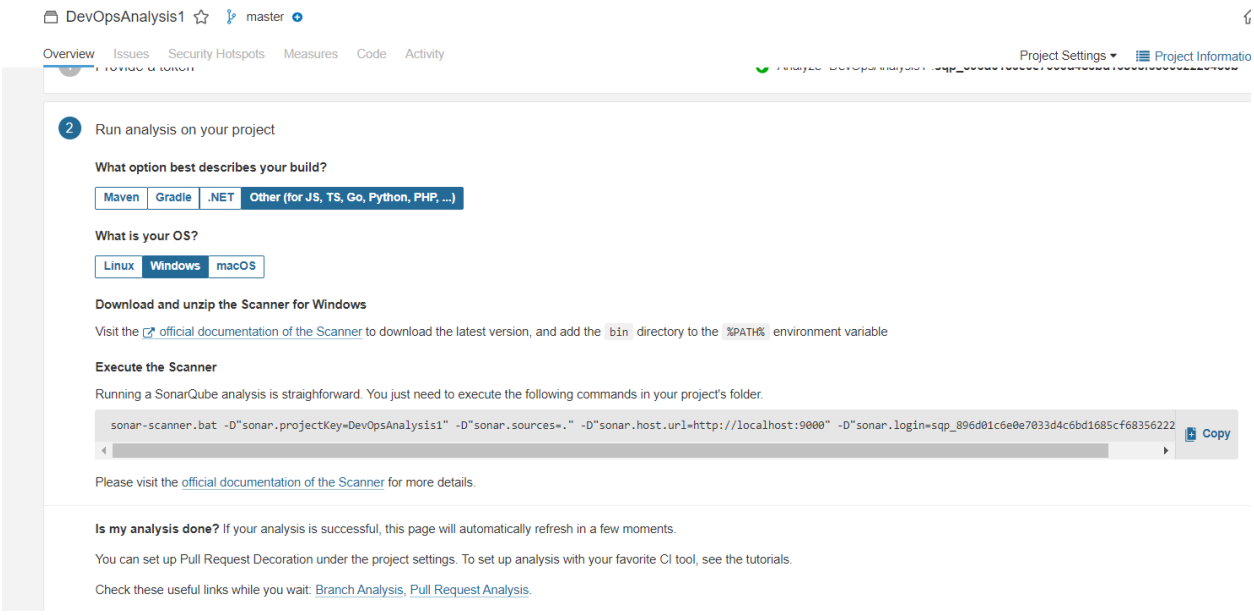


The following output will be received



15. After the token is copied and clicked on continue, then choose your project type

In this case it falls under others



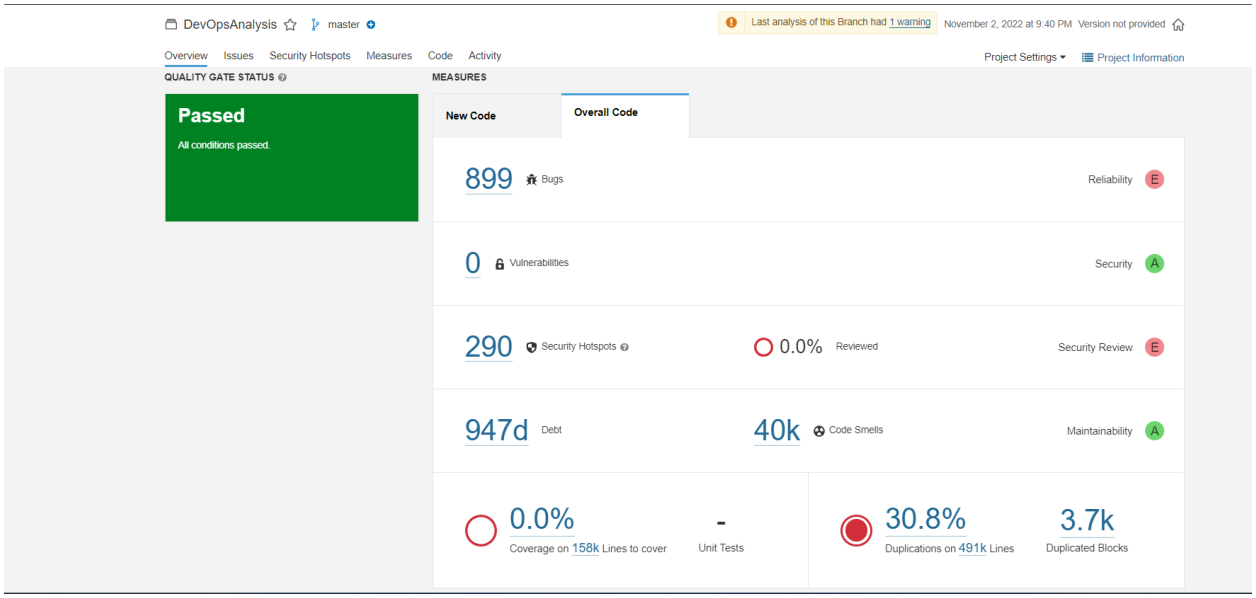
16. Now go the root of your project directory and run the command given by the sonar scanner

```
PS C:\Users\padhy\OneDrive\Documents\vs_code\CluedIn_web> sonar-scanner.bat -D"sonar.projectKey=DevOpsAnalysis" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=sqp_e98edb03ecfb6203979e61db4ba07e15c3830e0a"
INFO: Scanner configuration file: C:\sonar-scanner-4.7.0.2747-windows\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.7.0.2747
INFO: Java 11.0.14.1 Eclipse Adoptium (64-bit)
INFO: Windows 11 10.0 amd64
INFO: User cache: C:\Users\padhy\.sonar\cache
```

Once the scanning process is done, you'll see the following output

```
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AYQ5Jz5ZAcz65iYn77xT
INFO: Analysis total time: 13:12.190 s
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 13:20.389s
INFO: Final Memory: 17M/67M
INFO: ------------------------------------------------------------------------
PS C:\Users\padhy\OneDrive\Documents\vs_code\CluedIn_web>
```

Now we can see the results in our sonarqube dashboard under our project


DevOpsAnalysis
master
Last analysis of this Branch had 1 warning
November 2, 2022 at 9:40 PM
Version not provided
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
QUALITY GATE STATUS
Passed
All conditions passed.
MEASURES
New Code
Overall Code
899
Bugs
Reliability
E
0
Vulnerabilities
Security
A
290
Security Hotspots
0.0%
Reviewed
Security Review
E
947d
Debt
40k
Code Smells
Maintainability
A
0.0%
Coverage on 158k Lines to cover
-
Unit Tests
30.8%
Duplications on 491k Lines
3.7k
Duplicated Blocks

ADVANCE DEVOPS LAB
Name : Tushar Padhy
Roll no: 42
Exp - 09

Aim : To Understand Continuous monitoring and Installation and configuration of Nagios Core, Nagios Plugins and NRPE (Nagios Remote Plugin Executor) on Linux Machine.

Theory : Continuous monitoring is a process of constant detecting, reporting, and
responding to risks and events within an IT system.
This process is a vital DevOps security practice and has multiple goals:

- Provide real-time insight into system performance.
- Offer feedback on the overall health and security of IT infrastructure.
- Enhance visibility across IT operations and the DevOps pipeline.
- Identify the cause of incidents and apply mitigation before the problem results in downtime or a data breach

Output :

STEP 1 : Install Prerequisite Packages

```
tushar@maverick:~$ mkdir nagisos
tushar@maverick:~$ cd nagisos/
tushar@maverick:~/nagisos$ sudo wget https://assets.nagios.com/downloads/nagiosc
ore/releases/nagios-4.4.6.tar.gz
--2022-10-07 00:51:53--  https://assets.nagios.com/downloads/nagioscore/releases
/nagios-4.4.6.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 45.79.49.120, 2600:3c00::f03c
:92ff:fef7:45ce
Connecting to assets.nagios.com (assets.nagios.com)|45.79.49.120|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 11333414 (11M) [application/x-gzip]
Saving to: ‘nagios-4.4.6.tar.gz’

nagios-4.4.6.tar.gz 100%[===================>]  10.81M  1.45MB/s    in 25s

2022-10-07 00:52:20 (441 KB/s) - ‘nagios-4.4.6.tar.gz’ saved [11333414/11333414]

tushar@maverick:~/nagisos$
```

STEP 2 : Install Nagios core

```
tushar@maverick:~/nagisos$ sudo tar -xvf nagios-4.4.6.tar.gz
nagios-4.4.6/
nagios-4.4.6/.gitignore
nagios-4.4.6/.travis.yml
nagios-4.4.6/CONTRIBUTING.md
nagios-4.4.6/Changelog
nagios-4.4.6/INSTALLING
nagios-4.4.6/LEGAL
nagios-4.4.6/LICENSE
nagios-4.4.6/Makefile.in
nagios-4.4.6/README.md
nagios-4.4.6/THANKS
nagios-4.4.6/UPGRADING
nagios-4.4.6/aclocal.m4
nagios-4.4.6/autoconf-macros/
nagios-4.4.6/autoconf-macros/.gitignore
nagios-4.4.6/autoconf-macros/CHANGELOG.md
nagios-4.4.6/autoconf-macros/LICENSE
nagios-4.4.6/autoconf-macros/LICENSE.md
nagios-4.4.6/autoconf-macros/README.md
nagios-4.4.6/autoconf-macros/add_group_user
nagios-4.4.6/autoconf-macros/ax_nagios_get_distrib
```

```
tushar@maverick:~/nagisos$ cd nagios-4.4.6/
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo ./configure --with-httpd-conf=/etc/
apache2/sites-enabled
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether make sets $(MAKE)... yes
checking whether ln -s works... yes
checking for strip... /usr/bin/strip
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for ANSI C header files... yes
```

```
*** Configuration summary for nagios 4.4.6 2020-04-28 ***:

 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagios
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  /run/nagios.lock
   Check result directory:  /usr/local/nagios/var/spool/checkresults
           Init directory:  /lib/systemd/system
  Apache conf.d directory:  /etc/apache2/sites-enabled
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll

 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):


Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

STEP 3: Run the make all command to compile the program alongside the CGIs:

```
Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.

tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make all
cd ./base && make
make[1]: Entering directory '/home/tushar/nagisos/nagios-4.4.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
nagios.c: In function ‘main’:
nagios.c:611:25: warning: ignoring return value of ‘asprintf’ declared with attr
ibute ‘warn_unused_result’ [-Wunused-result]
  611 |                         asprintf(&mac->x[MACRO_PROCESSSTARTTIME], "%llu"
, (unsigned long long)program_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c:841:25: warning: ignoring return value of ‘asprintf’ declared with attr
ibute ‘warn_unused_result’ [-Wunused-result]
  841 |                         asprintf(&mac->x[MACRO_EVENTSTARTTIME], "%llu",
(unsigned long long)event_start);
      |                         ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c: In function ‘nagios_core_worker’:
nagios.c:176:17: warning: ignoring return value of ‘read’ declared with attribut
e ‘warn_unused_result’ [-Wunused-result]
  176 |                 read(sd, response + 3, sizeof(response) - 4);
      |                 ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
nagios.c: In function ‘test_path_access’:
nagios.c:122:17: warning: ignoring return value of ‘asprintf’ declared with attr
ibute ‘warn_unused_result’ [-Wunused-result]
```

STEP 4: Create group users and install sample config files

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make install-groups-users
groupadd -r nagios
useradd -g nagios nagios
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo usermod -a -G nagios www-data
tushar@maverick:~/nagisos/nagios-4.4.6$
```

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make install
cd ./base && make install
make[1]: Entering directory '/home/tushar/nagisos/nagios-4.4.6/base'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagios /usr/local/nagios/bin
/usr/bin/install -c -s -m 774 -o nagios -g nagios nagiostats /usr/local/nagios/b
in
make[1]: Leaving directory '/home/tushar/nagisos/nagios-4.4.6/base'
cd ./cgi && make install
make[1]: Entering directory '/home/tushar/nagisos/nagios-4.4.6/cgi'
make install-basic
make[2]: Entering directory '/home/tushar/nagisos/nagios-4.4.6/cgi'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/sbin
for file in *.cgi; do \
        /usr/bin/install -c -s -m 775 -o nagios -g nagios $file /usr/local/nagio
s/sbin; \
done
make[2]: Leaving directory '/home/tushar/nagisos/nagios-4.4.6/cgi'
make[1]: Leaving directory '/home/tushar/nagisos/nagios-4.4.6/cgi'
```

```
make[1]: Leaving directory '/home/tushar/nagisos/nagios-4.4.6'
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make install-init
/usr/bin/install -c -m 755 -d -o root -g root /lib/systemd/system
/usr/bin/install -c -m 755 -o root -g root startup/default-service /lib/systemd/
system/nagios.service
tushar@maverick:~/nagisos/nagios-4.4.6$
```

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***
```

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc/objects
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/nagios.cfg /usr/
local/nagios/etc/nagios.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/cgi.cfg /usr/loc
al/nagios/etc/cgi.cfg
/usr/bin/install -c -b -m 660 -o nagios -g nagios sample-config/resource.cfg /us
r/local/nagios/etc/resource.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template-object/
templates.cfg /usr/local/nagios/etc/objects/templates.cfg
```

STEP 5: Set up Apache and Nagios UI

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-enabled/n
agios.conf
if [ 0 -eq 1 ]; then \
        ln -s /etc/apache2/sites-enabled/nagios.conf /etc/apache2/sites-enabled/
nagios.conf; \
fi

*** Nagios/Apache conf file installed ***
```

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo a2enmod rewrite cgi
Enabling module rewrite.
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
tushar@maverick:~/nagisos/nagios-4.4.6$
```

STEP 6: Create nagios user and set password

```
tushar@maverick:~/nagisos/nagios-4.4.6$ sudo htpasswd -c /usr/local/nagios/etc/h
tpasswd.users nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
tushar@maverick:~/nagisos/nagios-4.4.6$
```

Step 7: Install nagios plugin

```
tushar@maverick:~/nagisos/nagios-4.4.6$ cd ..
tushar@maverick:~/nagisos$ sudo apt install monitoring-plugins nagios-nrpe-plugi
n -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdbi1 libmysqlclient21 libradcli4 liburiparser1 monitoring-plugins-basic
  monitoring-plugins-common monitoring-plugins-standard mysql-common
  python3-gpg python3-samba python3-tdb rpcbind samba-common samba-common-bin
  samba-dsdb-modules smbclient
Suggested packages:
  icinga2 nagios-plugins-contrib fping postfix | sendmail-bin
  | exim4-daemon-heavy | exim4-daemon-light qstat heimdal-clients
  python3-markdown python3-dnspython cifs-utils
The following NEW packages will be installed:
  libdbi1 libmysqlclient21 libradcli4 liburiparser1 monitoring-plugins
  monitoring-plugins-basic monitoring-plugins-common
```

```
tushar@maverick:/usr/local/nagios/etc$ sudo vim nagios.cfg
[sudo] password for tushar:
sudo: vim: command not found
tushar@maverick:/usr/local/nagios/etc$ sudo nano nagios.cfg
tushar@maverick:/usr/local/nagios/etc$
```

```
  GNU nano 6.2                          nagios.cfg *
# directive as shown below:

cfg_dir=/usr/local/nagios/etc/servers
cfg_dir=/usr/local/nagios/etc/printers
cfg_dir=/usr/local/nagios/etc/switches
cfg_dir=/usr/local/nagios/etc/routers




# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts.  The CGIs read object definitions from
# this cache file (rather than looking at the object config files
# directly) in order to prevent inconsistencies that can occur
# when the config files are modified after Nagios starts.

Save modified buffer?
 Y Yes
 N No           ^C Cancel
```

```
tushar@maverick:/usr/local/nagios/etc$ sudo nano nagios.cfg
tushar@maverick:/usr/local/nagios/etc$ sudo mkdir servers printers switches rout
ers
tushar@maverick:/usr/local/nagios/etc$ sudo nano resource.cfg
tushar@maverick:/usr/local/nagios/etc$
```

```
  GNU nano 6.2                        resource.cfg *
# event handlers - if you decide to move the plugins or event handlers to
# a different directory in the future, you can just update one or two
# $USERx$ macros, instead of modifying a lot of command definitions.
#
# The CGIs will not attempt to read the contents of resource files, so
# you can set restrictive permissions (600 or 660) on them.
#
# Nagios supports up to 256 $USERx$ macros ($USER1$ through $USER256$)
#
# Resource files may also be used to store configuration directives for
# external data sources like MySQL...
#
###############################################################################

# Sets $USER1$ to be the path to the plugins
#$USER1$=/usr/local/nagios/libexec


^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
# Nagios supports up to 256 $USERx$ macros ($USER1$ through $USER256$)
#
# Resource files may also be used to store configuration directives for
# external data sources like MySQL...
#
##############################################################################

$USER1$=/usr/lib/nagios/plugins
# Sets $USER1$ to be the path to the plugins
#$USER1$=/usr/local/nagios/libexec

# Sets $USER2$ to be the path to event handlers
#$USER2$=/usr/local/nagios/libexec/eventhandlers

# Store some usernames and passwords (hidden from the CGIs)
#$USER3$=someuser

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
tushar@maverick:/usr/local/nagios/etc$ cd objects/
tushar@maverick:/usr/local/nagios/etc/objects$ sudo nano contacts.cfg
```

```
  GNU nano 6.2                        contacts.cfg *
# This contact definition inherits a lot of default values from the
# 'generic-contact' template which is defined elsewhere.

define contact {

    contact_name            nagiosadmin             ; Short name of user
    use                     generic-contact         ; Inherit default values fr>
    alias                   Nagios Admin            ; Full name of user
    email                   padhytushar4303@gmail.com ; <<***** CHANGE THIS TO >
}



##############################################################################
#
# CONTACT GROUPS
#

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

```
tushar@maverick:/usr/local/nagios/etc/objects$ sudo nano commands.cfg
tushar@maverick:/usr/local/nagios/etc/objects$
```

```
    command_name    process-host-perfdata
    command_line    /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTA>
}



define command {

    command_name    process-service-perfdata
    command_line    /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERV>
}


define command{
        command_name check_nrpe
        command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Step 8 : Start the Nagios Daemon and check status

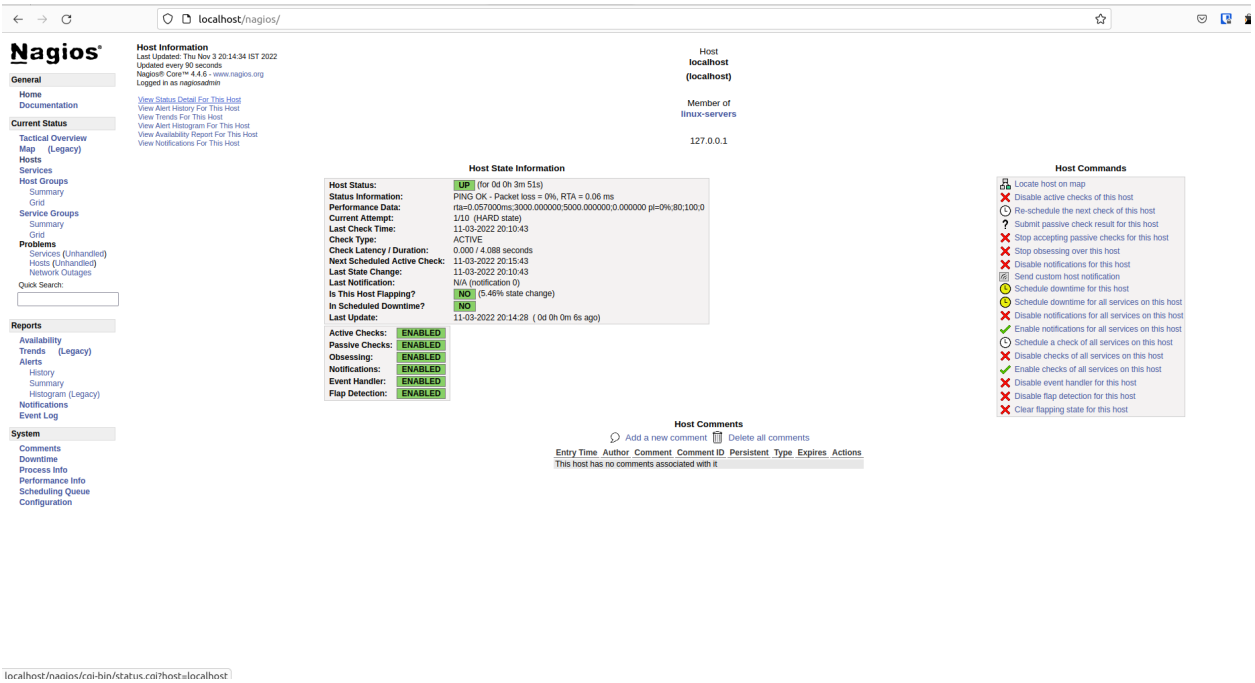
```
tushar@maverick:~$ sudo systemctl restart apache2
tushar@maverick:~$
```

```
tushar@maverick:~$ sudo systemctl restart nagios
tushar@maverick:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
tushar@maverick:~$ sudo systemctl enable nagios
Created symlink /etc/systemd/system/multi-user.target.wants/nagios.service → /li
b/systemd/system/nagios.service.
tushar@maverick:~$
```

```
tushar@maverick:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Fri 2022-10-07 01:56:24 IST; 1min 58s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 21471 (apache2)
      Tasks: 6 (limit: 8632)
     Memory: 10.1M
        CPU: 51ms
     CGroup: /system.slice/apache2.service
             ├─21471 /usr/sbin/apache2 -k start
             ├─21472 /usr/sbin/apache2 -k start
             ├─21473 /usr/sbin/apache2 -k start
             ├─21474 /usr/sbin/apache2 -k start
             ├─21475 /usr/sbin/apache2 -k start
             └─21476 /usr/sbin/apache2 -k start

Oct 07 01:56:24 maverick systemd[1]: Starting The Apache HTTP Server...
Oct 07 01:56:24 maverick apachectl[21470]: AH00558: apache2: Could not reliably>
Oct 07 01:56:24 maverick systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

```
● nagios.service - Nagios Core 4.4.6
     Loaded: loaded (/lib/systemd/system/nagios.service; enabled; vendor preset>
     Active: active (running) since Fri 2022-10-07 01:56:57 IST; 2min 4s ago
       Docs: https://www.nagios.org/documentation
   Main PID: 21508 (nagios)
      Tasks: 14 (limit: 8632)
     Memory: 4.3M
        CPU: 77ms
     CGroup: /system.slice/nagios.service
             ├─21508 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagi>
             ├─21509 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21510 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21511 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21512 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21513 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21514 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21515 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21516 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21517 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
             ├─21518 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/va>
lines 1-20
```

Access the tool by opening the browser and navigating to the http://server-IP/nagios URL.


localhost/nagios/
Nagios
General
Home
Documentation
Current Status
Tactical Overview
Map (Legacy)
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services (Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:
Reports
Availability
Trends (Legacy)
Alerts
History
Summary
Histogram (Legacy)
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration
Host Information
Last Updated: Thu Nov 3 20:14:34 IST 2022
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin
View Status Detail For This Host
View Alert History For This Host
View Trends For This Host
View Alert Histogram For This Host
View Availability Report For This Host
View Notifications For This Host
Host
localhost
(localhost)
Member of
linux-servers
127.0.0.1
Host State Information
Host Status: UP (for 0d 0h 3m 51s)
Status Information: PING OK - Packet loss = 0%, RTA = 0.06 ms
Performance Data: rta=0.057000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt: 1/10 (HARD state)
Last Check Time: 11-03-2022 20:10:43
Check Type: ACTIVE
Check Latency / Duration: 0.000 / 4.088 seconds
Next Scheduled Active Check: 11-03-2022 20:15:43
Last State Change: 11-03-2022 20:10:43
Last Notification: N/A (notification 0)
Is This Host Flapping? NO (5.46% state change)
In Scheduled Downtime? NO
Last Update: 11-03-2022 20:14:28 (0d 0h 0m 6s ago)
Active Checks: ENABLED
Passive Checks: ENABLED
Obsessing: ENABLED
Notifications: ENABLED
Event Handler: ENABLED
Flap Detection: ENABLED
Host Commands
Locate host on map
Disable active checks of this host
Re-schedule the next check of this host
Submit passive check result for this host
Stop accepting passive checks for this host
Stop obsessing over this host
Disable notifications for this host
Send custom host notification
Schedule downtime for this host
Schedule downtime for all services on this host
Disable notifications for all services on this host
Enable notifications for all services on this host
Schedule a check of all services on this host
Disable checks of all services on this host
Enable checks of all services on this host
Disable event handler for this host
Disable flap detection for this host
Clear flapping state for this host
Host Comments
Add a new comment
Delete all comments
Entry Time
Author
Comment
Comment ID
Persistent
Type
Expires
Actions
This host has no comments associated with it

ADVANCED DEVOPS LAB

Name : Tushar Padhy

Roll no : 42

Experiment No : 10

Aim:

To perform Port, Service monitoring, Windows/Linux server monitoring using Nagios.

Theory:

Nagios is an open source monitoring system for computer systems. It was designed to run on the Linux operating system and can monitor devices running Linux, Windows and Unix operating systems (OSes).

Nagios runs both agent-based and agentless configurations. Independent agents are installed on any hardware or software system to collect data that is then reported back to the management server. Agentless monitoring uses existing protocols to emulate an agent. Both approaches can monitor file system usage, OS metrics, service and process states and more.

Here are the important reasons to use Nagios monitoring tool:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitors and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

Output:

1. Install nrpe service nrpe plugins on linux host

```
tushar@tushar-HP-Laptop-14s:~$ sudo apt install nagios-nrpe-server nagios-plugins -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
monitoring-plugins is already the newest version (2.3.1-1ubuntu4).
Suggested packages:
  xinetd | inetd
The following NEW packages will be installed:
  nagios-nrpe-server
0 upgraded, 1 newly installed, 0 to remove and 189 not upgraded.
Need to get 359 kB of archives.
After this operation, 477 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 nagios-nrpe-server amd64 4.0.3-1ubuntu
59 kB]
Fetched 359 kB in 1s (343 kB/s)
Selecting previously unselected package nagios-nrpe-server.
(Reading database ... 169461 files and directories currently installed.)
Preparing to unpack .../nagios-nrpe-server_4.0.3-1ubuntu2_amd64.deb ...
Unpacking nagios-nrpe-server (4.0.3-1ubuntu2) ...
```

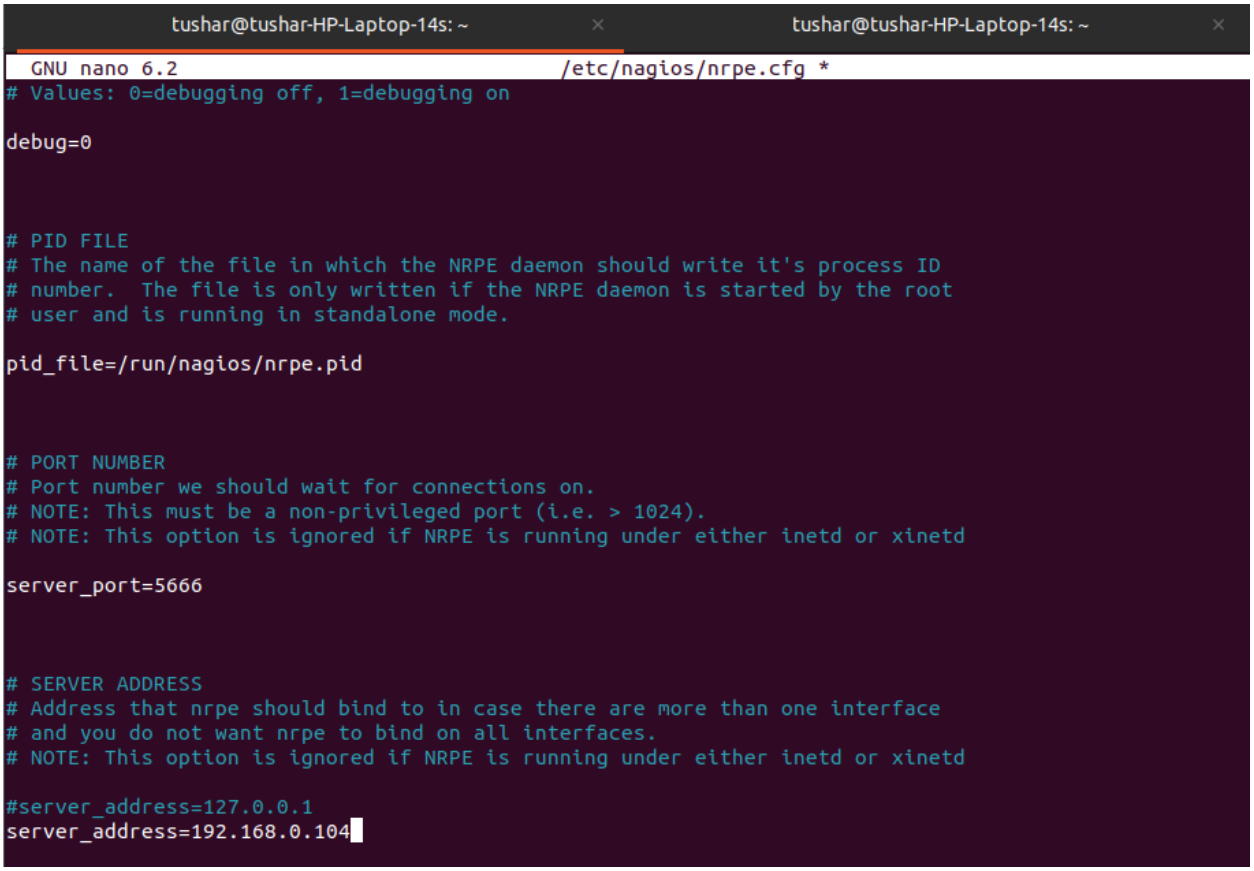
2. Configuring nrpe Plugins and services

```
tem/nagios-nrpe-server.service.
Processing triggers for man-db (2.10.2-1) ...
tushar@tushar-HP-Laptop-14s:~$ sudo nano /etc/nagios/nrpe.cfg
tushar@tushar-HP-Laptop-14s:~$
```

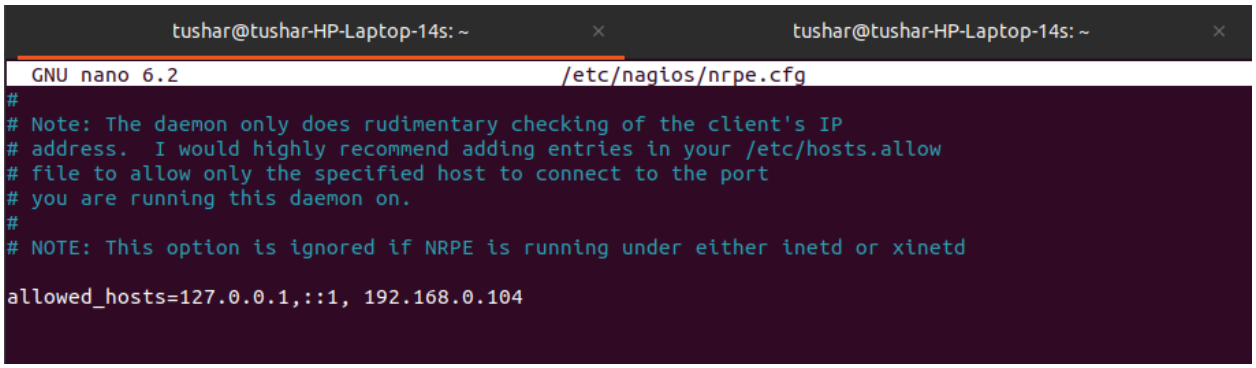
3. Add your IP address (host) as server address

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ sudo chmod 775 servers/
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ sudo chown nagios:nagios servers/
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ ll
total 168
drwxrwxr-x 7 nagios nagios  4096 Nov  3 17:15 ./
drwxr-xr-x 8 root   root    4096 Oct 28 13:36 ../
-rw-rw-r-- 1 nagios nagios 13710 Nov  3 17:06 cgi.cfg
-rw-rw-r-- 1 nagios nagios 13710 Oct 28 13:57 cgi.cfg~
-rw-r--r-- 1 root   root      50 Nov  3 17:08 htpasswd.users
-rw-rw-r-- 1 nagios nagios 45838 Nov  3 17:13 nagios.cfg
-rw-rw-r-- 1 nagios nagios 45843 Oct 28 13:57 nagios.cfg~
drwxrwxr-x 2 nagios nagios  4096 Nov  3 17:19 objects/
drwxr-xr-x 2 root   root    4096 Nov  3 17:14 printers/
-rw-rw---- 1 nagios nagios  1312 Nov  3 17:15 resource.cfg
-rw-rw---- 1 nagios nagios  1312 Oct 28 13:57 resource.cfg~
drwxr-xr-x 2 root   root    4096 Nov  3 17:14 routers/
drwxrwxr-x 2 nagios nagios  4096 Nov  4 00:16 servers/
drwxr-xr-x 2 root   root    4096 Nov  3 17:14 switches/
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$
```

4. Edit /etc/nagios/nrpe.cfg file to configure the nrpe project



5. Add the host IP address in allowed_hosts



6. Restart and enable nagios nrpe server to make the changes active

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ sudo systemctl restart nagios-nrpe
-server
[sudo] password for tushar:
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ sudo systemctl enable nagios-nrpe-
server
Synchronizing state of nagios-nrpe-server.service with SysV service script with /lib/
systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nagios-nrpe-server
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$
```

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ cd servers
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ touch tushar-HP-Laptop-14s.cfg
touch: cannot touch 'tushar-HP-Laptop-14s.cfg': Permission denied
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ sudo touch tushar-HP-Laptop-14s.cfg
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$
```

7. Change permission on servers folders

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ sudo chmod 775 servers
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$ sudo chown nagios:nagios servers/
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc$
```

8. Go to servers directory and create a new config file with "hostname".cfg to define linux host and perform sudo nano to make changes

   In this case it will be tushar-HP-laptop-14s.cfg

```
tushar@tushar-HP-Laptop-14s: /usr/local/nagios/etc/servers
  GNU nano 6.2                          tushar-HP-Laptop-14s.cfg
#Replace :
# host_name = Your-Hostname
# alias = Your-Alias
# address = Your-IP address of host

define host{
use                     linux-server
host_name               tushar-HP-Laptop-14s
alias                   SlaveDNS
address                 192.168.0.104
}

define service{
use                             local-service
host_name                       tushar-HP-Laptop-14s
service_description             Root / Partition
check_command                   check_nrpe!check_disk
}

define service{
use                             local-service
host_name                       tushar-HP-Laptop-14s
```

9. Change permission on servers folder

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ sudo chown nagios:nagios tushar-HP-Laptop-14s.cfg
[sudo] password for tushar:
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ sudo chmod 664 tushar-HP-Laptop-14s.cfg
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$
```

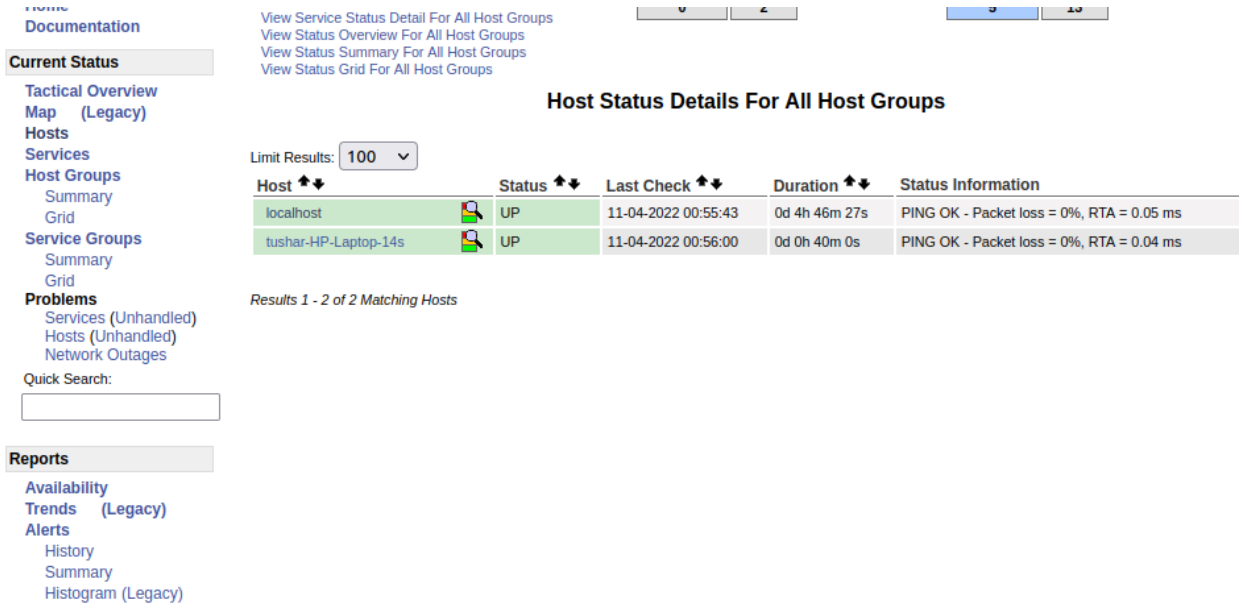
10. Verify nagios configuration file for any errors in the nagios server

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ sudo systemctl restart nagios
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios
/etc/nagios.cfg
[sudo] password for tushar:

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL

Website: https://www.nagios.org
Reading configuration data...
   Read main config file okay...
   Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
        Checked 13 services.
        Checked 2 hosts.
        Checked 1 host groups.
        Checked 0 service groups.
        Checked 1 contacts.
        Checked 1 contact groups.
        Checked 25 commands.
        Checked 5 time periods.
        Checked 0 host escalations.
        Checked 0 service escalations.
Checking for circular paths...
        Checked 2 hosts
        Checked 0 service dependencies
        Checked 0 host dependencies
        Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors:   0
```

11. Restart Nagios

```
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$ sudo systemctl restart nagios
tushar@tushar-HP-Laptop-14s:/usr/local/nagios/etc/servers$
```

12. The host has been added as seen below



Conclusion:

Thus port monitoring is completed successfully.

Reference:

https://www.youtube.com/watch?v=UMHgRnPXoEw

https://www.youtube.com/watch?v=6T_RCywnLB8&list=PLvoGk4CSiH0sfH21iVgaPKx3DlI8dJYte&index=4

ADVANCED DEVOPS LAB

Name : Tushar Padhy

Roll no : 42

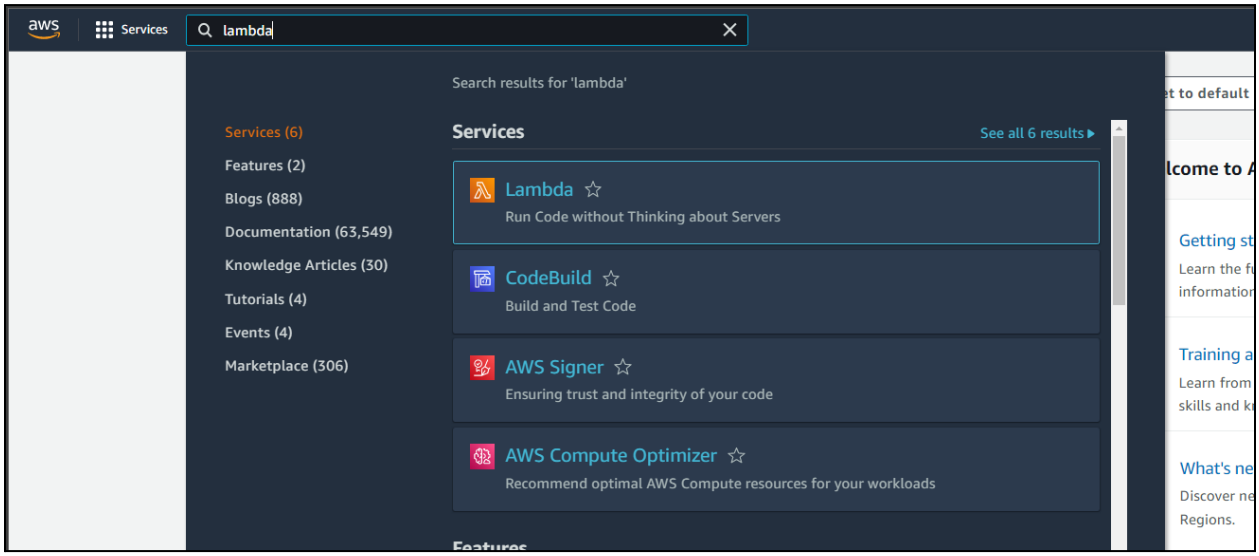
Experiment No : 11

Aim:

To understand AWS Lambda, its workflow, and various functions and create your first Lambda functions using Python / Java / Nodejs

Output:

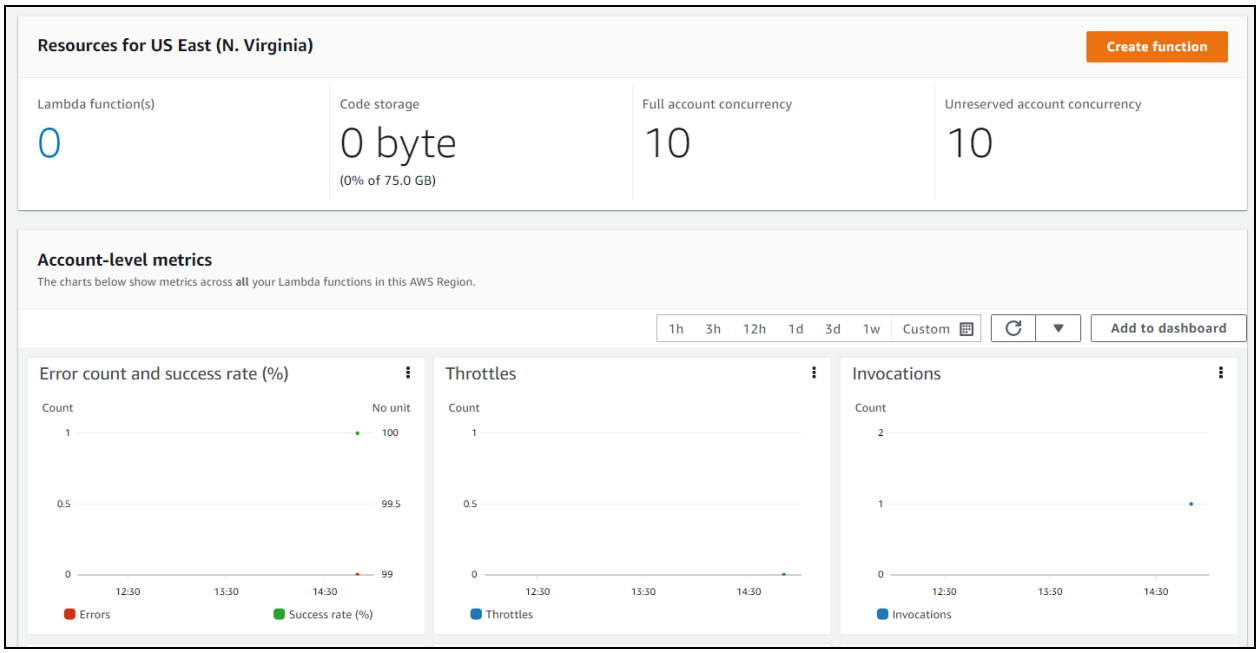
Step 1 : Enter The lambda console



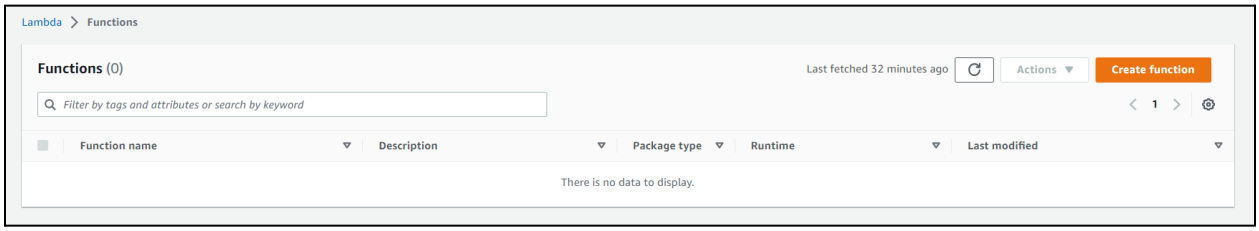
Step 2 : Select a lambda blueprint

Blueprints provide example code to do some minimal processing. Most blueprints process events from specific event sources, such as Amazon S3, Amazon DynamoDB, or a custom application.

a. In the AWS Lambda console, choose Create function.



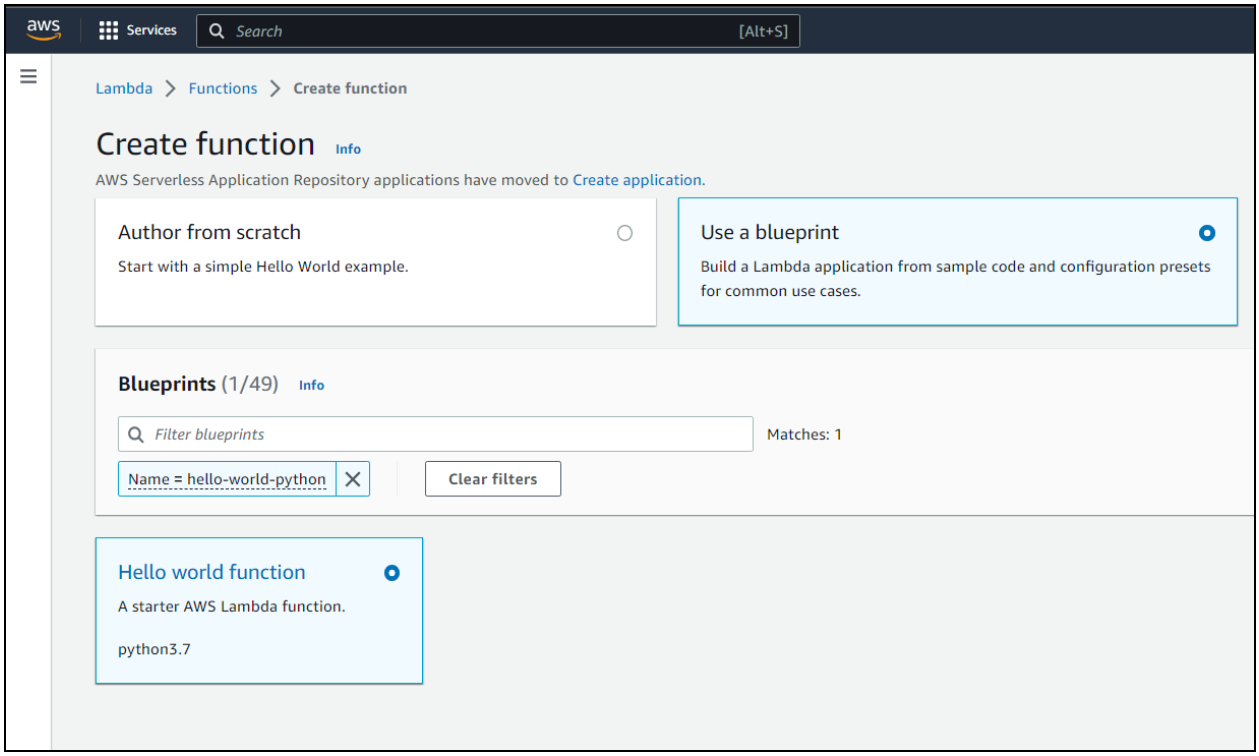
b. In functions, the create function is present



c. Select use a blueprint.

d. In the Filter box, enter hello-world-python and select the hello-world-python blueprint.

e. Then choose Configure.



Step 3 : Configure and create your lambda function

A Lambda function consists of code you provide, associated dependencies, and configuration. The configuration information you provide includes the compute resources you want to allocate (for example, memory), execution timeout, and an IAM role that AWS Lambda can assume to execute your Lambda function on your behalf.

a. You will now enter Basic information about your Lambda function.

Basic information:

Name: You can name your Lambda function here. For this tutorial, enter hello-world-python.

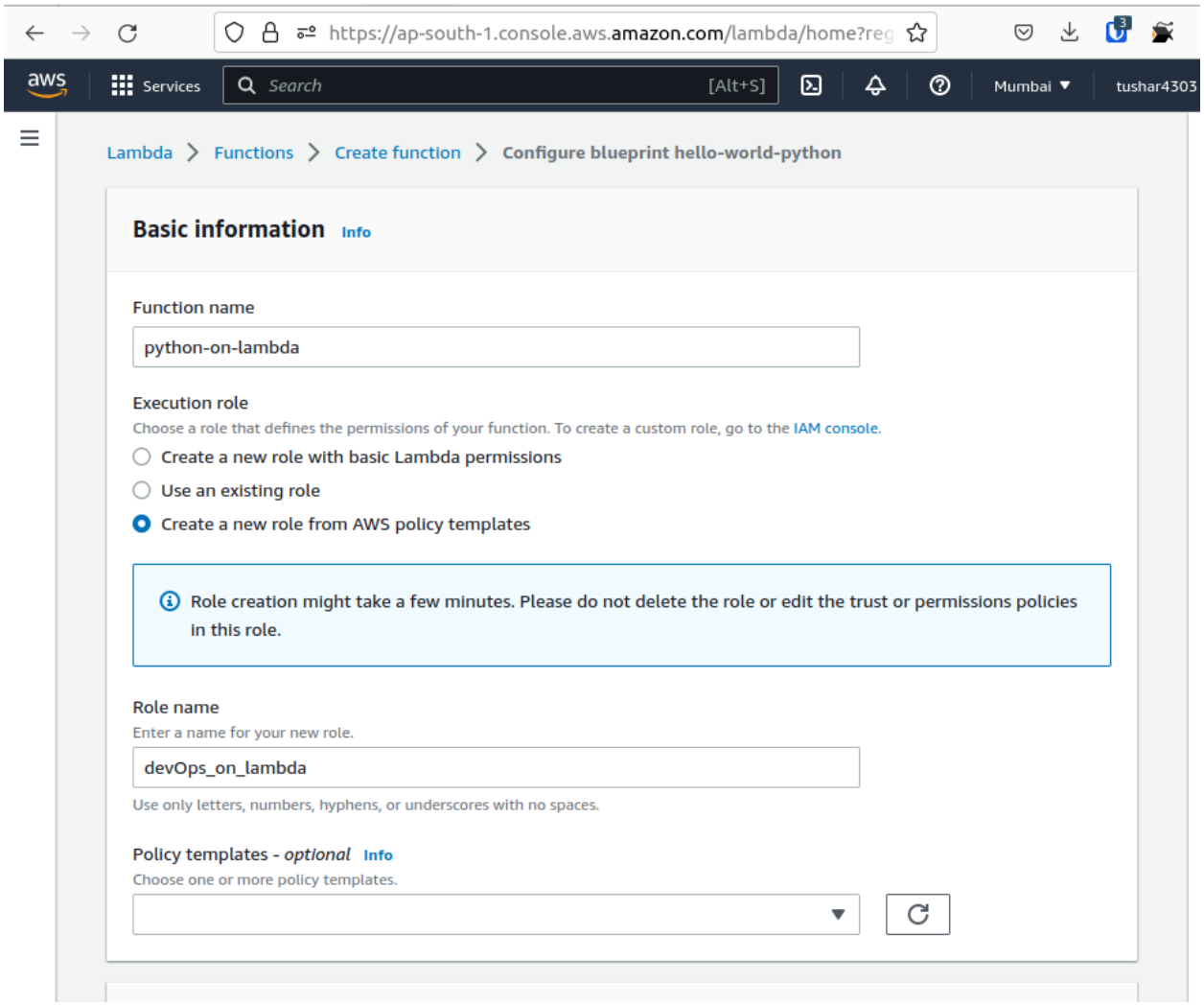
Role: You will create an IAM role (referred to as the execution role) with the necessary permissions that AWS Lambda can assume to invoke your Lambda

function on your behalf. Select Create a new role from AWS policy templates.

Role name: type lambda_basic_execution.

Lambda function code:

b. Go to the bottom of the page and choose the Create function.



Lambda function code

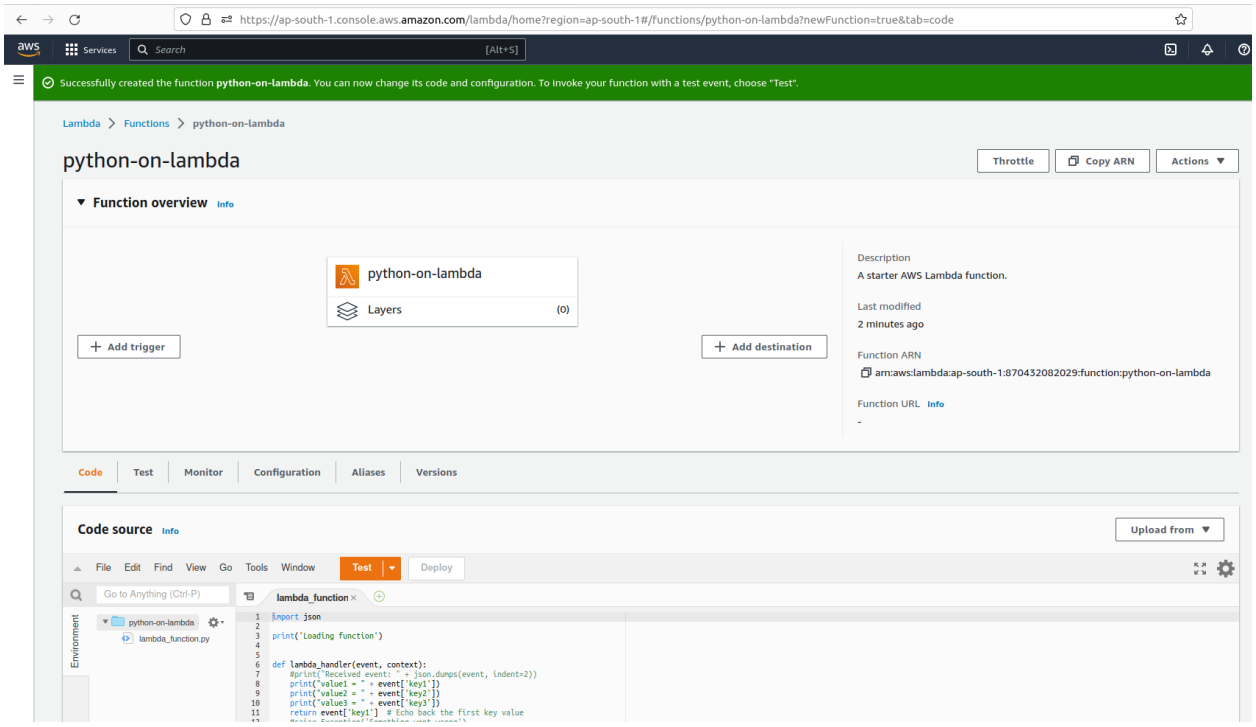
Code is preconfigured by the chosen blueprint. You can configure it after you create the function. Learn more about deploying Lambda functions.

This function contains external libraries.

Runtime	Architecture
Python 3.7	x86_64

```
1  import json
2
3  print('Loading function')
4
5
6  def lambda_handler(event, context):
7      #print("Received event: " + json.dumps(event, indent=2))
8      print("value1 = " + event['key1'])
9      print("value2 = " + event['key2'])
10     print("value3 = " + event['key3'])
11     return event['key1']  # Echo back the first key value
12     #raise Exception('Something went wrong')
13
```

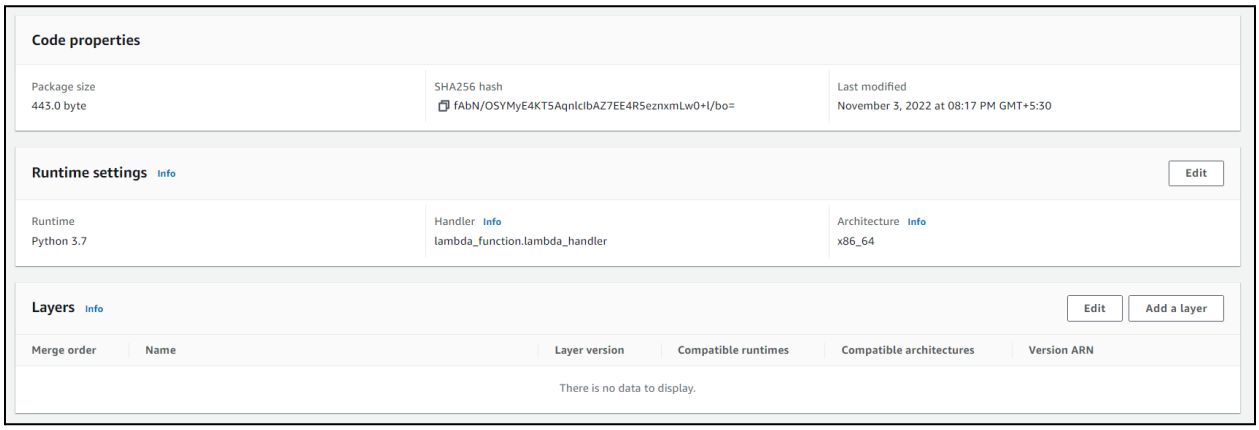
Cancel Create function



c. Runtime: Currently, you can author your Lambda function code in Java, Node.js, C#, Go, or Python. For this tutorial, use Python 3.7 as the runtime.

d. Handler: You can specify a handler (a method/function in your code) where AWS Lambda can begin executing your code. AWS Lambda provides event data as input to this handler, which processes the event.

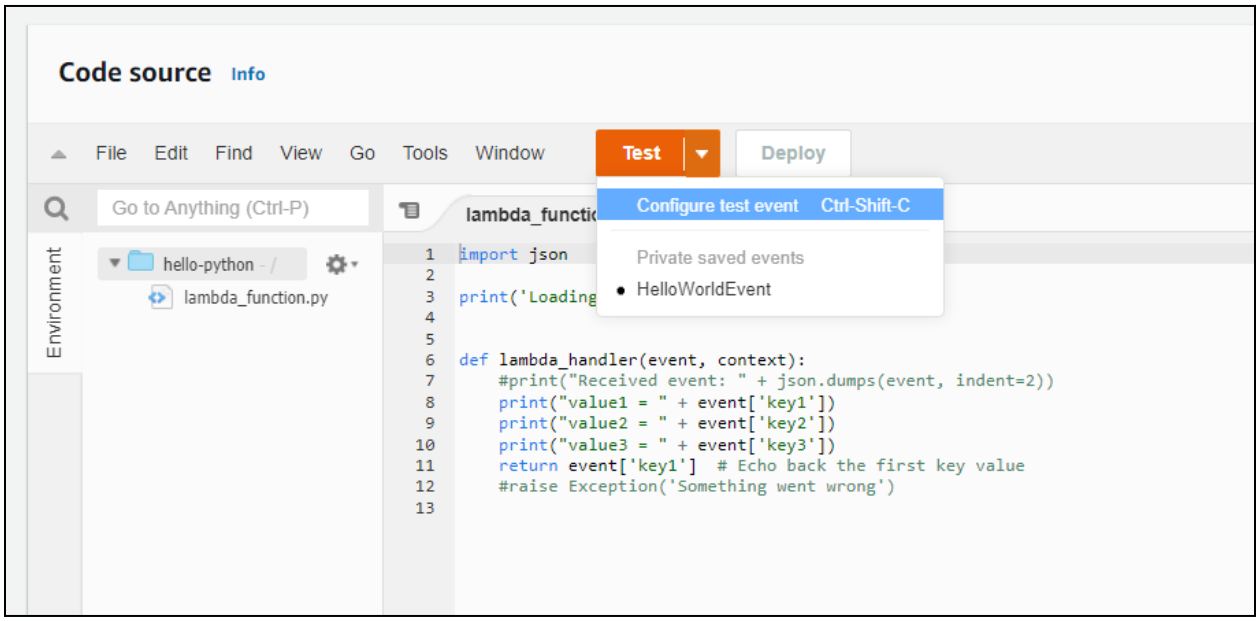
In this example, Lambda identifies this from the code sample and this should be pre-populated with lambda_function.lambda_handler.



Step 4 : Invoke lambda function and verify results

The console shows the hello-world-python Lambda function. You can now test the function, verify results, and review the logs.

a. Select Configure Test Event from the drop-down menu called Test.

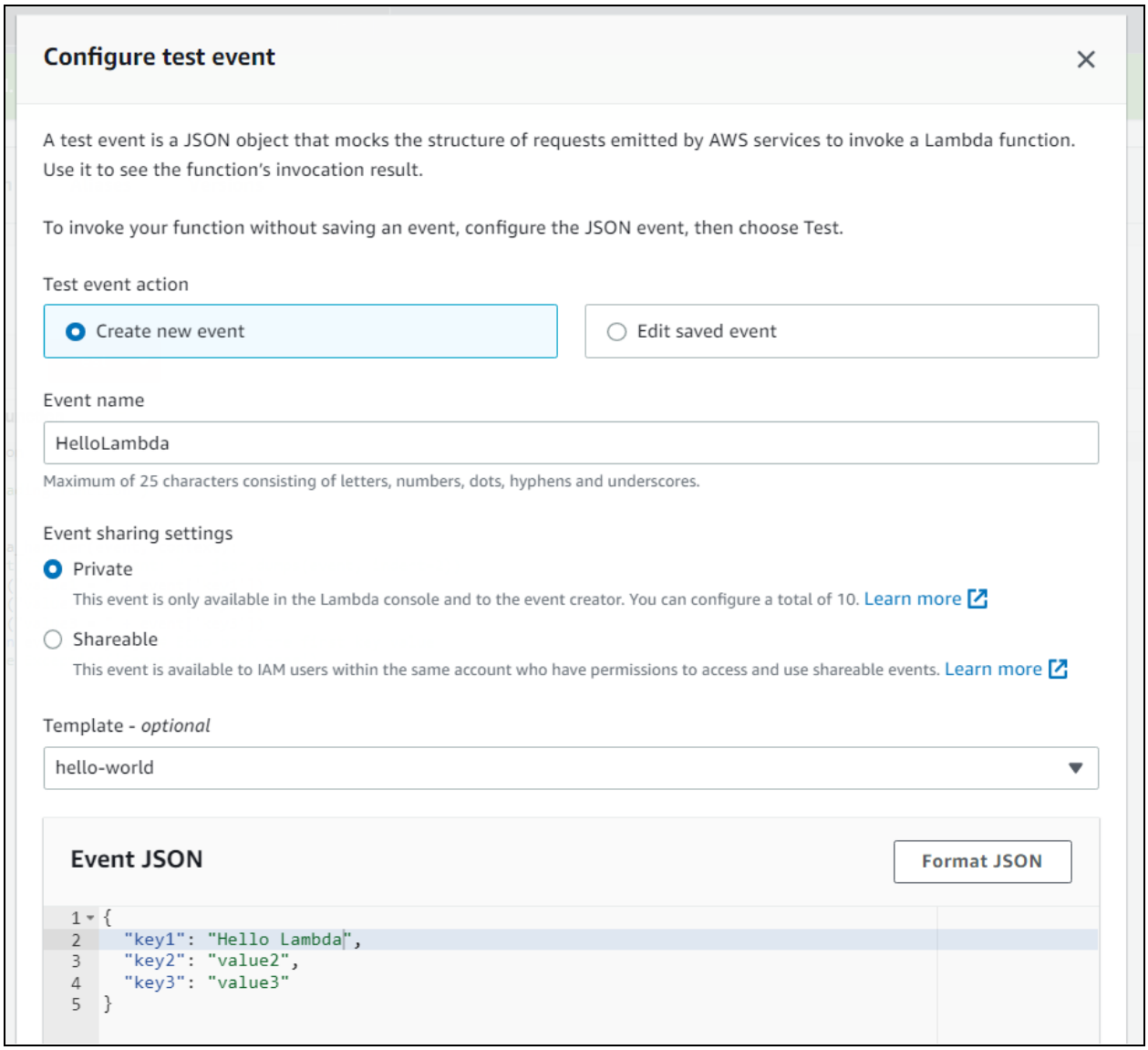


b. The editor pops up so you can enter an event to test your function.

- Select Create new event.

- Type in an event name like HelloWorldEvent.
- Retain default setting of Private for Event sharing settings.
- Choose hello-world from the template list.
- You can change the values in the sample JSON, but don’t change the event structure.

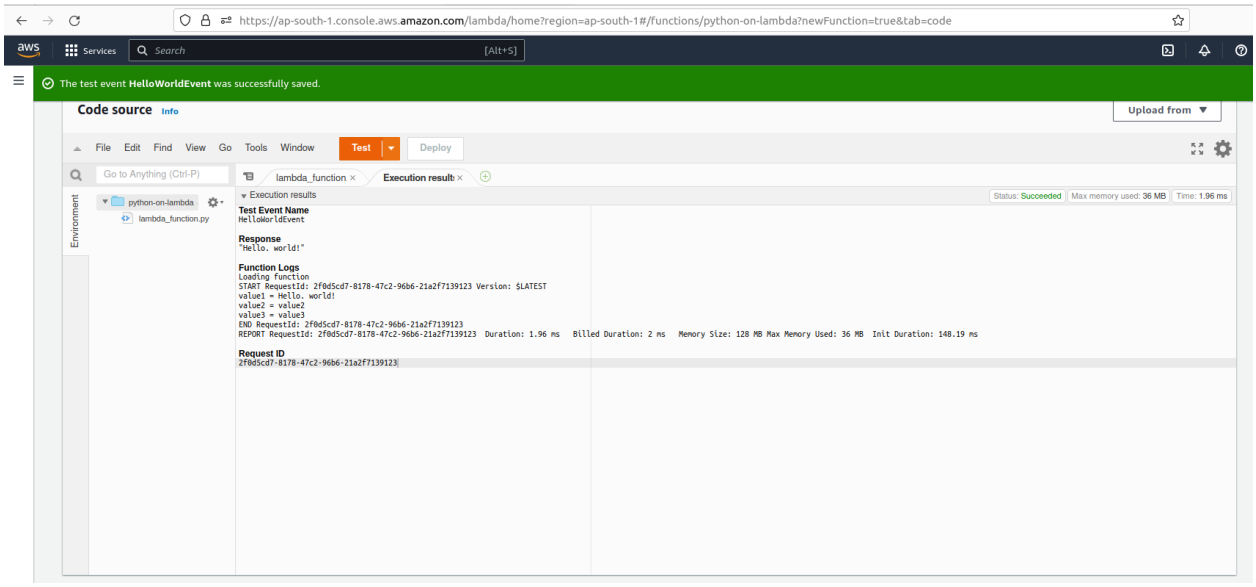
For now, replace value1 with hello world!. Select Create.



c. Choose Test.

d. Upon successful execution, view the results in the console:

- The Execution results tab verifies that the execution succeeded.
- The Function Logs section will show the logs generated by the Lambda function execution as well as key information reported in the Log output.



Step 5 : Monitor your metrics

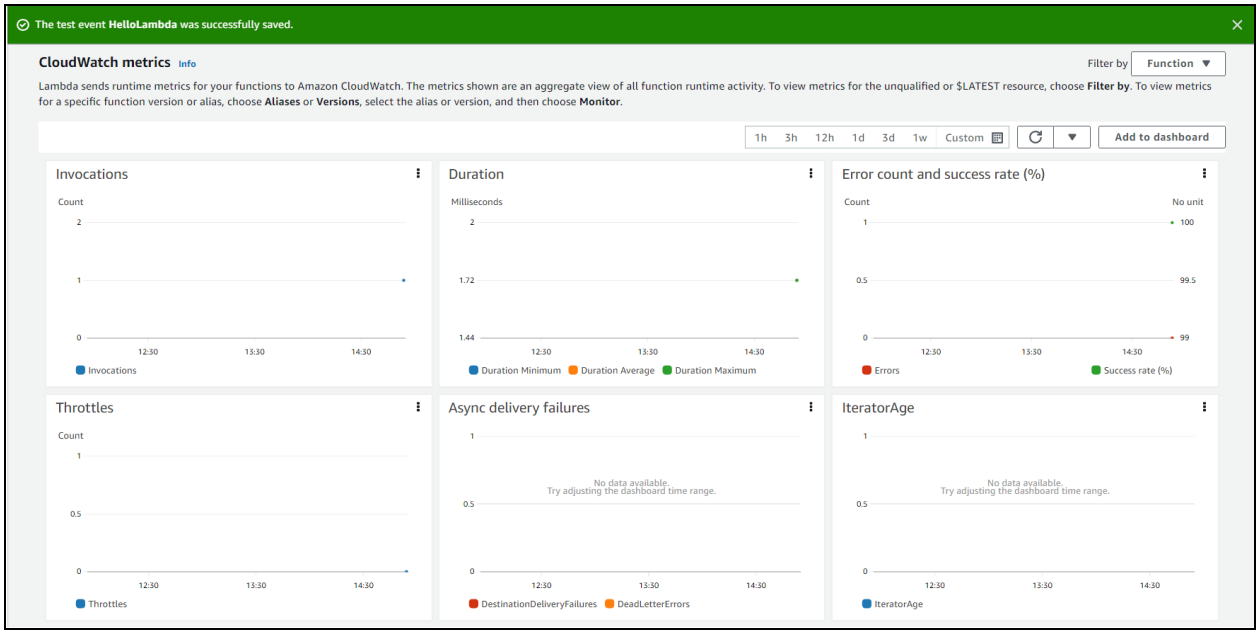
AWS Lambda automatically monitors Lambda functions and reports metrics through Amazon CloudWatch. To help you monitor your code as it executes, Lambda automatically tracks the number of requests, the latency per request, and the number of requests resulting in an error and publishes the associated metrics.

a. Invoke the Lambda function a few more times by repeatedly choosing the Test button. This will generate the metrics that can be viewed in the next step.

b. Select the Monitor tab to view the results.

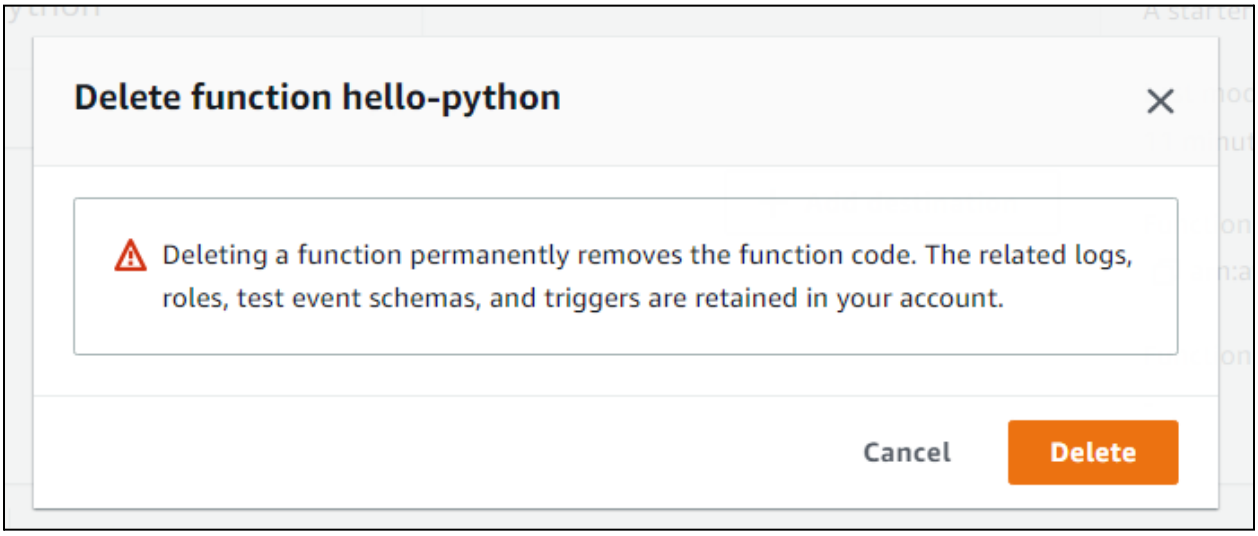
c. Scroll down to view the metrics for your Lambda function. Lambda metrics are reported through Amazon CloudWatch. You can leverage these metrics to set custom alarms. For more information about CloudWatch, see the Amazon CloudWatch Developer Guide.

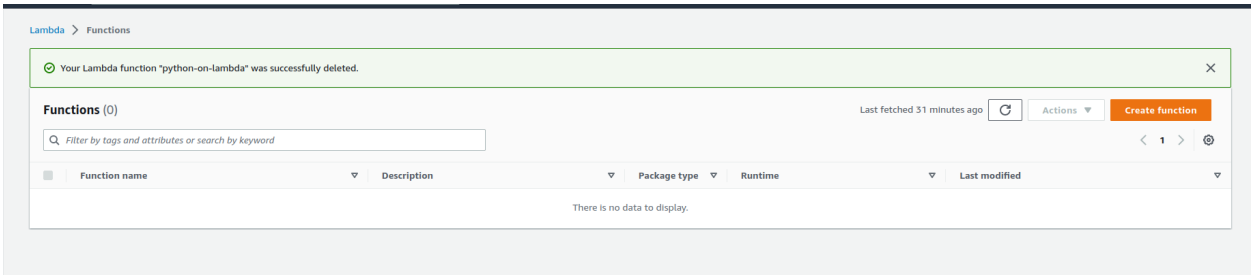
The Monitoring tab will show seven CloudWatch metrics: Invocations, Duration, Error count and success rate (%), Throttles, Async delivery failures, IteratorAge, and Concurrent executions.



Step 6 : Delete the lambda function

a. Select the Actions button and select Delete function.





Conclusion:

First AWS Lambda function is created successfully.

Reference:

[Run a Serverless "Hello, World!" with AWS Lambda]

ADVANCED DEVOPS LAB

Name : Tushar Padhy

Roll no : 42

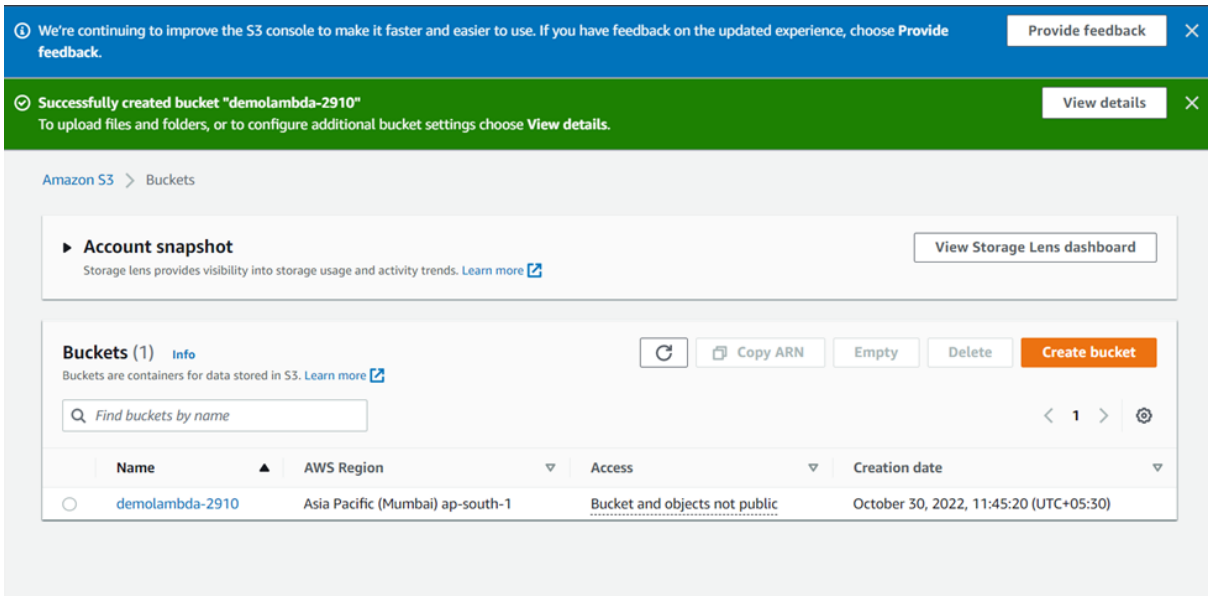
Experiment No : 12

To start using AWS Lambda with Amazon S3, we need the following −

1. Create S3 Bucket
2. Create role which has permission to work with s3 and lambda
3. Create a lambda function and add s3 as the trigger.

Create S3 Bucket

Step 1

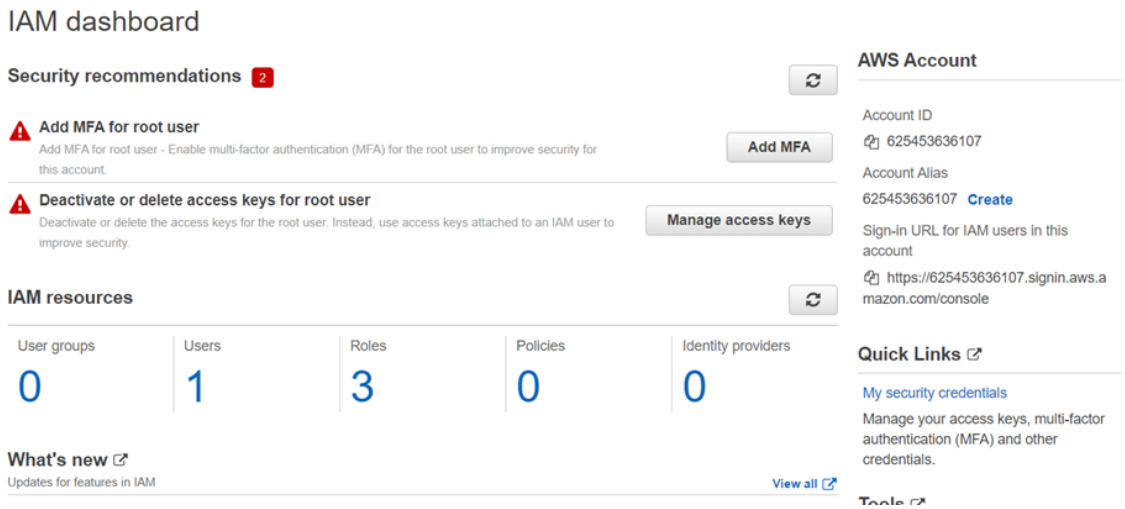


Create Role that Works with S3 and Lambda

To create role that works with S3 and Lambda, follow the Steps given below

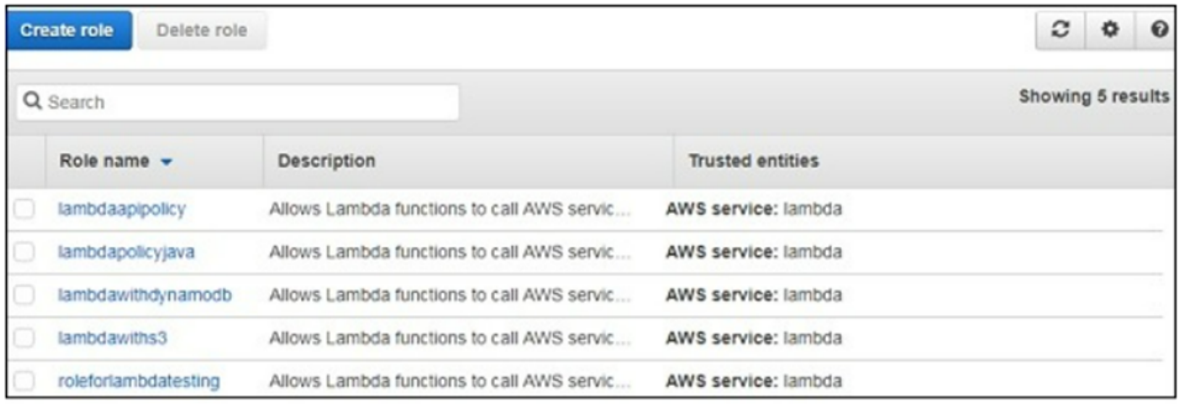
Step 2

Go to AWS services and select IAM as shown below −



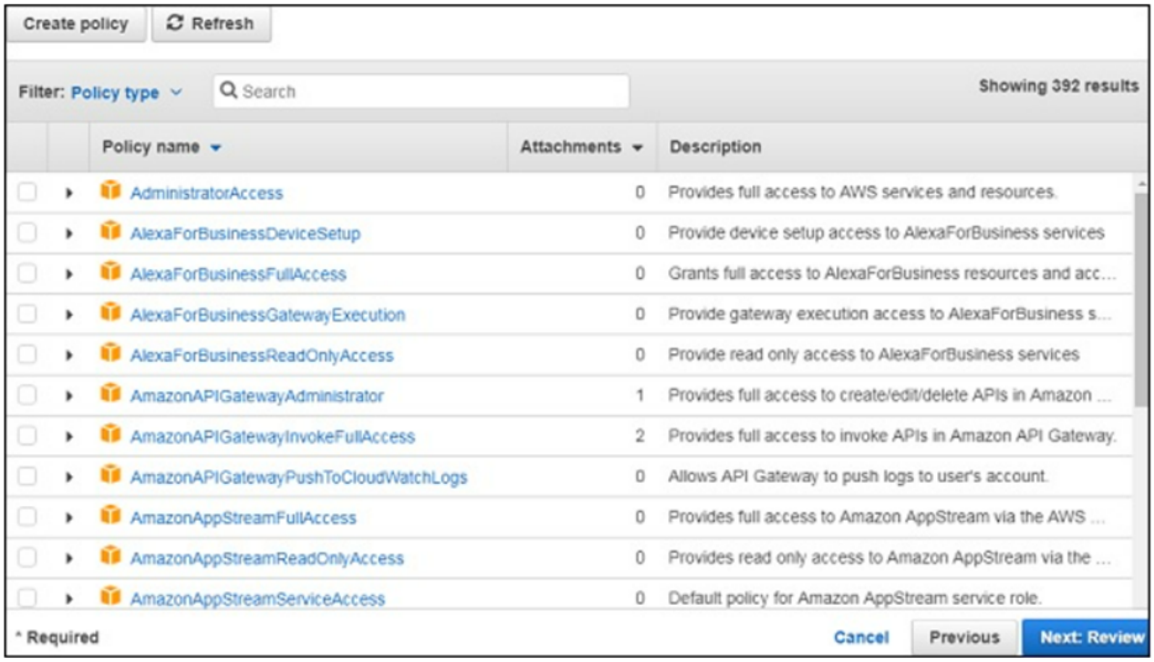
Step 3

Now, go to IAM and give the following permissions-



Step 4

Add the permission and click Review.



Step 5

Choose the following permissions −

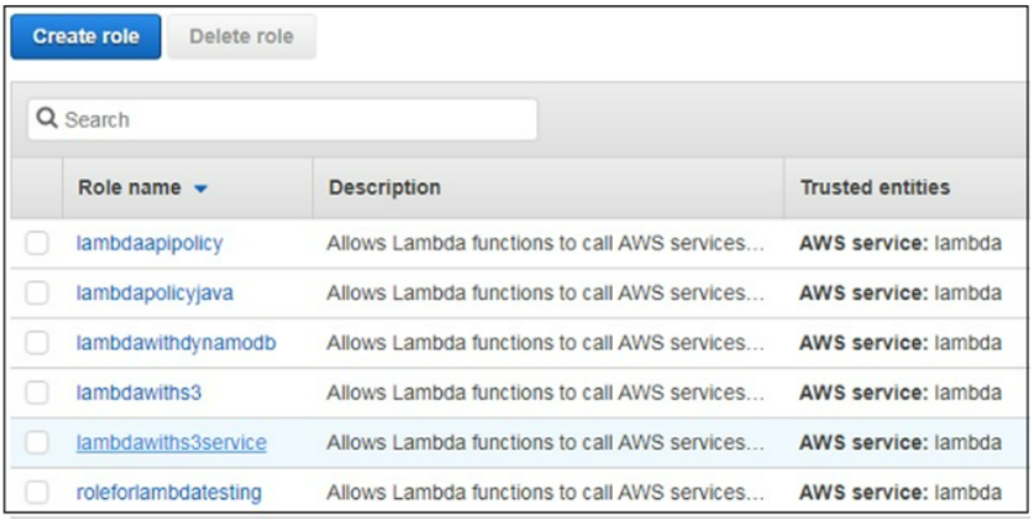
AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess.

Step 6

Create Lambda function and Add S3 Trigger

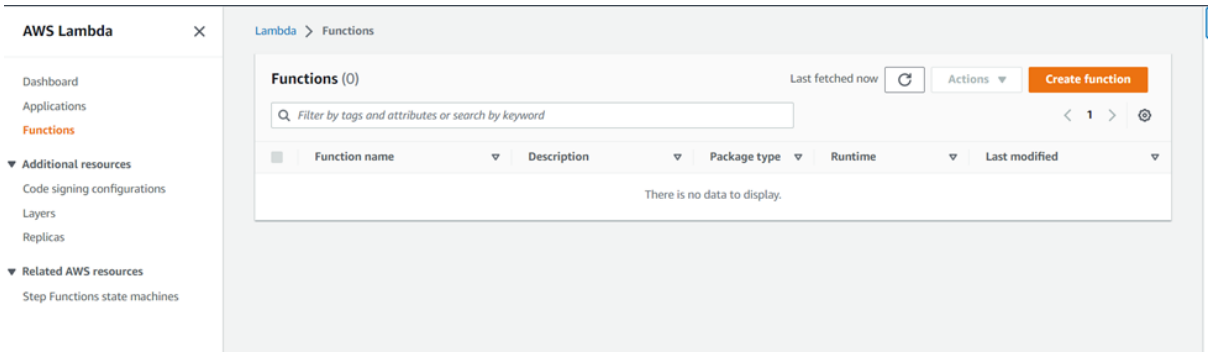
In this section, let us see how to create a Lambda function and add a S3 trigger to it. For this purpose, you will have to follow the Steps given below −

Now, enter the Role name, Role description and click ***Create Role*** button at the bottom.



Step 7

Go to AWS Services and select Lambda as shown below −



Step 8

Click Lambda and follow the process for adding Name. Choose the Runtime, Role etc. and create the function. The Lambda function that we have created is shown in the screenshot below −

Basic information

Function name

Enter a name that describes the purpose of your function.

demolambdacode

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 16.x

Architecture Info

Choose the instruction set architecture you want for your function code.

- (•) x86_64
- ( ) arm64

Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

- ( ) Create a new role with basic Lambda permissions
- (•) Use an existing role
- ( ) Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

demolambdarole

View the demolambdarole role on the IAM console.

Lambda > Functions

Functions (1)

Last fetched 3 minutes ago | Actions | Create function

Filter by tags and attributes or search by keyword

1

Function name	Description	Package type	Runtime	Last modified
demolambda	-	Zip	Node.js 16.x	2 minutes ago

demolambdacode

Throttle | Copy ARN | Actions

▼ Function overview Info

+ Add trigger

+ Add destination

Description
-

Last modified
28 seconds ago

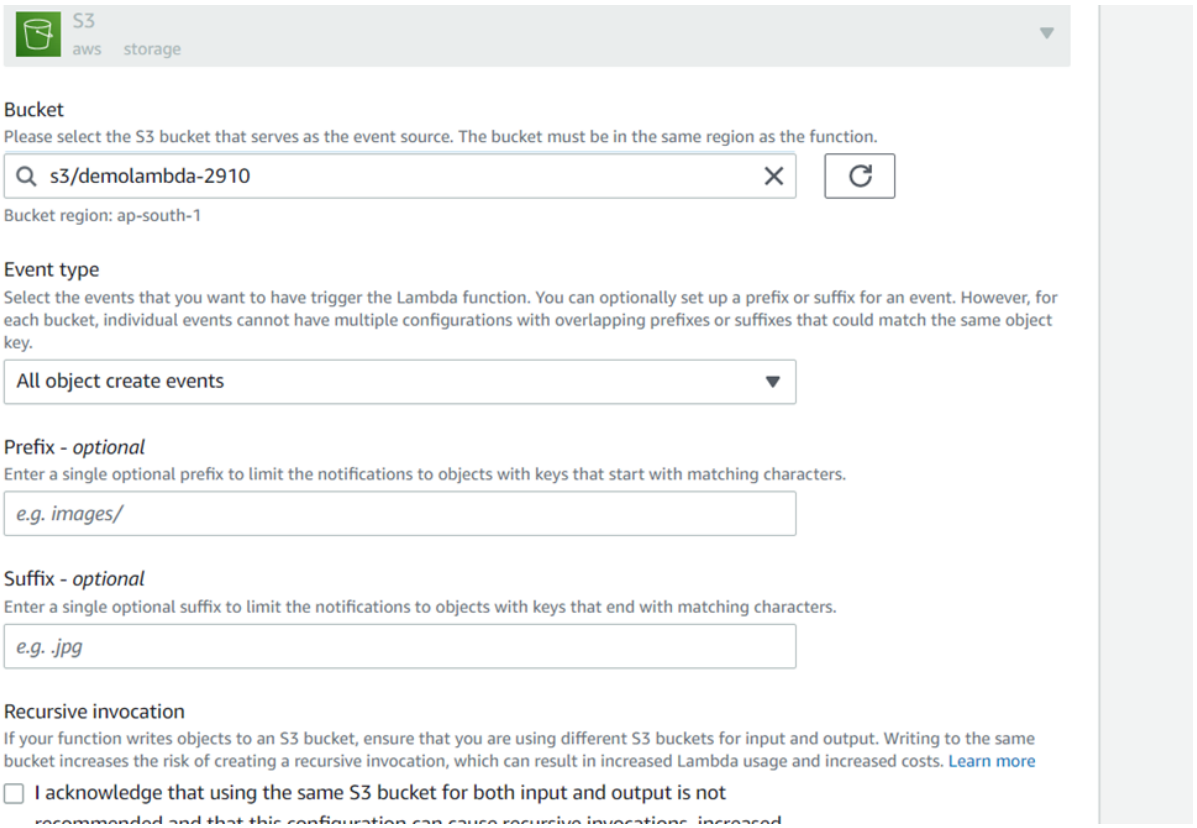
Function ARN
arn:aws:lambda:ap-south-1:625453636107:function:demolambdacode

Function URL Info
-

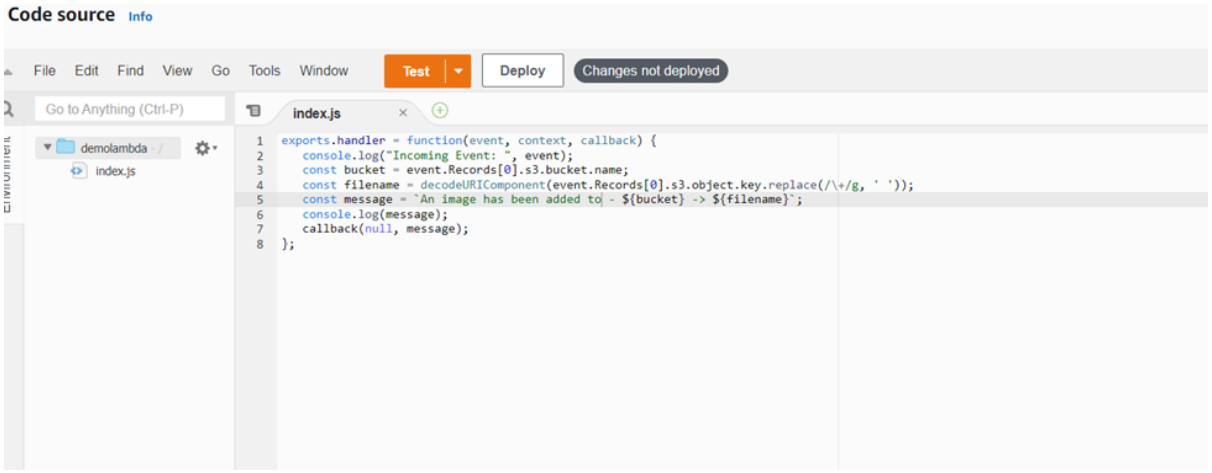
Code | Test | Monitor | Configuration | Aliases | Versions

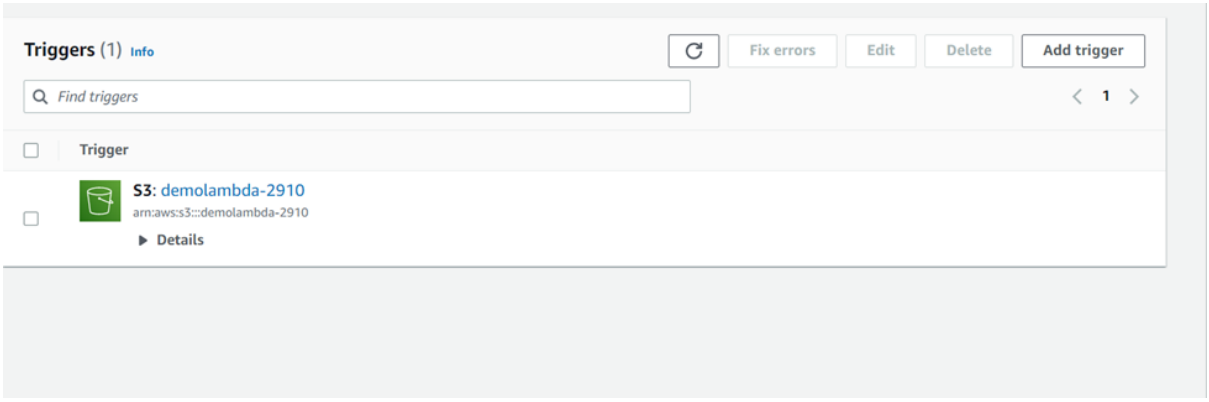
Hence a lambda function is created

Now Let's Create a trigger



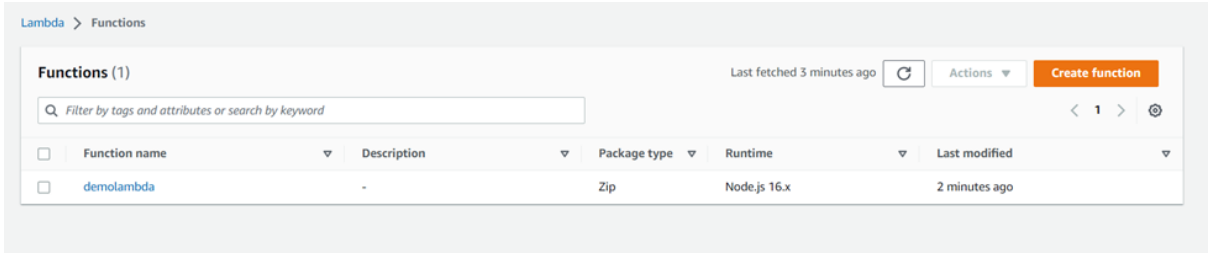
Below is the code written in node.js to pinned a log about an object that has been added to s3.





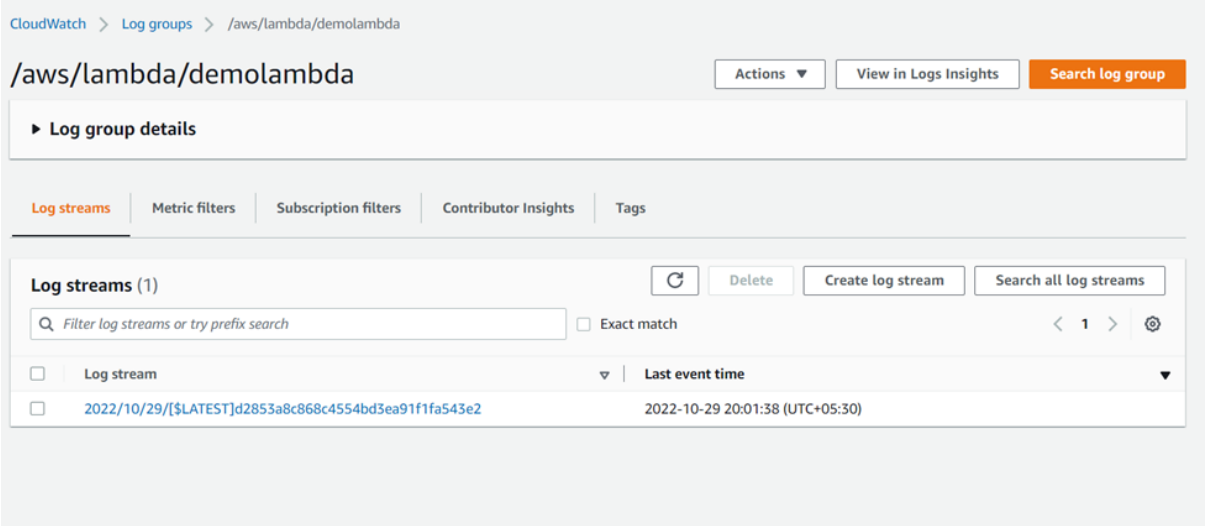
Now to test our lambda function with s3 trigger.

Step 9.add a image to s3 bucket



Step 10.

Now go to cloudwatch we'll see the logs of newly happened activity in s3.


CloudWatch > Log groups > /aws/lambda/demolambda
/aws/lambda/demolambda
Actions
View in Logs Insights
Search log group
Log group details
Log streams
Metric filters
Subscription filters
Contributor Insights
Tags
Log streams (1)
Delete
Create log stream
Search all log streams
Filter log streams or try prefix search
Exact match
1
Log stream
Last event time
2022/10/29/[$LATEST]d2853a8c868c4554bd3ea91f1fa543e2
2022-10-29 20:01:38 (UTC+05:30)