Bansilal Ramnath Agarwal Charitable Trust's

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY,

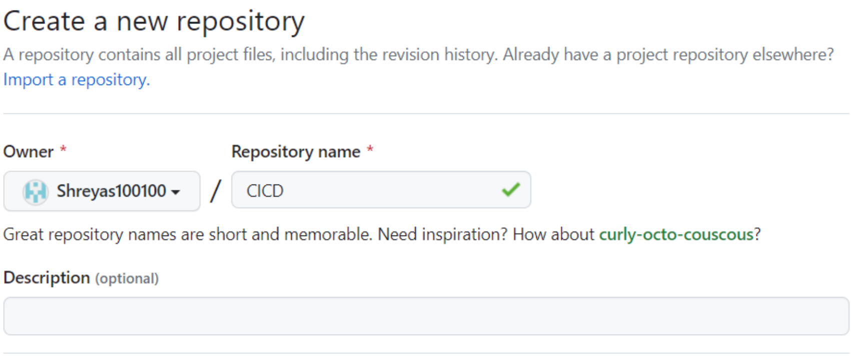
PUNE-48 Department of Information Technology

ITUA32202: CLOUD COMPUTING SCE

AIM: Deploy a CICD Pipeline using GitHub Actions for a Nodejs application.

Step by Step Implementation of the Project

-> Create a GitHub Repository named CICD



Create a new repository

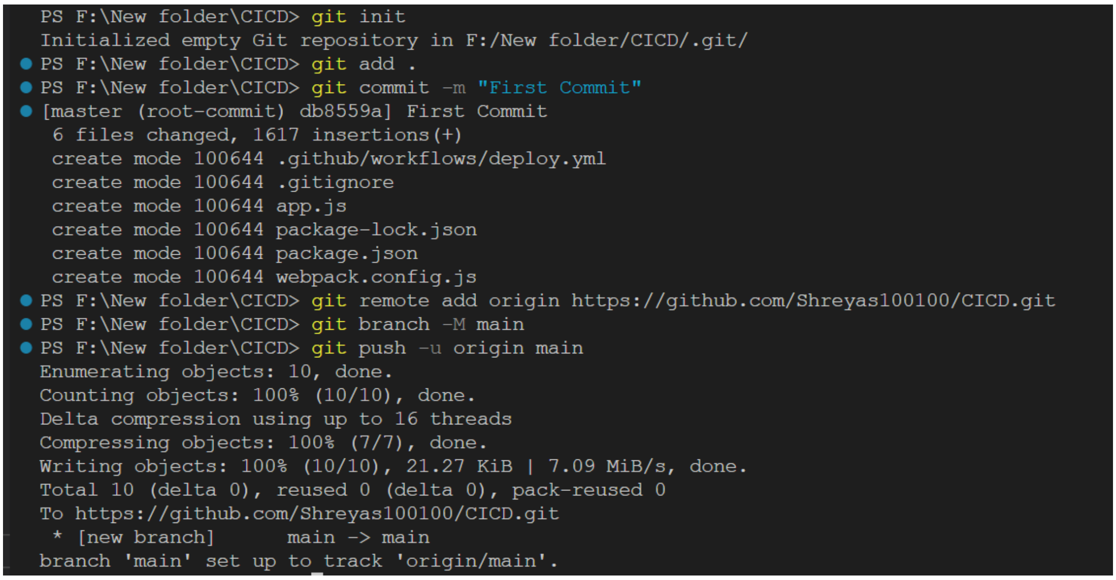
A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Shreyas100100 / Repository name * CICD ✓

Great repository names are short and memorable. Need inspiration? How about curly-octo-couscous?

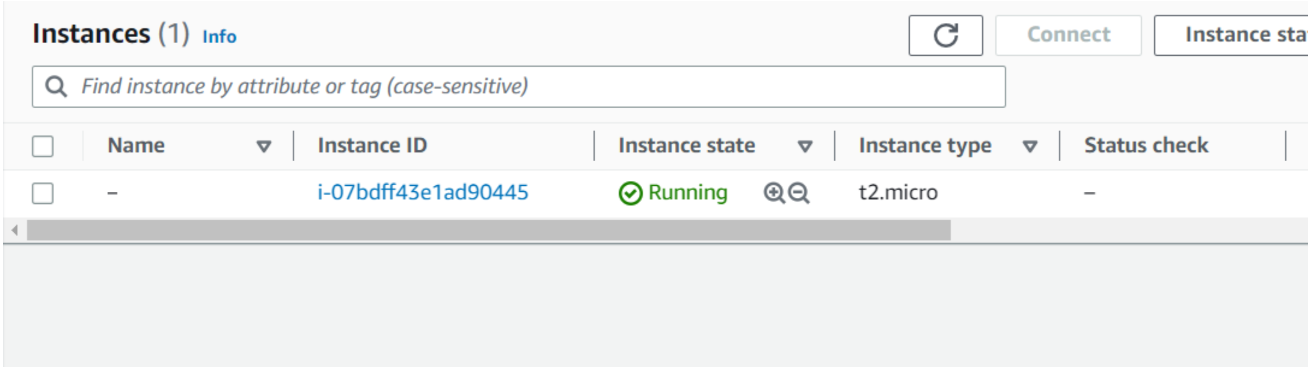
Description (optional)

-> Push the local machine code files into the GitHub Repository



```
PS F:\New folder\CICD> git init
Initialized empty Git repository in F:/New folder/CICD/.git/
PS F:\New folder\CICD> git add .
PS F:\New folder\CICD> git commit -m "First Commit"
[master (root-commit) db8559a] First Commit
 6 files changed, 1617 insertions(+)
 create mode 100644 .github/workflows/deploy.yml
 create mode 100644 .gitignore
 create mode 100644 app.js
 create mode 100644 package-lock.json
 create mode 100644 package.json
 create mode 100644 webpack.config.js
PS F:\New folder\CICD> git remote add origin https://github.com/Shreyas100100/CICD.git
PS F:\New folder\CICD> git branch -M main
PS F:\New folder\CICD> git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 21.27 KiB | 7.09 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Shreyas100100/CICD.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

-> Create an ec2 Instance



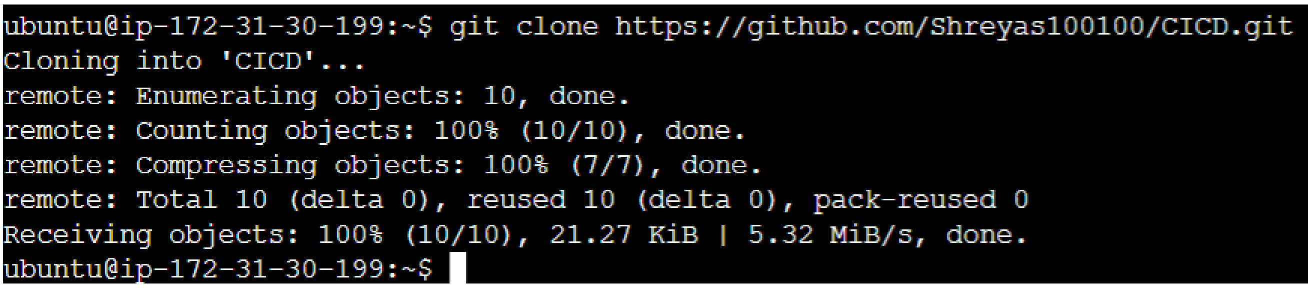
Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Status check
□	–	i-07bdff43e1ad90445	✓ Running	t2.micro	–

-> Run the following commands on the Terminal of the VM on ec2 Instance

```
sudo apt update
sudo apt install nodejs npm git
sudo npm install pm2 -g
```

-> Clone the GitHub Repository on the VM

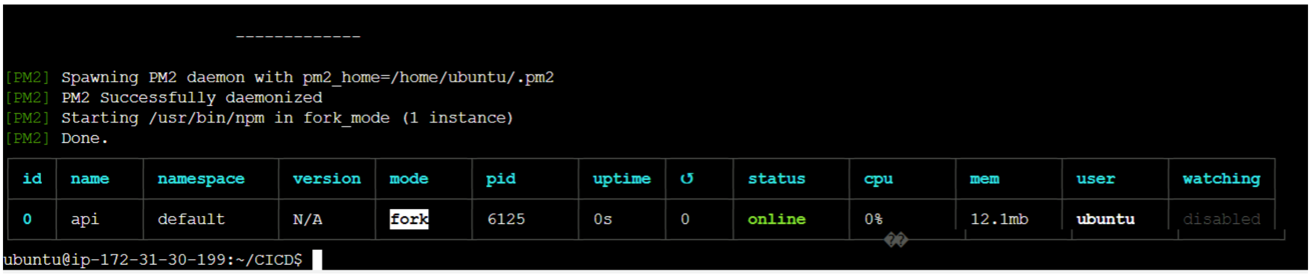


```
ubuntu@ip-172-31-30-199:~$ git clone https://github.com/Shreyas100100/CICD.git
Cloning into 'CICD'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 21.27 KiB | 5.32 MiB/s, done.
ubuntu@ip-172-31-30-199:~$
```

-> Start the pm2 (process manager)

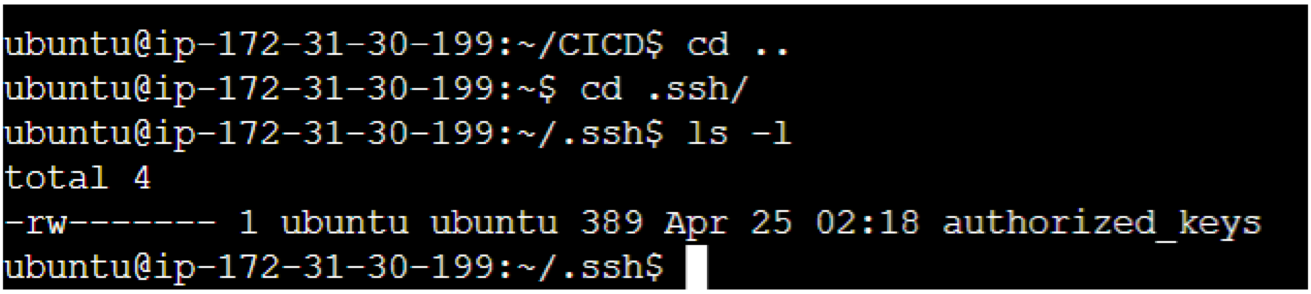
```
pm2 start npm --name api -- run start:prod
```

this will start our Node js application named api using pm2 Process Manager.



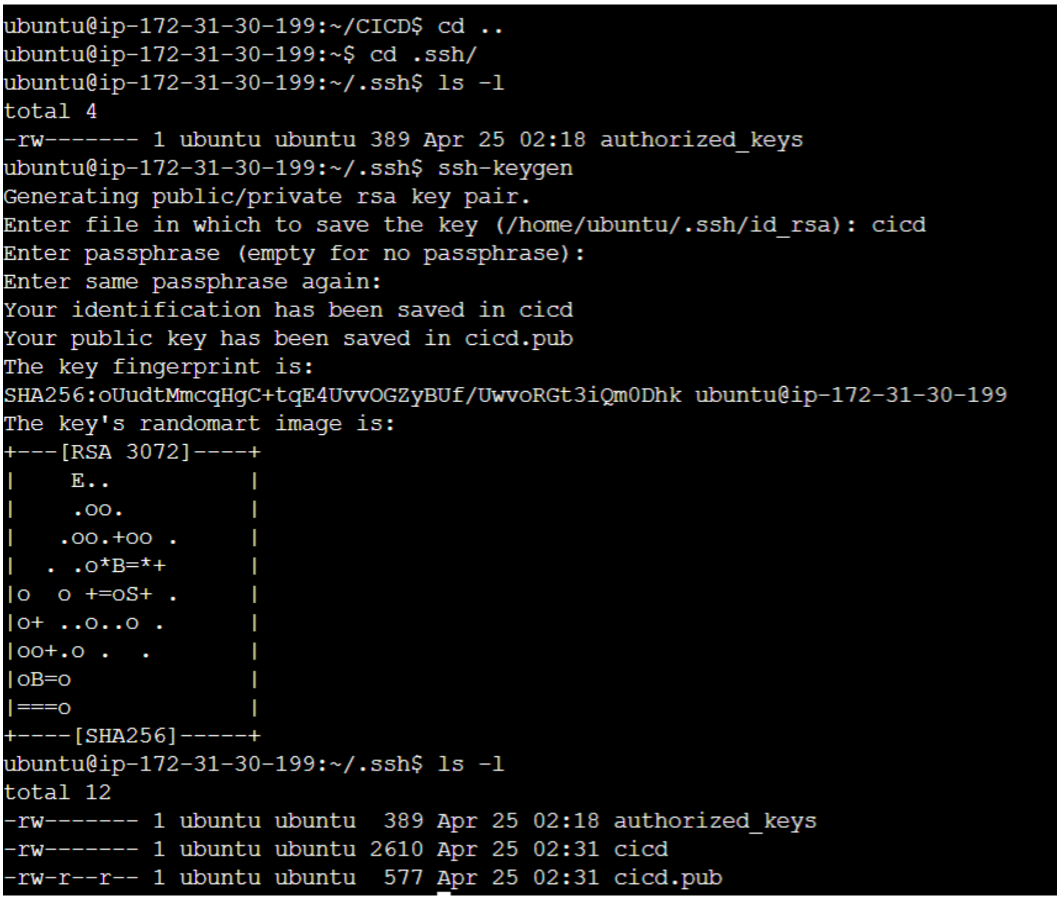
```
                        -------------

[PM2] Spawning PM2 daemon with pm2_home=/home/ubuntu/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /usr/bin/npm in fork_mode (1 instance)
[PM2] Done.
│ id │ name │ namespace │ version │ mode │ pid  │ uptime │ ↺ │ status │ cpu │ mem    │ user   │ watching │
│ 0  │ api  │ default   │ N/A     │ fork │ 6125 │ 0s     │ 0 │ online │ 0%  │ 12.1mb │ ubuntu │ disabled │
ubuntu@ip-172-31-30-199:~/CICD$
```

-> Change Directory to .ssh/ folder and check the authorization keys



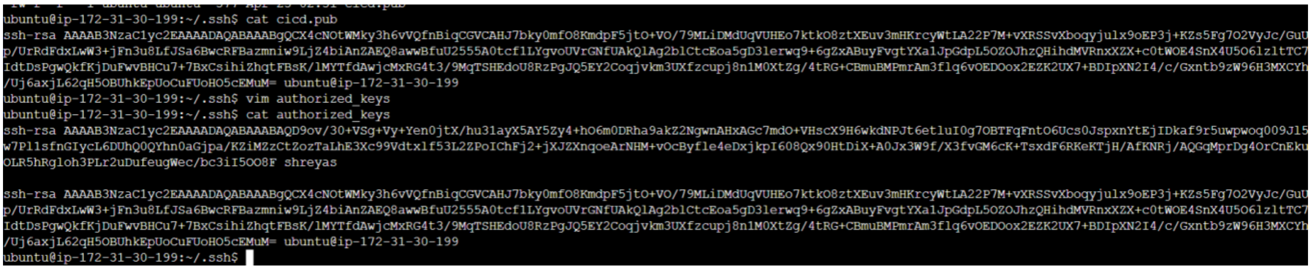
```
ubuntu@ip-172-31-30-199:~/CICD$ cd ..
ubuntu@ip-172-31-30-199:~$ cd .ssh/
ubuntu@ip-172-31-30-199:~/.ssh$ ls -l
total 4
-rw------- 1 ubuntu ubuntu 389 Apr 25 02:18 authorized_keys
ubuntu@ip-172-31-30-199:~/.ssh$
```

-> using `ssh-keygen` we will create public & private keys



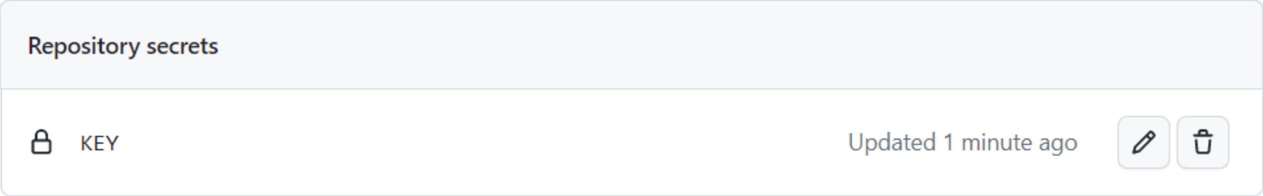
```
ubuntu@ip-172-31-30-199:~/CICD$ cd ..
ubuntu@ip-172-31-30-199:~$ cd .ssh/
ubuntu@ip-172-31-30-199:~/.ssh$ ls -l
total 4
-rw------- 1 ubuntu ubuntu 389 Apr 25 02:18 authorized_keys
ubuntu@ip-172-31-30-199:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): cicd
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cicd
Your public key has been saved in cicd.pub
The key fingerprint is:
SHA256:oUudtMmcqHgC+tqE4UvvOGZyBUf/UwvoRGt3iQm0Dhk ubuntu@ip-172-31-30-199
The key's randomart image is:
+---[RSA 3072]----+
|     E..         |
|     .oo.        |
|    .oo.+oo .    |
|   . .o*B=*+     |
|o   o +=oS+ .    |
|o+ ..o..o .      |
|oo+.o .   .      |
|oB=o             |
|===o             |
+----[SHA256]-----+
ubuntu@ip-172-31-30-199:~/.ssh$ ls -l
total 12
-rw------- 1 ubuntu ubuntu  389 Apr 25 02:18 authorized_keys
-rw------- 1 ubuntu ubuntu 2610 Apr 25 02:31 cicd
-rw-r--r-- 1 ubuntu ubuntu  577 Apr 25 02:31 cicd.pub
```

-> Now we must put the public key into the authorized_key



```
-rw-r--r-- 1 ubuntu ubuntu  577 Apr 25 02:31 cicd.pub
ubuntu@ip-172-31-30-199:~/.ssh$ cat cicd.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCX4cNOtWMky3h6vVQfnBiqCGVCAHJ7bky0mfO8KmdpF5jtO+VO/79MLiDMdUqVUHEo7ktkO8ztXEuv3mHKrcyWtLA22P7M+vXRSSvXboqyjulx9oEP3j+KZs5Fg7O2VyJc/GuU
p/UrRdFdxLwW3+jFn3u8LfJSa6BwcRFBazmniw9LjZ4biAnZAEQ8awwBfuU2555A0tcf1LYgvoUVrGNfUAkQlAg2blCtcEoa5gD3lerwq9+6gZxABuyFvgtYXa1JpGdpL5OZOJhzQHihdMVRnxXZX+c0tWOE4SnX4U5O6lzltTC7
IdtDsPgwQkfKjDuFwvBHCu7+7BxCsihiZhqtFBsK/lMYTfdAwjcMxRG4t3/9MqTSHEdoU8RzPgJQ5EY2Coqjvkm3UXfzcupj8n1M0XtZg/4tRG+CBmuBMPmrAm3flq6vOEDOox2EZK2UX7+BDIpXN2I4/c/Gxntb9zW96H3MXCYh
/Uj6axjL62qH5OBUhkEpUoCuFUoHO5cEMuM= ubuntu@ip-172-31-30-199
ubuntu@ip-172-31-30-199:~/.ssh$ vim authorized_keys
ubuntu@ip-172-31-30-199:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQD9ov/30+VSg+Vy+Yen0jtX/hu31ayX5AY5Zy4+hO6m0DRha9akZ2NgwnAHxAGc7mdO+VHscX9H6wkdNPJt6etluI0g7OBTFqFntO6Ucs0JspxnYtEjIDkaf9r5uwpwoq009Jl5
w7Pl1sfnGIycL6DUhQ0QYhn0aGjpa/KZiMZzCtZozTaLhE3Xc99Vdtx1f53L2ZPoIChFj2+jXJZXnqoeArNHM+vOcByfle4eDxjkpI608Qx90HtDiX+A0Jx3W9f/X3fvGM6cK+TsxdF6RKeKTjH/AfKNRj/AQGqMprDg4OrCnEku
OLR5hRgloh3PLr2uDufeugWec/bc3iI5OO8F shreyas

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCX4cNOtWMky3h6vVQfnBiqCGVCAHJ7bky0mfO8KmdpF5jtO+VO/79MLiDMdUqVUHEo7ktkO8ztXEuv3mHKrcyWtLA22P7M+vXRSSvXboqyjulx9oEP3j+KZs5Fg7O2VyJc/GuU
p/UrRdFdxLwW3+jFn3u8LfJSa6BwcRFBazmniw9LjZ4biAnZAEQ8awwBfuU2555A0tcf1LYgvoUVrGNfUAkQlAg2blCtcEoa5gD3lerwq9+6gZxABuyFvgtYXa1JpGdpL5OZOJhzQHihdMVRnxXZX+c0tWOE4SnX4U5O6lzltTC7
IdtDsPgwQkfKjDuFwvBHCu7+7BxCsihiZhqtFBsK/lMYTfdAwjcMxRG4t3/9MqTSHEdoU8RzPgJQ5EY2Coqjvkm3UXfzcupj8n1M0XtZg/4tRG+CBmuBMPmrAm3flq6vOEDOox2EZK2UX7+BDIpXN2I4/c/Gxntb9zW96H3MXCYh
/Uj6axjL62qH5OBUhkEpUoCuFUoHO5cEMuM= ubuntu@ip-172-31-30-199
ubuntu@ip-172-31-30-199:~/.ssh$
```

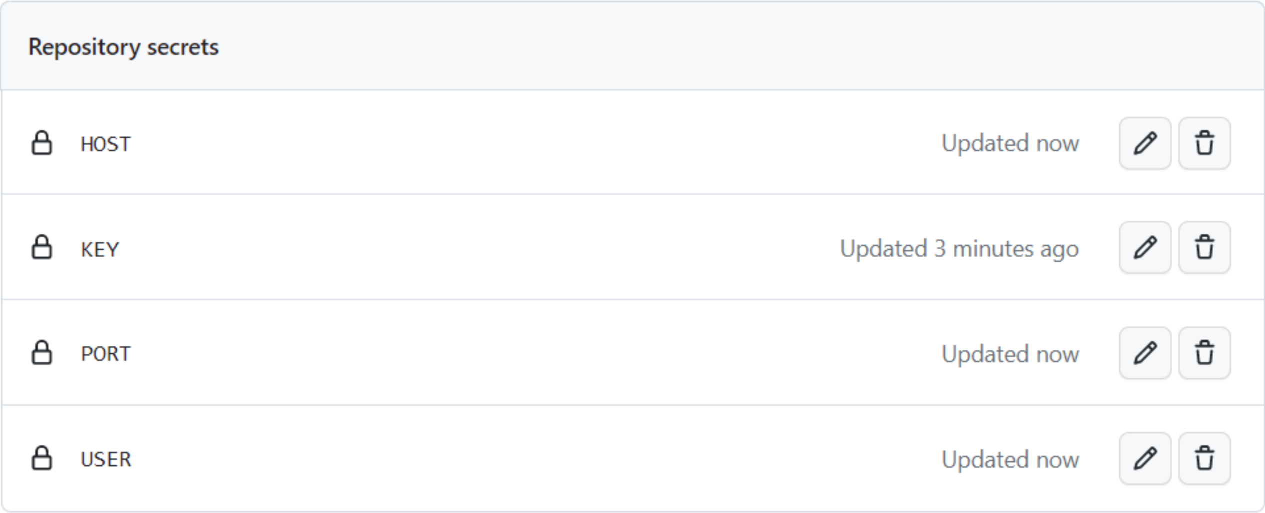
-> Copy the private key and add it as a secret named KEY in GitHub Actions of our Repository



Repository secrets

KEY Updated 1 minute ago

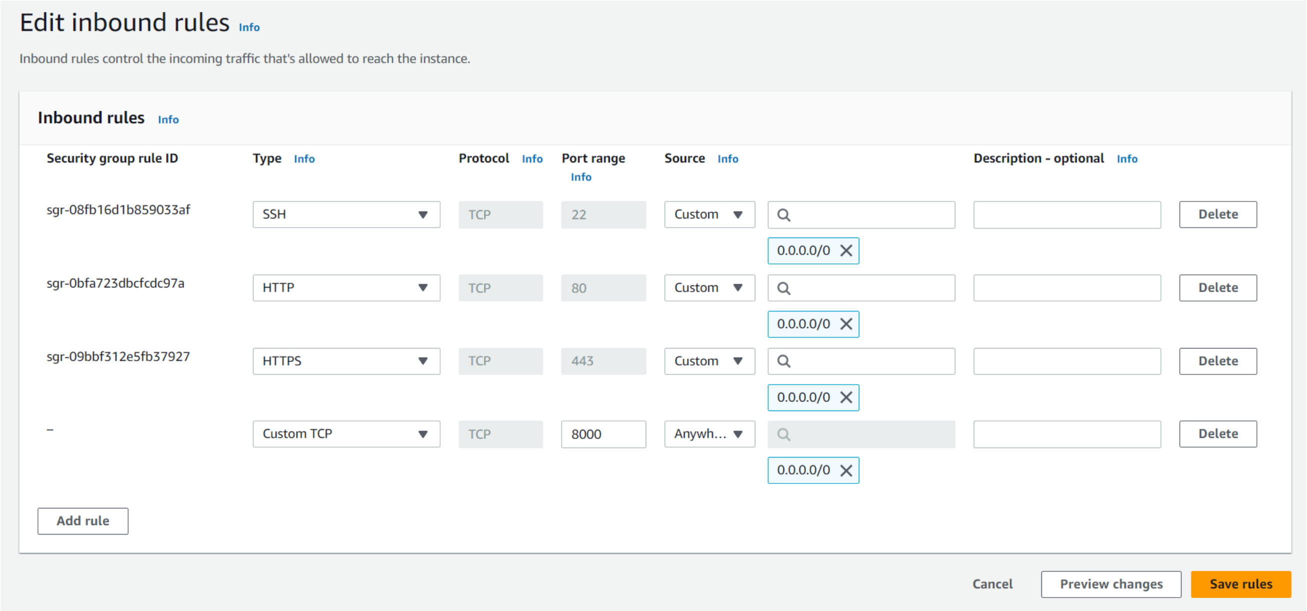
-> Similarly add the Secrets for HOST, USER & PORT in GitHub Actions



Repository secrets

HOST	Updated now	
KEY	Updated 3 minutes ago	
PORT	Updated now	
USER	Updated now	

-> Our Nodejs application runs on Port 8000 so we need to update the inbound rules.



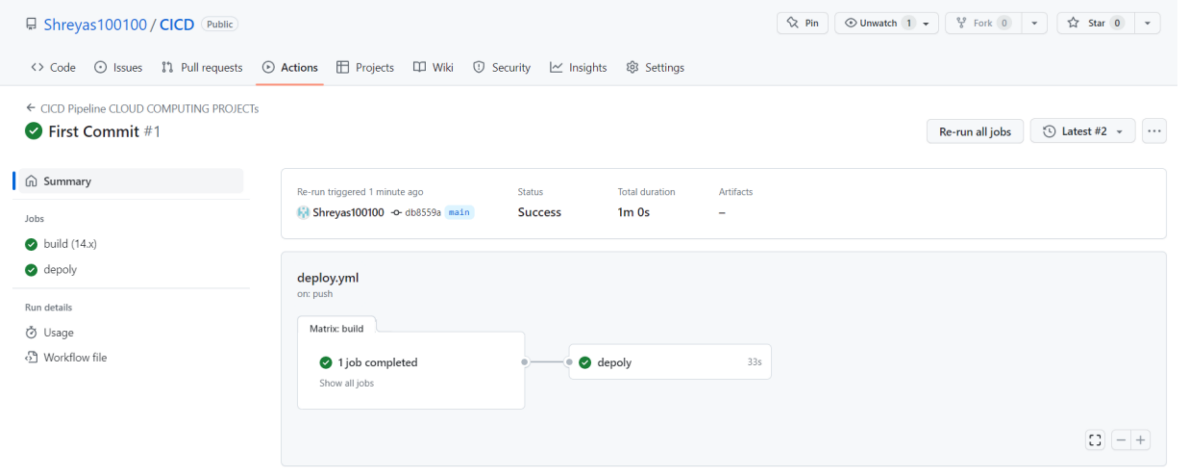
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-08fb16d1b859033af	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0bfa723dbcfcdc97a	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
sgr-09bbf312e5fb37927	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete
–	Custom TCP	TCP	8000	Anywh...	0.0.0.0/0		Delete

-> On the GitHub Repo after adding the secrets we must re-run the .yml file



Shreyas100100 / CICD Public

← CICD Pipeline CLOUD COMPUTING PROJECTs

First Commit #1

Jobs

- build (14.x)
- depoly

Run details

Re-run triggered 1 minute ago	Status	Total duration	Artifacts
Shreyas100100 db8559a main	Success	1m 0s	–

deploy.yml

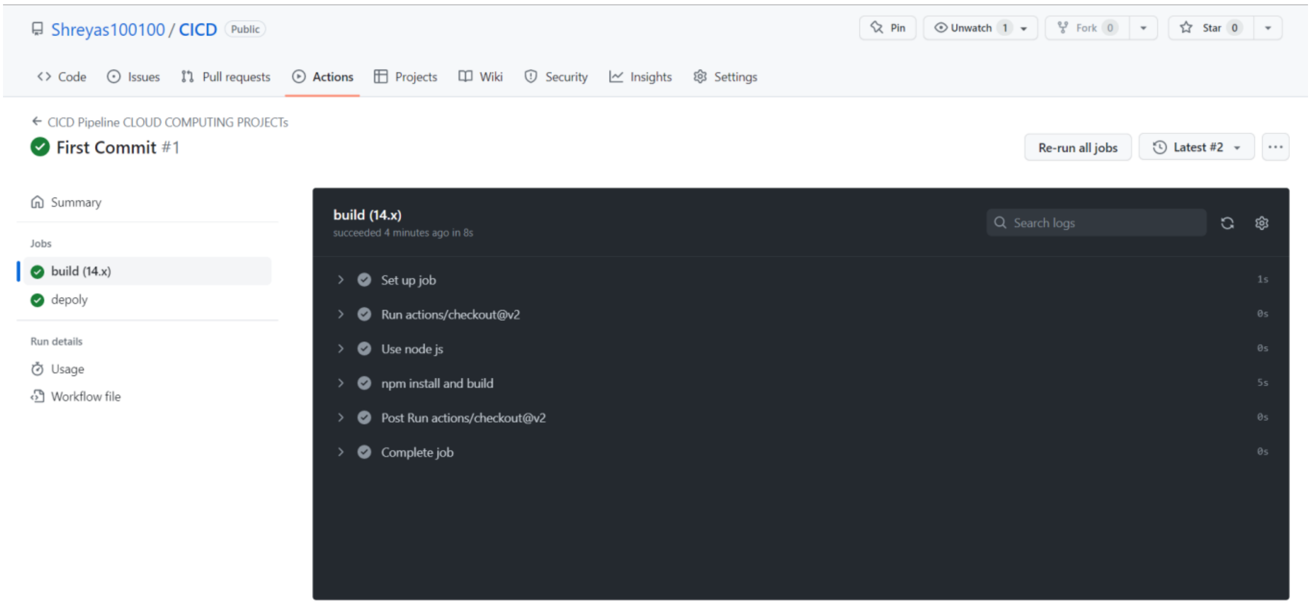
on: push

Matrix: build

1 job completed

Show all jobs

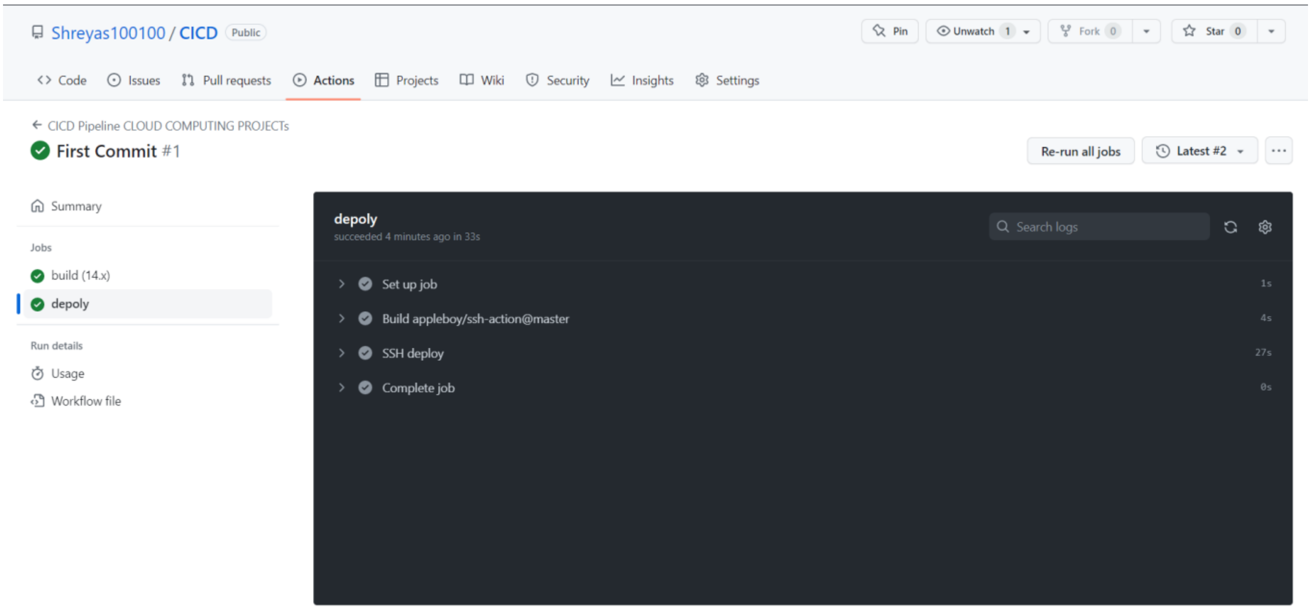
depoly 33s



First Commit #1

build (14.x)
succeeded 4 minutes ago in 8s

```
> ✓ Set up job                       1s
> ✓ Run actions/checkout@v2          0s
> ✓ Use node js                      0s
> ✓ npm install and build            5s
> ✓ Post Run actions/checkout@v2     0s
> ✓ Complete job                     0s
```

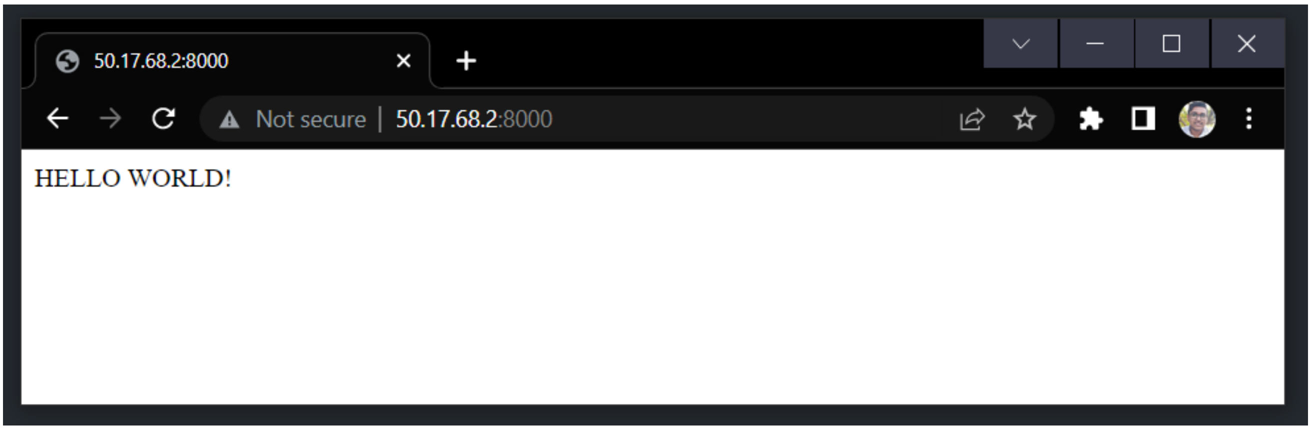


First Commit #1

depoly
succeeded 4 minutes ago in 33s

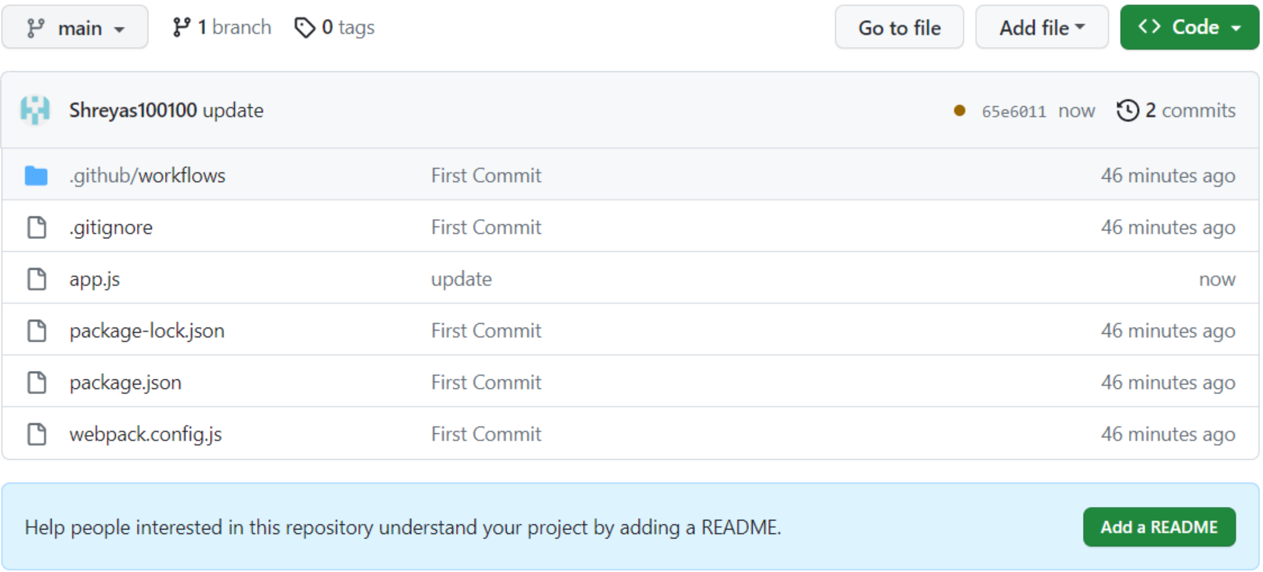
```
> ✓ Set up job                            1s
> ✓ Build appleboy/ssh-action@master      4s
> ✓ SSH deploy                           27s
> ✓ Complete job                          0s
```

-> As the Build and Deploy both are successful, it means that our CICD pipeline is successfully working and our Nodejs application is live on the port 8000 of our public IP Address.



HELLO WORLD!

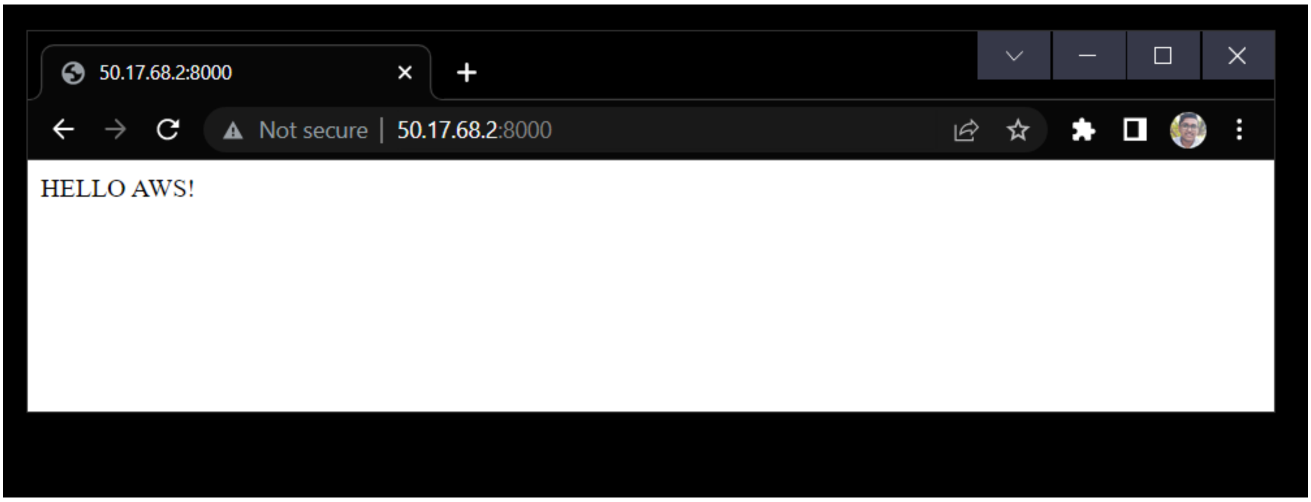
-> Once any collaborator updates the code and commit it to the GitHub Repository. Our CICD pipeline will get in action & after the build, deployment the changes will be reflected on the Public Ip port 8000.



Shreyas100100 update ● 65e6011 now 2 commits

.github/workflows	First Commit	46 minutes ago
.gitignore	First Commit	46 minutes ago
app.js	update	now
package-lock.json	First Commit	46 minutes ago
package.json	First Commit	46 minutes ago
webpack.config.js	First Commit	46 minutes ago

Help people interested in this repository understand your project by adding a README.



HELLO AWS!

<u>Conclusion:</u> We have successfully Deployed a CICD pipeline for Nodejs Application using GitHub Actions & AWS.