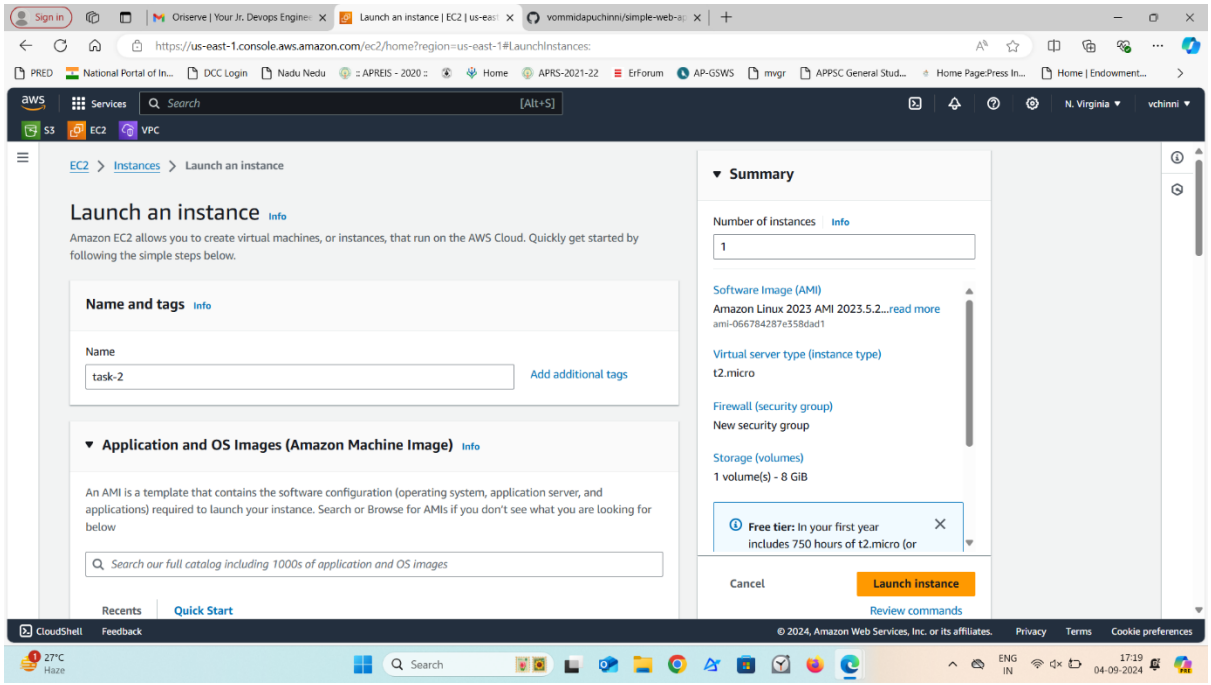
Write a shell script to create a file in multiple folder which contains a folder name oriserve only.

Git repo: https://github.com/vommidapuchinni/task-2-oriserve.git

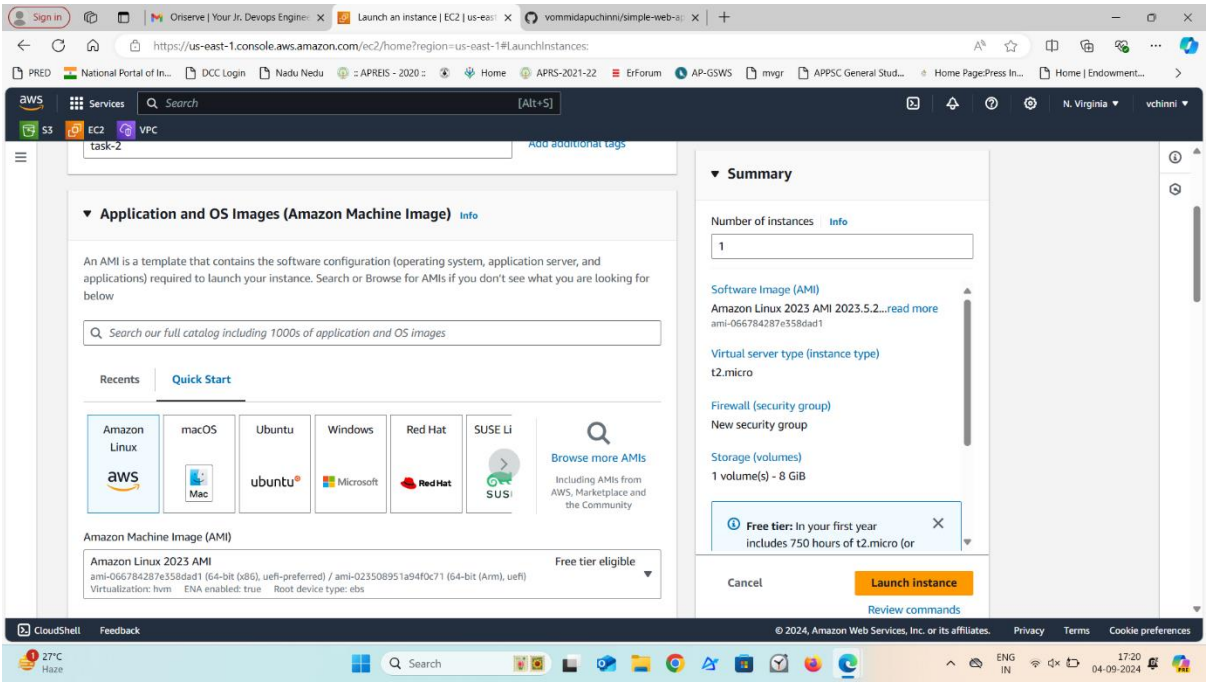
This shell script is designed to automate the creation of a specific directory structure and manage files within it. The script creates a projects directory containing multiple subdirectories, some of which contain a folder named oriserve. The script then finds all oriserve folders and creates a test.txt file inside each one, containing the text "oriserve".

Step 1: Set Up an AWS EC2 Instance

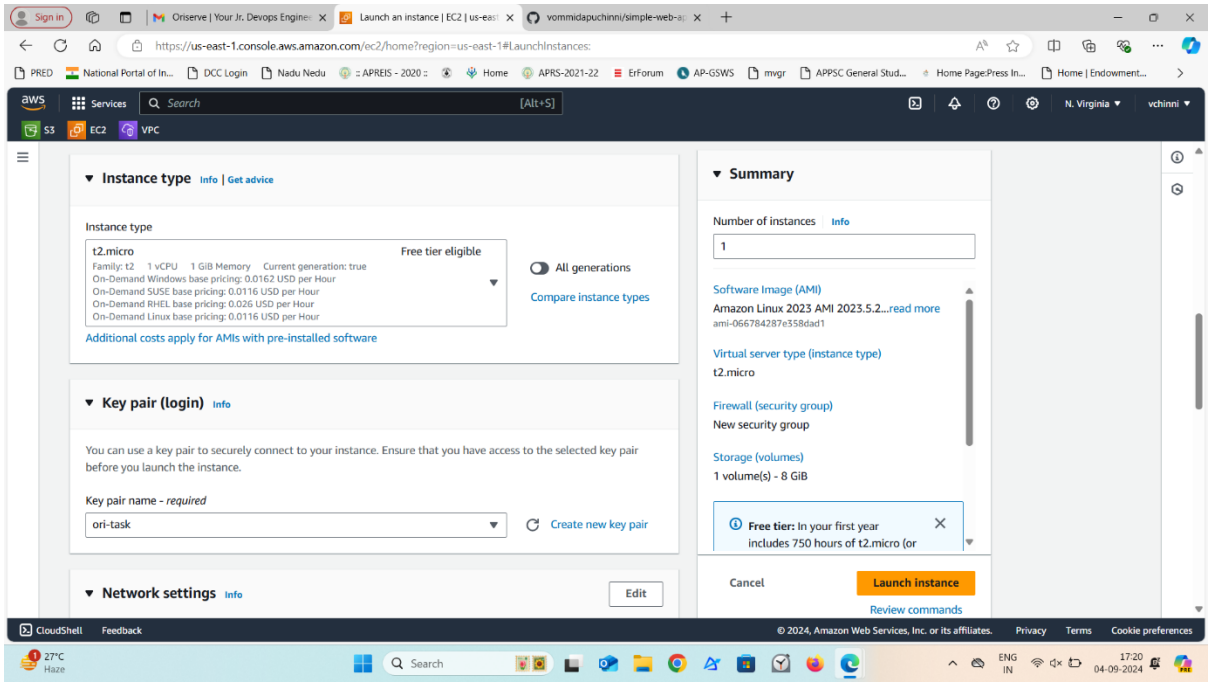
1. **Log in to AWS Management Console**:
    - Navigate to the AWS Management Console.
2. **Launch an EC2 Instance**:
    - Go to the **EC2 Dashboard**.



    - Click on **Launch Instance**.
    - Choose an **Amazon Machine Image (AMI)** (e.g., Amazon Linux 2).

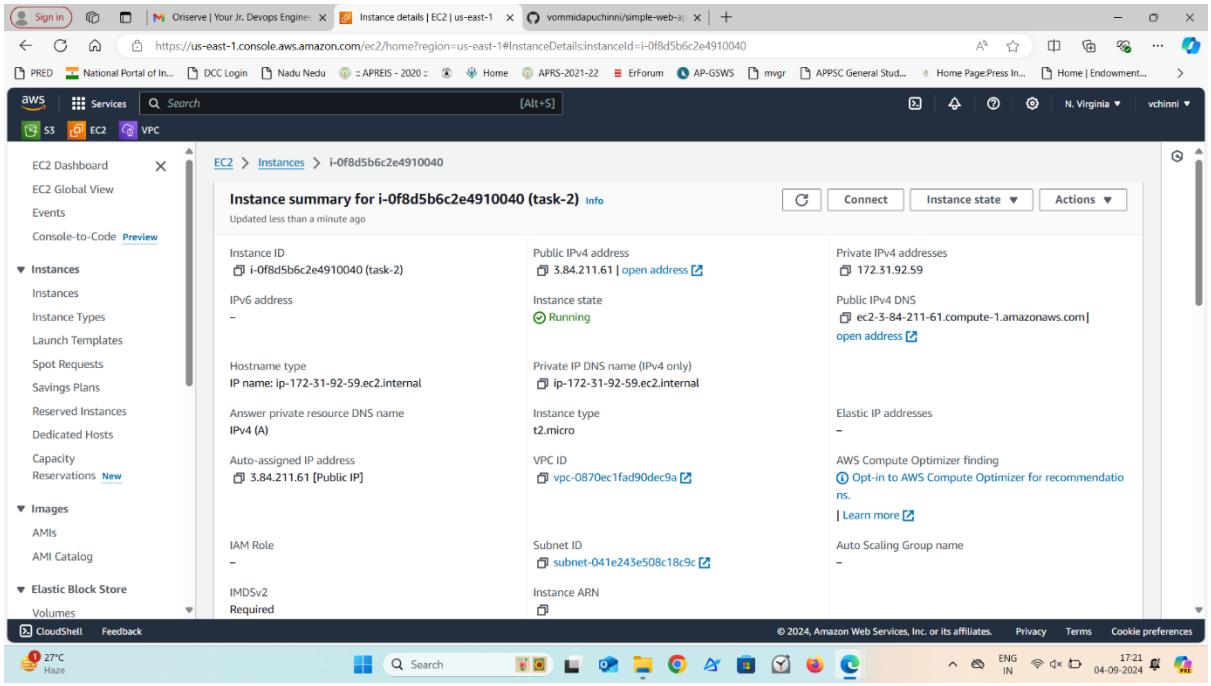


- Select an **Instance Type** (e.g., t2.micro for free tier eligibility).
- Choose or create a **Key Pair** (you'll use this to SSH into the instance).



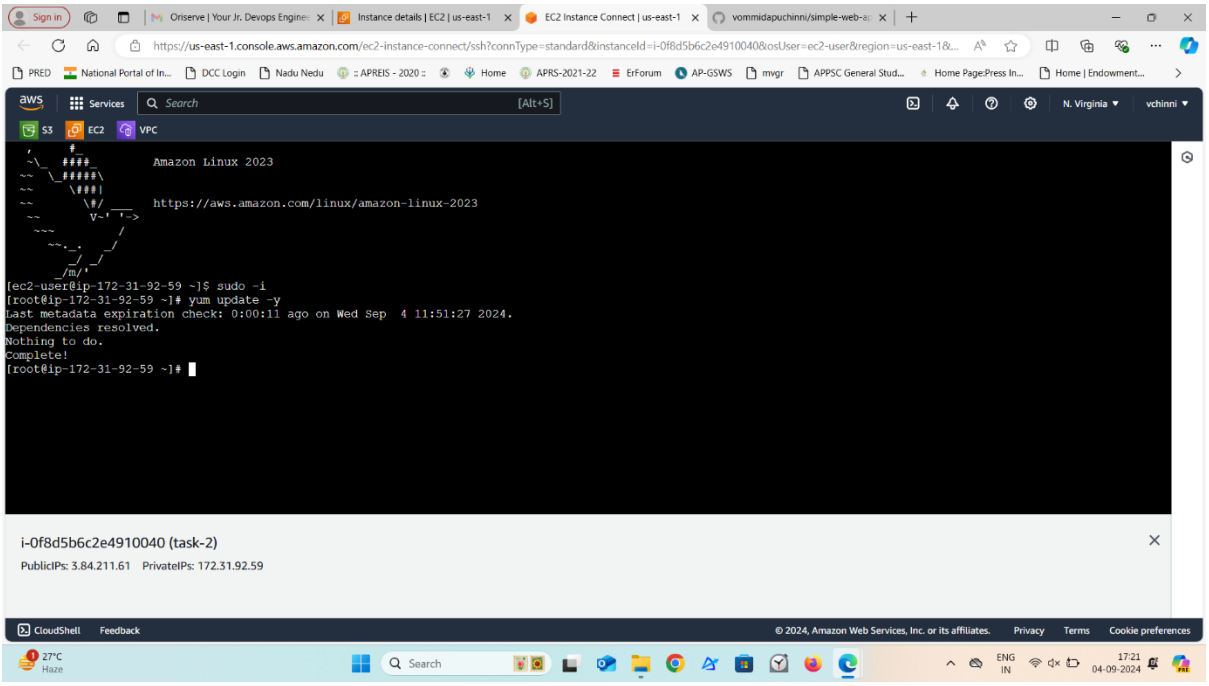
- Configure the **Security Group** to allow SSH (port 22) access from your IP.
- Review and **Launch** the instance.

3. **Connect to Your EC2 Instance**:
   - Once the instance is running, click **Connect**.
   - Follow the instructions to SSH into the instance using the .pem key file.
   - sudo -i



Step 2: Prepare the EC2 Instance Environment

1. **Update Packages**: yum update -y

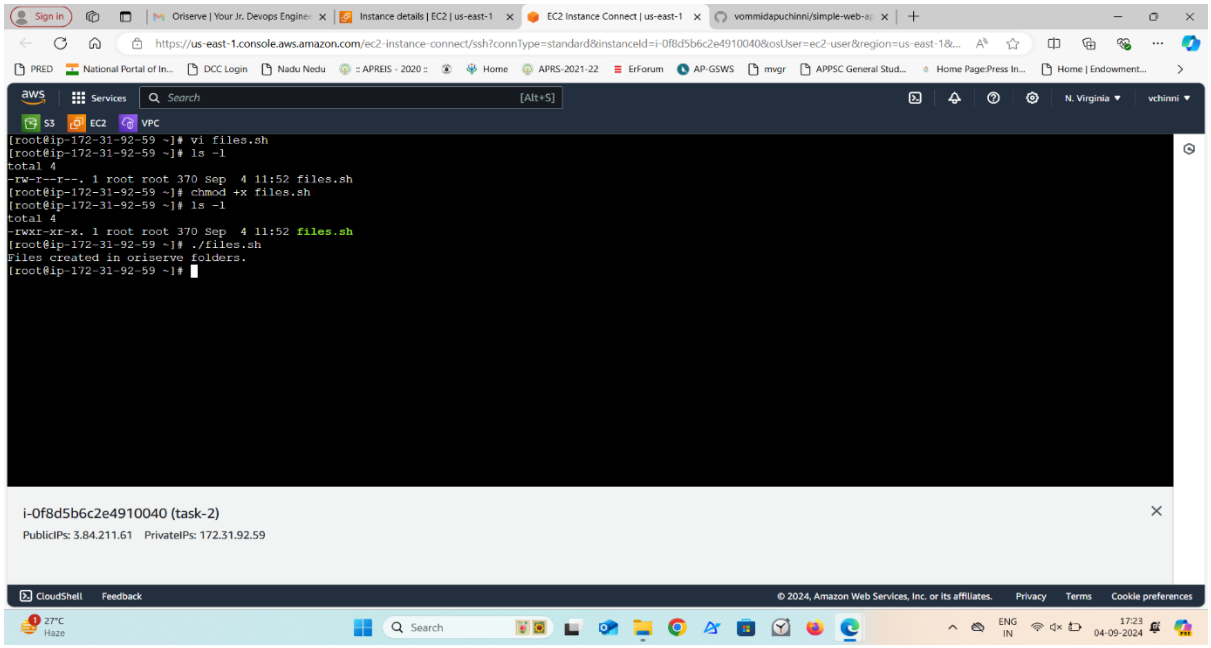


2. **Create the Shell Script**: vi files.sh
3. **Make the Script Executable**: chmod +x create_files.sh

Step 3: Execute the Shell Script

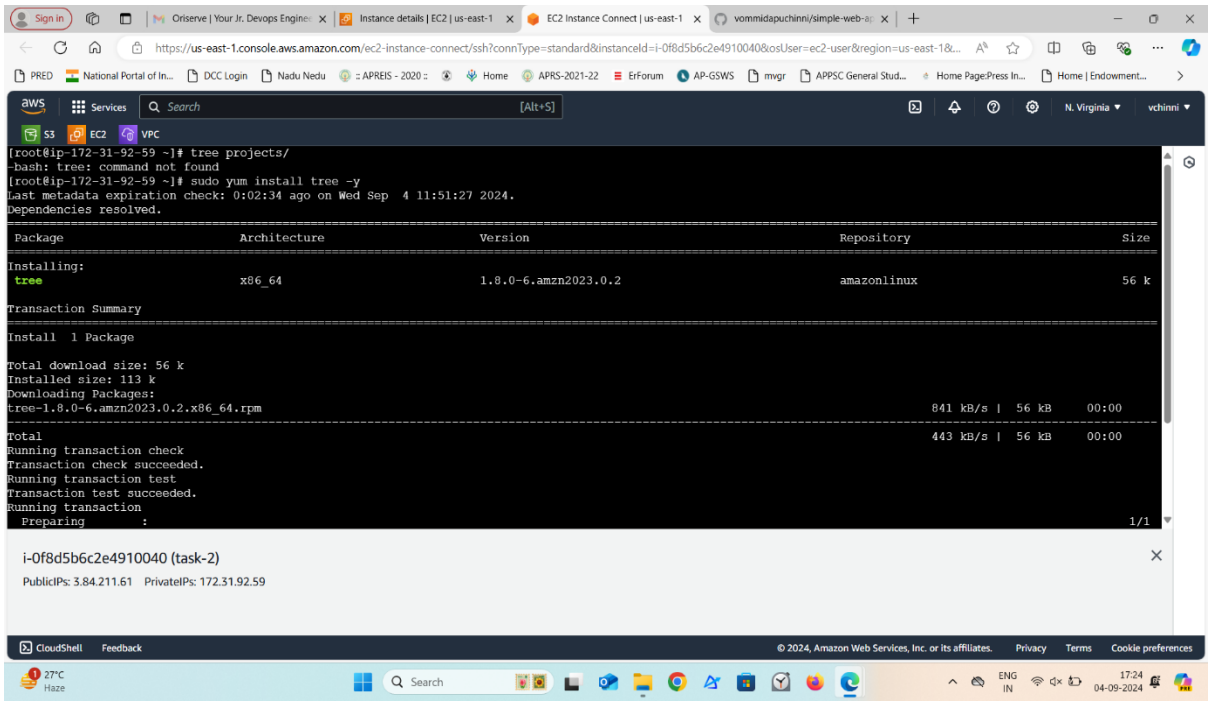
1. **Run the Script**:
    - Execute the script to create the directory structure and files:

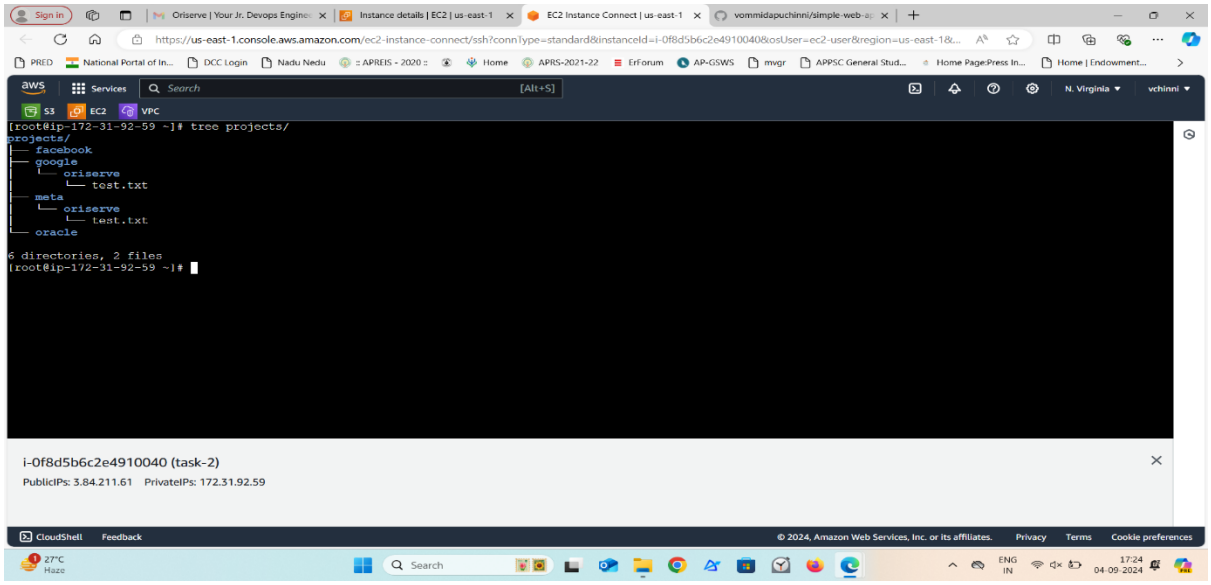
      ./create_files.sh

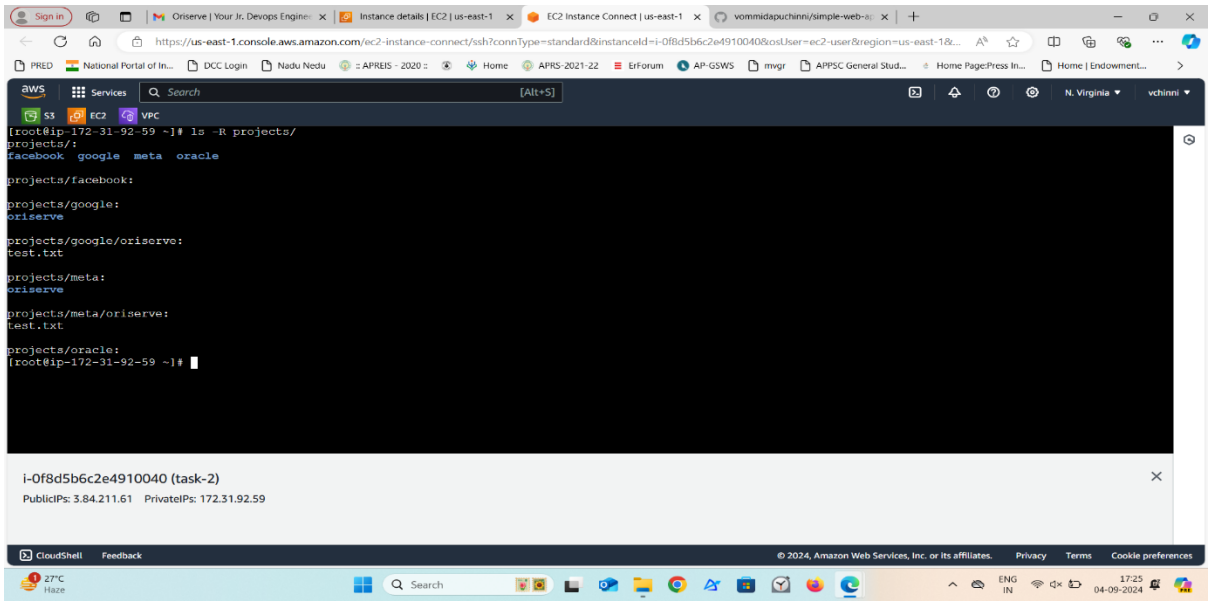


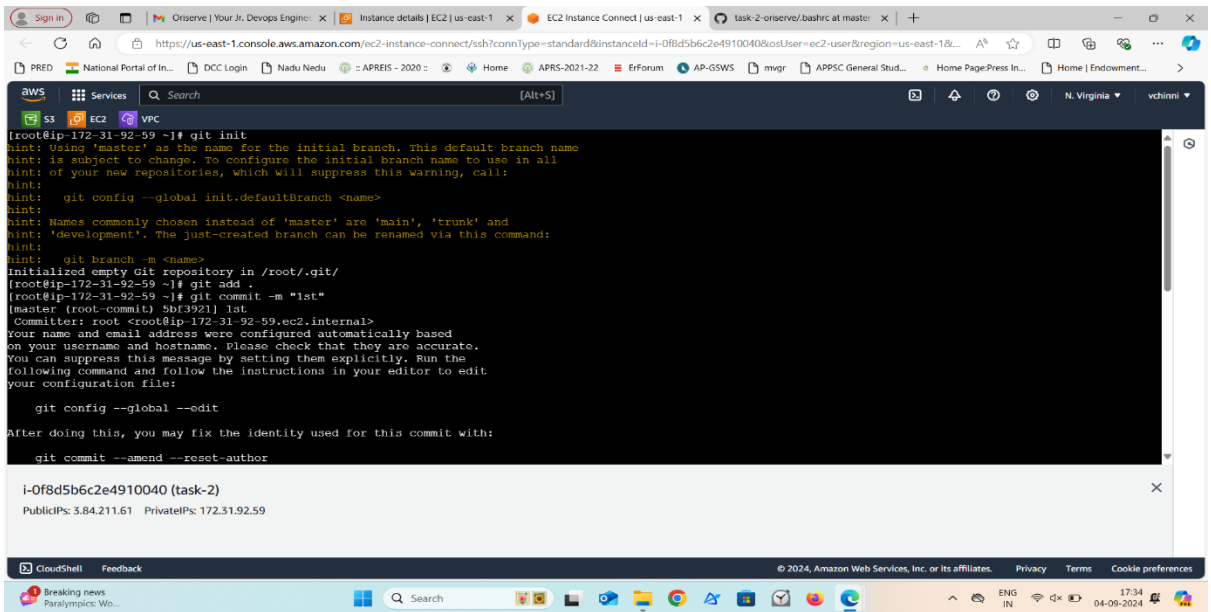
2. **Verify the Results**:
   - After running the script, check the directory structure:

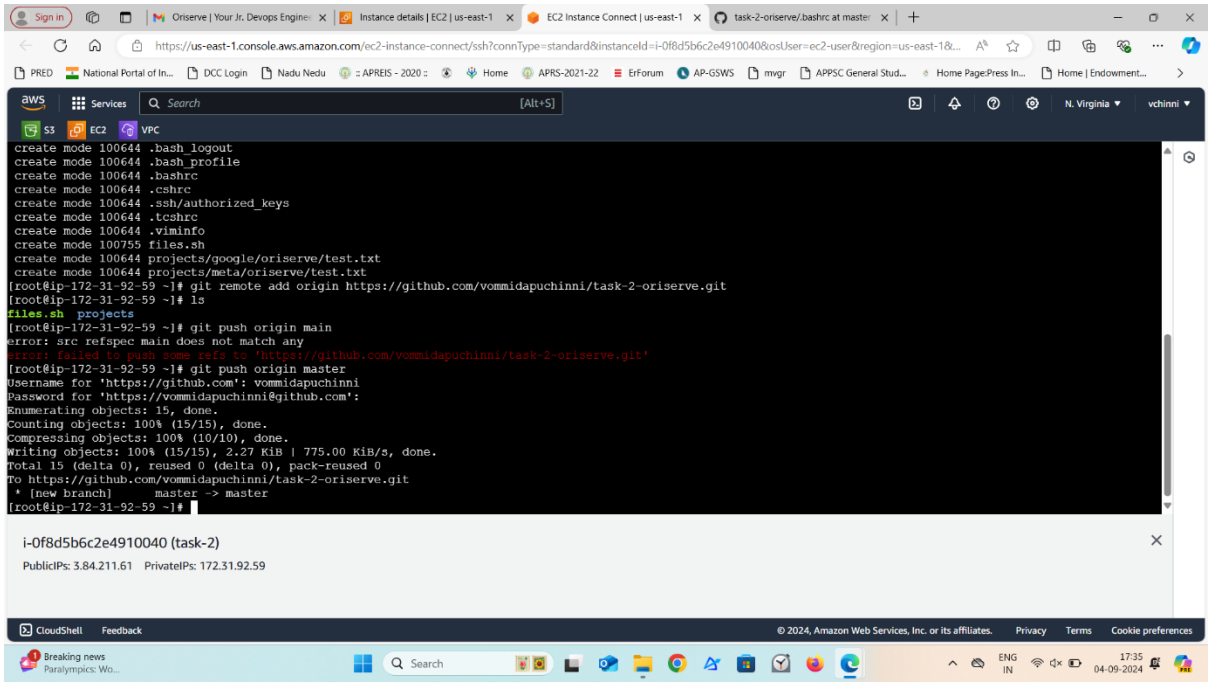
     tree projects/
   - The tree command shows a visual representation of the directory structure. If tree is not installed, you can use:
   - ls -R projects/




[root@ip-172-31-92-59 ~]# tree projects/
projects/
├── facebook
├── google
│   └── oriserve
│   └── test.txt
├── meta
│   └── oriserve
│   └── test.txt
└── oracle
6 directories, 2 files
[root@ip-172-31-92-59 ~]#
i-0f8d5b6c2e4910040 (task-2)
PublicIPs: 3.84.211.61 PrivateIPs: 172.31.92.59


[root@ip-172-31-92-59 ~]# ls -R projects/
projects/:
facebook google meta oracle
projects/facebook:
projects/google:
oriserve
projects/google/oriserve:
test.txt
projects/meta:
oriserve
projects/meta/oriserve:
test.txt
projects/oracle:
[root@ip-172-31-92-59 ~]#
i-0f8d5b6c2e4910040 (task-2)
PublicIPs: 3.84.211.61 PrivateIPs: 172.31.92.59


[root@ip-172-31-92-59 ~]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint: git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /root/.git/
[root@ip-172-31-92-59 ~]# git add .
[root@ip-172-31-92-59 ~]# git commit -m "1st"
[master (root-commit) 5bf3921] 1st
Committer: root <root@ip-172-31-92-59.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
i-0f8d5b6c2e4910040 (task-2)
PublicIPs: 3.84.211.61 PrivateIPs: 172.31.92.59



Conclusion:

The shell script successfully demonstrates an efficient method for managing directory structures and file creation. By automating these tasks, it reduces manual effort and ensures accuracy. This approach is particularly useful for streamlining repetitive setup processes, making it easier to maintain consistent environments and manage files effectively.