

User Manual

Users of this project are DevOps engineers who wants to take look how we can develop software with the help of DevOps methodology and the industries who want to shift from agile to DevOps can also refer this project manual to create and follows the DevOps principal to develop and deliver software.

6.1 Prerequisite to be installed in local system.

1. Git

Steps to install git:-

- i) Browse url <https://git-scm.com/downloads>

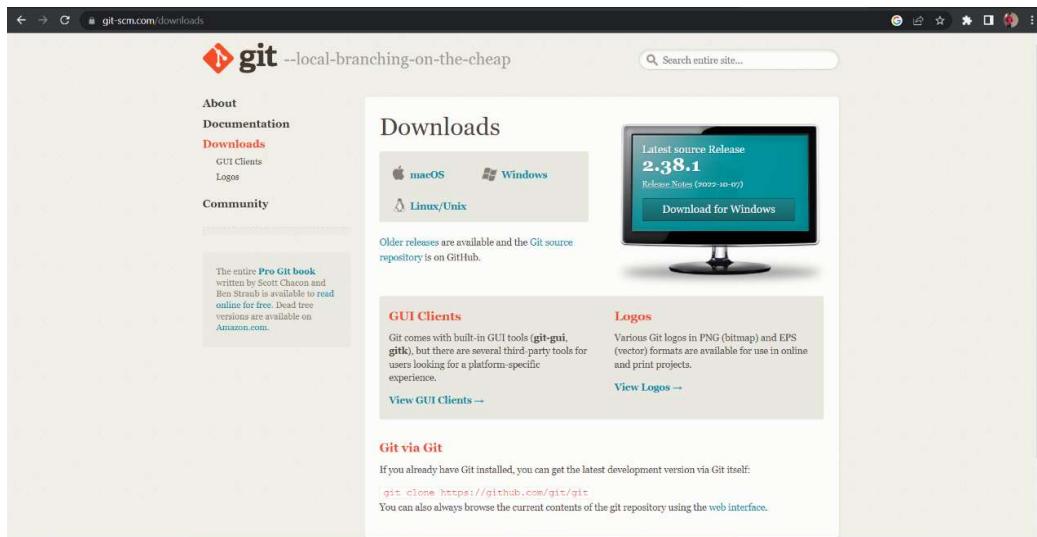


Figure 6.1.1 (git download page)

- ii) Click on the operation system which you are using

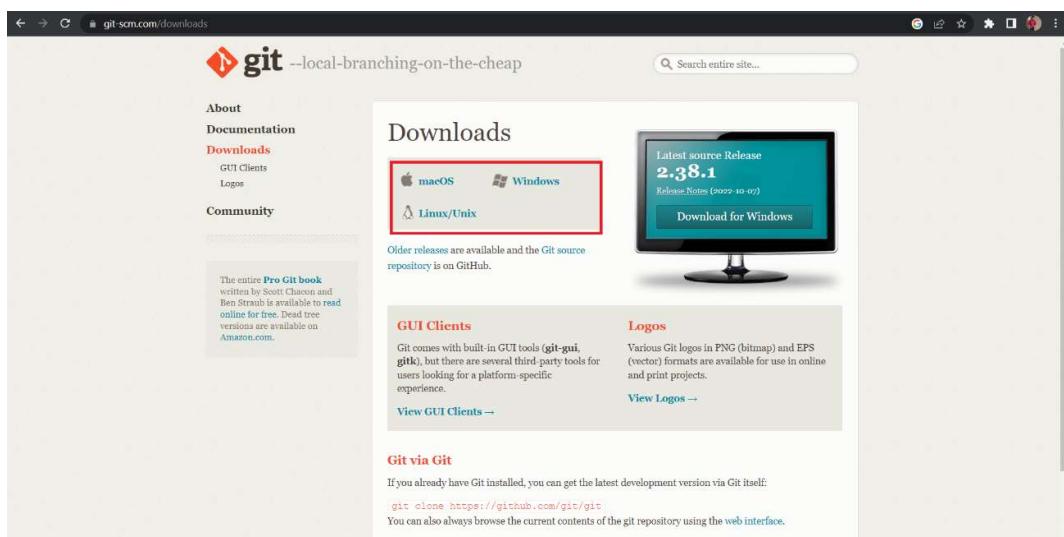


Figure 6.1.2 (Choose OS)

iii) Click on click here to download to start downloading the git

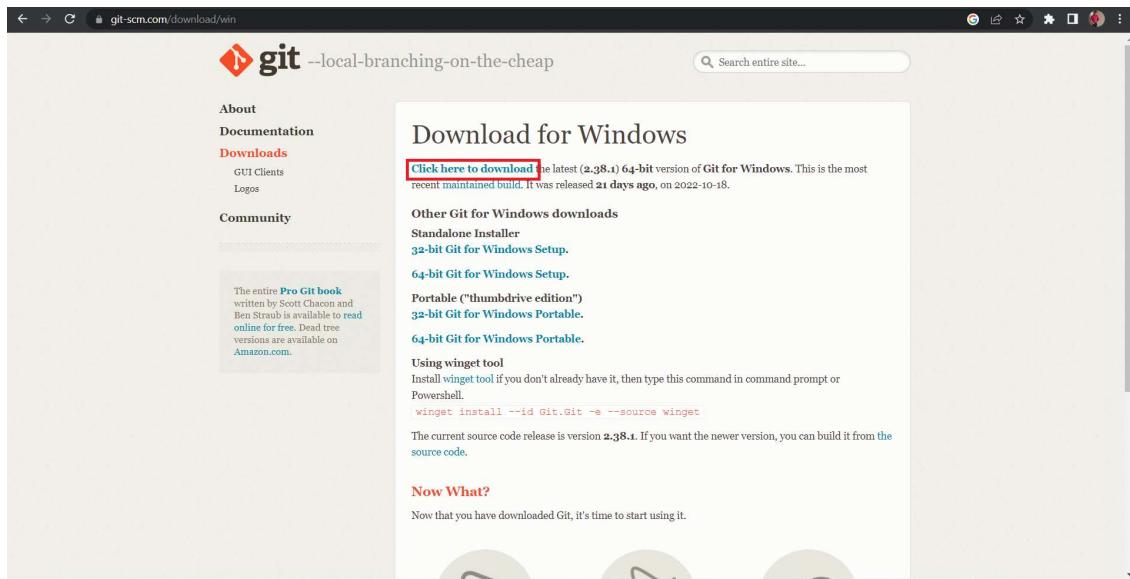


Figure 6.1.3 (download git)

iv) Wait for git to be downloaded

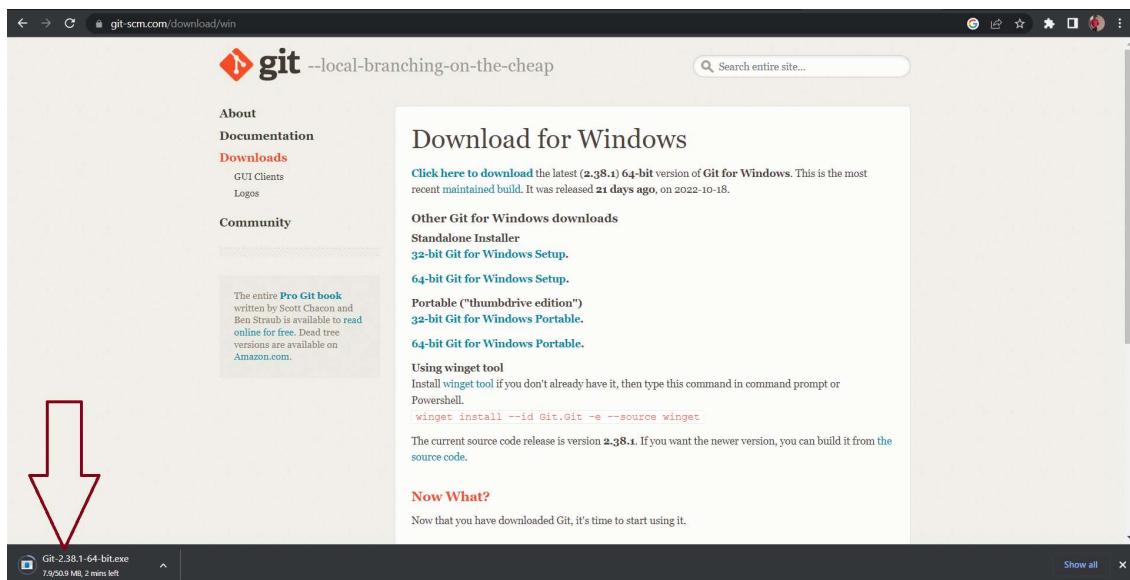


Figure 6.1.4 (wait for download)

v) Now open the folder where git is downloaded

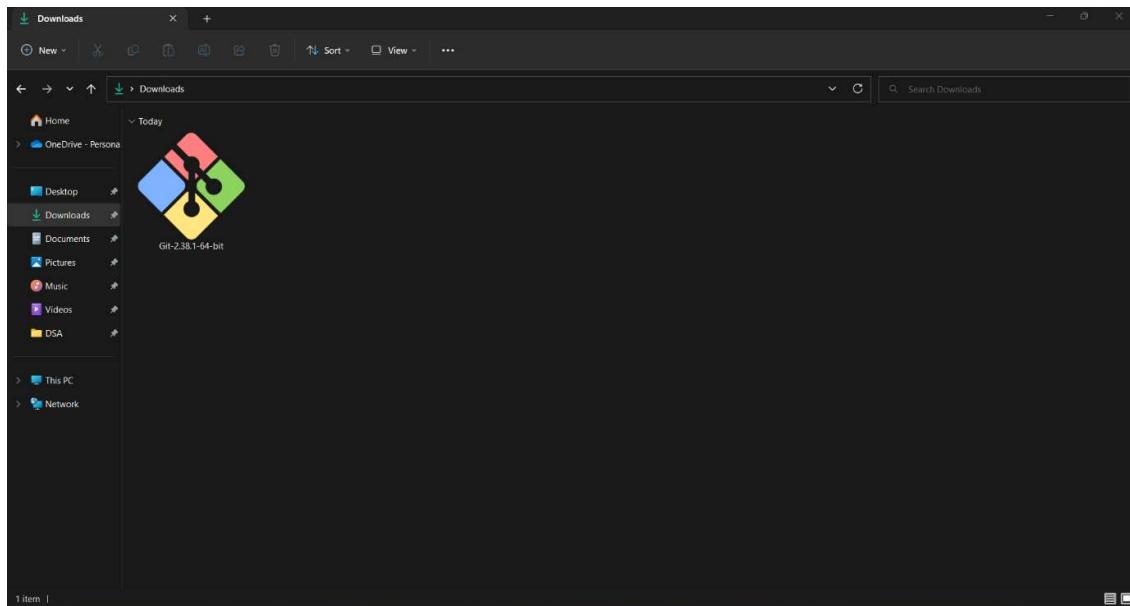


Figure 6.1.5 (open folder)

vi) Double click on the git icon and install git on the machine

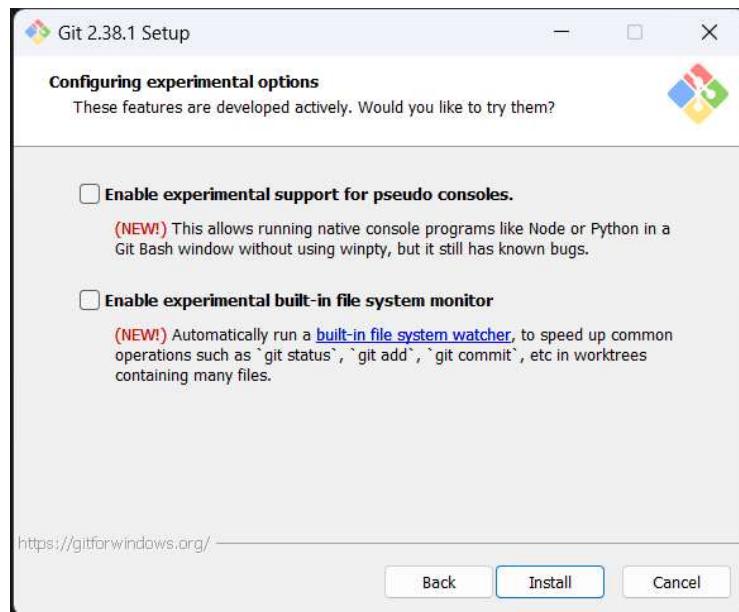


Figure 6.1.6 (Install Git)

vii) Wait for installation to finish

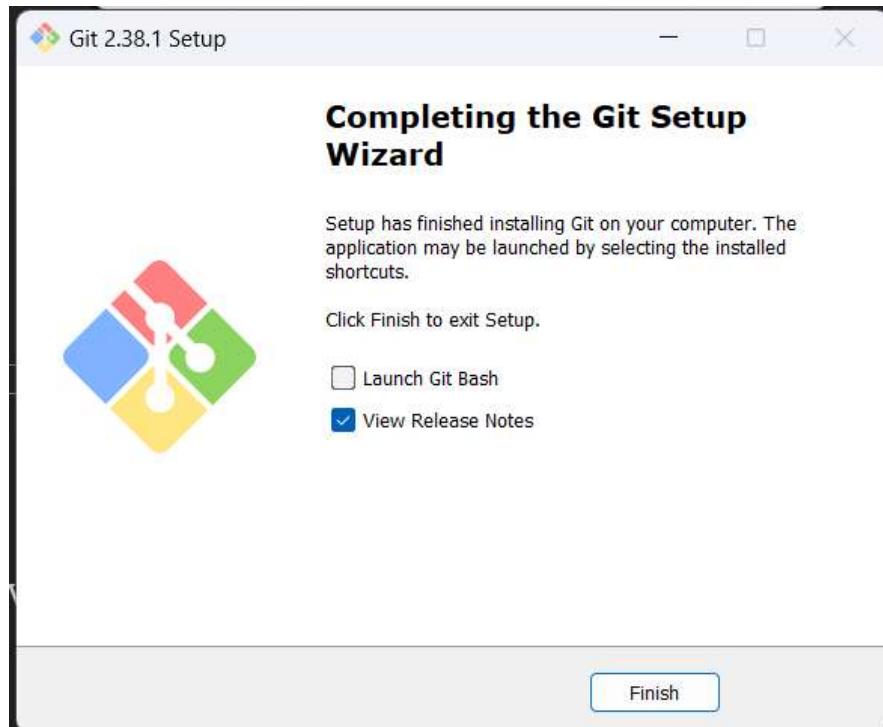


Figure 6.1.7 (Finish Installation)

viii) Open CMD and run git --version to check if git is properly installed

A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command "git --version" was run, and the output shows "git version 2.38.1.windows.1". The window has a dark theme.

Figure 6.1.8 (Check version of git)

2. VirtualBox version lesser than 7

Steps to install VirtualBox

i) Browse URL

https://www.virtualbox.org/wiki/Download_Old_Builds_6_1

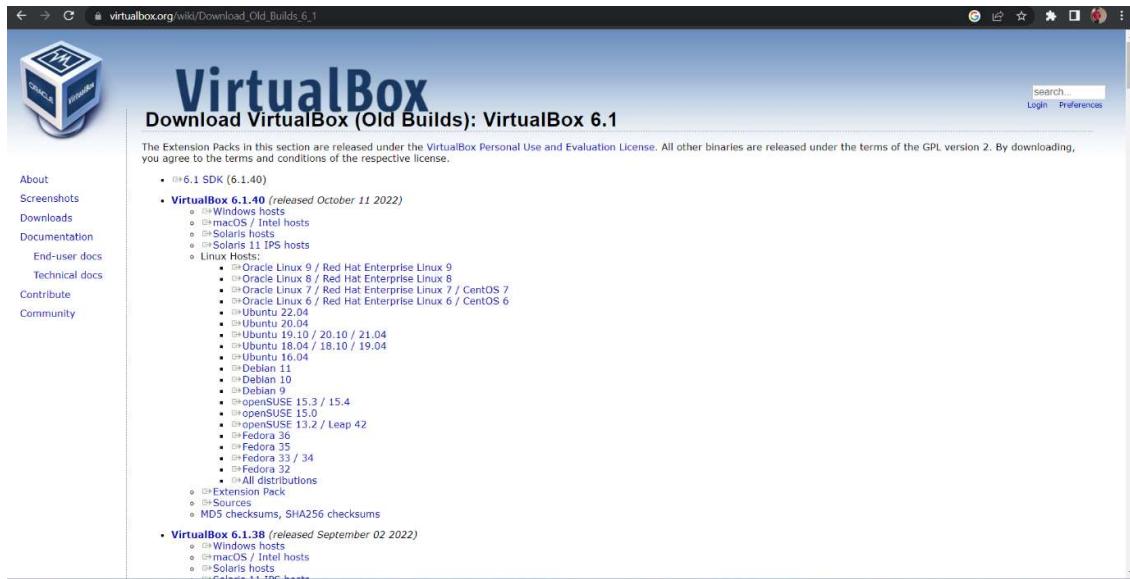


Figure 6.1.9 (open VirtualBox Download page)

ii) Click on the host according to your machine for me it's windows hosts, and wait for download to be completed

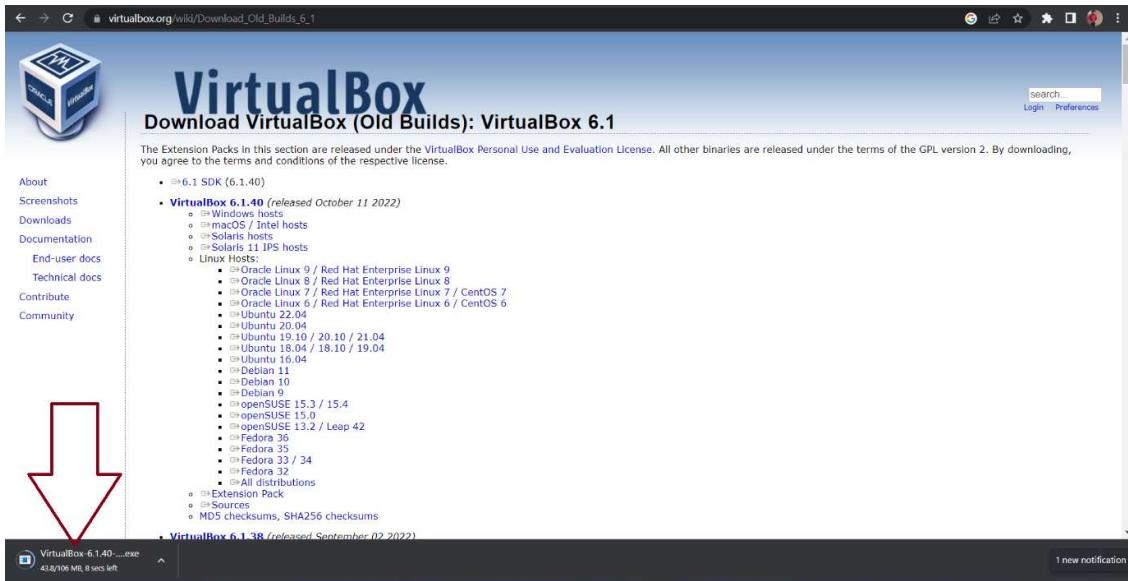


Figure 6.1.10 (Download VirtualBox)

iii) Open folder where VirtualBox is downloaded

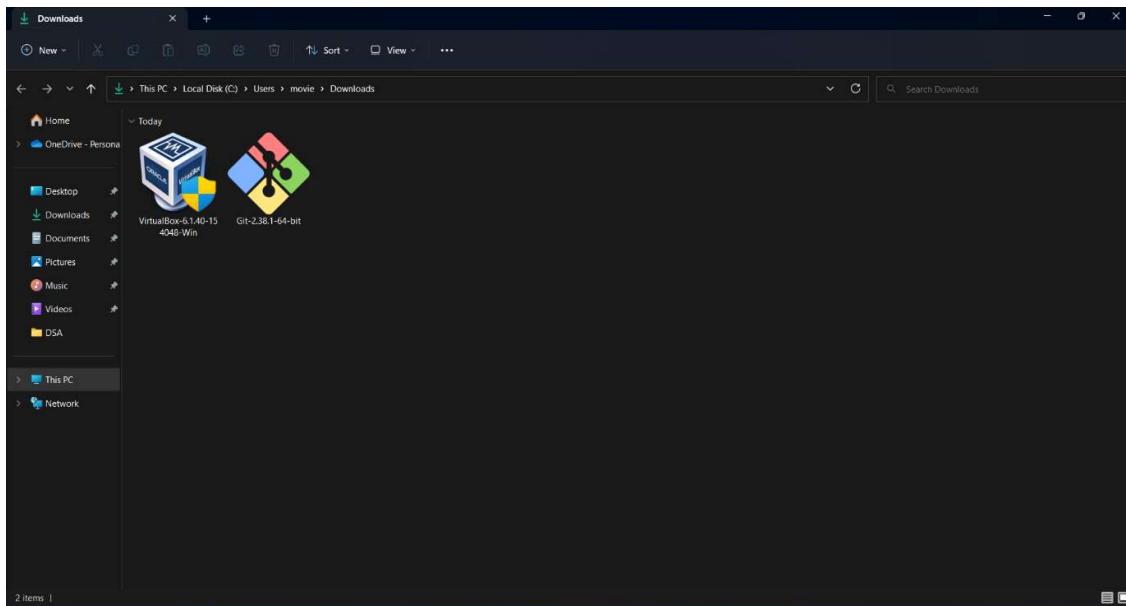


Figure 6.1.11 (Open download Folder)

- iv) Double click on VirtualBox icon and install virtual box

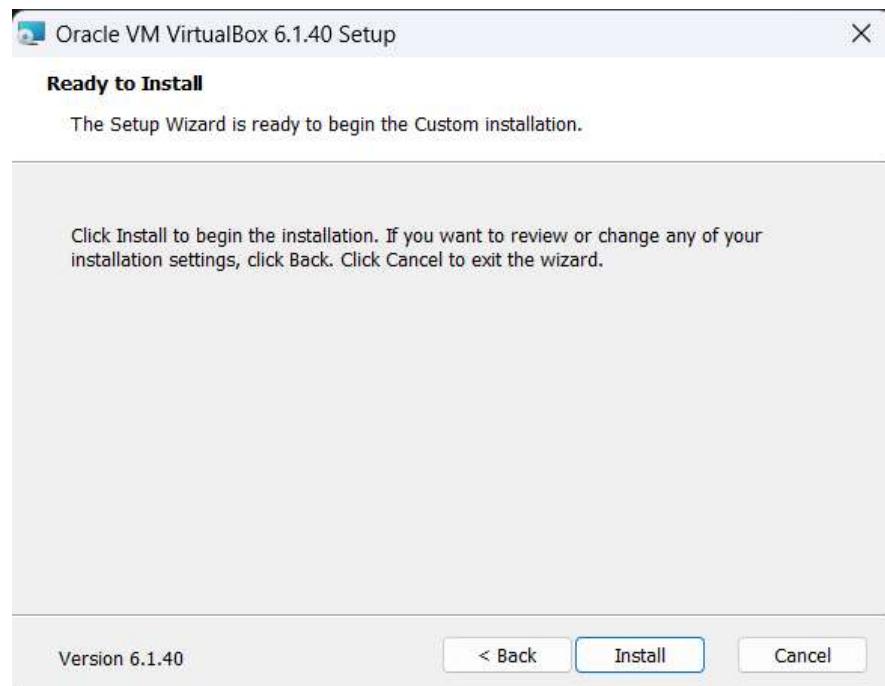


Figure 6.1.12 (install VirtualBox)

- v) Wait for installation to be finished



Figure 6.1.13 (Installation Finished)

vi) Check VirtualBox is installed properly or not

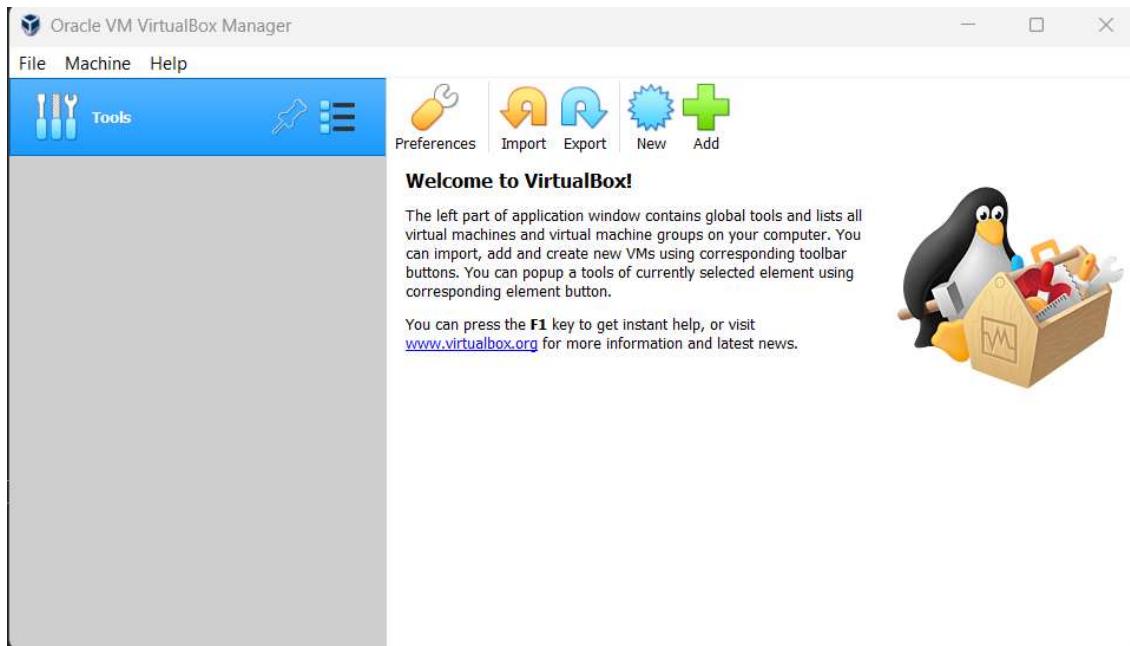


Figure 6.1.14 (Installed VirtualBox)

3. Chocolatey package manager

Steps to install Chocolatey package manager for windows

- i) Browse URL <https://chocolatey.org/install>

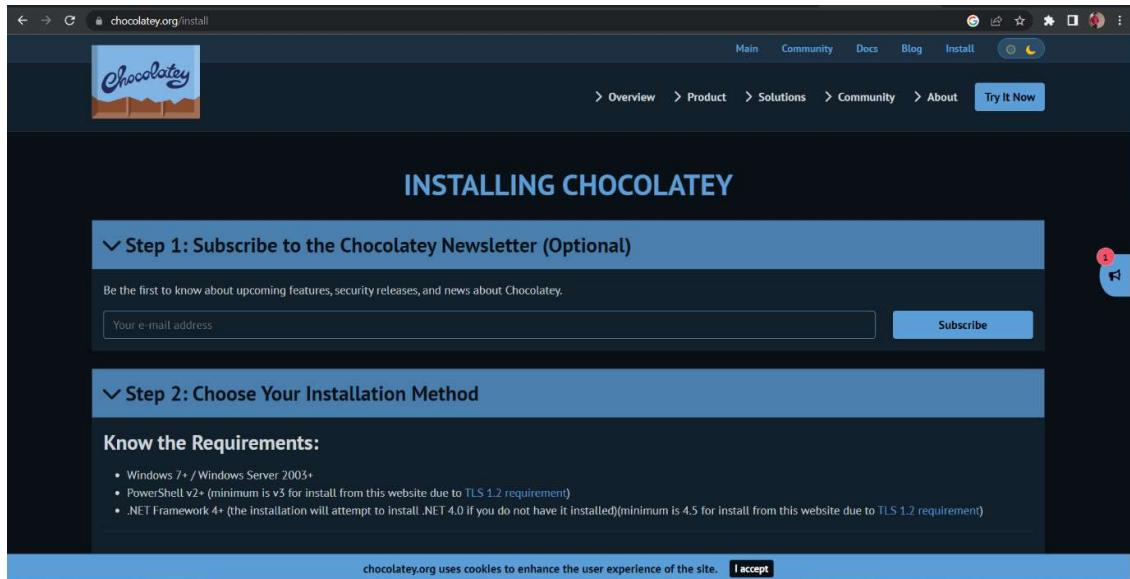


Figure 6.1.15 (Open Chocolatey installation page)

- ii) Scroll down and copy the installation command

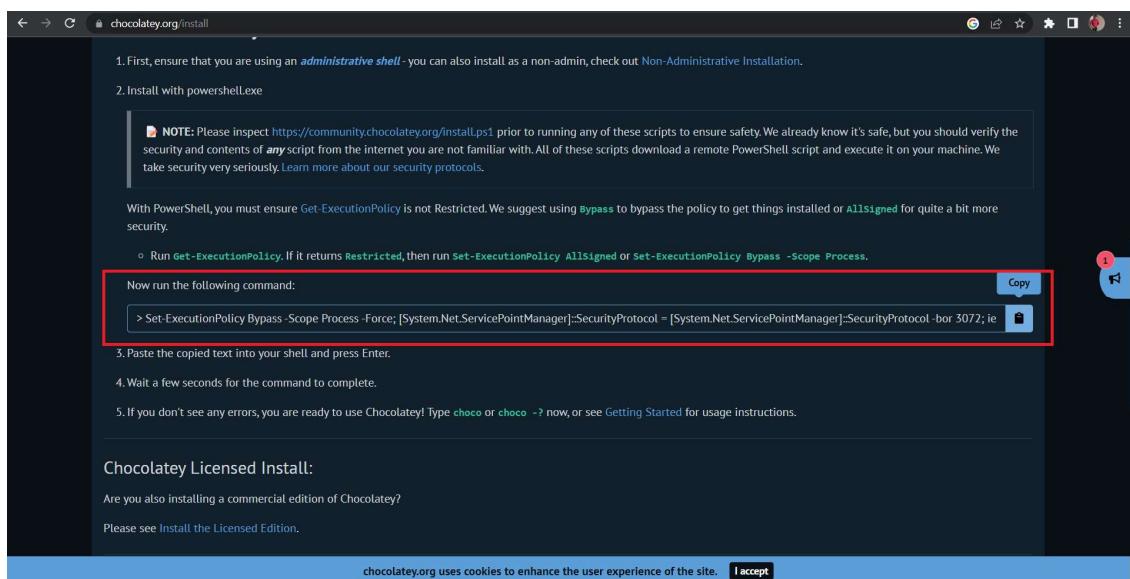


Figure 6.1.16 (copy installation command)

iii) Open PowerShell with administrative privileges

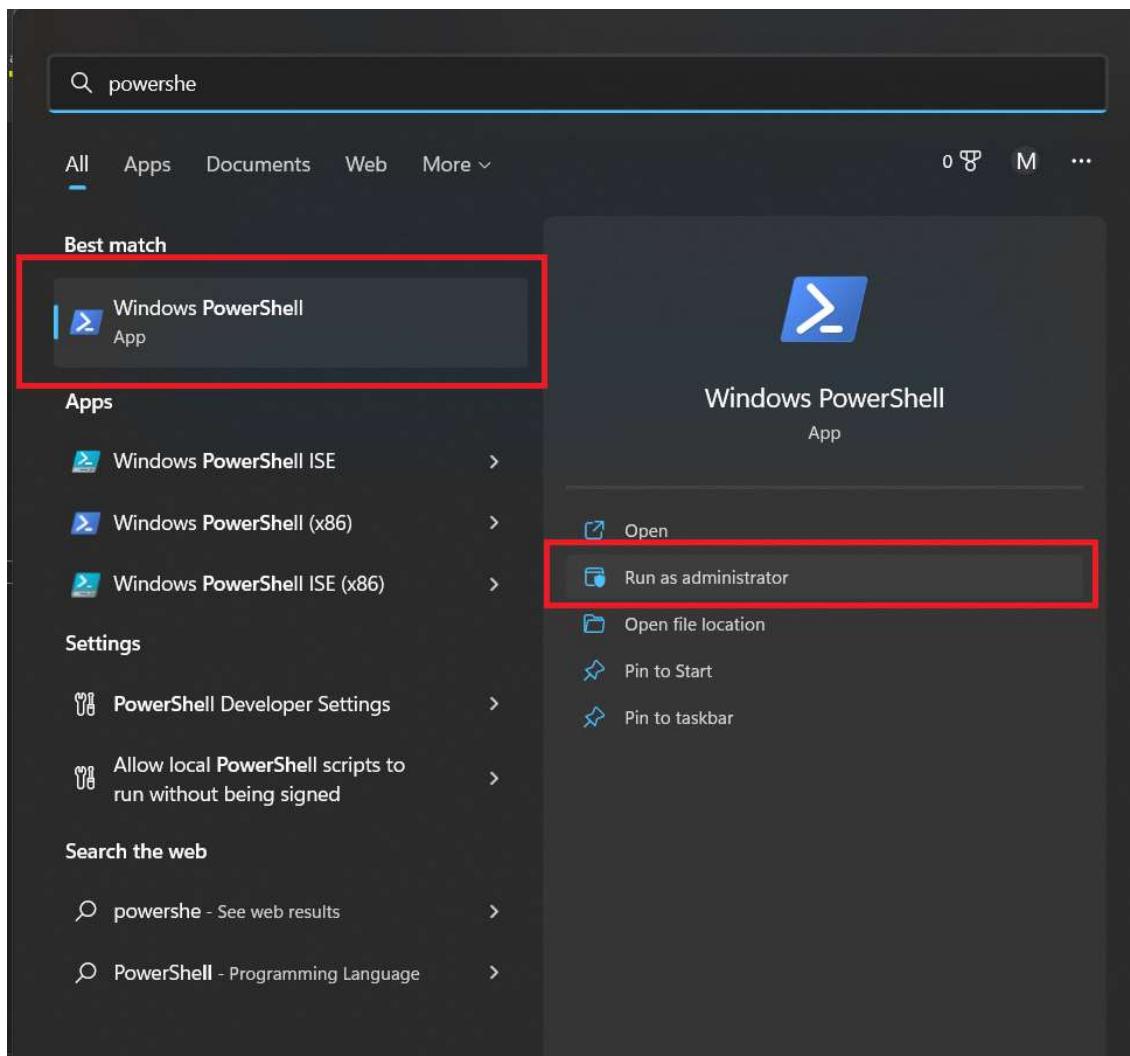
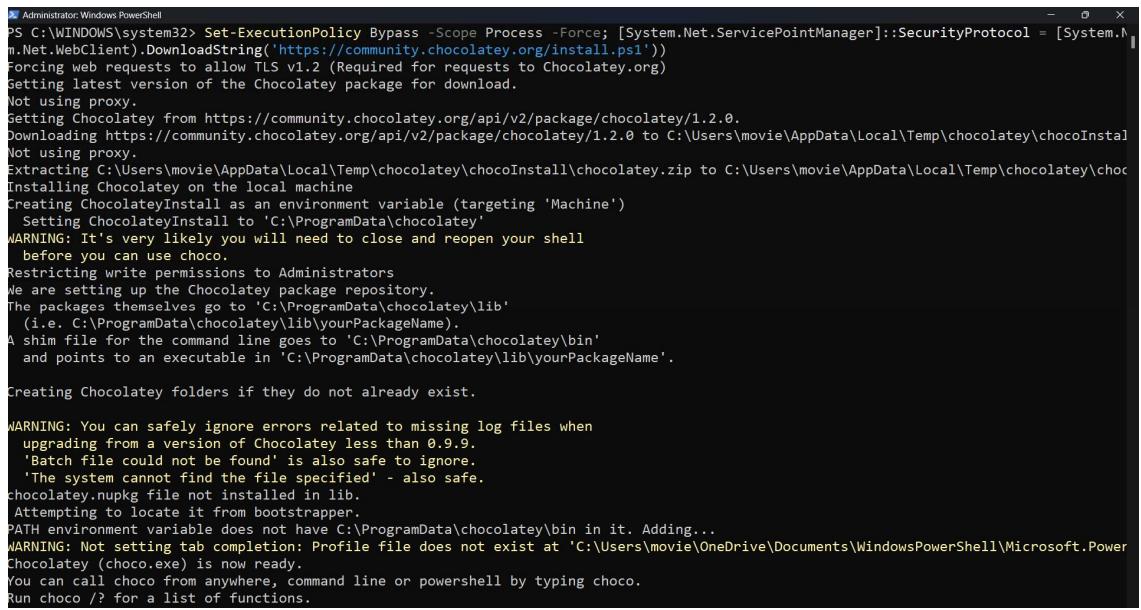


Figure 6.1.17 (open powershell)

iv) Paste the copied link on PowerShell command line and press enter



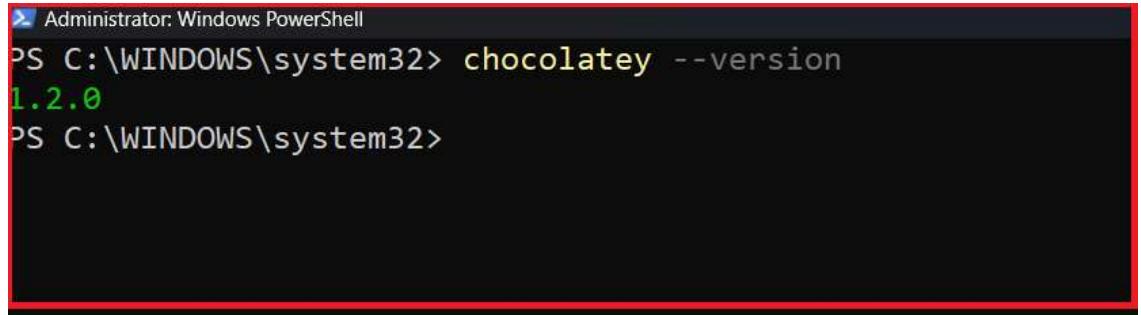
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.N
n.Net.WebClient].DownloadString('https://community.chocolatey.org/install.ps1')
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/1.2.0.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/1.2.0 to C:\Users\movie\AppData\Local\Temp\chocolatey\chocoInstall
Not using proxy.
Extracting C:\Users\movie\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\movie\AppData\Local\Temp\chocolatey\choc
Installing Chocolatey on the local machine
Creating ChocolateyInstall as an environment variable (targeting 'Machine')
Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib',
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.
Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'Batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at 'C:\Users\movie\OneDrive\Documents\WindowsPowerShell\Microsoft.Power
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
```

Figure 6.1.18 (install chocolatey)

v) Check version of chocolatey to ensure if chocolatey properly installed

or not.



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> chocolatey --version
1.2.0
PS C:\WINDOWS\system32>
```

Figure 6.1.19 (check version of chocolatey)

4. terraform and vagrant

Steps to install terraform and vagrant using chocolatey package manager

- i) Open PowerShell with administrative access

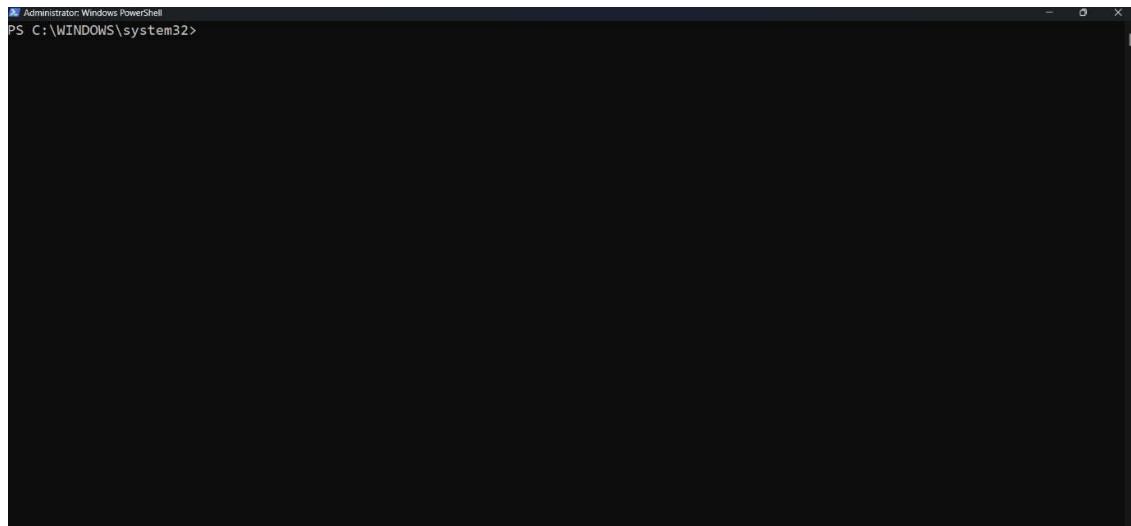


Figure 6.1.20 (Open PowerShell)

- ii) Run command choco install terraform vagrant -y

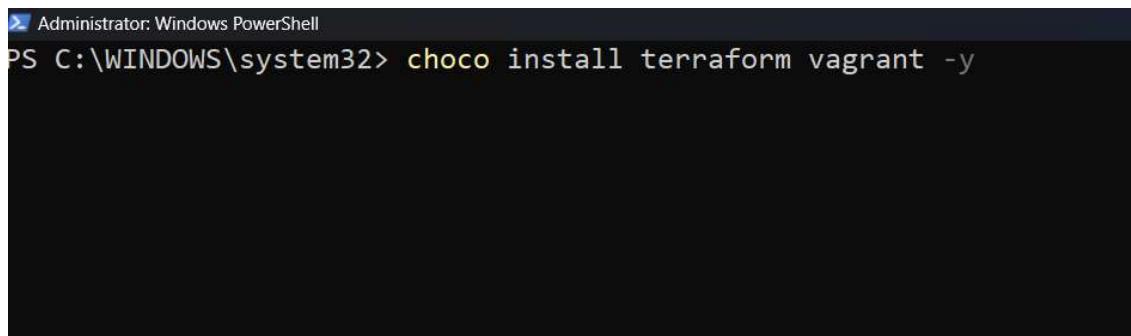
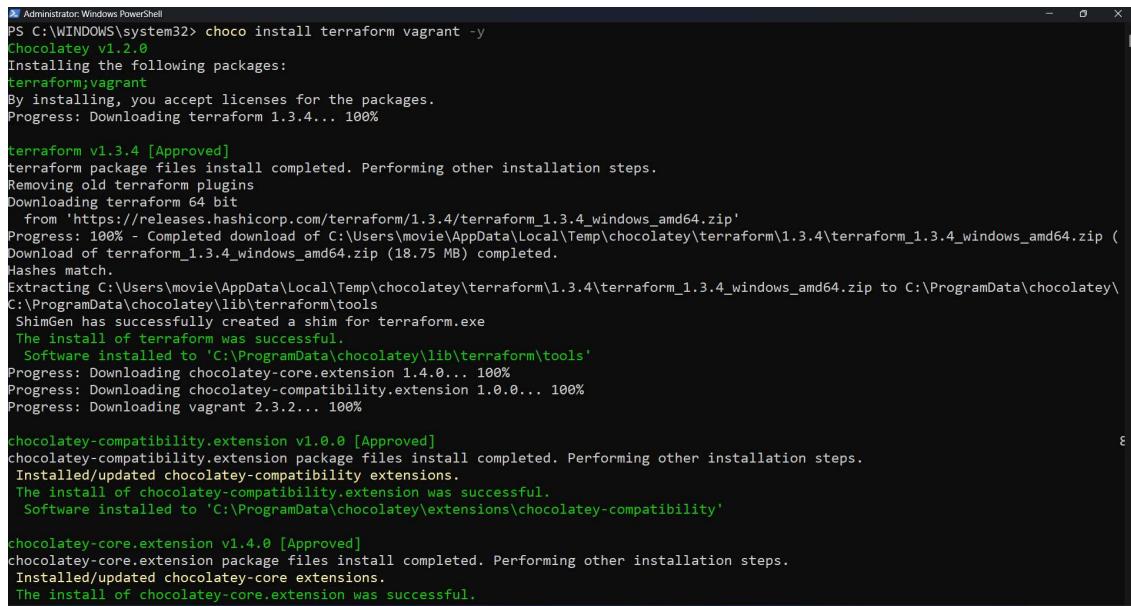


Figure 6.1.21 (install terraform and vagrant)

iii) Wait for installation of both packages



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco install terraform vagrant -y
Chocolatey v1.2.0
Installing the following packages:
terraform;vagrant
By installing, you accept licenses for the packages.
Progress: Downloading terraform 1.3.4... 100%

terraform v1.3.4 [Approved]
terraform package files install completed. Performing other installation steps.
Removing old terraform plugins
Downloading terraform 64 bit
  from 'https://releases.hashicorp.com/terraform/1.3.4/terraform_1.3.4_windows_amd64.zip'
Progress: 100% - Completed download of C:\Users\movie\AppData\Local\Temp\chocolatey\terraform\1.3.4\terraform_1.3.4_windows_amd64.zip (Download of terraform_1.3.4_windows_amd64.zip (18.75 MB) completed.
Hashes match.
Extracting C:\Users\movie\AppData\Local\Temp\chocolatey\terraform\1.3.4\terraform_1.3.4_windows_amd64.zip to C:\ProgramData\chocolatey\
C:\ProgramData\chocolatey\lib\terraform\tools
ShimGen has successfully created a shim for terraform.exe
The install of terraform was successful.
  Software installed to 'C:\ProgramData\chocolatey\lib\terraform\tools'

Progress: Downloading chocolatey-core.extension 1.4.0... 100%
Progress: Downloading chocolatey-compatibility.extension 1.0.0... 100%
Progress: Downloading vagrant 2.3.2... 100%

chocolatey-compatibility.extension v1.0.0 [Approved]
chocolatey-compatibility.extension package files install completed. Performing other installation steps.
  Installed/updated chocolatey-compatibility extensions.
The install of chocolatey-compatibility.extension was successful.
  Software installed to 'C:\ProgramData\chocolatey\extensions\chocolatey-compatibility'

chocolatey-core.extension v1.4.0 [Approved]
chocolatey-core.extension package files install completed. Performing other installation steps.
  Installed/updated chocolatey-core extensions.
The install of chocolatey-core.extension was successful.
```

Figure 6.1.22 (install terraform and vagrant)

iv) Installation of vagrant requires reboot of machine

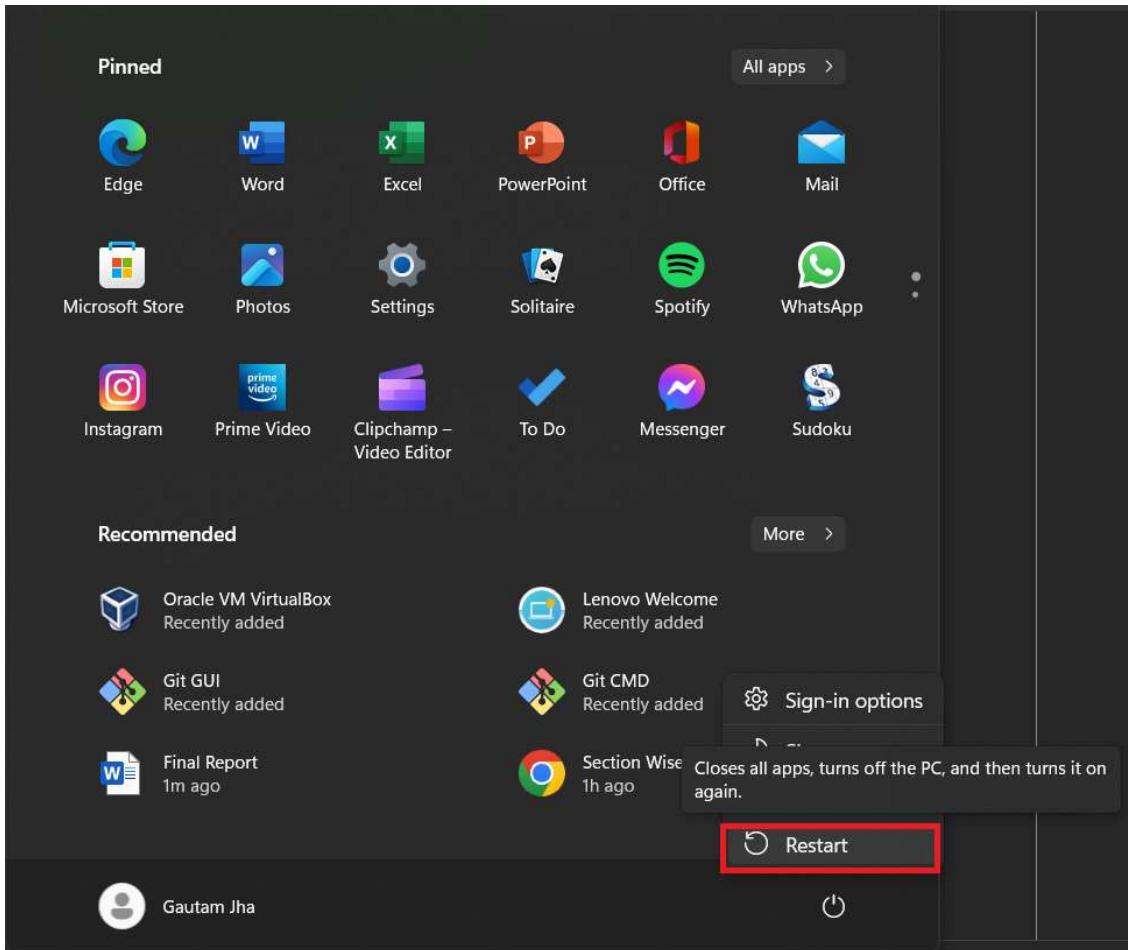


Figure 6.1.23 (Restart PC)

- v) Check the version of both packages to ensure both are properly installed

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> terraform --version
Terraform v1.3.4
on windows_amd64
PS C:\WINDOWS\system32> vagrant --version
Vagrant 2.3.2
PS C:\WINDOWS\system32>
```

Figure 6.1.24 (Checking version of terraform and vagrant)

5. minikube and kubectl

Steps to install minikube and kubectl

- i) Open PowerShell with administrative privileges

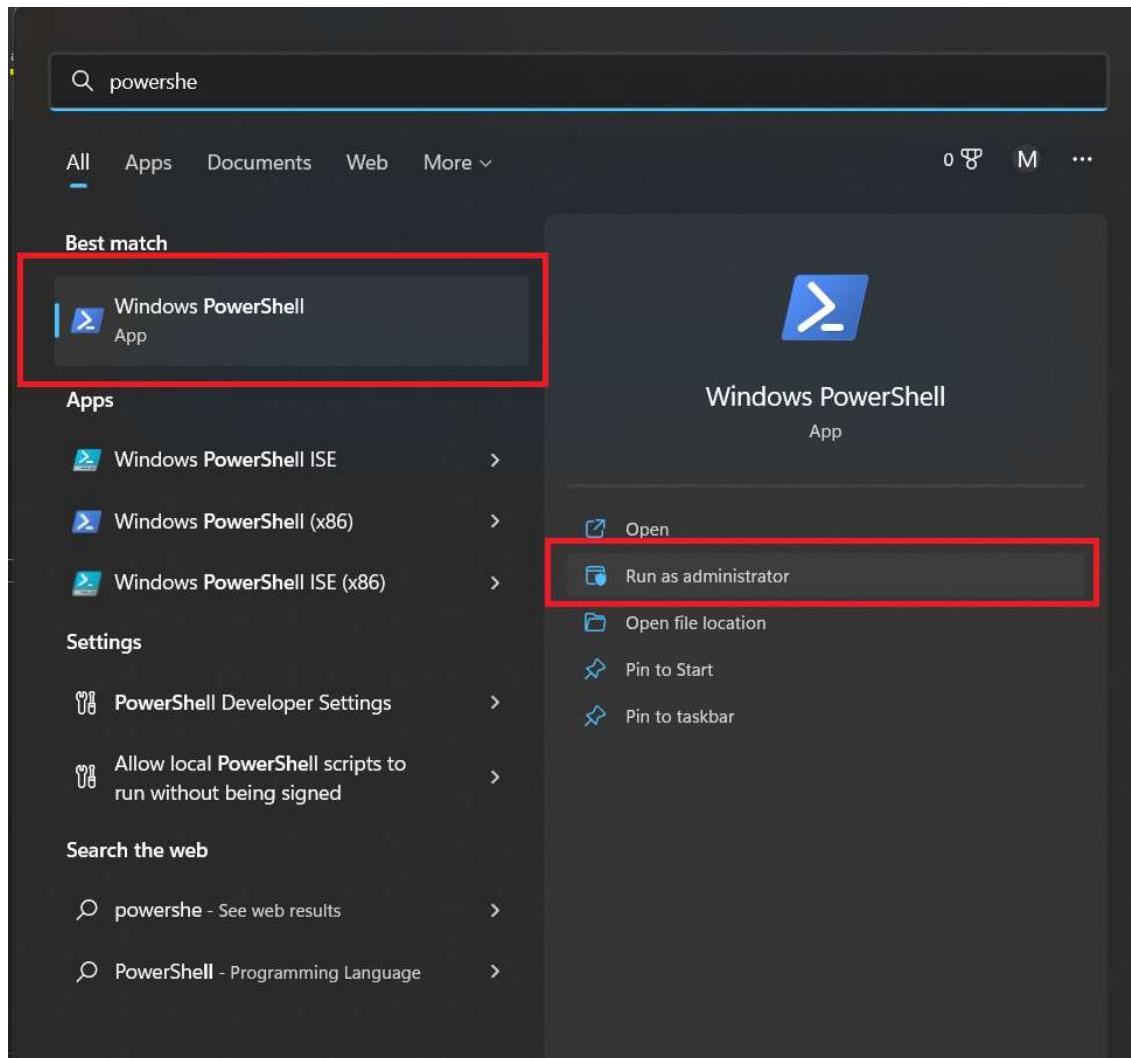
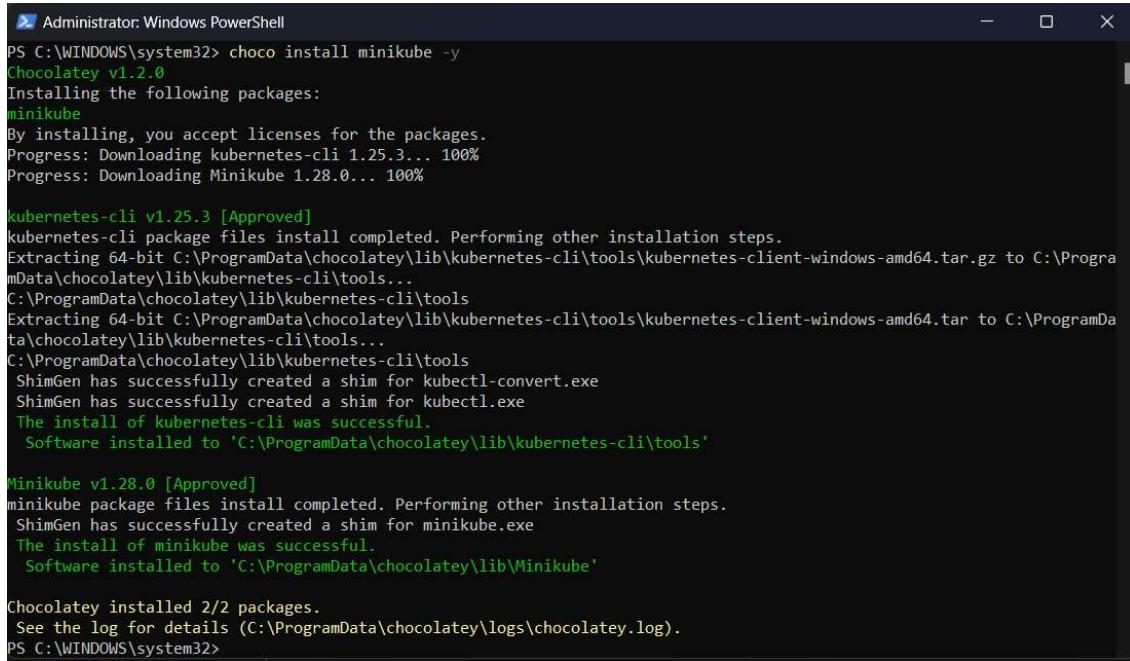


Figure 6.1.25 (Open PowerShell)

ii) Run command choco install minikube -y



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco install minikube -y
Chocolatey v1.2.0
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.25.3... 100%
Progress: Downloading Minikube 1.28.0... 100%

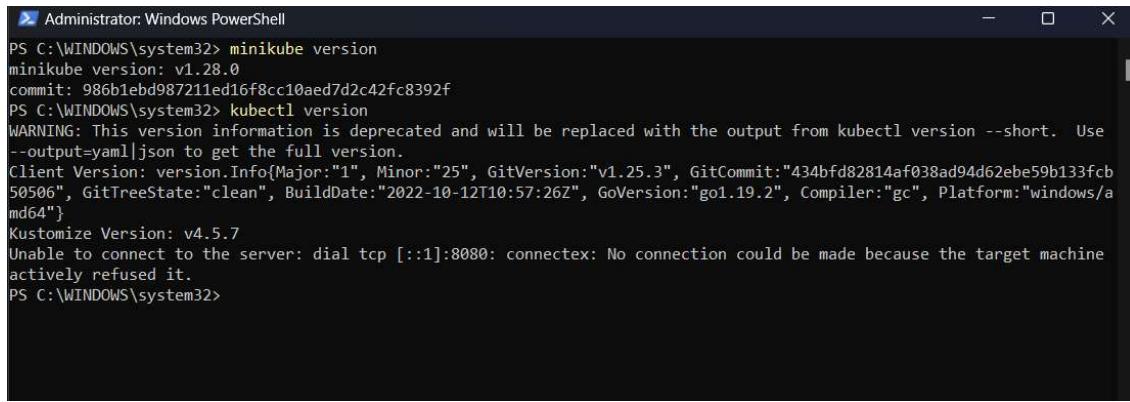
kubernetes-cli v1.25.3 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
ShimGen has successfully created a shim for kubectl-convert.exe
ShimGen has successfully created a shim for kubectl.exe
The install of kubernetes-cli was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'

Minikube v1.28.0 [Approved]
minikube package files install completed. Performing other installation steps.
ShimGen has successfully created a shim for minikube.exe
The install of minikube was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\Minikube'

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32>
```

Figure 6.1.26 (install minikube)

iii) Check the version of minikube and kubectl to ensure both packages are installed properly



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube version
minikube version: v1.28.0
commit: 986b1ebd987211ed16f8cc10aed7d2c42fc8392f
PS C:\WINDOWS\system32> kubectl version
WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.3", GitCommit:"434bfd82814af038ad94d62ebe59b133fcb50506", GitTreeState:"clean", BuildDate:"2022-10-12T10:57:26Z", GoVersion:"go1.19.2", Compiler:"gc", Platform:"windows/amd64"}
Kustomize Version: v4.5.7
Unable to connect to the server: dial tcp [::]:8080: connectex: No connection could be made because the target machine actively refused it.
PS C:\WINDOWS\system32>
```

Figure 6.1.27 (checking version of minikube and kubectl)

6.2 Setup AWS Credentials on environment variable

to work with terraform we need to setup AWS credential on environment variable

Steps:-

- i) Open AWS Console

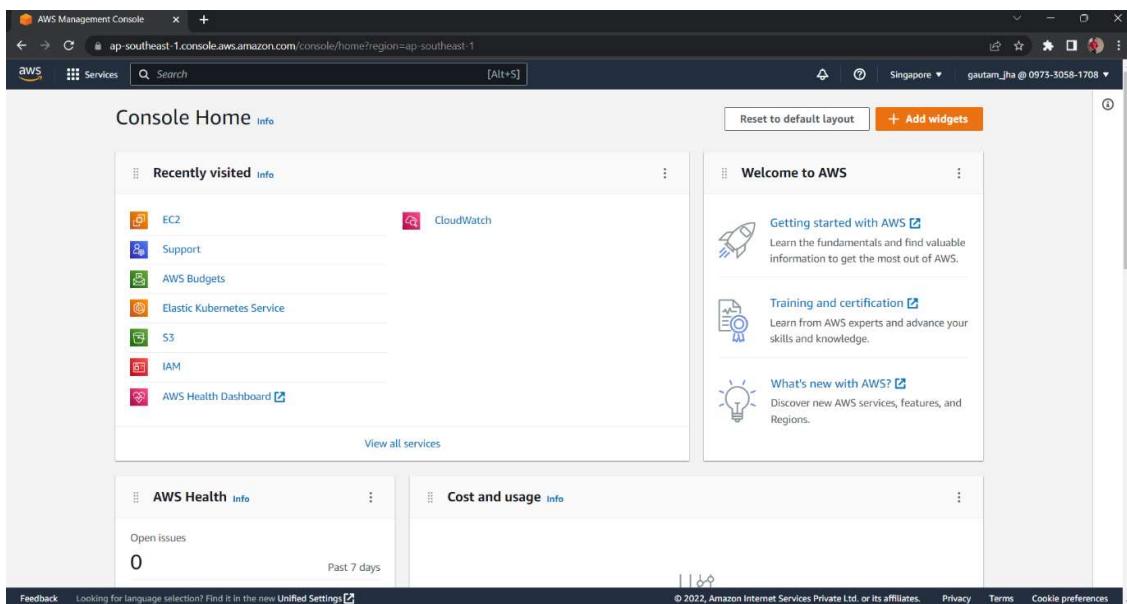


Figure 6.2.1 (AWS Web Console)

- ii) click on username dropdown menu from top right corner and select security credential option

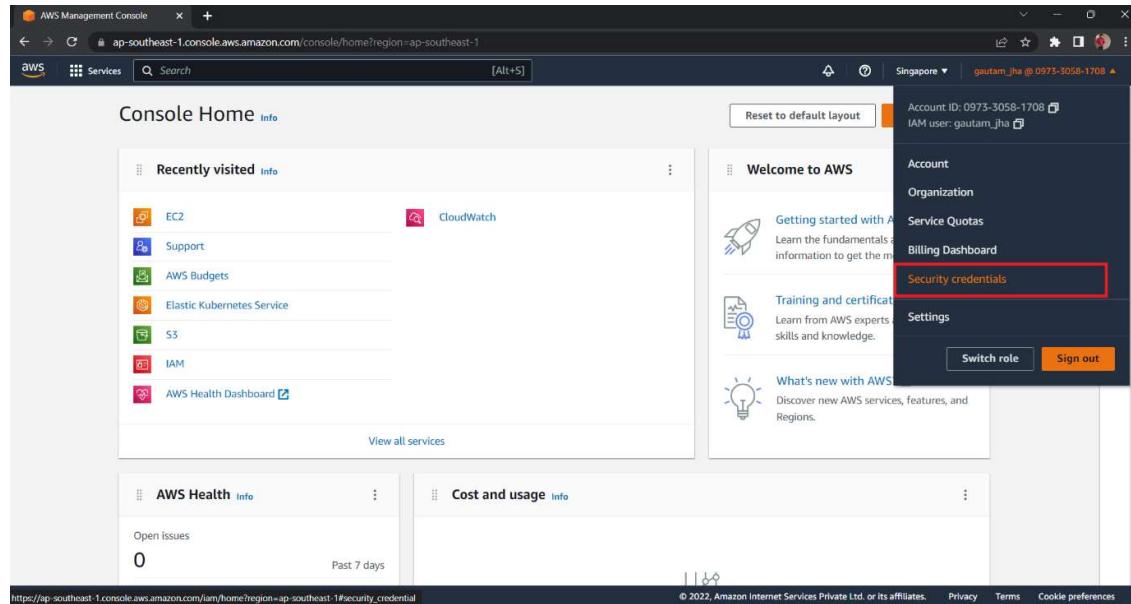


Figure 6.2.2 (Open username Dropdown)

iii) Click on create access keys

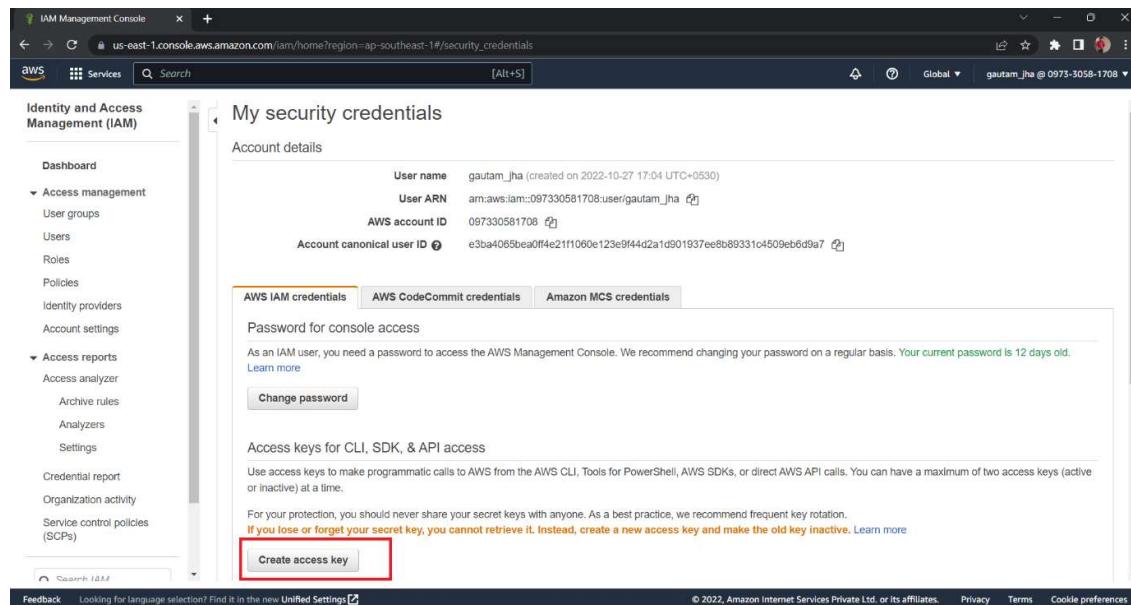


Figure 6.2.3 (Create access keys)

- iv) Copy your Access key ID to clipboard

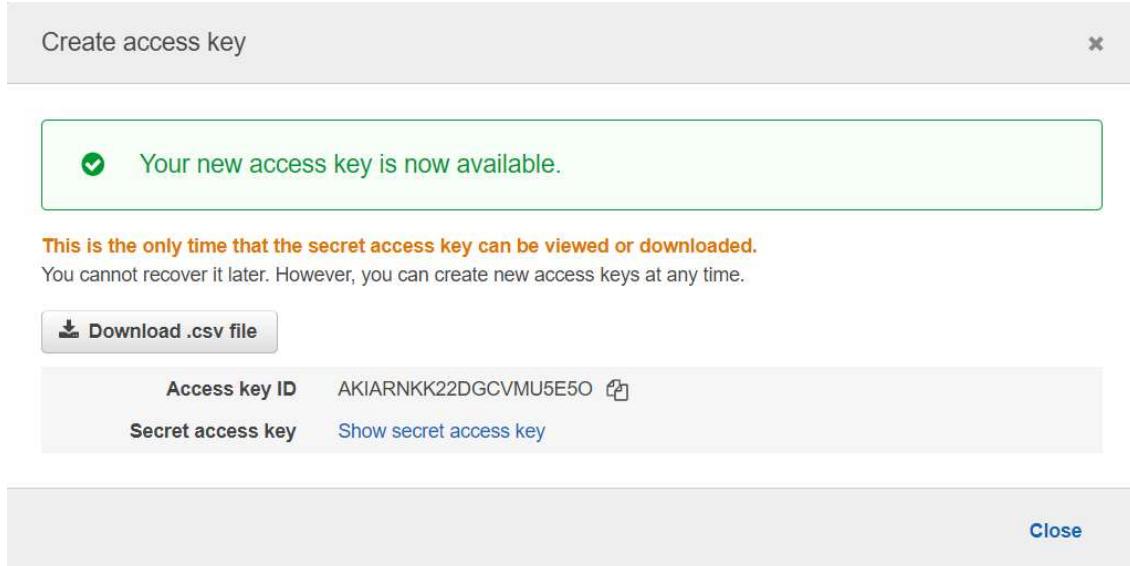


Figure 6.2.4 (Copy Access keys)

- v) Open environment variable and add a system variable called AWS_ACCESS_KEY_ID and set it's value as the access key id given by AWS

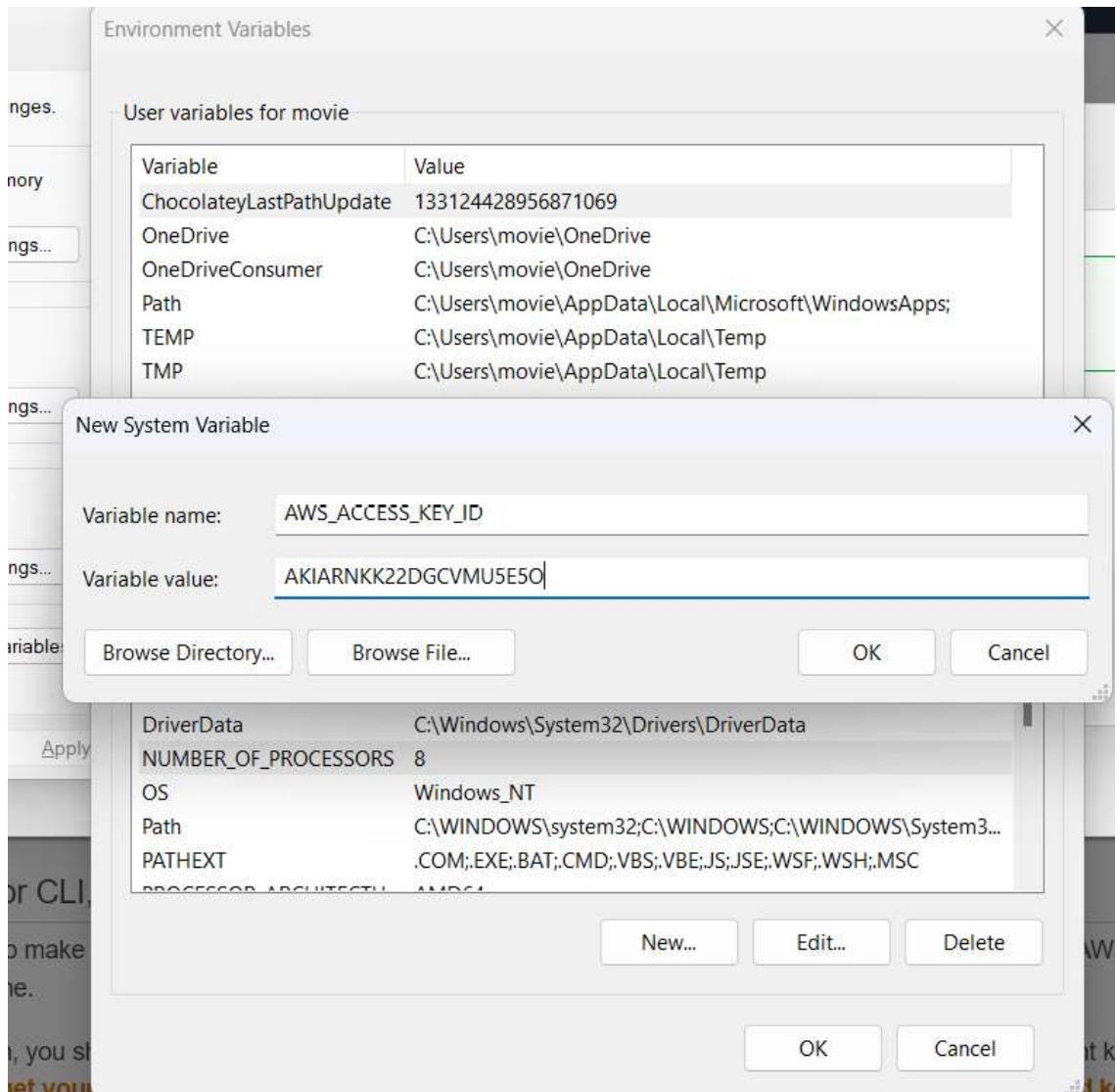


Figure 6.2.5 (Set AWS Access key id)

- vi) Now add another environment variable with name AWS_SECRET_ACCESS_KEY and set its value as the secret access key provided by AWS

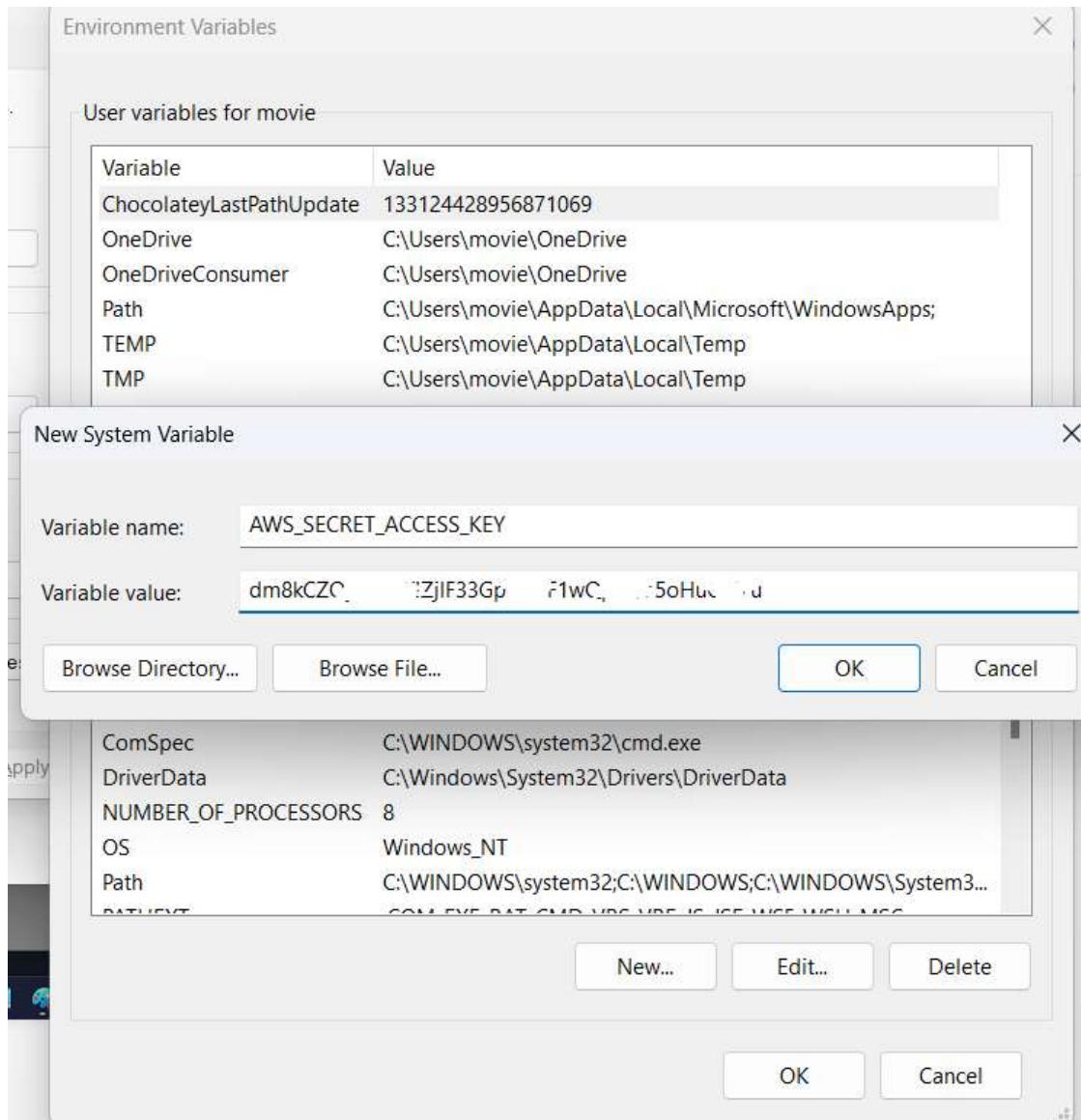


Figure 6.2.6 (Set AWS Secret Access key)

6.3 Get code on local system

Steps :-

1. Open any location in your file explorer where you want your code should be.

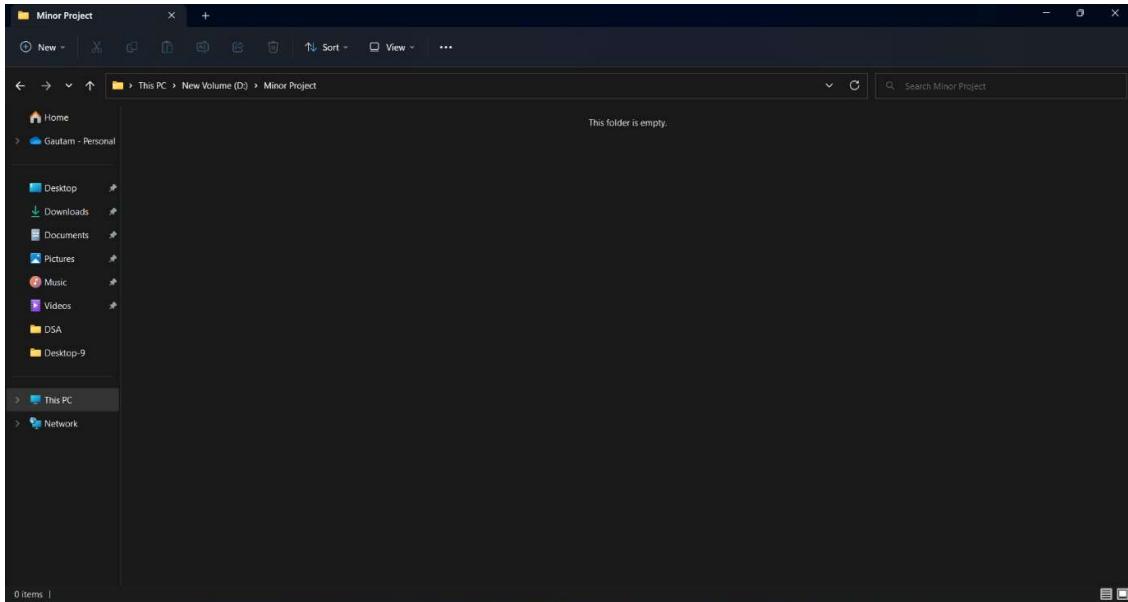


Figure 6.3.1 (Open Folder)

2. Open Same location on any terminal of your choice like git bash
PowerShell, windows terminal etc.

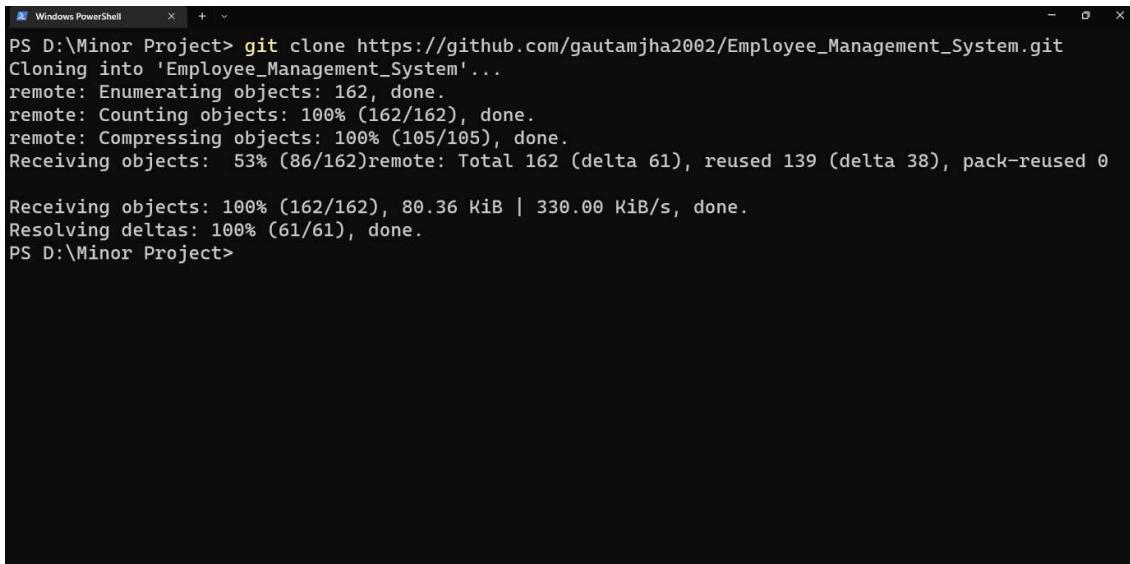


```
Windows PowerShell -> PS D:\Minor Project> pwd  
Path  
----  
D:\Minor Project  
PS D:\Minor Project>
```

Figure 6.3.2 (Open in terminal)

3. Clone the git repository using command

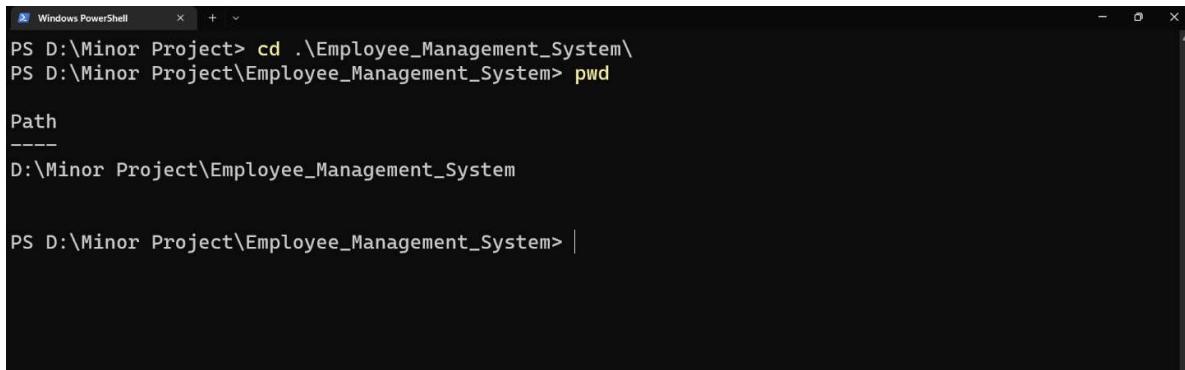
```
git clone https://github.com/gautamjha2002/Employee\_Management\_System.git
```



```
Windows PowerShell -> PS D:\Minor Project> git clone https://github.com/gautamjha2002/Employee_Management_System.git  
Cloning into 'Employee_Management_System'...  
remote: Enumerating objects: 162, done.  
remote: Counting objects: 100% (162/162), done.  
remote: Compressing objects: 100% (105/105), done.  
Receiving objects: 53% (86/162)remote: Total 162 (delta 61), reused 139 (delta 38), pack-reused 0  
Receiving objects: 100% (162/162), 80.36 KiB | 330.00 KiB/s, done.  
Resolving deltas: 100% (61/61), done.  
PS D:\Minor Project>
```

Figure 6.3.3 (Cloning the code)

4. Change directory to the folder where all the code for application is present, using cd command



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the following command and its output:

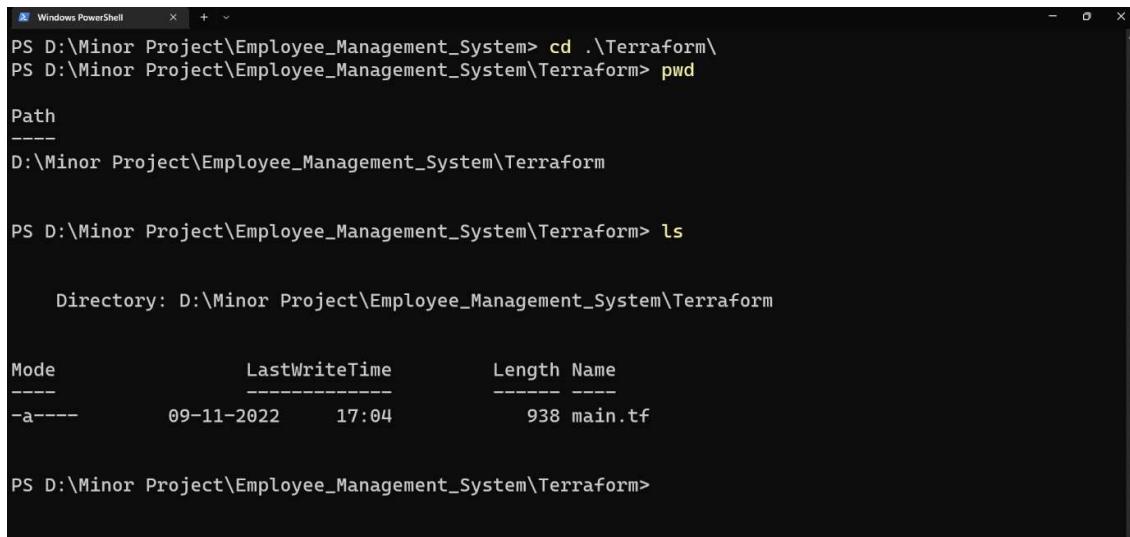
```
PS D:\Minor Project> cd .\Employee_Management_System\  
PS D:\Minor Project\Employee_Management_System> pwd  
Path  
----  
D:\Minor Project\Employee_Management_System  
  
PS D:\Minor Project\Employee_Management_System> |
```

Figure 6.3.4 (change directory)

6.4 Create the infrastructure using terraform and vagrant

Steps :-

1. Get inside Terraform folder using cd command



```
Windows PowerShell
PS D:\Minor Project\Employee_Management_System> cd .\Terraform\
PS D:\Minor Project\Employee_Management_System\Terraform> pwd

Path
-----
D:\Minor Project\Employee_Management_System\Terraform

PS D:\Minor Project\Employee_Management_System\Terraform> ls

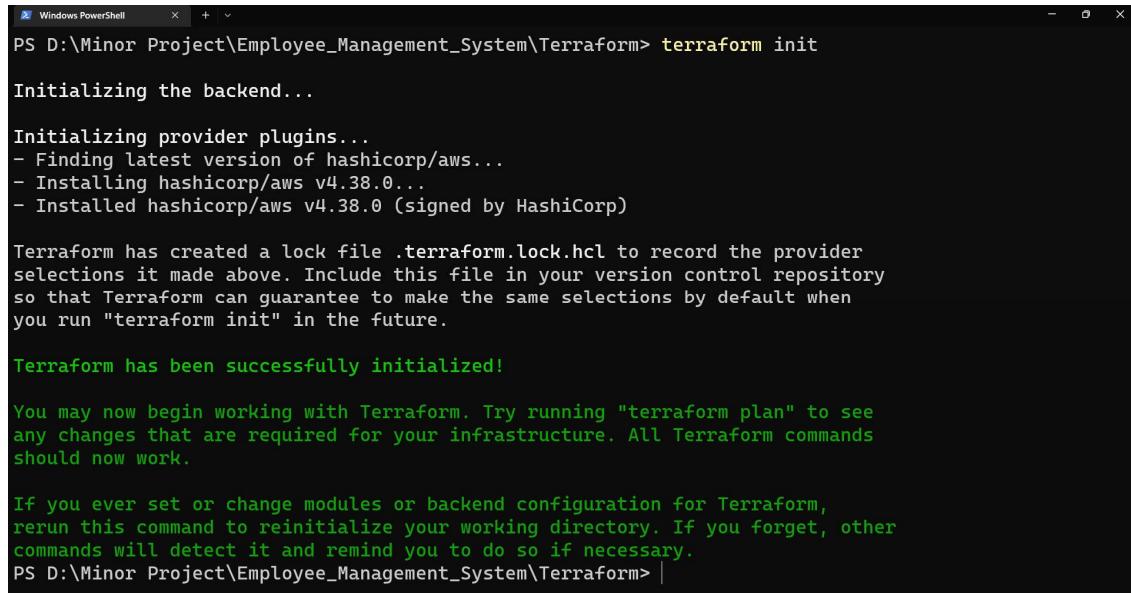
Directory: D:\Minor Project\Employee_Management_System\Terraform

Mode                LastWriteTime        Length Name
----                -----          ---- - 
-a---    09-11-2022     17:04         938 main.tf

PS D:\Minor Project\Employee_Management_System\Terraform>
```

Figure 6.4.1 (Change Directory to terraform)

2. Run command terraform init to initialize terraform to work with AWS



```
PS D:\Minor Project\Employee_Management_System\Terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.38.0...
- Installed hashicorp/aws v4.38.0 (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.

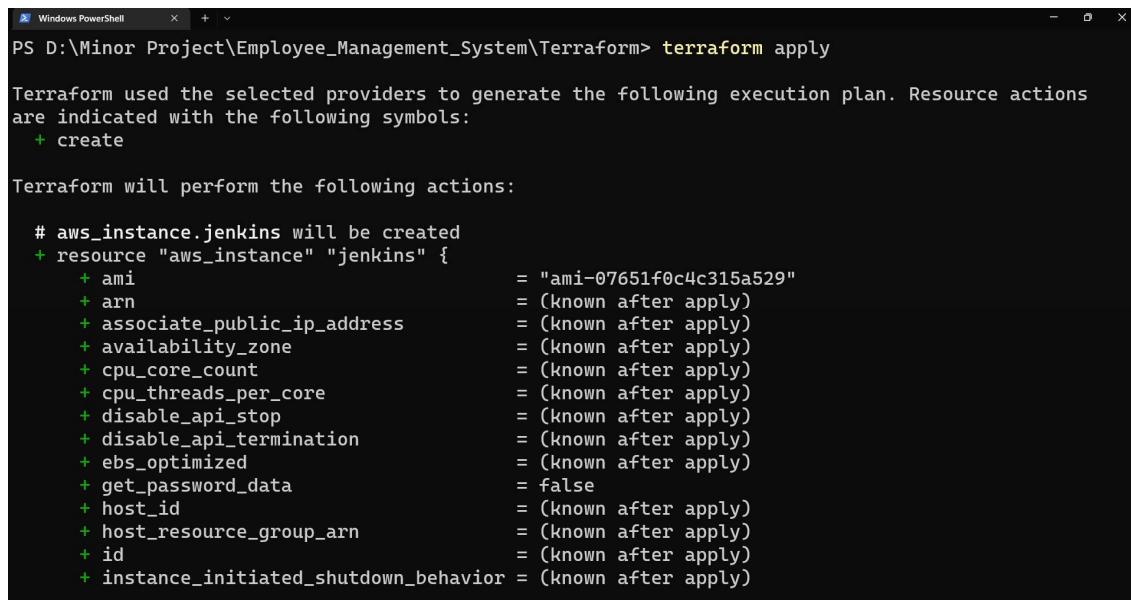
Terraform has been successfully initialized!

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.
PS D:\Minor Project\Employee_Management_System\Terraform> |
```

Figure 6.4.2 (Initialize Terraform)

3. Run terraform apply to apply the infrastructure setup



```
PS D:\Minor Project\Employee_Management_System\Terraform> 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.jenkins will be created
+ resource "aws_instance" "jenkins" {
    + ami = "ami-07651f0c4c315a529"
    + 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)
    + host_resource_group_arn = (known after apply)
    + id = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
```

Figure 6.4.3 (Apply Terraform code)

4. Check on AWS EC2 console if the instance is created or not

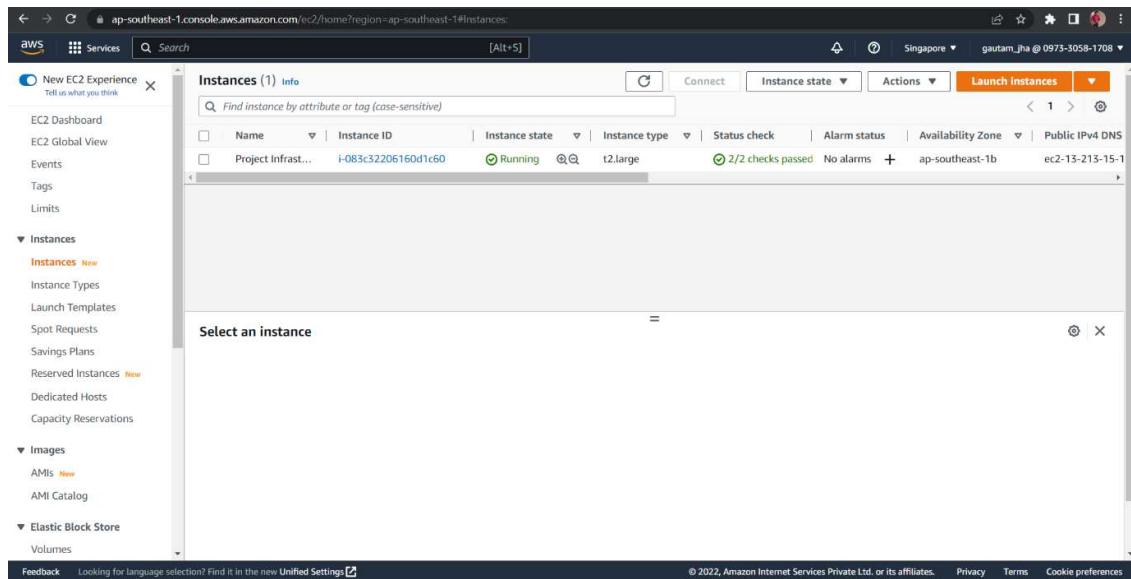


Figure 6.4.4 (Infrastructure on cloud)

5. Change directory to Vagrant

```
Windows PowerShell

PS D:\Minor Project\Employee_Management_System> cd VM
PS D:\Minor Project\Employee_Management_System\VM> pwd

Path
-----
D:\Minor Project\Employee_Management_System\VM

PS D:\Minor Project\Employee_Management_System\VM> ls

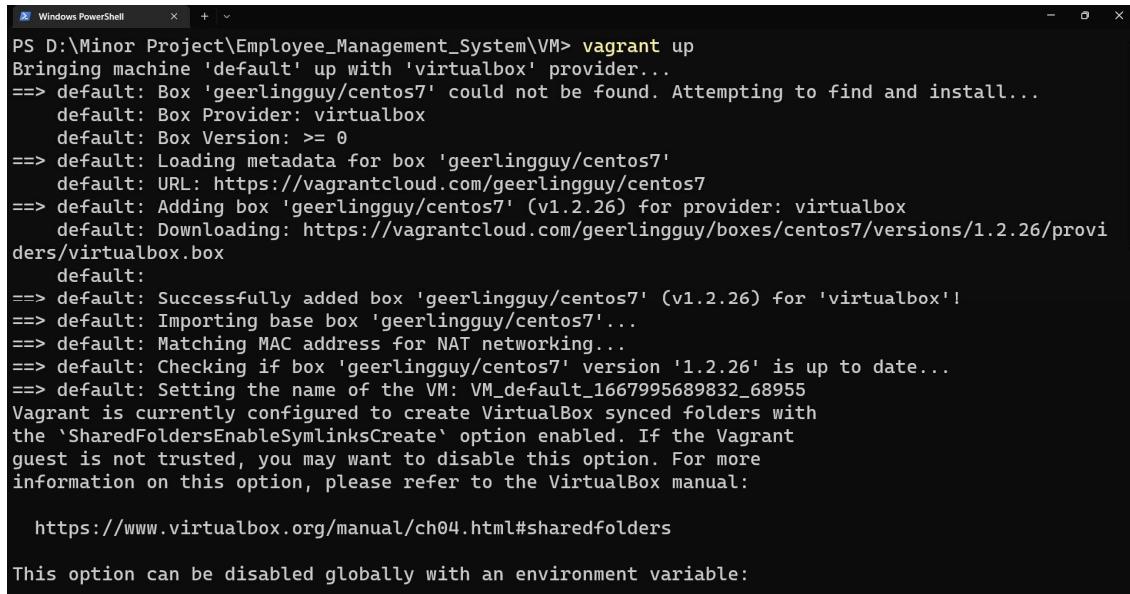
Directory: D:\Minor Project\Employee_Management_System\VM

Mode                LastWriteTime          Length Name
----                -----          ----
-a---- 09-11-2022     17:04            3090 Vagrantfile

PS D:\Minor Project\Employee_Management_System\VM> |
```

Figure 6.4.5 (Change directory to VM)

6. Create a Centos machine using vagrant use command vagrant up



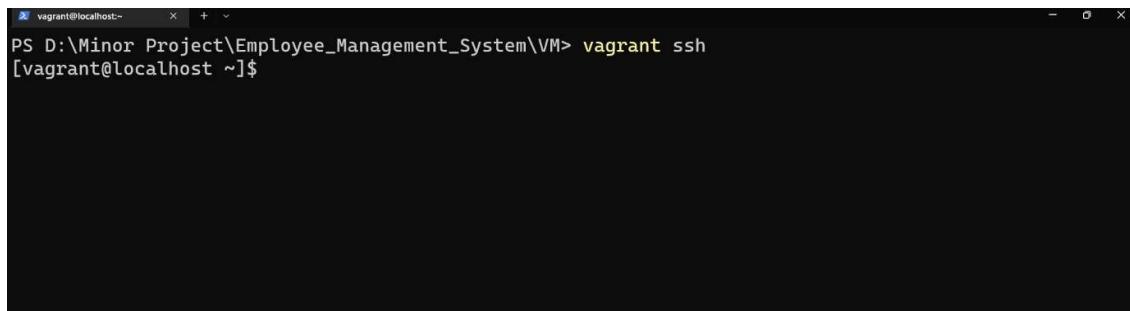
```
PS D:\Minor Project\Employee_Management_System\VM> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'geerlingguy/centos7' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'geerlingguy/centos7'
    default: URL: https://vagrantcloud.com/geerlingguy/centos7
==> default: Adding box 'geerlingguy/centos7' (v1.2.26) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/geerlingguy/boxes/centos7/versions/1.2.26/providers/virtualbox.box
    default:
    default: Successfully added box 'geerlingguy/centos7' (v1.2.26) for 'virtualbox'!
==> default: Importing base box 'geerlingguy/centos7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'geerlingguy/centos7' version '1.2.26' is up to date...
==> default: Setting the name of the VM: VM_default_1667995689832_68955
Vagrant is currently configured to create VirtualBox synced folders with
the 'SharedFoldersEnableSymlinksCreate' option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:
```

Figure 6.4.6 (installing CentOS)

7. SSH in the machine using command vagrant ssh



```
vagrant@localhost:~ % PS D:\Minor Project\Employee_Management_System\VM> vagrant ssh
[vagrant@localhost ~]$
```

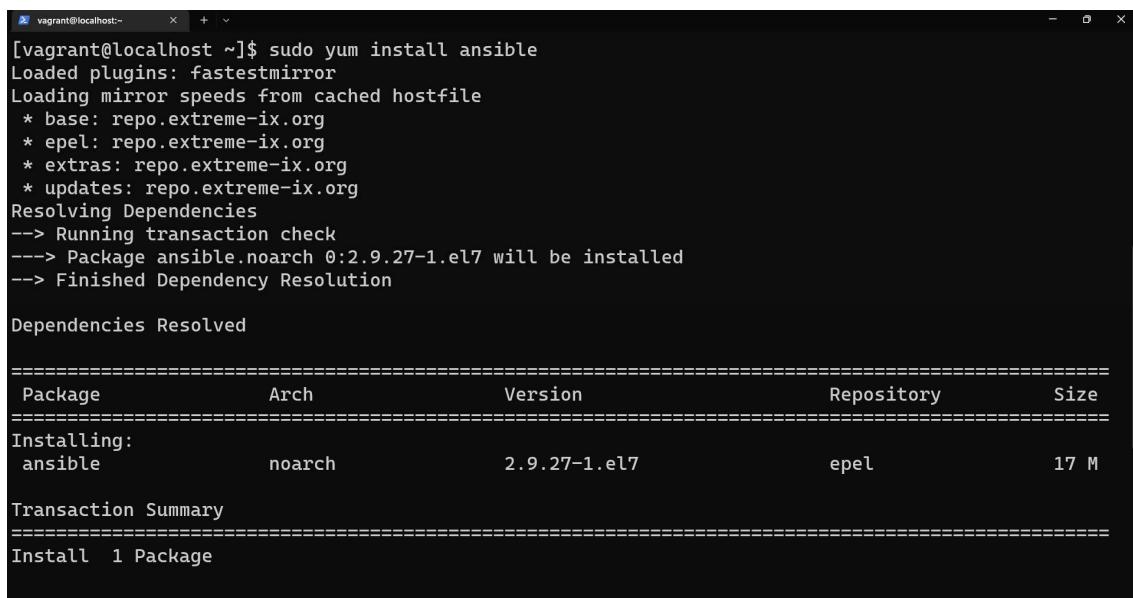
Figure 6.4.7 (SSH into the machine)

6.5 Install ansible and setup user and password

Install ansible on VM machine and create user ansible on both the machine
(VM machine and cloud machine)

Steps :-

1. Install ansible on VM machine using command sudo yum install ansible



```
vagrant@localhost ~]$ sudo yum install ansible
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: repo.extreme-ix.org
 * epel: repo.extreme-ix.org
 * extras: repo.extreme-ix.org
 * updates: repo.extreme-ix.org
Resolving Dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.27-1.el7 will be installed
--> Finished Dependency Resolution

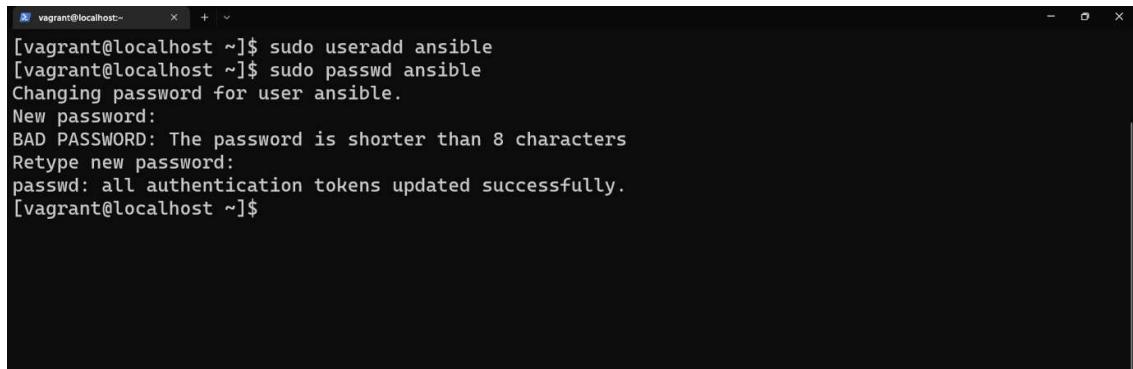
Dependencies Resolved

=====
Package           Arch      Version       Repository      Size
=====
Installing:
ansible          noarch   2.9.27-1.el7    epel        17 M

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

Figure 6.5.1 (Installing Ansible)

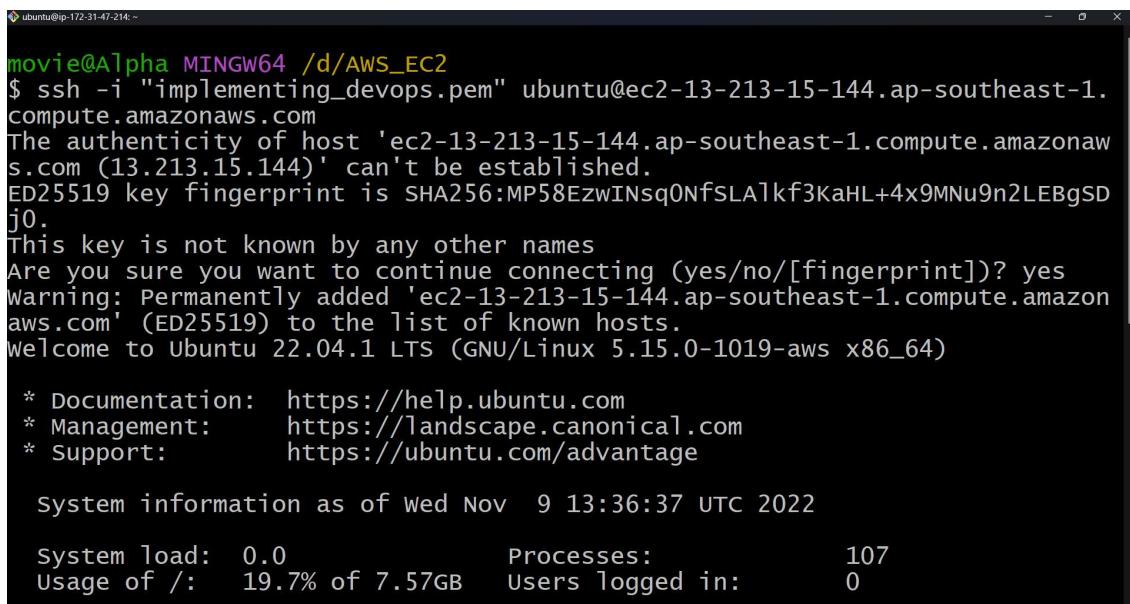
2. Create user ansible on VM machine and give some password to it



```
vagrant@localhost ~]$ sudo useradd ansible
[vagrant@localhost ~]$ sudo passwd ansible
Changing password for user ansible.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[vagrant@localhost ~]$
```

Figure 6.5.1 (Add user ansible to the machine)

3. SSH into the cloud machine using command ssh -i "implementing_devops.pem" ubuntu@ec2-13-213-15-144.ap-southeast-1.compute.amazonaws.com (Note that the pem file should be with you on the same folder from where you are trying to ssh into the cloud machine)



```
ubuntu@ip-172-31-47-214:~$ ssh -i "implementing_devops.pem" ubuntu@ec2-13-213-15-144.ap-southeast-1.compute.amazonaws.com
The authenticity of host 'ec2-13-213-15-144.ap-southeast-1.compute.amazonaws.com (13.213.15.144)' can't be established.
ED25519 key fingerprint is SHA256:MP58EzwINsq0NfSLA1kf3KaHL+4x9MNu9n2LEBgSDj0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-213-15-144.ap-southeast-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

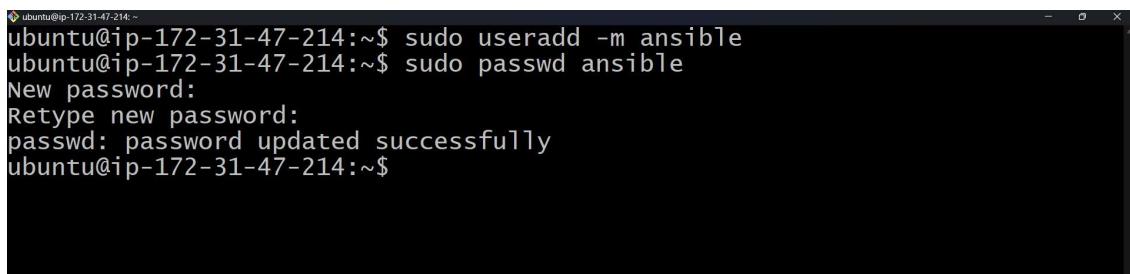
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Wed Nov  9 13:36:37 UTC 2022

 System load:  0.0          Processes:           107
 Usage of /:   19.7% of 7.57GB    Users logged in:      0
```

Figure 6.5.2 (ssh into the cloud machine)

4. Add user ansible and password for the same user



```
ubuntu@ip-172-31-47-214:~$ sudo useradd -m ansible
ubuntu@ip-172-31-47-214:~$ sudo passwd ansible
New password:
Retype new password:
passwd: password updated successfully
ubuntu@ip-172-31-47-214:~$
```

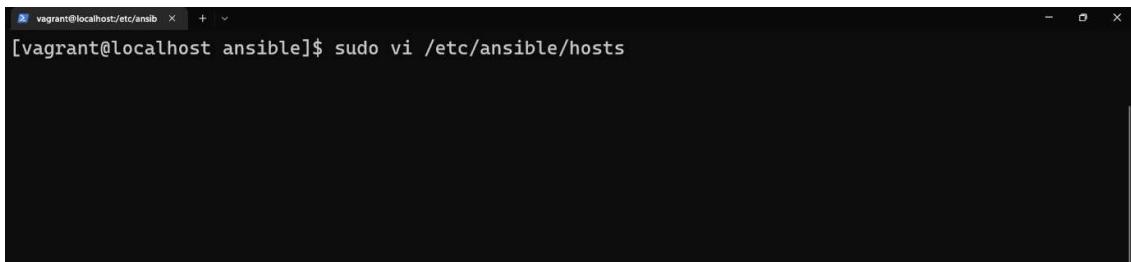
Figure 6.5.3 (adding user and password on cloud machine)

6.6 Add IP to ansible host file

Add cloud machine IP address to the ansible hosts file of VM machine host

Steps :-

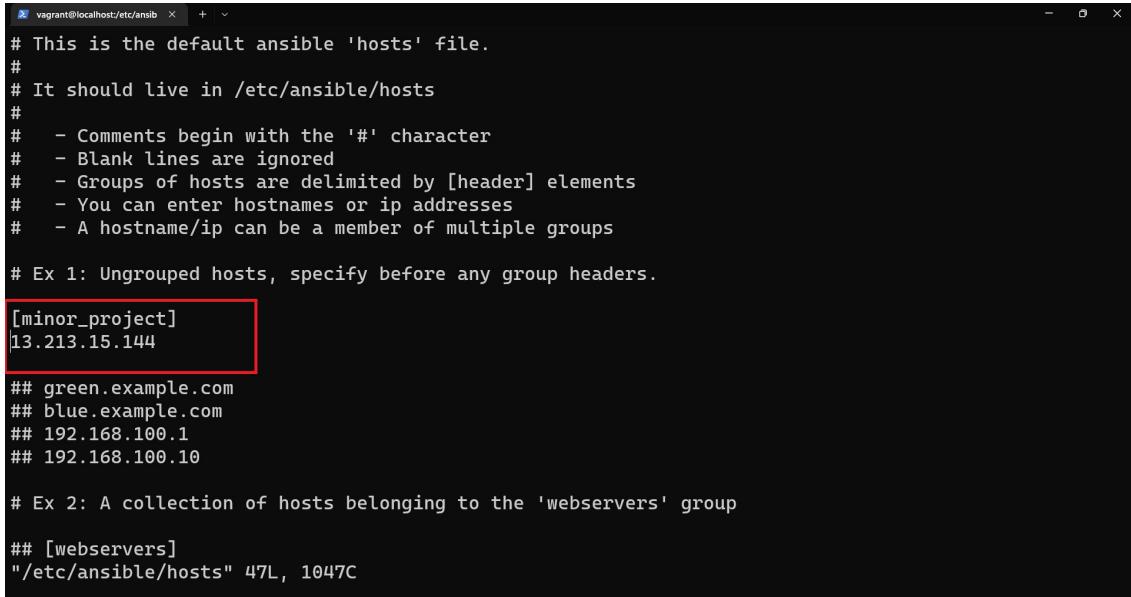
1. Open file using command sudo /etc/ansible/hosts



```
vagrant@localhost:~$ sudo vi /etc/ansible/hosts
```

Figure 6.6.1 (open ansible hosts file)

2. Add IP address of cloud machine with name minor_project



```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

[minor_project]
13.213.15.144

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
"/etc/ansible/hosts" 47L, 1047C
```

Figure 6.6.2 (adding ip address of cloud machine)

3. Try to ping the cloud machine using ansible to check if ansible is able to reach the cloud machine using command ansible all -m ping



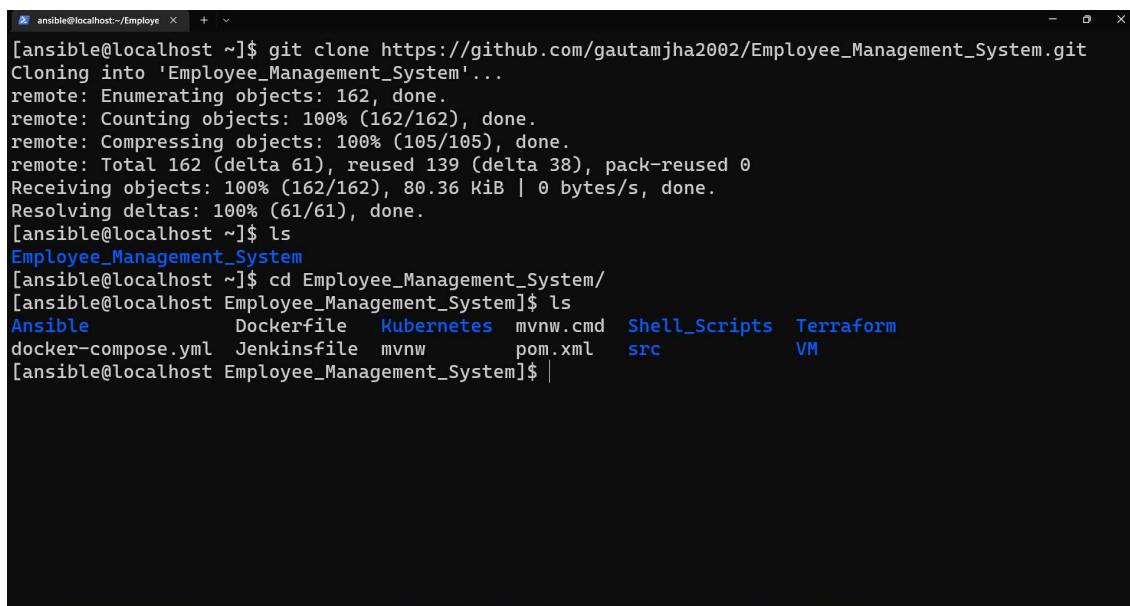
```
[ansible@localhost ~]$ ansible all -m ping
13.213.15.144 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[ansible@localhost ~]$ |
```

Figure 6.6.3 (pinging the cloud machine)

6.7 Configure the cloud machine using ansible

Steps :-

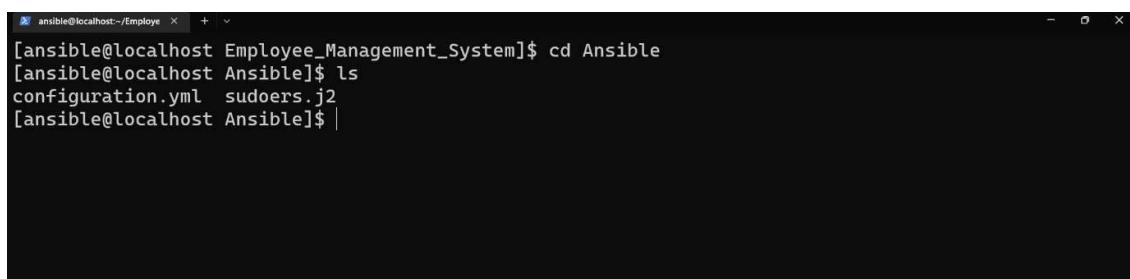
1. Clone the git repository again here and change the directory to it for the code of ansible



```
[ansible@localhost ~]$ git clone https://github.com/gautamjha2002/Employee_Management_System.git
Cloning into 'Employee_Management_System'...
remote: Enumerating objects: 162, done.
remote: Counting objects: 100% (162/162), done.
remote: Compressing objects: 100% (105/105), done.
remote: Total 162 (delta 61), reused 139 (delta 38), pack-reused 0
Receiving objects: 100% (162/162), 80.36 KiB | 0 bytes/s, done.
Resolving deltas: 100% (61/61), done.
[ansible@localhost ~]$ ls
Employee_Management_System
[ansible@localhost ~]$ cd Employee_Management_System/
[ansible@localhost Employee_Management_System]$ ls
Ansible Dockerfile Kubernetes mvnw.cmd Shell_Scripts Terraform
docker-compose.yml Jenkinsfile mvnw pom.xml src VM
[ansible@localhost Employee_Management_System]$ |
```

Figure 6.7.1 (Clone git repo and change directory)

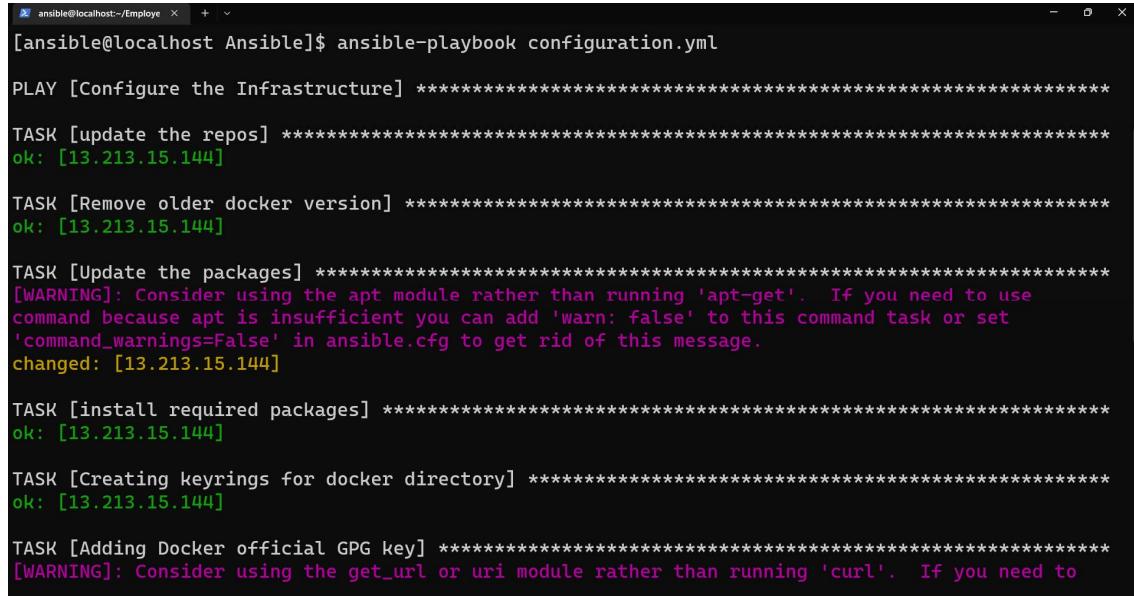
2. Again change directory to Ansible here the yaml file for ansible is located



```
[ansible@localhost ~]$ cd Employee_Management_System
[ansible@localhost Employee_Management_System]$ cd Ansible
[ansible@localhost Ansible]$ ls
configuration.yml sudoers.j2
[ansible@localhost Ansible]$ |
```

Figure 6.7.2 (Change directory to Ansible)

3. Configure the cloud machine using command ansible-playbook configuration.yml



```
[ansible@localhost Ansible]$ ansible-playbook configuration.yml

PLAY [Configure the Infrastructure] ****
TASK [update the repos] ****
ok: [13.213.15.144]

TASK [Remove older docker version] ****
ok: [13.213.15.144]

TASK [Update the packages] ****
[WARNING]: Consider using the apt module rather than running 'apt-get'. If you need to use
command because apt is insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [13.213.15.144]

TASK [install required packages] ****
ok: [13.213.15.144]

TASK [Creating keyrings for docker directory] ****
ok: [13.213.15.144]

TASK [Adding Docker official GPG key] ****
[WARNING]: Consider using the get_url or uri module rather than running 'curl'. If you need to
```

Figure 6.7.3 (configure infrastructure using ansible)

6.8 Setup Jenkins Web

Steps :-

1. Open Jenkins web application with Instance IP address and port 8080

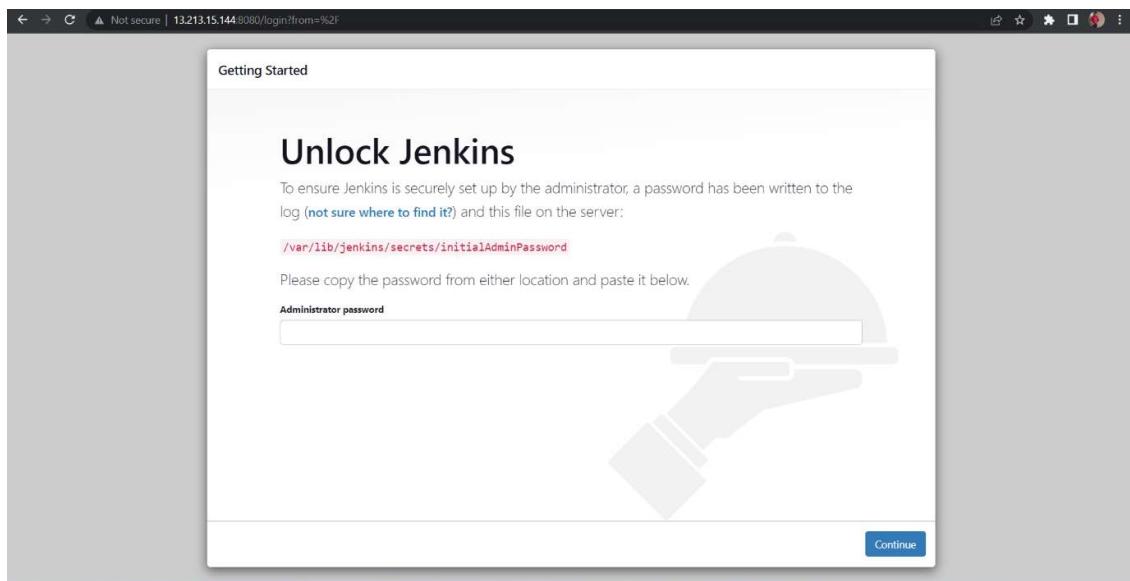


Figure 6.8.1 (Jenkins web application)

2. Get Password to unlock Jenkins using command

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```

A screenshot of a terminal window on a Linux system. The command 'cat /var/lib/jenkins/secrets/initialAdminPassword' is run, and the output is a long, randomly generated string of characters: '305c9f73ed694242969347121b81308b'. The terminal prompt is 'root@ip-172-31-47-214:~#'. The background of the terminal window is black.

Figure 6.8.2 (Password for jenkins)

3. Copy the password and paste it into jenkins web application and click on continue

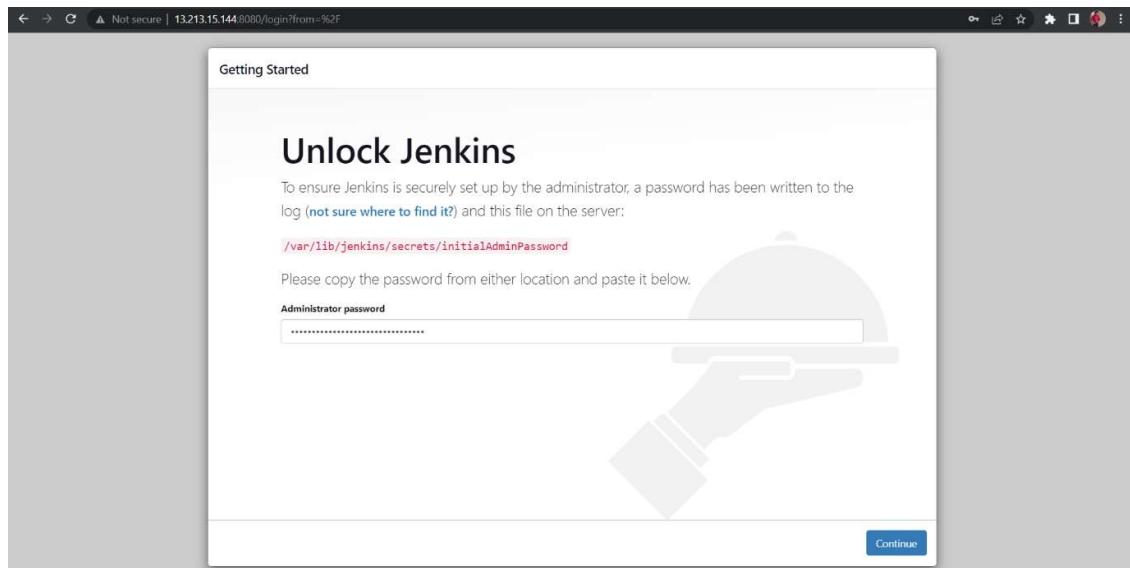


Figure 6.8.3 (paste Jenkins password)

4. Click on install suggested plugins

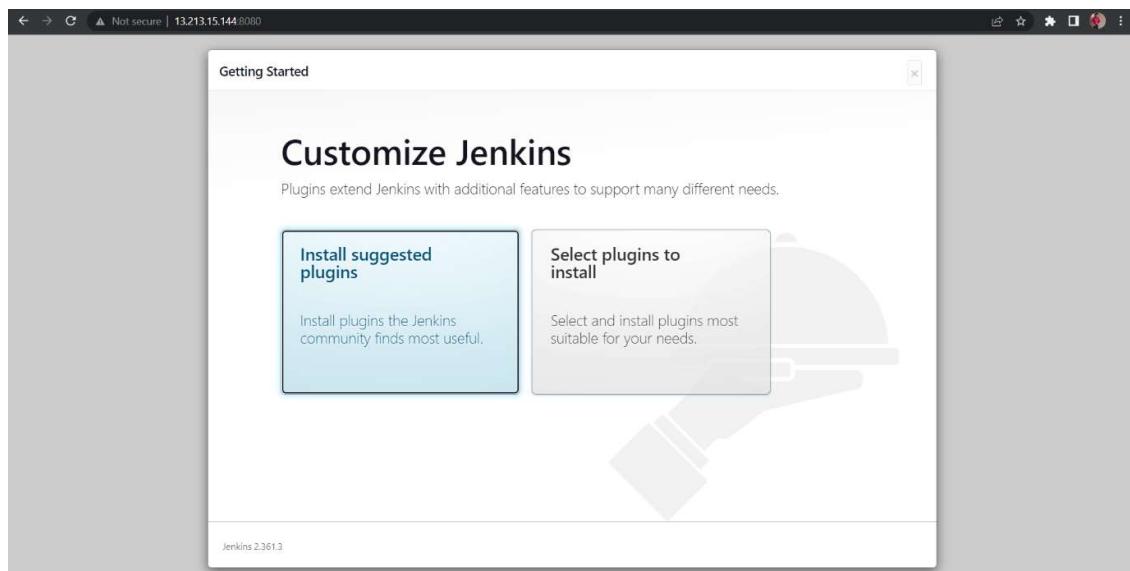


Figure 6.8.4 (Install Suggested Plugin)

5. Wait until all plugin is installed, and setup Admin username then click on save and continue.

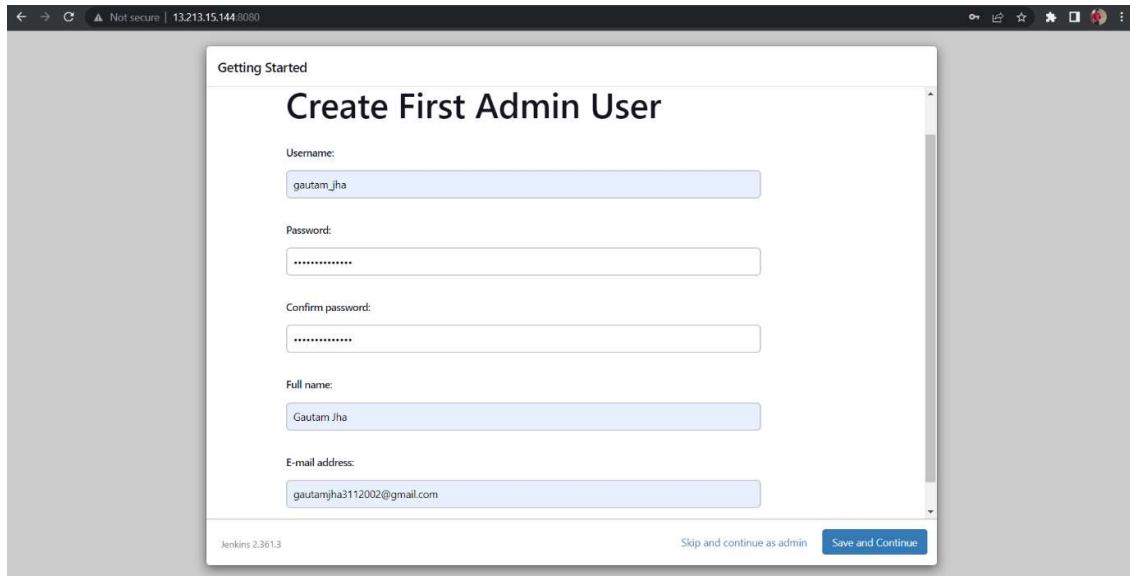


Figure 6.8.5 (Create Admin User)

6. You will be redirected to Jenkins Home page.

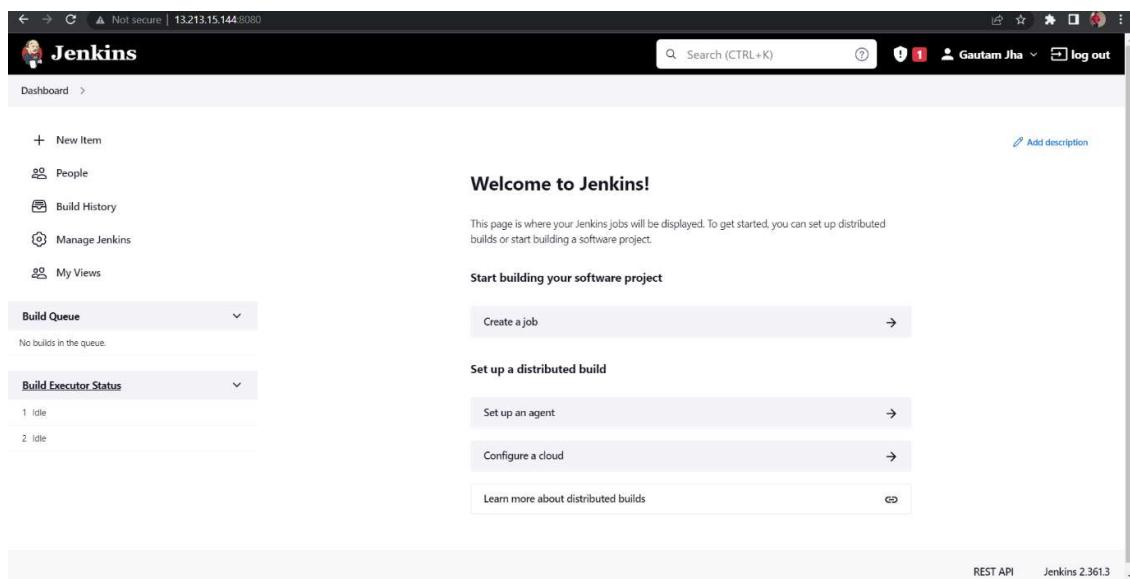


Figure 6.8.6 (Jenkins Home page)

6.9 Configure MAVEN tool Slack plugin

Steps :-

1. From Home page in left side click on manage Jenkins

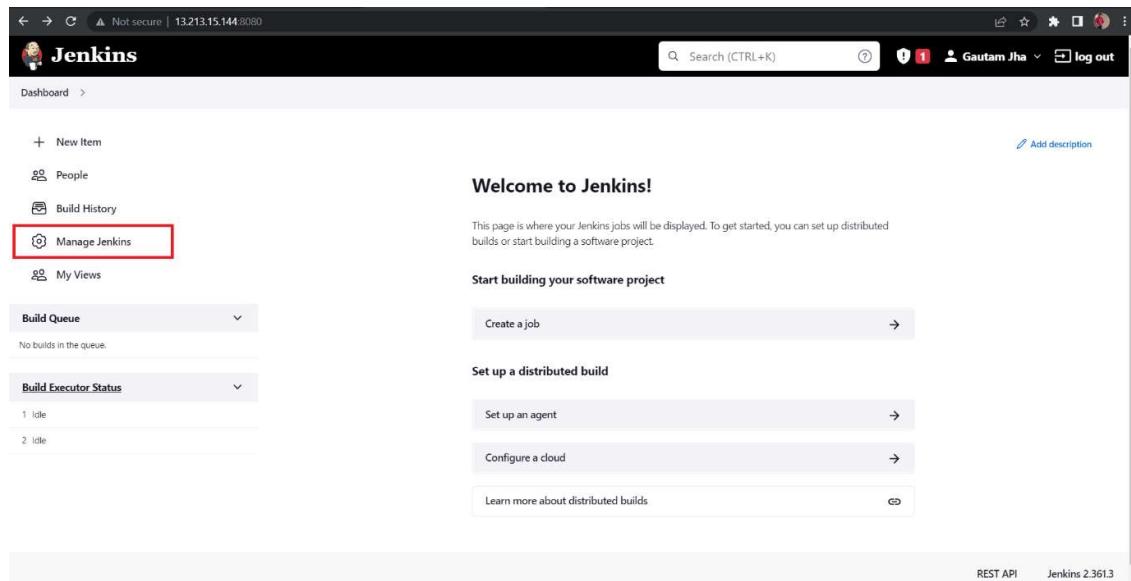


Figure 6.9.1 (Manage Jenkins)

2. Click on global tool configuration

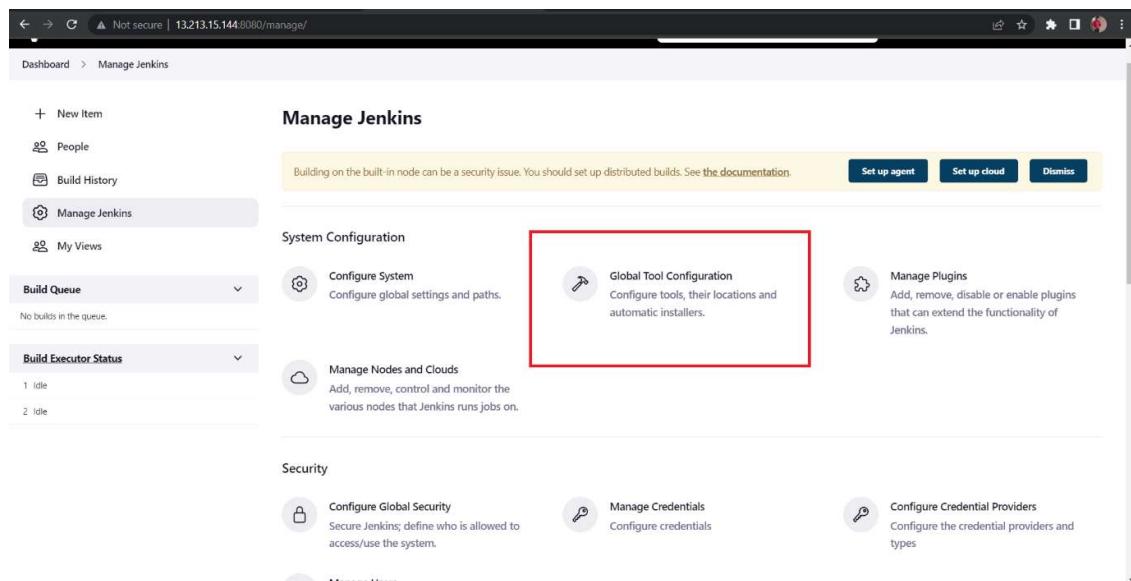


Figure 6.9.2 (Global Configuration)

3. Scroll down and click on Add Maven

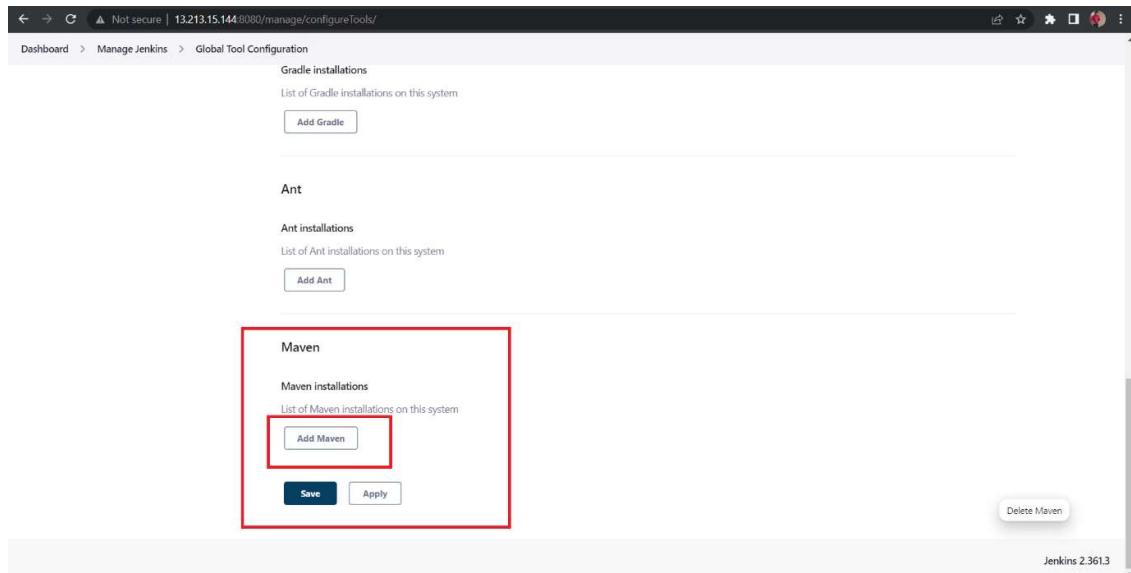


Figure 6.9.3 (Add Maven)

4. Select Maven version and give name MAVEN and click on save

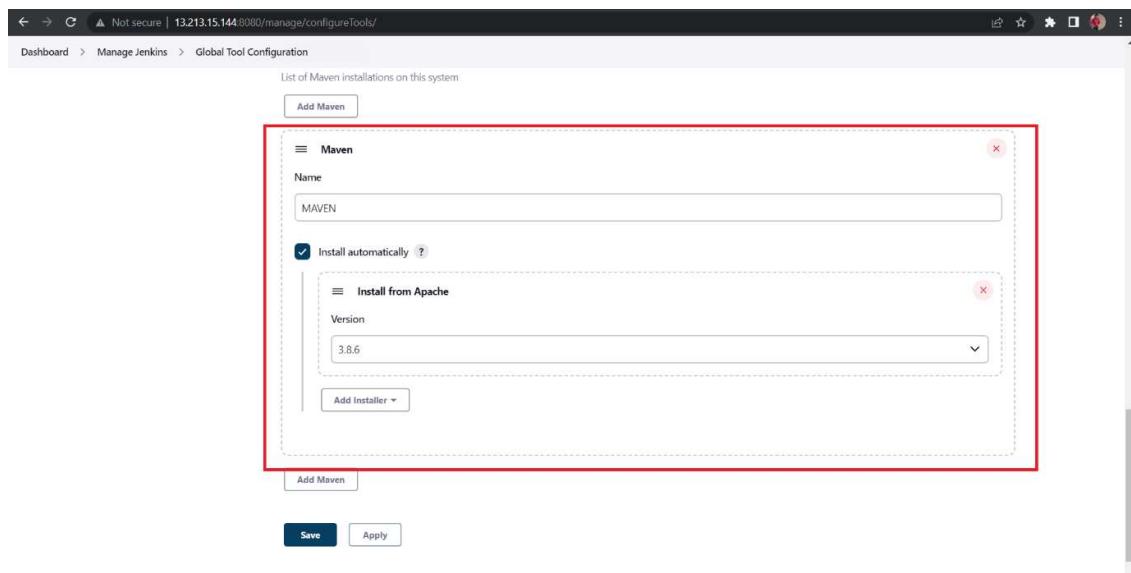


Figure 6.9.4 (Select version and give name to maven)

5. Now Click on manage Plugins

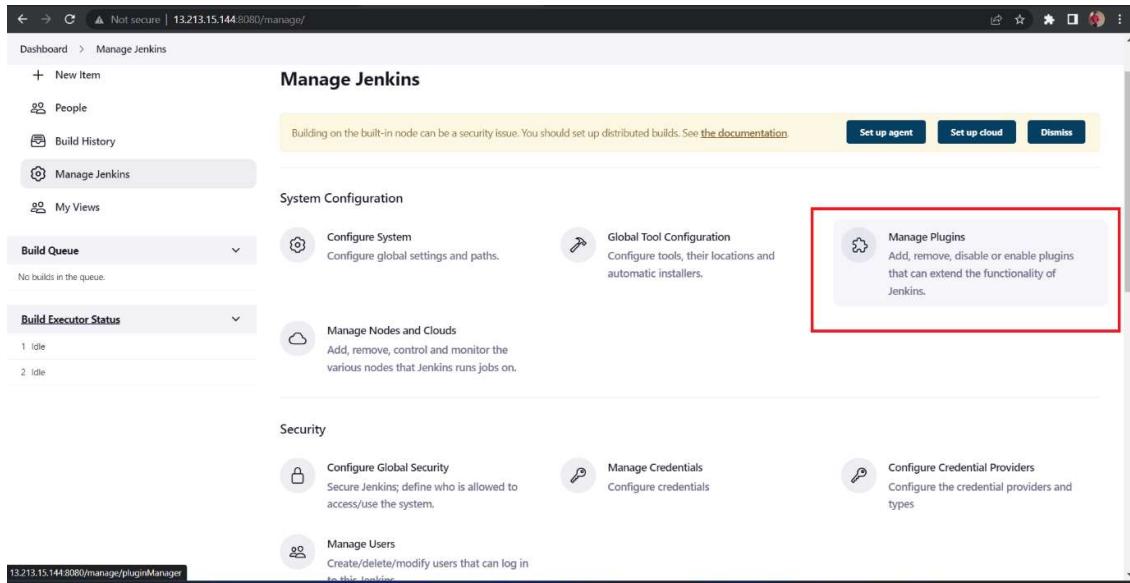


Figure 6.9.5 (Manage plugin)

6. In available section search for slack Notification plugin and then install it without restart

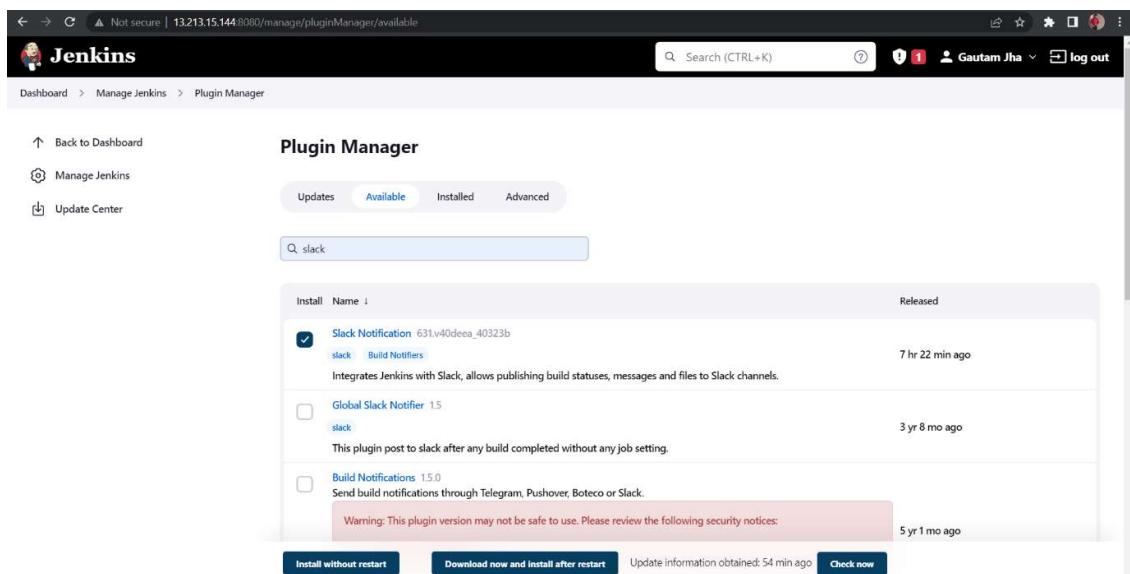


Figure 6.9.6 (Installing Slack Notification Plugin)

7. Wait until plugin is installed

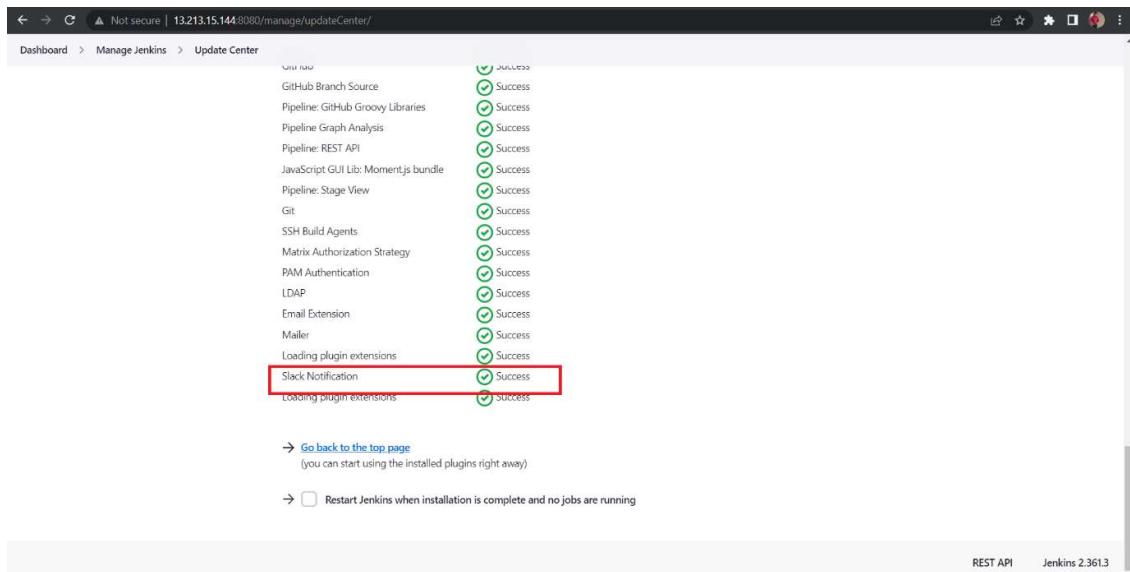


Figure 6.9.7 (Installed Slack plugin)

8. Again Go back to Manage Jenkins section and click on Manage credential

Figure 6.9.8 (Manage credential)

6.10 Configure Slack to work with Jenkins

Steps :-

1. Login to slack and either create workspace or choose from existing one

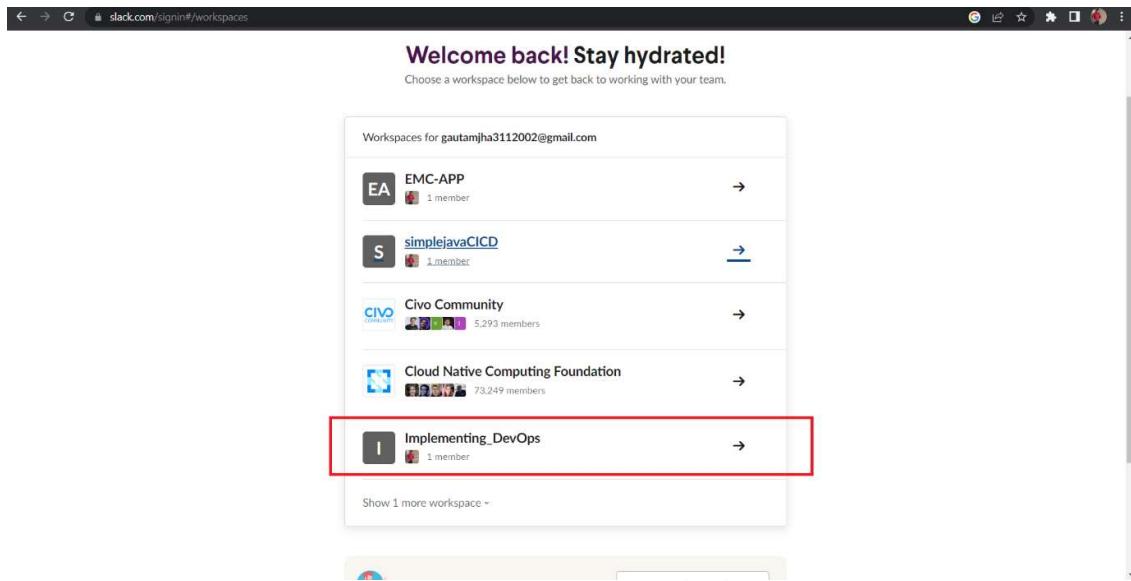


Figure 6.10.1 (Login & creating Workspace)

2. Create channel minor project

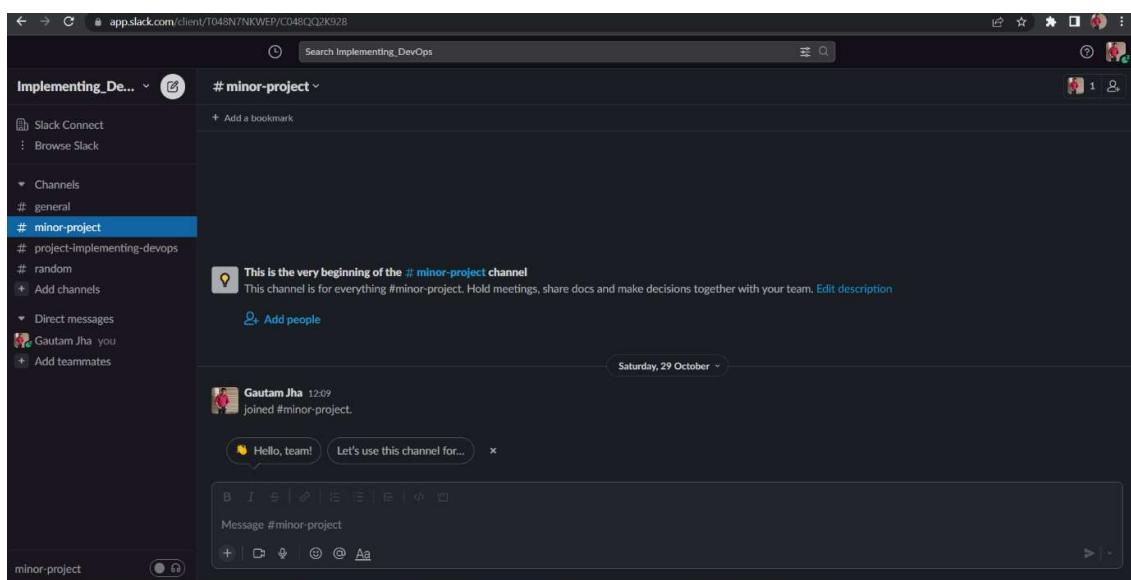


Figure 6.10.2 (channel minor-project)

3. Add Jenkins CI app to slack by clicking on Add to slack

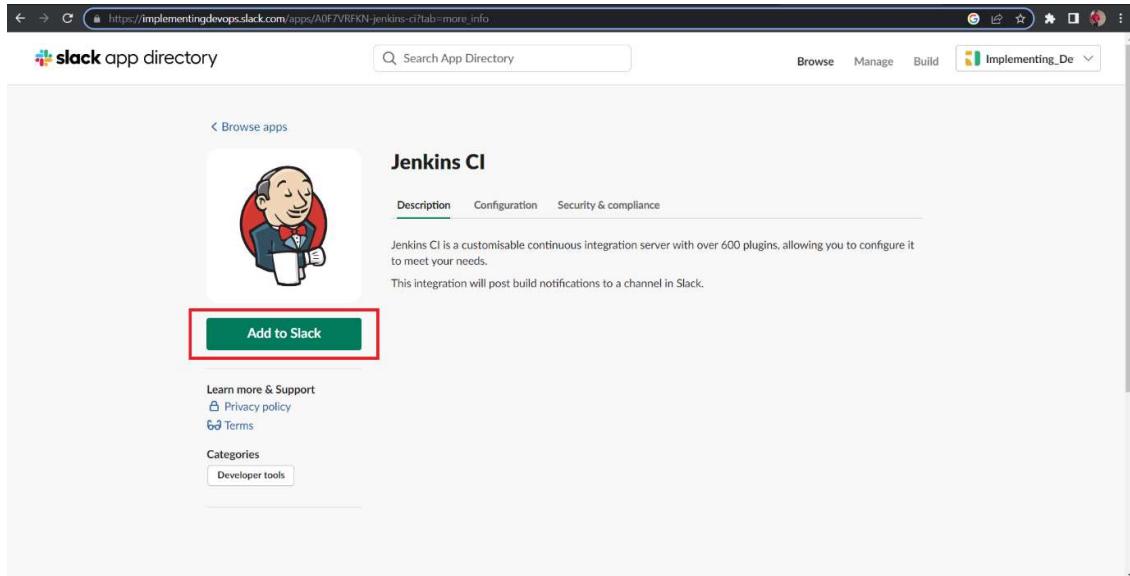


Figure 6.10.3 (Add Jenkins CI to Slack)

4. Choose channel and click on Add Jenkins CI Integration

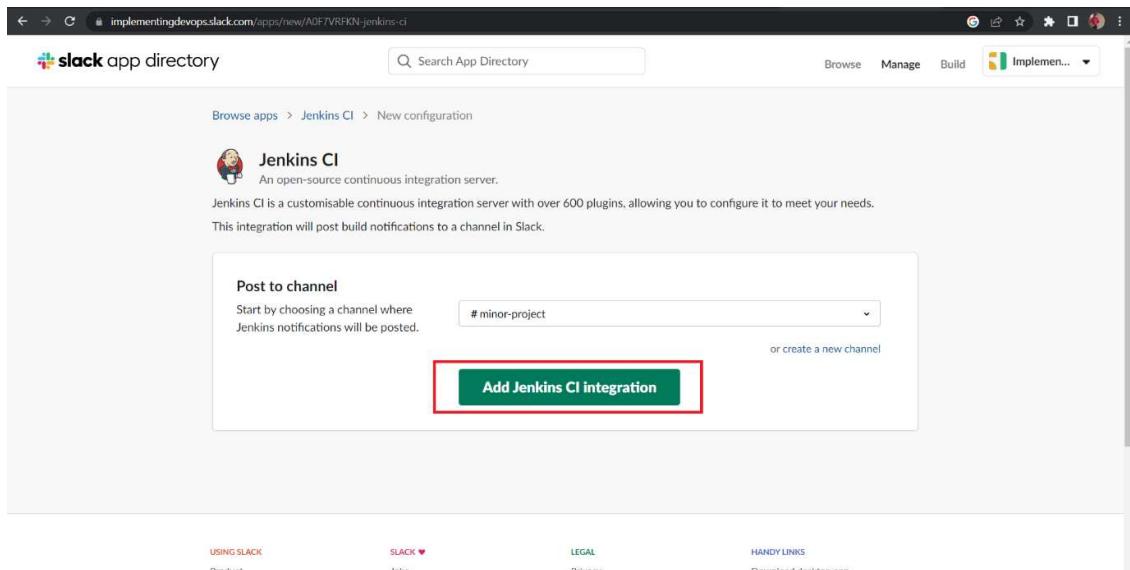


Figure 6.10.4 (Adding Jenkins CI integration)

5. Scroll down and copy token

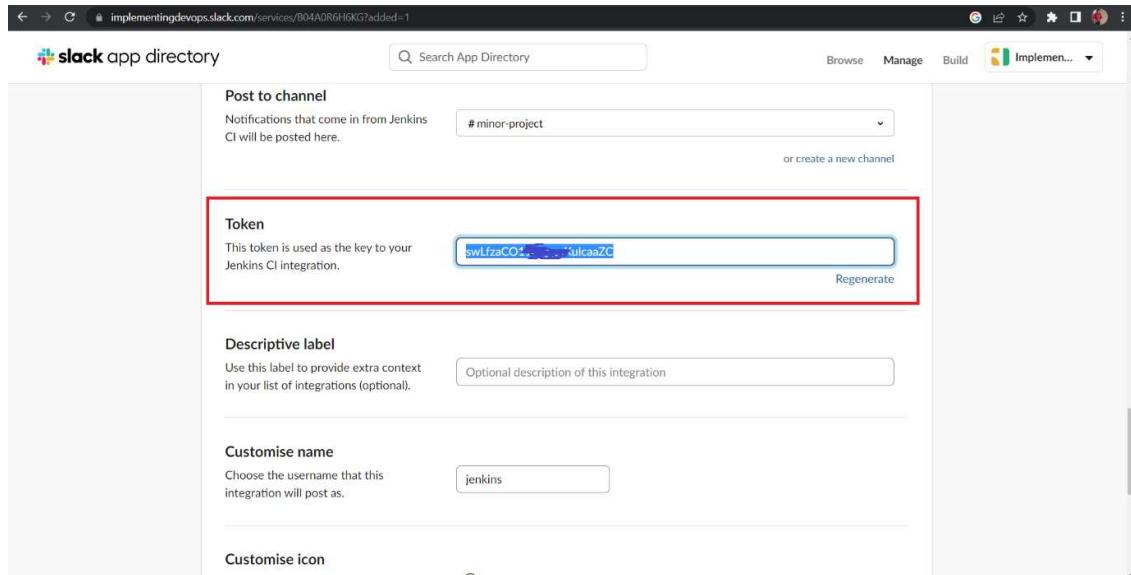


Figure 6.10.5 (Copy slack token)

6. In Jenkins provide go to configure system and search for slack

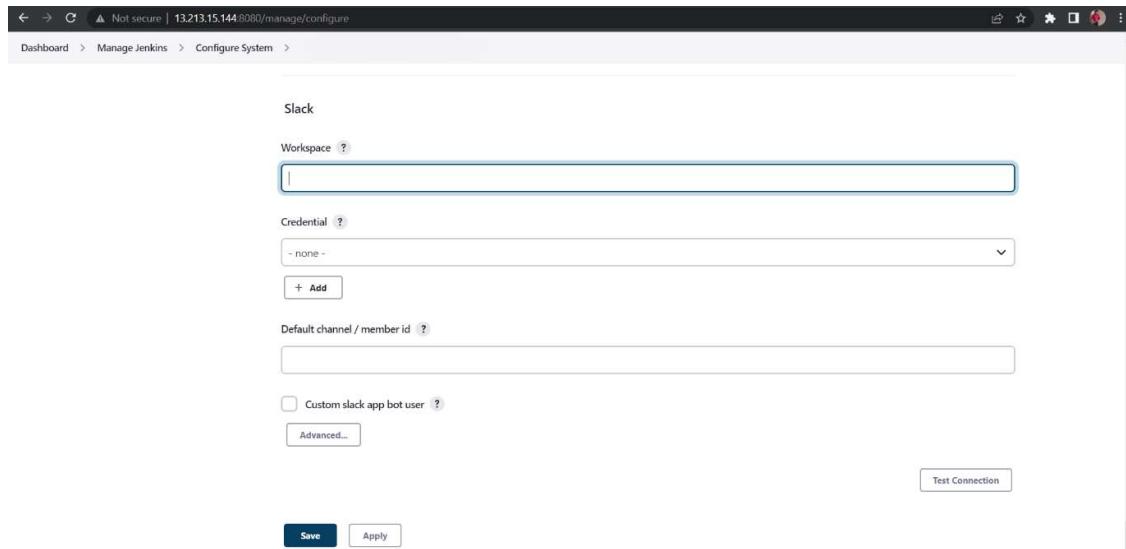


Figure 6.10.6 (configure slack)

7. Add workspace name and default channel

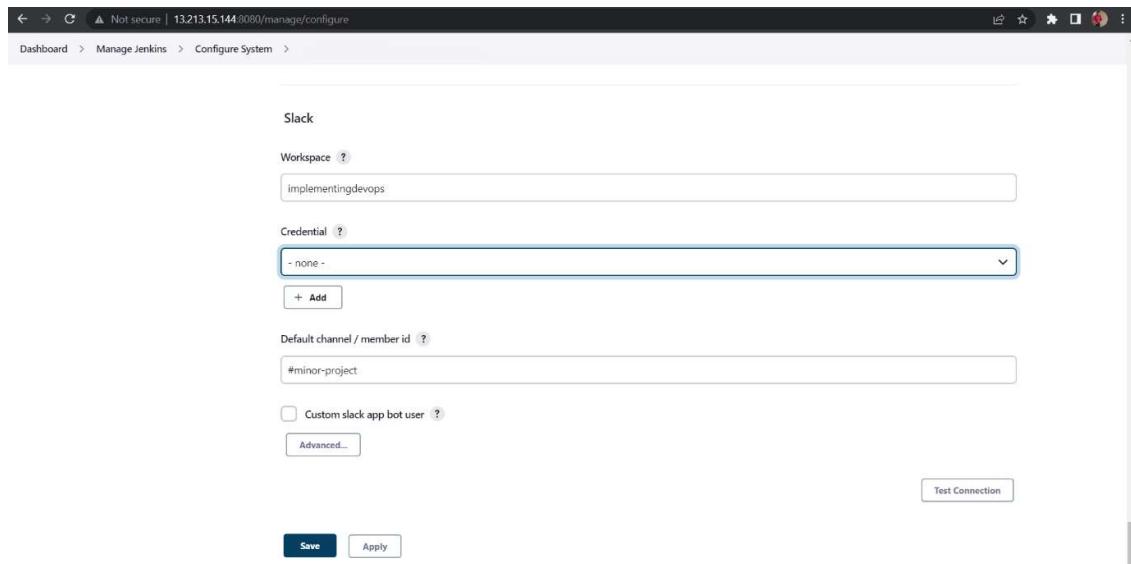


Figure 6.10.7 (Add workspace and channel name)

8. Add slack token as credential as a secret text and give it name as slacktoken

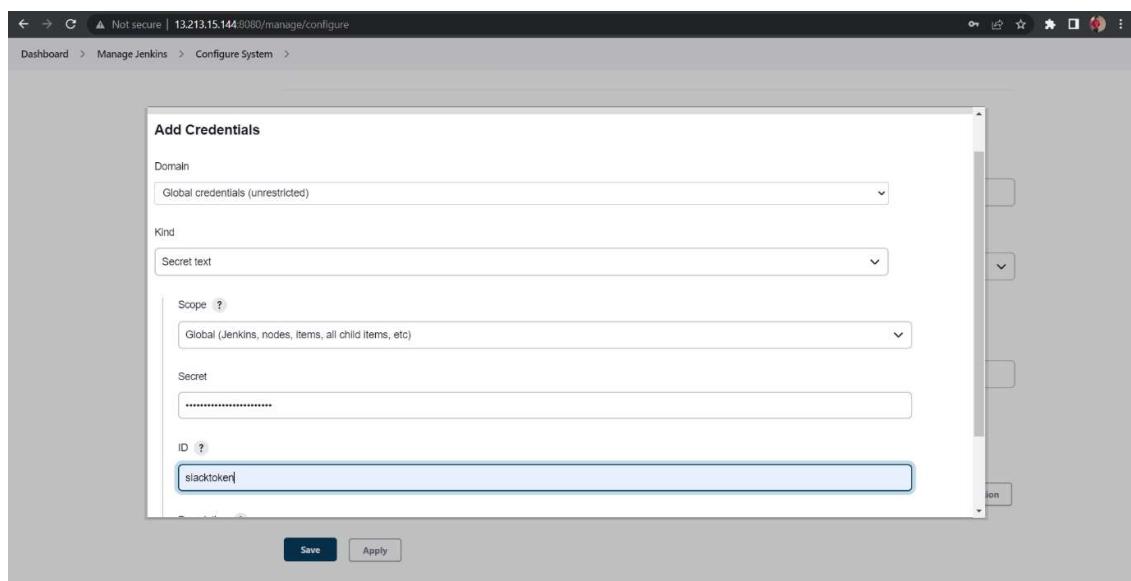


Figure 6.10.8 (Add slack credential)

9. Click on test connection if it is success or not

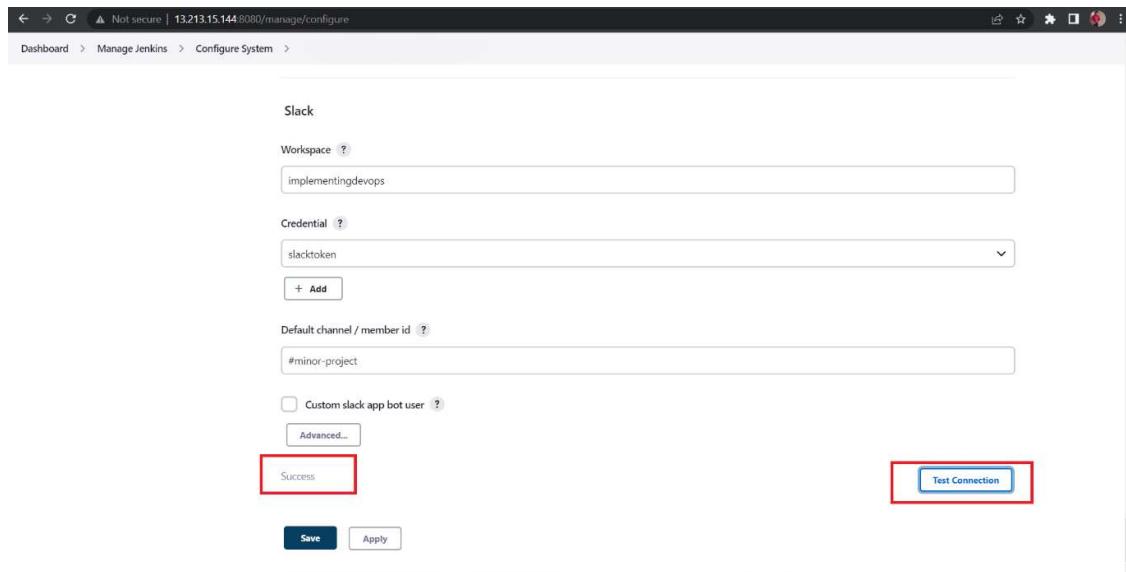


Figure 6.10.9 (Test slack connection)

10 Click on save and exit

6.11 Create Pipeline

Steps :-

1. From Jenkins home page click on New Item

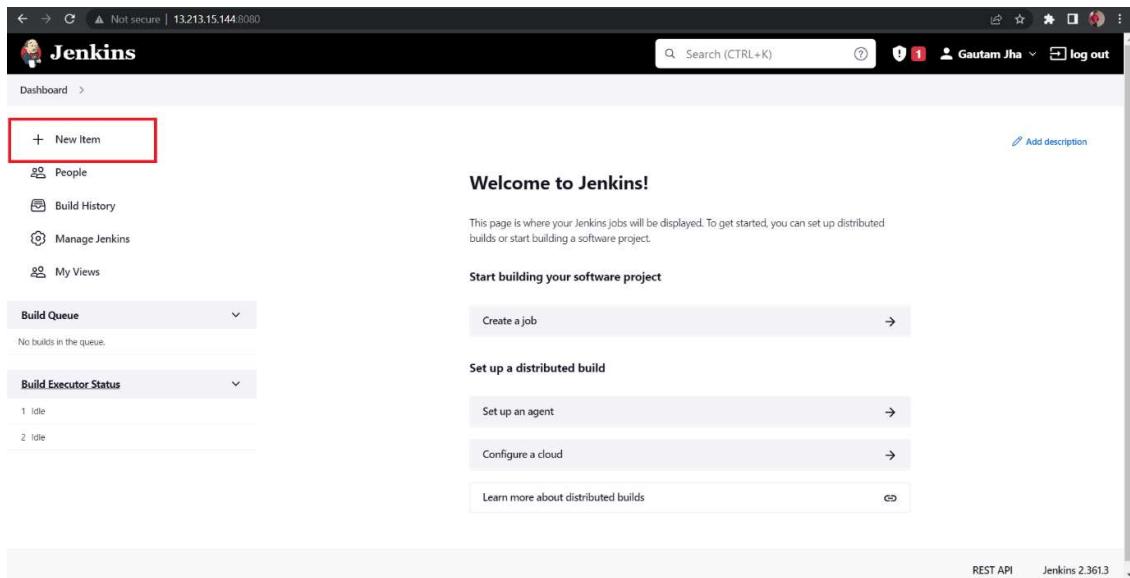


Figure 6.11.1 (Click on New item)

2. Select pipeline project and give name of the pipeline then click on OK

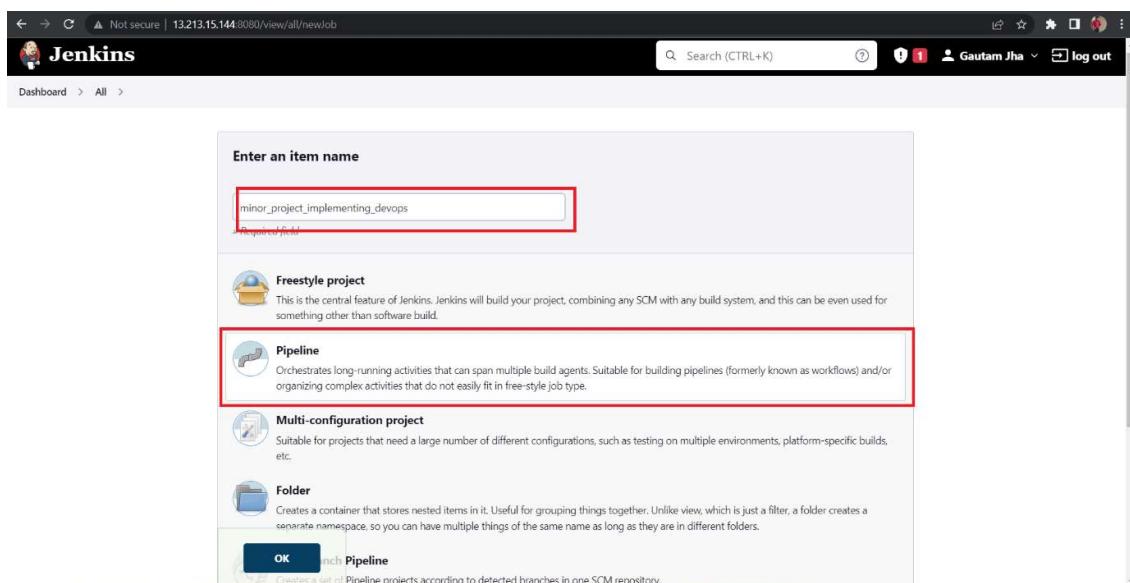


Figure 6.11.2 (pipeline project)

3. Give description and click on discard old builds and give value as 3 days with 3 builds

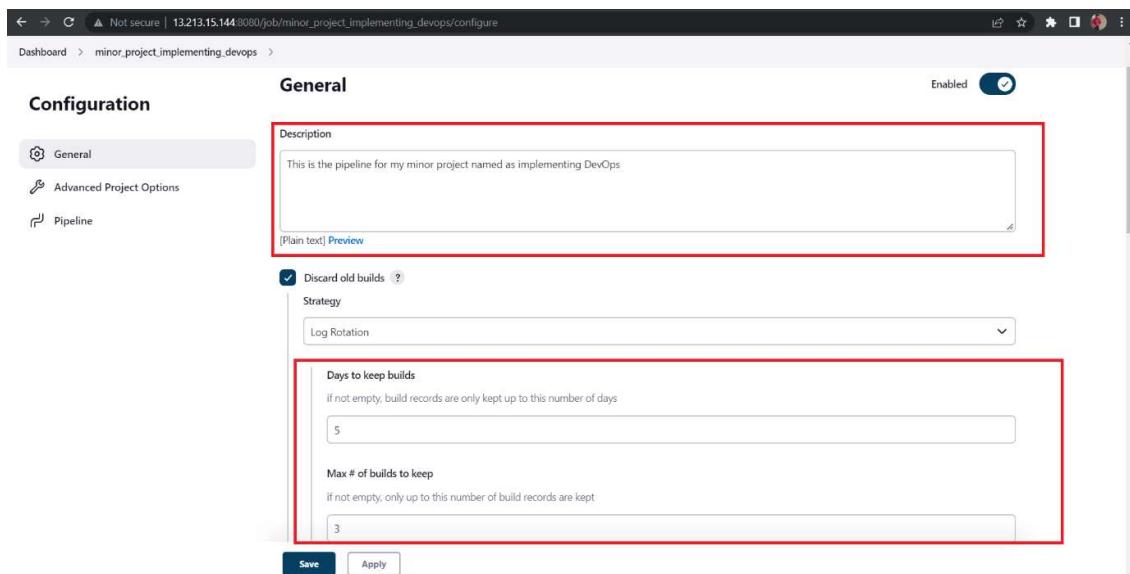


Figure 6.11.3 (providing general information to project)

4. In Build trigger Option choose poll scm and set it to poll every minute using 5 start (*)

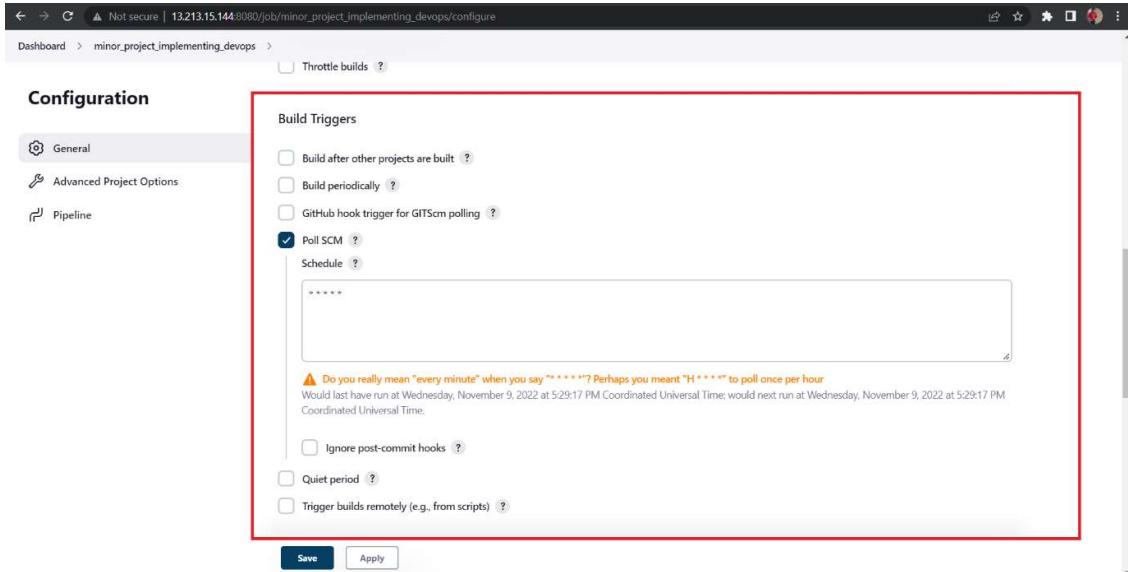


Figure 6.11.4 (Poll SCM every minute)

5. In pipeline section use pipeline script from SCM and choose Git as option and provide the repository URL

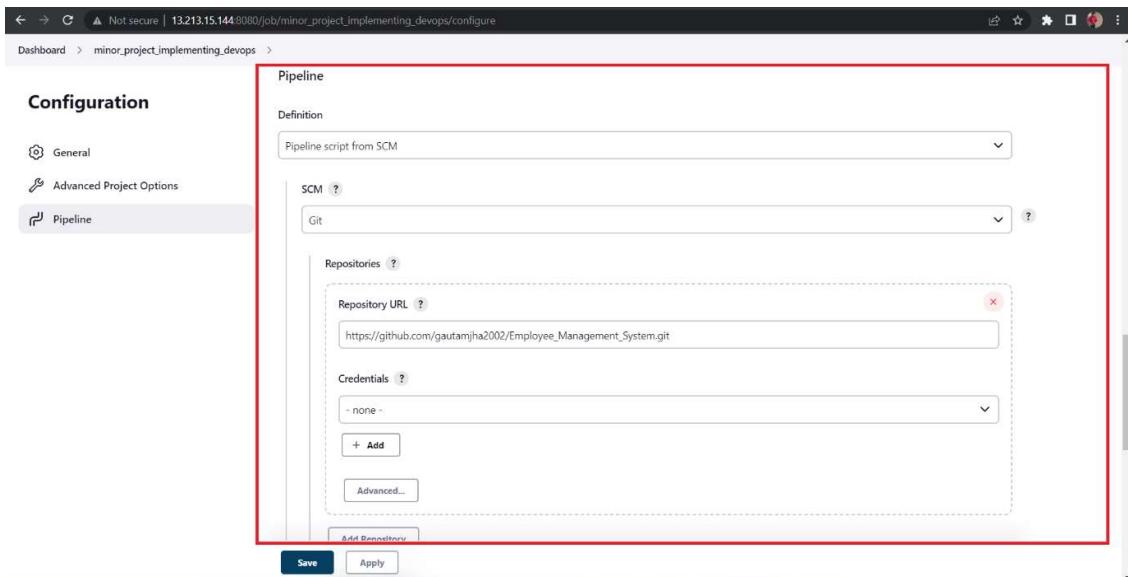


Figure 6.11.5 (Setup pipeline script from SCM)

6. Save this Item using save button

7. From Left side navigation click on build now option to build the pipeline

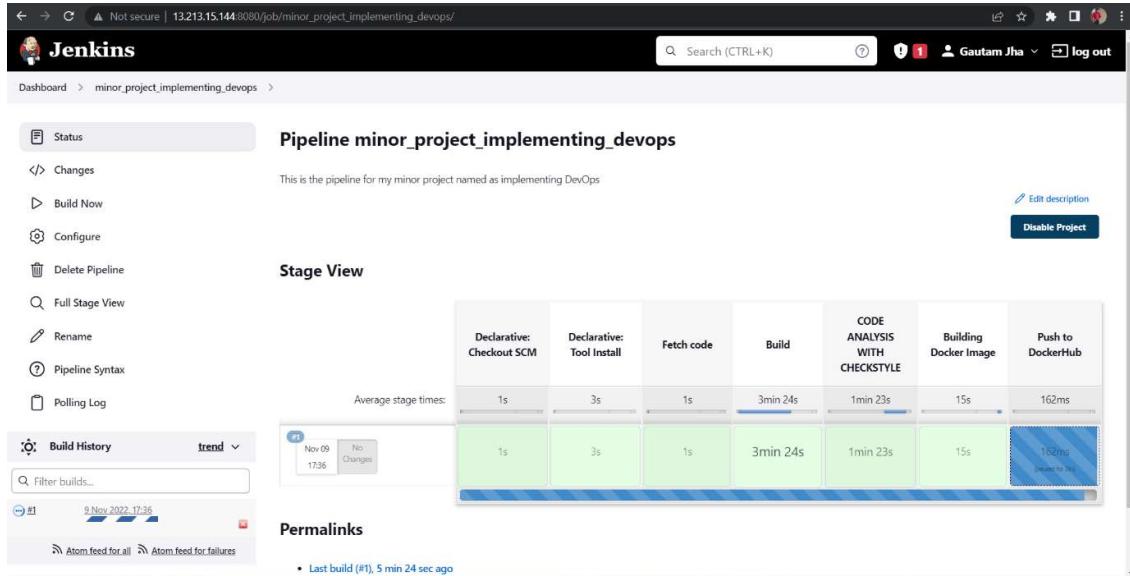


Figure 6.11.6 (Building pipeline)

8. Click on proceed to push the docker image to docker Hub

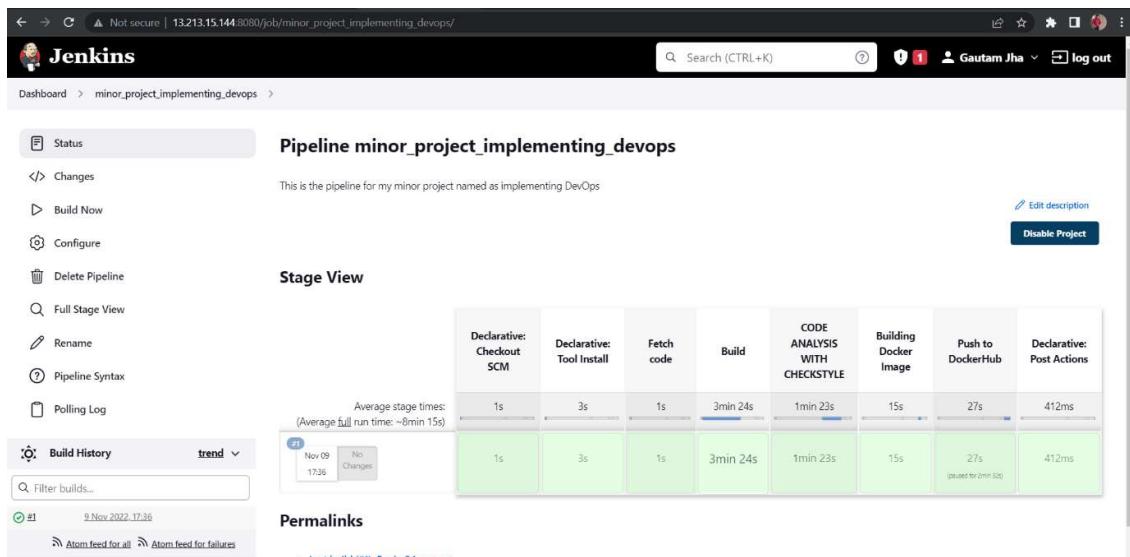


Figure 6.11.7 (Push to docker hub using pipeline)

9. Check the slack notification it will be send after the pipeline is finished

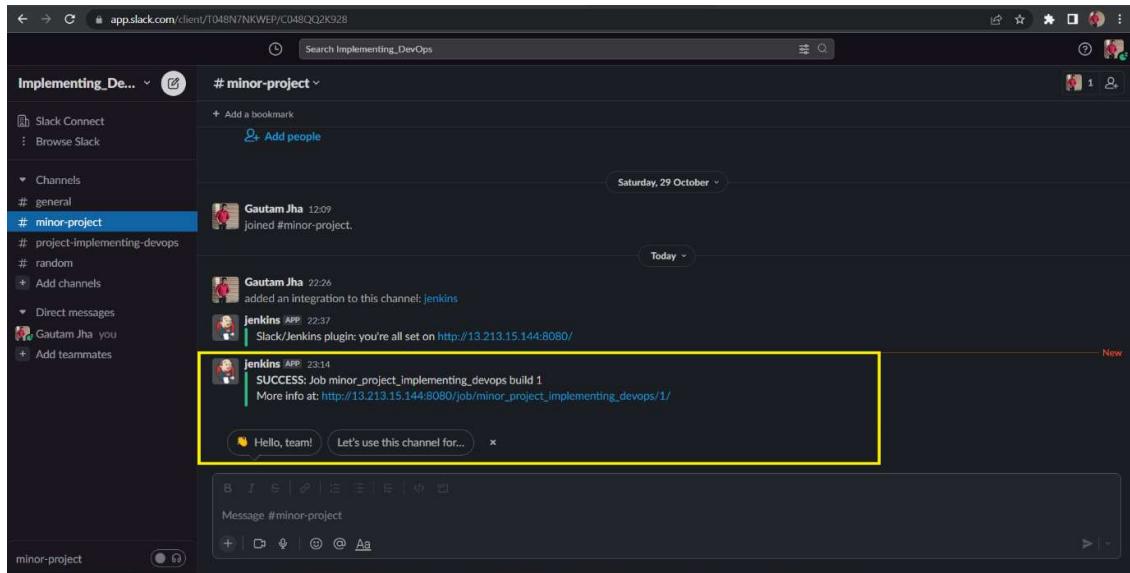


Figure 6.11.8 (Slack Notification)

10. Check Docker Hub there image of application will be pushed

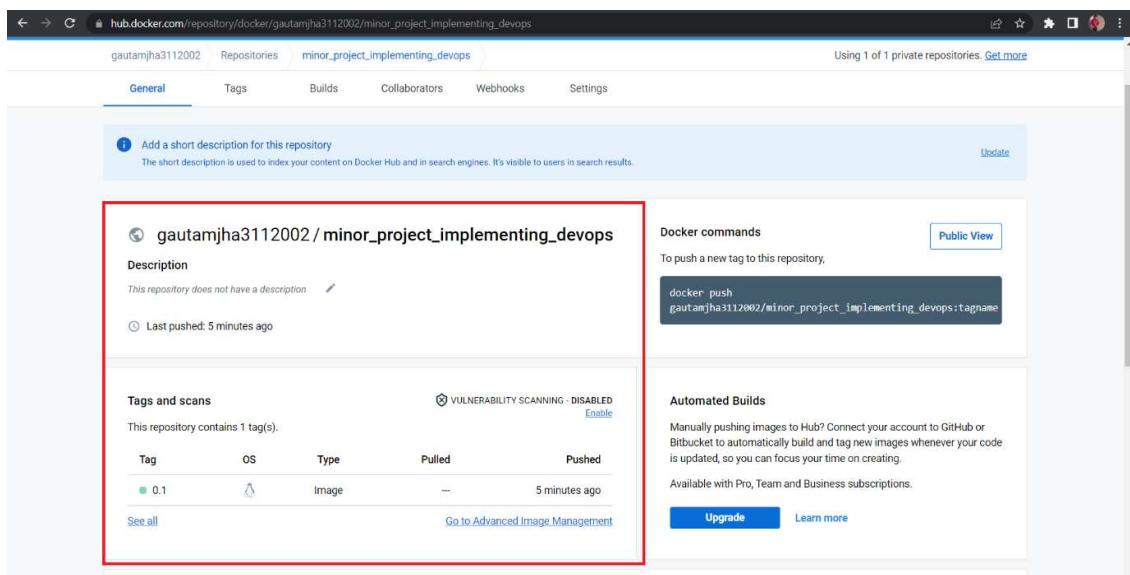
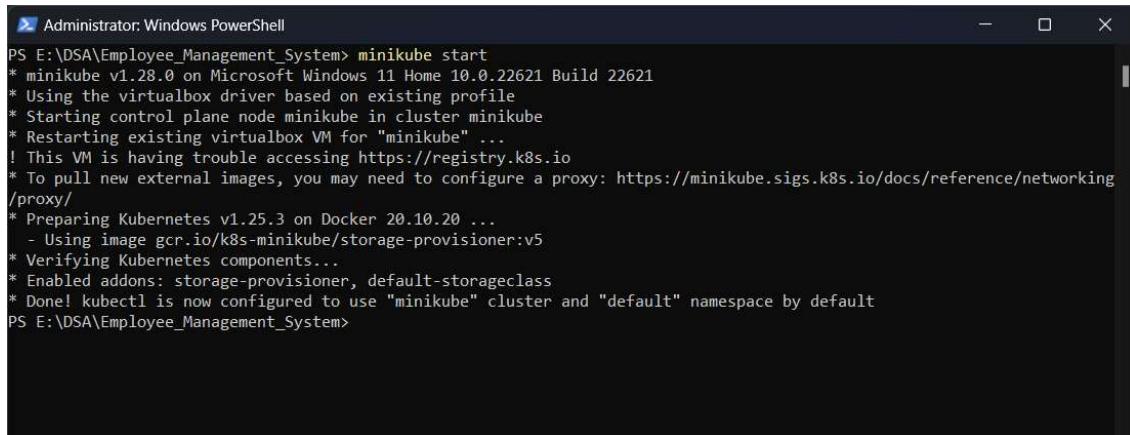


Figure 6.11.9 (Image to Docker Hub)

6.12 Deploy on Kubernetes (Minikube)

Steps :-

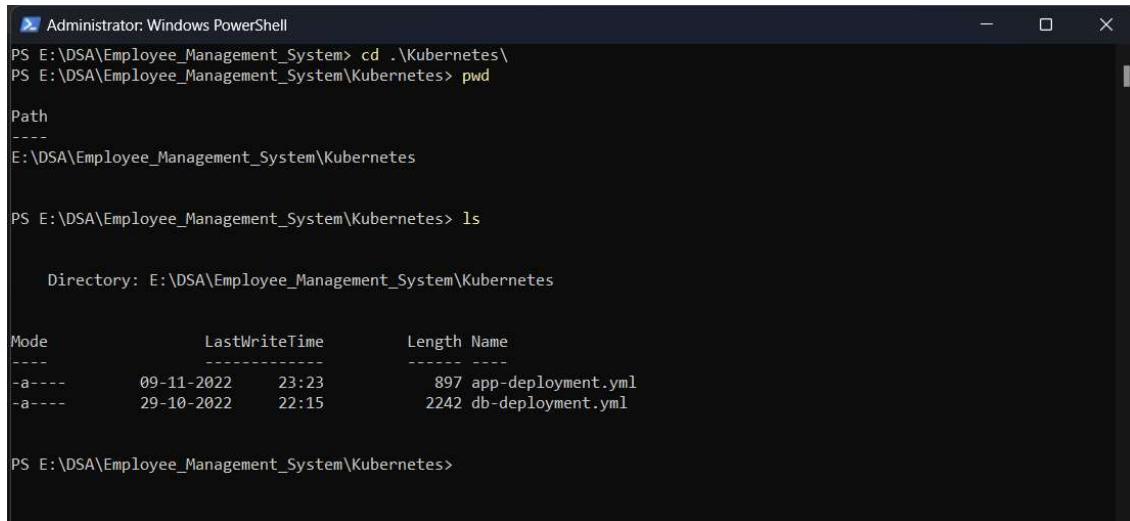
1. Start minikube using command minikube start



```
Administrator: Windows PowerShell
PS E:\DSA\Employee_Management_System> minikube start
* minikube v1.28.0 on Microsoft Windows 11 Home 10.0.22621 Build 22621
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Restarting existing virtualbox VM for "minikube" ...
! This VM is having trouble accessing https://registry.k8s.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS E:\DSA\Employee_Management_System>
```

Figure 6.12.1 (start minikube)

2. Change directory to Kubernetes to apply configuration



```
Administrator: Windows PowerShell
PS E:\DSA\Employee_Management_System> cd .\Kubernetes\
PS E:\DSA\Employee_Management_System\Kubernetes> pwd
Path
-----
E:\DSA\Employee_Management_System\Kubernetes

PS E:\DSA\Employee_Management_System\Kubernetes> ls

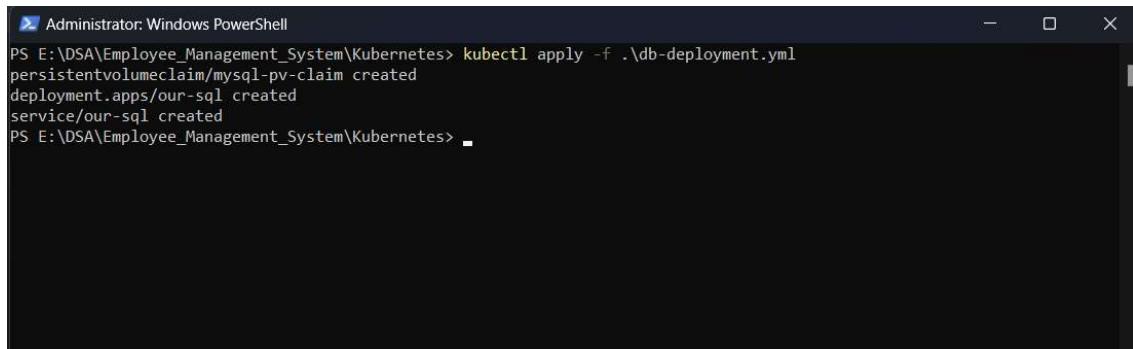
Directory: E:\DSA\Employee_Management_System\Kubernetes

Mode                LastWriteTime        Length Name
----                -----          ---- -
-a---     09-11-2022    23:23            897 app-deployment.yml
-a---     29-10-2022    22:15          2242 db-deployment.yml

PS E:\DSA\Employee_Management_System\Kubernetes>
```

Figure 6.12.2 (change dir to kubernetes)

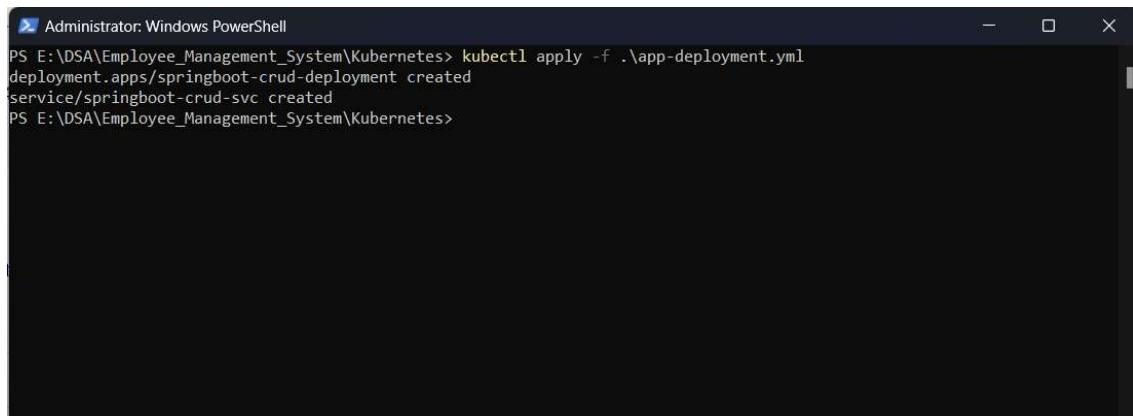
3. First create database deployment use command kubectl apply -f db-deployment.yml



```
Administrator: Windows PowerShell
PS E:\DSA\Employee_Management_System\Kubernetes> kubectl apply -f ./db-deployment.yml
persistentvolumeclaim/mysql-pv-claim created
deployment.apps/our-sql created
service/our-sql created
PS E:\DSA\Employee_Management_System\Kubernetes>
```

Figure 6.12.3 (create database deployment)

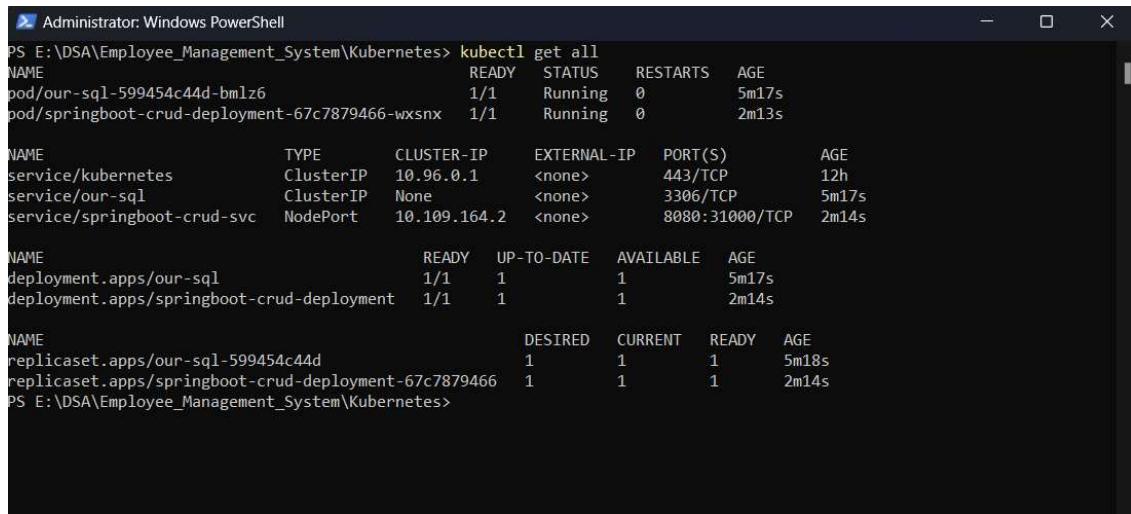
4. Now create deployment for the app using kubectl apply -f app-deployment.yml



```
Administrator: Windows PowerShell
PS E:\DSA\Employee_Management_System\Kubernetes> kubectl apply -f ./app-deployment.yml
deployment.apps/springboot-crud-deployment created
service/springboot-crud-svc created
PS E:\DSA\Employee_Management_System\Kubernetes>
```

Figure 6.12.4 (create deployment for app)

- Now check all the object running in the Kubernetes cluster using kubectl
get all



```
Administrator: Windows PowerShell
PS E:\DSA\Employee_Management_System\Kubernetes> kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/our-sql-599454c44d-bmlz6                1/1     Running   0          5m17s
pod/springboot-crud-deployment-67c7879466-wxsnx 1/1     Running   0          2m13s

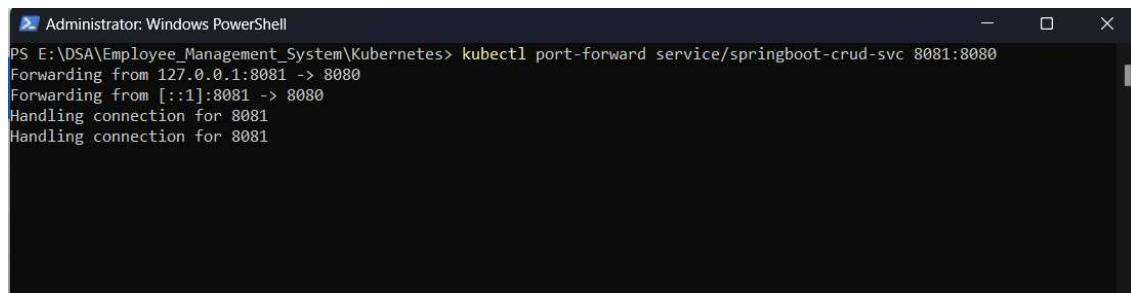
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP       12h
service/our-sql     ClusterIP   None         <none>        3306/TCP      5m17s
service/springboot-crud-svc   NodePort   10.109.164.2  <none>        8080:31000/TCP  2m14s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/our-sql   1/1      1           1          5m17s
deployment.apps/springboot-crud-deployment 1/1      1           1          2m14s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/our-sql-599454c44d   1         1         1          5m18s
replicaset.apps/springboot-crud-deployment-67c7879466 1         1         1          2m14s
PS E:\DSA\Employee_Management_System\Kubernetes>
```

Figure 6.12.5 (get all object of Kubernetes running on the cluster)

- Port-forward app service to be able to connect with localhost using
kubectl port-forward service/springboot-crud-svc 8081:8080



```
Administrator: Windows PowerShell
PS E:\DSA\Employee_Management_System\Kubernetes> kubectl port-forward service/springboot-crud-svc 8081:8080
Forwarding from 127.0.0.1:8081 -> 8080
Forwarding from [::1]:8081 -> 8080
Handling connection for 8081
Handling connection for 8081
```

Figure 6.12.6 (port forward app deployment service)

6.13 Access the application on localhost:8081

1. Home page

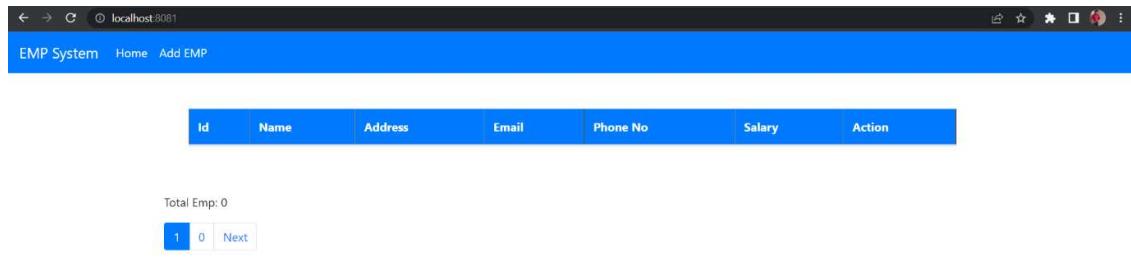


Figure 6.13.1 (Home page of application)

2. Add Employee

The screenshot shows a web browser window with the URL 'localhost:8081/addemp'. The title bar says 'EMP System - Home Add EMP'. The main content area is titled 'Add Emp' and contains five input fields: 'Enter Full Name' (Gautam Jha), 'Enter Address' (Delhi), 'Enter Email' (gautamjha3112002@gmail.com), 'Enter Phone No' (9650976986), and 'Enter Salary' (150000). A blue 'Submit' button is at the bottom.

Figure 6.13.2 (Add employee)

3. Return to homepage after adding employee

The screenshot shows a web browser window with the URL `localhost:8081` in the address bar. The title bar says "EMP System Home Add EMP". The main content area displays a table with one row of data:

ID	Name	Address	Email	Phone No	Salary	Action
1	Gautam Jha	Delhi	gautamjha3112002@gmail.com	9650976986	150000	Edit Delete

Total Emp: 1

1

Figure 6.13.3 (Home page)

4. Edit Employee

The screenshot shows a web browser window with the URL `localhost:8081/edit/1` in the address bar. The title bar says "EMP System Home Add EMP". The main content area displays a form titled "Edit Emp" with five input fields:

Enter Full Name
Gautam Jha

Enter Address
Delhi

Enter Email
gautamjha3112002@gmail.com

Enter Phone No
9650976986

Enter Salary
15000

Submit

Figure 6.13.4 (Edit the employee)

5. Return to homepage after Editing employee

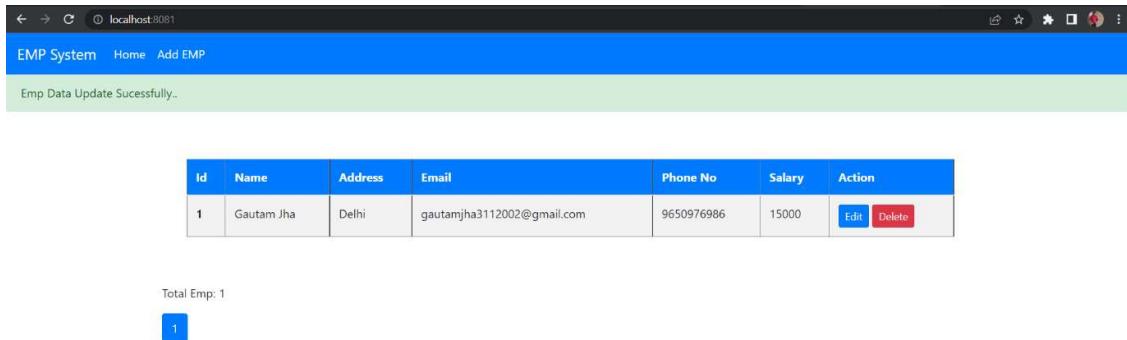


Figure 16.13.5 (Home page)

6. Delete Employee

- Homepage before deleting employee

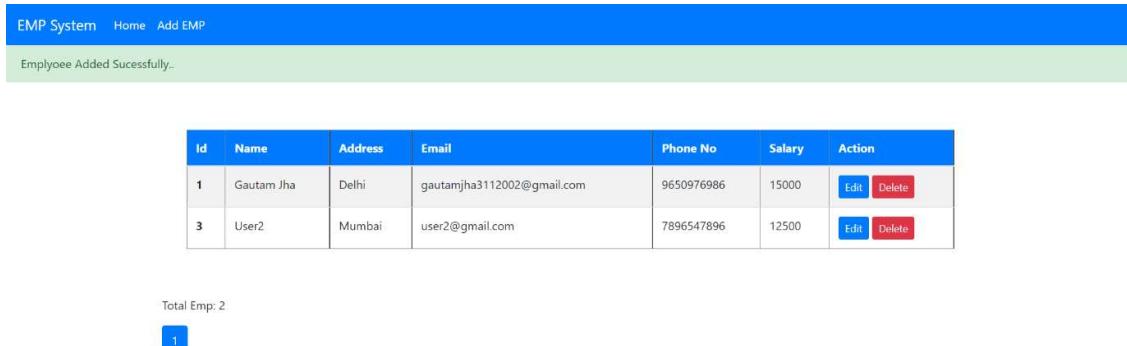


Figure 16.13.6 (home page before deleting employee)

➤ Homepage after deleting employee

The screenshot shows a web browser window with the URL `localhost:8081` in the address bar. The title bar says "EMP System Home Add EMP". A green banner at the top states "Emp Data Delete Sucessfully..". Below it is a table with one row of data. At the bottom left, it says "Total Emp: 1" with a blue button containing the number "1".

Id	Name	Address	Email	Phone No	Salary	Action
1	Gautam Jha	Delhi	gautamjha3112002@gmail.com	9650976986	15000	Edit Delete

Total Emp: 1

1

Figure 16.13.7 (Homepage after deleting employee)