Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 9**	
Title of Lab Assignment: To implement continuous deployment using Ansible/chef.		
DOP: 28-03-2024	**DOS: 03-04-2024**	
CO Mapped: **CO5**	**PO Mapped:** **PO1, PO2, PO3, PO5, PSO1, PSO2**	**Signature:**

Practical No. 9

Aim: To implement continuous deployment using Ansible/chef.

Introduction:

In the world of software development, continuous integration and continuous delivery (CI/CD) pipelines have become essential for efficient and rapid code delivery. CI/CD pipelines automate the process of building, testing, and deploying software, enabling faster product delivery to customers while minimizing manual intervention. In this blog,we will explore the stages of a CI/CD pipeline and demonstrate how to set up these stages using Ansible Playbook.

Goal of this ansible playbook:

The goal of the Ansible playbook mentioned in this blog is to automate the stages of the CI/CD pipeline using the principles of continuous integration and continuous delivery. By leveraging Ansible, we aim to achieve the following:

1. Install Required Software:
   The playbook will ensure that the necessary software, such as openjdk-8-jdk, git, maven, tomcat9, and tomcat9-admin, is installed on all relevant hosts.
2. Continuous Download and Continuous Build:
   The playbook will download the development code from a specified Git repository, create an artifact from the code,and fetch the artifact to the Ansible controller machine.
3. Continuous Deployment and Continuous Testing:
   The playbook will deploy the artifact into the Tomcat server on the QA server, download Selenium scripts from a specified Git repository, execute the Selenium scripts, and perform continuous testing.
4. Continuous Delivery:
   Finally, the playbook will deploy the artifact into the Tomcat server on the production server, enabling continuous delivery of the application.

By automating these stages with Ansible, we can achieve faster and more reliable software delivery, reduce manual intervention, and ensure consistent deployments across different environments.

CI/CD stages

A typical CI/CD pipeline includes the following stages. These stages make up the development lifecycle and workflow from source code to production:

Source:

1. Triggered when a developer commits code changes to a source code repository
2. Common triggers include scheduled or user-initiated workflows
3. Popular repository tools like GitHub and SVN are used

Build:

1. Creation and compilation of source code
2. Compiler gathers code features and dependencies, establishes links, and compiles into a new build
3. Quick detection and resolution of issues or conflicts in the build

Test:

1. Automated testing to validate code accuracy and ensure quality
2. Functional and unit testing for new features and functions
3. Integration and regression testing to ensure stability of existing features
4. Feedback loop for issue analysis and rectification in subsequent builds

Deliver:

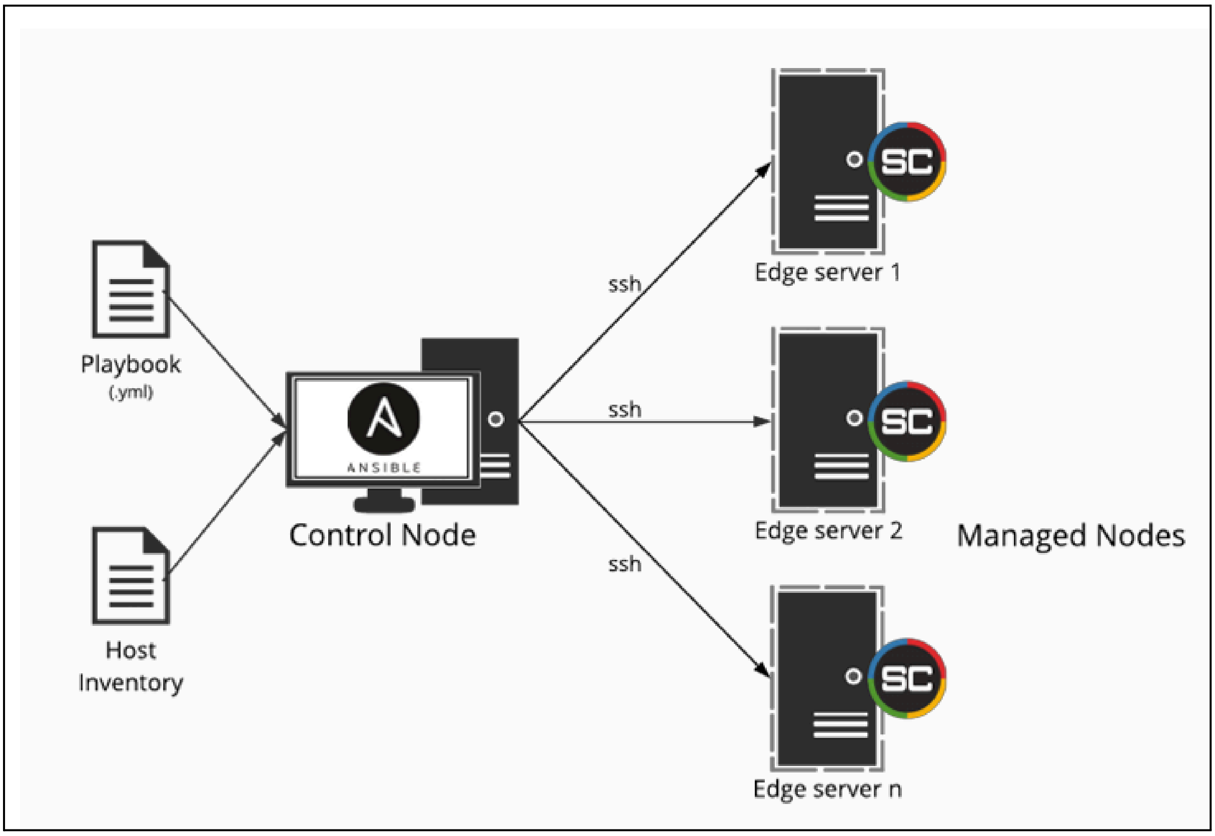
1. Approval of codebase from testing stage
2. Delivery to a runtime environment for integration, quality assurance, and pre production
3. Functional and performance testing conducted on the application

Deploy:

1. Deployment of code changes to the production environment
2. Orchestration of software releases by DevOps teams
3. Flexibility to schedule code deployments or target specific user groups
4. Rollback mechanism in case of issues

These above stages are created by using Ansible

Setting Up the Ansible Environment



Installing Ansible

1. Update the apt repository

   sudo apt-get update

2. Install software-properties-common

   sudo apt-get install -y software-properties-common

3. Add the latest version of Ansible to apt repository

   sudo apt-add-repository ppa:ansible/ansible

4. Install Ansible

   sudo apt-get install -y ansible

5. Update your ansible inventory hosts

   sudo vim /etc/ansible/host

```
[webserver]
172.31.30.86

[appserver]
172.31.92.137
```

[dbserver]

172.31.86.213

6. To check the version of Ansible

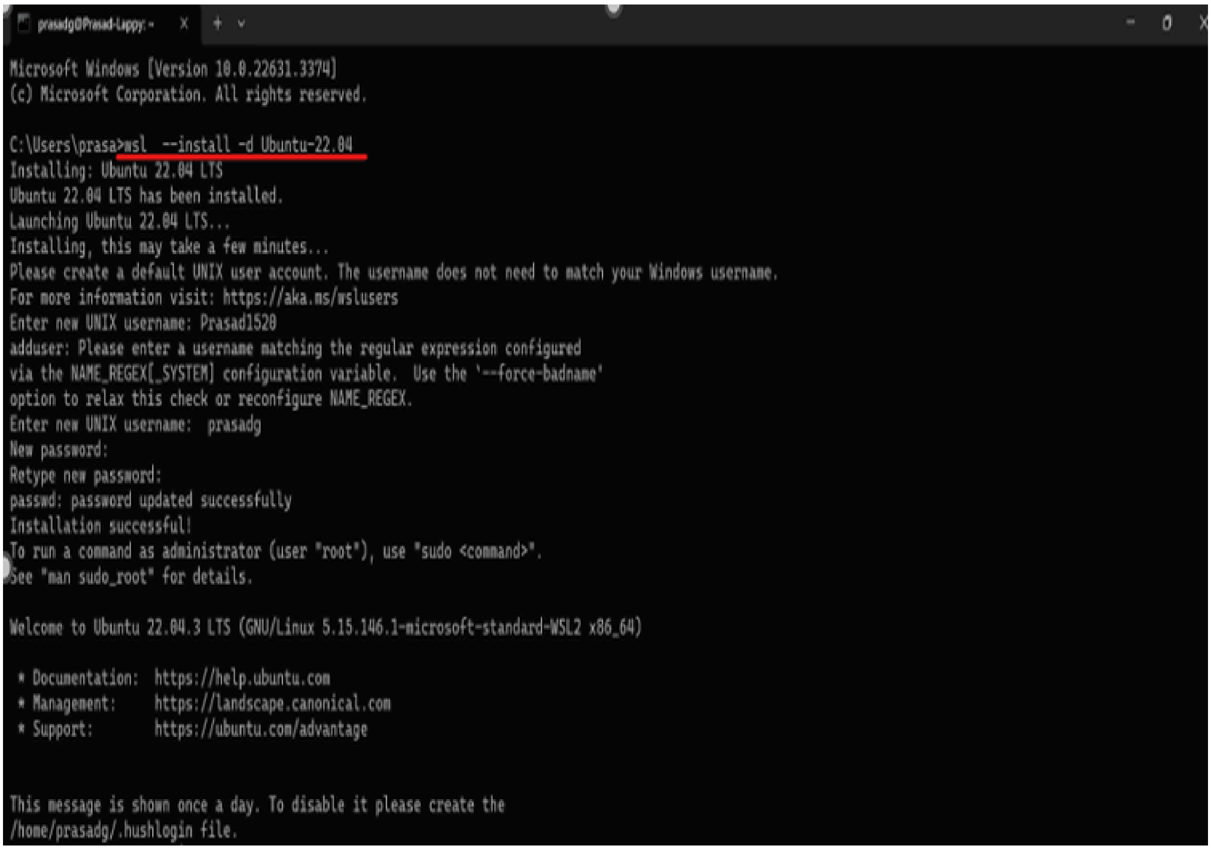
   ansible –version

Steps:

Step 1: To perform and install the ansible we need a linux server so to work on the linux server we will download the “Ubuntu 22.04” either from the command prompt or from the Microsoft store.

Command prompt:

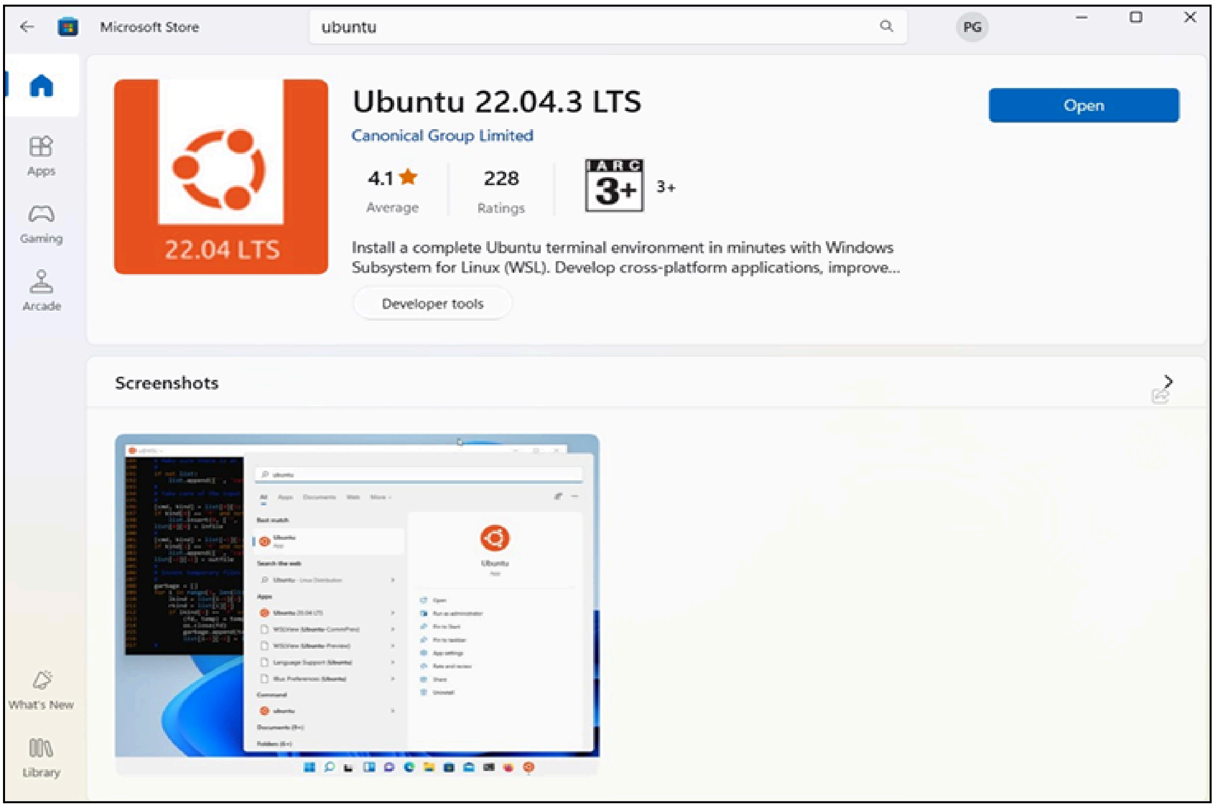
Run Command = wsl –install -d Ubuntu-22.04



```
Microsoft Windows [Version 10.0.22631.3374]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prasa>wsl  --install -d Ubuntu-22.04
Installing: Ubuntu 22.04 LTS
Ubuntu 22.04 LTS has been installed.
Launching Ubuntu 22.04 LTS...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: Prasad1520
adduser: Please enter a username matching the regular expression configured
via the NAME_REGEX[_SYSTEM] configuration variable.  Use the '--force-badname'
option to relax this check or reconfigure NAME_REGEX.
Enter new UNIX username:  prasadg
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.146.1-microsoft-standard-WSL2 x86_64)

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


This message is shown once a day. To disable it please create the
/home/prasadg/.hushlogin file.
```

Through Microsoft store:

Open the Microsoft store and search for unbuntu and then click on install.



Ubuntu 22.04.3 LTS

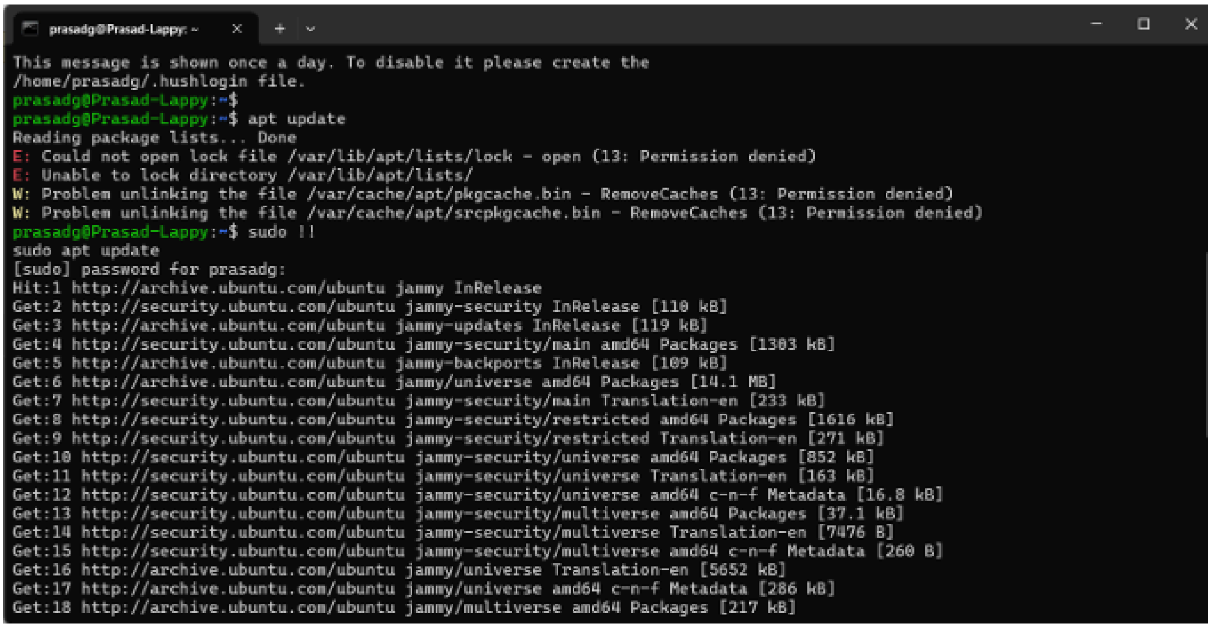
Canonical Group Limited

4.1 ★ Average · 228 Ratings · IARC 3+

Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform applications, improve...

Step 2: Once the installation is successful, open the ubuntu command line and perform below commands.

1. apt update (used to update the list of available packages)
2. sudo !! (re-run the previous command with sudo privileges)

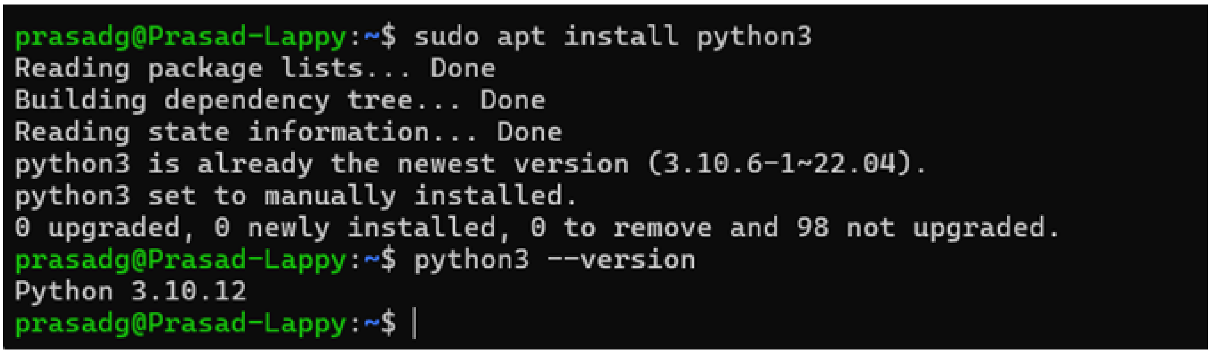


```
This message is shown once a day. To disable it please create the
/home/prasadg/.hushlogin file.
prasadg@Prasad-Lappy:~$
prasadg@Prasad-Lappy:~$ apt update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)
prasadg@Prasad-Lappy:~$ sudo !!
sudo apt update
[sudo] password for prasadg:
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1303 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [233 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1616 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [271 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [852 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [163 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
```

Step 3: We need to install the python before installing ansible as it is one of the pre-requisites.

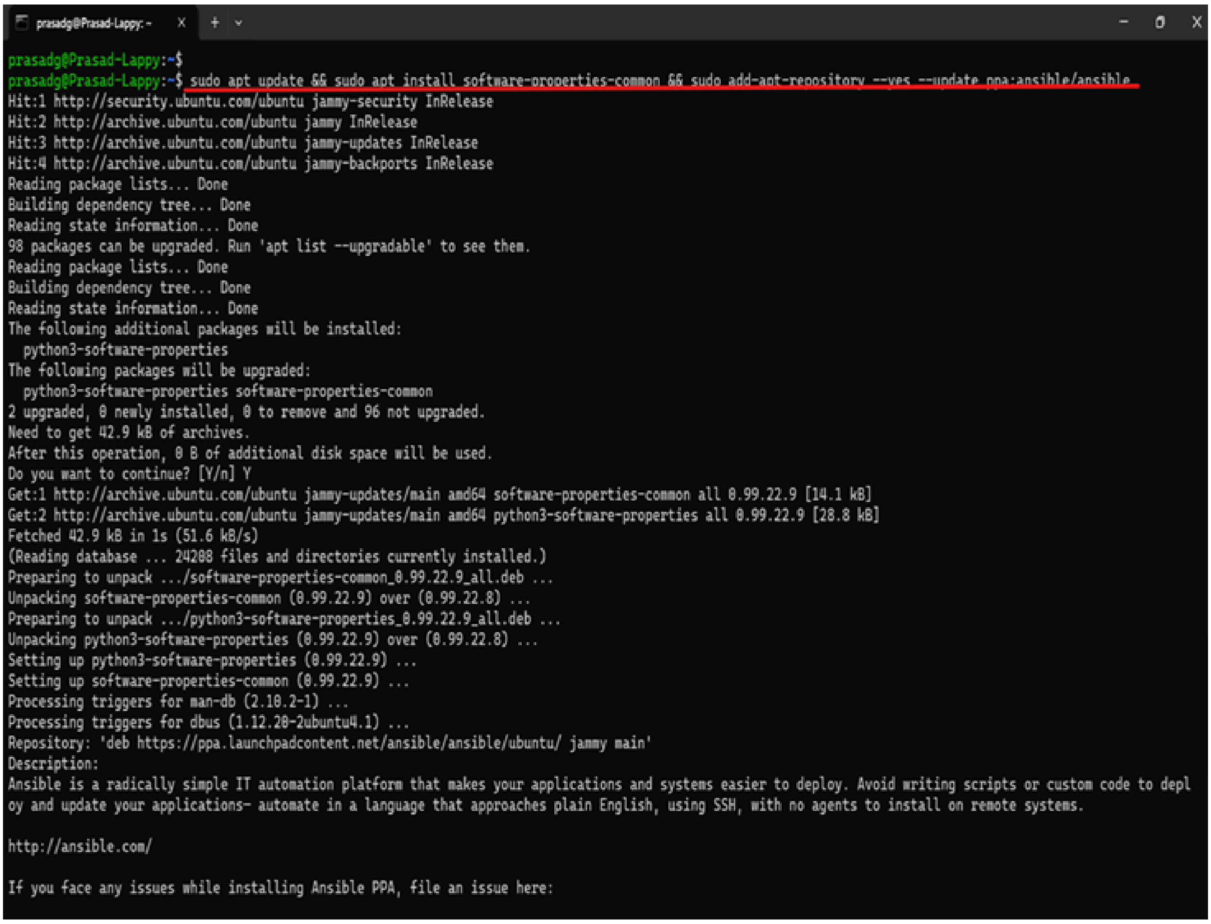
Command: sudo apt install python3 (Installation)

Command: python3 —-version (To Check the version)



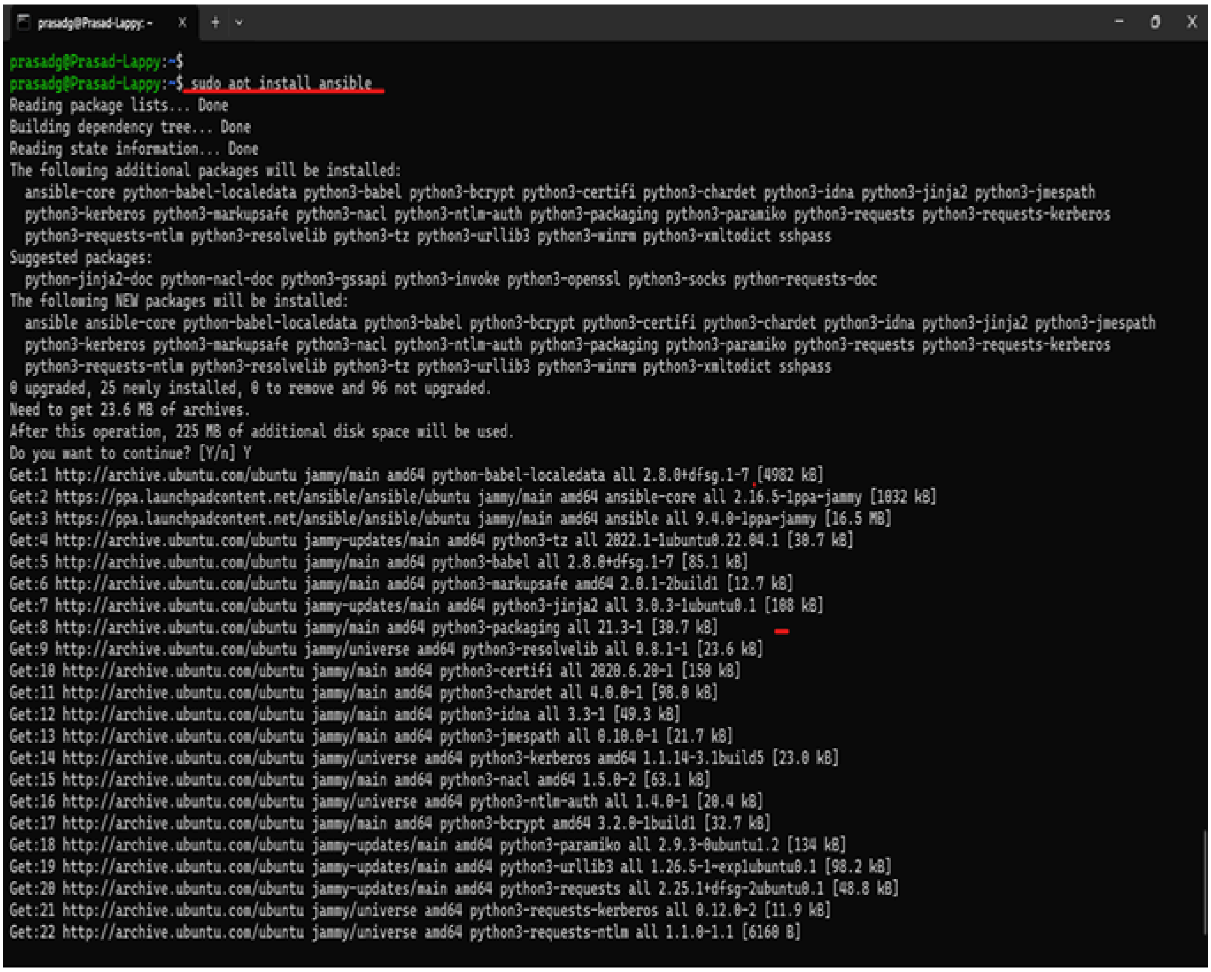
```
prasadg@Prasad-Lappy:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 98 not upgraded.
prasadg@Prasad-Lappy:~$ python3 --version
Python 3.10.12
prasadg@Prasad-Lappy:~$ |
```

Step4: Use below command to update the ansible packages available in linux



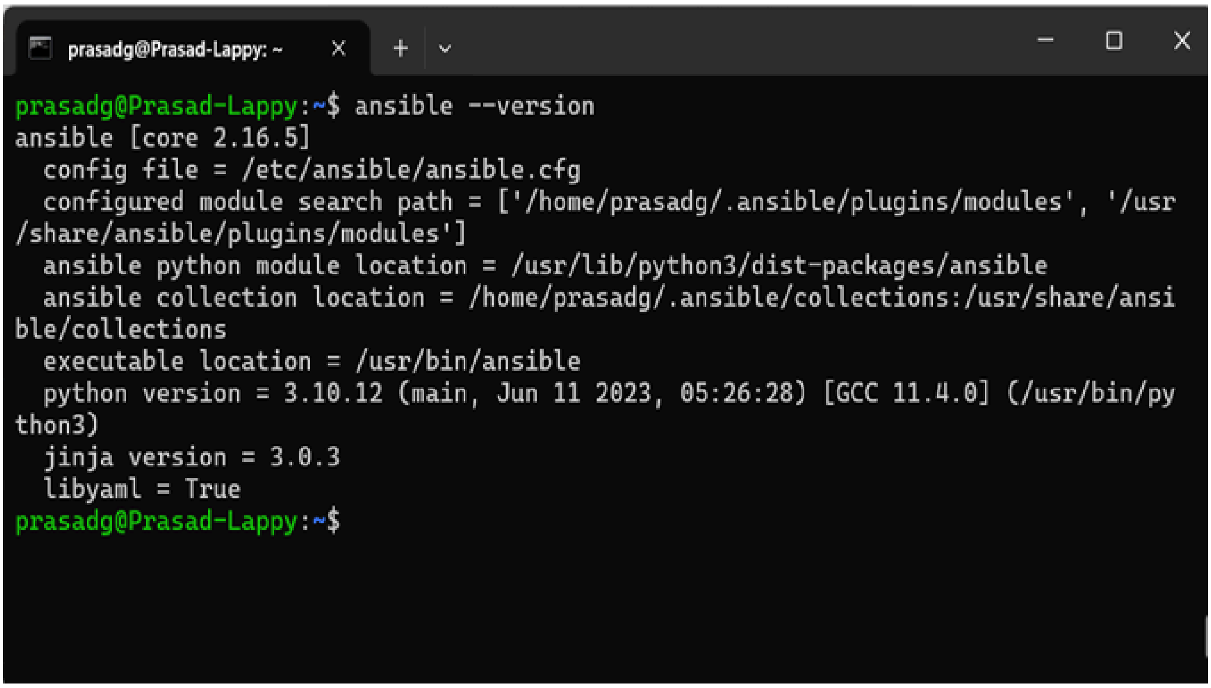
```
prasadg@Prasad-Lappy:~$
prasadg@Prasad-Lappy:~$ sudo apt update && sudo apt install software-properties-common && sudo add-apt-repository --yes --update ppa:ansible/ansible
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
98 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 0 newly installed, 0 to remove and 96 not upgraded.
Need to get 42.9 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 software-properties-common all 0.99.22.9 [14.1 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-software-properties all 0.99.22.9 [28.8 kB]
Fetched 42.9 kB in 1s (51.6 kB/s)
(Reading database ... 24208 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.99.22.9_all.deb ...
Unpacking software-properties-common (0.99.22.9) over (0.99.22.8) ...
Preparing to unpack .../python3-software-properties_0.99.22.9_all.deb ...
Unpacking python3-software-properties (0.99.22.9) over (0.99.22.8) ...
Setting up python3-software-properties (0.99.22.9) ...
Setting up software-properties-common (0.99.22.9) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to depl
oy and update your applications- automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
```

Step 5: Install ansible with below commands.



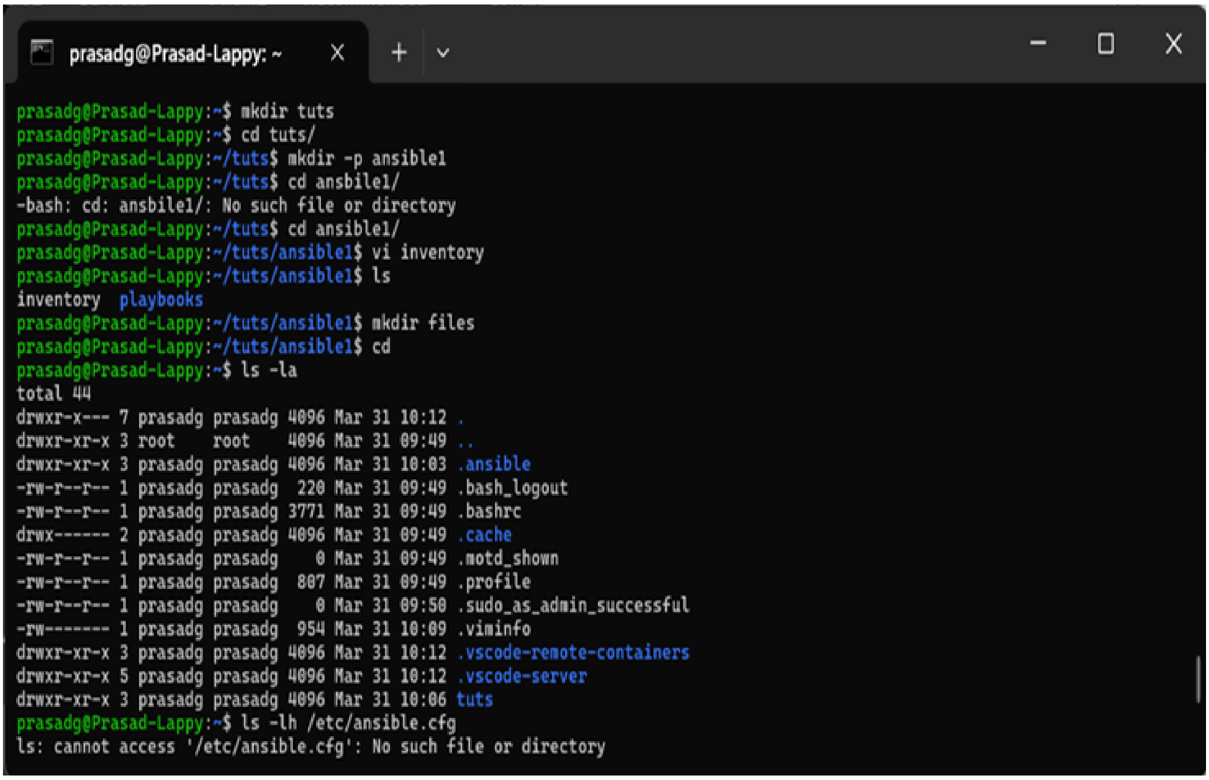
```
prasadg@Prasad-Lappy:~$
prasadg@Prasad-Lappy:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python-babel-localedata python3-babel python3-bcrypt python3-certifi python3-chardet python3-idna python3-jinja2 python3-jmespath
  python3-kerberos python3-markupsafe python3-nacl python3-ntlm-auth python3-packaging python3-paramiko python3-requests python3-requests-kerberos
  python3-requests-ntlm python3-resolvelib python3-tz python3-urllib3 python3-winrm python3-xmltodict sshpass
Suggested packages:
  python-jinja2-doc python-nacl-doc python3-gssapi python3-invoke python3-openssl python3-socks python-requests-doc
The following NEW packages will be installed:
  ansible ansible-core python-babel-localedata python3-babel python3-bcrypt python3-certifi python3-chardet python3-idna python3-jinja2 python3-jmespath
  python3-kerberos python3-markupsafe python3-nacl python3-ntlm-auth python3-packaging python3-paramiko python3-requests python3-requests-kerberos
  python3-requests-ntlm python3-resolvelib python3-tz python3-urllib3 python3-winrm python3-xmltodict sshpass
0 upgraded, 25 newly installed, 0 to remove and 96 not upgraded.
Need to get 23.6 MB of archives.
After this operation, 225 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 python-babel-localedata all 2.8.0+dfsg.1-7 [4982 kB]
Get:2 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible-core all 2.16.5-1ppa~jammy [1032 kB]
Get:3 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible all 9.4.0-1ppa~jammy [16.5 MB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-tz all 2022.1-1ubuntu0.22.04.1 [30.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-babel all 2.8.0+dfsg.1-7 [85.1 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-markupsafe amd64 2.0.1-2build1 [12.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-jinja2 all 3.0.3-1ubuntu0.1 [108 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-packaging all 21.3-1 [30.7 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 python3-resolvelib all 0.8.1-1 [23.6 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-certifi all 2020.6.20-1 [150 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-chardet all 4.0.0-1 [98.0 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-idna all 3.3-1 [49.3 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-jmespath all 0.10.0-1 [21.7 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 python3-kerberos amd64 1.1.14-3.1build5 [23.0 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-nacl amd64 1.5.0-2 [63.1 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 python3-ntlm-auth all 1.4.0-1 [20.4 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-bcrypt amd64 3.2.0-1build1 [32.7 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-paramiko all 2.9.3-0ubuntu1.2 [134 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-urllib3 all 1.26.5-1~exp1ubuntu0.1 [98.2 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-requests all 2.25.1+dfsg-2ubuntu0.1 [48.8 kB]
Get:21 http://archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-ntlm all 1.1.0-1.1 [6160 B]
```

Step 6: Check the ansible version.



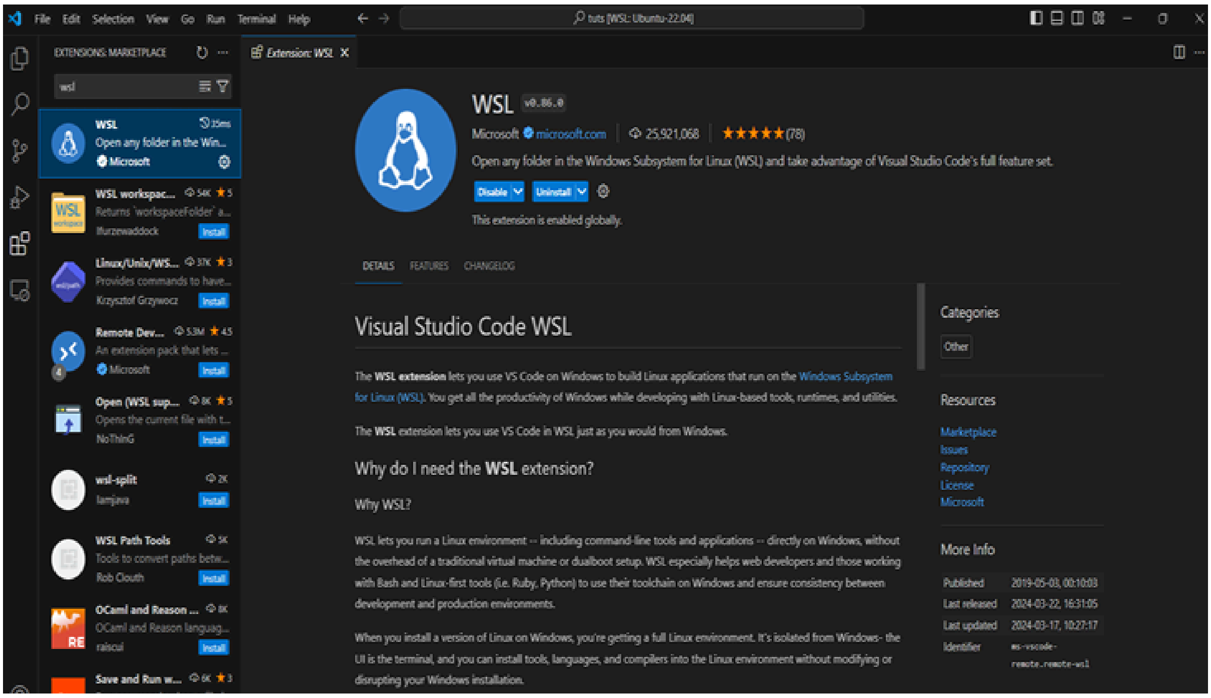
```
prasadg@Prasad-Lappy:~$ ansible --version
ansible [core 2.16.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/prasadg/.ansible/plugins/modules', '/usr
/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/prasadg/.ansible/collections:/usr/share/ansi
ble/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] (/usr/bin/py
thon3)
  jinja version = 3.0.3
  libyaml = True
prasadg@Prasad-Lappy:~$
```

Step 7: Create a new directory and follow the below steps to create new files.



```
prasadg@Prasad-Lappy:~$ mkdir tuts
prasadg@Prasad-Lappy:~$ cd tuts/
prasadg@Prasad-Lappy:~/tuts$ mkdir -p ansible1
prasadg@Prasad-Lappy:~/tuts$ cd ansbile1/
-bash: cd: ansbile1/: No such file or directory
prasadg@Prasad-Lappy:~/tuts$ cd ansible1/
prasadg@Prasad-Lappy:~/tuts/ansible1$ vi inventory
prasadg@Prasad-Lappy:~/tuts/ansible1$ ls
inventory  playbooks
prasadg@Prasad-Lappy:~/tuts/ansible1$ mkdir files
prasadg@Prasad-Lappy:~/tuts/ansible1$ cd
prasadg@Prasad-Lappy:~$ ls -la
total 44
drwxr-x--- 7 prasadg prasadg 4096 Mar 31 10:12 .
drwxr-xr-x 3 root    root    4096 Mar 31 09:49 ..
drwxr-xr-x 3 prasadg prasadg 4096 Mar 31 10:03 .ansible
-rw-r--r-- 1 prasadg prasadg  220 Mar 31 09:49 .bash_logout
-rw-r--r-- 1 prasadg prasadg 3771 Mar 31 09:49 .bashrc
drwx------ 2 prasadg prasadg 4096 Mar 31 09:49 .cache
-rw-r--r-- 1 prasadg prasadg    0 Mar 31 09:49 .motd_shown
-rw-r--r-- 1 prasadg prasadg  807 Mar 31 09:49 .profile
-rw-r--r-- 1 prasadg prasadg    0 Mar 31 09:50 .sudo_as_admin_successful
-rw------- 1 prasadg prasadg  954 Mar 31 10:09 .viminfo
drwxr-xr-x 3 prasadg prasadg 4096 Mar 31 10:12 .vscode-remote-containers
drwxr-xr-x 5 prasadg prasadg 4096 Mar 31 10:12 .vscode-server
drwxr-xr-x 3 prasadg prasadg 4096 Mar 31 10:06 tuts
prasadg@Prasad-Lappy:~$ ls -lh /etc/ansible.cfg
ls: cannot access '/etc/ansible.cfg': No such file or directory
```

Step 8: To create a playbook and other files we will use vscode. Open the vscode and install the extension for WSL.


File Edit Selection View Go Run Terminal Help
tuts [WSL: Ubuntu-22.04]
EXTENSIONS: MARKETPLACE
wsl
WSL 35ms Open any folder in the Win... Microsoft
WSL workspac... 54K ★5 Returns `workspaceFolder` a... lfurzewaddock Install
Linux/Unix/WS... 37K ★3 Provides commands to have... Krzysztof Grzywocz Install
Remote Dev... 5.3M ★4.5 An extension pack that lets ... Microsoft Install
Open (WSL sup... 8K ★5 Opens the current file with t... NoThinG Install
wsl-split 2K lamjava Install
WSL Path Tools 5K Tools to convert paths betw... Rob Clouth Install
OCaml and Reason ... 8K OCaml and Reason languag... raiscui Install
Save and Run w... 6K ★3
Extension: WSL
Disable
Uninstall
DETAILS
FEATURES
CHANGELOG

WSL v0.86.0

Microsoft microsoft.com | 25,921,068 | ★★★★★ (78)

Open any folder in the Windows Subsystem for Linux (WSL) and take advantage of Visual Studio Code's full feature set.

This extension is enabled globally.

Visual Studio Code WSL

The **WSL extension** lets you use VS Code on Windows to build Linux applications that run on the Windows Subsystem for Linux (WSL). You get all the productivity of Windows while developing with Linux-based tools, runtimes, and utilities.

The **WSL** extension lets you use VS Code in WSL just as you would from Windows.

Why do I need the **WSL** extension?

Why WSL?

WSL lets you run a Linux environment -- including command-line tools and applications -- directly on Windows, without the overhead of a traditional virtual machine or dualboot setup. WSL especially helps web developers and those working with Bash and Linux-first tools (i.e. Ruby, Python) to use their toolchain on Windows and ensure consistency between development and production environments.

When you install a version of Linux on Windows, you're getting a full Linux environment. It's isolated from Windows- the UI is the terminal, and you can install tools, languages, and compilers into the Linux environment without modifying or disrupting your Windows installation.

Categories

Other

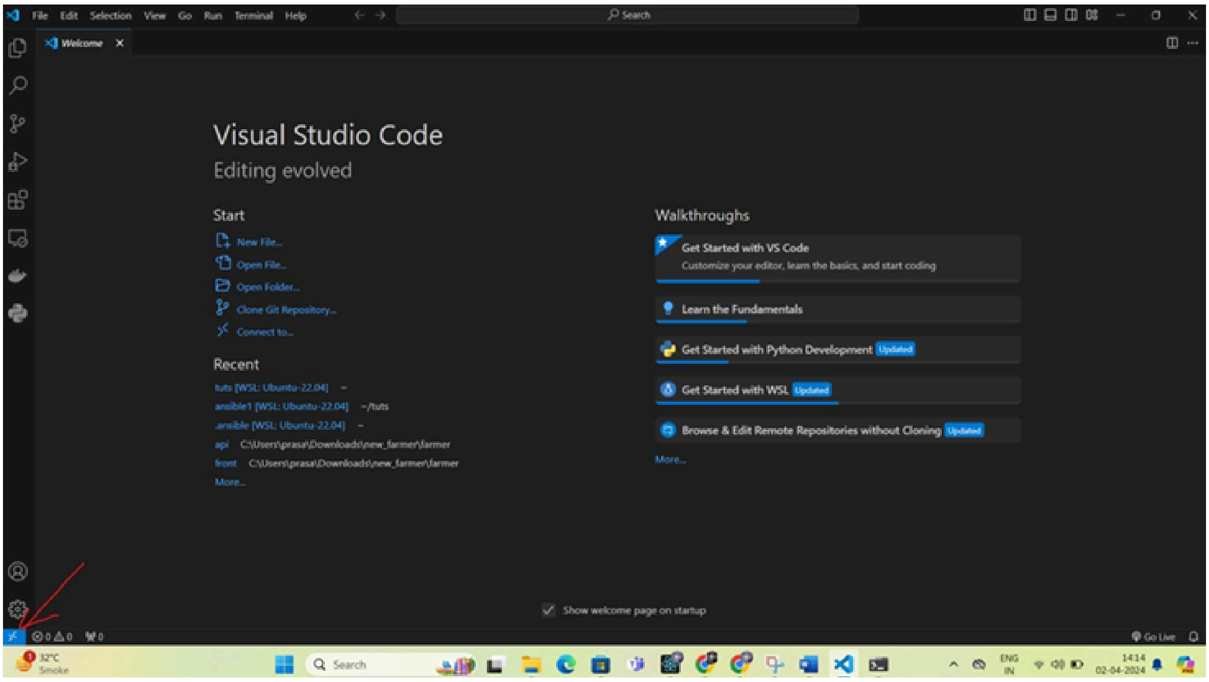
Resources

Marketplace
Issues
Repository
License
Microsoft

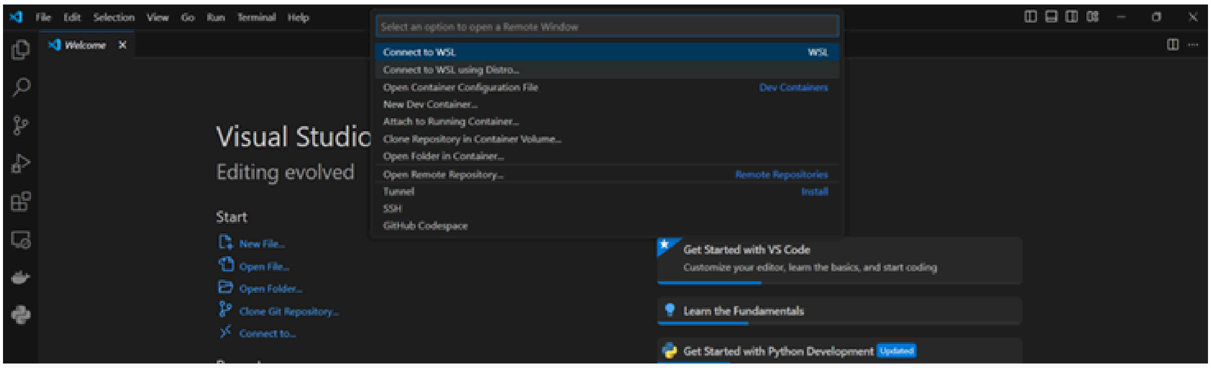
More Info

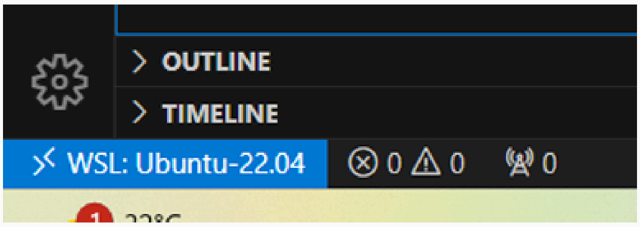
Published	2019-05-03, 00:10:03
Last released	2024-03-22, 16:31:05
Last updated	2024-03-17, 10:27:17
Identifier	ms-vscode-remote.remote-wsl

Step 9 : Connect to the ubuntu server through vscode.

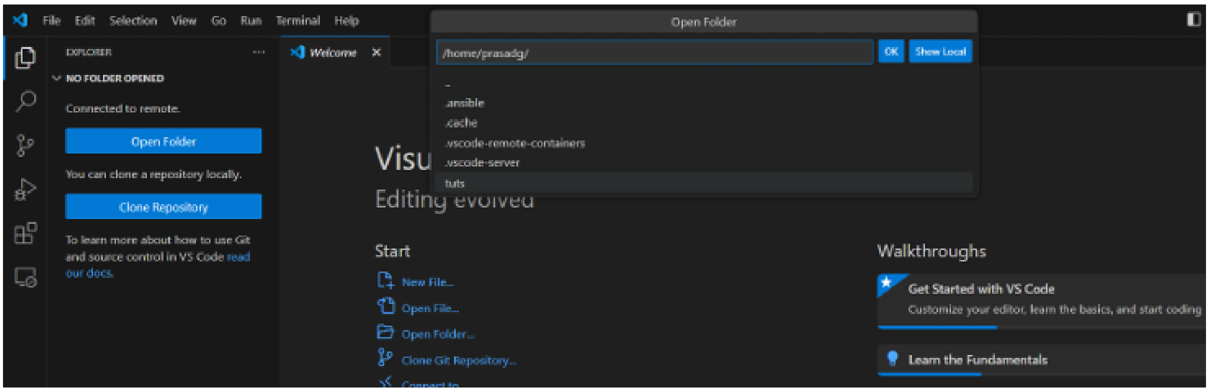

File Edit Selection View Go Run Terminal Help
Search
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Recent
tuts [WSL: Ubuntu-22.04] ~
ansible1 [WSL: Ubuntu-22.04] ~/tuts
ansible [WSL: Ubuntu-22.04] ~
api C:\Users\prasa\Downloads\new_farmer\farmer
front C:\Users\prasa\Downloads\new_farmer\farmer
More...
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
Get Started with Python Development Updated
Get Started with WSL Updated
Browse & Edit Remote Repositories without Cloning Updated
More...
Show welcome page on startup
Go Live
32°C Smoke
Search
ENG IN
14:14 02-04-2024

Click on “Connect to WSL using Distro” then select the name of your ubuntu server.


File Edit Selection View Go Run Terminal Help
Welcome
Select an option to open a Remote Window
Connect to WSL WSL
Connect to WSL using Distro...
Open Container Configuration File Dev Containers
New Dev Container...
Attach to Running Container...
Clone Repository in Container Volume...
Open Folder in Container...
Open Remote Repository... Remote Repositories
Tunnel Install
SSH
GitHub Codespace
Visual Studio
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Connect to...
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
Get Started with Python Development Updated



Step 10: Once connected click on the open folder. And select the directory which we created earlier.



/home/prasadg/

Step 11: Create a folder name Inventory and create a file named “Inventory.ini”

Code:

[webservers]

web1.example.com

web2.example.com

[dbservers]

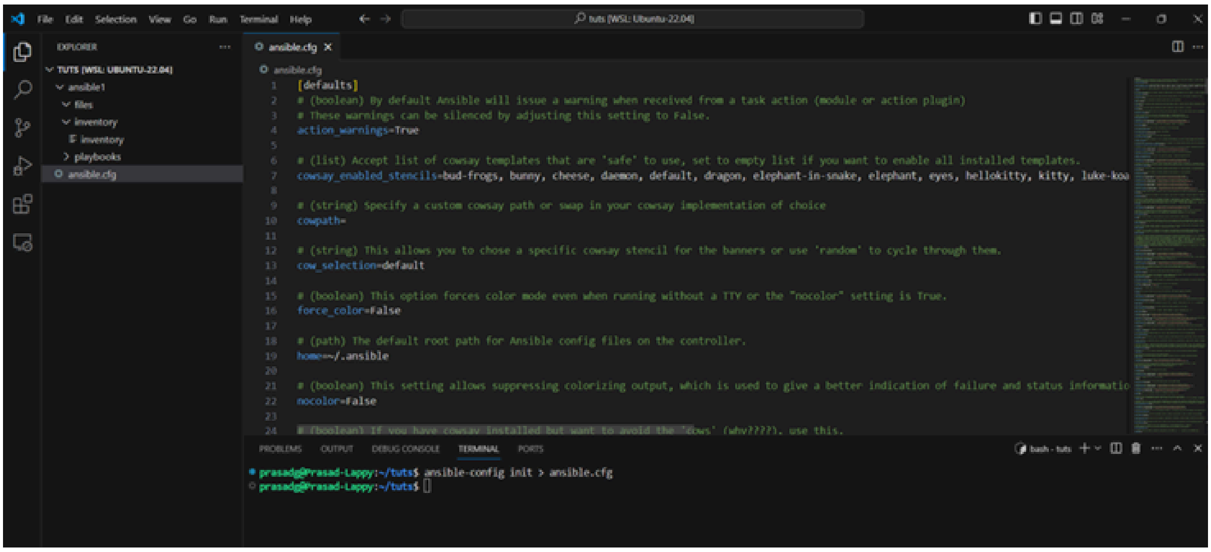
one.example.com

two.example.com

three.example.com

Step 12: Create a ansible configuration file with below command

$ ansible-config init > ansible.cfg

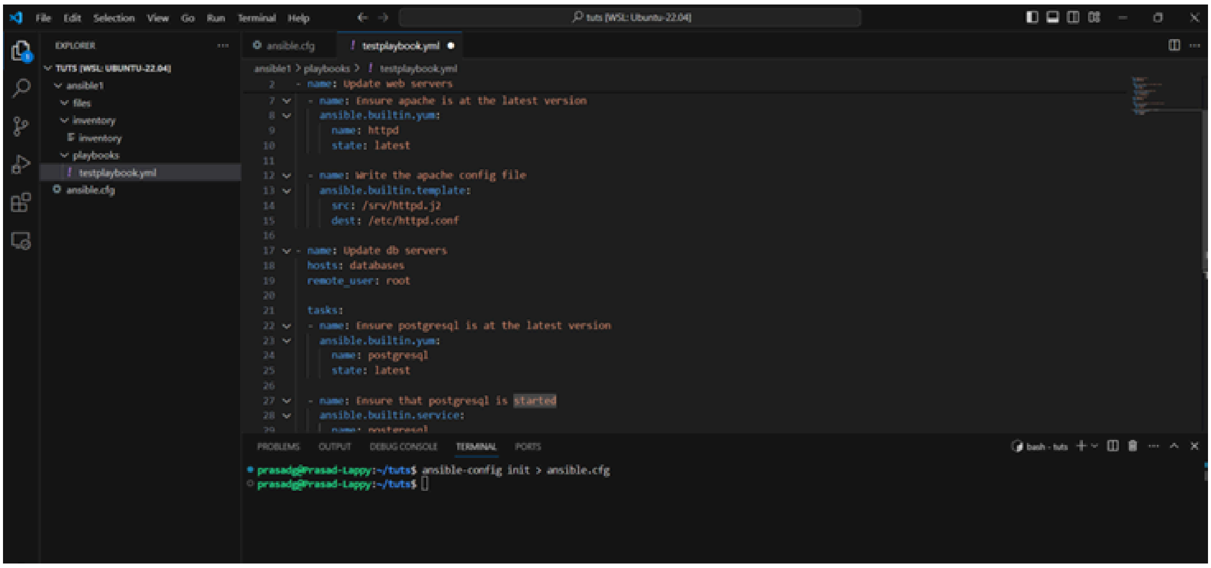


```
[defaults]
# (boolean) By default Ansible will issue a warning when received from a task action (module or action plugin)
# These warnings can be silenced by adjusting this setting to False.
action_warnings=True

# (list) Accept list of cowsay templates that are 'safe' to use, set to empty list if you want to enable all installed templates.
cowsay_enabled_stencils=bud-frogs, bunny, cheese, daemon, default, dragon, elephant-in-snake, elephant, eyes, hellokitty, kitty, luke-koa

# (string) Specify a custom cowsay path or swap in your cowsay implementation of choice
cowpath=

# (string) This allows you to chose a specific cowsay stencil for the banners or use 'random' to cycle through them.
cow_selection=default

# (boolean) This option forces color mode even when running without a TTY or the "nocolor" setting is True.
force_color=False

# (path) The default root path for Ansible config files on the controller.
home=~/.ansible

# (boolean) This setting allows suppressing colorizing output, which is used to give a better indication of failure and status informatio
nocolor=False

# (boolean) If you have cowsay installed but want to avoid the 'cows' (why????), use this.
```

```
prasadg@Prasad-Lappy:~/tuts$ ansible-config init > ansible.cfg
prasadg@Prasad-Lappy:~/tuts$
```

Step 13: Create a Ansible Playbook with .yml extension

```
--- - name: Update web servers
  hosts: webservers
  remote_user: root

  tasks:
  - name: Ensure apache is at the latest version
    ansible.builtin.yum:
      name: httpd
      state: latest

  - name: Write the apache config file
    ansible.builtin.template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
 - name: Update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: Ensure postgresql is at the latest version
    ansible.builtin.yum:
      name: postgresql
      state: latest

  - name: Ensure that postgresql is started
    ansible.builtin.service:
      name: postgresql
      state: started
```



```yaml
- name: Update web servers
  - name: Ensure apache is at the latest version
    ansible.builtin.yum:
      name: httpd
      state: latest

  - name: Write the apache config file
    ansible.builtin.template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- name: Update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: Ensure postgresql is at the latest version
    ansible.builtin.yum:
      name: postgresql
      state: latest

  - name: Ensure that postgresql is started
    ansible.builtin.service:
      name: postgresql
```

```
prasadg@Prasad-Lappy:~/tuts$ ansible-config init > ansible.cfg
prasadg@Prasad-Lappy:~/tuts$ 
```

Conclusion: In this practical we learned about the Ansible tool and how to create a playbook in it.