Atria Institute of Technology

Department of Information Science and Engineering

Big Data Analytics (18CS72)

Assignment-1

SUBMITTED BY

Name: Supriya Daspalle

USN: 1AT20IS096

Section: 7TH SEM 'B'

Submission Date: 27-11-2023

Course Handling Faculty Name:

Dr. K S Ananda Kumar
Associate Professor
Dept of ISE, Atria IT.

Table of contents

Sl. No	Description
1	1. create an **EC2 Linux** instance in AWS Cloud /Any cloud INSTANCE NAME - **YOUR NAME** INSTANCE TYPE - t2.micro/any other also. key pair name- your name storage - 10 GB Take the screenshot of instance running status Mention the private IP address and Public IP address. (Execute this program/concept and take a screenshot of the output)
2	Execute the basic Linux commands/ simple program on the instance (Execute this program and take a screenshot of the output)
3	Create the **GitHub** Account with your credentials, Same things stored in public repository in Github. Share the assignment in github link.

Note:

1. Minimum 10 Screenshots with proper explanation
2. Minimum no of pages – 10
3. Submit your Assignment soft copy (Word & PDF) to anandakumar.ks@atria.edu.
   Subject Line in mail: Student_Name_USN_BDA_Assignment1
4. Share your assignment Github link in Assignment Document.
5. Submit Assignment on or before **27th Nov 2023.**

Instance Creation-01

/ List the steps with proper explanation & Screenshots/

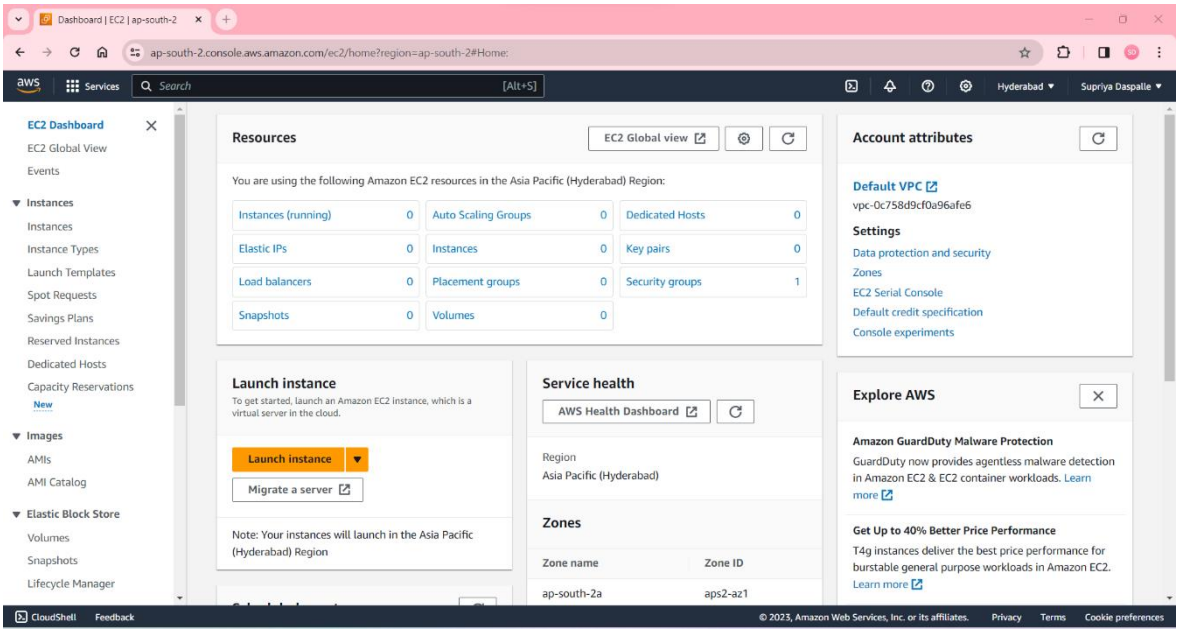
/Any number of pages/

Step 1: Sign in to the AWS Management Console

1. Go to the AWS Management Console at https://aws.amazon.com/ and sign in using your AWS account credentials.

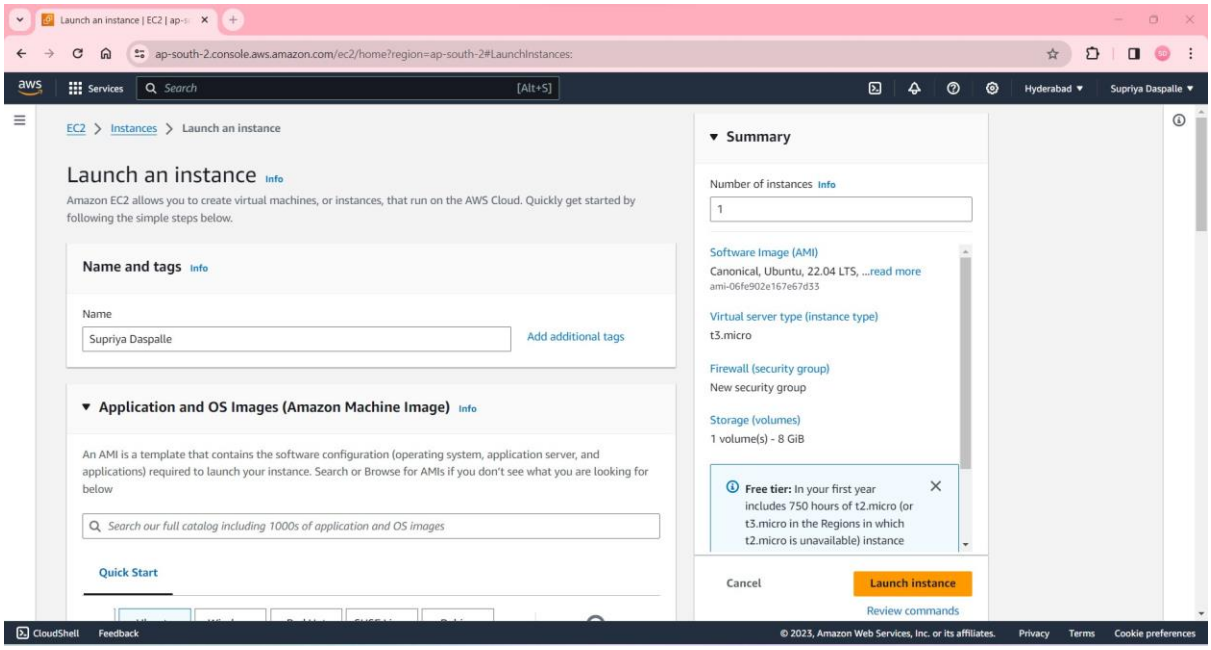
Step 2: Access EC2 Service

1. Once logged in, navigate to the "Services" dropdown at the top left corner of the page.
2. Select "EC2" under the "Compute" section. This will take you to the EC2 Dashboard.



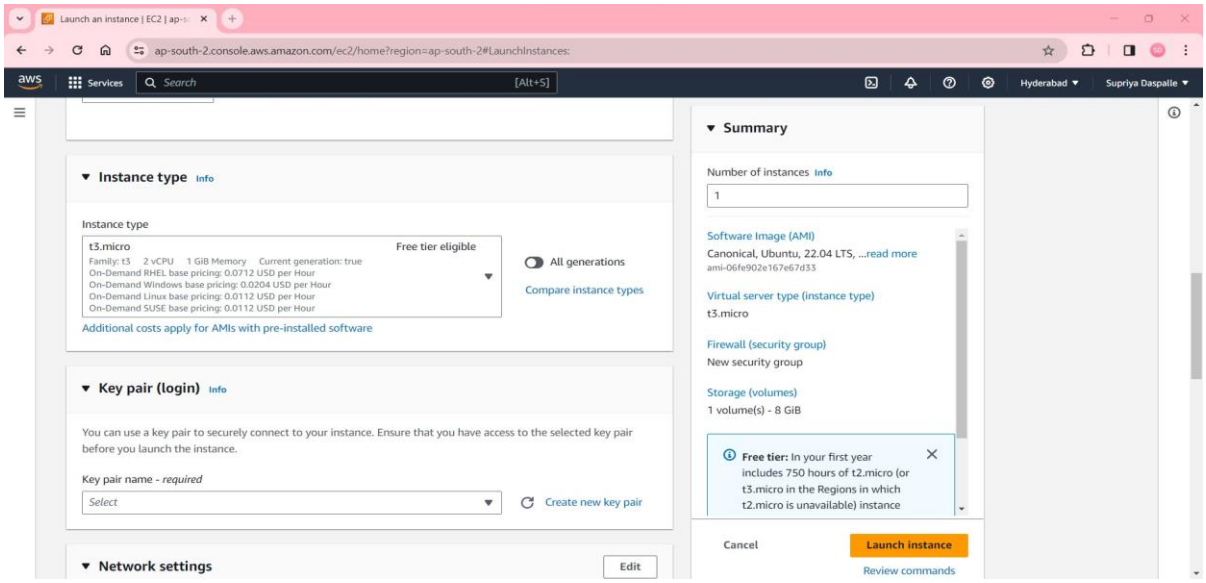
Step 3: Launch an Instance

1. On the EC2 Dashboard, click the "Launch Instance" button.
2. You'll be directed to the AWS Marketplace to choose an Amazon Machine Image (AMI). Select an AMI based on your requirements (Amazon Linux, Ubuntu, Windows, etc.). Click "Select" to proceed.



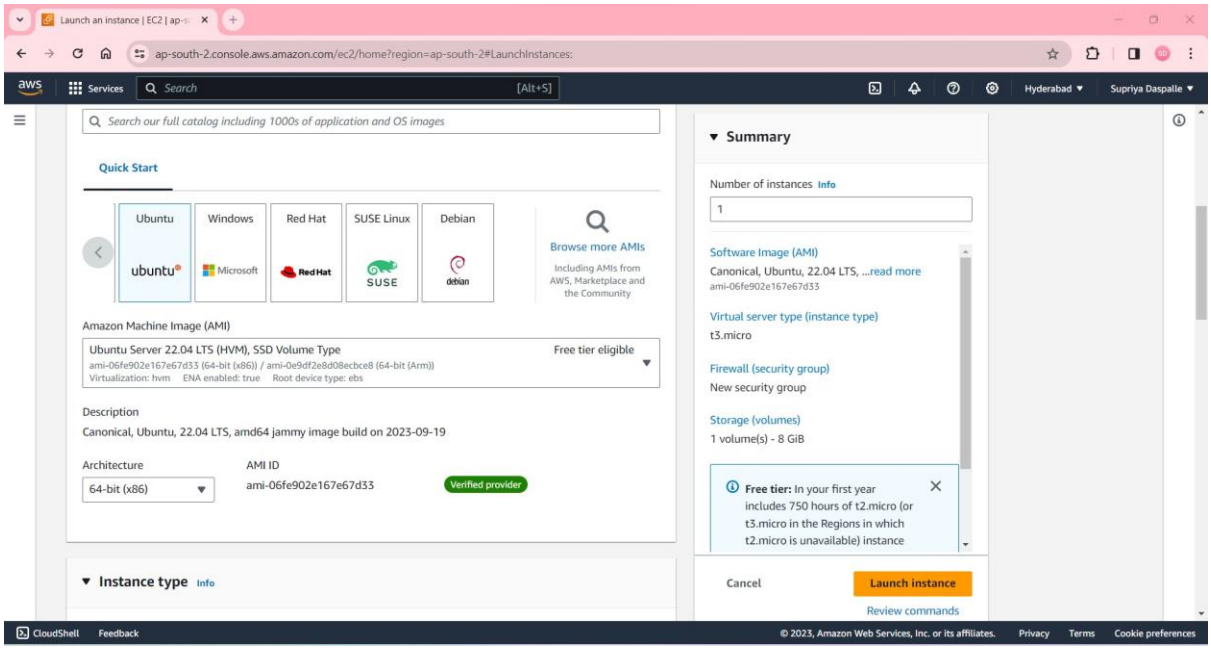
Step 4: Choose an Instance Type

1. Select an instance type based on your needs (e.g., t2.micro, t3.medium, m5.large). Each type varies in CPU, memory, storage, and network capacity. Click "Next: Configure Instance Details" once you've made your selection.



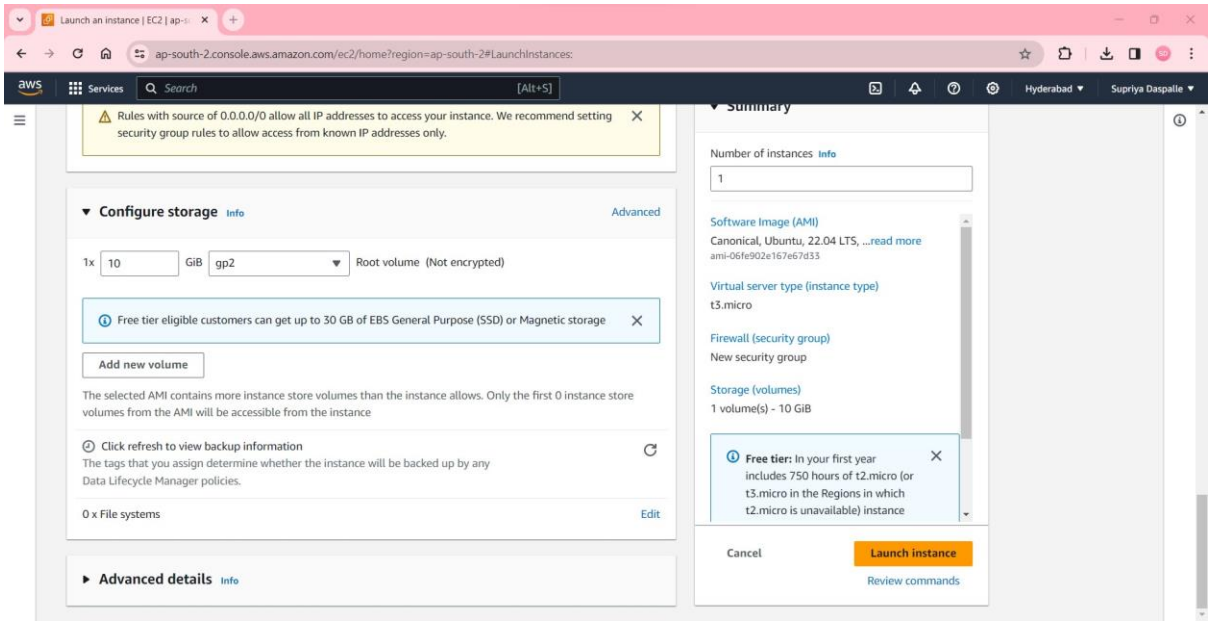
Step 5: Configure Instance Details

1. Configure settings like the number of instances, network settings, subnet, IAM role (if needed), etc. Adjust according to your requirements and click "Next: Add Storage."



Step 6: Add Storage

1. Define the storage requirements for your instance. You can add or modify storage volumes as needed. Once done, click "Next: Add Tags."



Step 7: Add Tags (Optional but Recommended)

1. Tags help you identify and manage your instances. Add key-value pairs to organize and categorize instances. Click "Next: Configure Security Group."

Step 8: Configure Security Group

1. A security group acts as a virtual firewall that controls the traffic for your instance.

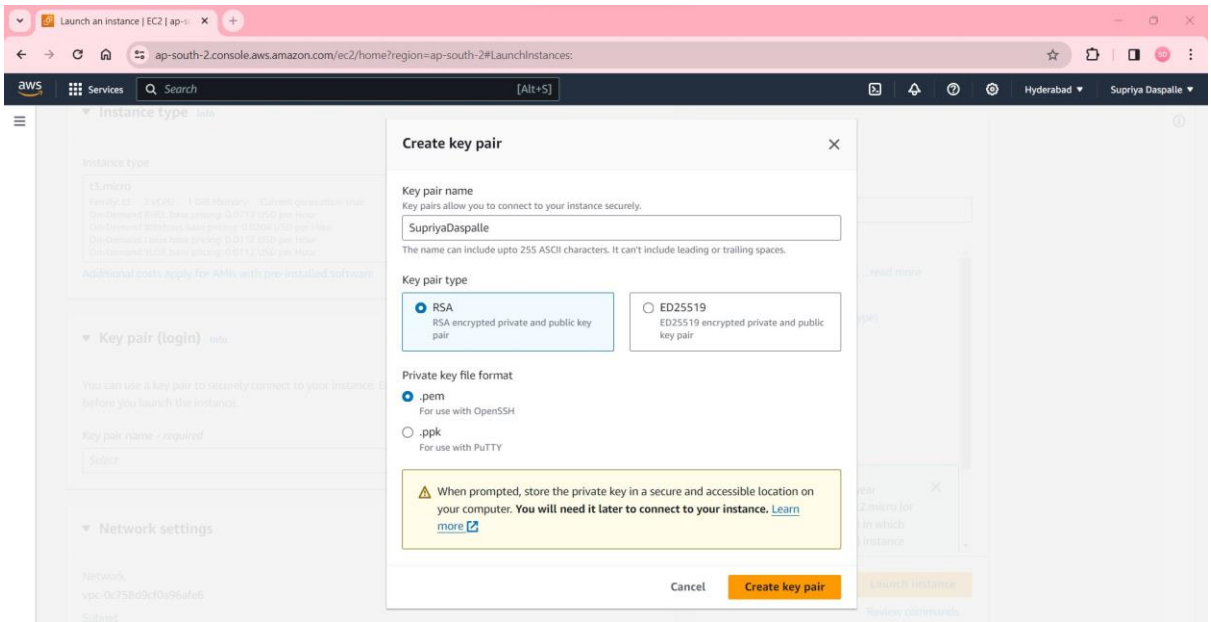
2. Create a new security group or select an existing one. Define inbound and outbound rules (e.g., SSH, HTTP, HTTPS) to control traffic. Click "Review and Launch" when finished.

Step 9: Review and Launch

1. Review the configuration details of your instance. Make sure everything is set up as desired.
2. Click "Launch" to initiate the instance creation process.

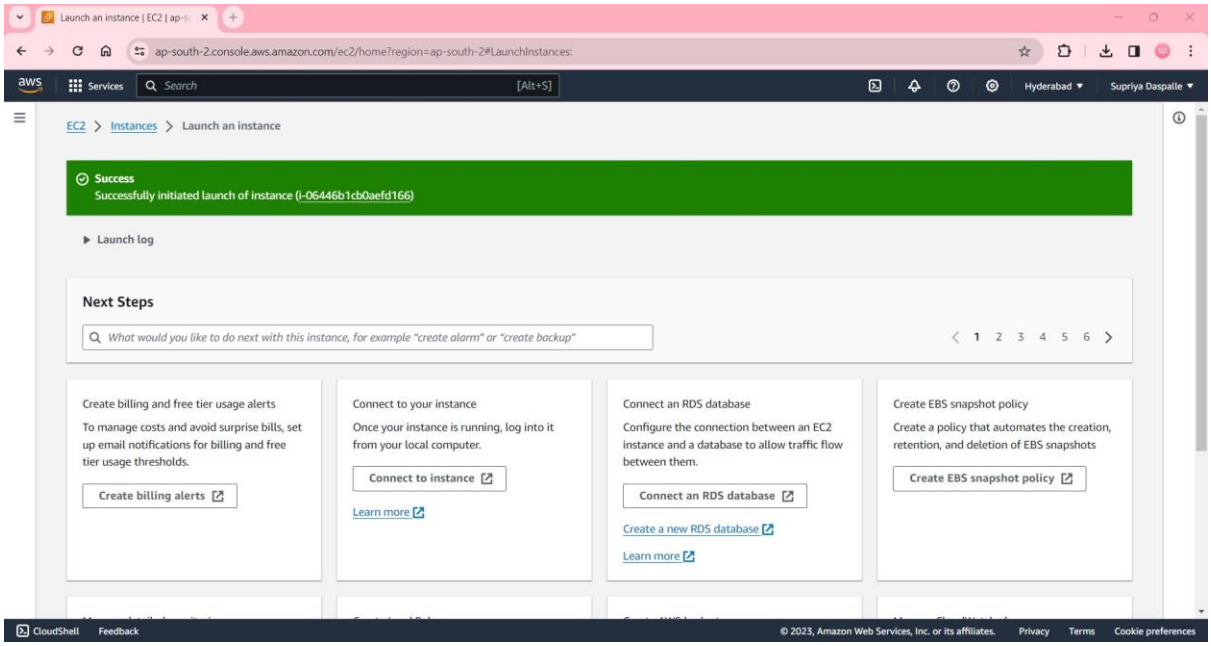
Step 10: Create a Key Pair

1. A key pair is necessary to connect securely to your instance. Select "Create a new key pair" from the dropdown menu.
2. Enter a name for your key pair and download the .pem file. **IMPORTANT**: Keep this file secure as it's required to access your instance.



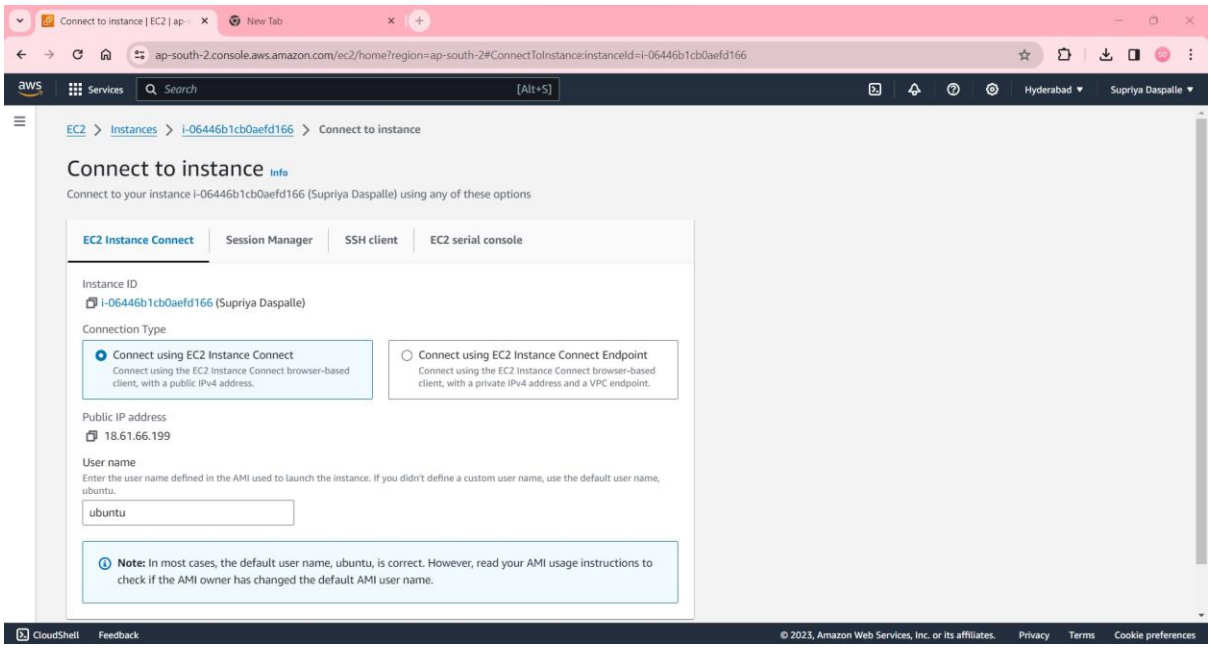
Step 11: Launch the Instance

1. Click "Launch Instances." AWS will start provisioning your instance. You'll be redirected to the Instances section of the EC2 Dashboard.
2. Monitor the instance's status. Once it shows "running," your instance is ready.



Step 12: Access Your Instance

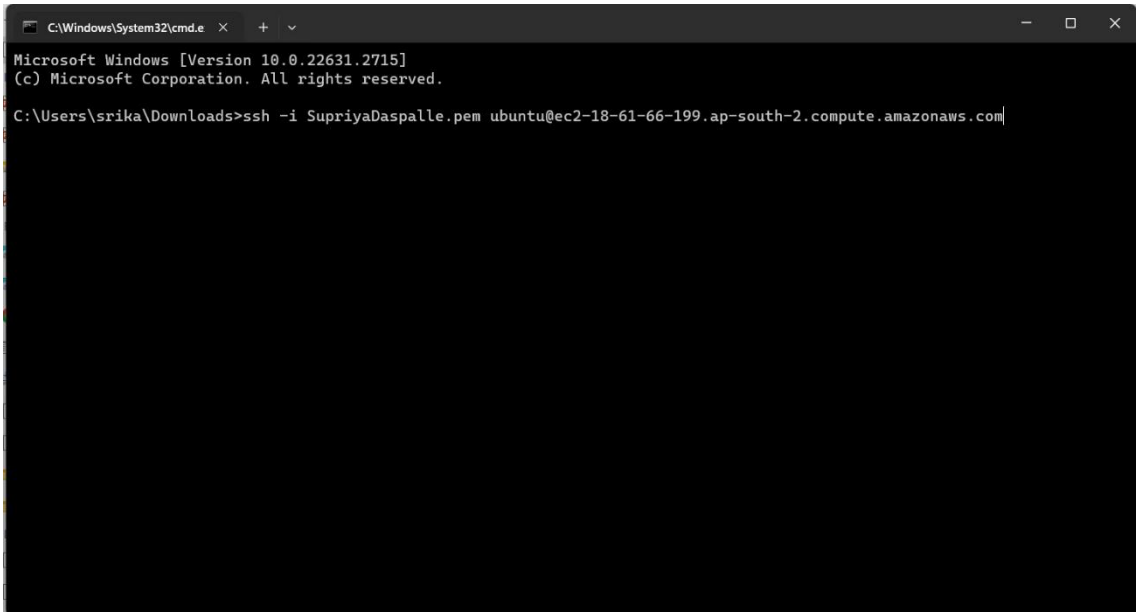
1. Use an SSH client (e.g., PuTTY for Windows, Terminal for macOS/Linux) to connect to your instance using the downloaded .pem key pair file and the public DNS or IP address provided by AWS.



Execute the basic Linux commands

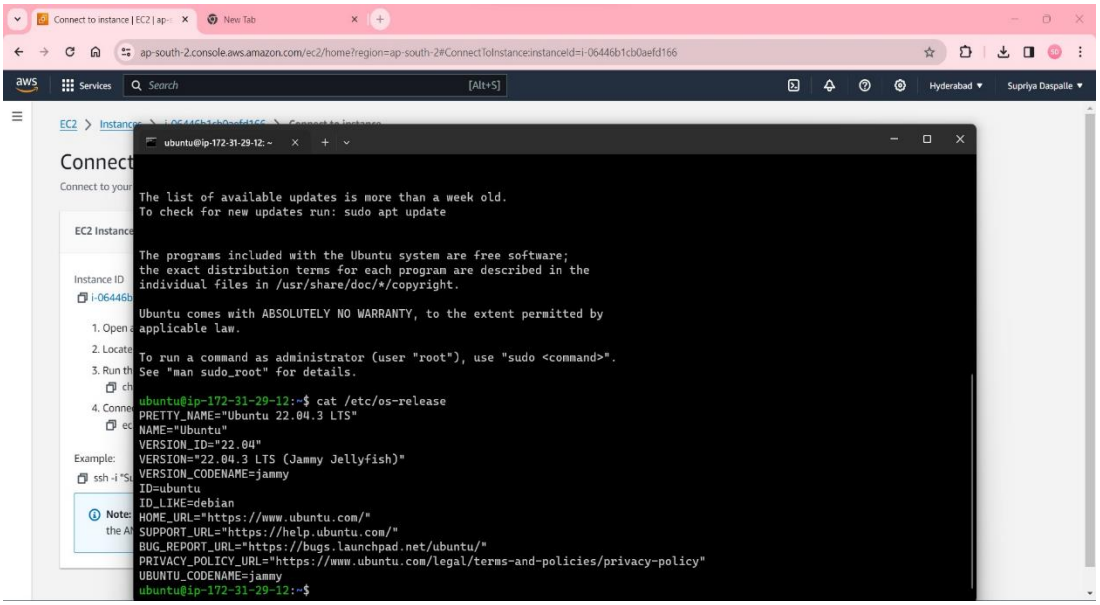
/ List the steps with proper explanation & Screenshots/

/Any number of pages/

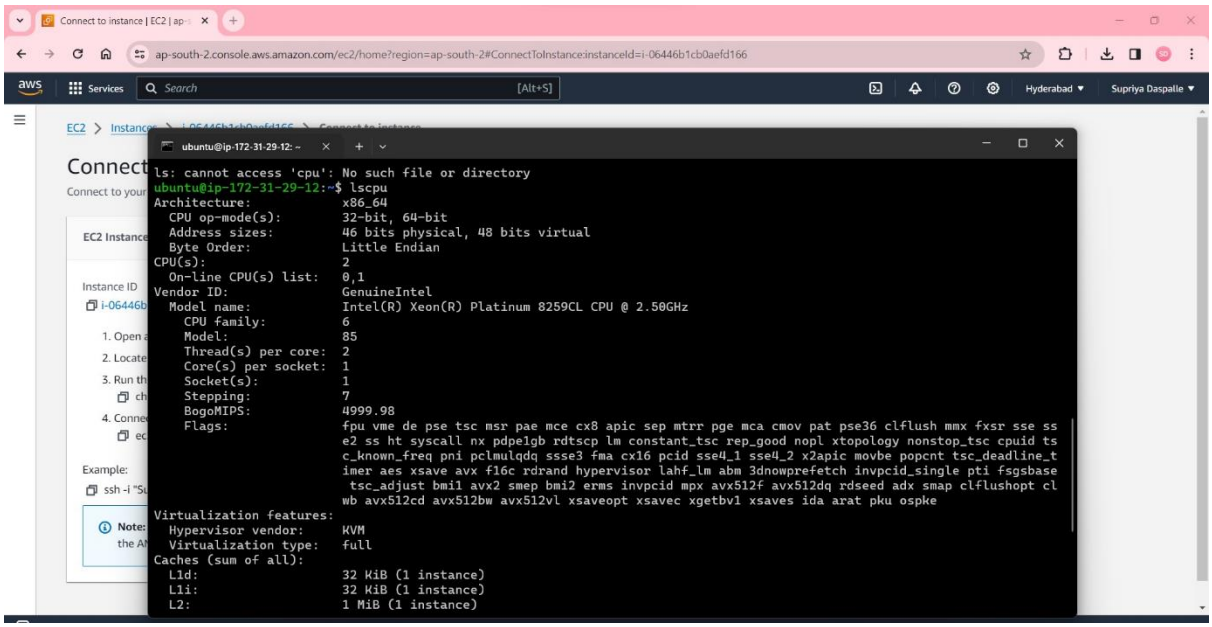


The command **cat /etc/os-release** is used in Linux to display information about the operating system (OS) release or distribution.

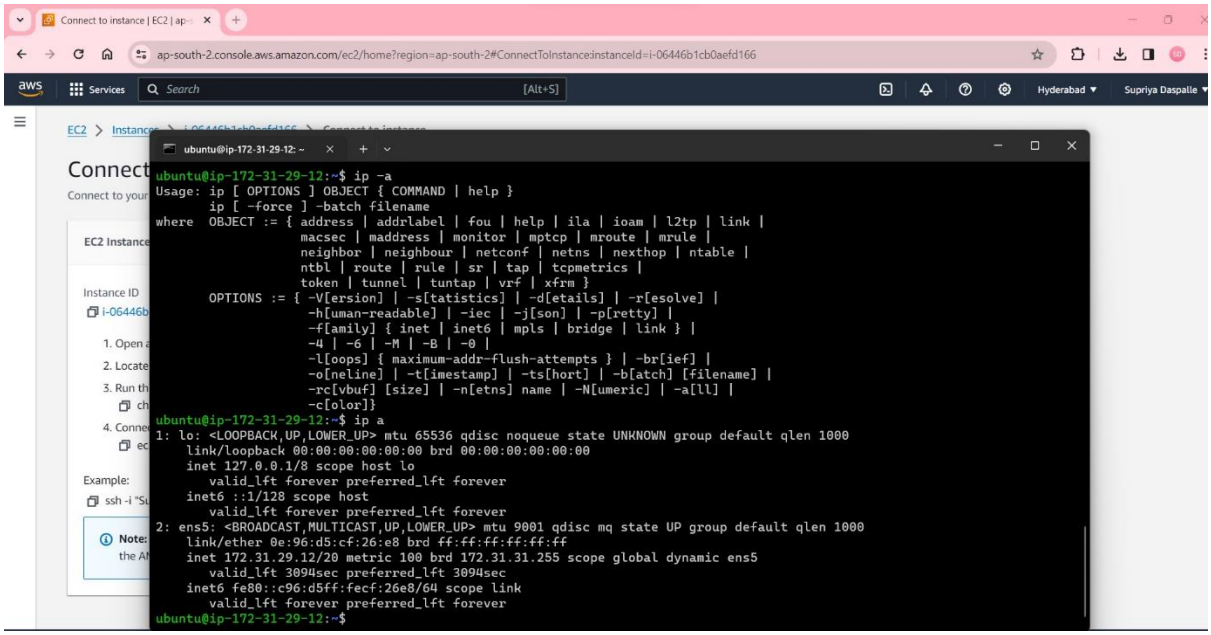
1. **cat**: This command is used to concatenate and display the content of files. When used with **/etc/os-release**, it displays the content of the file **os-release**.
2. **/etc/os-release**: This file contains specific information about the operating system release and distribution. It's commonly found on Linux distributions that adhere to the systemd system and service manager.

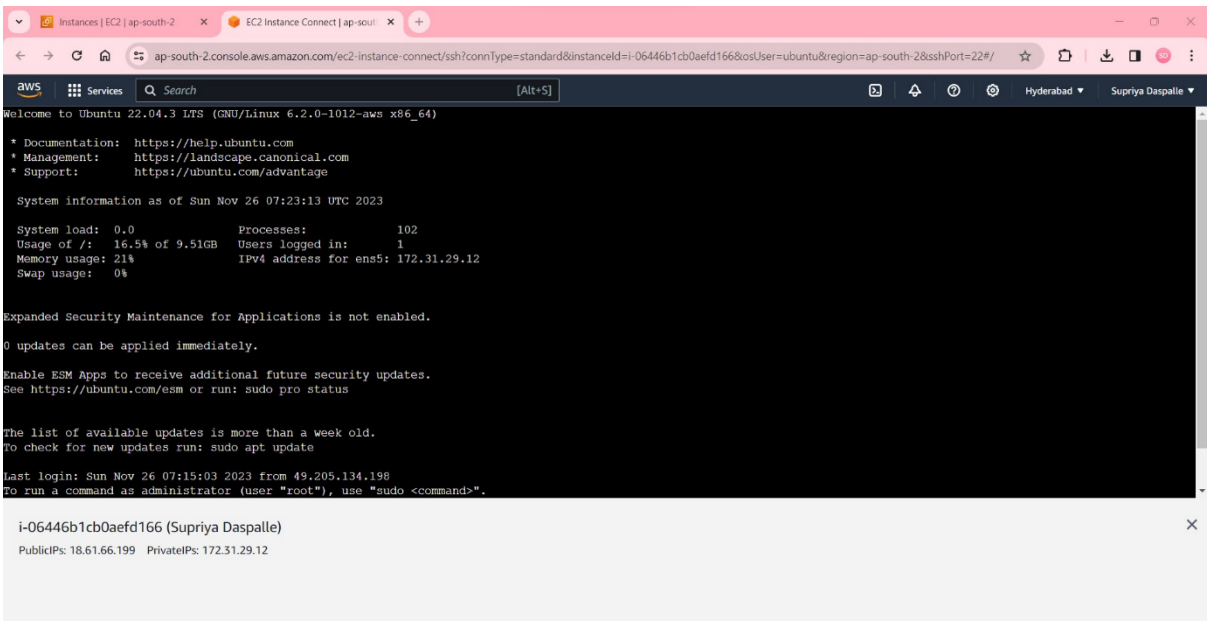


The **lscpu** command in Linux is used to display detailed information about the CPU (Central Processing Unit) and its characteristics. It provides comprehensive information about the processor(s) installed on the system.



The **ipa** command is primarily associated with Identity Management in Linux, particularly in Red Hat-based systems. It stands for "Identity, Policy, and Audit," and it is a command-line interface for managing Identity Management (IdM) services.


Connect to instance | EC2 | ap-
ap-south-2.console.aws.amazon.com/ec2/home?region=ap-south-2#ConnectToInstance:instanceId=i-06446b1cb0aefd166
Services
Search
[Alt+S]
Hyderabad
Supriya Daspalle
EC2 > Instances
Connect
EC2 Instance
Instance ID
i-06446b
1. Open
2. Locate
3. Run th
4. Conne
Example:
ssh -i "St
Note:
the AN
ubuntu@ip-172-31-29-12: ~
ubuntu@ip-172-31-29-12:~$ ip -a
Usage: ip [OPTIONS] OBJECT { COMMAND | help }
ip [-force] -batch filename
where OBJECT := { address | addrlabel | fou | help | ila | ioam | l2tp | link |
macsec | maddress | monitor | mptcp | mroute | mrule |
neighbor | neighbour | netconf | netns | nexthop | ntable |
ntbl | route | rule | sr | tap | tcpmetrics |
token | tunnel | tuntap | vrf | xfrm }
OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
-h[uman-readable] | -iec | -j[son] | -p[retty] |
-f[amily] { inet | inet6 | mpls | bridge | link } |
-4 | -6 | -M | -B | -0 |
-l[oops] { maximum-addr-flush-attempts } | -br[ief] |
-o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
-rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
-c[olor]}
ubuntu@ip-172-31-29-12:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
link/ether 0e:96:d5:cf:26:e8 brd ff:ff:ff:ff:ff:ff
inet 172.31.29.12/20 metric 100 brd 172.31.31.255 scope global dynamic ens5
valid_lft 3094sec preferred_lft 3094sec
inet6 fe80::c96:d5ff:fecf:26e8/64 scope link
valid_lft forever preferred_lft forever
ubuntu@ip-172-31-29-12:~$


Instances | EC2 | ap-south-2
EC2 Instance Connect | ap-sout
ap-south-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-06446b1cb0aefd166&osUser=ubuntu&region=ap-south-2&sshPort=22#/
Services
Search
[Alt+S]
Hyderabad
Supriya Daspalle
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
System information as of Sun Nov 26 07:23:13 UTC 2023
System load: 0.0 Processes: 102
Usage of /: 16.5% of 9.51GB Users logged in: 1
Memory usage: 21% IPv4 address for ens5: 172.31.29.12
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Sun Nov 26 07:15:03 2023 from 49.205.134.198
To run a command as administrator (user "root"), use "sudo <command>".
i-06446b1cb0aefd166 (Supriya Daspalle)
PublicIPs: 18.61.66.199 PrivateIPs: 172.31.29.12

Running sample Program on Linux Instance

Step 1: Launch an EC2 Instance

1. **Sign in to AWS Management Console:**
   - Go to the AWS Management Console.
   - Sign in using your credentials.

2. **Navigate to EC2:**
   - Click on the "Services" dropdown.
   - Select "EC2" under the "Compute" section.

3. **Launch an Instance:**
   - Click on "Instances" in the EC2 Dashboard.
   - Click the "Launch Instance" button.

4. **Choose an Amazon Machine Image (AMI):**
   - Select an appropriate AMI (e.g., Amazon Linux, Ubuntu) based on your requirements.

5. **Choose an Instance Type:**
   - Select the instance type based on your computing needs.

6. **Configure Instance:**
   - Configure instance details like network, subnet, etc.

7. **Add Storage:**
   - Configure the storage requirements for the instance.

8. **Add Tags (Optional):**
   - Optionally, add tags for better organization and identification.

9. **Configure Security Group:**
   - Create or select a security group to define inbound/outbound rules (allowing SSH access, etc.).

10. **Review and Launch:**
    - Review the configuration settings and launch the instance.
    - Choose an existing key pair or create a new one to access the instance securely.

Step 2: Connect to the EC2 Instance

SSH into the Instance:

- Use an SSH client like Terminal (Mac/Linux) or PuTTY (Windows) to connect to the instance.
- Example command: **ssh -i /path/to/your-key.pem ec2-user@your-instance-public-ip**

Step 3: Setup Python Environment

1. **Check Python Installation:**
    - By default, most EC2 instances come with Python pre-installed.
    - Check the installed version: **python --version** or **python3 –version**

2. **Install or Update Python (if required):**
    - Use package manager like **yum** (Amazon Linux) or **apt** (Ubuntu) to install or update Python if needed.
        - For example: **sudo yum install python3** or **sudo apt-get install python3**

3. **Install Additional Packages (if needed):**
    - Use **pip** to install any additional Python packages required for your script.
        - Example: **pip install package_name**

Step 4: Upload and Run Python Code

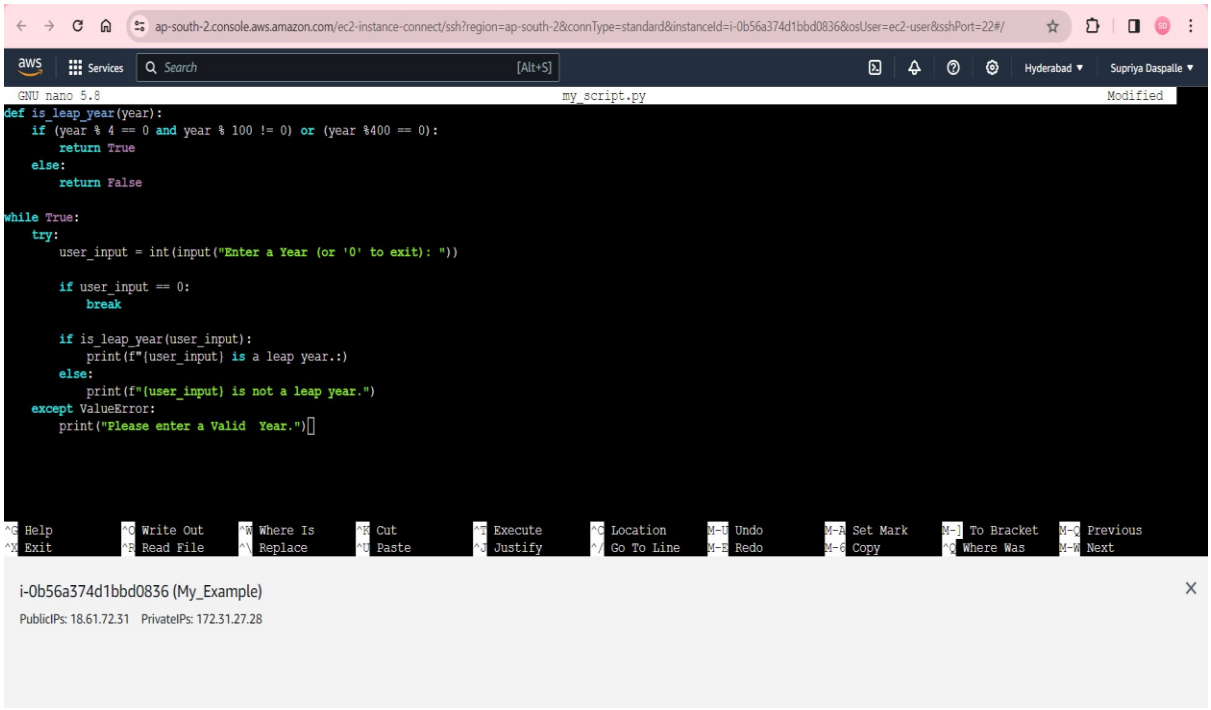
1. **Upload Python Script:**
    - Use SCP (Secure Copy Protocol), SFTP (Secure File Transfer Protocol), or tools like **scp** or **rsync** to transfer your Python script to the EC2 instance.

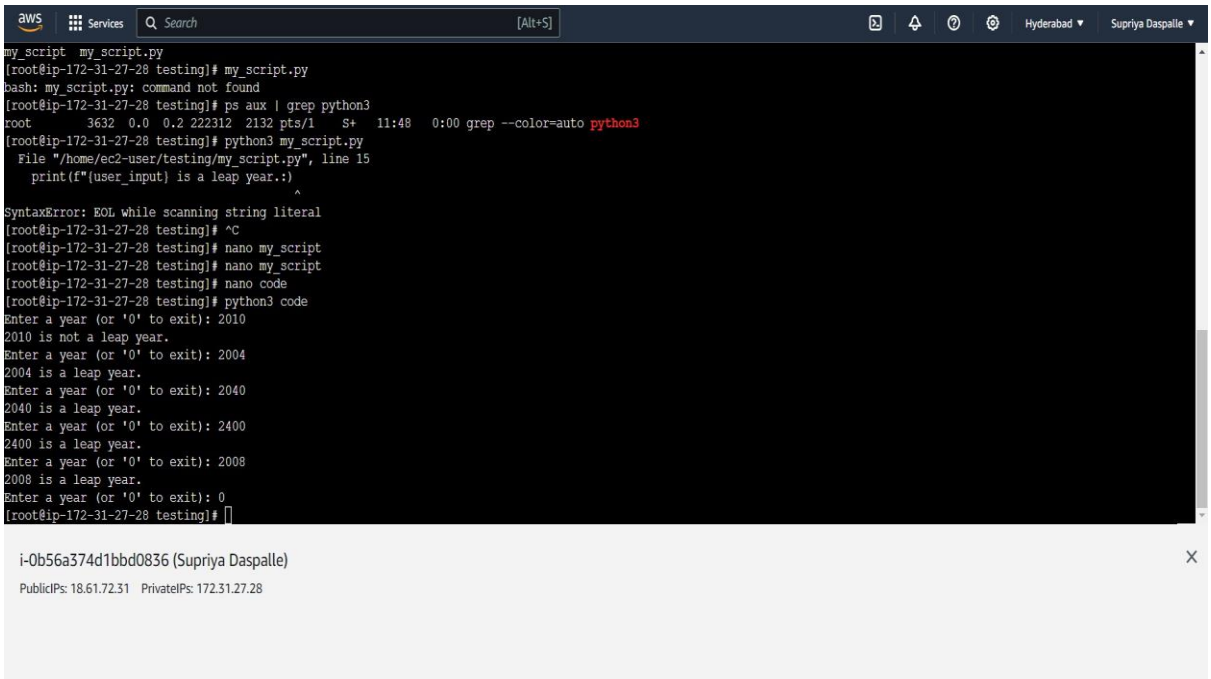
2. **Run Python Script:**
    - Navigate to the directory where your Python script is located.
    - Run the script using the Python interpreter: **python script_name.py** or **python3 script_name.py**

Step 5: Monitoring and Management

1. **Monitoring and Logs:**
    - Monitor your EC2 instance through the AWS Management Console.
    - Use CloudWatch for logging and monitoring system performance.

2. **Instance Management:**
    - Terminate or stop the instance when it's no longer needed to avoid unnecessary charges.


ap-south-2.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-2&connType=standard&instanceId=i-0b56a374d1bbd0836&osUser=ec2-user&sshPort=22#/
Services
Search
[Alt+S]
Hyderabad
Supriya Daspalle
GNU nano 5.8
my_script.py
Modified
def is_leap_year(year):
if (year % 4 == 0 and year % 100 != 0) or (year %400 == 0):
return True
else:
return False
while True:
try:
user_input = int(input("Enter a Year (or '0' to exit): "))
if user_input == 0:
break
if is_leap_year(user_input):
print(f"{user_input} is a leap year.:)
else:
print(f"{user_input} is not a leap year.")
except ValueError:
print("Please enter a Valid Year.")
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next
i-0b56a374d1bbd0836 (My_Example)
PublicIPs: 18.61.72.31 PrivateIPs: 172.31.27.28


Services
Search
[Alt+S]
Hyderabad
Supriya Daspalle
my_script my_script.py
[root@ip-172-31-27-28 testing]# my_script.py
bash: my_script.py: command not found
[root@ip-172-31-27-28 testing]# ps aux | grep python3
root 3632 0.0 0.2 222312 2132 pts/1 S+ 11:48 0:00 grep --color=auto python3
[root@ip-172-31-27-28 testing]# python3 my_script.py
File "/home/ec2-user/testing/my_script.py", line 15
print(f"{user_input} is a leap year.:)
^
SyntaxError: EOL while scanning string literal
[root@ip-172-31-27-28 testing]# ^C
[root@ip-172-31-27-28 testing]# nano my_script
[root@ip-172-31-27-28 testing]# nano my_script
[root@ip-172-31-27-28 testing]# nano code
[root@ip-172-31-27-28 testing]# python3 code
Enter a year (or '0' to exit): 2010
2010 is not a leap year.
Enter a year (or '0' to exit): 2004
2004 is a leap year.
Enter a year (or '0' to exit): 2040
2040 is a leap year.
Enter a year (or '0' to exit): 2400
2400 is a leap year.
Enter a year (or '0' to exit): 2008
2008 is a leap year.
Enter a year (or '0' to exit): 0
[root@ip-172-31-27-28 testing]#
i-0b56a374d1bbd0836 (Supriya Daspalle)
PublicIPs: 18.61.72.31 PrivateIPs: 172.31.27.28

Assignment GitHub Link (https://github.com/login). (using this link able to access your work)

https://github.com/SupriyaDaspalle/Supriya-Daspalle