UPES
UNIVERSITY OF THE FUTURE

University of Petroleum and Energy Studies

System Provisioning and Configuration Management Lab File

Name: Aanchal Tailwal **Faculty: Mrs. Silky Goel**

Branch: BTech CSE

Batch: DevOps B-4 (NH)

Sap Id: 500097386

Roll No: R2142211334

INDEX

EXPERIMENT-1

AIM : Terraform Installation

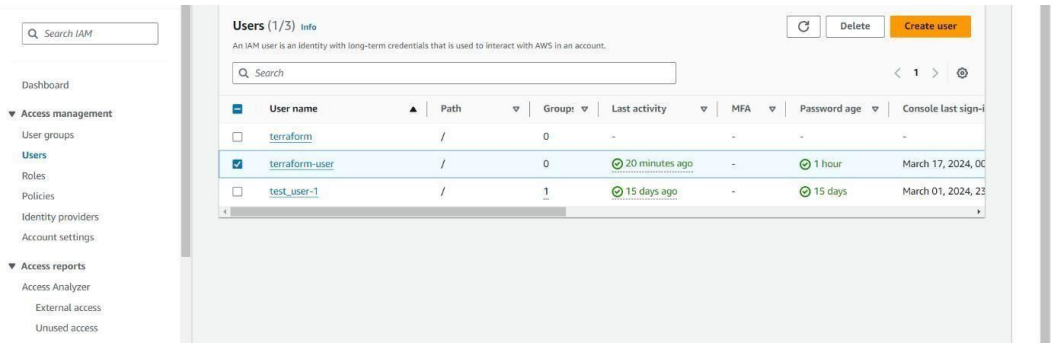
Step 1: go to https://www.terraform.io/downloads.html
Step 2: Scroll down a bit and you can see Windows. There will be 32-bit and 64-bit. As per your
processor, click on the link and the zip file will start getting downloaded. It will be downloaded in your Downloads folder.
Step 3: Unzip the zip file.

Step 4: Open a command prompt and write the below commands: mkdir terraform cd terraform/
Step 5: The zipped file which has been unzipped in Step 3, copy that file and paste it in the folder directory created in step 4.
Step 6: In the command prompt window where you entered the commands mentioned in Step 4 just write the below command set PATH=%PATH%;C:\Users\Asus\terraform.
Step 7: Now write in the command prompt — terraform. Below is the screenshot of what it looks like
in the machine.

Terraform 0.13 and later:

Copy

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-east-1"
}

# Create a VPC
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}
```



EXPERIMENT-2

Aim:

- **Download aws provider through terraform.**
- **Provide the aws api access to terraform.**
- **Create an EC2 instance.**
- **Create an EC2 instance and add a tag.**

Created IAM Admin User — terraform-user

Then Created AWS API Access key for IAM User — terraform-user

```
resource "aws_instance" "spcm-lab1" {
  ami = "ami-0d3f444bc76de0a79"
  instance_type = "t2.micro"
  tags = {"Name":"terraformlab1"}
}
```

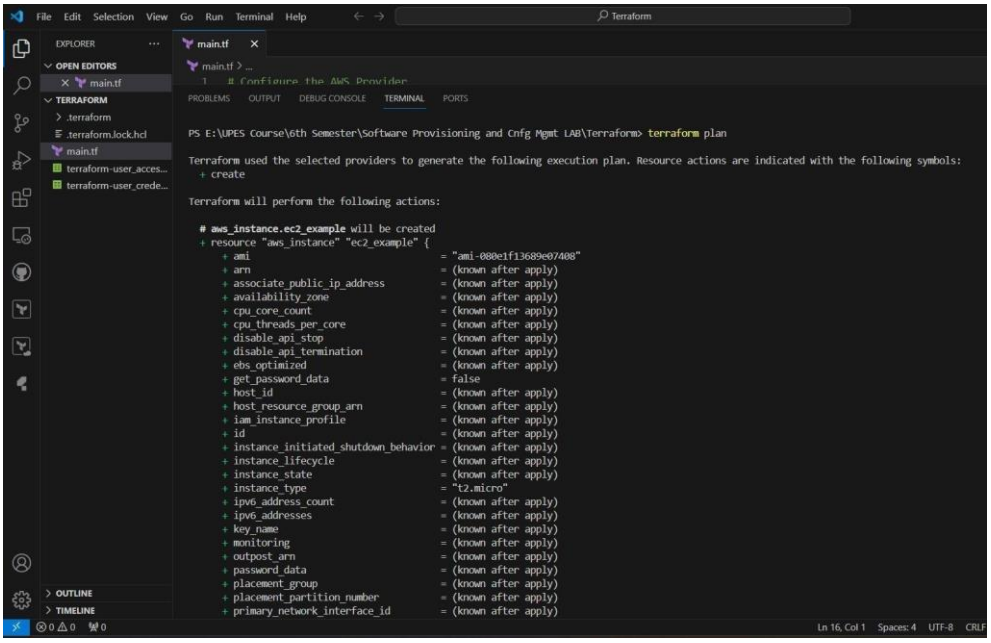
Terraform commands

1) **terraform init:** This command is responsible for downloading all the dependencies which are required for the provider AWS on your local machine.

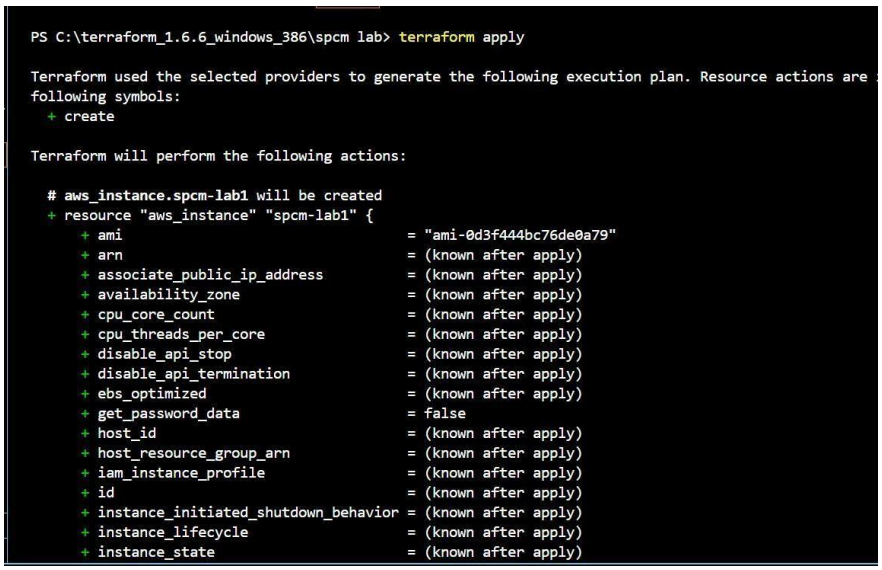
```
main.tf
main.tf > ...
# Configure the AWS Provider
provider "aws" {
    region = "us-east-1"
    access_key = "AKIAR7TVWYL33L2QMRV6"
    secret_key = "9glt71SIV1Vt33FrnqHFDqwlI7UraIAj+j7ZDwcp"
}

# Creating AWS EC2 Instance
resource "aws_instance" "ec2_example" {
    ami = "ami-080e1f13689e07408"
    instance_type = "t2.micro"
    tags = {
        Name = "Terraform EC2"
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\UPES Course\6th Semester\Software Provisioning and Cnfg Mgmt LAB\Terraform> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.41.0...
- Installed hashicorp/aws v5.41.0 (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
```

2) **terraform plan:** This command will help you to understand how many resources you are gonna add or delete.

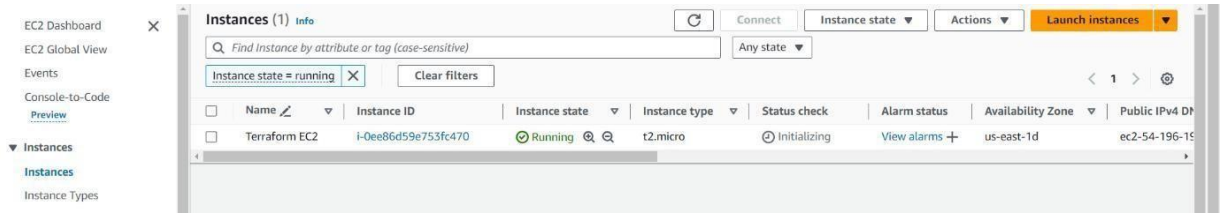


3) **terraform apply:** This command will do some real stuff on AWS. Once you issue this command, it will be going to connect to AWS and then finally going to provision AWS instances.

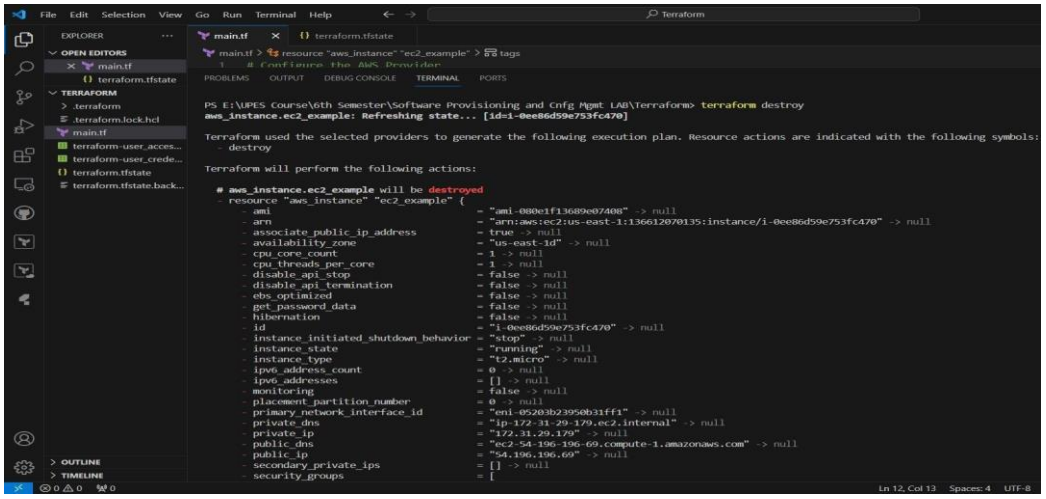


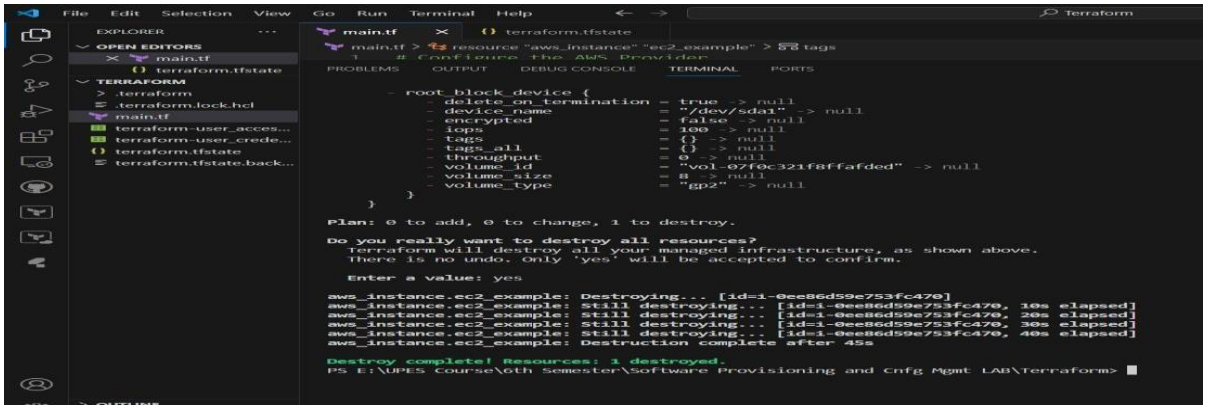
```
PS C:\terraform_1.6.6_windows_386\spcm lab> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are i
following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.spcm-lab1 will be created
  + resource "aws_instance" "spcm-lab1" {
      + ami                                  = "ami-0d3f444bc76de0a79"
      + 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)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
```

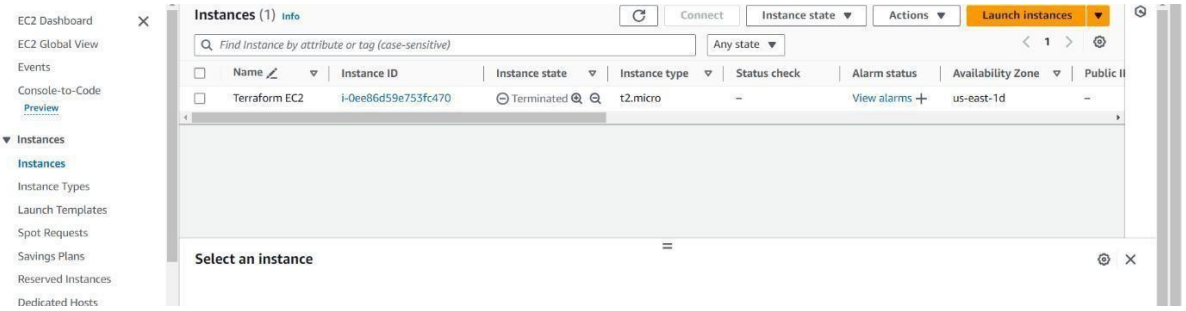
```
        }
      + tags_all                                = {
          + "Name" = "terraformlab1"
        }
      + tenancy                                 = (known after apply)
      + user_data                               = (known after apply)
      + user_data_base64                        = (known after apply)
      + user_data_replace_on_change             = false
      + vpc_security_group_ids                  = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.spcm-lab1: Creating...
aws_instance.spcm-lab1: Still creating... [10s elapsed]
aws_instance.spcm-lab1: Still creating... [20s elapsed]
aws_instance.spcm-lab1: Still creating... [30s elapsed]
aws_instance.spcm-lab1: Creation complete after 34s [id=i-0240f85e17f4c4a43]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



4) **terraform destroy:** It will remove all the running EC2 Instances.






EC2 Dashboard
EC2 Global View
Events
Console-to-Code
Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Terraform EC2
i-0ee86d59e753fc470
Terminated
t2.micro
View alarms
us-east-1d
Select an instance

Experiment- 3

Aim:

Write the terraform script to perform the following task:

- **create a S3 bucket**
- **enable the version of the bucket**
- **add an upload folder**

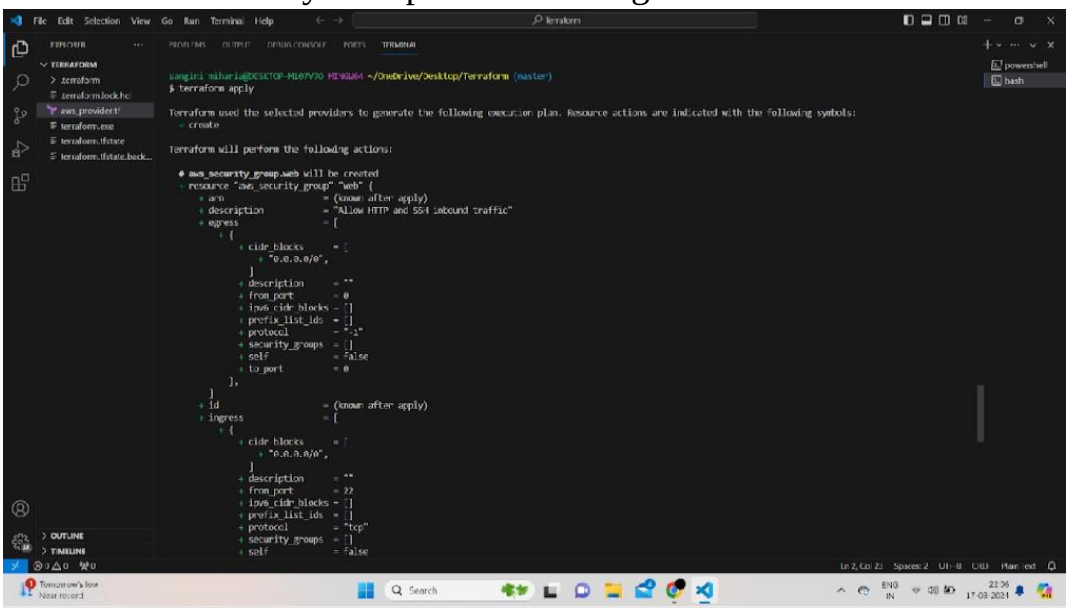
Terraform script:

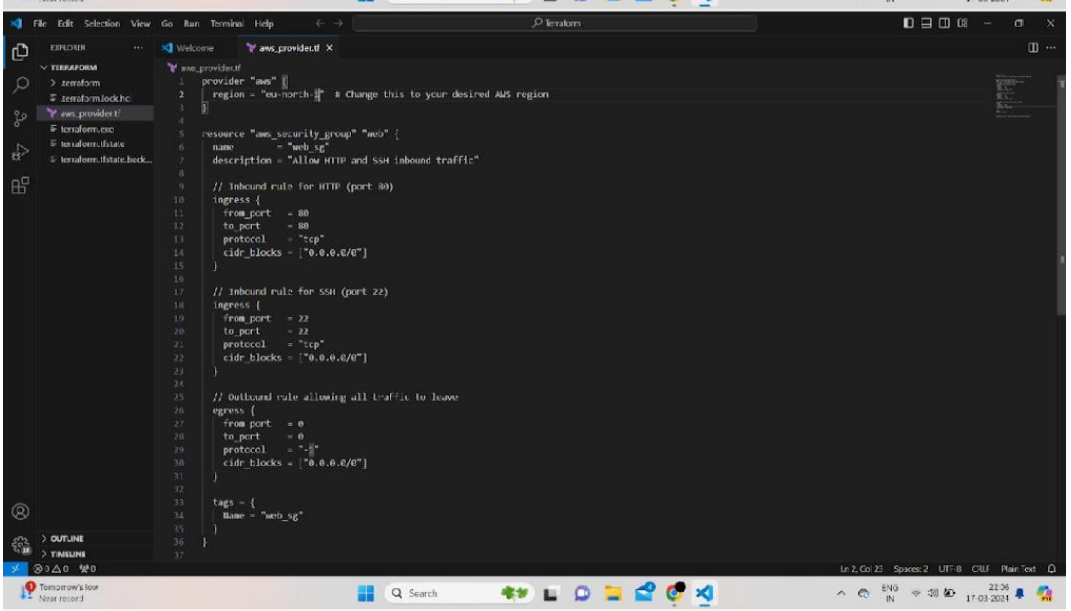
```
provider "aws" {
  access_key = "AKIAVURU2DW5CV6PQ7EV"
  secret_key = "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"
  region = "ap-south-1"

}
resource "aws_s3_bucket" "lab_5_sanskar" {
  bucket="lab-5-sanskar"

  tags = {
    "Name": "Lab5-sanskar"
  }
}
```

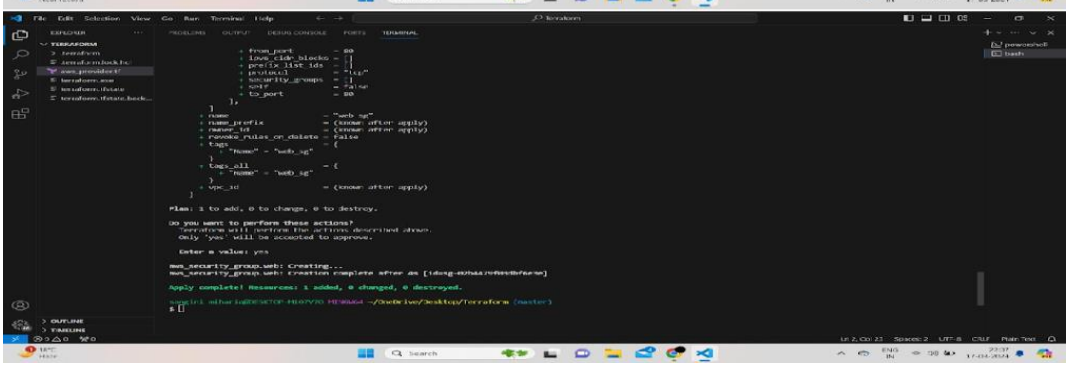
```
resource "aws_s3_bucket_versioning" "lab_5_versioning" {
  bucket = aws_s3_bucket.lab_5_sanskar.id

  versioning_configuration {
    status = "Enabled"
  }

}
```

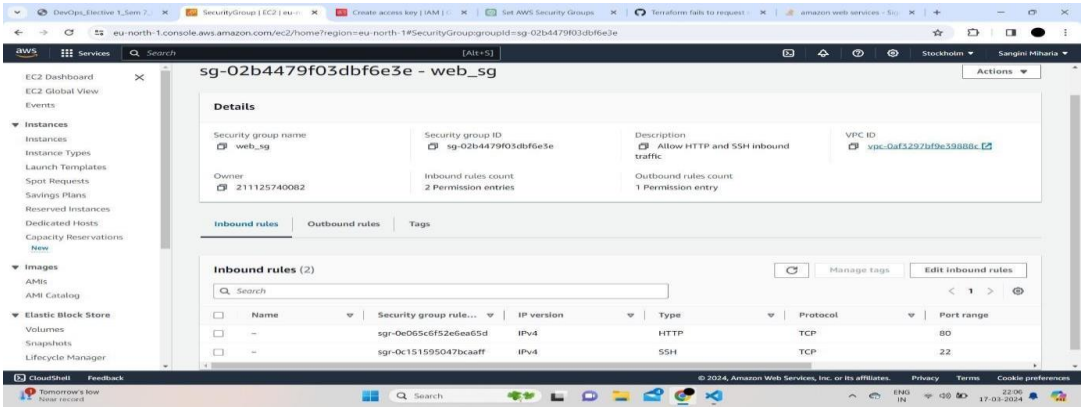
EXPERIMENT- 4

AIM: Create Security Group on AWS using Terraform








sg-02b4479f03dbf6e3e - web_sg
Actions
Details
Security group name
web_sg
Security group ID
sg-02b4479f03dbf6e3e
Description
Allow HTTP and SSH inbound traffic
VPC ID
vpc-0af5297bf9e39886c
Owner
211125740082
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules (2)
Manage tags
Edit inbound rules
Name
Security group rule...
IP version
Type
Protocol
Port range
sgr-0e065c6f52e6ea65d
IPv4
HTTP
TCP
80
sgr-0c151595047bcaaff
IPv4
SSH
TCP
22

EXPERIMENT- 5

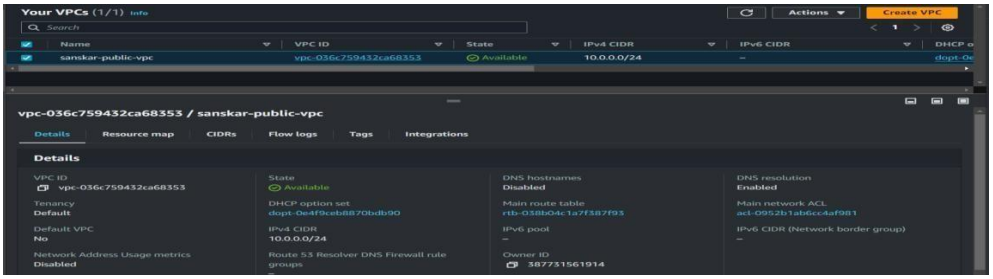
Aim:

Write the terraform script to create the VPC.

```
resource "aws_vpc" "sanskar_public_vpc" {
  cidr_block ="10.0.0.0/24"
  instance_tenancy ="default"
  tags={
    Name = "sanskar-public-vpc"
  }
}
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Terraform will perform the following actions:

  # aws_vpc.sanskar_public_vpc will be created
  + resource "aws_vpc" "sanskar_public_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/24"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "sanskar-public-vpc"
        }
      + tags_all                             = {
          + "Name" = "sanskar-public-vpc"
        }
    }
```

```
      + tags                                 = {
          + "Name" = "sanskar-public-vpc"
        }
      + tags_all                             = {
          + "Name" = "sanskar-public-vpc"
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.sanskar_public_vpc: Creating...
aws_vpc.sanskar_public_vpc: Creation complete after 2s [id=vpc-03b672a9081f166e2]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



EXPERIMENT- 6 : Write the script to create multiple instances using the variable having different configurations.

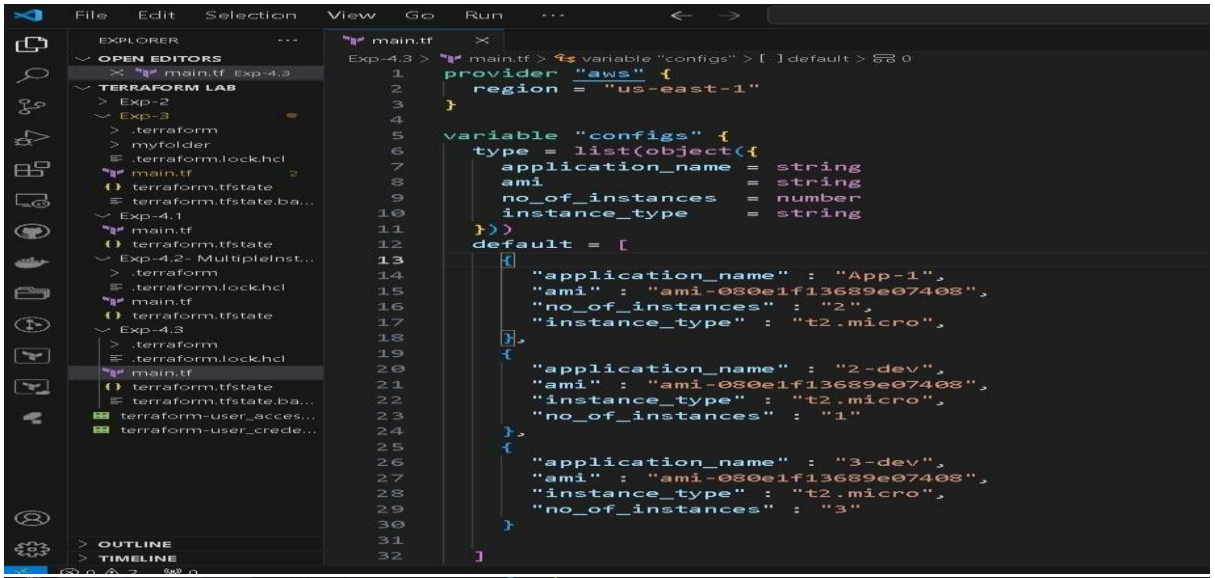
Aim:

1) Variables

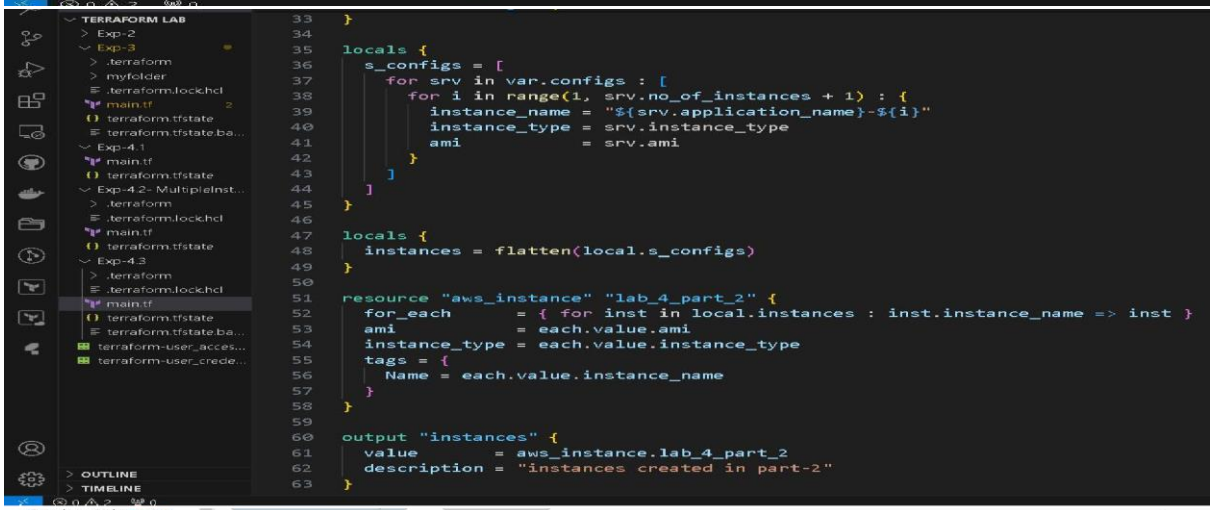
```
variable "template" {
  type = string
  default = "01000000-0000-4000-8000-000030080200"
}
output "string_output" {
   value = var.template
}

variable "users" {
  type = list
  default = ["root", "user1", "user2"]
}

output "list_output"{
      value=var.users
}

variable "plans" {
  type = map
  default = {
  "5USD" = "1xCPU-1GB"
  "10USD" = "1xCPU-2GB"
  "20USD" = "2xCPU-4GB"
  }
}
output "Maps_output"{
      value=var.plans
}

variable "set_password" {
  default = false
}
output "Boolean_output"{
      value=var.set_password
}
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

Boolean_output = false
Maps_output = tomap({
  "10USD" = "1xCPU-2GB"
  "20USD" = "2xCPU-4GB"
  "5USD" = "1xCPU-1GB"
})
list_output = tolist([
  "root",
  "user1",
  "user2",
])
string_output = "01000000-0000-4000-8000-000030080200"
```

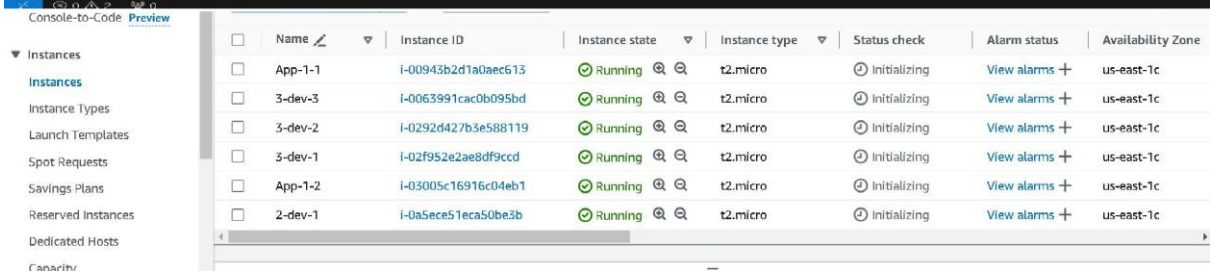
Create multiple instances using the variable having different configurations



```
provider "aws" {
  region = "us-east-1"
}

variable "configs" {
  type = list(object({
    application_name = string
    ami              = string
    no_of_instances  = number
    instance_type    = string
  }))
  default = [
    {
      "application_name" : "App-1",
      "ami" : "ami-080e1f13689e07408",
      "no_of_instances" : "2",
      "instance_type" : "t2.micro",
    },
    {
      "application_name" : "2-dev",
      "ami" : "ami-080e1f13689e07408",
      "instance_type" : "t2.micro",
      "no_of_instances" : "1"
    },
    {
      "application_name" : "3-dev",
      "ami" : "ami-080e1f13689e07408",
      "instance_type" : "t2.micro",
      "no_of_instances" : "3"
    }

  ]
```



```
}

locals {
  s_configs = [
    for srv in var.configs : [
      for i in range(1, srv.no_of_instances + 1) : {
        instance_name = "${srv.application_name}-${i}"
        instance_type = srv.instance_type
        ami           = srv.ami
      }
    ]
  ]
}

locals {
  instances = flatten(local.s_configs)
}

resource "aws_instance" "lab_4_part_2" {
  for_each      = { for inst in local.instances : inst.instance_name => inst }
  ami           = each.value.ami
  instance_type = each.value.instance_type
  tags = {
    Name = each.value.instance_name
  }
}

output "instances" {
  value       = aws_instance.lab_4_part_2
  description = "instances created in part-2"
}
```



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
App-1-1	i-00943b2d1a0aec613	Running	t2.micro	Initializing	View alarms +	us-east-1c
3-dev-3	i-0063991cac0b095bd	Running	t2.micro	Initializing	View alarms +	us-east-1c
3-dev-2	i-0292d427b3e588119	Running	t2.micro	Initializing	View alarms +	us-east-1c
3-dev-1	i-02f952e2ae8df9ccd	Running	t2.micro	Initializing	View alarms +	us-east-1c
App-1-2	i-03005c16916c04eb1	Running	t2.micro	Initializing	View alarms +	us-east-1c
2-dev-1	i-0a5ece51eca50be3b	Running	t2.micro	Initializing	View alarms +	us-east-1c

EXPERIMENT 7

Write the steps for setting up the Ansible.

- sudo apt-get -y upgrade && sudo apt-get -y update

```
root@Sanskar:~#  sudo apt-get -y upgrade && sudo apt-get -y update
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  python3-update-manager ubuntu-advantage-tools update-manager-core
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Hit:1 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:3 http://archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2821 kB]
Hit:6 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [952 kB]
Fetched 4000 kB in 4s (1084 kB/s)
Reading package lists... Done
root@Sanskar:~# sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.99.9.12).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@Sanskar:~# sudo apt-add-repository --yes --update ppa:ansible/ansible
Hit:1 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
root@Sanskar:~#  sudo apt install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-bcrypt python3-jmespath python3-kerberos python3-ntlm-auth python3-packaging python3-paramiko
  python3-pyparsing python3-requests-kerberos python3-requests-ntlm python3-resolvelib python3-winrm python3-xmltodict
  sshpass
```

- sudo apt install software-properties-common

```
0 upgraded, 15 newly installed, 0 to remove and 3 not upgraded.
Need to get 22.3 MB of archives.
After this operation, 323 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 python3-pyparsing all 2.4.6-1 [61.3 kB]
Get:2 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 python3-resolvelib all 0.5.4-2ppa~focal [12.6 kB]
Get:3 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 ansible-core all 2.12.10-1ppa~focal [943 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/main amd64 python3-packaging all 20.3-1 [26.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jmespath all 0.9.4-2ubuntu1 [21.5 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 python3-kerberos amd64 1.1.14-3.1build1 [22.6 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/universe amd64 python3-ntlm-auth all 1.1.0-1 [19.6 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/main amd64 python3-bcrypt amd64 3.1.7-2ubuntu1 [30.4 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-paramiko all 2.6.0-2ubuntu0.3 [123 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-ntlm all 1.1.0-1 [6004 B]
Get:12 http://archive.ubuntu.com/ubuntu focal/universe amd64 python3-xmltodict all 0.12.0-1 [12.6 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal/universe amd64 python3-winrm all 0.3.0-2 [21.7 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal/universe amd64 sshpass amd64 1.06-1 [10.5 kB]
Get:15 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 ansible all 5.10.0-1ppa~focal [21.0 MB]
Fetched 22.3 MB in 5s (4704 kB/s)
Selecting previously unselected package python3-pyparsing.
(Reading database ... 32760 files and directories currently installed.)
Preparing to unpack .../00-python3-pyparsing_2.4.6-1_all.deb ...
Unpacking python3-pyparsing (2.4.6-1) ...
Selecting previously unselected package python3-packaging.
Preparing to unpack .../01-python3-packaging_20.3-1_all.deb ...
Unpacking python3-packaging (20.3-1) ...
Selecting previously unselected package python3-resolvelib.
Preparing to unpack .../02-python3-resolvelib_0.5.4-2ppa~focal_all.deb ...
Unpacking python3-resolvelib (0.5.4-2ppa~focal) ...
Selecting previously unselected package ansible-core.
Preparing to unpack .../03-ansible-core_2.12.10-1ppa~focal_all.deb ...
Unpacking ansible-core (2.12.10-1ppa~focal) ...
Selecting previously unselected package ansible.
Preparing to unpack .../04-ansible_5.10.0-1ppa~focal_all.deb ...
Unpacking ansible (5.10.0-1ppa~focal) ...
Selecting previously unselected package python3-jmespath.
Preparing to unpack .../05-python3-jmespath_0.9.4-2ubuntu1_all.deb ...
Unpacking python3-jmespath (0.9.4-2ubuntu1) ...
Selecting previously unselected package python3-kerberos.
Preparing to unpack .../06-python3-kerberos_1.1.14-3.1build1_amd64.deb ...
```

- sudo apt-add-repository --yes --update ppa:ansible/ansible
- sudo apt install ansible

```
Setting up python3-pyparsing (2.4.6-1) ...
Setting up python3-jmespath (0.9.4-2ubuntu1) ...
Setting up python3-requests-kerberos (0.12.0-2) ...
Setting up python3-paramiko (2.6.0-2ubuntu0.3) ...
Setting up python3-requests-ntlm (1.1.0-1) ...
Setting up python3-packaging (20.3-1) ...
Setting up python3-winrm (0.3.0-2) ...
Setting up ansible-core (2.12.10-1ppa~focal) ...
Setting up ansible (5.10.0-1ppa~focal) ...
Processing triggers for man-db (2.9.1-1) ...
root@Sanskar:~# ansible --version
ansible [core 2.12.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 22 2023, 10:22:35) [GCC 9.4.0]
  jinja version = 2.10.1
  libyaml = True
root@Sanskar:~#
```

EXPERIMENT- 8

WRITE AN ANSIBLE PLAYBOOK TO PRINT HELLO-WORLD USING YML

```
  GNU nano 4.8                                          hello.yml
---
- name: Generic Playbook Name
  hosts: "{{ target_hosts | default('localhost') }}"
  tasks:
    - name: Print Hello, World!
      debug:
        msg: "Hello, World!"
```

```
root@Sanskar:~# nano hello.yml
root@Sanskar:~# ansible-playbook hello.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match
'all'

PLAY [Generic Playbook Name] ***************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [localhost]

TASK [Print Hello, World!] *****************************************************************************************
ok: [localhost] => {
    "msg": "Hello, World!"
}

PLAY RECAP *********************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@Sanskar:~#
```

EXPERIMENT- 9

WRITE AN ANSIBLE PLAYBOOK AND ANSIBLE SHELL COMMAND USING YAML.

```
  GNU nano 4.8                                                          e
---
- name: Execute Shell Command
  hosts: localhost
  tasks:
    - name: Run a Shell Command
      shell: echo "Hello, World!"
      register: shell_output

    - name: Display Output
      debug:
        var: shell_output.stdout
```

```
root@Sanskar:~# nano hello.yml
root@Sanskar:~# nano exp2.yml
root@Sanskar:~# ansible-playbook exp2.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match
'all'

PLAY [Execute Shell Command] ***************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [localhost]

TASK [Run a Shell Command] *****************************************************************************************
changed: [localhost]

TASK [Display Output] **********************************************************************************************
ok: [localhost] => {
    "shell_output.stdout": "Hello, World!"
}

PLAY RECAP *********************************************************************************************************
localhost                  : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

EXPERIMENT- 10

WRITE AN ANSIBLE PLAYBOOK TO DECLARE THE VARIABLE USING YAML

```
---
- name: Example playbook with variable declaration
  hosts: localhost
  vars:
    message: "Hello, World!"
    number: 42
    fruits:
      - apple
      - banana
      - orange
  tasks:
    - name: Print message variable
      debug:
        msg: "{{ message }}"

    - name: Print number variable
      debug:
        msg: "{{ number }}"

    - name: Print fruits variable
      debug:
        msg: "{{ fruits }}"
```

```
root@Sanskar:~# nano exp2.yml
root@Sanskar:~# nano exp3.yml
root@Sanskar:~# ansible-playbook exp3.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Example playbook with variable declaration] ******************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************
ok: [localhost]

TASK [Print message variable] ***************************************************************************************************************
ok: [localhost] => {
    "msg": "Hello, World!"
}

TASK [Print number variable] ****************************************************************************************************************
ok: [localhost] => {
    "msg": 42
}

TASK [Print fruits variable] ****************************************************************************************************************
ok: [localhost] => {
    "msg": [
        "apple",
        "banana",
        "orange"
    ]
}

PLAY RECAP **********************************************************************************************************************************
localhost                  : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

EXPERIMENT -11

1. Write a playbook to print the default ipv4 address of each host along with the host name.

```
root@Sanskar:~# nano /etc/ansible/hosts
root@Sanskar:~# ansible-playbook -i /etc/ansible/hosts print_ipv4.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match
'all'

PLAY [Print Default IPv4 Addresses of Hosts] ******************************************************************
skipping: no hosts matched

PLAY RECAP ****************************************************************************************************
```

Command -: $ ansible localhost -u ansible -m setup

```
root@Sanskar:~#  ansible localhost -u ansible -m setup
localhost | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "172.19.30.14"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::215:5dff:feda:65e4"
        ],
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "",
        "ansible_bios_vendor": "",
        "ansible_bios_version": "",
        "ansible_board_asset_tag": "",
        "ansible_board_name": "",
        "ansible_board_serial": "",
        "ansible_board_vendor": "",
        "ansible_board_version": "",
        "ansible_chassis_asset_tag": "",
        "ansible_chassis_serial": "",
        "ansible_chassis_vendor": "",
        "ansible_chassis_version": "",
        "ansible_cmdline": {
            "WSL_ROOT_INIT": "1",
```

EXPERIMENT- 12

Write a playbook to print the user defined variables from the command line.

```
---
- name: Print User-Defined Variables
  hosts: localhost
  gather_facts: false
  vars:
    user_defined_var1: "{{ user_var1 }}"
    user_defined_var2: "{{ user_var2 }}"

  tasks:
    - name: Display User-Defined Variables
      debug:
        msg: "User-defined Variable 1: {{ user_defined_var1 }}, User-defined Variable 2: {{ user_defined_var2 }}"
```

```
root@Sanskar:~# ansible-playbook -i localhost, print_user_vars.yml -e "user_var1=value1 user_var2=value2"

PLAY [Print User-Defined Variables] ********************************************************************************

TASK [Display User-Defined Variables] ******************************************************************************
ok: [localhost] => {
    "msg": "User-defined Variable 1: value1, User-defined Variable 2: value2"
}

PLAY RECAP *********************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
---
- name: Display Gym and Yoga Products
  hosts: localhost
  gather_facts: false

  tasks:
    - name: Display Gym Product
      debug:
        msg: "Gym Product: {{ gym_product }}"

    - name: Display Yoga Product
      debug:
        msg: "Yoga Product: {{ yoga_product }}"
```

```
root@Sanskar:~# nano multivariableplaybook.yml
root@Sanskar:~# ansible-playbook -i localhost multivariableplaybook.yml -e "gym_product=Dumbbells" -e "yoga_product=Yoga
 Mat"
[WARNING]: Unable to parse /root/localhost as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match
'all'

PLAY [Display Gym and Yoga Products] *******************************************************************************

TASK [Display Gym Product] *****************************************************************************************
ok: [localhost] => {
    "msg": "Gym Product: Dumbbells"
}

TASK [Display Yoga Product] ****************************************************************************************
ok: [localhost] => {
    "msg": "Yoga Product: Yoga"
}

PLAY RECAP *********************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

EXPERIMENT- 13

Write the playbook to show the working of loops.

```
- name: Generate Gym Products List
  hosts: localhost
  gather_facts: false

  vars:
    gym_products:
      - name: "Dumbbells"
        weight: "5kg"
        price: "$20"
      - name: "Jump Rope"
        length: "Adjustable"
        price: "$10"
      - name: "Yoga Mat"
        size: "Standard"
        price: "$15"
      - name: "Resistance Bands"
        level: "Medium"
        price: "$25"

  tasks:
    - name: Display Gym Products
      debug:
        msg: |
          Product: {{ item.name }}
          {% if item.weight is defined %}
```

```
root@Sanskar:~# nano print_user_vars.yml
root@Sanskar:~# nano loopingplaybook.yml
root@Sanskar:~# ansible-playbook -i localhost loopingplaybook.yml
[WARNING]: Unable to parse /root/localhost as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match
'all'

PLAY [Generate Gym Products List] ***********************************************************************************

TASK [Display Gym Products] *****************************************************************************************
ok: [localhost] => (item={'name': 'Dumbbells', 'weight': '5kg', 'price': '$20'}) => {
    "msg": "Product: Dumbbells\nWeight: 5kg\nPrice: $20\n"
}
ok: [localhost] => (item={'name': 'Jump Rope', 'length': 'Adjustable', 'price': '$10'}) => {
    "msg": "Product: Jump Rope\nLength: Adjustable\nPrice: $10\n"
}
ok: [localhost] => (item={'name': 'Yoga Mat', 'size': 'Standard', 'price': '$15'}) => {
    "msg": "Product: Yoga Mat\nSize: Standard\nPrice: $15\n"
}
ok: [localhost] => (item={'name': 'Resistance Bands', 'level': 'Medium', 'price': '$25'}) => {
    "msg": "Product: Resistance Bands\nLevel: Medium\nPrice: $25\n"
}

PLAY RECAP **********************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

EXPERIMENT- 14

Write a playbook to print the user defined variables from the command line.

```
---
- name: Print User-Defined Variables
  hosts: localhost
  gather_facts: false
  vars:
    user_defined_var1: "{{ user_var1 }}"
    user_defined_var2: "{{ user_var2 }}"

  tasks:
    - name: Display User-Defined Variables
      debug:
        msg: "User-defined Variable 1: {{ user_defined_var1 }}, User-defined Variable 2: {{ user_defined_var2 }}"
```

```
root@Sanskar:~# ansible-playbook -i localhost, print_user_vars.yml -e "user_var1=value1 user_var2=value2"

PLAY [Print User-Defined Variables] ************************************************************************************

TASK [Display User-Defined Variables] ***********************************************************************************
ok: [localhost] => {
    "msg": "User-defined Variable 1: value1, User-defined Variable 2: value2"
}

PLAY RECAP *************************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
---
- name: Display Gym and Yoga Products
  hosts: localhost
  gather_facts: false

  tasks:
    - name: Display Gym Product
      debug:
        msg: "Gym Product: {{ gym_product }}"

    - name: Display Yoga Product
      debug:
        msg: "Yoga Product: {{ yoga_product }}"
```

```
root@Sanskar:~# nano multivariableplaybook.yml
root@Sanskar:~# ansible-playbook -i localhost multivariableplaybook.yml -e "gym_product=Dumbbells" -e "yoga_product=Yoga
 Mat"
[WARNING]: Unable to parse /root/localhost as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match
'all'

PLAY [Display Gym and Yoga Products] ************************************************************************************

TASK [Display Gym Product] **********************************************************************************************
ok: [localhost] => {
    "msg": "Gym Product: Dumbbells"
}

TASK [Display Yoga Product] *********************************************************************************************
ok: [localhost] => {
    "msg": "Yoga Product: Yoga"
}

PLAY RECAP *************************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```