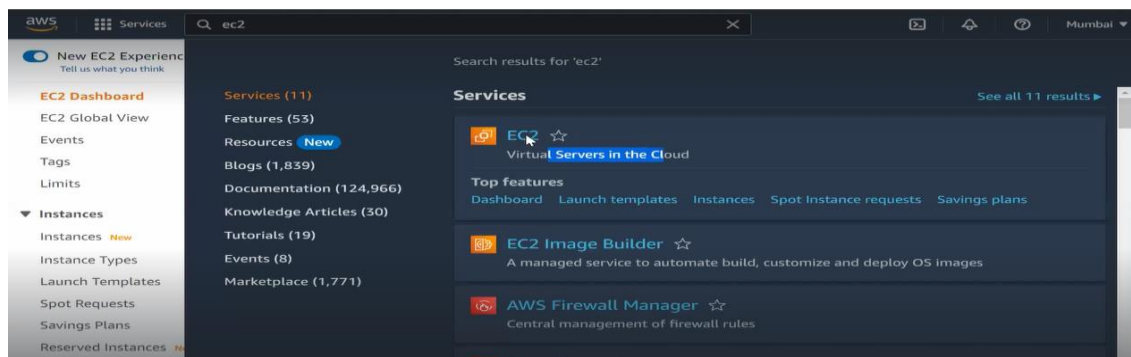




Ansible Automation

Session 1 – 26th Nov 2022 Summary

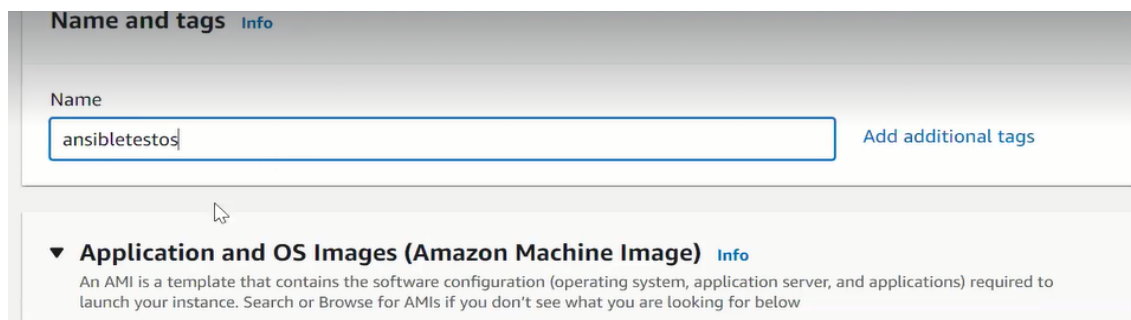
- Launch the RHEL9 OS on AWS Cloud - login into AWS account, select **EC2** user-



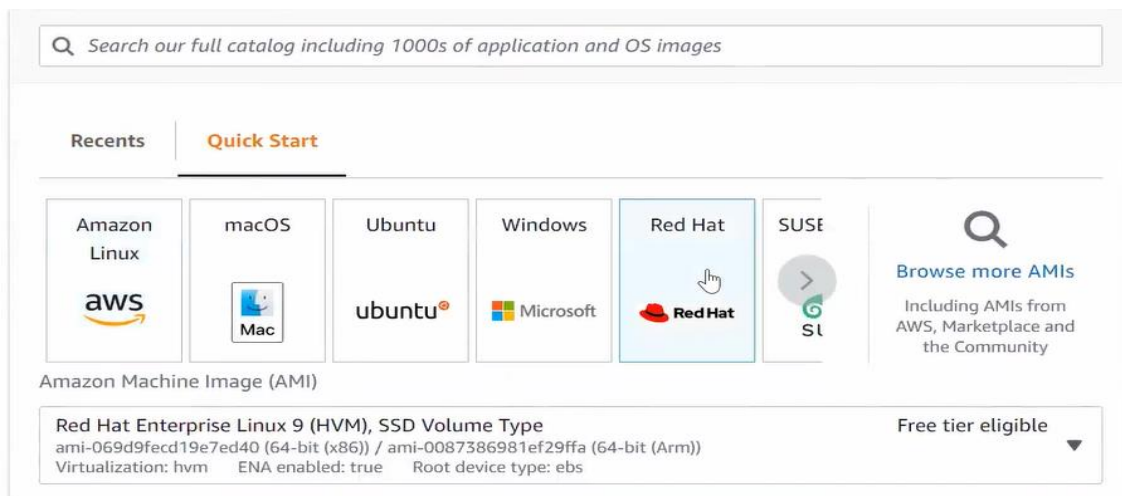
- Click on **Launch instance**, to launch an instance



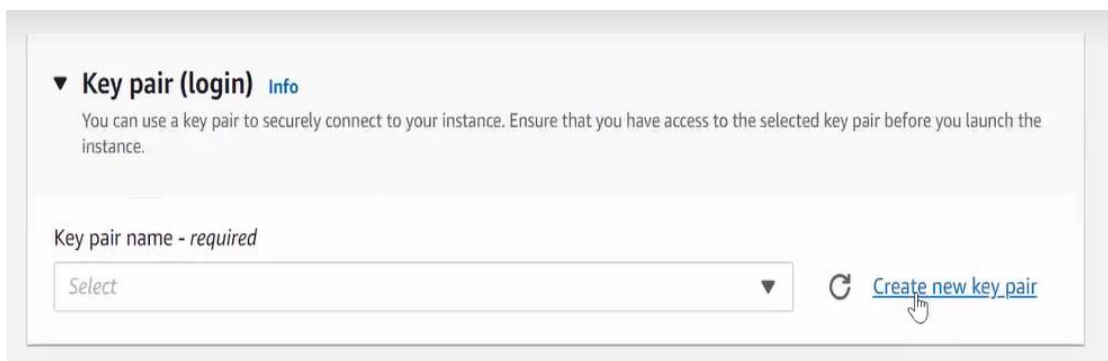
- Give a name to the instance



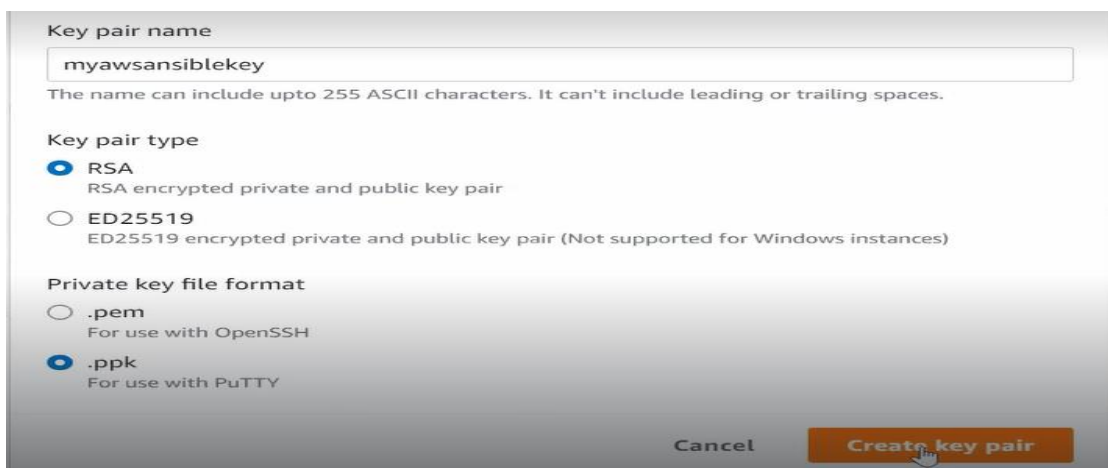
- Select RedHat OS – as here we are launching RHEL9 OS on AWS Cloud



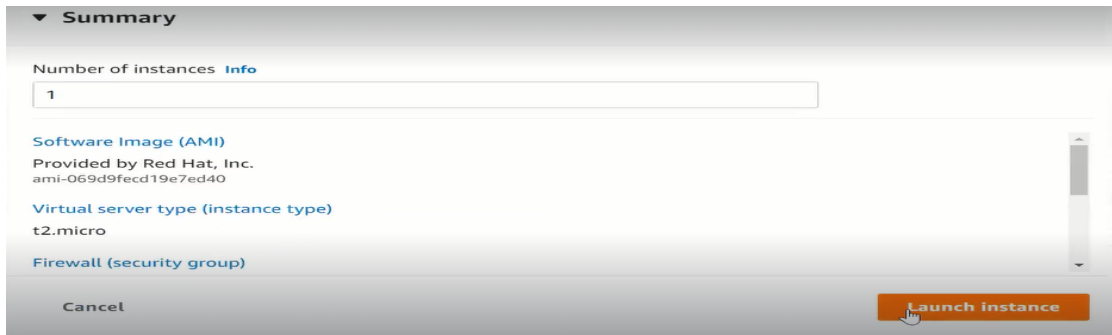
- The OS is launched in cloud, for login we need password – for this click on **Create new key pair**



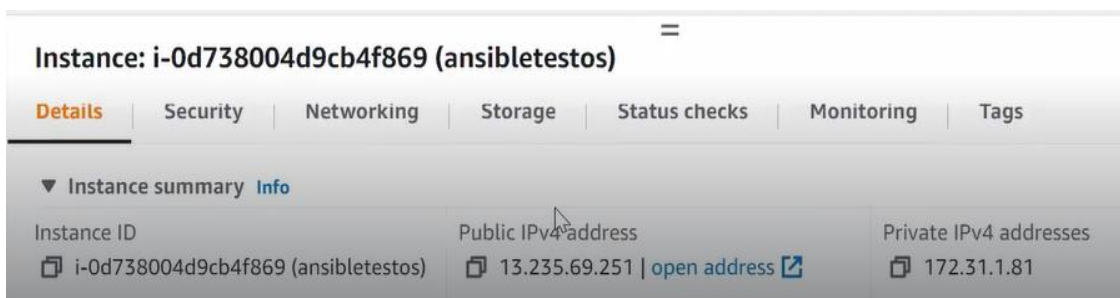
- Specify the **key pair name** and download the key in **.ppk** format, putty supports .ppk format and then click on **Create key pair**



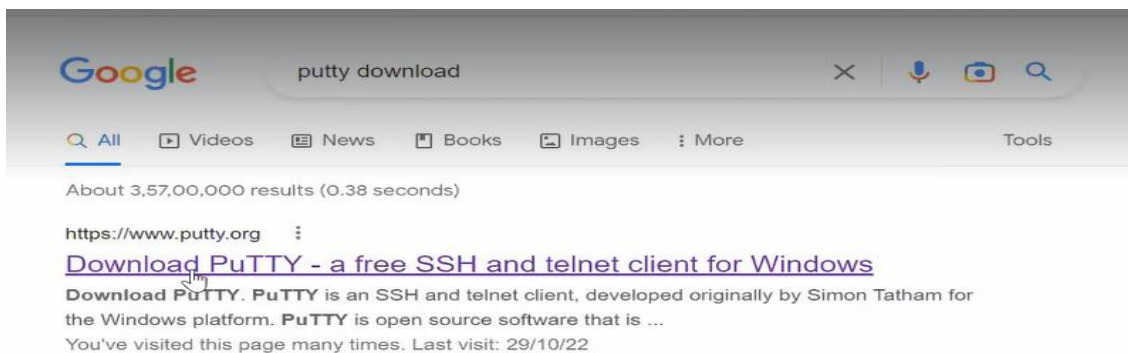
- Click on **Launch instance**

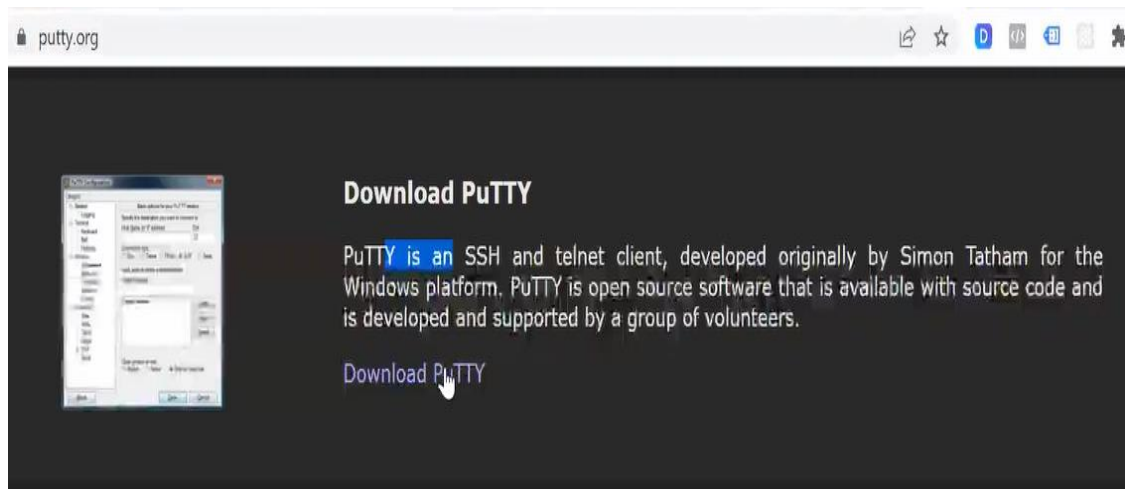


- The OS is successfully launched- click on the instance **id** – we see the detailed information like Public IP and Private IP address of the instance launched

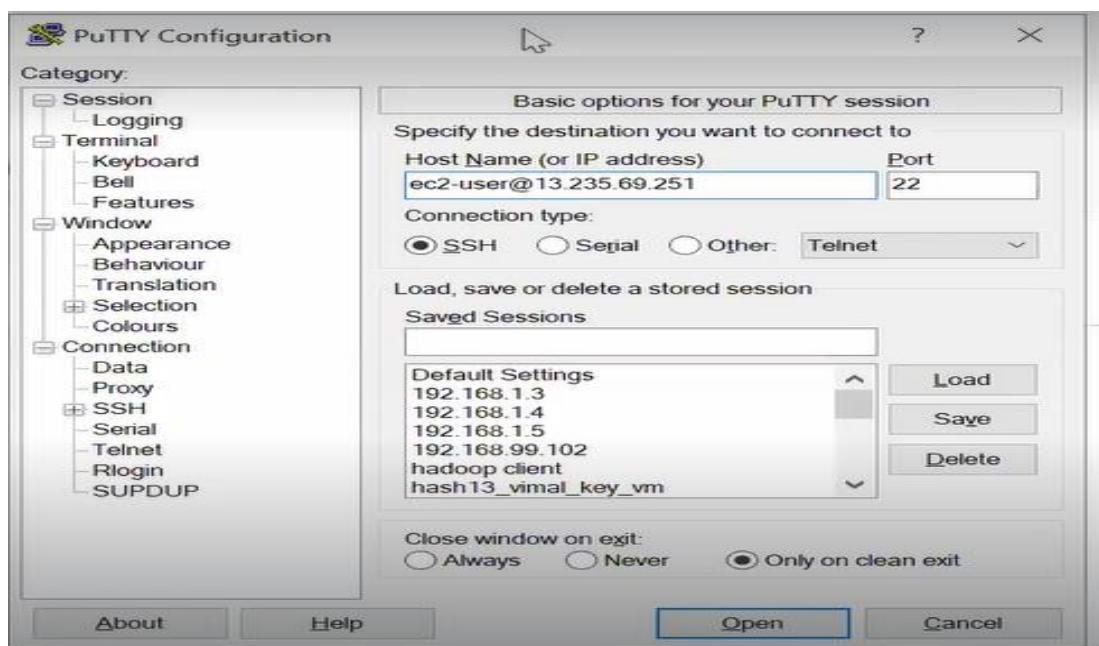


- We can use Putty to connect to the instance -

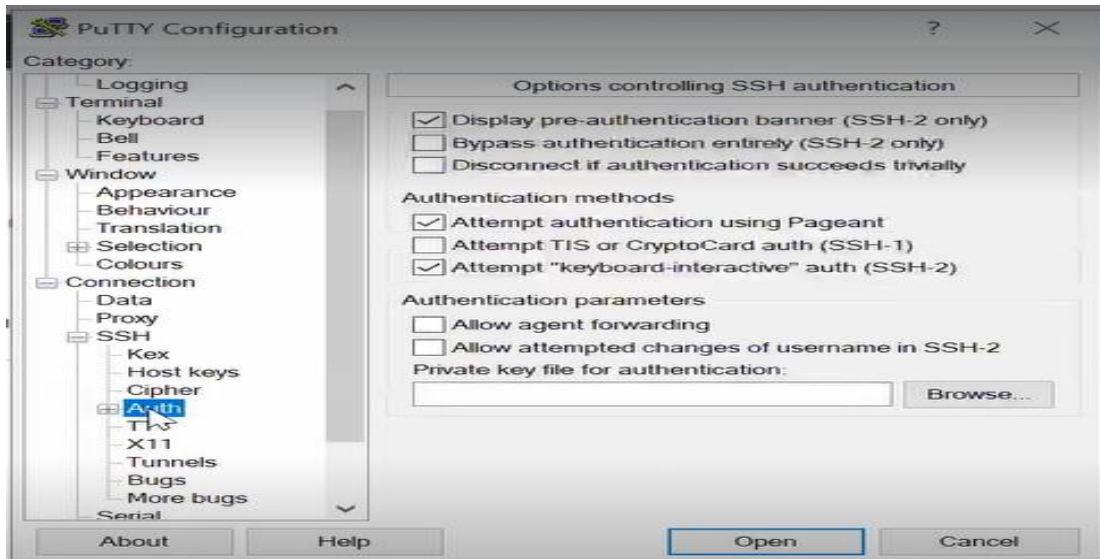




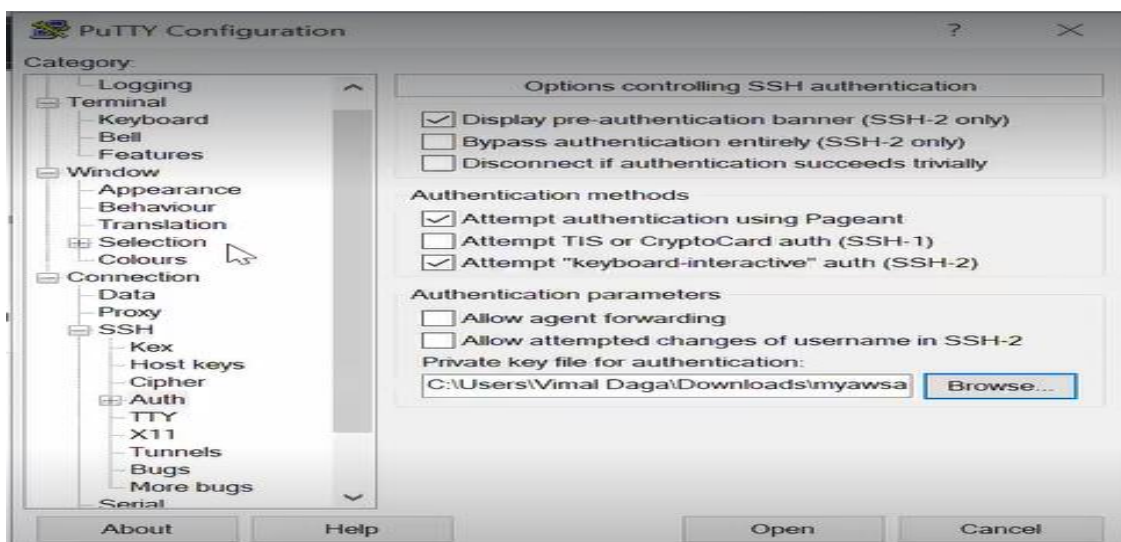
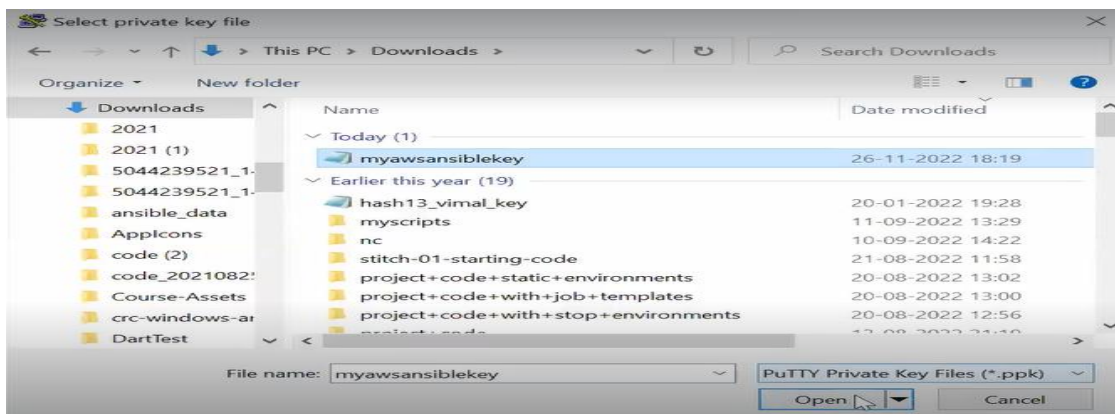
- Open the Putty software to connect to the instance - specify the **username** and **IP address** of the instance



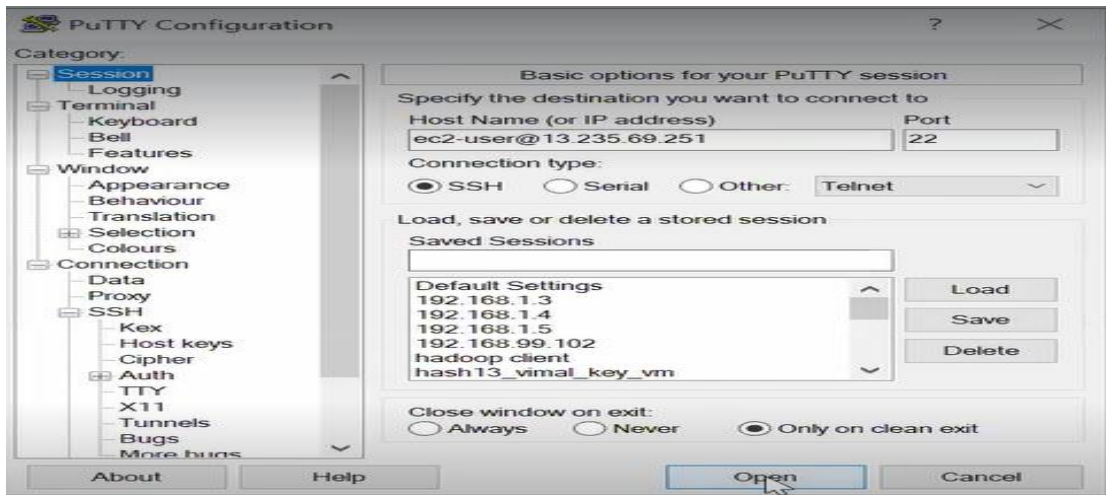
- To login to OS, we require password, the password is private key- for this click on Connection → SSH → Auth → Browse



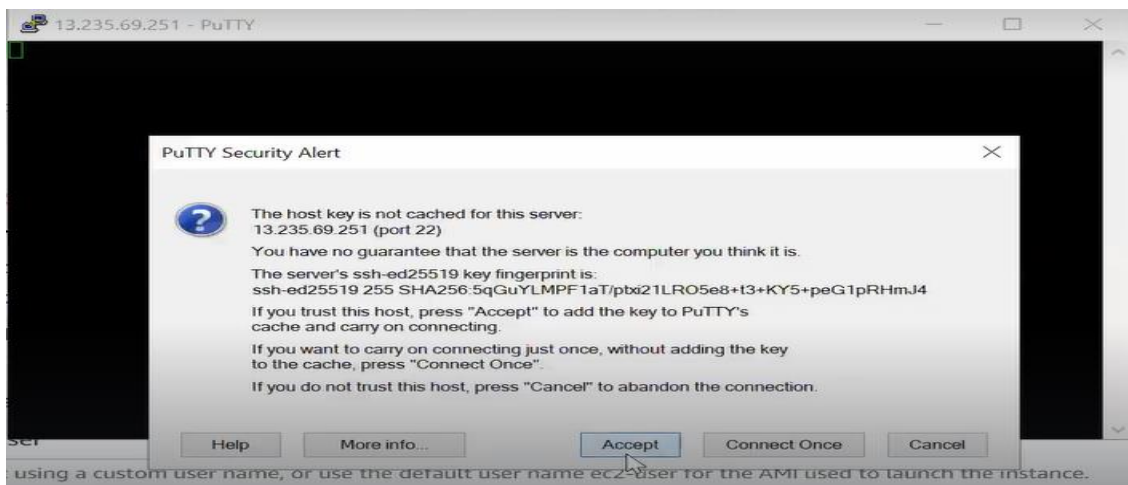
- We have to browse the key and connect to key that we downloaded



- Then click on **Sessions** → **Open**



- Click on **Accept**



- The command to login to the OS with the admin account is “**sudo su – root**”

```
[ec2-user@ip-172-31-1-81 ~]$  
[ec2-user@ip-172-31-1-81 ~]$ whoami  
ec2-user  
[ec2-user@ip-172-31-1-81 ~]$  
[ec2-user@ip-172-31-1-81 ~]$ sudo su - root  
[root@ip-172-31-1-81 ~]# whoami  
root  
[root@ip-172-31-1-81 ~]#
```

- The command to install ansible “**yum install ansible-core**”

```
[root@ip-172-31-1-81 ~]# yum install ansible-core
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:01:29 ago on Sat 26 Nov 2022 12:57:00 PM UTC.
Dependencies resolved.
```

- The command to check the version of ansible installed “**ansible --version**”

```
[root@ip-172-31-1-81 ~]# ansible --version
ansible [core 2.13.3]
  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/python3.9/site-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.10 (main, Feb  9 2022, 00:00:00) [GCC 11.2.1 20220127 (Red Hat 11.2.1-9)]
  jinja version = 3.1.2
  libyaml = True
[root@ip-172-31-1-81 ~]#
```

- Ansible tool can be used either **manually**(ad hoc commands) or **automatically**(playbook)
- The command used to query software “**dialog**” installed or not “**rpm -q dialog**”

```
[root@ip-172-31-1-81 ~]# rpm -q dialog
package dialog is not installed
```

- The command to install dialog software “**dnf install dialog**”

```
[root@ip-172-31-1-81 ~]# dnf install dialog
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:01:02 ago on Sat 26 Nov 2022 01:02:03 PM UTC.
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
dialog x86_64 1.3-32.20210117.el9 rhel-9-appstream-rhui-rpms 300 k
=====
```

- The command used to verify software “dialog” installed “**rpm -q dialog**”

```
[root@ip-172-31-1-81 ~]# rpm -q dialog
dialog-1.3-32.20210117.el9.x86_64
[root@ip-172-31-1-81 ~]#
```

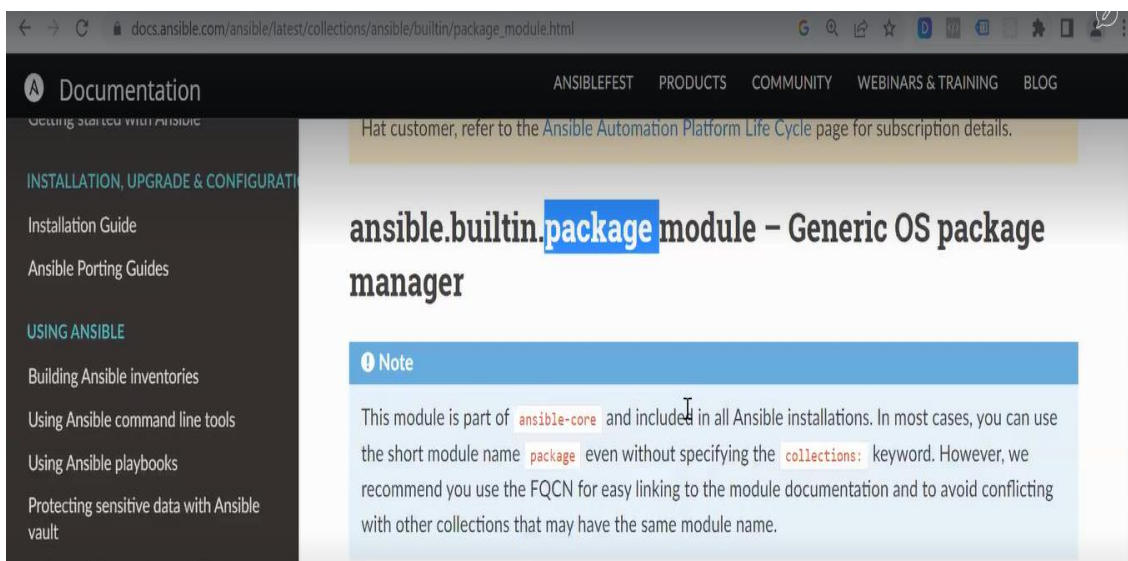
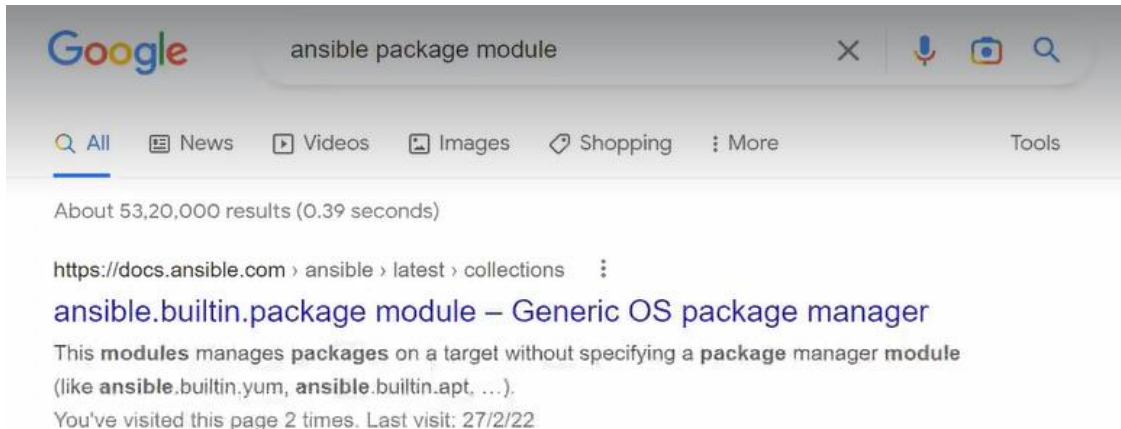
- The command to remove dialog software “**dnf remove dialog**”

```
[root@ip-172-31-1-81 ~]# dnf remove dialog
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Removing:
dialog x86_64 1.3-32.20210117.el9 @rhel-9-appstream-rhui-rpms 581 k
=====
```


- The package management can be done using “ansible adhoc commands”. The keyword used for package management is “**package**” and for this we have **modules** (- **m**). To specify the name of a package we have a keyword **name**. To specify either to install software (**present**) or remove a software (**absent**) we have a “**state**” keyword, the **name** and **state** keyword are put inside a **double quotes**, these are called as **attributes** (-**a**)



- When ansible installed on a system, we can specify two things
 - What to do - Ex:- install a software or package
 - Location – either the local host or others systems



- The ansible is not the one that is installing the software, it auto detects that the OS is RHEL9 , the command is “yum” or “dnf” to install the software and call the command behind the scene

```
[root@ip-172-31-1-81 ~]# ansible localhost -m package -a "name=dialog state=present"
localhost | CHANGED => {
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Installed: dialog-1.3-32.20210117.el9.x86_64"
  ]
}
```

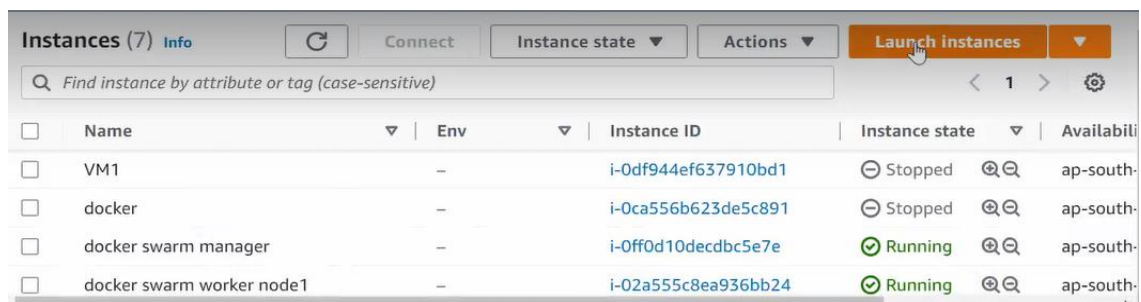
```
[root@ip-172-31-1-81 ~]#
[root@ip-172-31-1-81 ~]# rpm -q dialog
dialog-1.3-32.20210117.el9.x86_64
[root@ip-172-31-1-81 ~]#
```

- The ansible adhoc command to remove the software “dialog”

```
[root@ip-172-31-1-81 ~]# rpm -q dialog
dialog-1.3-32.20210117.el9.x86_64
[root@ip-172-31-1-81 ~]# ansible localhost -m package -a "name=dialog state=absent"
localhost | CHANGED => {
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Removed: dialog-1.3-32.20210117.el9.x86_64"
  ]
}

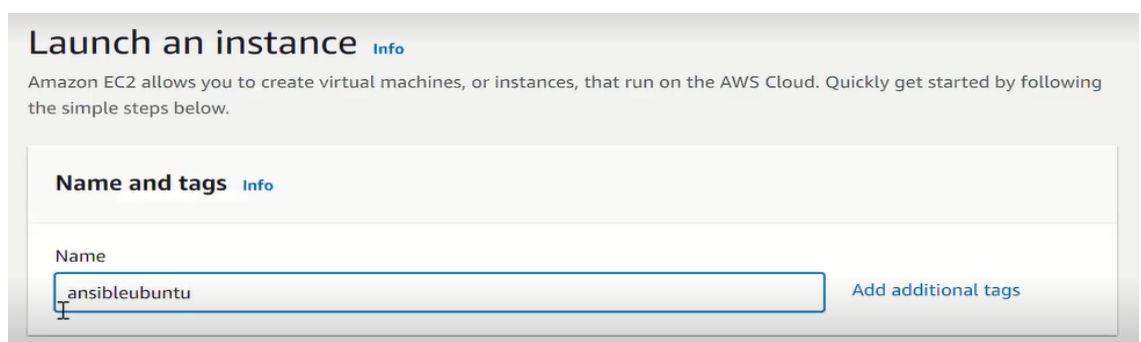
[root@ip-172-31-1-81 ~]# rpm -q dialog
package dialog is not installed
```

- To launch one more instance, the Ubuntu OS on AWS Cloud-



The screenshot shows the AWS Management Console 'Instances' page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below these is a search bar and a table of instances. The table has columns for Name, Env, Instance ID, Instance state, and Availability Zone. There are five instances listed: VM1 (Stopped), docker (Stopped), docker swarm manager (Running), and docker swarm worker node1 (Running).

<input type="checkbox"/>	Name	Env	Instance ID	Instance state	Availability
<input type="checkbox"/>	VM1	-	i-0df944ef637910bd1	⊖ Stopped	ap-south-
<input type="checkbox"/>	docker	-	i-0ca556b623de5c891	⊖ Stopped	ap-south-
<input type="checkbox"/>	docker swarm manager	-	i-0ff0d10decdbc5e7e	✔ Running	ap-south-
<input type="checkbox"/>	docker swarm worker node1	-	i-02a555c8ea936bb24	✔ Running	ap-south-



The screenshot shows the 'Launch an instance' wizard. The first step is 'Name and tags'. The 'Name' field is filled with 'ansibleubuntu'. There is a button 'Add additional tags' to the right of the field.

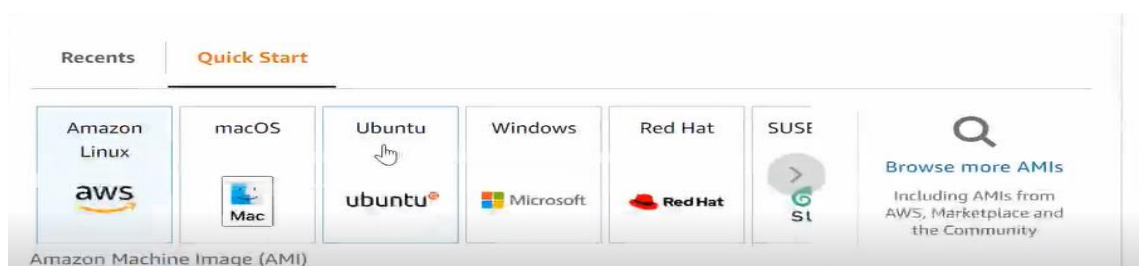
Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

ansibleubuntu [Add additional tags](#)



▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

myawsansiblekey

Create new key pair

▼ Summary

Number of instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-062df10d14676e201

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Cancel Launch instance

➤ Login to Ubuntu OS

Instances (1/1) Info Refresh Connect Instance state ▼ Actions ▼ Launch instances ▼

Find instance by attribute or tag (case-sensitive)

Instance ID = i-0e692b8ee9129a3c1 X Clear filters

<input checked="" type="checkbox"/>	Name	Env	Instance ID	Instance state	Availability
<input checked="" type="checkbox"/>	ansibleubuntu	-	i-0e692b8ee9129a3c1	Running	ap-south-

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID

i-0e692b8ee9129a3c1 (ansibleubuntu)

Public IP address

15.206.128.216

User name

ubuntu

Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel Connect

➤ The command to login to the Ubuntu OS with the admin account is “**sudo su – root**”

```
ubuntu@ip-172-31-6-207:~$
ubuntu@ip-172-31-6-207:~$ sudo su - root
root@ip-172-31-6-207:~#
```

- The command to verify software installed or not “**dpkg - - list dialog**” in Ubuntu OS

```
root@ip-172-31-6-207:~#  
root@ip-172-31-6-207:~# dpkg --get-selections | grep dialog  
Desired=Unknown/Install/Remove/Purge/Hold  
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend  
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)  
||/ Name           Version           Architecture Description  
++-=====+-----+-----+-----+  
un dialog          <none>           <none>         (no description available)  
root@ip-172-31-6-207:~#
```

- The command used to update the system about **setting-up repository** “**apt-get update**”

```
root@ip-172-31-6-207:~# apt-get update  
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]  
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [114 kB]  
Fetched 338 kB in 2s (169 kB/s)  
Reading package lists... Done
```

- The command to install ansible “**apt-get install ansible**”

```
root@ip-172-31-6-207:~# apt-get install ansible  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages will be installed:  
  ansible  
The following NEW packages will be installed:  
  ansible  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 1.1 MB of archives.  
After this operation, 4.2 MB of additional disk space will be used.  
Do you want to continue? [Y/n]  
Y  
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ansible amd64 2.10.8-1ubuntu0.1 [1.1 MB]  
Fetched 1.1 MB in 2s (540 kB/s)  
debconf: delaying package configuration, since apt-utils is not installed  
(Reading database ... 1234567890 files and directories currently installed.)  
Preparing to unpack .../ansible_2.10.8-1ubuntu0.1_amd64.deb ...  
Unpacking ansible (2.10.8-1ubuntu0.1) ...  
Setting up ansible (2.10.8-1ubuntu0.1) ...  
root@ip-172-31-6-207:~#
```

- The command used to check the version of ansible installed “**ansible - - version**”

```
root@ip-172-31-6-207:~# ansible --version  
ansible 2.10.8  
  config file = None  
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']  
  ansible python module location = /usr/lib/python3/dist-packages/ansible  
  executable location = /usr/bin/ansible  
  python version = 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0]  
root@ip-172-31-6-207:~#
```


- The command used to install the software “**dialog**” is “**ansible localhost -m package -a "name=dialog state=present"**”. Here in Ubuntu OS, we need not know the command to install the software, the ansible detects the OS is Ubuntu, figures out the command and installs the software. The ansible gives output in **orange colour**, it means it has **changed the state**.

```
root@ip-172-31-6-207:~# ansible localhost -m package -a "name=dialog state=present"
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | CHANGED => {
  "cache_update_time": 1669469243,
  "cache_updated": false,
  "changed": true,
  "stderr": "",
  "stderr_lines": [],
```

- The command to verify software installed or not “**dpkg - - list dialog**” in Ubuntu OS

```
root@ip-172-31-6-207:~# dpkg --get-selections | grep dialog
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version        Architecture Description
++-=====
ii dialog          1.3-20211214-1 amd64        Displays user-friendly dialog boxes from shell scripts
root@ip-172-31-6-207:~#
```

- If you try to install the “**dialog**” software again, the ansible detects that the software is already installed, it gives output in green colour – it means what we are looking for is available in the OS, the software “**dialog**” that we are trying to install is already in the system. This concept is called as **Idempotent**.

```
root@ip-172-31-6-207:~# ansible localhost -m package -a "name=dialog state=present"
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | SUCCESS => {
  "cache_update_time": 1669469243,
  "cache_updated": false,
  "changed": false
```