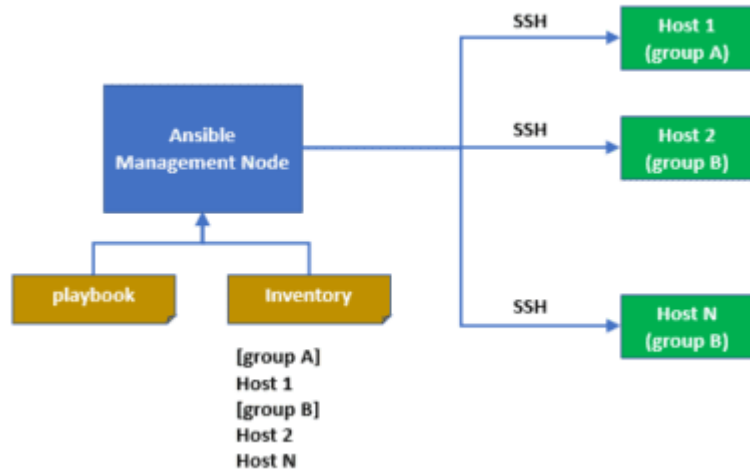


Ansible on AWS



- **Create EC2 Named Controller**
- **Choice Ubuntu as an Operating System**

The screenshot shows the AWS Management Console interface for launching an EC2 instance. The page is titled "Launch an instance" and includes a "Summary" sidebar on the right.

Name and tags: The instance name is set to "Controller".

Application and OS Images (Amazon Machine Image): The "Quick Start" section shows various operating system options. **Ubuntu** is selected, with the "ubuntu" logo visible. Other options include Amazon Linux, macOS, Windows, Red Hat, and SUSE Linux.

Summary sidebar:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 22.04 LTS, ...read more (ami-053b0453c279acc90)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

Buttons at the bottom right include "Cancel", "Launch instance", and "Review commands".

- Create a ppk Key
- Review and Create the EC2 (controller)

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

PPK Key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair

Private key file format

☐ .pem
For use with OpenSSH

☒ .ppk
For use with PuTTY

⚠

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel

Create key pair

- Create Two EC2 (worker)
- Choose RedHat as an Operating System

EC2 > Instances > Launch an instance

Launch an instance

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

Name

Redhat

Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Li

Browse more AMIs

Summary

Number of instances

1

Software Image (AMI)

Provided by Red Hat, Inc.

ami-026ebd4cfe2c043b2

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-1

Storage (volumes)

1 volume(s) - 10 GiB

Cancel

Launch instance

Review commands

- Select the same key for EC2 (Worker) and EC2 (controller)
- Select the Same SG for all instance

▼ 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*

PPK Key

Q |

Proceed without a key pair (Not recommended) Default value

PPK Key

Type: rsa

Create new key pair

Edit

▼ Network settings Info

Network Info

vpc-08a00530d2770b20f

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups Info

Select security groups

launch-wizard-1 sg-0167b08c2cd21db0a X

VPC: vpc-08a00530d2770b20f

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Edit

▼ Summary

Number of instances Info

1

Software Image (AMI)

Provided by Red Hat, Inc.

ami-02b0b04fc2c0403d2

Virtual server type (instance type)

t2.micro

Firewall (security group)

launch-wizard-1

Storage (volumes)

1 volume(s) - 10 GiB

Cancel

Launch instance

Review commands

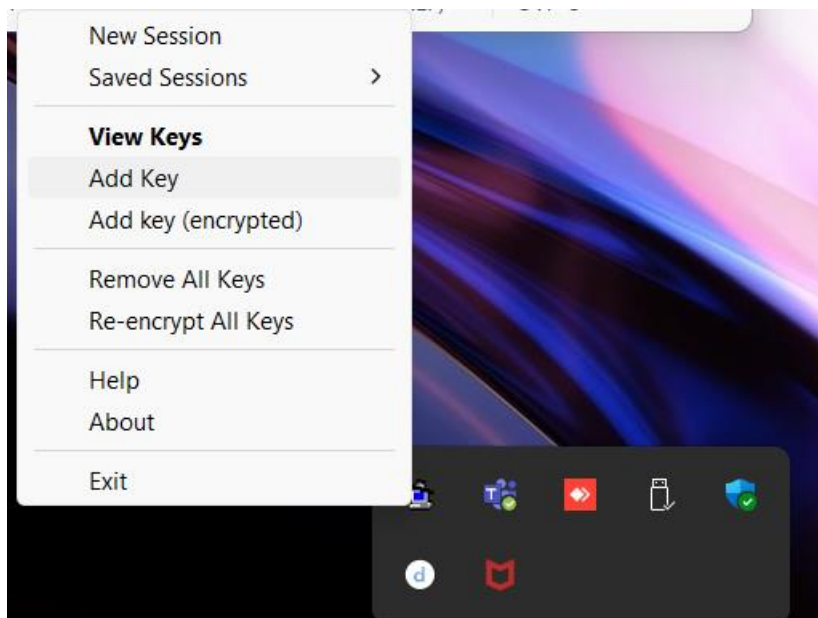
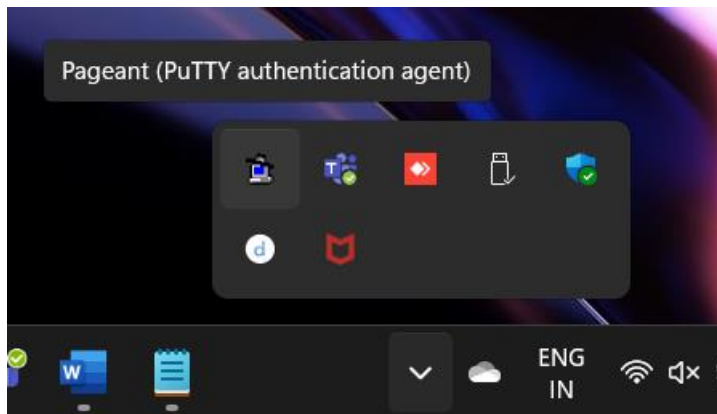
Instances (3) Info									
Find instance by attribute or tag (case-sensitive)									
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	
<input type="checkbox"/>	Redhat	i-007a01b16b807363d	Running	t2.micro	Initializing	0 in alarm	us-east-1b	ec2-18-215-231	
<input type="checkbox"/>	Redhat	i-099368be16cd1169f	Running	t2.micro	Initializing	0 in alarm	us-east-1b	ec2-34-239-48-	
<input type="checkbox"/>	Controller	i-045d66aabd9abee71	Running	t2.micro	2/2 checks passed	0 in alarm	us-east-1b	ec2-54-162-54-	

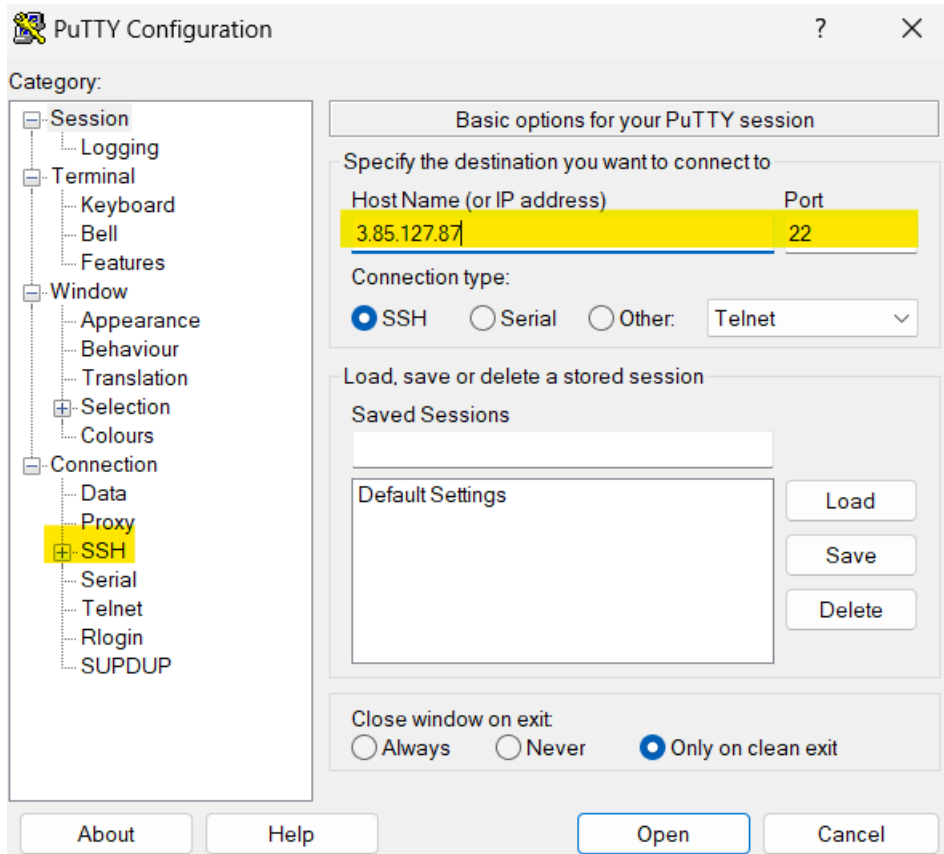
Select an instance

=

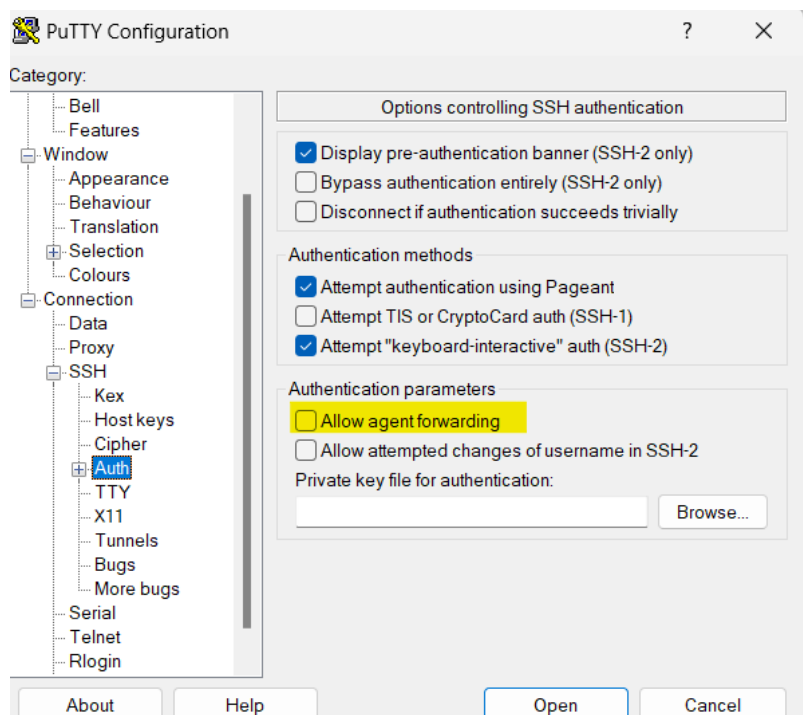
⚙️ ✕

- Download Putty Pageant (putty.org)
- Add ppk Key to Pageant
- Open Putty
- Connect to EC2 (Controller)





- **Login/Connect to Controller EC2 Instance**
- **Click on SSH**
- **Inside SSH click on Auth and allow Agent Forwarding**



- Once Connected Install Ansible

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-22-04>

Command to Install Ansible in ubuntu 22

- `sudo apt-add-repository ppa:ansible/ansible` (This will create Repository for Ansible)
- `sudo apt update` (Updating the operating System)
- `sudo apt install ansible` (This will Install Ansible)

```
ubuntu@ip-172-31-40-196: ~
ubuntu@ip-172-31-40-196:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (8.4.0-lppa~jammy).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-40-196:~$
```

Setting Up the Inventory File

- `sudo nano /etc/ansible/hosts` (This command Displays the Path of Inventory Files)

```
GNU nano 6.2 /etc/ansible/hosts *
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':

## [Debian]
## alpha.example.org
## beta.example.org

## [openSUSE]
## green.example.com
## blue.example.com

[Server1]
172.31.45.71
[Server2]
172.31.47.39
```

- List the Worker EC2 instance Private IP
- To save Ctrl X
- Y to save

- Next Click Enter
- Use the following command to list the inventory
- `ansible-inventory --list -y`

```
ubuntu@ip-172-31-40-196: ~
ubuntu@ip-172-31-40-196:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ansible is already the newest version (8.4.0-1ppa~jammy).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-40-196:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-40-196:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-40-196:~$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-40-196:~$ ansible-inventory --list -y
all:
  children:
    Server1:
      hosts:
        172.31.45.191: {}
    Server2:
      hosts:
        172.31.43.108: {}
ubuntu@ip-172-31-40-196:~$
```

Create Connections

- Use the following command to create connections
- `ssh ec2-user@privateip`

```
ec2-user@ip-172-31-45-191:~
[ec2-user@ip-172-31-45-191 ~]$ ssh ec2-user@172.31.43.108
The authenticity of host '172.31.43.108 (172.31.43.108)' can't be established.
ED25519 key fingerprint is SHA256:goJRZ+cZiwiYz/Tg0kVgjnluW3RcVncTAcl+QyTqLm4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.43.108' (ED25519) to the list of known hosts.
ec2-user@172.31.43.108: Permission denied (publickey,gssapi-keyex,gssapi-with-mi
c).
[ec2-user@ip-172-31-45-191 ~]$
```

Testing Connection

- Use the following command to test the connection Between Controller and Worker EC2 Instance
- `ansible -m ping all - -user=ec2-user`

```
ubuntu@ip-172-31-40-196: ~  
ubuntu@ip-172-31-40-196:~$ ansible -m ping all --user=ec2-user  
172.31.43.108 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
172.31.45.191 | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
ubuntu@ip-172-31-40-196:~$
```

Check the storage on the Worker Ec2 Instance

- **ansible -a "df -h" all - -user=ec2-user**

```
ubuntu@ip-172-31-40-196: ~  
ubuntu@ip-172-31-40-196:~$ ansible -a "df -h" all --user=ec2-user  
172.31.43.108 | CHANGED | rc=0 >>  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        4.0M   0    4.0M   0% /dev  
tmpfs           385M   0    385M   0% /dev/shm  
tmpfs           154M  4.4M   150M   3% /run  
/dev/xvda4       9.4G  1.3G   8.1G  14% /  
/dev/xvda3       495M  153M   343M  31% /boot  
/dev/xvda2       200M   8.0K   200M   1% /boot/efi  
tmpfs            77M    0     77M   0% /run/user/1000  
172.31.45.191 | CHANGED | rc=0 >>  
Filesystem      Size  Used Avail Use% Mounted on  
devtmpfs        4.0M   0    4.0M   0% /dev  
tmpfs           385M   0    385M   0% /dev/shm  
tmpfs           154M  4.4M   150M   3% /run  
/dev/xvda4       9.4G  1.3G   8.1G  14% /  
/dev/xvda3       495M  153M   343M  31% /boot  
/dev/xvda2       200M   8.0K   200M   1% /boot/efi  
tmpfs            77M    0     77M   0% /run/user/1000  
ubuntu@ip-172-31-40-196:~$
```


Check Partition on the Worker Ec2 instance

- `ansible -a "lsblk" all - -user=ec2-user`

```
ubuntu@ip-172-31-40-196:~$ ansible -a "lsblk" all --user=ec2-user
172.31.43.108 | CHANGED | rc=0 >>
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda        202:0    0   10G  0 disk
├─xvda1     202:1    0    1M  0 part
├─xvda2     202:2    0 200M  0 part /boot/efi
├─xvda3     202:3    0 500M  0 part /boot
└─xvda4     202:4    0  9.3G  0 part /
172.31.45.191 | CHANGED | rc=0 >>
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda        202:0    0   10G  0 disk
├─xvda1     202:1    0    1M  0 part
├─xvda2     202:2    0 200M  0 part /boot/efi
├─xvda3     202:3    0 500M  0 part /boot
└─xvda4     202:4    0  9.3G  0 part /
ubuntu@ip-172-31-40-196:~$
```

Check RAM on Worker Ec2 instance

- `ansible -a "free -h" all - -user=ec2-user`

```
ubuntu@ip-172-31-40-196:~$ ansible -a "free -h" all --user=ec2-user
172.31.43.108 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:          768Mi        259Mi        321Mi        5.0Mi        306Mi        509Mi
Swap:           0B           0B           0B
172.31.45.191 | CHANGED | rc=0 >>
              total        used        free      shared  buff/cache   available
Mem:          768Mi        280Mi        295Mi        5.0Mi        311Mi        487Mi
Swap:           0B           0B           0B
ubuntu@ip-172-31-40-196:~$
```

Create group Using Ansible

- **ansible all -m group -a "name=devops state=present" - -user=ec2-user -b**

```
ubuntu@ip-172-31-40-196:~$ ansible all -m group -a "name=devops state=present" -
-user=ec2-user -b
172.31.45.191 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 1001,
  "name": "devops",
  "state": "present",
  "system": false
}
172.31.43.108 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 1001,
  "name": "devops",
  "state": "present",
  "system": false
}
```

Create Users

- **ansible all -m user -a "name=user1 append=yes groups=devops" - -user=ec2-user -b**

```
ubuntu@ip-172-31-40-196:~$ ansible all -m user -a "name=user1 append=yes groups=
devops" --user=ec2-user -b
172.31.45.191 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1002,
  "groups": "devops",
  "home": "/home/user1",
  "name": "user1",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1001
}
```

Ansible Playbook

- Download a Code with a simple language Or Refer with the below mentioned example
- Reference From operating system (ansible -doc)

Ping function

- Create a folder to store the code
- Use nano command to insert the code in the file
- Copy and paste the code (right click to paste on ubuntu)
- Ctrl X to exit
- Y and enter to save

```
ubuntu@ip-172-31-40-196: ~  
GNU nano 6.2 ping.yaml  
--  
- hosts: all  
  user: ec2-user  
  tasks:  
    - name: Ping all hosts  
      ping:  
...  
[ Read 7 lines ]  
^G Help      ^O Write Out ^W Where Is  ^K Cut  
^X Exit      ^R Read File ^\ Replace   ^U Paste
```

- To run the code us (ansible-playbook ping.yaml)

```
ubuntu@ip-172-31-40-196:~$ ansible-playbook ping.yaml

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [172.31.43.108]
ok: [172.31.45.191]

TASK [Ping all hosts] *****
ok: [172.31.43.108]
ok: [172.31.45.191]

PLAY RECAP *****
172.31.43.108      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.45.191    : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-40-196:~$
```

Refer the below Diagram for Ansible

