

# Experiment-11

## *Ansible Installation*

1. Create 2 AWS EC2 Instance with same key-pair.

**Description**  
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01

Architecture: 64-bit (x86) AMI ID: ami-0b8b44ec9a8f90422 Verified provider

**▼ Instance type** [Info](#) | [Get advice](#)

Instance type: t2.micro  
Family: t2 1 vCPU 1 GiB Memory Current generation: true Free tier eligible  
On-Demand Linux base pricing: 0.0116 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand RHEL base pricing: 0.0716 USD per Hour

**▼ 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: Ansiblekey Create new key pair

**▼ Summary**

Number of instances: 2 [Info](#)

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-0b8b44ec9a8f90422

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per ...

Cancel Launch instance [Review commands](#)

1. Install Ansible in one of the Instance.

```
52 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-44-171:~$ sudo apt-add repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible
```

```
Selecting previously unselected package python3-ntlm-auth.
Preparing to unpack .../12-python3-ntlm-auth_1.4.0-1_all.deb ...
Unpacking python3-ntlm-auth (1.4.0-1) ...
Selecting previously unselected package python3-requests-kerberos.
Preparing to unpack .../13-python3-requests-kerberos_0.12.0-2_all.deb ...
Unpacking python3-requests-kerberos (0.12.0-2) ...
(#####.....)
```

2. Verify installation by checking version.

```
ubuntu@ip-172-31-44-171:~$ ansible --version
ansible [core 2.16.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
ubuntu@ip-172-31-44-171:~$
```

### 3. Make a directory and using scp copy key-pair PEM file.

```
PS C:\Users\pulki\Downloads> scp -i "Ansiblekey.pem" Ansiblekey.pem ubuntu@ec2-3-144-76-19.us-east-2.compute.amazonaws.com:/home/ubuntu/keys
The authenticity of host 'ec2-3-144-76-19.us-east-2.compute.amazonaws.com (3.144.76.19)' can't be established.
ED25519 key fingerprint is SHA256:A8HYIniES1/6HhmwGB5gP7VoXZOMckamevfnXe88PA4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added 'ec2-3-144-76-19.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Ansiblekey.pem                               100% 1674    2.4KB/s   00:00
PS C:\Users\pulki\Downloads>
```

### 3. Edit **ansible host** file and add the public IP of another server.

```
GNU nano 6.2                                     hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers:
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
#
# Ex 2: A collection of hosts belonging to the 'webserver' group:
[webserver]
server1 ansible_host=18.217.29.22
[webserver:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansiblekey.pem
```

### 4. Now check connection by **ansible webserver -m ping**

```
ubuntu@ip-172-31-44-171:/etc/ansible$ chmod 0600 /home/ubuntu/keys/Ansiblekey.pem
ubuntu@ip-172-31-44-171:/etc/ansible$ ansible webserver -m ping
server1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-44-171:/etc/ansible$
```

# Experiment-12

## *Ansible Playbook*

1. Create a **nginxplay.yml** file.

```
GNU nano 6.2 nginxplay.yml
--
name: Install and start Nginx
hosts: webserver
become: yes
tasks:
  - name: Install Nginx
    apt:
      name: nginx
      state: latest

  - name: Start Nginx
    service:
      name: nginx
      state: started
      enabled: yes
```

2. Execute this playbook by **ansible-playbook nginxplay.yml**

```
ubuntu@ip-172-31-44-171:~$ ansible-playbook nginxplay.yml

PLAY [Install and start Nginx] *****

TASK [Gathering Facts] *****
ok: [server1]

TASK [Install Nginx] *****
ok: [server1]

TASK [Start Nginx] *****
ok: [server1]

PLAY RECAP *****
server1 : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

ubuntu@ip-172-31-44-171:~$
```

3. To verify open the IP of another server and **Nginx** webpage will be up.

