# Hands-On Lab (HOL): Ansible Setup on Amazon Linux 2 & Ubuntu

---

## Lab Objective

In this HOL, we will **set up Ansible step by step** on:

- **Amazon Linux 2** (Master node only)
- **Ubuntu** (Master node)

You will clearly understand:

- Why each command is used
- How Ansible communicates with managed nodes
- How passwordless (key-based) SSH works

This lab is written in a **trainer-style**, beginner-friendly way.

---

## Lab Architecture (Common for Both OS)

- Total Machines: **3 EC2 / VMs**
- 1 × Ansible Master
- 2 × Managed Nodes
- All machines **must communicate with each other** (same VPC / network)
- Use **private IPs** for Ansible inventory

---

# PART 1: Ansible Setup on Amazon Linux 2

> ⚠️**All Ansible installation work is done ONLY on the Master node**

---

## Step 1: Enable EPEL Repository

```
sudo amazon-linux-extras install epel -y
```

**What this does:**

- Enables the **EPEL (Extra Packages for Enterprise Linux)** repo

- Ansible packages are available via EPEL

---

## Step 2: Update the System

```
sudo yum update -y
```

**Why:**

- Brings all packages to the latest stable version
- Avoids dependency issues

---

## Step 3: Install Required Packages

```
yum install git python python-level python-pip -y
```

**Explanation:**

- `git` → used for pulling Ansible roles/playbooks
- `python` → Ansible is Python-based
- `python-pip` → Python package manager

---

## Step 4: Install Ansible

```
yum install ansible -y
```

**What happens:**

- Installs Ansible engine and CLI

---

## Step 5: Install OpenSSL

```
yum install openssl -y
```

**Why:**

- Required for secure SSH communication

## Step 6: Verify Installation

```
ansible --version
```

**Confirms:**

- Ansible is installed correctly

## Step 7: Configure Inventory File

```
vi /etc/ansible/hosts
```

Add managed node IPs:

```
[dev]
172.31.44.103 ansible_python_interpreter=/usr/bin/python3
172.31.38.223 ansible_python_interpreter=/usr/bin/python3
```

**Explanation:**

- `[dev]` → inventory group name
- `ansible_python_interpreter` → tells Ansible which Python to use

## Step 8: Configure Ansible Defaults

```
vi /etc/ansible/ansible.cfg
```

Uncomment and configure:

- inventory
- sudo user

**Why:**

- Makes Ansible execution smoother without extra flags

## Step 9: Create Ansible User (All Machines)

```
adduser ansible
passwd ansible
```

**Why:**

- Best practice: don’t use root directly

---

## Step 10: Grant Sudo Access

```
visudo
```

Add:

```
ansible ALL=(ALL) NOPASSWD: ALL
```

**Meaning:**

- `NOPASSWD` allows Ansible automation without password prompts

---

## Step 11: SSH Configuration (All Machines)

```
vi /etc/ssh/sshd_config
```

Make these changes:

```
PermitRootLogin yes
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

Restart SSH:

```
service sshd restart
```

**Why:**

- Enables password-based SSH temporarily
- Required for key exchange

---

### Step 12: Passwordless SSH Setup

(Login as `ansible` user on **Master only**)

```
ssh-keygen
cd ~/.ssh
ssh-copy-id ansible@172.31.38.223
ssh-copy-id ansible@172.31.44.103
```

**Explanation:**

- Generates SSH key pair
- Copies public key to managed nodes
- Enables **passwordless authentication**

---

## PART 2: Ansible Setup on Ubuntu

---

### Step 1: Update System (All Machines)

```
sudo -i
apt update -y
```

**Why:**

- Ensures clean package installation

---

### Step 2: Set Root Password (All Machines)

```
sudo passwd root
```

**Why:**

- Needed if you plan to use root login

## Step 3: Install Ansible (Master Node Only)

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y software-properties-common apt-transport-https ca-certificates curl
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt update
sudo apt install -y ansible
```

**Explanation:**

- Adds official Ansible PPA
- Installs latest stable Ansible

## Step 4: Verify Installation

```
ansible --version
```

## Step 5: Inventory Configuration

```
vi /etc/ansible/hosts

Note : In this filw add managed node private IP
172.31.44.103 and 172.31.38.223 in my managed node private ip
```

```
[dev]
172.31.44.103 ansible_python_interpreter=/usr/bin/python3
172.31.38.223 ansible_python_interpreter=/usr/bin/python3
```

## Step 6: Configure Ansible Defaults

```
vi /etc/ansible/ansible.cfg
```

Paste:

```
[defaults]
# Path to your inventory (hosts) file
inventory = hosts

# Default user to connect to managed nodes
remote_user = root

# Disable host key checking (useful for lab/testing)
host_key_checking = False

# Enable privilege escalation (sudo)
become = True
become_method = sudo
become_ask_pass = False   # Set to True if you want Ansible to prompt for sudo password

# Reduce the SSH timeout waiting period
timeout = 30

[persistent_connection]
connect_timeout = 30
command_timeout = 30

[privilege_escalation]
become=True
become_method=sudo
become_user=root
```

**Why:**

- Avoids repetitive flags
- Enables privilege escalation

---

## Step 7: SSH Configuration (All Machines)

```
vi /etc/ssh/sshd_config
```

```
PermitRootLogin yes
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```

Restart:

```
service sshd restart
```

---

## Step 8: Passwordless SSH (Master Node)

```
ssh-keygen
cd ~/.ssh
ssh-copy-id root@172.31.38.223
ssh-copy-id root@172.31.44.103
```

---

## Final Validation

```
ansible dev -m ping
```

**Expected Output:**

```
SUCCESS => pong
```

---

## Lab Outcome

✔️Ansible installed ✔️Inventory configured ✔️SSH trust established ✔️Ready for Playbooks

---