Ansible Master-Slave Setup Guide on AWS

This guide walks you through setting up Ansible on AWS EC2 instances with a Master-Slave SSH configuration. Follow each step carefully. Commands assume Ubuntu on both instances and that you have SSH access.

Prerequisites

- 2 AWS EC2 instances running Ubuntu (one Master, one Slave).
- Hostnames changed to Master and Slave (use sudo hostname <name> to set).
- You can SSH into both instances and have sudo/root access.
- Connectivity between Master and Slave (security group allowing SSH from Master to Slave).

High-level Steps

1. Create and SSH into two EC2 instances (Master and Slave).
2. Install Ansible on the Master.
3. Generate SSH keys on Master and copy public key to Slave authorized_keys.
4. Configure SSH on Slave to allow root login via key and enable PubkeyAuthentication.
5. Test Ansible connectivity and run example modules to install/remove packages.

Master Node Setup

Switch to root and update packages:

```
sudo su

sudo apt update -y
```

Add Ansible PPA and install Ansible:

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

apt-get install ansible -y
```

Verify installation:

```
ansible --version
```

Edit the Ansible inventory to add the Slave IP:

```
nano /etc/ansible/hosts
```

Add at the end of the file:

```
[client_1]
<IP_Address_of_Slave>
```

(for exit from nano file ctrl+x -> Y -> enter)

Create SSH keypair on Master (if not already created):

```
ssh-keygen -t rsa
```

Switch to root and list .ssh:

```
cd /root
cd .ssh/
ls
```

Slave Node Configuration

Switch to root on Slave and open SSH folder:

```
sudo su

cd /root/.ssh/
ls
```

Open authorized_keys and paste the remove all and paste Master public key (id_rsa.pub):

```
nano authorized_keys
```

- Paste the full content of id_rsa.pub from Master into this file and save.

(for exit from nano file ctrl+x -> Y -> enter)

Edit SSH daemon configuration to allow root login and pubkey auth:

```
nano /etc/ssh/sshd_config
```

Ensure the following lines are set:

```
PermitRootLogin yes
```

(for exit from nano file ctrl+x -> Y -> enter)

After editing, restart SSH:

```
systemctl restart sshd || systemctl restart ssh
```

Back to Master — Test Ansible connectivity

From Master, test ping via Ansible:

```
ansible -m ping all
```

{

(optional if error found)

If host key prompt appears when using ssh, accept it once:

```
ssh root@<IP_Address_of_Slave>
```

When prompted 'Are you sure you want to continue connecting (yes/no)?' type 'yes' and press Enter.

```
exit

ansible -m ping all
```

}

```
ansible client_1 -m setup
```

ON Master Node

Step 1: Clone Ansible Playbook Repository

```
cd ~

mkdir ansible-lab

cd ansible-lab/

git clone https://github.com/sujataoak799/ansible-codes.git

cd ansible-codes/
```

ls

(Tip: change the index.html file according to your name…etc)

You should see files like: lampstack_1.yml, index.html, config.php, etc.

Step 2: Review/Edit LAMP Stack Playbook

ansible-playbook lampstack_1.yml

on Slave Node

STEP 3: Verify Installation on Slave Node

mysql

php –version

service apache2 status

STEP 4: Access Web Application

1. **Open AWS EC2 Console**
2. **Copy the Public IPv4 address of Slave instance**
3. **Paste it in your browser: http://<Slave-Public-IP>**
4. **You should see the deployed web page!**