**Project Title: Establishing Passwordless SSH Authentication to AWS EC2 Instances from a Local Server**

**Project Overview:**

In this project, I set up passwordless SSH authentication from my local server (laptop) to two identical Amazon EC2 instances. The goal was to configure these EC2 instances for seamless Ansible management. The process involved two methods: using an SSH key and using a password.

**Project Steps:**

1. **Passwordless Authentication Using SSH Key:**
   * **Connecting to EC2 Instance:**
     + Used the following command to connect to the EC2 instance and copy the SSH key:

sh

Copy code

ssh-copy-id -f "-o IdentityFile=keypair.pem" ubuntu@public-ip-address

* + - The above command asked for first-time connection verification with:

sh

Copy code

ssh -o 'IdentityFile=keypair.pem' 'ubuntu@public-ip-address'

* + **Restarting SSH Service:**
    - Restarted the SSH service on the EC2 instance to apply changes:

sh

Copy code

sudo systemctl restart ssh

* + **Verifying Passwordless Authentication:**
    - Confirmed passwordless authentication by executing:

sh

Copy code

ssh ubuntu@public-ip-address

1. **Passwordless Authentication Using Password:**
   * **Initial Connection:**
     + Connected to the EC2 instance using:

sh

Copy code

ssh -i keypair.pem ubuntu@public-ip-address

* + **Editing SSH Configuration:**
    - Opened the cloud-init script to enable password authentication:

sh

Copy code

sudo vim /etc/ssh/sshd\_config.d/60-cloudimg-settings.conf

* + - Modified the line **PasswordAuthentication no** to **PasswordAuthentication yes**.
  + **Alternative Method:**
    - Edited the main SSH configuration file:

sh

Copy code

sudo vim /etc/ssh/sshd\_config

* + - Uncommented the line **#PasswordAuthentication yes** to **PasswordAuthentication yes**.
  + **Restarting SSH Service:**
    - Restarted the SSH service to apply changes:

sh

Copy code

sudo systemctl restart ssh

* + **Verifying Passwordless Authentication:**
    - Confirmed passwordless authentication by executing:

sh

Copy code

ssh ubuntu@public-ip-address

**Conclusion:**

Through this project, I successfully established passwordless SSH authentication from my local server to two Amazon EC2 instances using two different methods. This setup is essential for seamless Ansible configuration management, ensuring efficient and secure server interactions.

**Commands Used:**

1. **Passwordless Authentication Using SSH Key:**

sh

Copy code

ssh-copy-id -f "-o IdentityFile=keypair.pem" ubuntu@public-ip-address ssh -o 'IdentityFile=keypair.pem' 'ubuntu@public-ip-address' sudo systemctl restart ssh ssh ubuntu@public-ip-address

1. **Passwordless Authentication Using Password:**

sh

Copy code

ssh -i keypair.pem ubuntu@public-ip-address sudo vim /etc/ssh/sshd\_config.d/60-cloudimg-settings.conf sudo vim /etc/ssh/sshd\_config sudo systemctl restart ssh ssh ubuntu@public-ip-address

**Errors and Solutions:**

1. **Error: Incorrect SSH Key Path**
   * **Description:** Encountered an error due to incorrect path or name of the SSH key.
   * **Solution:** Ensured the correct path and file name were used in the command.
2. **Error: SSH Service Restart Issues**
   * **Description:** SSH service did not restart correctly after changes.
   * **Solution:** Verified changes in the configuration files and restarted the SSH service using **systemctl** commands.

**Difficulties Faced:**

1. **SSH Key Configuration:**
   * Ensuring the SSH key was correctly copied to the EC2 instances.
   * Solution: Followed precise steps for **ssh-copy-id** and verified with manual SSH login.
2. **Editing Configuration Files:**
   * Correctly modifying the SSH configuration files without syntax errors.
   * Solution: Carefully edited the files using **vim** and double-checked changes.
3. **SSH Service Management:**
   * Restarting the SSH service to apply new configurations.
   * Solution: Used **systemctl restart ssh** and verified the service status.

This project effectively demonstrates the setup of passwordless SSH authentication, essential for smooth Ansible operations across multiple EC2 instances from a local server.