Lab Setup Instructions

Requirements

- Virtual Machine with Linux OS

- SSH key file

- Anaconda installer - Terminal with SSH client

- Web browser for accessing Jupyter Notebook

Steps:

To ensure secure access to the VM, the SSH key file permissions are set to read-only for the owner.

```
chmod 400 ravivm_key.pem
```

Connecting to the Virtual Machine

Access the VM using SSH with the provided key and user credentials.

```
ssh -i ravivm_key.pem azureuser@172.191.241.245
```

Installing Anaconda**

Ensure the home directory is the working directory, then download and install the Anaconda distribution.

```
cd $HOME

wget https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh

bash Anaconda3-2022.05-Linux-x86_64.sh
```

Follow the installer prompts to complete the installation.

Configuring Anaconda Environment**

After installation, Anaconda may not be added to the PATH. Activate it by restarting the shell or sourcing the appropriate configuration.

```
source ~/.bashrc
```

Alternatively, activate the Anaconda environment manually:

```
source ~/anaconda3/bin/activate
```

Verify the installation by checking the paths for python and jupyter:

```
which python
which jupyter
```

Expected outputs:

```
/home/azureuser/anaconda3/bin/python
/home/azureuser/anaconda3/bin/jupyter
```

3.2.5. Launching Jupyter Notebook

Start the Jupyter Notebook server to allow remote access.

```
jupyter notebook --ip=0.0.0.0 --no-browser --port=8888
```

The server will provide a URL with a token, e.g., http://172.191.241.245:8888/?token=abc123. Copy this URL for browser access.

Results

Upon successful execution of the above steps:

- The VM is accessible via SSH with secure key-based authentication.

- Anaconda is installed and added to the PATH, with Python and Jupyter correctly configured.

- The Jupyter Notebook server is running and accessible via the provided URL.

The setup process ensures a robust environment for Python-based computational tasks. Key considerations include:

- **Security**: The SSH key permissions (chmod 400) prevent unauthorized access.

- **Environment Isolation**: Anaconda provides a self-contained environment, avoiding conflicts with system Python installations.

- **Remote Access**: Configuring Jupyter with --ip=0.0.0.0 allows access from any network, though additional firewall rules may be needed for security.

Potential issues include network restrictions blocking port 8888 or incorrect PATH configurations. These can be resolved by checking firewall settings or reinstalling Anaconda.

Conclusion

The procedure successfully configures a virtual machine for Python development using Anaconda and Jupyter Notebook. This setup is suitable for data science, machine learning, or other computational experiments requiring a scalable and reproducible environment.

References

- Anaconda Documentation: https://docs.anaconda.com/

- Jupyter Notebook Documentation: https://jupyter.org/