# Lab Instructions: Automating Local and Remote System Queries with Fabric

## Step 1: Prerequisites

1. Install Python:

- Ensure Python (3.6 or later) is installed on your system.

- Verify the installation:

python3 --version

2. Install Fabric:

- Fabric is required for running the fab tasks. Install it using pip:

pip install fabric

3. Create a Virtual Environment:

- Use a Python virtual environment to isolate your Fabric project.

python3 -m venv venv

- Activate the virtual environment:

source venv/bin/activate # For Linux/macOS

.\venv\Scripts\activate # For Windows

- Install Fabric within the virtual environment:

pip install fabric

## Step 2: Verify fabfile.py Existence

1. Check for the Fabfile:

- Ensure that a file named fabfile.py exists in your working directory:

ls fabfile.py

2. Create a Fabfile if It Doesn't Exist:

- If the file doesn't exist, create fabfile.py with the following content:

from fabric import task, Connection  
import subprocess  
  
REMOTE\_HOST = "192.168.1.166"  
USERNAME = "rps"  
PASSWORD = "rps@123"  
  
@task  
def query\_local\_and\_remote\_info(c):  
 """Query system information for both local and remote hosts."""  
 try:  
 print("\n--- Local System Information ---\n")  
 local\_commands = [  
 ("Hostname", "hostname"),  
 ("Network Configuration", "ifconfig"),  
 ("System Uptime", "uptime"),  
 ("Current Users", "who"),  
 ("Memory Usage", "free -h"),  
 ("Disk Usage", "df -h"),  
 ]  
 for desc, command in local\_commands:  
 print(f"> {desc}:")  
 result = subprocess.run(command, shell=True, capture\_output=True, text=True)  
 if result.returncode == 0:  
 print(result.stdout.strip(), "\n")  
 else:  
 print(f"Error executing '{command}': {result.stderr.strip()}\n")  
 except Exception as e:  
 print(f"Error obtaining local system information: {e}")  
  
 try:  
 print("\n--- Remote System Information ---\n")  
 conn = c.Connection if hasattr(c, 'connection') else Connection(  
 host=REMOTE\_HOST,  
 user=USERNAME,  
 connect\_kwargs={"password": PASSWORD},  
 )  
 conn.open()  
 remote\_commands = [  
 ("Hostname", "hostname"),  
 ("Network Configuration", "ifconfig"),  
 ("System Uptime", "uptime"),  
 ("Current Users", "who"),  
 ("Memory Usage", "free -h"),  
 ("Disk Usage", "df -h"),  
 ]  
 for desc, command in remote\_commands:  
 print(f"> {desc}:")  
 result = conn.run(command, hide=True)  
 print(result.stdout.strip(), "\n")  
 conn.close()  
 except Exception as e:  
 print(f"Error obtaining remote system information: {e}")

## Step 3: List Tasks in Fabfile

1. List Tasks:

- Use the fab command to list available tasks in the fabfile.py:

fab --list

2. Expected Output:

- You should see the following task listed:

query\_local\_and\_remote\_info Query system information for both local and remote hosts.

## Step 4: Execute Tasks

1. Run the Task:

- Execute the query\_local\_and\_remote\_info task:

fab query\_local\_and\_remote\_info

## Step 5: Verify Virtual Environment

1. Check Active Virtual Environment:

- Ensure the virtual environment is active. The prompt should include (venv).

2. Deactivate When Done:

- Deactivate the virtual environment to exit:

deactivate