This appendix outlines the steps necessary to set up and operate Guardian for QKD purposes. For detail can refer to https://github.com/s-fifteen-instruments/guardian/blob/2node_networksep/info/QuickStart.md.

Step 1: Download and Set Up Two Ubuntu Virtual Machines

1. Donload Ubuntu ISO
    - Visit the Ubuntu webside (https://ubuntu.com/download/desktop) and download the ISO file for the latest Ubuntu LTS version.
2. Set Up Virtual Machines
    - Use a virtualization platform like VirtualBox, VMware, or KVM to create two VMs.
    - Allocate at least:
        - 1 CPUs and 4 GB of RAM per VM.
        - 20 GB of disk space.
        - Set the Network Adapter to NAT for internet access.
    - Attach the Ubuntu ISO to the VM as a bootable disk.
3. Install Ubuntu
    - Boot the VMs and follow the Ubuntu installation process:
        - Configure a hosname (e.g., qkde0001 and qkde0002).
        - Set up a user account for administration.
4. Install Required Tools and Dependencies
   After setting up the Ubuntu VMs, install the following tools to prepare the environment for Guardian:
    - Update system packages:
        - sudo apt update && sudo apt upgrade -y
    - OpenSSH Server for secure remote access:
        - sudo apt install openssh-server -y
        - sudo systemctl enable ssh
        - sudo systemctl start ssh
    - Networking Tools for managing network interfaces:
        - sudo apt install net-tools -y
    - Git for cloning the Guardian repository:
        - sudo apt install git -y
    - Make for compiling dependencies if needed:
        - sudo apt install make -y
    - Docker for containerized environments:
        - sudo snap install docker
    - Vault by HashiCorp for managing certificates and secrets:
        - sudo snap install vault
    - Python and Pip (if not pre-installed) for running Python-based scripts:
        - sudo apt install python3 python3-pip -y
    - OpenSSL for generating certificates:
        - sudo apt install openssl -y
5. Edit the /etc/hosts File
   After installing dependencies, determine the IP address of each VM and edit the /etc/hosts file:

- Use ifconfig (installed as part of net-tools) to find the IP address of the VM:
  - Ifconfig
  - Look for the inet address under the appropriate network interface (e.g., eth0).
- Edit the /etc/hosts file to map the IP addresses to the hostnames:
  - sudo nano /etc/hosts
- Add the following lines, replacing the IP addresses with those of your VMs:
  - 192.168.24.139 qkde0001.internal
    192.168.24.140 qkde0002.internal
    192.168.24.139 qkde0001.public
    192.168.24.140 qkde0002.public
- Save and exit the editor (CTRL+O, ENTER, CTRL+X for nano).
-

Step 2: Clone the Guardian Repository

1. Clone the Guardian Repository
   - On each VM, clone the Guardian repository with the 2node_networksep branch:
     - git clone -b 2node_networksep https://github.com/s-fifteen-instruments/guardian.git
   - Navigate to the repository directory:
     - cd guardian
     -

Step 3: Configure SSH Access Between VMs

1. Generate SSH Keys
   On each VM, generate an RSA key pair for secure, passwordless access:
   - ssh-keygen -t rsa -b 4096
2. Copy the SSH Key to the Other VM
   - Use the ssh-copy-id command to share the key with the other VM:
     - ssh-copy-id <username>@<IP_of_other_VM>
     - Replace <username> with your VM's user account and <IP_of_other_VM> with the other VM's IP address.
   - Test the connection by logging into the other VM:
     - ssh <username>@<IP_of_other_VM>

Step 4: Configure SSH Access Between VMs

1. Set Required Permissions for Vault
   - cd guardian
2. Run the script to initialize permissions for Vault:
   - sudo ./scripts/init_permissions.sh ${CURR_BRANCH_NAME}
   - Replace ${CURR_BRANCH_NAME} with the branch name you are using (e.g., 2node_networksep).

Step 5: Generate Configuration Files

1. Run the following command to generate initial configuration files:
   - sudo make generate_config

- This will create default configuration files. You can modify these files as needed to customize your setup.

Step 6: Initialize the Test Environment

1. Run the initialization command on both VMs to generate certificates and configuration:
   - make init_test

Step 7: Edit the Makefile

1. Open the Makefile for editing:
   - sudo nano Makefile
2. Edit the Following Sections:
   - For qkde0001 (Master):
     - # Passwordless SSH access must be set up to the remote directory.
       export LOCAL_KME_ADDRESS ?= qkde0001.public
       export REMOTE_KME_ADDRESS ?= qkde0002.public
       export LOCAL_KME_ADD_SSH ?= qkde0001.internal
       export REMOTE_KME_ADD_SSH ?= qkde0002.internal
       export REMOTE_KME_DIR_SSH ?=
       qitlab@$(REMOTE_KME_ADD_SSH):~/guardian

       # Identity strings for QKDE and KME, with an initial local SAE bootstrapped, swap at remote
       export LOCAL_QKDE_ID ?= QKDE0001
       export REMOTE_QKDE_ID ?= QKDE0002
       export LOCAL_KME_ID ?= KME-S15-Guardian-001-Guardian
       export REMOTE_KME_ID ?= KME-S15-Guardian-002-Guardian
       export LOCAL_SAE_ID ?= SAE-S15-Test-001-sae1
   - For qkde0002 (Slave):
     - # Passwordless SSH access must be set up to the remote directory.
       export LOCAL_KME_ADDRESS ?= qkde0002.public
       export REMOTE_KME_ADDRESS ?= qkde0001.public
       export LOCAL_KME_ADD_SSH ?= qkde0002.internal
       export REMOTE_KME_ADD_SSH ?= qkde0001.internal
       export REMOTE_KME_DIR_SSH ?=
       qitlab@$(REMOTE_KME_ADD_SSH):~/guardian

       # Identity strings for QKDE and KME, with an initial local SAE bootstrapped, swap at remote
       export LOCAL_QKDE_ID ?= QKDE0002
       export REMOTE_QKDE_ID ?= QKDE0001
       export LOCAL_KME_ID ?= KME-S15-Guardian-002-Guardian
       export REMOTE_KME_ID ?= KME-S15-Guardian-001-Guardian
       export LOCAL_SAE_ID ?= SAE-S15-Test-002-sae2
   - Save and exit the editor (CTRL+O, ENTER, CTRL+X for nano).

Step 8: Initialize the Guardian Setup

1. On Both VMs (qkde0001 and qkde0002), run the following command to initialize the Guardian setup:
    - sudo make init
    - This will set up the necessary directories, permissions, and configurations for each VM.

Step 9: Establish Connection Between VMs

1. On the Master VM (qkde0001), wait for the remote VM (qkde0002) to complete its initialization.
2. Once the remote VM (qkde0002) is ready, run the following command on the master:
    - make connect
    - This establishes a secure connection between the local and remote KMEs.

Step 10: Running the REST Service and Key Generation

1. Start the REST Service:
    - sudo make rest
2. Verify the Docker Containers:
    - sudo docker ps -a
3. Monitor the Key Generation Process:
    - View the logs of the guardian-qkdsim-1 container to ensure keys are being generated:
        - sudo docker logs -f guardian-qkdsim-1
4. Monitor the Key Ingestion Process:
    - Check the logs of the guardian-watcher-1 container to confirm that generated keys are being ingested:
        - sudo docker logs -f guardian-watcher-1

Step 11: Stopping and Cleaning Up the Services

1. Stop Key Generation
    - Stop the guardian-qkdsim-1 Container:
        - sudo docker stop guardian-qkdsim-1
2. Clean Up the KME Environment
    - Tear down the KME setup:
        - sudo make clean