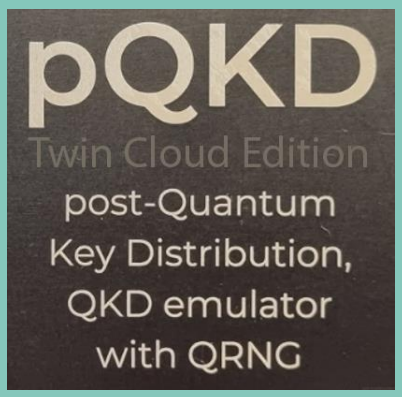
pQKD Twin Cloud Edition



Quantum-Enhanced VPN Connectivity for AWS Cloud.
Delivering ETSI QKD Compatibility,
Genuine Quantum Entropy,
and Post-Quantum Security.

Created by:
Ryszard Olejnik,
Piotr Łuniewski,
and Mirek Sopek

Quantum Blockchains, Inc.

Table of Contents

pQKD Twin Cloud Edition

I. Solution presentation

pQKD Twin Cloud Edition is a solution for high-security connectivity between an AWS VPC (Virtual Private Cloud) and a client network, based on the principles of quantum cryptography realized through QKD emulation.
It follows standard VPN principles, enhanced with secure symmetric key exchange provided by QKD emulation technology. This technology ensures full ETSI QKD compatibility, genuine quantum entropy from a quantum random number generator, and uses a standard post-quantum key encapsulation mechanism (FIPS 203 - Module-Lattice-Based Key-Encapsulation Mechanism Standard known also as CRYSTALS Kyber).

The solution consists of the following elements:

1. An EC2 server instance on AWS (with the necessary services installed)
2. A client computer (with the necessary services installed)
3. A pQKD device.

The solution schema is presented below:

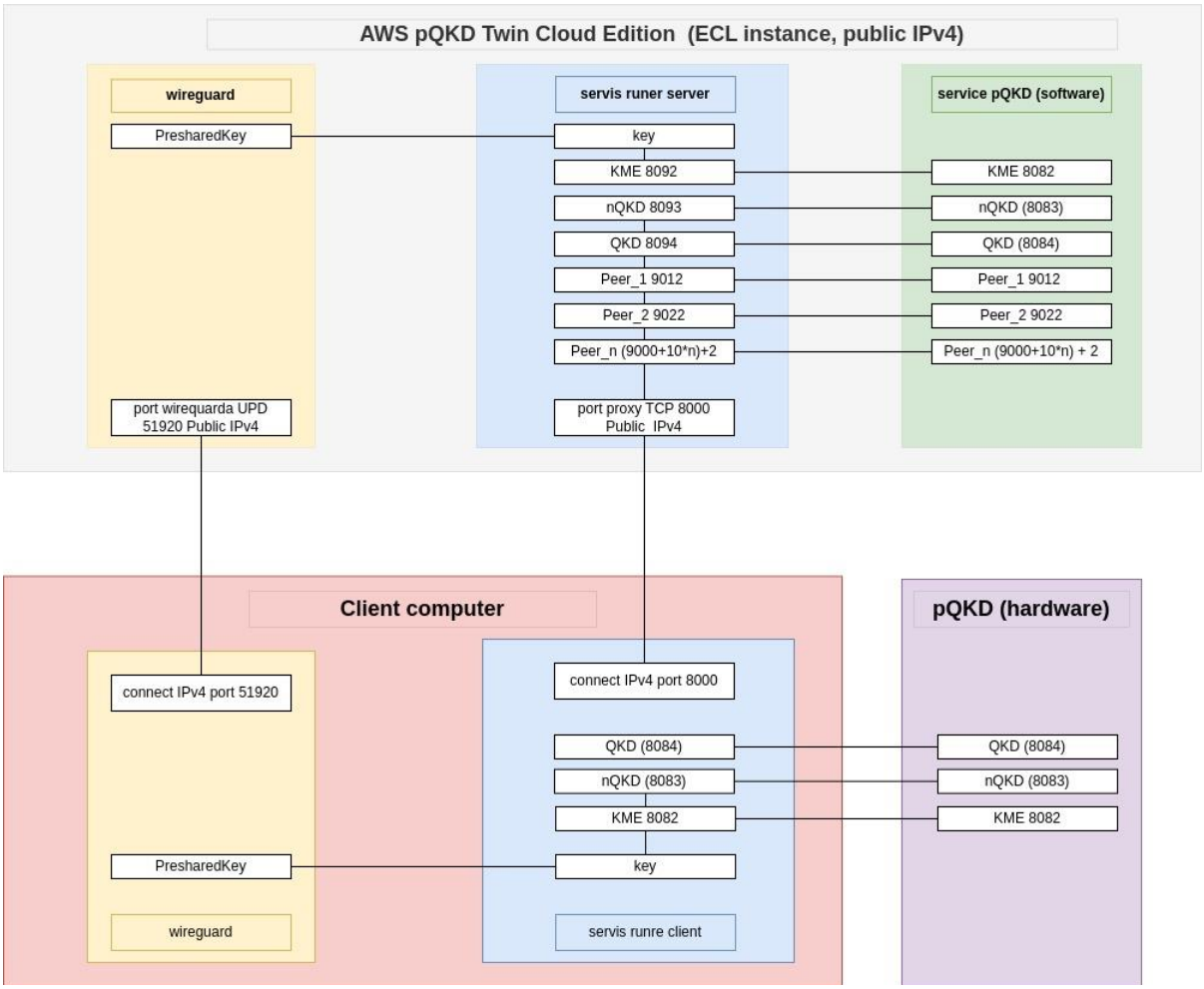


Figure 1 pQKD Twin Cloud Edition schematics

As shown in Figure 1, the VPN network is based on the efficient and secure WireGuard VPN. In our solution, we do not modify the essential security components of the VPN. WireGuard establishes a UDP connection (on port 51920), creating a new virtual network interface in the system. Authentication and encryption key exchange are still performed using the RSA algorithm. However, for maximum security, the transmitted standard key is encrypted with a presharedKey—identical on both the server and client sides—via an XOR operation, typical for One-Time Pad (OTP) mechanisms.

pQKD Twin (on the cloud side) and the pQKD device (on the client network/computer side) provide mechanisms for distributing the presharedKey.

Consequently, our solution is a hybrid approach, combining the WireGuard system with a post-quantum key exchange based on quantum entropy. The service requests key generation by connecting to the pQKD service on AWS via the KME (Key Management Entity) port.

The pQKD service communicates through a TCP link (port 8000) with the client service and its pQKD device (where the key is generated). The keys obtained on both the AWS server and client sides are then incorporated into the WireGuard VPN on each end.

On the AWS side, there is an EC2 instance with the following services installed:
1. WireGuard
2. A runner service for communication with WireGuard, pQKD, and the client
3. A software-based implementation of pQKD

This server configuration has been saved as an AMI image on AWS.

On the client side, the following components are present:
1. WireGuard
2. A service for communication with WireGuard, pQKD, and the AWS server
3. A pQKD device connected to the client computer

II. Installation of pQKD Twin on AWS

We select the image (Images → AMIs) named: “pQKD Twin_Cloud_Edition.”

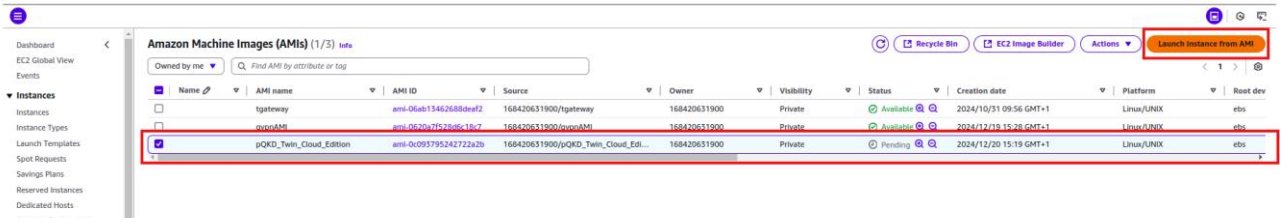


Figure 2

After choosing the right image, click the “Launch Instance from AMI” button:

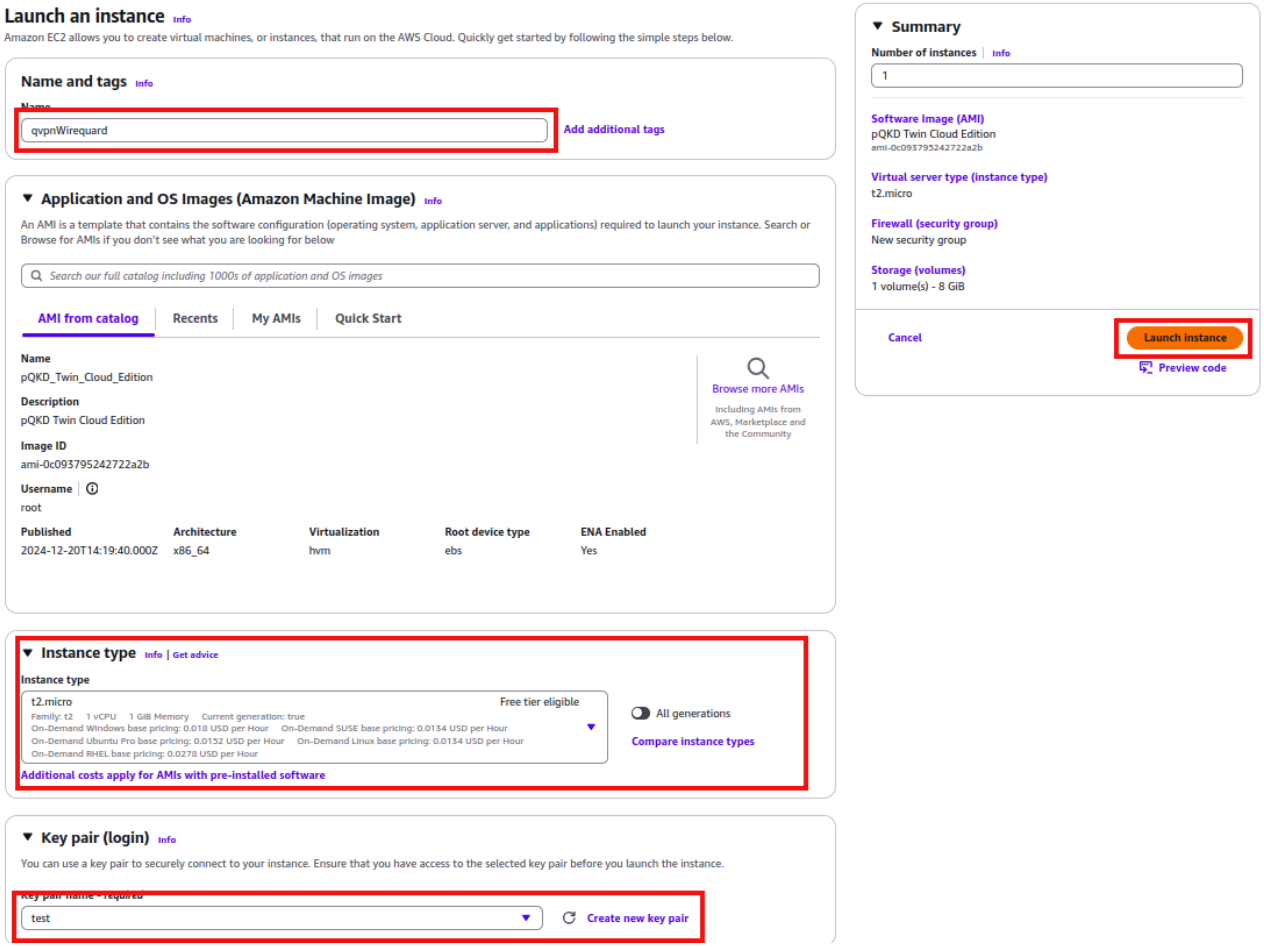


Figure 3

Then fill in the fields highlighted in red as follows:

1. Instance name
2. Instance type (vCPUs, CPU, disk size)
3. Select or generate the key for SSH terminal access

Next, click the “Launch Instance” button.

You should then see the following screen:

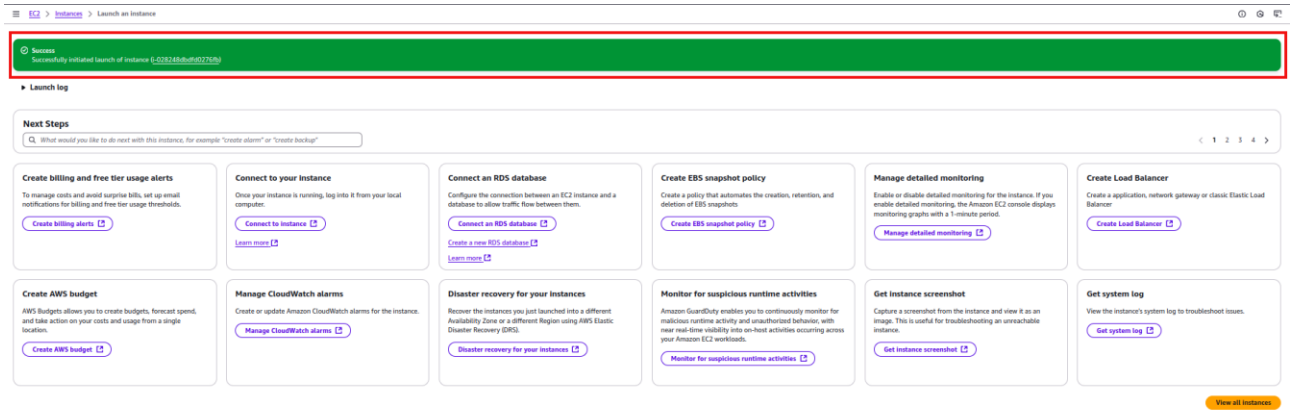


Figure 4

Which informs about creation of the instance based on the stored image.

After opening "EC2 → instance" in AWS we shall see the created instance:

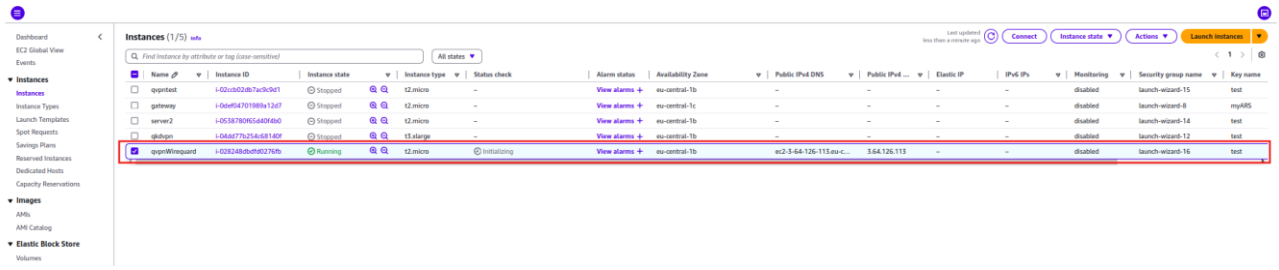


Figure 5

This instance is automatically launched. A "Public IPv4" number is also assigned, which may change after restarting the instance or stopping and resuming it. Therefore, we must associate a fixed IPv4 address with this instance, which will be needed in further system configuration.

To do this, open **"Network & Security" → "Elastic IPs"** in the AWS menu:

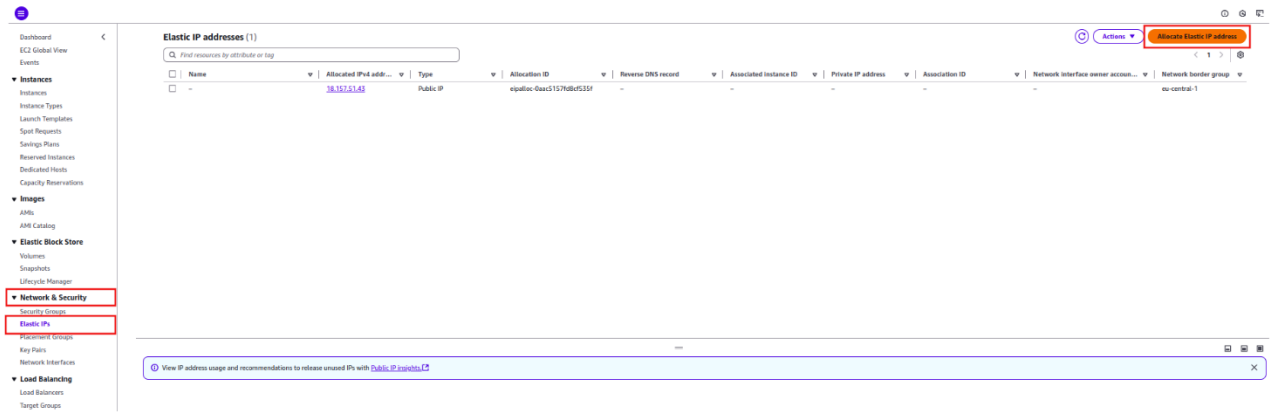


Figure 6

Now, click on *'Allocate Elastic IP address':*

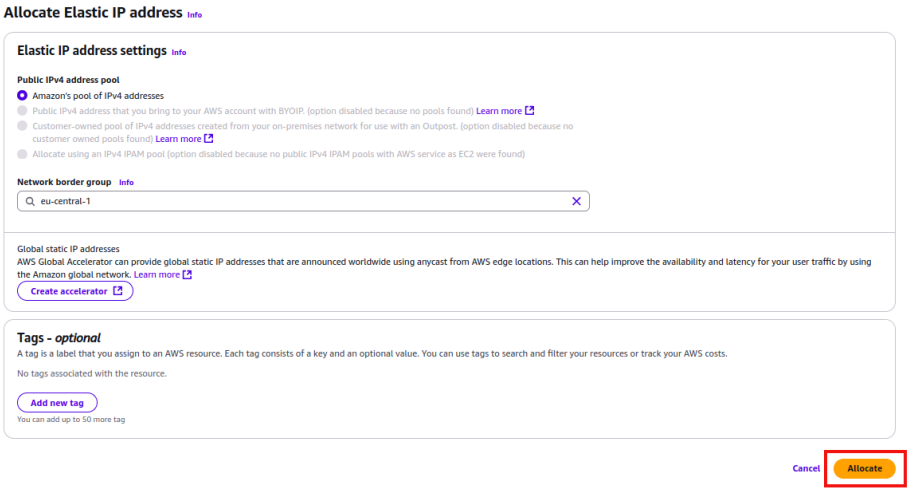


Figure 7

After clicking the "Allocate" button, a Public IPv4 address is generated (as shown in Figure 7); in our example, it is 18.157.51.43.

Now we assign the generated IP address to our newly created instance. To do this, right-click the IP address (the link in blue) and select:

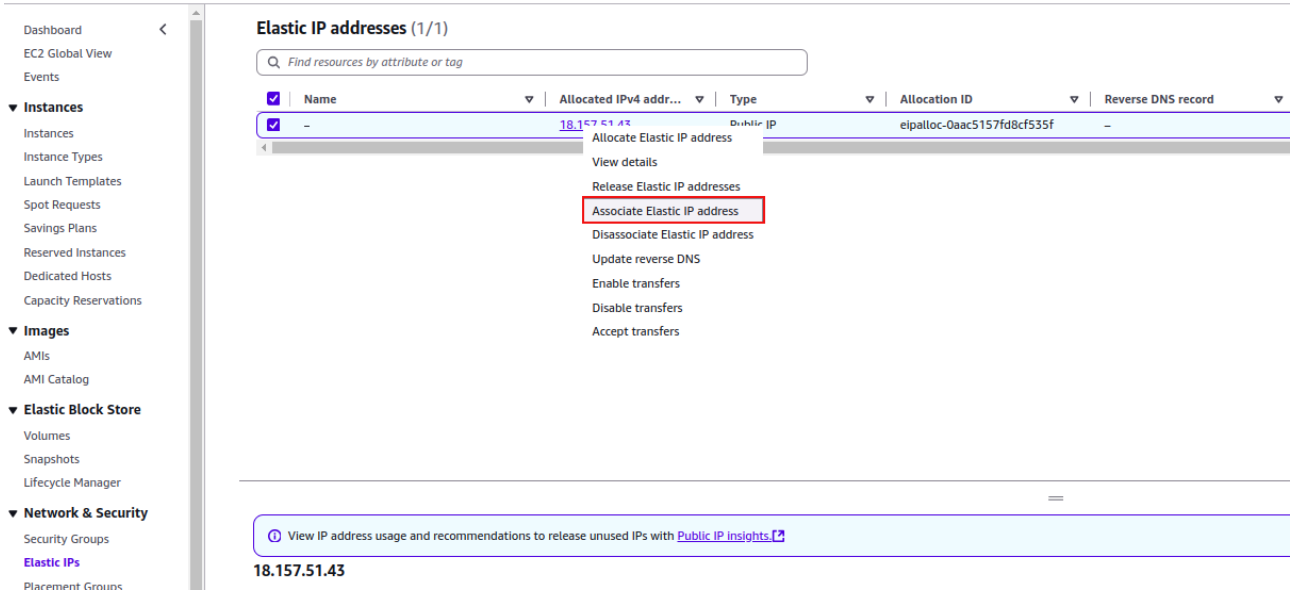


Figure 8

The screen appears:

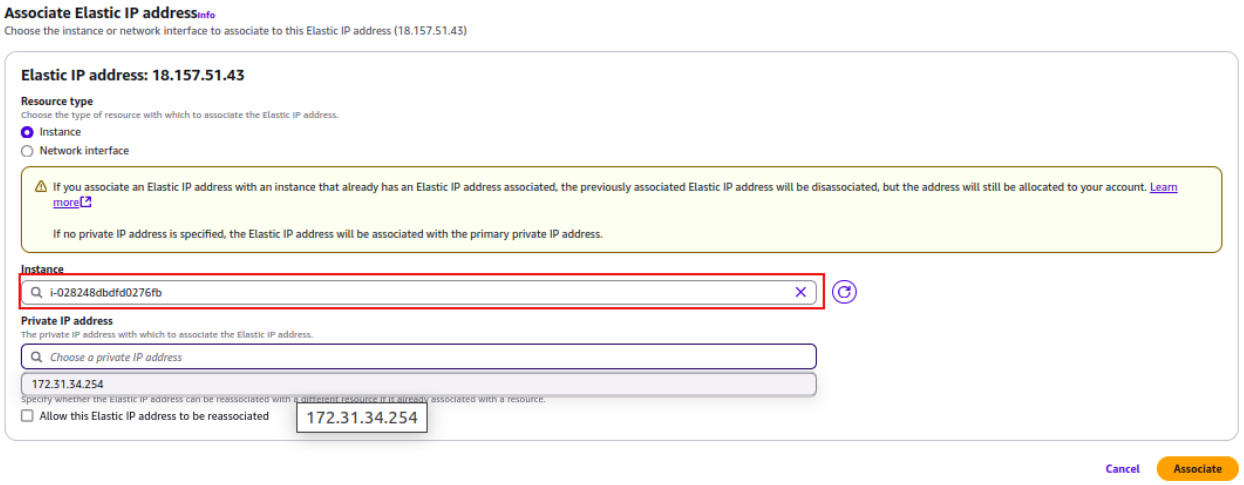


Figure 9

We select our instance and click on *'Associate':*

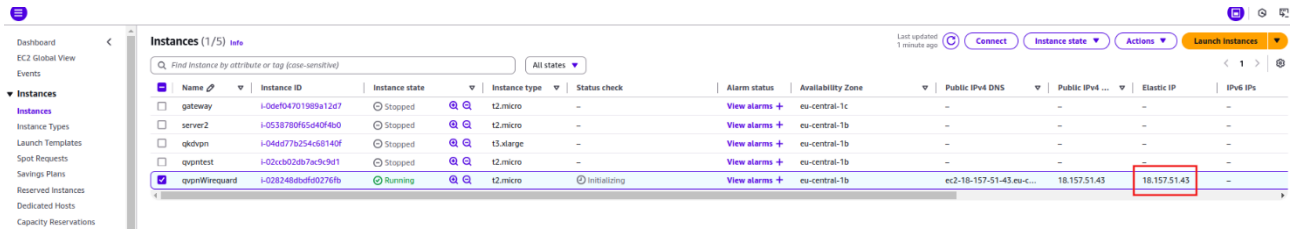


Figure 10

By navigating to *'EC2 → Instances'*, we can verify that our server has been assigned the *"Elastic IP.":*

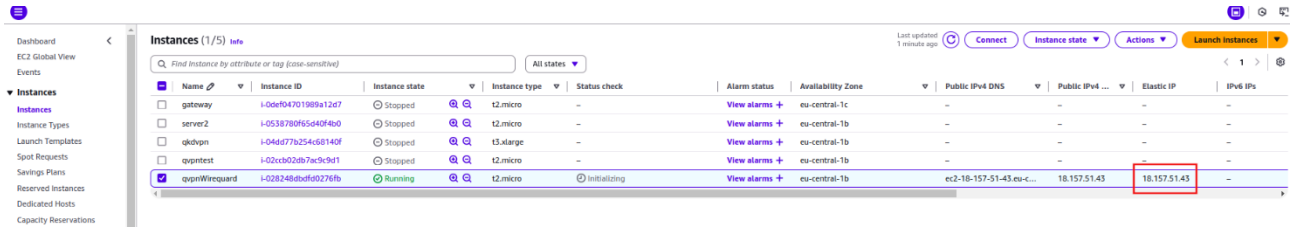


Figure 11

We still need to open the necessary ports for our services so they can communicate publicly. To do this, we select (using the checkbox) our server and edit the *"Secure"* security group:

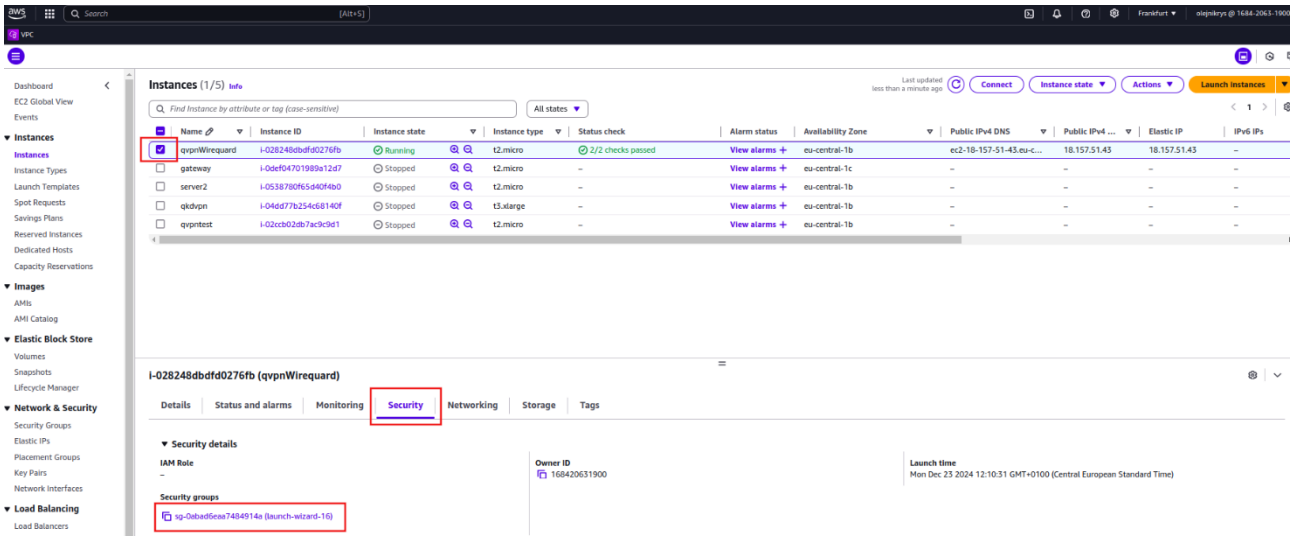


Figure 12

After clicking the link to the security group, we edit it (adding ports) as follows:

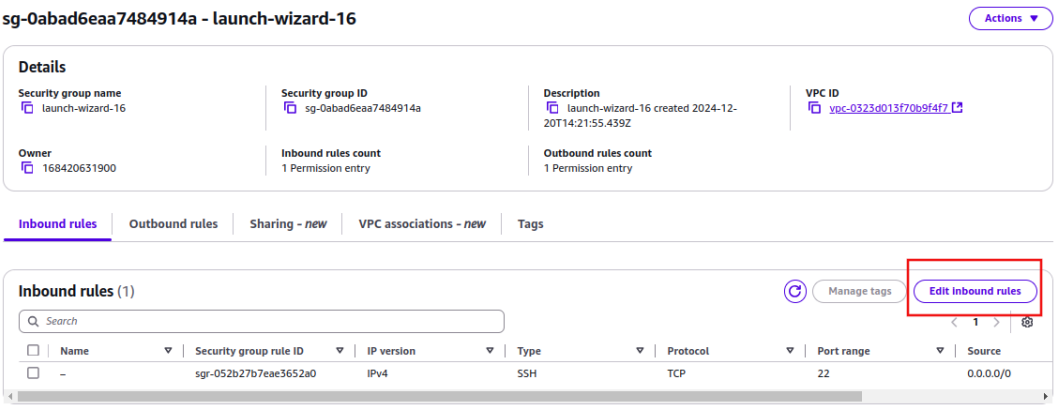


Figure 13

We now click on *'Edit inbound rules'*:

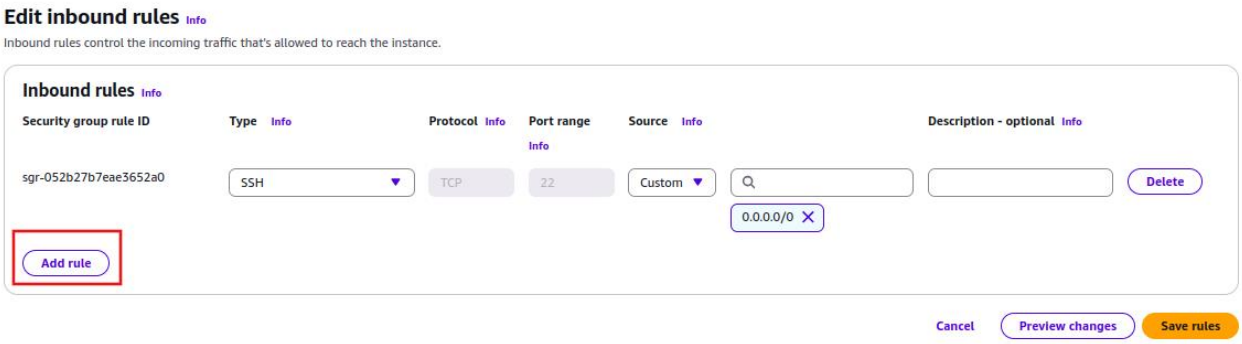


Figure 14

Click the *"Add rule"* button and add the respective ports (UDP for WireGuard, TCP for the service):

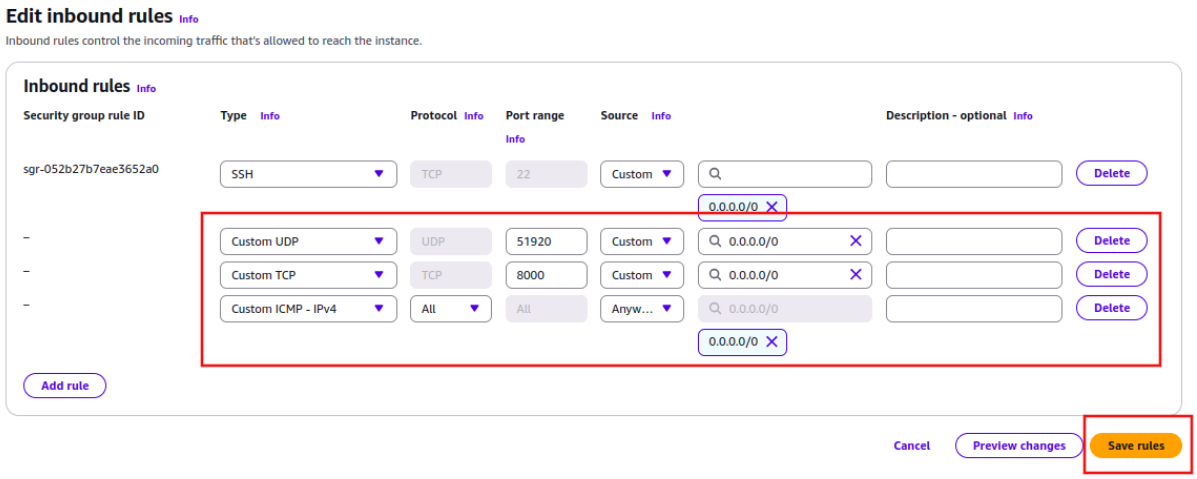


Figure 15

Support for *"Custom ICMP-IPv4"* (external pings, e.g., useful during system startup) has also been added. After clicking the "Save rules" button, we have:

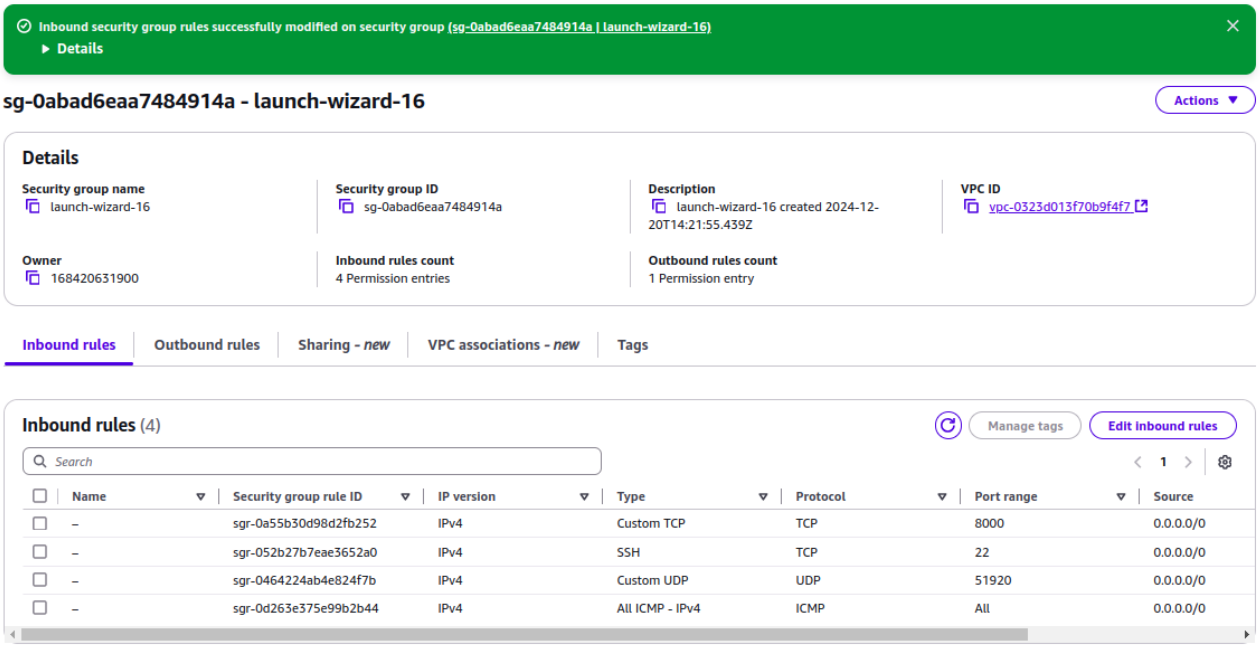


Figure 16

After selecting *"EC2 → Instances"*, our server appears as follows:

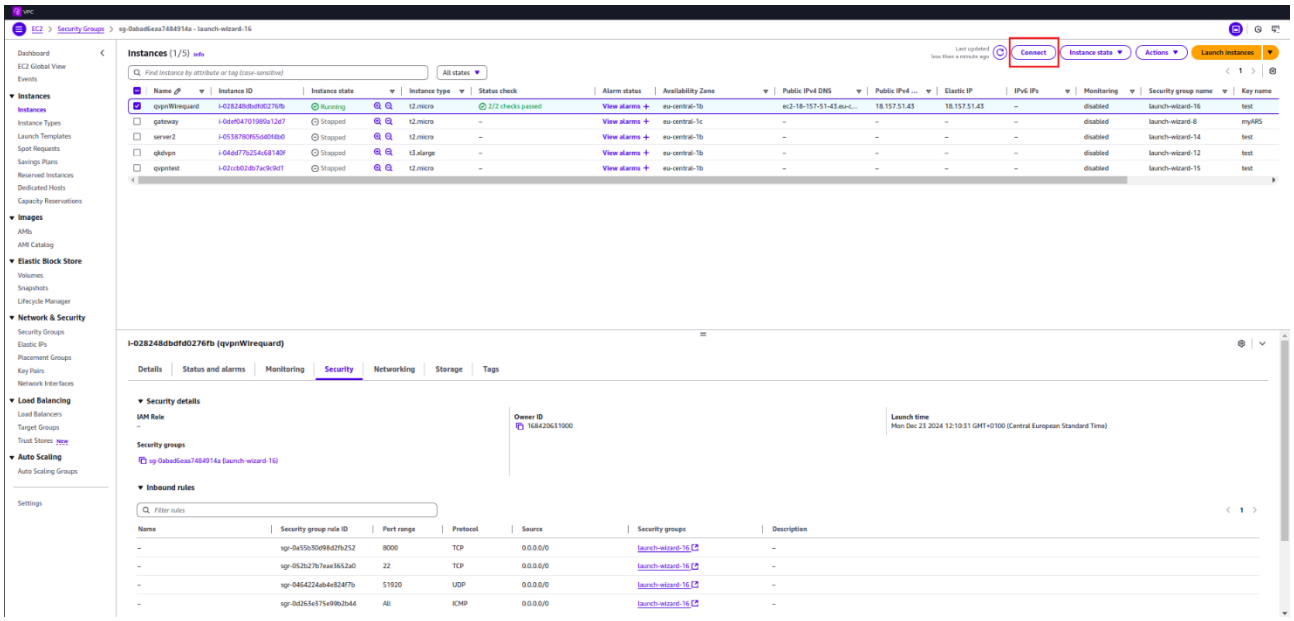


Figure 17

We connect to the server using *"EC2 Instance Connect"* or SSH.

The user in both cases is either **"ubuntu"** or **"root"**, for example:

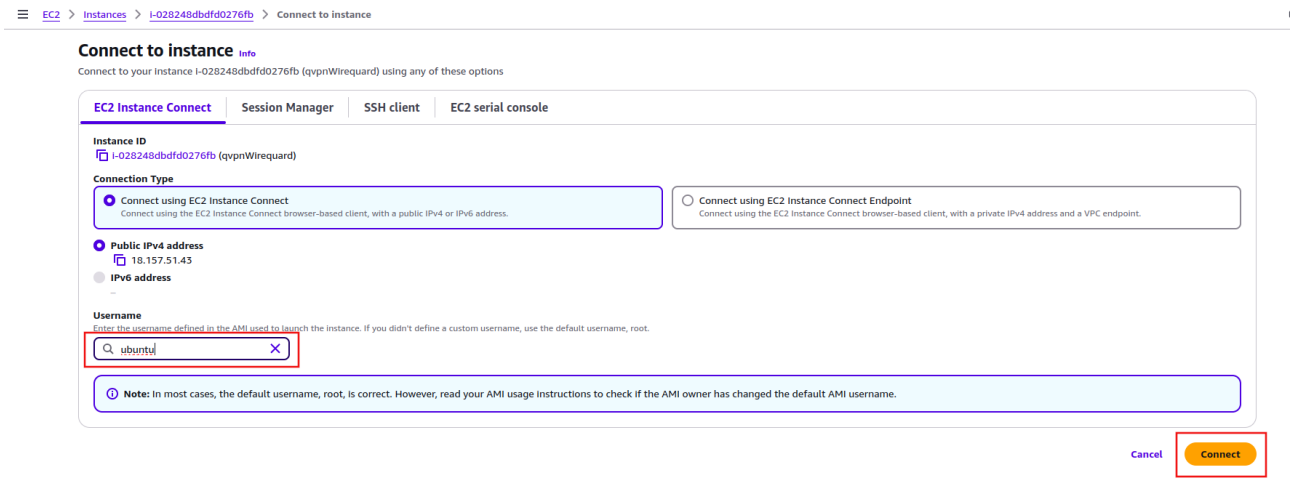


Figure 18

After clicking the *"Connect"* button, the terminal appears:

Figure 19

Of course, the terminal can also be accessed externally via SSH (see the *"SSH Client"* tab). The connection instructions are provided in that tab:

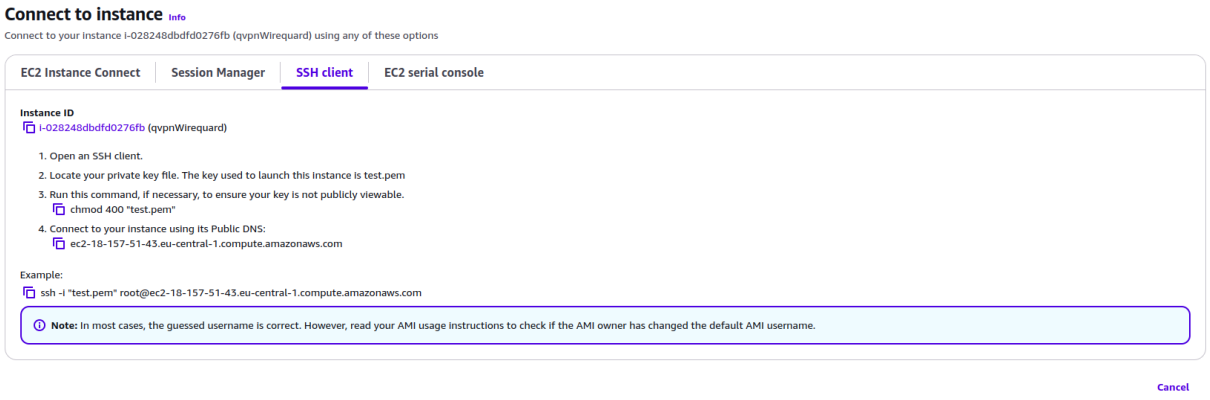


Figure 20

III. Configuration of the QKD Emulator – pQKD Device on the Client Side

The default configuration (factory reset) of the pQKD device uses a network interface with the IP address 192.168.1.80. A factory reset is performed by holding the button (on the back of the device) while connecting the power supply until the blue LED starts flashing).

You need to configure your computer to use the same network (i.e. assign it the IP address from the same network, e.g. 192.168.1.200), then connect the pQKD device using its ETH0 (rightmost) LAN connector to your computer using a network cable.

Figure 21. Desktop and Rack Mounted pQKD. Use the rightmost (ETH0) LAN RJ-45 connector.

After powering on the pQKD device, wait until the blue LED lights up steadily.

In your browser, enter the address: http://192.168.1.80. This will open a login page.

The default login credentials are:

Username: admin
Password: admin

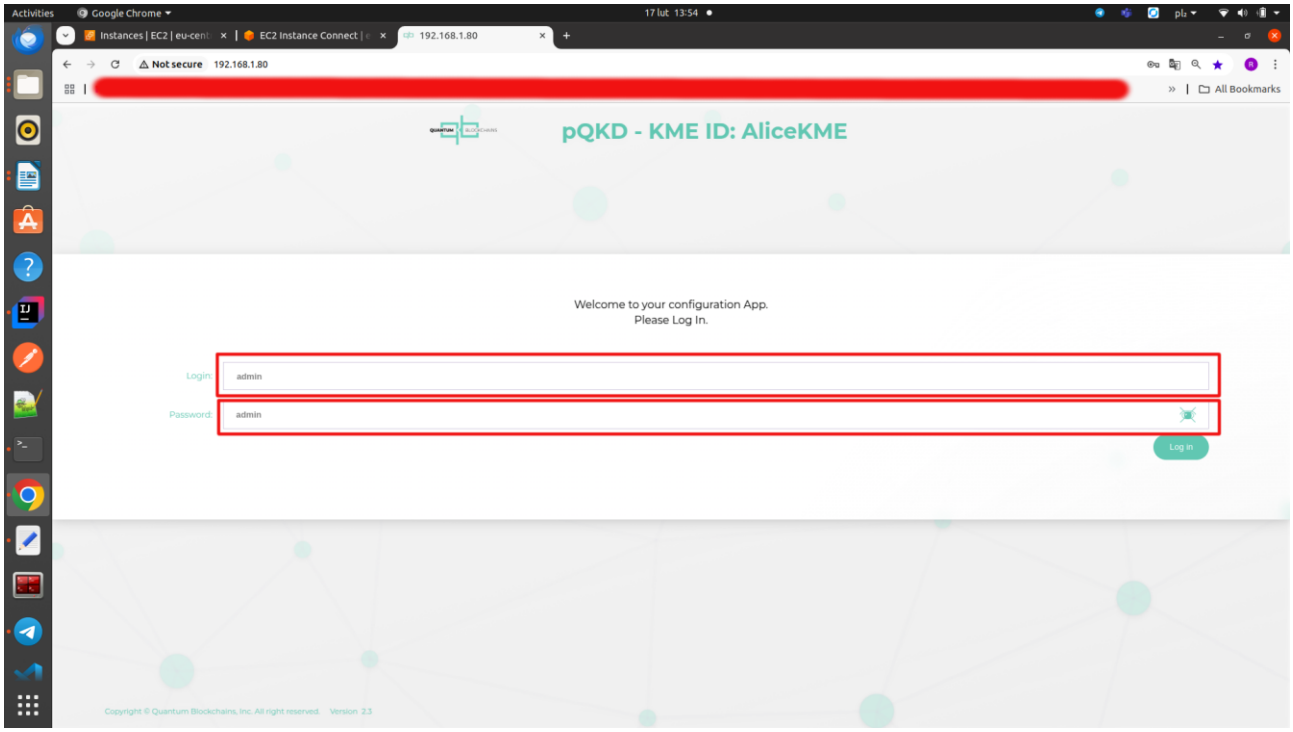


Figure 22

After logging in, a configuration wizard for the device will launch:

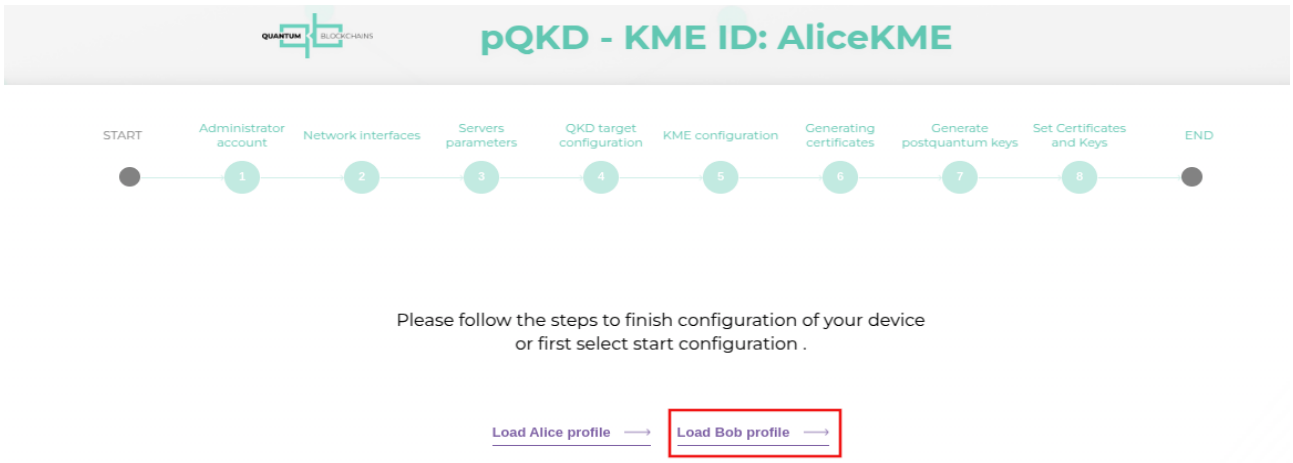


Figure 23

Select the profile: *"Load Bob profile"*. In the next step, set your own unique password for logging.

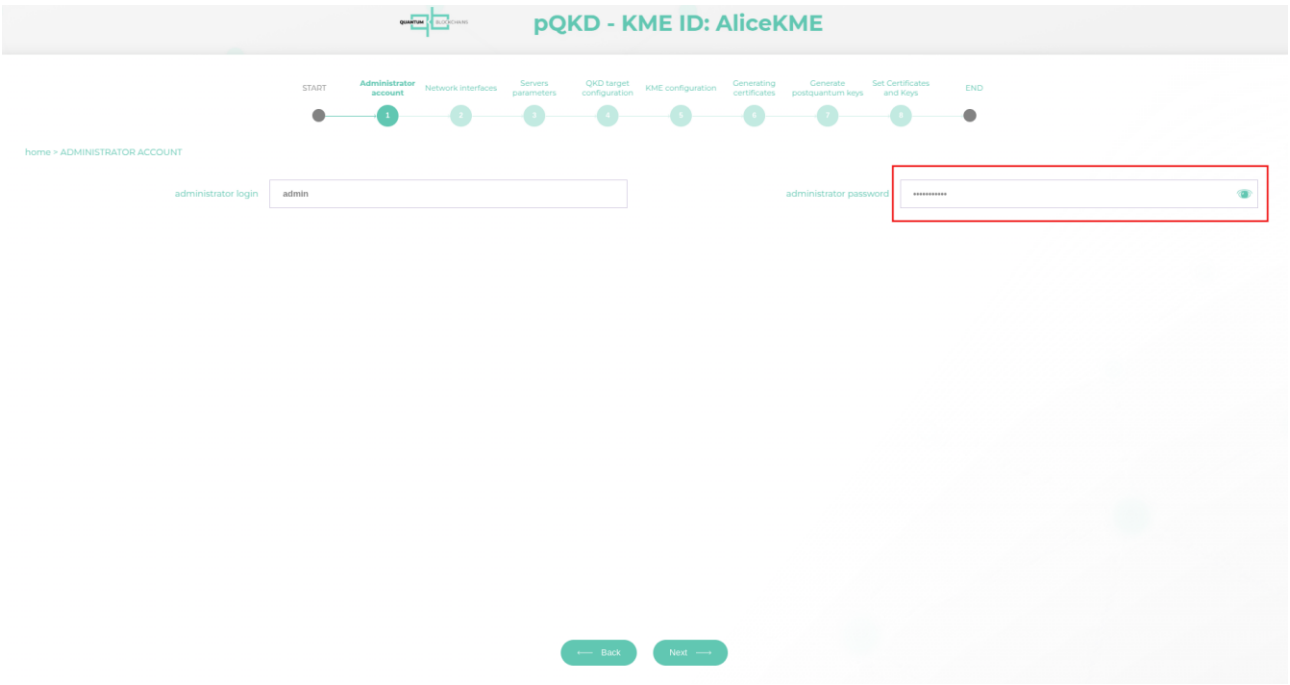


Figure 24

Set your access password.

In step (“NETWORK INTERFACES), set a custom IP address for the ETH0 interface that matches the network your computer will operating on. The ETH1 port will not be used in this configuration, so it can remain unchanged.

Press next step button:

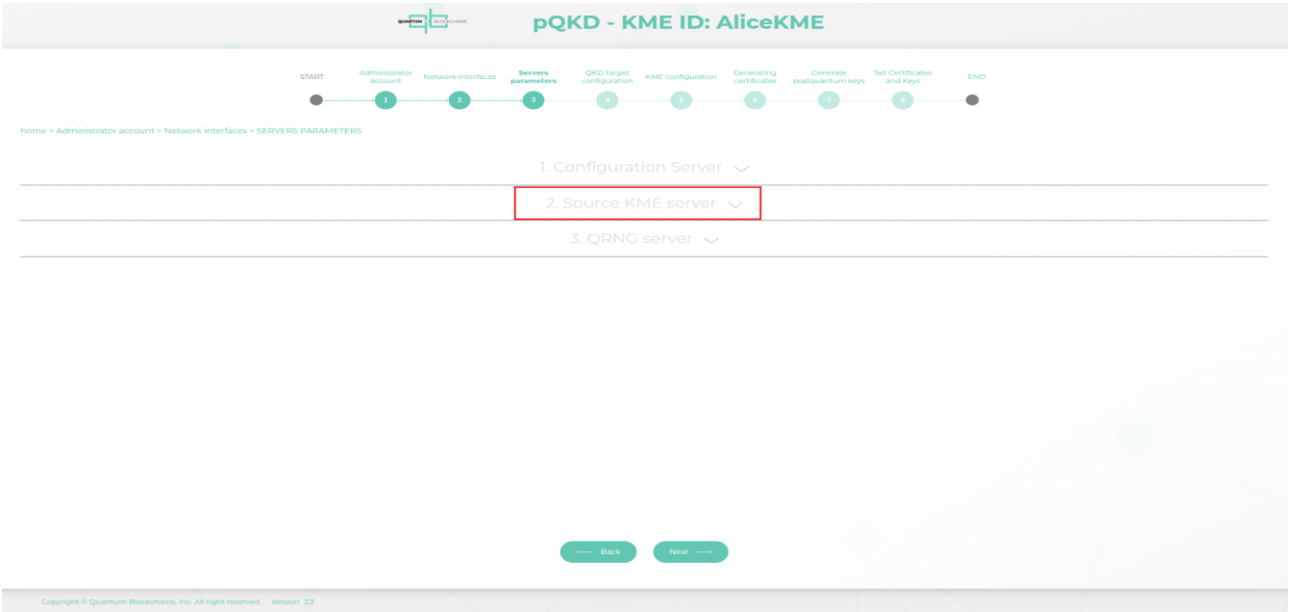


Figure 25

Open “2. Source KME server”:

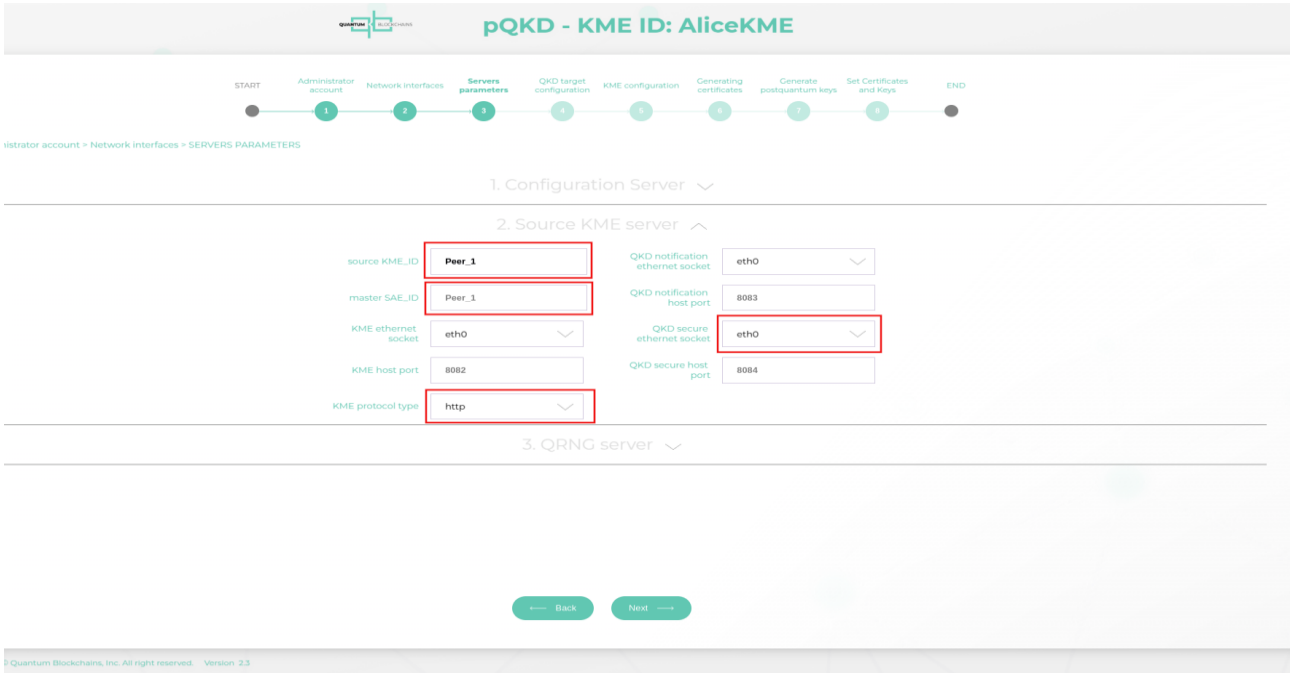


Figure 25

Additionally, change the Source Identifier (KME_ID)* to something like Peer_1 and the Master Identifier (SAE_ID) to Peer_1. The number must be used in the name. Based on this, the allocated pool of ports on AWS.

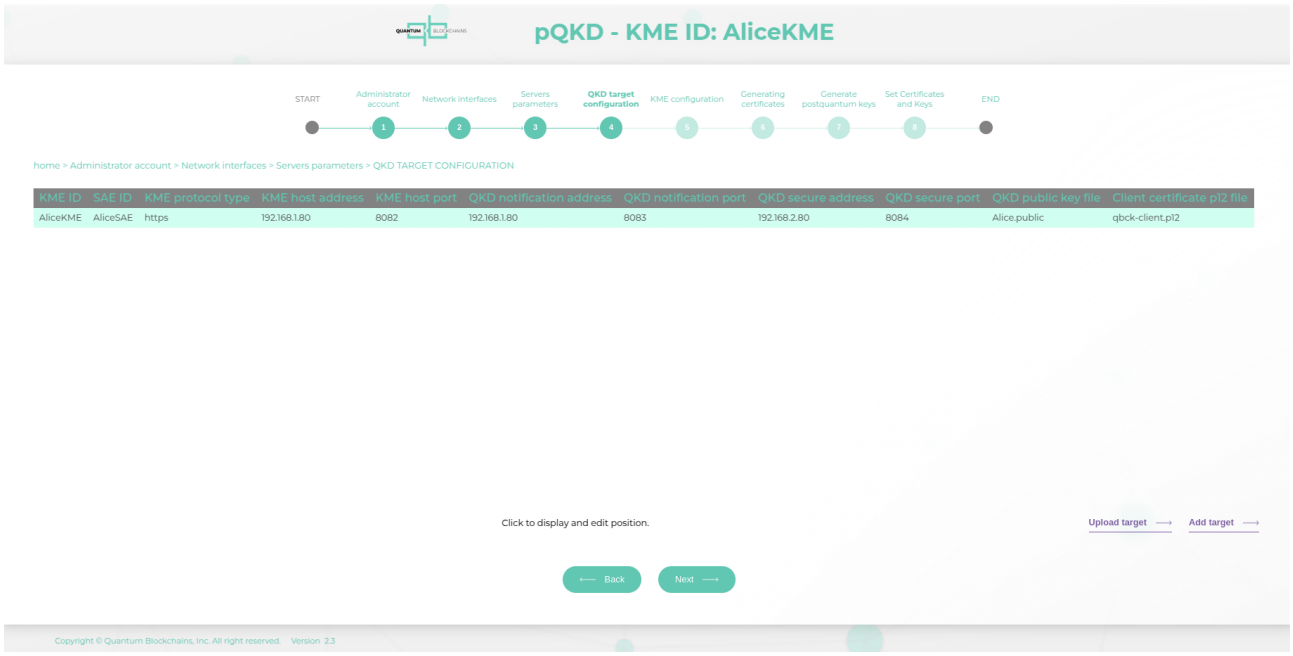


Figure 26

In the next step, *"QKD Target Configuration",* we configure access to the pQKD system located on AWS (pQKD Twin), which is accessed via the qVPN client installed on the client computer. After selecting:

Figure 26

Change HTTPS to HTTP and adjust the addresses to match your target network (your computer's ip).

Next, accept the modified values and proceed step by step, confirming *"KME Configuration"*, *"Generating Certificates"*, and finally *"Generate Post-Quantum Keys".*

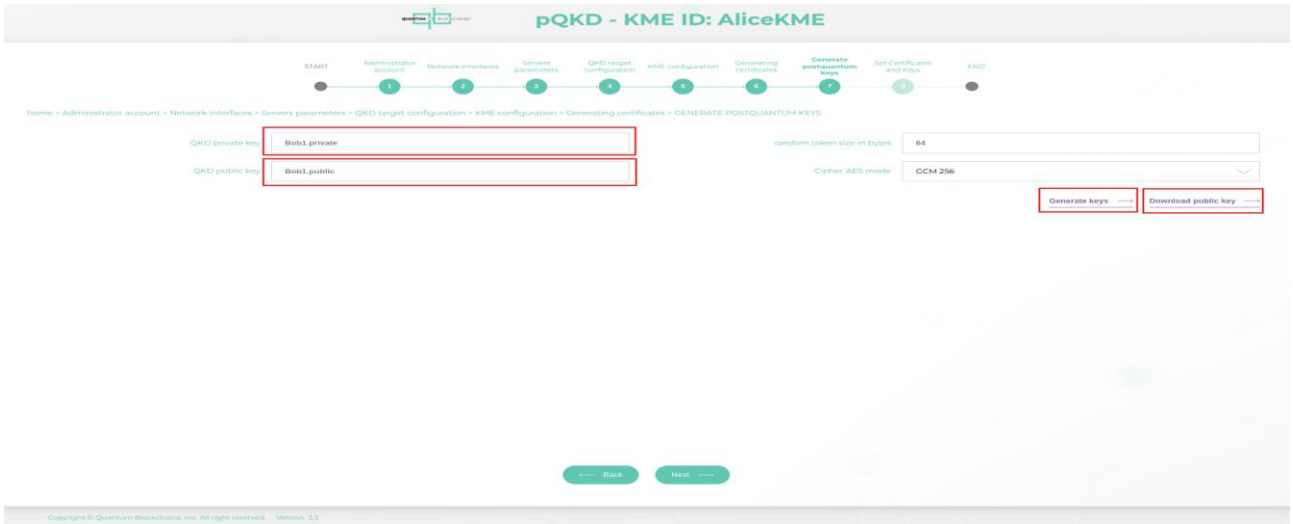


Figure 27

Change the key names, for example, from *"Bob.private"* to *"Bob1.private"* and from *"Bob.public"* to *"Bob1.public".* After renaming, generate the key by clicking the *"Generate Keys"* button and download the *Bob1.public* key to your computer (it will be needed for server configuration).

Proceed to the *"Set Certificates and Keys"* step. Set the newly generated keys (*Bob1.private*, *Bob1.public*) in the corresponding fields:

Figure 28

Accept the changes and proceed to the end of the wizard. Confirm the settings to save them. The device will restart automatically.

Now you can reconfigure your computer's network settings (IP address) to match the network you were previously using (in the example 192.168.1.90). To access the pQKD device setup from now on, you must use the IP address you configured (see).

If needed, you can adjust individual pQKD device parameters later by navigating through the appropriate menu options.

All pQKD connections within the client's private network connected to the pQKD device use HTTP.

All other external connections are encrypted with the post-quantum cryptographic algorithm "CRYSTALS Kyber" (FIPS 204 Key Encapsulation Mechanism).

IV. Client software installation

The system can currently be installed on Linux computers (Ubuntu, Debian, etc.). The installation files must be downloaded from: https://www.quantumblockchains.io/decks/qVPN_Linux.zip

The installation proceeds in the following steps:

1. Installation of WireGuard

The WireGuard installation instructions are available at: https://www.wireguard.com/install/

For Ubuntu run:

```
sudo apt update
sudo apt install wireguard
```

WireGuard must be installed without generating RSA keys (we are setting them upfront). Ensure that WireGuard Tools (wg-quick) are also installed. You can verify this with the following command:

wg-quick

The command should execute successfully, displaying the command's help output.

2. Installation of Java

Run the following commands:

```
sudo apt update
sudo apt install default-jre
```

You can verify the installation with:

```
java -version
```

3. Installation and Configuration of VPN (qVPN)

Copy the file qvpn.tar to your system, then extract it using the command:

```
tar -ux qvpn.tar
```

This will unpack the necessary files for further configuration.

A directory named qVPN will be created. Enter the directory using the command:

```
cd qVPN
```

Inside this directory, you'll find the following files:

```
init.config
qvpn.jar
qvpn.service
qvpn.sh
READ.ME
wgc.conf
```

Configuration Before Installation:

Before installation, you must configure *qvpn* service by editing the *wgc.conf* file:

```
#
# client
#
[Interface]
QKD_Host = 192.168.1.200
QKD_EndPointProxy = 18.157.51.43:8000
QKD_TimeOut = 10
Address = 10.0.0.2/32
DNS = 1.1.1.1
```

```
[Peer]
QKD_IDENT = Peer_1
QKD_KME = 8082, 192.168.1.90:8082
QKD_EQKD = 8083, 192.168.1.90:8083
QKD_QKD = 8084, 192.168.1.90:8084
Endpoint = 18.157.51.43:51920
AllowedIPs = 172.31.0.0/16
PersistentKeepalive = 25
```

Configuration File Explanation:

The configuration file format is similar to a standard WireGuard configuration file. However, it includes new commands prefixed with *QKD*. In addition to these new commands, you can still use all standard WireGuard commands.

[Interface] Section

- `QKD_Host`: Enter your computer's IP address (used for communication with the pQKD device). In the example, this is the address in the default pQKD network (i.e. 192.168.1.200).

- `QKD_TimeOut`: Maximum response time (in seconds) for pQKD to reply.

- `QKD_EndPointProxy`: The endpoint address (communication channel) exposed on AWS for auxiliary communication with the client.

- `Address`: The local address of the virtual network interface created on the computer. It's a good practice to establish a clear IP structure, e.g., `10.0.0.1` for the VPN server, `10.0.0.2` for Peer 1, 10.0.0.3 for Peer 2, and so on.

[Peer] Section

- `QKD_IDENT`: Identifier assigned to the Peer.

- `QKD_KME`: Parameters for communication with the *Key Management Entity (KME)*.

- `QKD_EQKD`: Parameters for communication with the emulated QKD service (for QKD notification channel).

- `QKD_QKD`: Parameters for communication with the QKD service.

These configurations (IP addresses and ports) must match those set in the pQKD device configuration.

Ensure consistency across all settings to guarantee proper communication and integration between qVPN and pQKD.

NOTICE:

Due to the definition of multiple ports on the qVPN server side, automatic port assignment has been introduced via the *QKD_IDENT* parameter.

The value of this parameter should follow the syntax:

`<name>_<number>`

Examples: `Peer_1`, `Bob_1`, `User_1`, etc.

The numbers should be assigned sequentially: 1, 2, 3, …. In the example above, the identifier is `Peer_1`. Name consistent with pQKD configuration (Figure 25).

Installation

After configuring the configuration file, run the installation command:

```
sudo sh qvpn.sh install
```

The qVPN service will also start automatically after restarting the computer. To uninstall the service, use:

```
sudo sh qvpn.sh uninstall
```

The program and configuration files are installed in the directory:

```
/usr/bin/qVPN
```

After uninstallation, all files are removed from this location.

Editing Configuration Parameters (wgc.conf)

You can modify the wgc.conf file in two ways:

1.Uninstall qVPN, Uninstall the service, Prepare the updated configuration file, Reinstall the service.

2. Direct File Editing: Edit the `wgc.conf` file located at:

```
/usr/bin/qVPN
```

Restart the service with the following command:

```
sudo systemctl restart qvpn
```

This ensures the changes take effect immediately.

V. qVPN configuration on AWS

To configure the qVPN server, open a terminal to the server instance on AWS (see Chapter II, Figure 19). After opening the terminal, the console will appear:

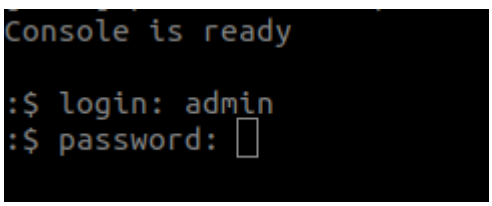


Figure 29

After entering the login (admin) and password (admin), proceed to enter commands.

You can view a list of available commands by typing: ? :

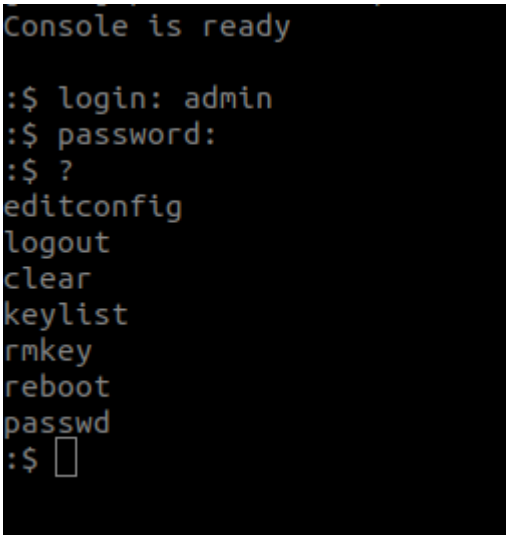


Figure 30

1. `editconfig` – Used to edit the configuration file.

After executing this command, you gain access to the qVPN server configuration file. It's a simple line-based editor with basic commands:

`? – Display the list of editor commands

`show` – Display the contents of the configuration file

`delete`– Delete a specific line in the file:

delete <line number>

`insert` – Insert a new line before a specified line:

insert <line number> <line content>

`edit` – Edit a specific line:

edit <line number> <new line content>

`test` – Test the correctness of the configuration file

`save` – Save the configuration file

`end` – Exit the configuration editor

2. `logout` – Log out (end session)

3. `clear` – Clear the screen

4. `keylist` – Display a list of post-quantum keys available in the system

5. `rmkey` – Delete a specific key file

6. `reboot` – Restart the system

7. `passwd` – Change the password

The keys available on the server can be displayed using the following command: `keylist`:

Figure 31

We need to upload the previously generated public key `Bob1.public` (during pQKD configuration) to this resource. To do this, copy the file `Bob1.public`to the `~/` directory of the Ubuntu user using the following command:

```
scp -i key.pem Bob1.public ubuntu@18.157.51.43:~/
```

The key `test.pem`is obtained during the creation of the EC2 instance from the AMI image (see Figure 3).

Proper permissions must be set for the `test.pem` key (see Figure 20).

In the example above, we assume that both the `test.pem` key and the `Bob1.public file are located in the same directory.

Of course, the IP address `18.157.51.43 is just an example here. You should replace it with your generated IP address (see Figure 7). After copying the key, we have:

Figure32

Now we can use the key in the configuration.

Edit the configuration using the command:

```
editconfig
```

After executing this command, the screen will appear as follows:

Figure 33

In this mode, we have entered the simple command editor (a list of available commands can be displayed using `?`).

After issuing the `show` command, the configuration will be displayed:

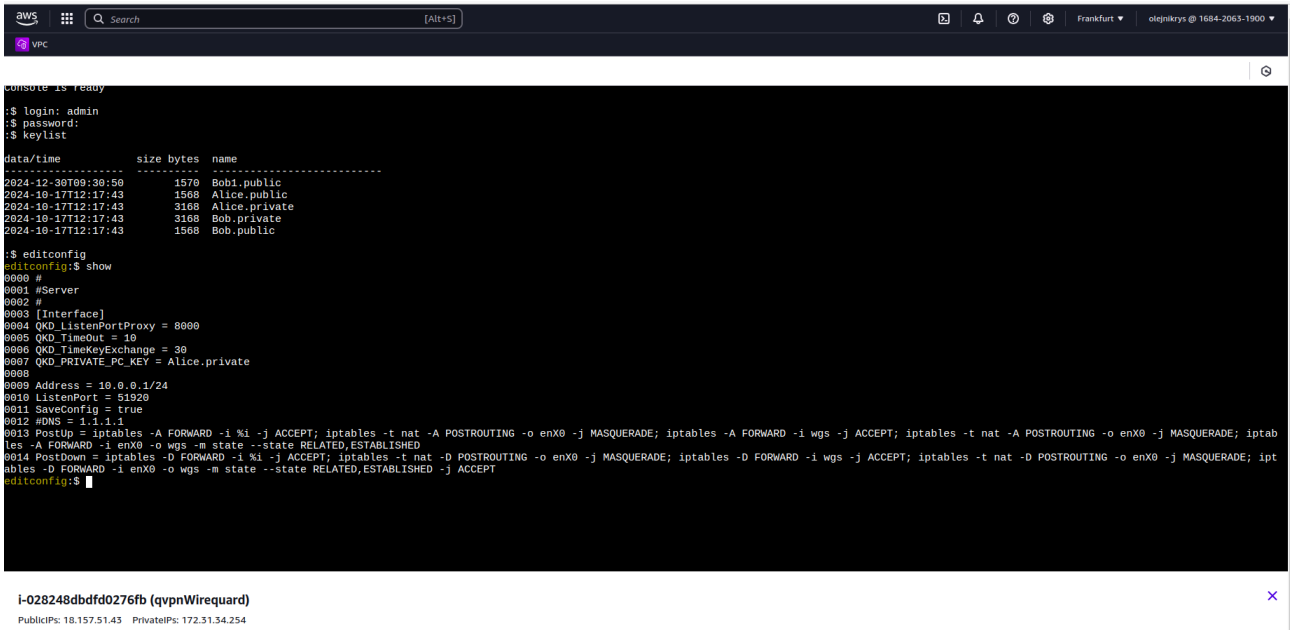


Figure 34

Just like during the configuration of the qVPN client, here we are also dealing with WireGuard commands along with additional commands prefixed with `QKD_`. The names of these variables describe their purpose.

If changes are required, such as different ports, alternative routing, or other parameters, they can be edited directly here.

To add a user (qVPN client), you need to add a [Peer] section. For our previous configuration, it will look like this:

insert 40
insert 40 [Peer]
insert 40 QKD_IDENT = Peer_1
insert 40 QKD_PUBLIC_PC_KEY = Bob1.public
insert 40 AllowedIPs = 10.0.0.2/32

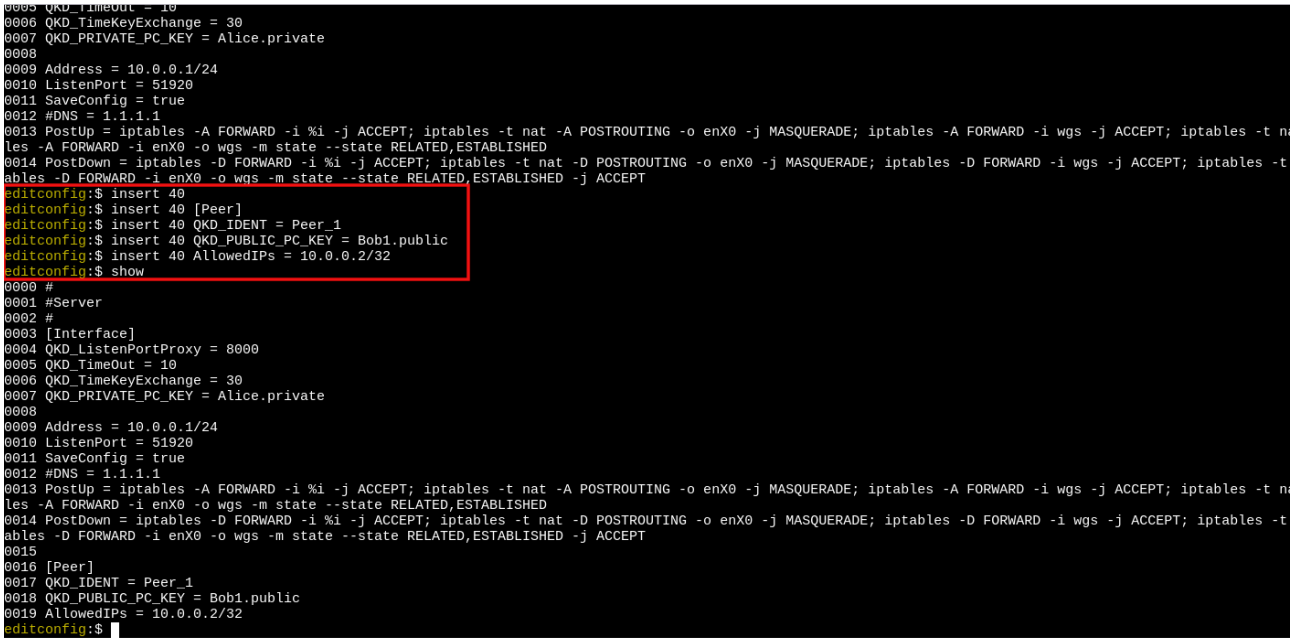


Figure 35

In the `insert` command, the line number 40 was used. Since the file has only 19 lines, the new lines will be added at the end of the file.

The file has been updated with the "Peer_1" configuration.

To verify the correctness of the entered data, use the command:

`test`

To save the changes, use:

`save`

After saving, exit the editing mode with: `end`

To apply the changes, restart the system with the following command: `reboot`:

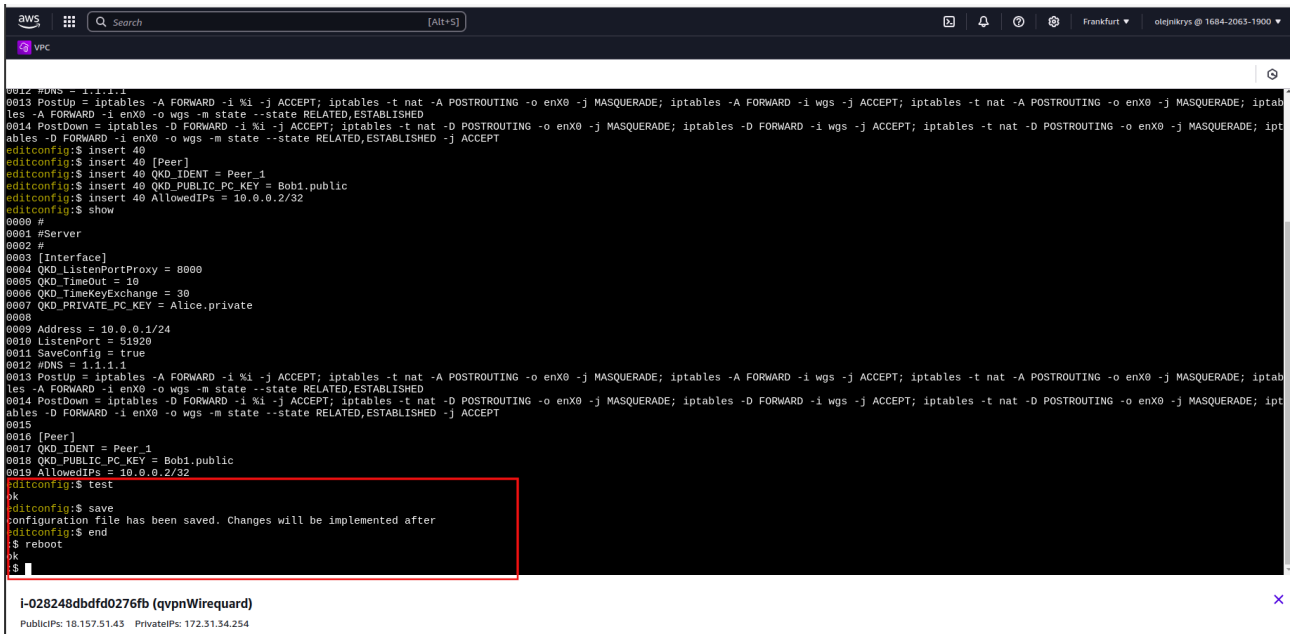


Figure 36

If everything has been configured correctly, on the client side, after executing the command: `ip`

you should see the network interface `wgc` listed:

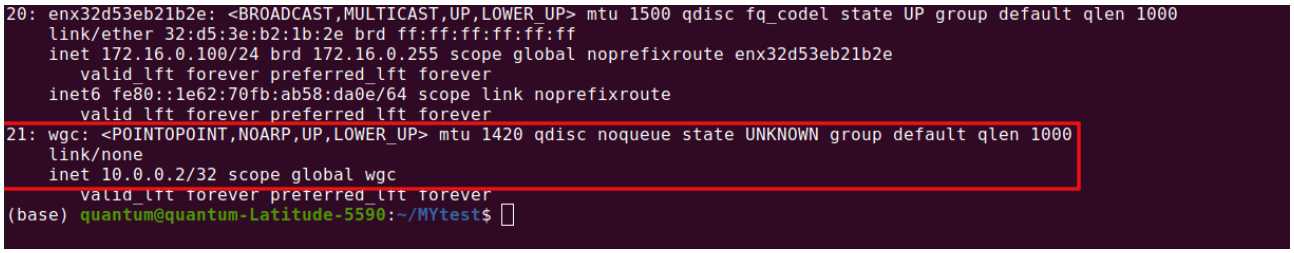


Figure 37

Our "VPN server" has a local IP address of 172.31.34.254, and the range of available addresses is defined in (Chapter III, Point 3) as: *AllowedIPs = 172.31.0.0/16*

Therefore, we should be able to ping the server using the following command:

```
ping 172.31.34.254
```

Figure 38

If there are any issues, you can inspect the qVPN service on the client side with the following command:

```
sudo systemctl status qvpn
```

To check the status and transmission info of WireGuard, use:

```
sudo wg show
```

Adding Additional Peers

Subsequent Peers can be added in a similar manner.

Of course, access to hosts in the AWS network can be configured using AWS tools (e.g., route tables, NAT, firewall rules, etc.).

Troubleshooting Connection Issues

If the connection is not established after installation, try uninstalling the qVPN client and reinstalling it.

The connection may take several tens of seconds to establish after starting the services. This delay is related to the refresh time of the post-quantum key.

Support

In case of any troubles contact our support team at: support@quantumblockchains.io