APPLICATION NOTE

QCS Configuration Tool

Introduction

The Configuration Tool is a Windows command-line tool designed for configuration and management of your QCS Multihost System. Configuration tool is installed along with other QCS components by TSM or the 5-part installation.

KEYSIGHT TECHNOLOGIES

Table of Contents

Launch configuration tool

Open PowerShell with Administrator Privileges

Use command “QcsConfigureCli” to launch.

Create System Topology

To create a graphical representation of the host PCs, the chassis and the connections between them in your system, we use the Graphviz DOT language.

The Graphviz DOT language is a plain text graph description language primarily used for defining the structural representation of diagrams and network graphs. Developed as part of the Graphviz software suite, DOT language allows users to create complex visualizations by specifying nodes, edges, and their attributes in a straightforward syntax. This language supports a wide range of graph types, including directed and undirected graphs, and provides options for customizing the appearance of nodes and edges through various attributes like shape, color, and labels. Typically saved with a .dot or .gv extension, DOT files can be processed by Graphviz tools to generate visual outputs in multiple formats, such as PNG, SVG, and PDF. The versatility and simplicity of the DOT language make it a powerful tool for developers, data scientists, and researchers looking to visualize relational data efficiently. https://graphviz.org/

Sample topology file showing 3 racks system

```
digraph Topology {
    // Version of the topology
        version="2.0";

    // Global graph attributes
    fontsize="20";
    fontname="Helvetica,Arial,sans-serif";
    bgcolor="lightyellow";
    rankdir=LR;
    node [shape=record, style=filled, fillcolor=green, fontname="Helvetica,Arial,sans-serif"];
    edge [color="darkorange", penwidth=3.0, arrowsize=1.5, arrowhead="open"];
    ////////////////////////////////////////////////////////////////////////////
    subgraph cluster_1 {
        label="Rack 1";
        style=filled;
        rankdir=TB;
        node [fillcolor=lightblue];
        R1C1 [label="1 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R1C2 [label="2 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R1C3 [label="3 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1 | <D2> SyncDown 2 | <D3> SyncDown 3 | <D4> SyncDown 4"];
        R1C4 [label="4 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1 | <D2> SyncDown 2 | <D3> SyncDown 3 | <D4> SyncDown 4"];
        R1C5 [label="5 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"] ;
        R1C6 [label="6 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
    }
    subgraph cluster_2 {
        label="Rack 2";
        style=filled;
        rankdir=TB;
        node [fillcolor=green];
        R2C1 [label="1 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R2C2 [label="2 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R2C3 [label="3 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1 | <D2> SyncDown 2 | <D3> SyncDown 3 | <D4> SyncDown 4"];
        R2C4 [label="4 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1 | <D2> SyncDown 2 | <D3> SyncDown 3 | <D4> SyncDown 4"];
        R2C5 [label="5 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R2C6 [label="6 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
    }
    subgraph cluster_3 {
        label="Rack 3";
        style=filled;
        rankdir=TB;
        node [fillcolor=orange];
        R3C1 [label="1 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R3C2 [label="2 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R3C3 [label="3 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1 | <D2> SyncDown 2 | <D3> SyncDown 3 | <D4> SyncDown 4"];
        R3C4 [label="4 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1 | <D2> SyncDown 2 | <D3> SyncDown 3 | <D4> SyncDown 4"];
        R3C5 [label="5 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
        R3C6 [label="6 | Chassis: MY00000000 | <U1> SyncUp | <D1> SyncDown 1"];
    }

    // Define connections between Rack chassis
    Start -> R1C3:U1 [label="Input"];
    R1C3:D1 -> R1C4:U1;
    R1C3:D2 -> R2C3:U1;
    R1C3:D3 -> R3C4:U1;
    R1C3:D4 -> R1C2:U1;
    R1C2:D1 -> R1C6:U1;
    R1C4:D2 -> R1C1:U1;
    R1C6:D1 -> R1C5:U1;

    R2C3:D4 -> R2C5:U1;
    R2C3:D1 -> R2C4:U1;
    R2C2:D1 -> R2C6:U1;
    R2C4:D2 -> R2C1:U1;
    R2C3:D2 -> R2C2:U1;

    R3C4:D1 -> R3C1:U1;
    R3C4:D2 -> R3C2:U1;
    R3C2:D1 -> R3C6:U1;
    R3C6:D1 -> R3C5:U1;
    R3C1:D1 -> R3C3:U1;
}
```

DOT File Visualization

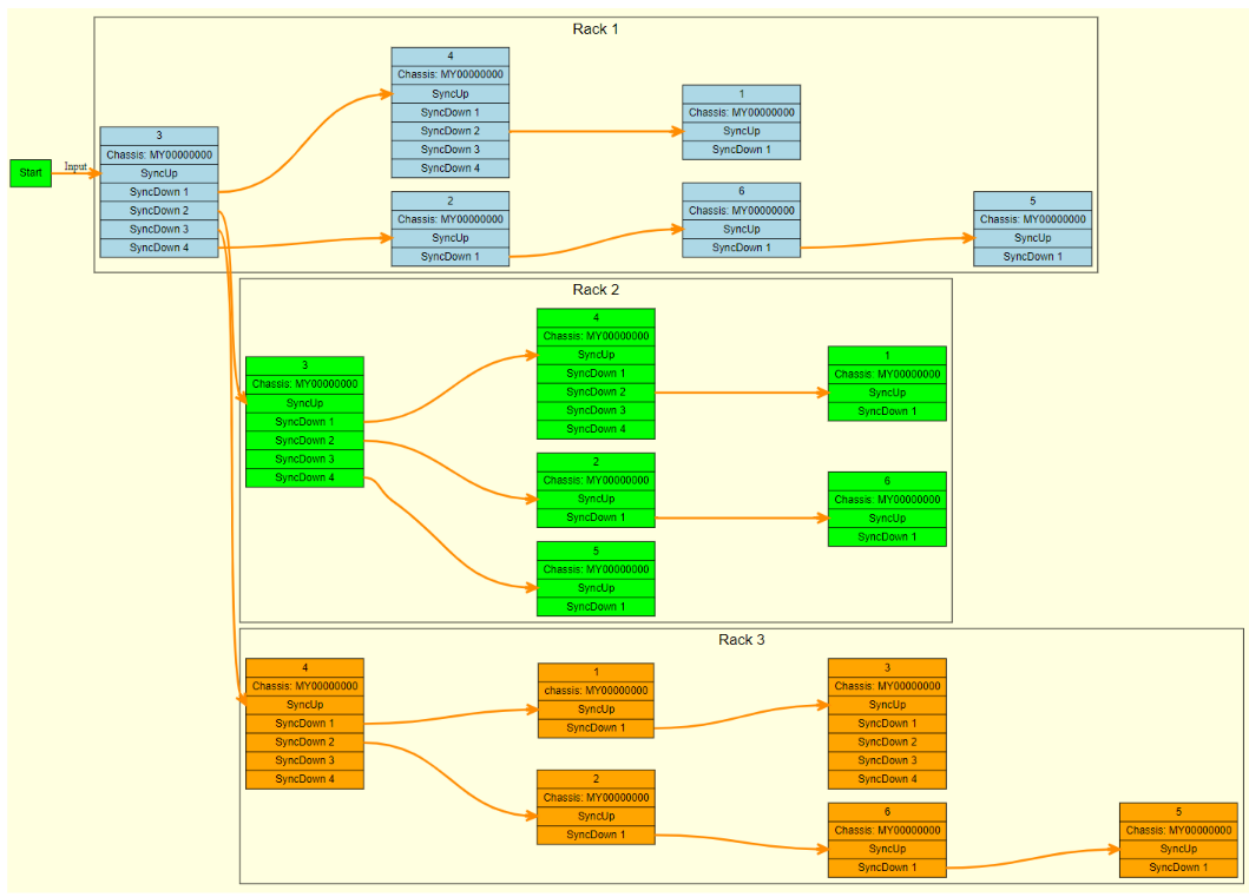

Figure 1. By using any online Graphviz free compiler we will see the image above

DOT file Explanation

This DOT file defines a directed graph named "Topology" with several visual attributes and subgraphs. Here's a breakdown of its structure:

1. Global Attributes:
    a. fontsize="20": Sets the font size for the graph to 20 points.
    b. fontname="Comic Sans MS": Uses "Comic Sans MS" as the font for text in the graph.
    c. bgcolor="lightyellow": Sets the background color of the graph to light yellow.
    d. TB =Top Bottom alignment
2. Node and Edge Attributes:

a. node [shape=record, style=filled, fillcolor=green, fontname="Helvetica,Arial,sans-serif"];: Configures nodes to have a record shape, be filled, and use "Comic Sans MS" for their labels.

b. edge [color="darkorange", penwidth=3.0, arrowsize=1.5, arrowhead="open"];: Sets edges to be blue with an open arrowhead, uses "Comic Sans MS" for edge labels, and adjusts label positioning.

3. Clusters:

   a. Cluster for Rack 1:

      i. subgraph cluster_1: Defines a subgraph (cluster) labeled "rack1" with light blue filled nodes.

      ii. Nodes R1C1 to R1C6 represent chassis within this rack, each labeled with a chassis number and a port (e.g., "Chassis: 1 | <p1> Port 1").

   b. Cluster for Rack 2:

      i. subgraph cluster_2: Defines a subgraph labeled "rack2" with yellow filled nodes.

      ii. Nodes R2C1 to R2C6 represent chassis within this rack, similarly labeled.

   c. c. Cluster for Rack 3:

      i. subgraph cluster_3: Defines a subgraph labeled "rack3" with red filled nodes.

      ii. Nodes R3C1 to R3C4 represent chassis within this rack, similarly labeled.

This DOT file structures a network topology diagram, visually grouping nodes (representing chassis) into racks and setting distinct visual styles for clarity. It defines connections (edges) between various nodes, each representing a chassis in different racks. Here's an explanation of the connections:

1. Start -> R1C1:U1 [label="Input"]

   This is the Input Stream coming to the ISM Leader host labeled as "start" to the upstream u1 of node R1C1 (Chassis 1 in Rack 1), with the edge labeled "Input stream".

2. R1C1:D1 -> R1C3:U1;

   There is a connection from R1C1 (Rack 1 Chassis 1) downstream d1 to R1C3 (Rack 1 Chassis 2) upstream u1

3. R1C1:D2 -> R2C1:U1;

   There is a connection from R1C1 (Rack 1 Chassis 1) downstream d2 to R2C1 (Rack 2 Chassis 1) upstream u1

4. R1C3:D1 -> R1C2:U1;

   There is a connection from R1C3 (Rack 1 Chassis 3) downstream d1 to R1C2 (Rack 1 Chassis 2) upstream u1

5. R2C1:D2 -> R2C3:U1;

There is a connection from R2C1 (Rack 2 Chassis 1) downstream d2 to R2C3 (Rack 2 Chassis 3) upstream u1

6. R2C1:D1 -> R2C2:U1;

   There is a connection from R2C1 (Rack 2 Chassis 1) downstream d1 to R2C2 (Rack 2 Chassis 2) upstream u1

7. R1C1:D3 -> R3C1:U1;

   There is a connection from R1C1 (Rack 1 Chassis 1) downstream d3 to R3C1 (Rack 3 Chassis 1) upstream u1 . ... etc

These connections form a network topology that links various chassis across different racks, indicating how data or signals flow between them. The labels and ports help identify the specific points of connection, enhancing the clarity and readability of the diagram.

DOT File Validation:

Once you have the DOT file saved in the file system, for instance c:\tmp\topology_template.dot, you can use the following command to validate.

```
qcsconfigurecli compile topology path "c:\tmp\topology_template.dot"
```

This will compile and validate the topology file offline and generate a sample System_definition.yml file locally for user review. During validation common problems such as missing semicolons or mistyped identifiers will be reported as errors, with the line number that needs to be fixed.

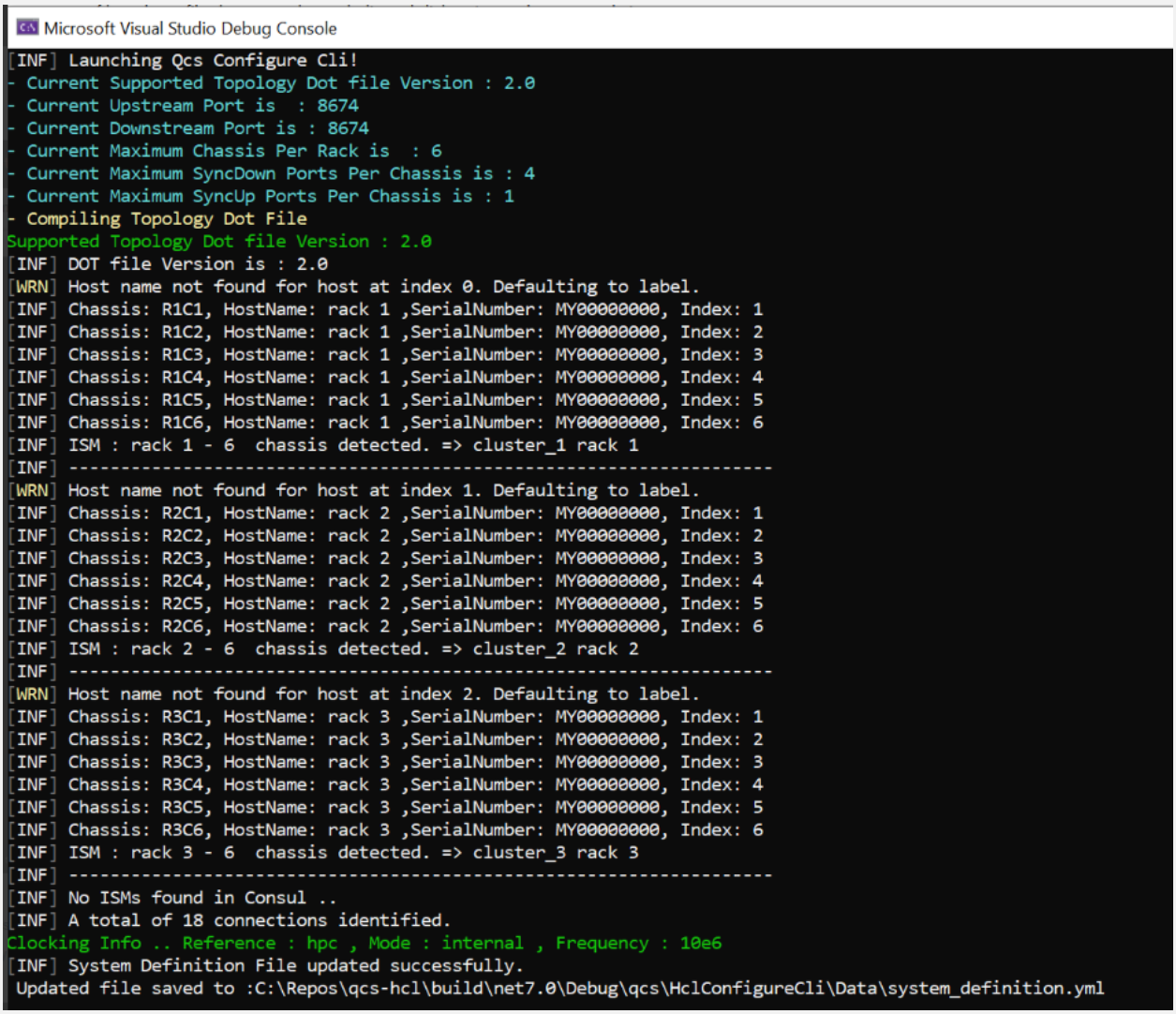

Figure 2. figure/caption goes here.

System Configuration

Once you are satisfied with the generated system_definition.yml file, you can proceed to system configuration.

System configuration is required when setting up the QCS system for the first time or when changes to system definition yml is needed. For a straightforward system configuration, the Auto Run Command simplifies the process by executing all the necessary configuration commands automatically. This command will handle the entire setup, including all the required configurations, and will prompt the user for any additional input as needed throughout the process.

The only parameter required by the Auto Run Command is the system topology file that you have created in the previous step. This file contains the details of your system's architecture, which the command will use to configure the system accordingly.

Auto Run Command

use the following syntax to run autorun command.

```
qcsconfigurecli.exe configure autorun path "c:\tmp\topology_template.dot"
```

In this command:

- qcsconfigurecli configure autorun: This initiates the auto-configuration process.
- path "c:\tmp\topology_template.dot": This specifies the path to the topology file that the Auto Run Command will use to guide the configuration.
- Optional the clocking Configurations mode external frequency 100e6

By providing the correct topology file, the Auto Run Command will proceed to configure the system in alignment with the defined architecture, ensuring that all necessary steps are completed with minimal manual intervention.

Running As: Administrator

The command must be executed with administrative privileges to ensure that it can make the necessary changes to the system configuration.

```
> qcsconfigurecli.exe configure autorun path "c:\tmp\topology_template.dot"
[INF] Launching Qcs Configure Cli!
- Current Supported Topology Dot file Version : 2.0
- Current Upstream Port is  : 8674
- Current Downstream Port is : 8674
- Current Maximum Chassis Per Rack is  : 6
- Current Maximum SyncDown Ports Per Chassis is : 4
- Current Maximum SyncUp Ports Per Chassis is : 1

--------------- HCL Consul Configurations ---------------
Running As : Administrator

Choose Network option
1. Auto Discovery of Network Devices
2. Use Pre-configured List of Devices
3. Clear Existing Consul Server Configurations
4. Configure pgBouncer as Database Proxy (Configuration Completion Required)
5. Exit Discovery and Continue Configuring
Enter your choice (1, 2, ...):
```

Figure 3. figure showing autorun command execution

Step 1: Network Device Selection

You will be prompted to choose how the network devices should be identified and configured.

1. Auto Discovery of Network Devices: The system will automatically detect available network interfaces.
2. Use Pre-configured List of Devices: You can provide a list of devices that have been pre-configured.
3. Clear Existing Consul Server Configurations: delete all Existing Configuration in Consul
4. Configure Pgbouncer as Database Proxy (Configuration Completion Required)
5. Exit Discovery and Continue Configuring:

In this example, you choose Option 1 (Auto Discovery of Network Devices). This choice instructs the system to automatically scan and present a list of available network interfaces.

Step 2: Selecting a Network Interface

Once the network devices have been discovered, the system will display a list of available network interfaces. Each interface is associated with an IP address, which represents different network segments.

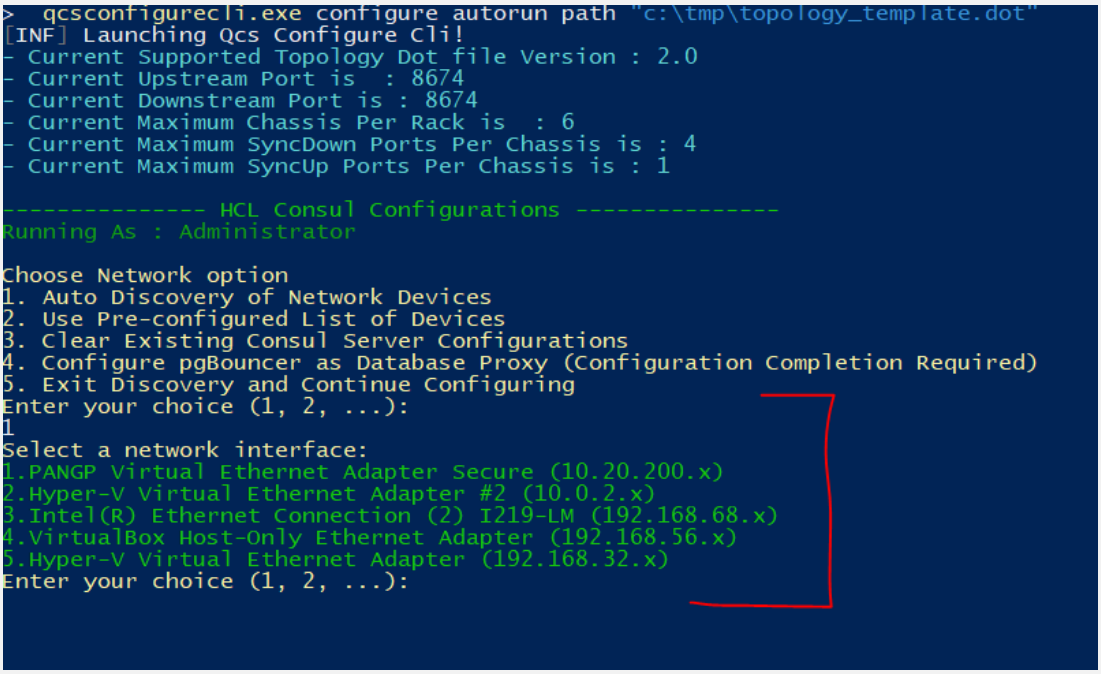

Figure 4. Example List of Interfaces.

Use Pre-configured List of Devices in the hosts.txt File

This option reads a file containing hostname and IP mappings and displays the available devices. The application will load the list of devices from a file named 'hosts.txt' located inside the 'Data' directory within the application directory.

The file must be placed in the 'Data' directory, which is inside the application directory. Data folder is located in C:\ProgramData\Keysight\Qcs\agent\Data

File Format Example

102.54.94.97,host1

102.54.94.98,host2

Ensure that the file is correctly formatted with each line containing an IP address and hostname separated by a comma. If the file is not found, the application will display an error message.

You will need to select the interface that corresponds to the network you want the system to use for communication.

Step 3: Subnet Confirmation and Private Link Configuration

After selecting a network interface, the system identifies the subnet associated with that interface. It then asks whether the identified subnet is a private link, which is crucial for configuring secure communications within the network.

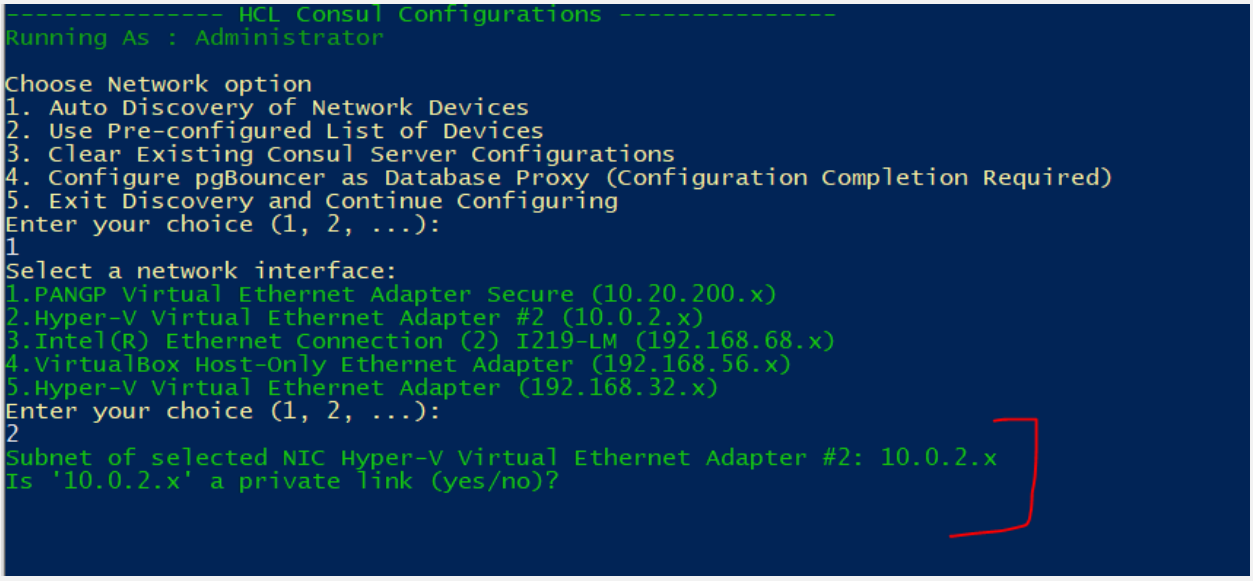

Figure 5. prompt to select private link.

Running the Network Discovery

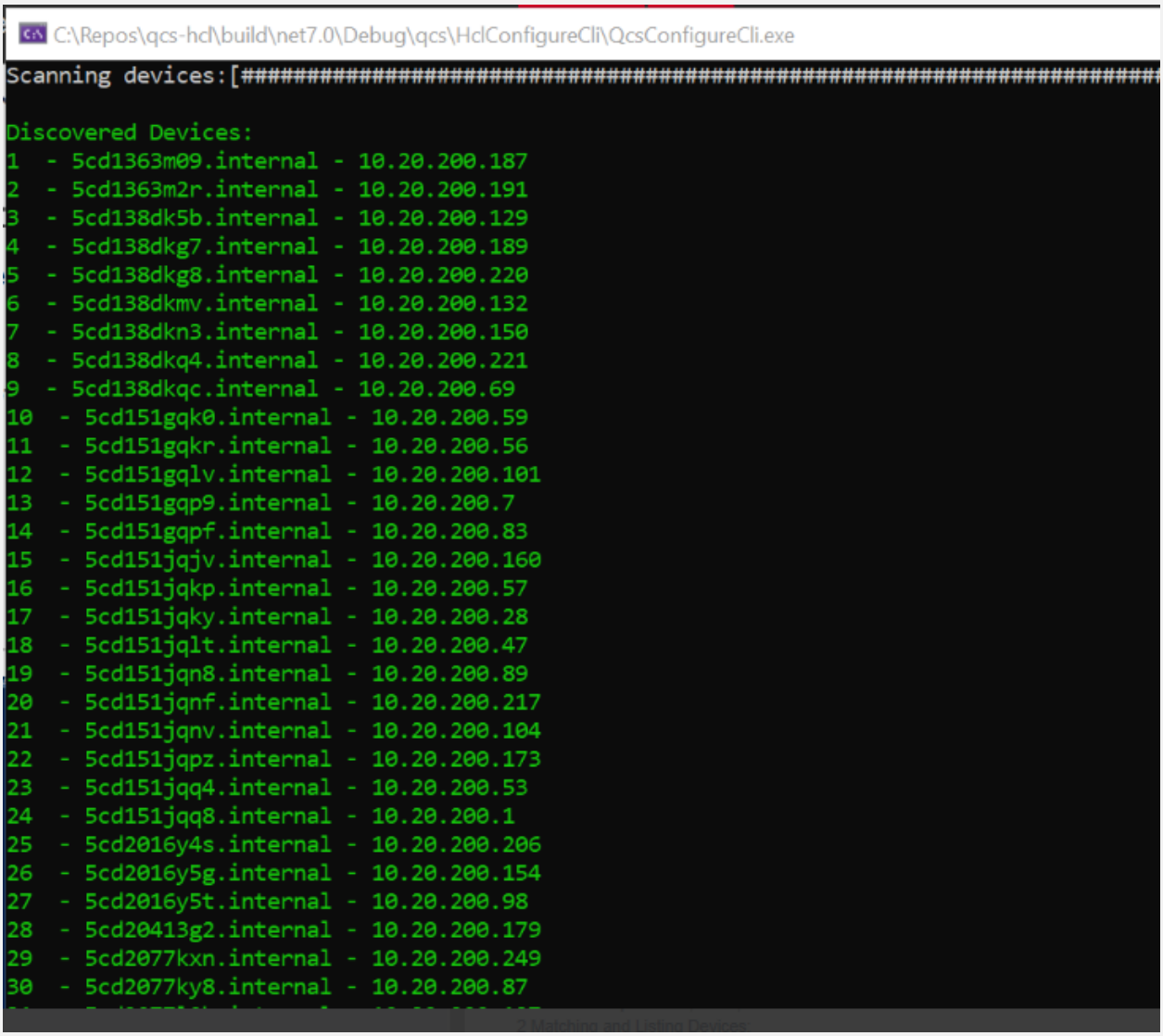

Figure 6. Screenshot showing discovered network devices.

Filtering Network Devices After Discovery

After discovering network devices, you can filter the list based on specific criteria. The filter allows you to enter multiple wildcard strings separated by commas. If any of the filter patterns match the devices, those devices will be listed. If none match, you will be prompted to try again.

How to Use the Filter?

1. Enter Filter Criteria:
    a. After the device discovery step, you will be prompted to enter filter criteria.
    b. Enter one or more wildcard patterns separated by commas.

For example: host*, *host, *host*

2. Matching and Listing Devices:
    a. The application will match the devices against the entered patterns.
    b. If any devices match, they will be displayed on the screen
3. Handling No Matches:
    a. If no devices match your filter criteria, the application will notify you and ask if you want to try again with different criteria.

        Example Usage

        Input: host*, *2

        Result: The application might display:

        1 - host1 - 102.54.94.97

        2 - server2 - 102.54.94.98

    b. If No Match:

        If no devices match the input, you will see a message - No devices matched the filter criteria. Please try again.

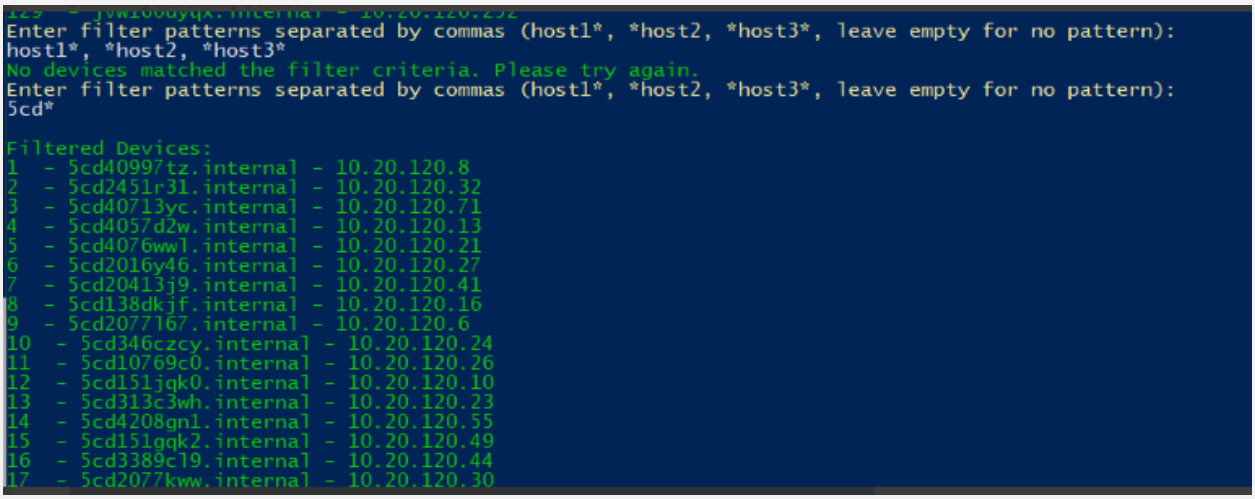

Figure Y. Filtering network devices.

After scanning, you'll need to match each discovered device with its corresponding component (e.g., Consul Agent, HCL, DB, ISMs). For example,

```
Enter your choice (1, 2, ...) for Consul server: 1

Enter your choice (1, 2, ...) for Database server: 1

Enter your choice (1, 2, ...) for HCL host: 1

Enter your choice (1, 2, ...) for ISM Leader host: 2

Enter your choice (1, 2, ...) (comma-separated list) for ISM Follower hosts: 3
```

The Consul server is crucial as it manages the service discovery and health checking for the infrastructure. Consul server resides on the main controller. Second question is to confirm to clean up an existing configuration in the Consul.

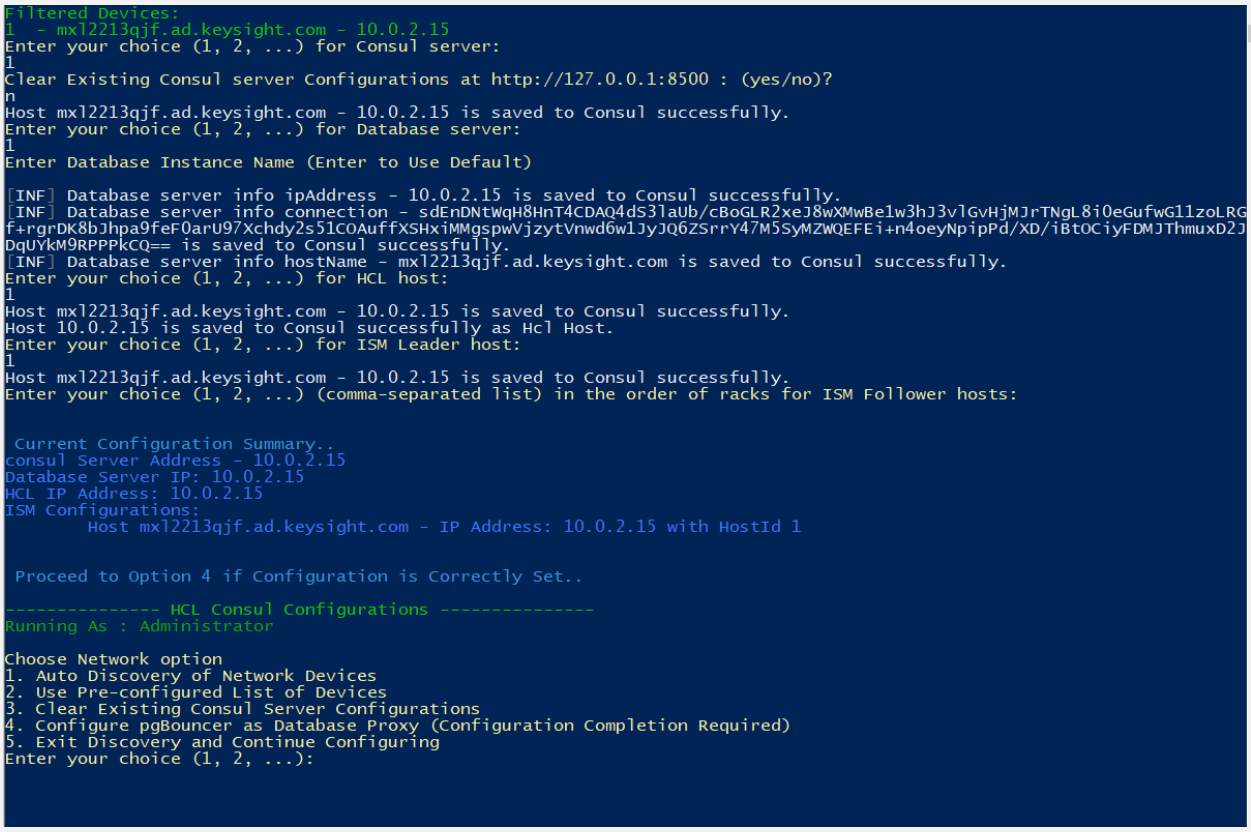

Figure 7. Screenshot showing device selection.

These steps in above screenshot demonstrate the sequential configuration of various components such as the Consul server, Database server, HCL host, ISM Leader host, and ISM Follower hosts. Each selection is registered within the Consul , ensuring that the infrastructure is correctly configured to function as a cohesive unit.

Last Step - Exit Discovery and Continue Configuring

User can exit from the above screen to Continue the configurations by selecting option 5.

After exiting, Configuration tool will run series of command to apply the configurations to all the hosts as shown in the screenshot below.

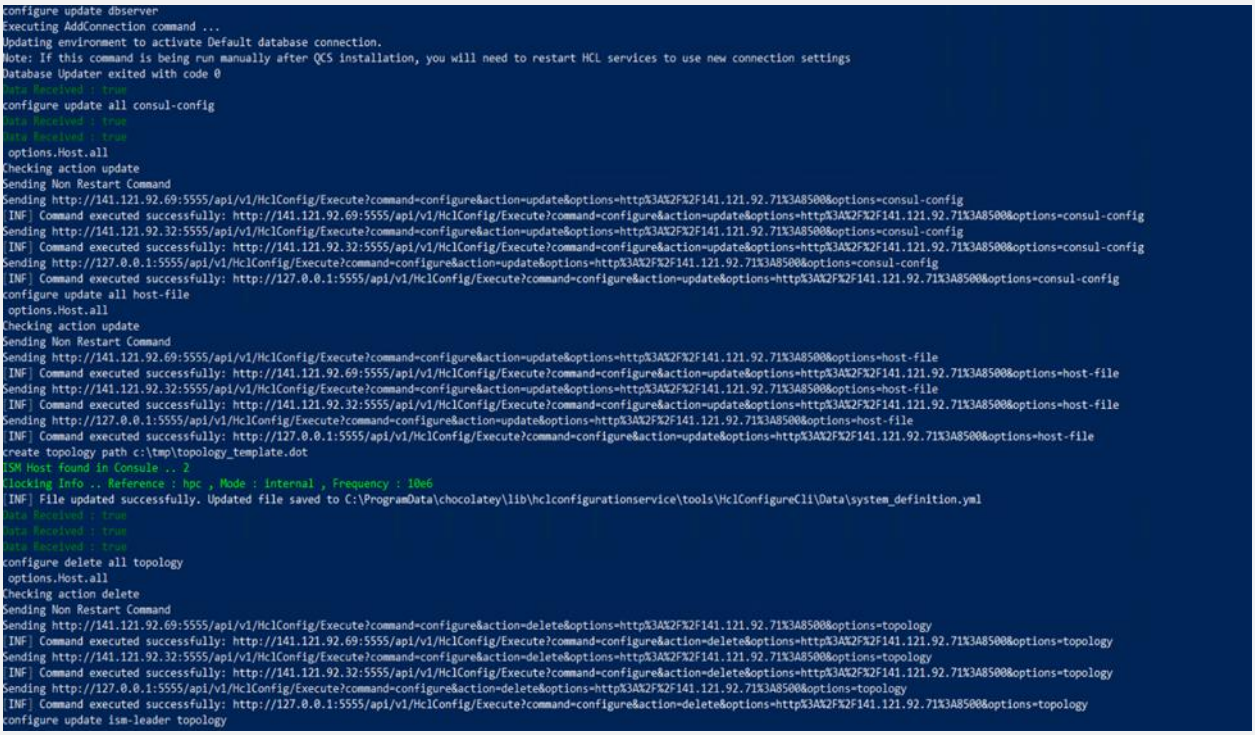

Figure 8. Configuration tool applying configurations to different hosts.

Setup the external IP for the database server

You will also be prompted to setup the external IP for the database server during this phase.

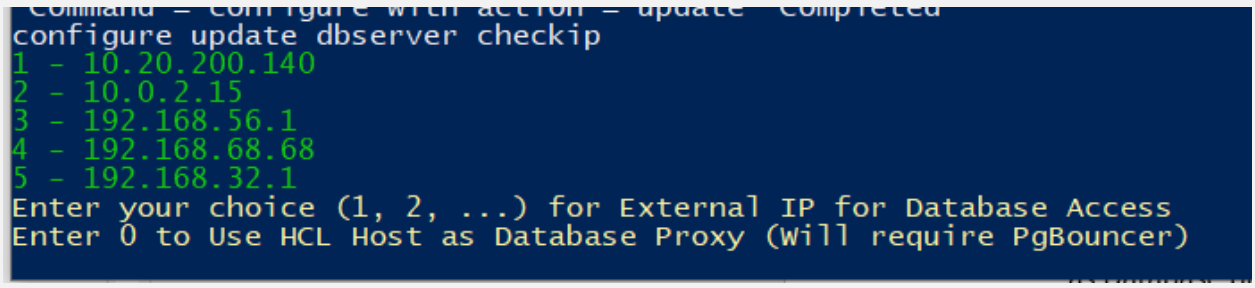

Figure 9. Prompt to choose IP address to be used by external application (SDK) for database access

On this prompt, you have option to either choose one of the displayed IP address as external IP for database access or enter 0 to use main controller as database proxy. This is discussed in next section. After this we are at the end of configuration.

Using Database Server with private IP

Operating Postgres on a database server with a private IP requires the user to run the configuration tool. Database server may be hosted on a Windows or Linux server. When the database server is a private IP, the HCL host uses Pgbouncer (which is installed via the 2025A TSM bundle) as a proxy to connect to that server. If the database server has a public IP available (such as through a dual network interface card), that public IP should be used instead of setting up the instructions below.

These instructions use a database server on a Redhat Linux, and the server does not have a public IP.

Locate the hosts.txt file in C:\ProgramData\Keysight\Qcs\agent\Data. Edit the file to include the IP address and host name of the private database server. Then run the configuration tool. During configuration, the tool will prompt you to select the external IP for Database access. Choose 0. This uses the main controller as a proxy to connect to the private database server.

- This is only required to allow SDK to connect to the Postgres database for remote program submission.
- If Database server has public IP, public IP should be used to connect to Database. You will get a prompt to select the IP address when Database host is being set by configuration tool

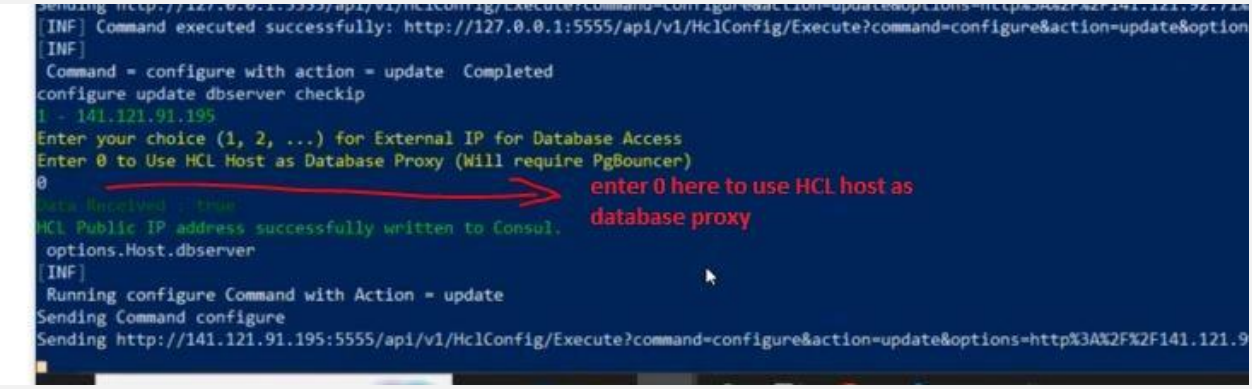

Figure 10. Selecting 0 to choose main controller as database proxy

- After that, the configuration tool will take some time to save and apply configurations and restart all the host machines.
- After the configuration is done, execute the autorun command again and enter option 4 - "Configure pgBouncer as DatabaseProxy"

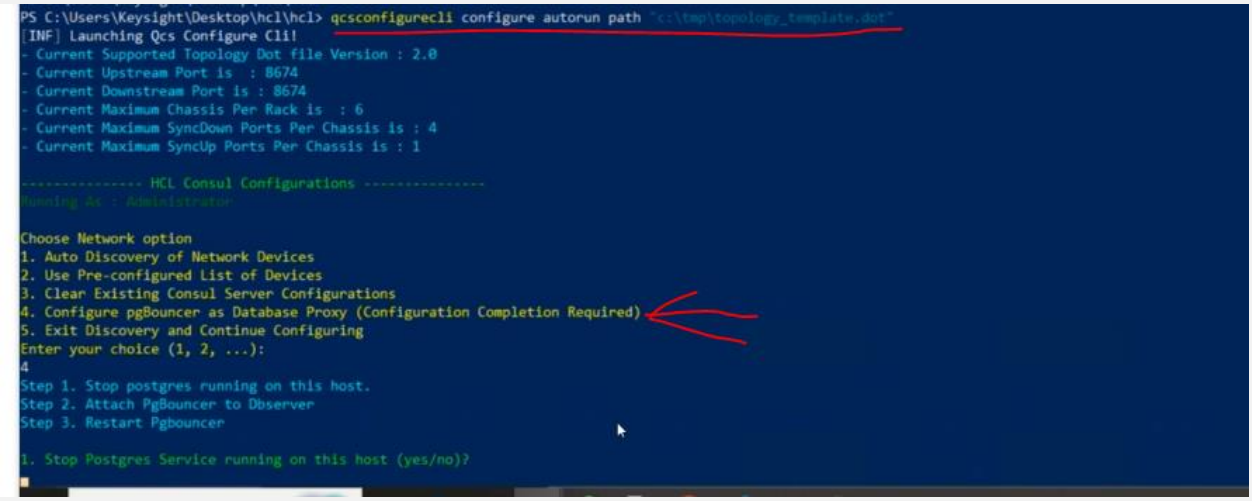

Figure 11. Execute autorun command again

- Step 1 asks if the user wants to stop any Postgres server running on the main controller. Enter "yes"

```
1. Stop Postgres Service running on this host (yes/no)?
```

Figure 12. prompt to stop postgres service.

- Step 2 asks if the user wants to attach Pgbouncer Service to Configured Database. This will reconfigure Pgbouncer and restart it. Enter "yes"

```
[INF] Service 'postgresql-x64-15' has been stopped successfully.
Stopped Postgres Service running on this host
2. Attach PgBouncer Service to Configured Database (yes/no)?
yes
```

Figure 13. Prompt to enable PgBouncer.

- Verify that the setup has succeeded in the Windows Task Manager on the HCL host. The Postgres service should be stopped and the Pgbouncer service should now be running.

Frequently used command

Update Database Server

```
qcsconfigurecli configure update dbserver
```

Restart ism services across the hosts

```
qcsconfigurecli service restart ism ism
```

Validate topology

```
qcsconfigurecli compile topology path "c:\tmp\topology_template.dot"
```

Network Discovery

```
qcsconfigurecli.exe run network-discovery
```

You'll be prompted to choose a network discovery option:

```
Choose Network option
1. Auto Discovery of Network Devices
2. Use Pre-configured List of Devices
3. Clear Existing Consul Server Configurations
4. Configure pgBouncer as Database Proxy (Configuration Completion
Required)
5. Exit Discovery and Continue Configuring
Enter your choice (1, 2, ...):
```

If you select option 1, you'll be asked to select a network interface and confirm if the selected NIC's subnet is a private link. The system will then scan the network and list discovered devices.

After scanning, you'll need to match each discovered device with its corresponding component (e.g., Consul Agent, HCL, DB, ISMs).

For example:

```
Enter your choice (1, 2, ...) for Consul server: 1
Enter your choice (1, 2, ...) for Database server: 1
Enter your choice (1, 2, ...) for HCL host: 1
Enter your choice (1, 2, ...) for ISM Leader host: 2
Enter your choice (1, 2, ...) (comma-separated list) for ISM Follower
hosts: 3
```

Consul Server Installed on host/IP No. 1 in the above list

Database server Installed on host/IP No. 1 in the above list

Database server Installed on host/IP No. 1 in the above list ... etc

Create Topology

```
qcsconfigurecli create topology path "c:\tmp\topology_template.dot"
```

Create the topology with a custom clocking configuration:

```
qcsconfigurecli create topology path "c:\tmp\topology_template.dot"  mode external frequency
100e6
```

Create topology will update the content of system definition file on the Consul server. To see the generated system definition file without pushing to Consul server, use command specified in #3 Validate Topology.