CQ QuarkLink User Guide

Version V1.41

Table of Contents

1 Scope

Welcome to the ***Crypto Quantique (CQ) QuarkLink User Guide***. This user guide is designed to provide information for people who will use QuarkLink on a day-to-day basis. This document is designed to be read by any user of QuarkLink, as most users will have access to the features documented here.

2 What is QuarkLink?

QuarkLink is Crypto Quantiques' universal IoT security platform that uses advanced cryptographic techniques to integrate with a hardware root of trust to provide provisioning, onboarding and monitoring for easy scalability and reliable security.

Unlike other IoT security platforms whose vulnerabilities make scaling a risk, QuarkLink provides the following advantages:

Covers all device deployment stages
Users can provision, onboard and monitor devices from a single software platform. That's a rarity in the IoT security market.

Easy root-of-trust integration
It takes other IoT software platforms substantial effort to integrate with a given root-of-trust. QuarkLink integrates easily with any root-of-trust and connects thousands of devices to servers in minutes.

Eliminate complexity
OEMs, system integrators and even end-users can easily manage their IoT estate without any specialist expertise.

3 General Overview

QuarkLink is an IoT platform that has been designed to be agnostic. It is based on Docker and therefore can run using any containerised technology. Examples of containerised technology are Kubernetes or Docker Compose which can be run on either Linux or Windows server based systems.

Since QuarkLink is agnostic, it can be installed on any public or private cloud platform regardless of the underlying operating system or other dependencies. Users can install QuarkLink on-premises or on multi-cloud or hybrid-cloud environments.

Once installed, connected devices can communicate with their assigned QuarkLink and, over a secure communications channel, can exchange digital certificates and instructions to enable onboarding to the assigned cloud service provider and web application.

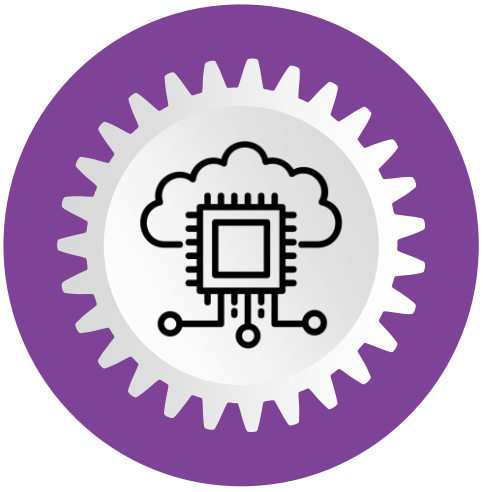
QuarkLink automatically creates the required Public Key Infrastructure (PKI) for a secure platform. A PKI ensures that the information being sent between entities is private and cannot be read and that the entity sending and receiving data is who they say they are.

QuarkLink manages the digital certificates used for identification. It can renew and revoke device certificates automatically.

QuarkLink provides a secure firmware download service through its firmware signing capability.

QuarkLink supports automated onboarding services for the following Cloud Service Providers:

- Amazon AWS
- Microsoft Azure (Hub & IoT Central)
- MQTT broker
- Database Direct



4 QuarkLink Resources (Cloud)

4.1 Hardware

Hardware resources can be divided into processing hardware and storage hardware. Processing hardware requirements for QuarkLink are dependent on the number and type of services that the QuarkLink is required to support. Large numbers of connected devices requiring simultaneous onboarding and a system wide software update could result in increases in hardware resources and should be considered when designing an IoT ecosystem. However, these are peak demands and will be handled automatically by the Kubernetes platform. Monitoring of the resources used by the QuarkLink over the lifetime of its deployment, with a view to alignment with the cloud service provider's pricing model, is required to ensure cost efficiency.

4.1.1 Compute

Current baseline QuarkLink instances are utilising the following compute hardware to support virtual machines:

Number of cores : 2
RAM static allocation : 8GB

It is recommended that the compute capability detailed above is a minimum allocation during early deployment where additional services such as logging and other debugging resources are needed for evaluation. The resources can then be reduced as the IoT ecosystem becomes stabilised and service requirement clearly understood.

4.1.2 Storage

QuarkLink instances do not rely on any Kubernetes (K8S) storage.

4.2 Software

QuarkLink requires the following cloud software services to operate.

4.2.1 Kubernetes Service Nodes

A QuarkLink instance requires 1 K8S node (virtual machine). It is recommended that an additional 2 nodes are utilised to support node outages and load balancing services.

4.2.2 IP Addresses

Typically, a QuarkLink instance requires 2 IP Addresses. One IP address for the QuarkLink URL (e.g. quarklink.io, quarklink.net) and one IP address for the standard transfer. An additional IP address may be used if the customer requires addition logging during commissioning of the service. IP address costs are typically low at around $1/month.

4.2.3 Load Balancers

A QuarkLink instance requires only one load balancer service. Most cloud service providers include a load balancer alongside a K8S managed service.

4.2.4 IoT Hub

There is no requirement for an IoT Hub service for the typical QuarkLink instance. However, during commission of a QuarkLink service it can prove to be useful for debugging and performance testing.

4.2.5 Logging

It is recommended that a QuarkLink instance is accompanied by a logging service. This can be used for long term performance testing and debugging.

4.2.6 Database

A QuarkLink instance requires a MongoDB database service. Many cloud providers provide a managed MongoDB service with varying pricing models, or this database could be managed by the customer. Crypto Quantique has experience with a variety of these options and typically makes use of the fully managed Atlas platform by MongoDB which is proving convenient. The storage requirements of Quarklink are not large and customer firmware images will likely be the biggest storage usage.

4.3 Cryptographic Technology

QuarkLink uses the latest cryptographic technology and is constantly automatically updated to use the latest algorithms and techniques. The currently technologies and algorithms utilised by QuarkLink are shown below:

- Cryptosystem - Elliptic Curve
- Supported curves - secp256r1
- Key length - 256 bits
- Public key certificate standard - X.509

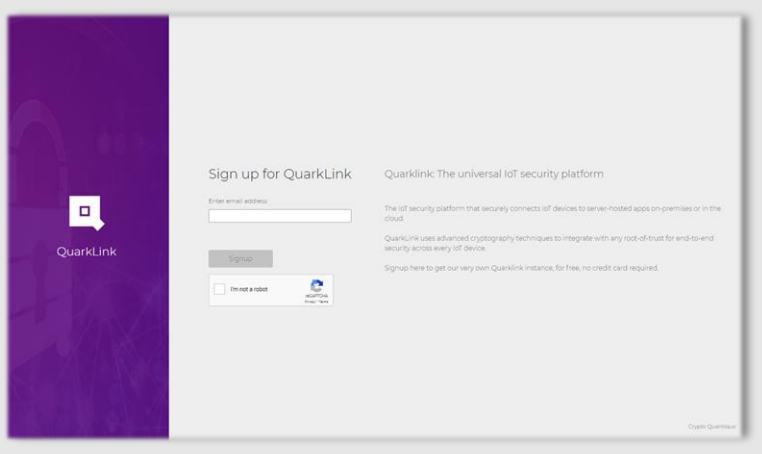
5 QuarkLink

QuarkLink is an IoT security platform which is designed to help users onboarding their IoT devices in a seamless and efficient manner. QuarkLink supports onboarding protocols for multiple cloud service providers and databases (see section 3) providing device connectivity and enrolment services, device certificate generation and provisioning, certificate renewal and revocation and firmware signing services.
QuarkLink will provide customers with efficient and non-complex onboarding services for their connected devices with minimal configuration.

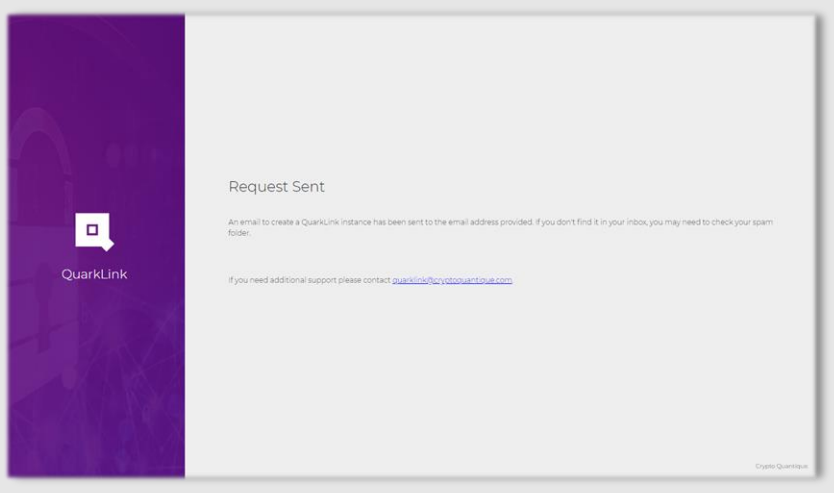
6 Signup

Access to a QuarkLink instance is provided through a browser interface.

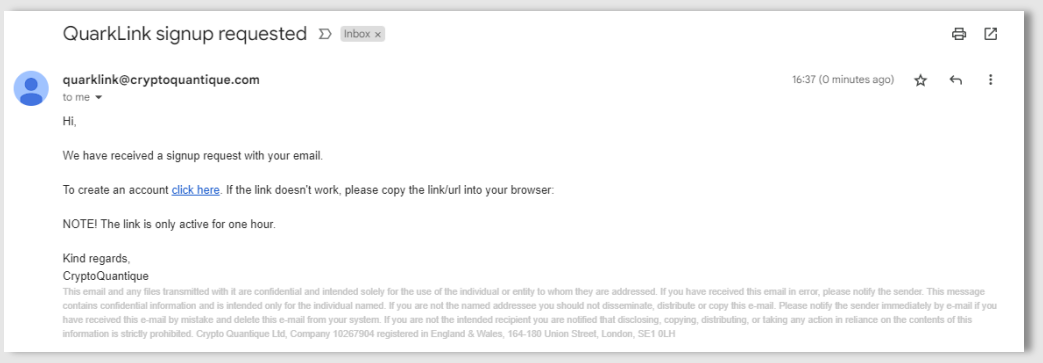
1. In a web browser, navigate to the QuarkLink Signup URL (e.g. https://signup.quarklink.io/).
2. The QuarkLink ***Login*** screen will appear as shown below:



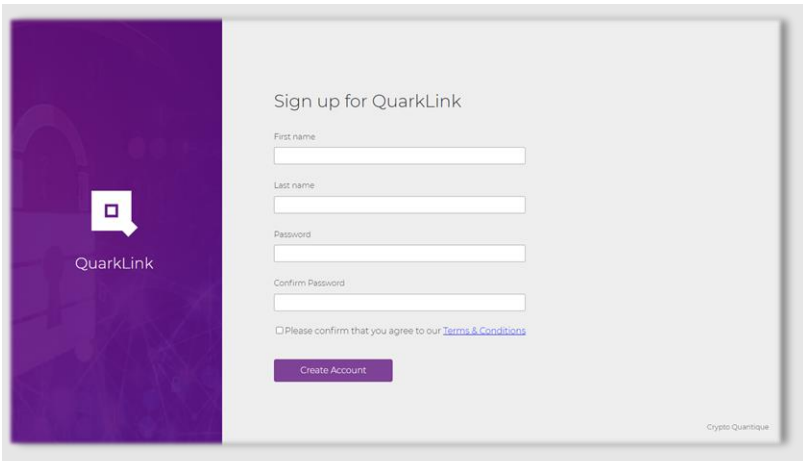
3. Enter a valid email address.
4. Click the “I’m not a robot” on the CAPTCHA. This will activate the ***Signup*** button.
5. Click the ***Signup*** button.
6. The dialog box shown below will be displayed.



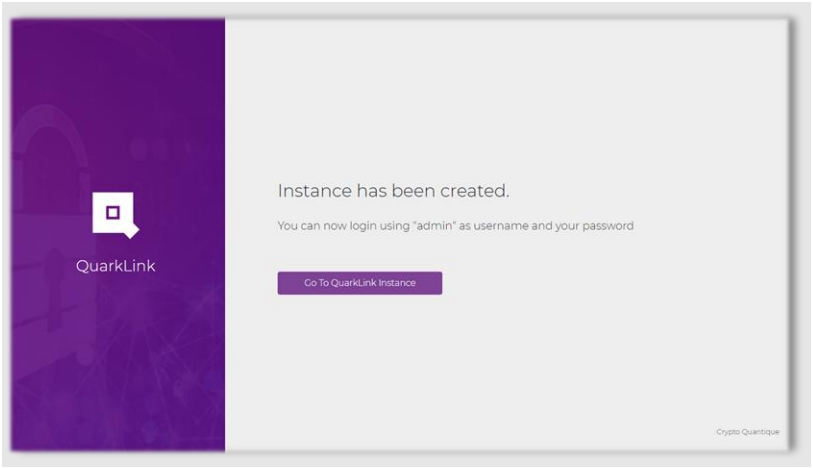
7. The user will receive an email with a link that allows sign up for a QuarkLink. An example email is shown below.



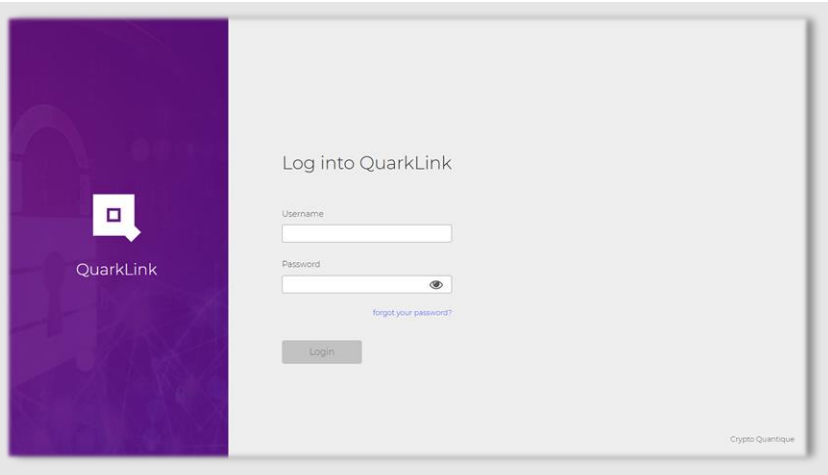
8. Click on the link provided in the email and the user will be transferred to the URL as shown below:



9. Complete the ***Sign up for QuarkLink*** registration and click the ***Create Account*** button. The dialogue box below will be displayed.



10. Click on the ***Go To QuarkLink Instance*** button and the dialogue box shown below will be displayed.

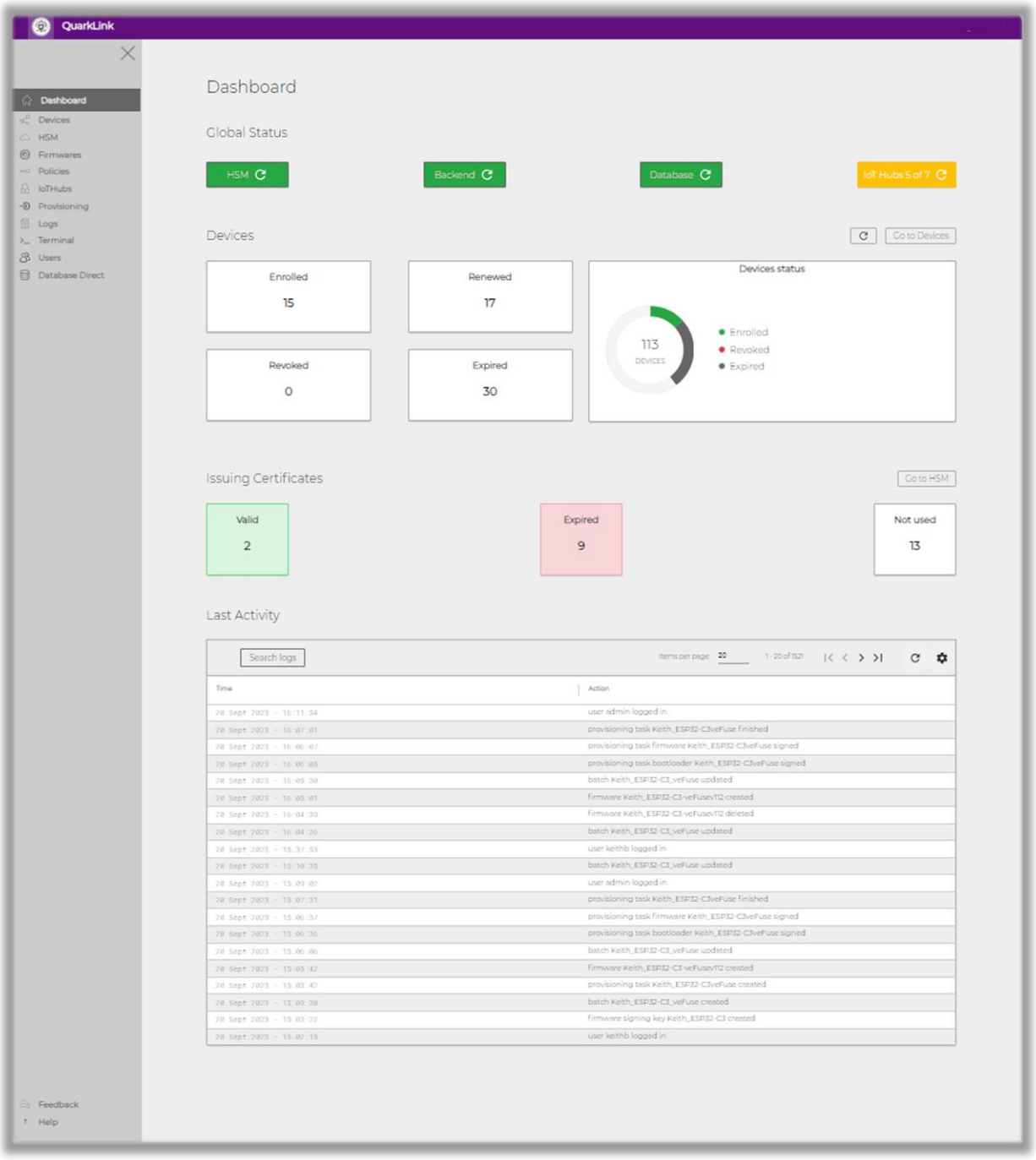


11. Log into the QuarkLink instance using the password entered in step 9 and the dashboard of the new QuarkLink instance will be displayed (see section 7).

Note : Two factor authentication is enabled by default when registering for a new QuarkLink. Please review the Crypto Quantique YouTube website for demonstration videos on how to use the two factor authentication (https://www.youtube.com/@CryptoQuantique).

7 Dashboard

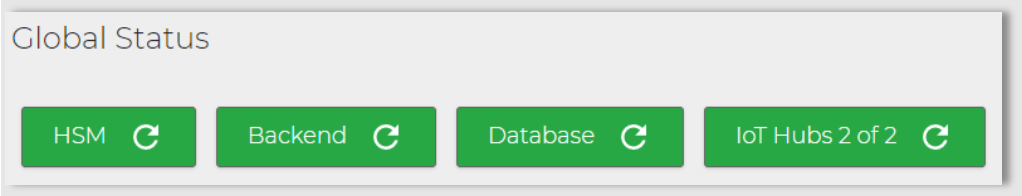
Once the user has logged into their QuarkLink instance the ***Dashboard*** will be displayed (see below). The ***Dashboard*** provides an at-a-glance view that shows a snapshot of the status of the QuarkLink. In the following sections we will describe the items displayed and their usage.



QuarkLink Dashboard

7.1 Global Status

The ***Dashboard*** provides real-time status of associated servers and functions using a simple colour coded system. The following status panels are shown:



7.1.1 Hardware Security Module

HSM – QuarkLink makes use of a cloud based Hardware Security Module (HSM) which is used to generate Public Key Infrastructures (PKI) using cryptographic keys and certificates. If coloured GREEN, the HSM is fully operational. Click on the icon to refresh the status. If coloured RED, QuarkLink is unable to contact the HSM. Contact your company IT department for additional help.

7.1.2 Backend

Backend – The QuarkLink Backend is the server that handles the device enrolment, management and communication with the remote IoT devices. If coloured in GREEN then the backend server is up and running with no problems, if it is coloured RED then there is a problem. Contact your company IT department for additional help.

7.1.3 Database

Database – All QuarkLink instances require a database to store critical information. If coloured GREEN there is a secure communication channel between the QuarkLink and its associated database. Click on the icon to refresh the status. If coloured RED, QuarkLink is unable to contact its database. Contact your company IT department for additional help.

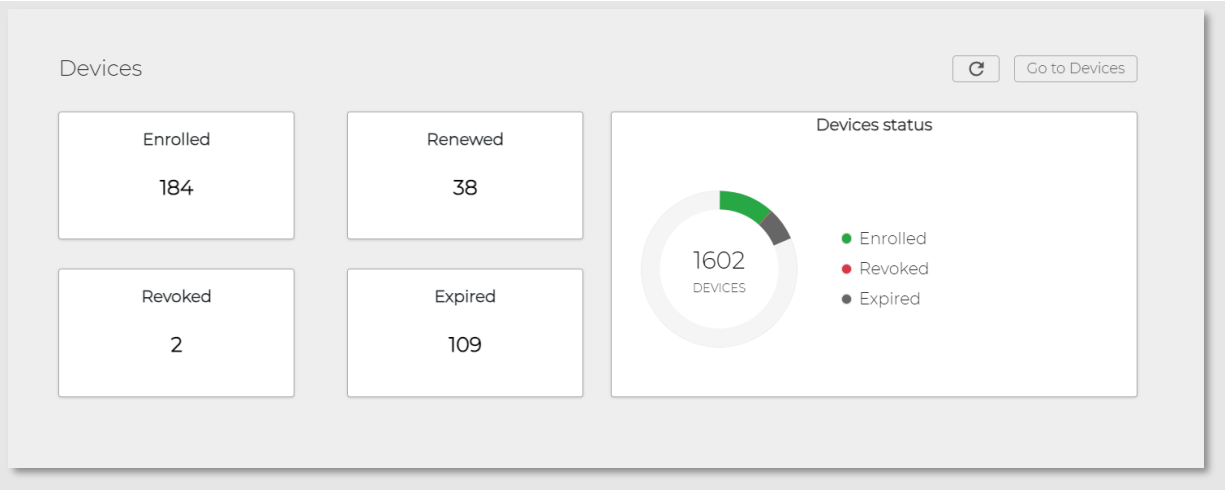
7.1.4 IoT Hubs

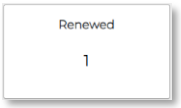
IoT Hubs – QuarkLink is required to make contact with IoT Hubs (AWS, Azure, etc) that have been setup in its configuration. If coloured GREEN, QuarkLink is able to make contact with all IoT Hubs that have been configured in this instance. Click on the icon to refresh the status. If coloured RED, QuarkLink is unable to contact all or some

of the IoT Hubs that have been configured. The status indicates how many of the IoT Hubs are contactable (e.g. 3 of 5 indicates 3 Hubs are contactable and coloured ORANGE). Hovering the mouse over the IoT Hubs button provides further details of the IoT Hub activity. Contact your company IT department for additional help.

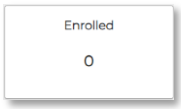
7.2 Device Status

The ***Dashboard*** provides status information regarding the IoT devices that are configured for this instance of QuarkLink. The following information is shown:

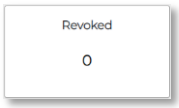




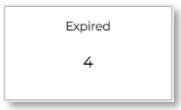
Renewed – The number of devices (identities) that have had their certificates renewed.



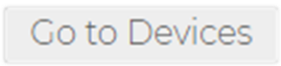
Enrolled – Enrolled devices are those which had connected to this instance of QuarkLink and have been recognised as a valid device. The device will have been connected to the QuarkLink and issues a valid device certificate.



Revoked – Revoked devices are those which had previously been successfully enrolled with this instance of QuarkLink but have since had their device certificates removed from the IoT Hub. This effectively disconnects the device from the IoT Hub it was associated with.



Expired – Expired devices are those whose certificate has exceeded the validity period.



The ***Go to Devices*** button provides the user with a shortcut navigation to the ***Batches*** configuration option of the QuarkLink. This button is equivalent to the ***Devices*** menu option on the left side of the ***Dashboard*** screen.

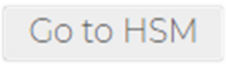
7.3 Certificate Status

The ***Dashboard*** provides status information regarding the digital certificates that have been generated by this instance of QuarkLink. The following information is shown:

Valid – The number of certificates that are currently within their expiration period.

Expired – The number of certificates that have exceeded their validation period.

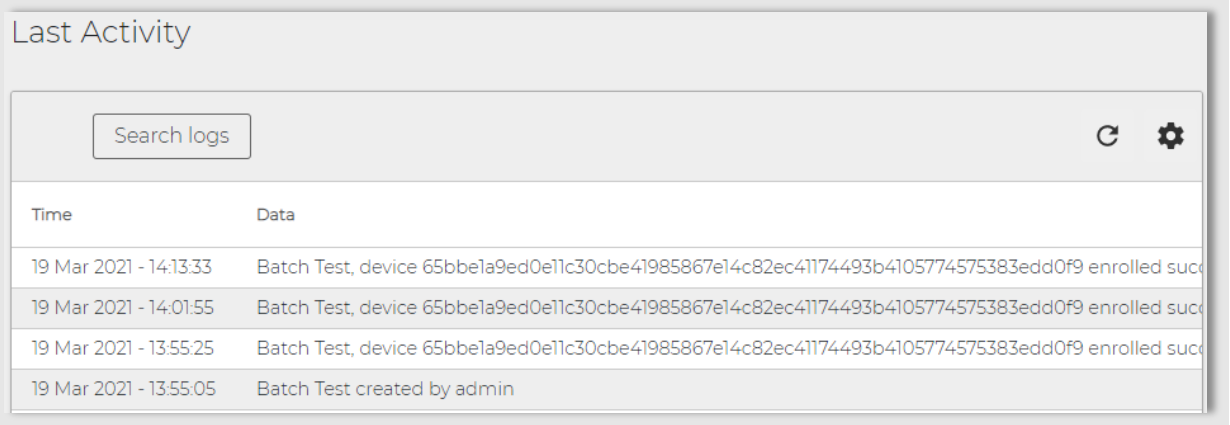
Not used – The number of certificates that have been created as part of a batch configuration but the device has not yet enrolled so the certificate has not been used.



The ***Go to HSM*** button provides the user with a shortcut navigation to the ***HSM*** display screen of the QuarkLink. This button is equivalent to the ***HSM*** menu option on the left side of the ***Dashboard*** screen

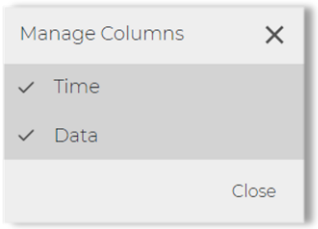
7.4 Last Activity

The ***Dashboard*** provides historical information regarding the activity that has taken place on this instance of QuarkLink. The first 10 entries are displayed. If additional log information is required click on the ***Logs*** link in the left hand menu list of the dashboard.



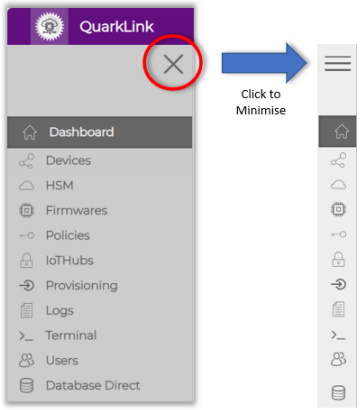
The ***Refresh*** button updates the last activity screen.

The ***Settings*** button allows the user to manage the columns that are required to be displayed in the ***Last Activity*** area (see below).



7.5 Menu Options

On the left side of the ***Dashboard*** screen are the global QuarkLink menu options (see below).



Dashboard — Return to ***Dashboard***

Devices — Add devices to a ***Batch*** (see section 8)

HSM — Examine PKI, certificates, keys and tokens (see section 9)

Firmwares — Add firmware images in preparation for download (see section 10)

Policies — Add ***Policies*** to a ***Batch*** (see section 11)

IoTHubs — Add ***IoTHub*** credentials (see section 12)

Provisioning — ***Provision*** (program) your IoT device with software (see section 13)

Logs — View ***Logs*** (see section 15)

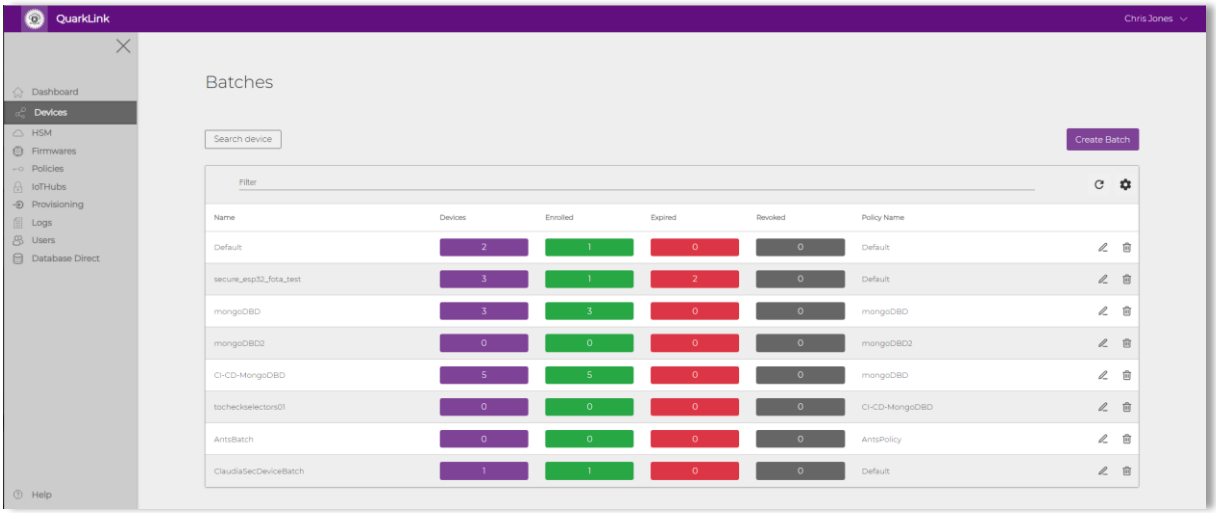
Terminal — Open the terminal emulator application (see section 17)

Users — Add ***Users*** to the QuarkLink (see section 14)

Database Direct — Configure for ***Database Direct*** connections (see section 16)

8 Devices

Devices refers to entities that are either unconnected, Enrolled or Revoked for this instance of QuarkLink. Devices are referenced by their identity which is a 256-bit number. QuarkLink requires information about the devices that are to be connected. Devices are grouped into ***Batches***. ***Batches*** can contain a single or multiple devices (identities). Device information is entered via the ***Create Batch*** dialog (see below).



A ***Batch*** of devices represents a group of devices that are to be connected to this instance of QuarkLink that have the same security ***Policy*** (section 11). Security policies can be created by clicking on the ***Policies*** link in the left hand menu list of the dashboard.

Creating a Batch: To create a new batch of devices click on the "***Create Batch***" button. In the dialog box that appears enter the following information:

Name - Enter a unique name for the batch.

Policy – Using the drop down menu, select a security policy. The list includes all those policies previously created under the ***Policies*** category. If a new policy is to be used for the batch, click the ***Create Policy*** button (see section 10). The security policy selected will be used for each device referenced in the batch.

Firmware – Using the drop-down menu, select the firmware that is required to be downloaded to the connected devices in the ***Batch***. Firmware updates are carried out on a ***Batch*** basis.
Firmware is uploaded to the QuarkLink via the ***Firmware*** page (see section 10). The Firmware update process to connected devices is triggered when the Batch ***Save*** button is clicked.

Select Signing Key – Use the drop-down menu to select the correct signing key that QuarkLink must use to verify the firmware that is to be associated with this batch (see section 9.3). If a signing key has not been defined, the user can create a signing key by clicking the "***Create New Key"*** button.

Filter by Attributes – Attributes are used to filter the types of firmware that are to be associated with the ***Batch***. This can be used to ensure that only firmware specific to the target hardware can be programmed into the ***Devices*** in the ***Batch***. Add attributes by specifying the name and value for each one. Clicking "Add New Filter" will allow the user to add more filter options. The "***Set Filters***" button will filter the firmware with the attributes matching the filters added (see section 10 for further details).

Add devices – There are two options available to add devices to the batch:

Devices – Type in the 256-bit identity of the device and then click "Add". Further devices can then be added individually. Once all device identities have been add, click "Save" to load the devices to the batch.

Upload File – Select a file that includes a list of devices that are to be added to the batch. This option is useful when large numbers of device identities are to be added. The format of the file is JSON script. An example file is shown below :

```
{"devices":[
"2fbdbebf711596348d85eb42fc02a1e392c4f3bcff841e27098f90fa14f7cce1",
"f016d74d4f9c3912be50cc5504b2d8cddfa7cb85a187b2169c5cb1d43c37eda9",
"73510f7a72304894e9263e1f14ea3ad7eb5fcaeeef47751ab34133fbfd611053",
"fb4831798f2af63cd9522d2d0767b3c7c3b215b3df6f44b9a5bd22cc9bffe6db",
"13321e68f019fe0a07943fd9ff07021b86d7d1b554034fa825f8bbe8a9fd212b",
"7da12f9318ca1cbd81d8e00d65cddd273a43238fe7527d09ad9c4b476e634ced",
"d557fb6992f9c4139ef514b22ef8f7f2cc5591dc948f4ae529f0a26140ba2b12",
"28bf78524d2b76b251e2925c896320e3ca9eb5d4d792fb5afd92bd653ad58eb4",
"e4be0f8c050ff0eb0dd34d475e03c0d0d59ebc6785e4d9fa3f1747651ab99042"
]}
```

Select the file (.txt) and click "***Open***" followed by clicking the "***Add File***" button. The file will then be loaded into the batch. Click "***Save***" to save the batch configuration.

Once a ***Batch*** has been created, devices with identities that match those included in the batch can be enrolled to the QuarkLink instance.

9 HSM

The Hardware Security Module (***HSM***) is an integral part of QuarkLink operation. It is responsible for creation of Public Key Infrastructures for the QuarkLink instance. The HSM generates cryptographic keys and certificates for use in a secure system. In this menu option, the certificates created/used by QuarkLink are displayed and can be downloaded by the user.

On installation of the QuarkLink a PKI is automatically generated. The Root certificate for the PKI instance is referred to as the ***OEM Root*** and is visible via the HSM menu option.

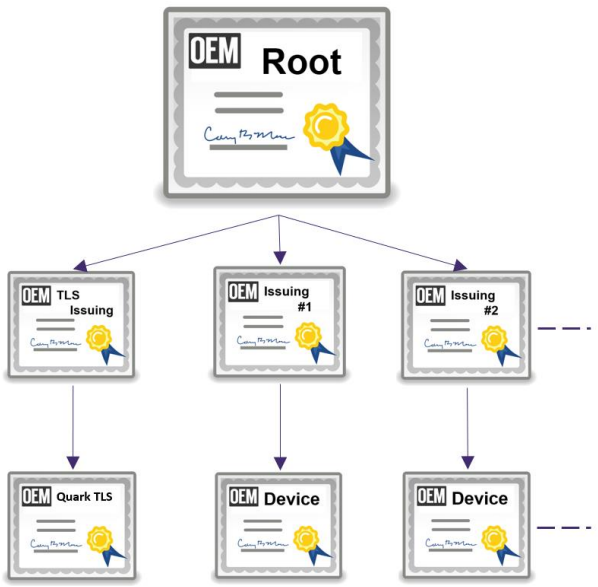
The HSM is cloud based but can be configured to use a customer specific HSM if required.

There are two types of certificates generated by the HSM:

Issuing – These are certificates that are used to issue device certificates. They form part of the chain of trust between the device certificate and the QuarkLinks root certificate.

TLS – These are certificates used when setting up a secure TLS communication between the QuarkLink instance and the device.

The diagram below shows a typical use of certificates in a QuarkLink instance :



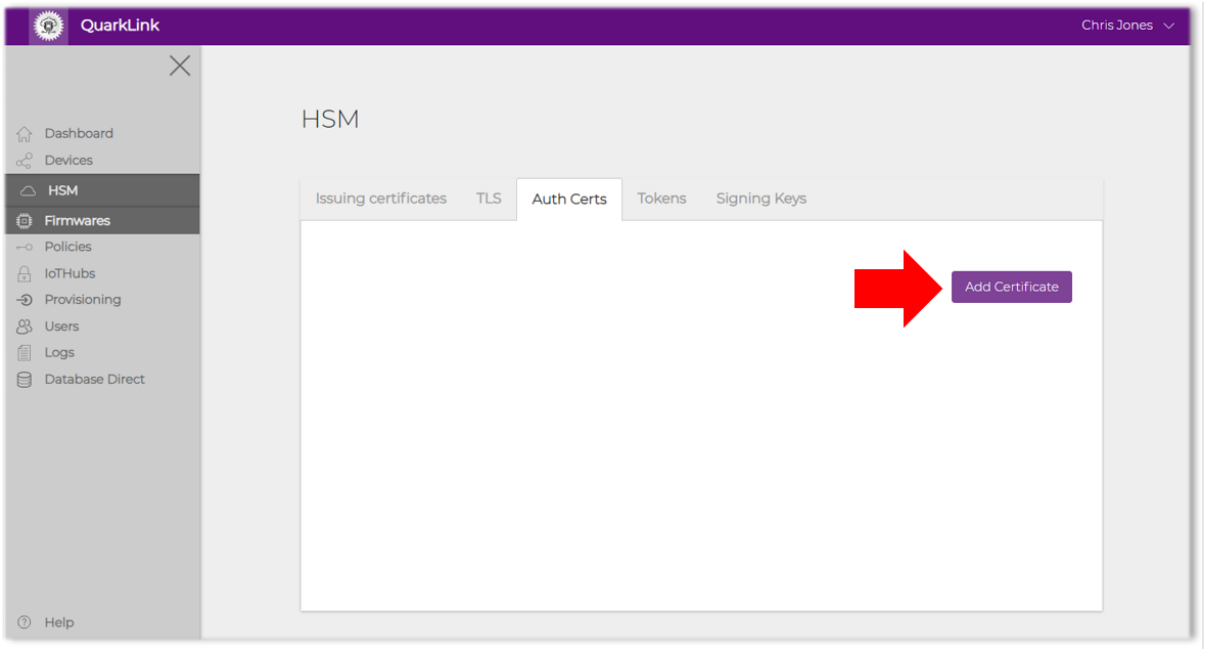
The ***HSM*** automatically generates cryptographic keys and certificates for use by QuarkLink. Users cannot create or manipulate keys and certificates. This menu option is only for information and access to certificate PEM files.

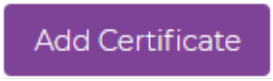
9.1 Authentication Certificates

QuarkLink has the ability to connect to multiple types of Roots of Trust. In cases where the Root of Trust is pre-provisioned with keys and certificates, QuarkLink includes the capability of utilising these pre-provisions keys & certificates to securely connect to the device. A typical case is where an IoT device hardware is supplemented with a pre-provisioned Secure Element as its Root of Trust.

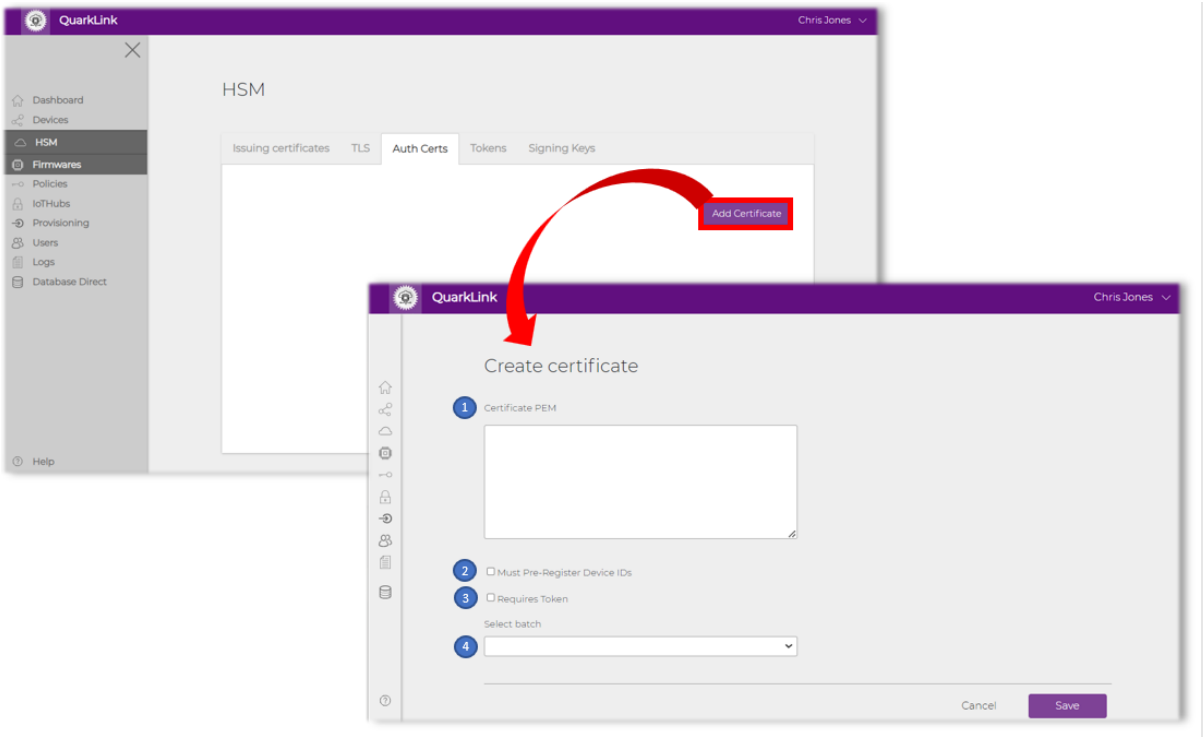
In cases where certificates are pre-provisioned into IoT devices, the authorised signer or Certificate Authority of the device certificate must be imported to the QuarkLink to allow authentication.

To add a CA certificate to the QuarkLink, click on the **HSM** menu and click on the "***Auth Certs"*** tab (see below).





Adding a Certificate – To add a new ***Certificate Authority*** certificate for use in enrolling devices with pre-provisioned certificates, click on the "***Add Certificate***" button. In the dialog box that appears there will be an option to import a certificate based on the PEM file format. Details of the CA certificate usage options available are shown below:



1. **Certificate PEM** – Paste the PEM file of the CA certificate to be used by QuarkLink to authenticate the connected device.
2. **Must Pre-Register Device IDs** – QuarkLink is to ignore any device requesting access to the QuarkLink unless its **DeviceID** is included in one of the configured ***Batches***.
3. **Requires Token** – QuarkLink is to ignore any device requesting access to the QuarkLink unless it has provided the correct ***Token*** (see section 9.2). In this option, the **DeviceID** will be ignored.
4. **Select batch** – Depending on the selection of options 1 or 2 above, the associated ***Batch*** is required to be selected.

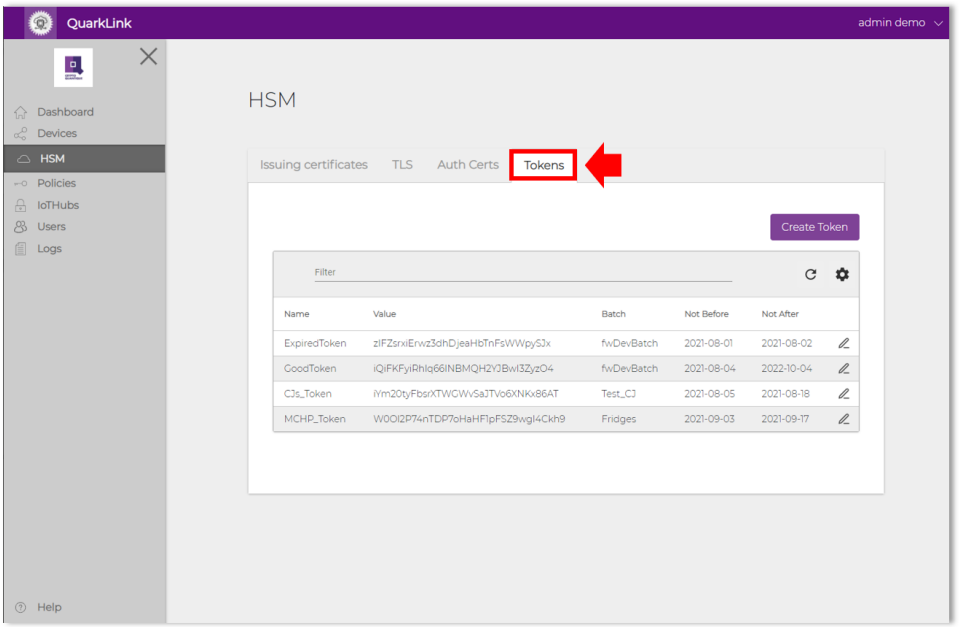
9.2 Tokens

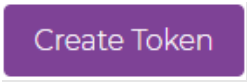
In cases where the Root of Trust is pre-provisioned with keys and certificates, QuarkLink includes an additional security feature which utilises tokens to securely connect to the device.

A typical case is where an IoT device hardware is supplemented with a pre-provisioned Secure Element as its Root of Trust but the customer wishes to further personalise their product and not rely on the generic certificate in the pre-provisioned Secure Element. A token can be included in the customer firmware to provide additional personalisation. The token is the same for each device that wishes to enrol with the QuarkLink. The token is used to ensure that the connecting device belongs to the right customer; otherwise, the device only authenticates to the Secure Element certificate and does not necessary ensure that the QuarkLink is connecting to the customer device.

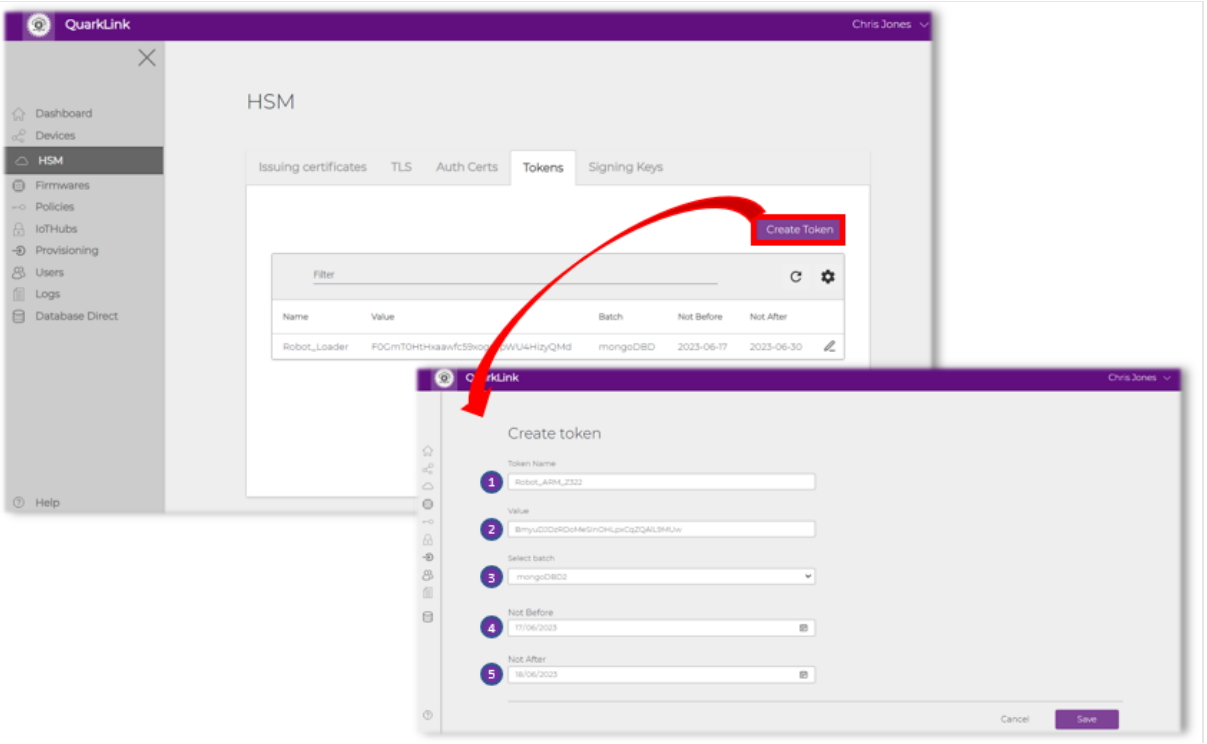
Tokens must be generated and saved in the QuarkLink prior to embedding in the customer firmware. This ensures that the token has sufficient entropy (e.g. be a 128 bit random string).

To create a Token, click on the **HSM** menu and click on the "***Tokens"*** tab (see below).





Creating a Token – To create a new ***Token*** for use in enrolling devices with pre-provisioned certificates, click on the "***Create Token***" button. In the dialog box that appears enter the following information:



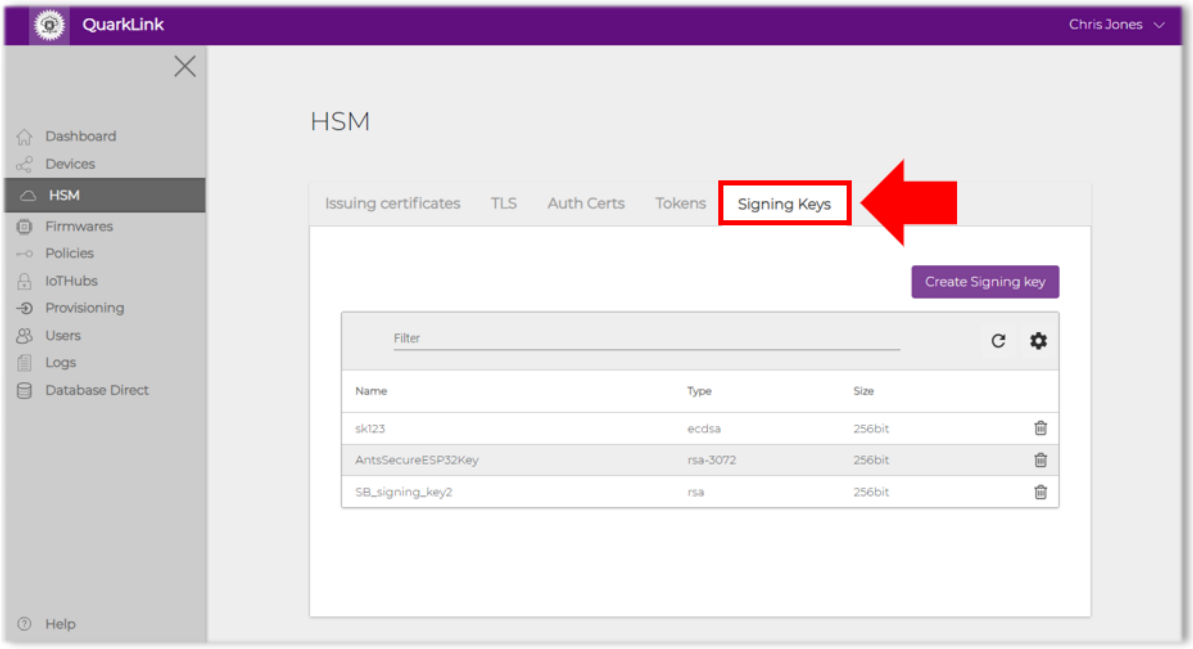
1. **Name*** – Enter a unique name for the ***Token***.
2. **Value –** The token is automatically created by QuarkLink and added here.
3. **Select Batch** – Use the drop down menu to select the ***Batch*** that is associated with this Token.
4. **Not Before** - Enter the date from which the token becomes valid.
5. **Not After** – Enter the date from which the token becomes invalid.

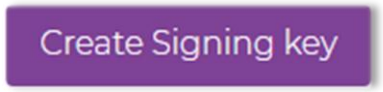
Click "***Save***" to save the new Token.

Once a ***Token*** has been created, devices with pre-provisioned certificates and the associated token can be enrolled to the QuarkLink instance. The token can be copied from the ***Value*** in the ***Create Token*** dialog box and embedded in the firmware of the device.

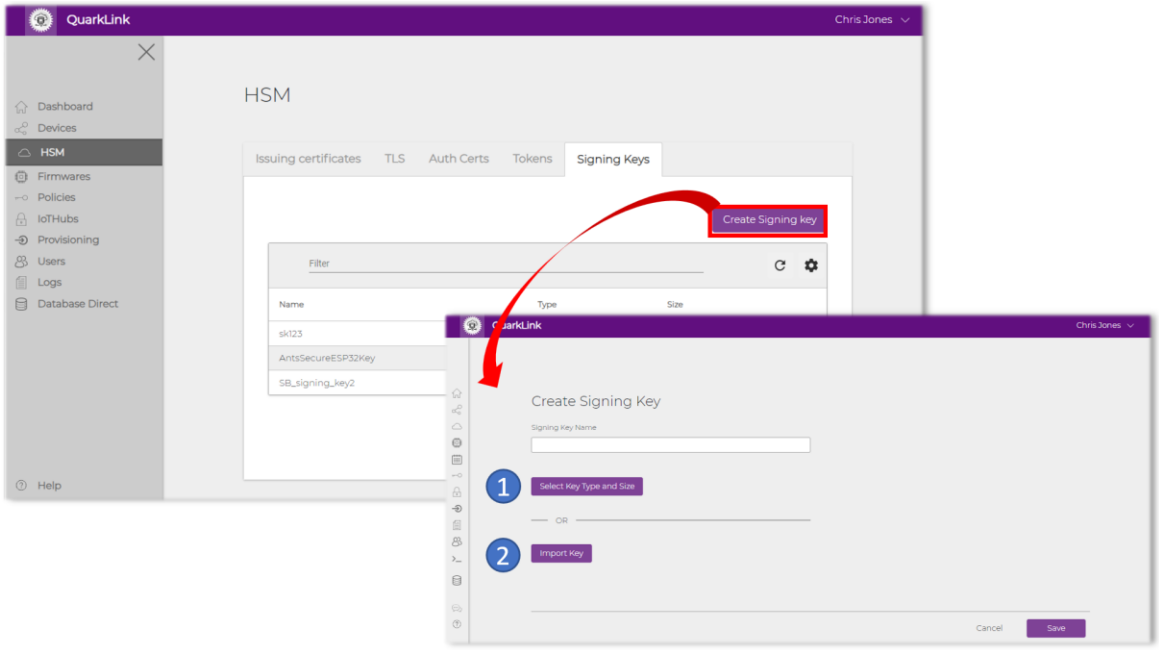
9.3 Signing Keys

QuarkLink includes a feature which allows the signing of firmware images that are to be used to download to connected devices. This facility is typically used in Firmware-Over-The-Air (FOTA) processes and part of the QuarkLink Provisioning function. QuarkLink has the capability to generate cryptographic keys needed to sign the firmware or import user keys for ease of product development. Click on the Signing Keys tab to enable QuarkLink to either generate signing keys or import them (see below):





Creating a Signing Key – To create a new ***Signing Key*** for use in firmware updates to connected devices, click on the "***Create Signing Key***" button. In the dialog box that appears there will be two options available:

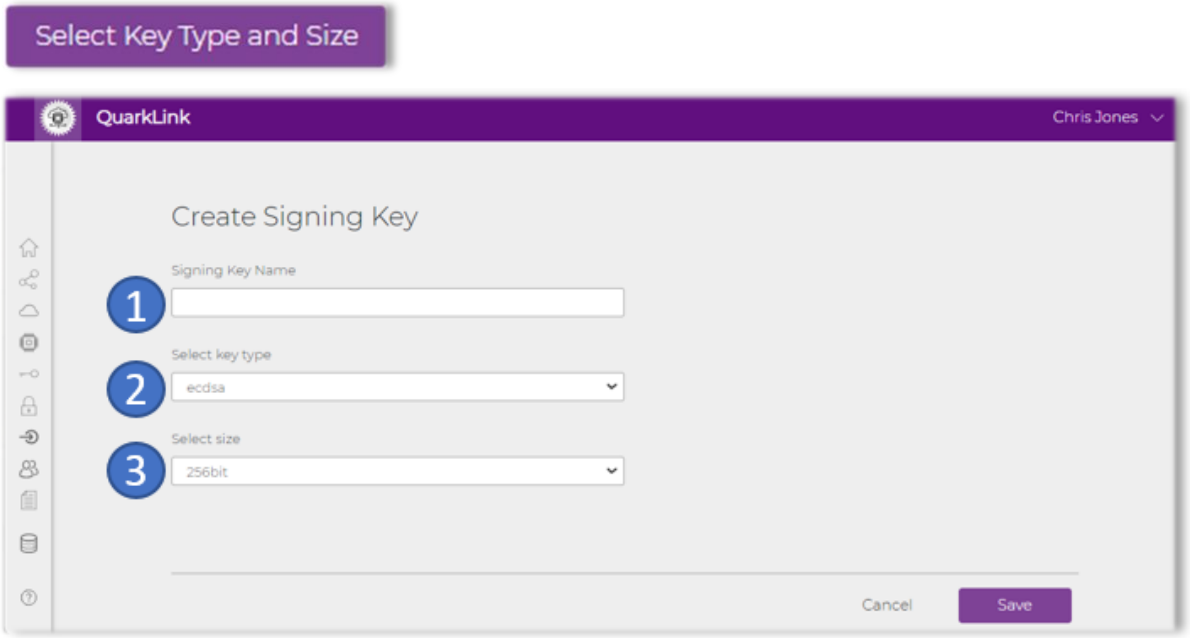


Option 1 is selected if the user requires a HSM generated key typically for use in production.

Option 2 is selected if the user is developing a product and wishes to import a non-secure key to be used for signing.

9.3.1 QuarkLink Generated Secure Key

Click the "***Select Key Type and Size"*** button for QuarkLink Generated Secure key generation. The following dialogue will be displayed:



A new signing key will require the input of the following information :

1. **Name*** – A unique name for the Signing Key.
2. **Key Type** – The user will be prompted to select a key type from a drop down list. The user can select either ECDSA[1] or RSA-3072[2].
3. **Key Size** – The user will be prompted to select a key size from a drop down list. The minimum specification will be a 256 bit key length

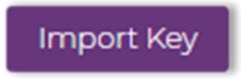
Click "***Save***" to save the new Signing Key.

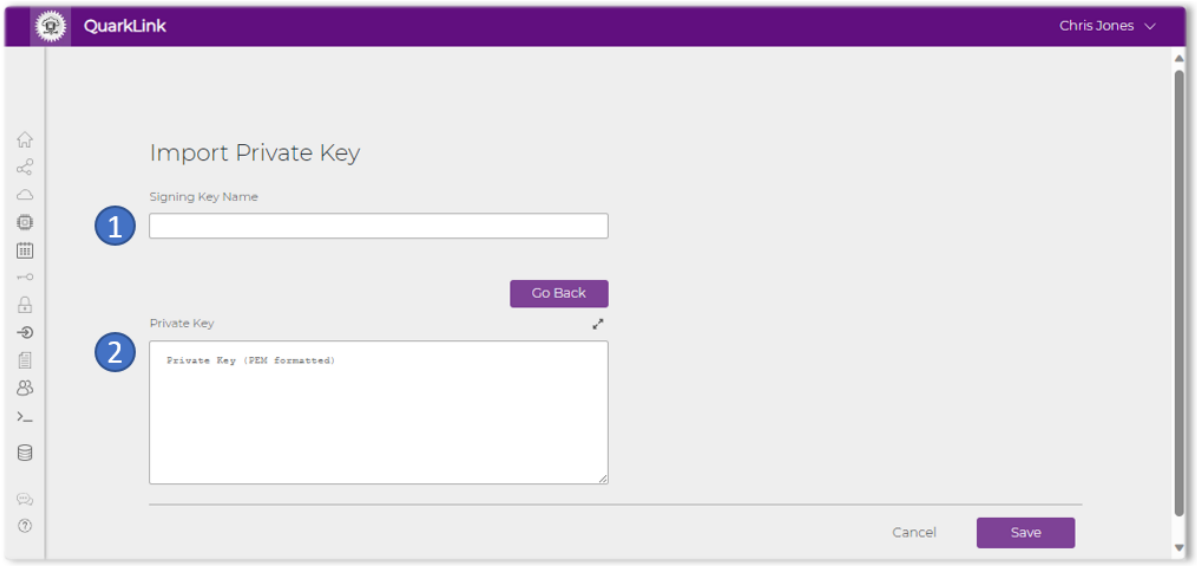
Once the user has input the information, QuarkLink will generate an asymmetric key pair using the HSM. The QuarkLink will display all ***Signing Keys*** generated by the QuarkLink instance and allow the user access to the public part of each of the ***Signing Keys*** (the private keys are not accessible by the user). The public part of the key pair will be used by the connected device to verify the firmware on delivery.

9.3.2 User Imported Signing Key

This feature is recommended for development purposes only as the imported key will be visible on the user's terminal during the import process. It is recommended that a QuarkLink generated signing key is used for production (section 9.3.1).

Click the "***Import Key***" button to import a user signing private key. The following dialogue will be displayed:





An imported signing key will require the input of the following information:

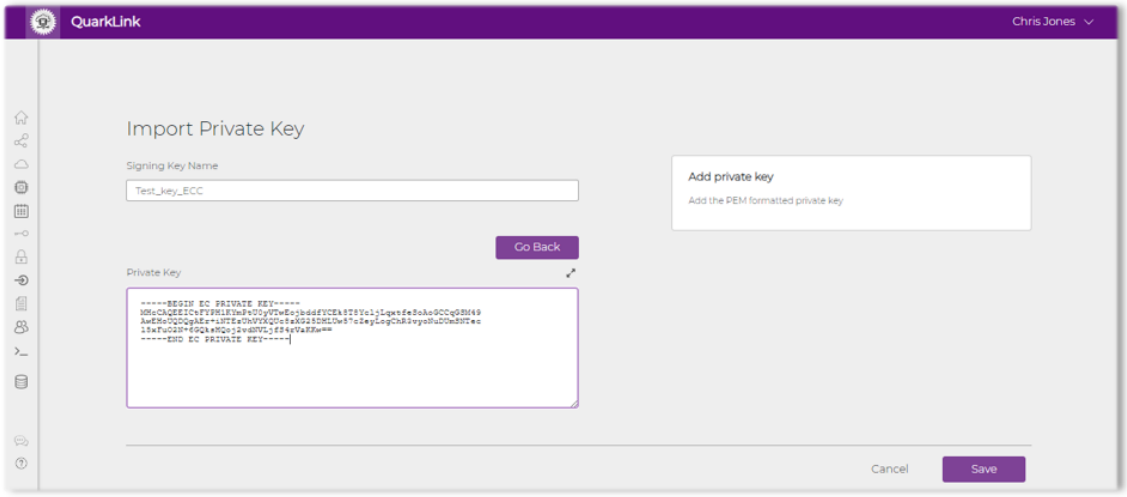
1. **Signing Key Name*** – A unique name for the imported Signing Key.
2. **Private Key (in PEM file format)** – The user will be prompted to import a PEM file for a private key type. The user can import either ECDSA[1] or RSA-3072[2].

Click "***Save***" to save the imported Signing Key.

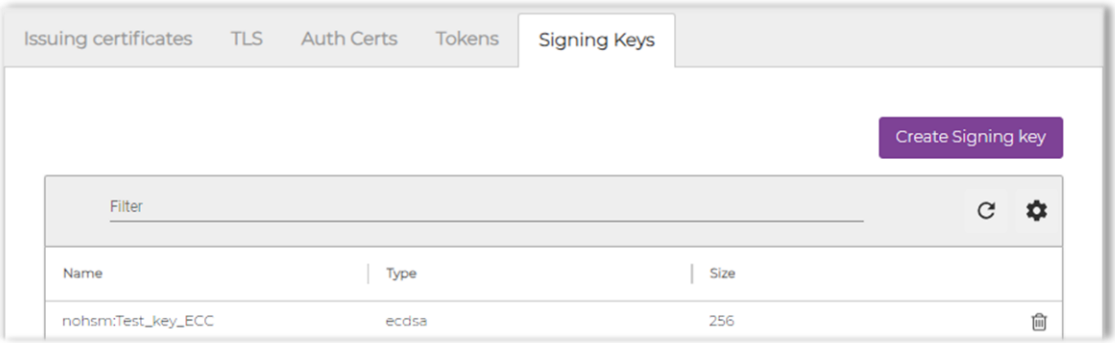
Note: A PEM encoded file includes Base64 data. The private key is prefixed with a "-----BEGIN PRIVATE KEY-----" line and postfixed with an "-----END PRIVATE KEY-----". An example is shown below:

```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDBj08sp5++4anG
cmQxJjAkBgNVBAoTHVByb2dyZXNzIFNvZnR3YXJlIENvcnBvcmF0aW9uMSAwHgYD
VQQDDBcqLmF3cy10ZXN0LnByb2dyZXNzLmNvbTCCASIwDQYJKoZIhvcNAQEBBQAD
...
bml6YXRpb252YWxzaGEyZzIuY3JsMIGgBggrBgEFBQcBAQSBkzCBkDBNBggrBgEF
BQcwAoZBaHR0cDovL3NlY3VyZS5nbG9iYWxzaWduLmNvbS9jYWNlcnQvZ3Nvcmdh
z3P668YfhUbKdRF6S42Cg6zn
-----END PRIVATE KEY-----
```

The dialogue box shown below is an example of importing a ECDSA private key:



Note: When keys are imported into QuarkLink they will appear on the HSM Signing Key list prefixed with the text “nohsm:” to indicate that the key is not a key that resides in the QuarkLink HSM (example below):



It is assumed that the user has access to the public key when utilising the import signing key function.

Signing Key are target hardware dependant. Please refer to the ***CQ Bootloaders*** application note for information on signing key types.

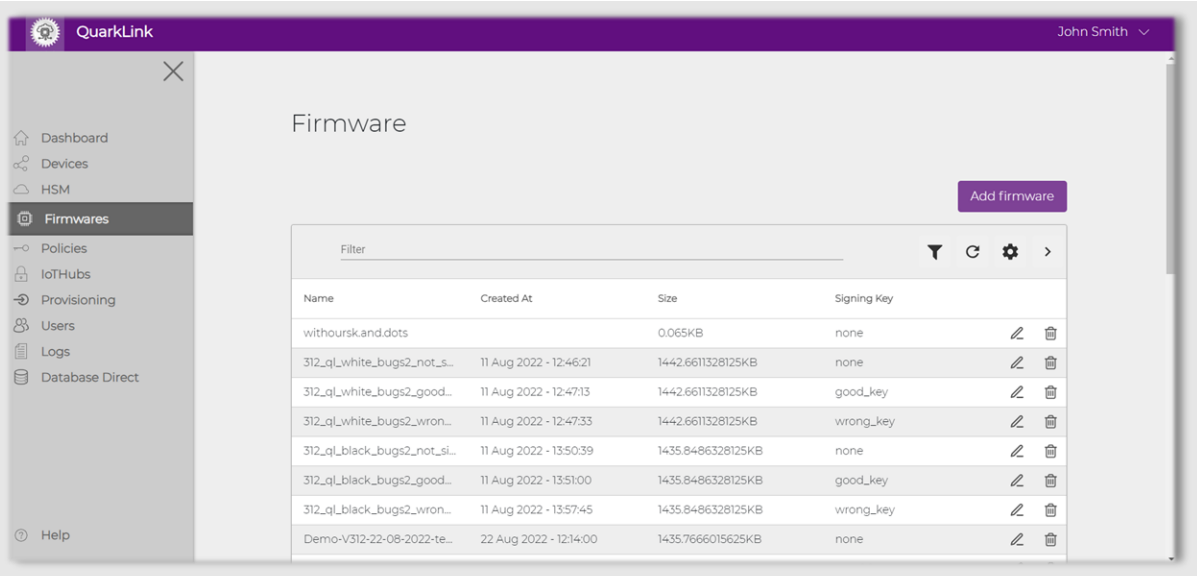
[1] The Elliptic Curve Digital Signature Algorithm (ECDSA) is a Digital Signature Algorithm (DSA) which uses keys derived from elliptic curve cryptography (ECC). It is a particularly efficient equation based on public key cryptography (PKC).

[2] Starting from May 28, 2021, 14:00 MDT (20:00 UTC), users are required to use 3072-bit RSA keys or larger for code signing certificates. This change is to comply with industry standards. These new RSA key size requirements apply to the complete certificate chain: end-entity, intermediate CA, and root. ECC key requirements however remain unchanged. Note that the QuarkLink provisioning function requires the use of 3078-bit RSA keys

- Code signing certificates issued before May 28 require no changes and will work until they expire.
- From May 28, all code signing certificates will require CSRs with 3072-bit or larger RSA keys.

10 Firmwares

QuarkLink has the capability to perform firmware updates to connected devices. Firmware images can be uploaded to QuarkLink in preparation for initiating a download process. Firmware can be uploaded via the ***Firmwares*** menu option. In the main ***Dashboard*** click on the ***Firmwares*** option in the left hand menu (see below).



All firmware uploaded to the QuarkLink instance will be displayed in the ***Firmware*** dialog box. The ***Firmware*** status dialog box will display all firmware images currently uploaded to the QuarkLink instance. Firmware can be deleted and edited.

Firmware images can be uploaded to QuarkLink in the following formats:

10.1 Binaries

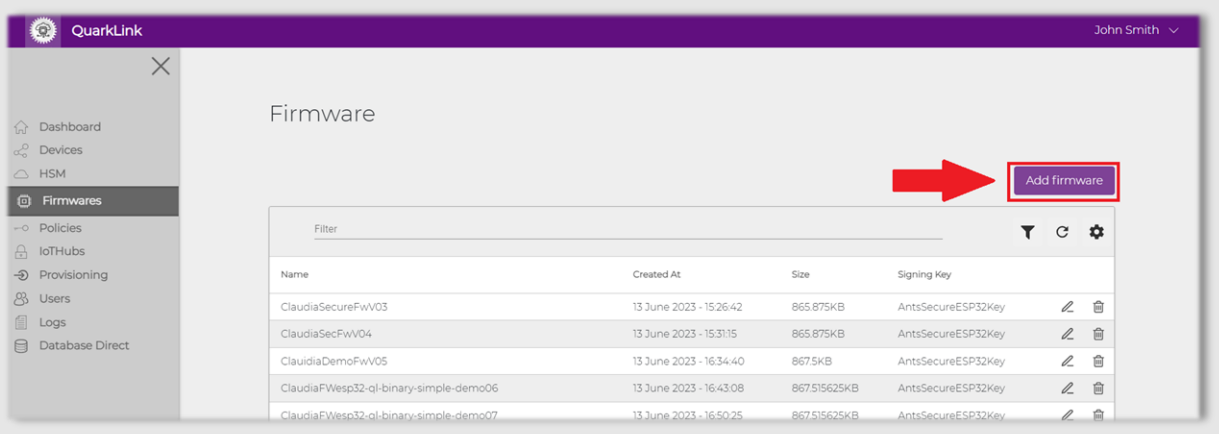
A binary file is a file whose content must be interpreted by a program or a hardware processor that understands, in advance, exactly how it is formatted.

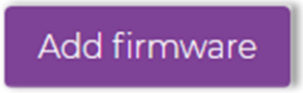
In general, executable (ready-to-run) programs are often identified as binary files and given a file name extension of ".bin". Programmers often talk about an executable program as a "binary". A synonym for this usage is object code. A binary file could also contain data ready to be used by a program.

10.2 Intel HEX

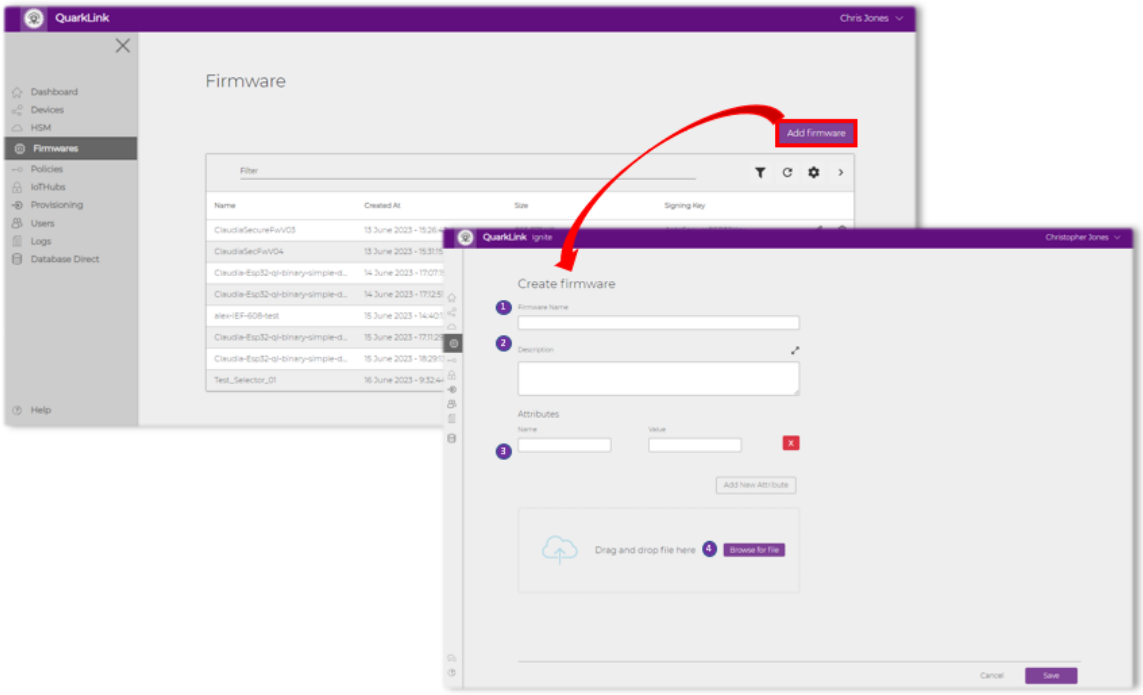
Intel Hex format is a standard layout for files produced by assemblers or C compilers when they compile source code. It is used by device programmers to program the target microcontroller with firmware. An Intel Standard HEX file is an ASCII file with one "record" per line.

To add firmware to the QuarkLink click on the "***Add firmware***" button. In the dialog box that appears enter the following information:





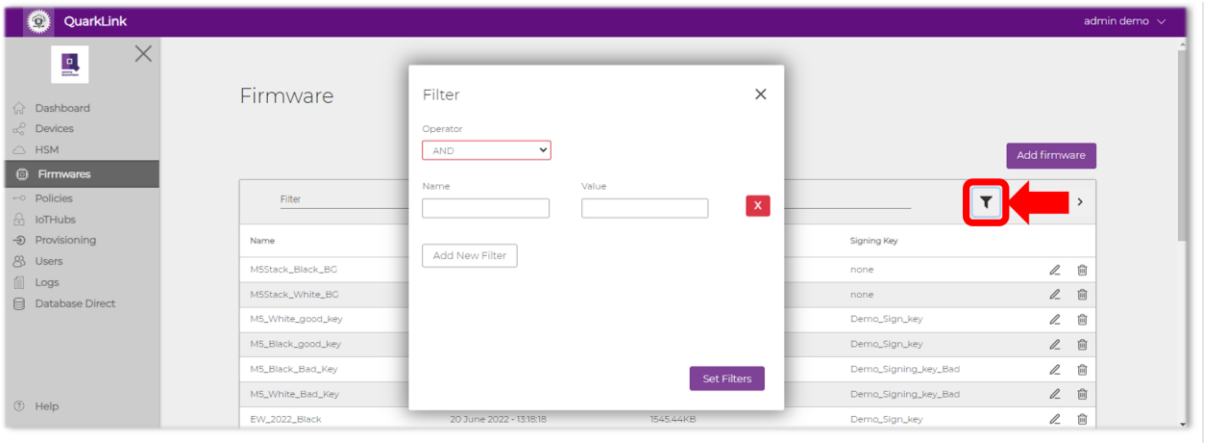
Adding Firmware – To add firmware to the QuarkLink instance, click on the "***Add firmware***" button. In the dialog box that appears enter the following information:



1. **Firmware Name** – Enter a unique name for the name of the firmware to upload.
2. **Description** – Enter a description of the firmware to be uploaded.
3. **Attributes –** Add attributes for the firmware specifying an attribute name and its associated value. Clicking "Add New Attribute" will allow to add more attributes.
4. **Browse for file** – Navigate to the firmware file to be uploaded.

Click "***Save***" to save the new firmware.

Filter Firmwares – Clicking the  button from the header of the table the ***Firmwares*** can be filtered adding attributes and selecting an “OR” or “AND” operator:

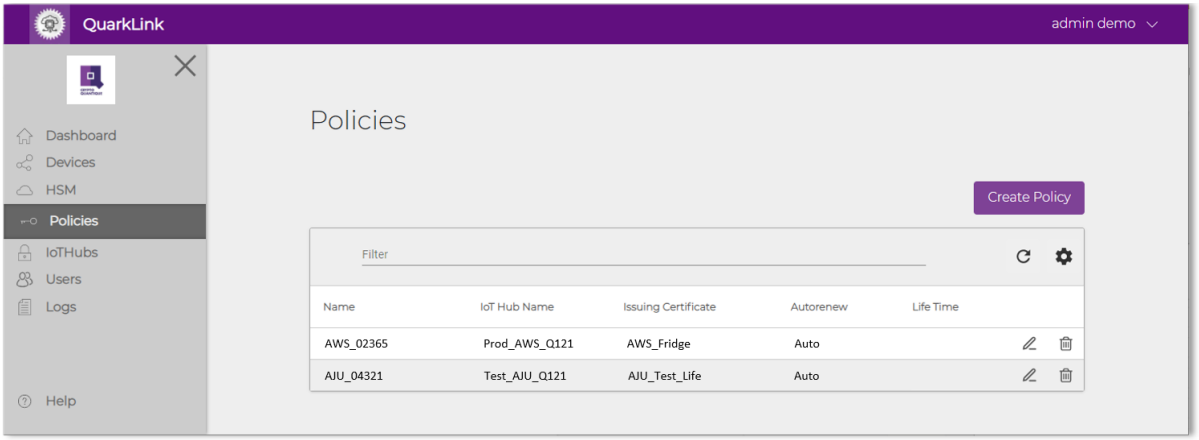


11 Policies

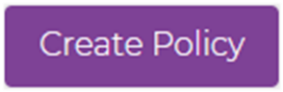
When configuring a device for connection to a cloud service provider a security policy is required. Generally, policies are described using JSON documents. Policies allow cloud service provider users to control access to services. The policy typically contains one or more policy statements. Each statement contains:

- Effect, which specifies whether the action is allowed or denied.
- Action, which specifies the action the policy is allowing or denying.
- Resource, which specifies the resource or resources on which the action is allowed or denied.

Please refer to the selected cloud service providers documentation for more detailed information regarding the contents and syntax of the security policy.



A ***Batch*** of devices requires a security policy which provides details of the IoT Hub access information and associated access control. Batches can be create by clicking the ***Devices*** link in the left hand menu list of the dashboard.



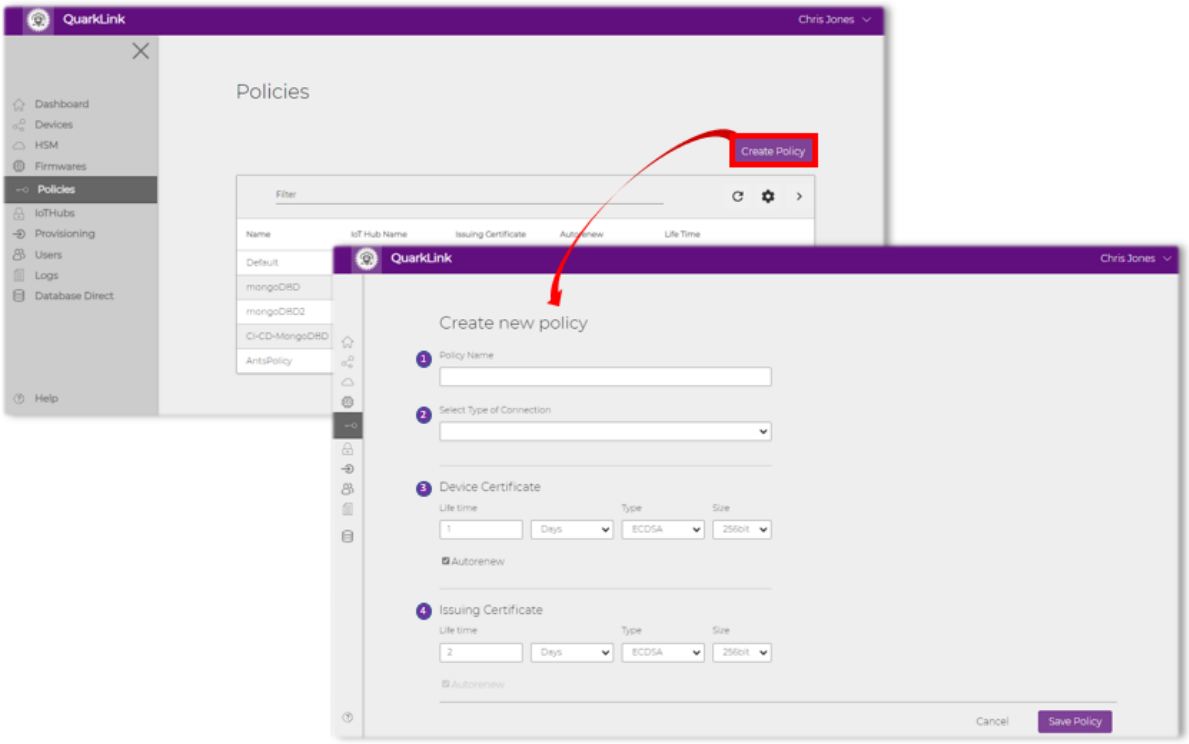
Creating a Policy – To create a new policy for devices click on the "***Create Policy***" button. In the dialog box that appears (see below) enter the following information:

1. **Policy Name** - Enter a unique name for the policy.
2. **Select Type of Connection**– Using the drop down menu, select the type of Connection required, options are :

    a. ***IoT Hub*** – This represents the connection to a supported cloud service provider (see section 12).

    b. ***Database Direct*** – This represents a connection to a users database (see section 16 for further configuration details).

If the ***Type of Connection*** is an ***IoT Hub***, the following additional data is required to be input :



c. **Device certificate life time** – Configure the device certificate by selecting :

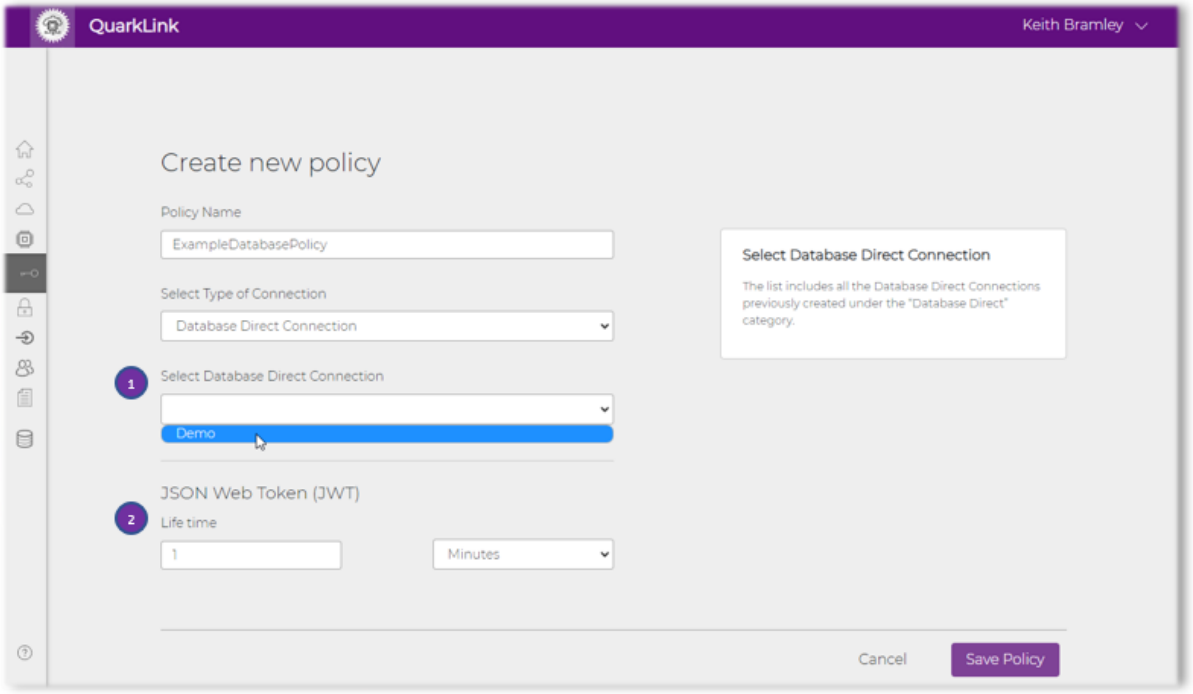
1. The number of minutes/hours/days/months/years for the validity period.
2. The signature type [ECDSA (default)]
3. Signature size [256 bits (default)]
4. Autorenew (select this option if you wish the QuarkLink to automatically re-issue a new certificate when the current validity period expires).

d. **Issuing certificate life time** – Configure the issuing certificate by selecting :

1. The number of minutes/hours/days/months/years for the validity period.
2. The signature type [ECDSA (default)]
3. Signature size [256 bits (default)]
4. Autorenew (select this option if you wish the QuarkLink to automatically re-issue a new certificate when the current validity period expires).

See section 9 for further information regards certificate usage.

If the ***Type of Connection*** is a ***Database Direct Connection***, the following additional data is required to be input :



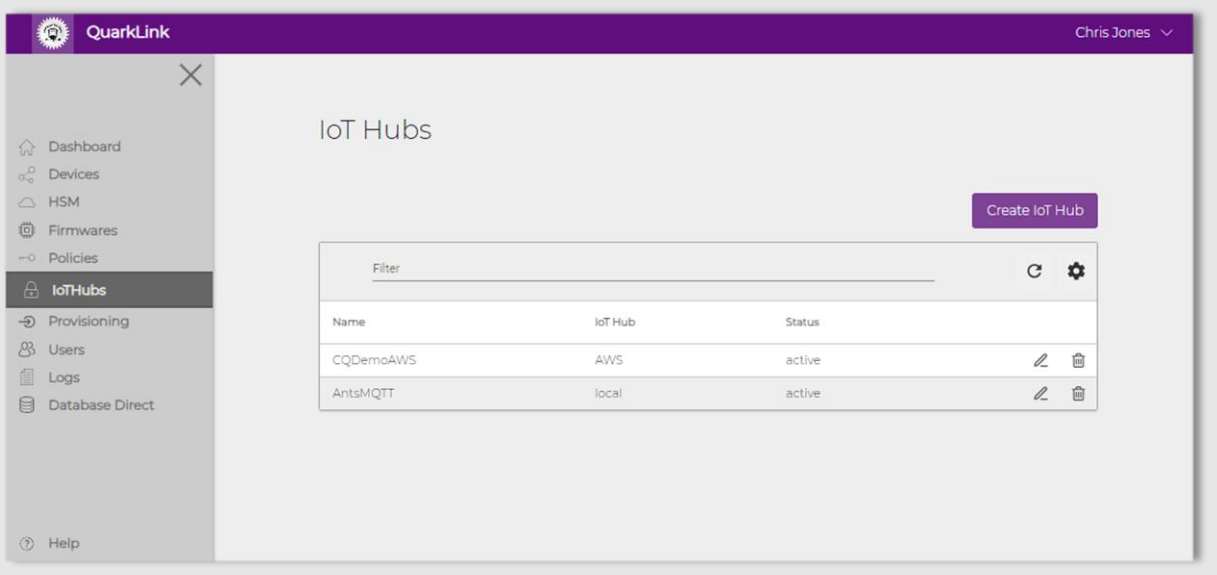
1. **Select Database Direct Connection** – From the drop down menu select the required database that the user wishes to connect to.
2. **JSON Web Token (JWT)** – Configure the life time of the JWT. Shorter lifetime tokens are safer, but require renewing more often. Select a JWT lifetime suitable for your application.

Click the "***Save Policy***" button to save the new ***Policy***.

Note : The ***Database Direct Connection*** used in the policy must be setup prior to creating the policy associated with it. Please see section 16 for more details on the configuration of a Database Direct connection.

12 IoTHubs

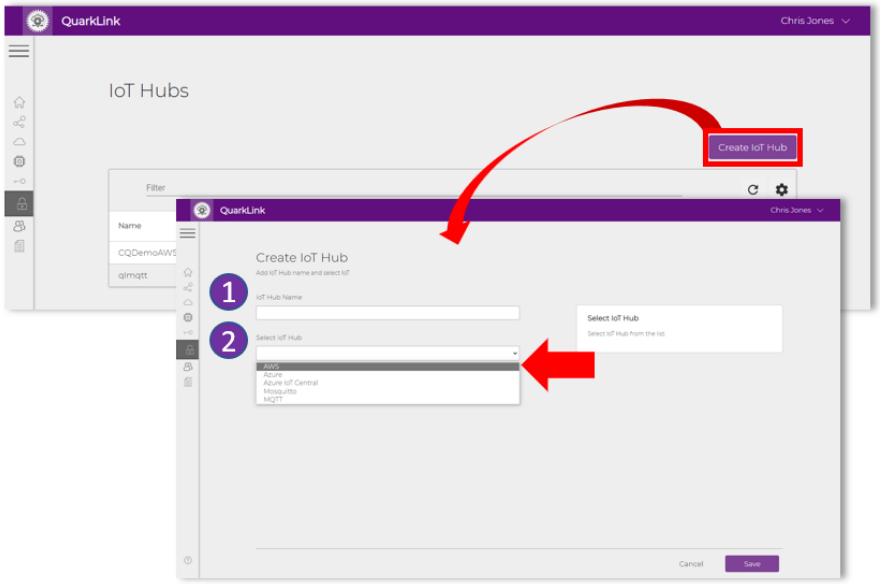
This menu option shows the IoT Hubs currently accessible by the QuarkLink instance. QuarkLink users with "***Administrator***" privileges can create new IoT Hub configurations. Users with "***Contributor***" privileges can only view the available IoT Hubs.



An IoTHub is a representation of the users cloud service providers account information that the QuarkLink must use in order to transfer certificates and register the enrolled devices with the service. The sections below provide details of the IoT Hub configurations for the cloud service providers supported by the QuarkLink.

Create IoT Hub

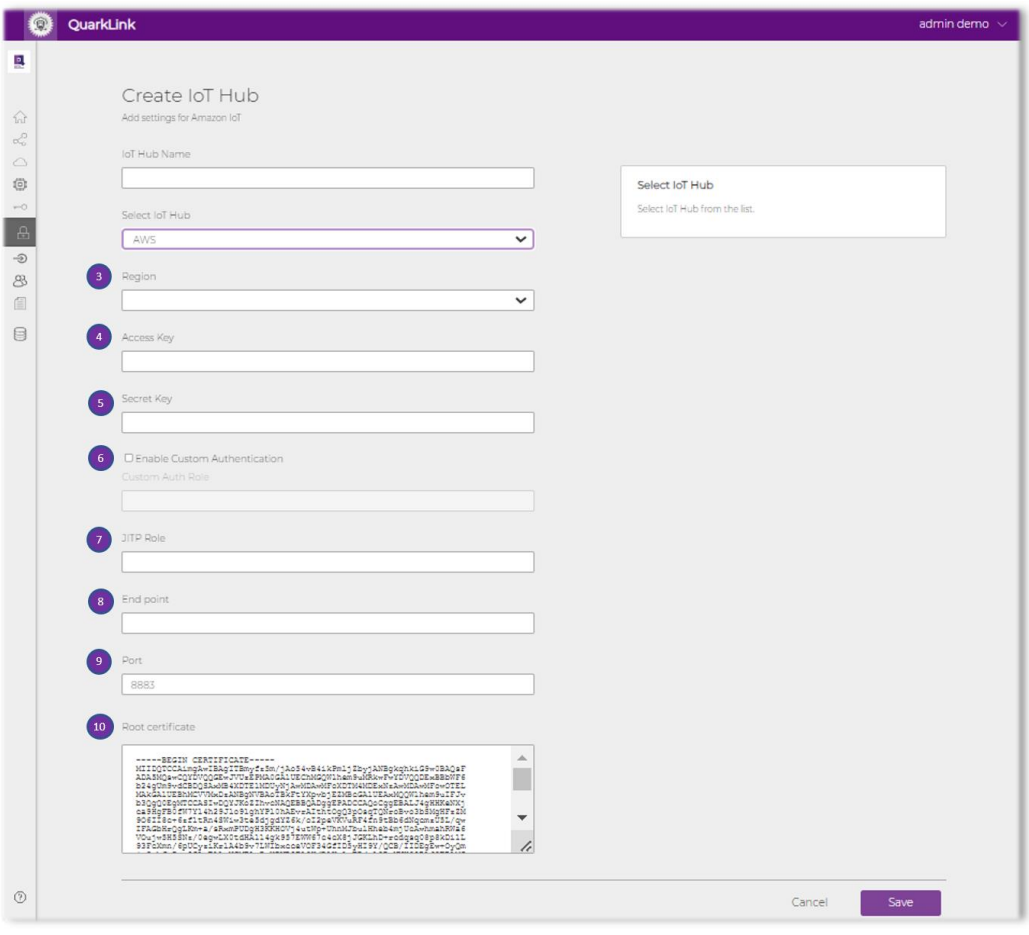
Creating a IoT Hub – To create a new ***IoTHub*** for use in policies click on the "***Create IoT Hub***" button. In the dialog box that appears (see below) enter the following information :



1. **IoT Hub Name** - Enter a unique name for the ***IoT Hub***.
2. **IoT Hub Type** – Using the drop down menu, select the type of connection which is either a cloud service provider or a MQTT broker (see below for details for each option).

12.1 Amazon AWS

When the Amazon AWS IoTHub is selected for configuration the screen below will be displayed. The user is required to enter their AWS account information. Details of each required entry are described below:



Please refer to Amazon AWS documentation for further details of the required IoT Hub entries detailed below. Most of the information below will be provided by the users AWS account:

3. **Region** – AWS Regions are separate geographic areas that AWS uses to house its infrastructure. The closer the region is to the user, the better, please refer to Amazon AWS documentation for further details.
4. **Access Key** – Access keys are long-term credentials for an IAM user or the AWS account root user. This entry will be obscured once entered.
5. **Secret Key** – Secret access keys are, as the name implies, secrets, like your password. This entry will be obscured once entered.

6. **Custom Auth Role** – Enable QuarkLink proprietary authentication protocol.
7. **JITP Role** – Amazon Resource Names (ARNs) uniquely identify AWS resources.
8. **End point** – To connect programmatically to an AWS service, you use an endpoint. An *endpoint* is the URL of the entry point for an AWS web service.
9. **Port** – No entry required here, default port included.
10. **Root certificate** – This is populated by QuarkLink automatically and is the Root certificate for Amazon AWS (see below). This certificate is used by the QuarkLink during secure communications with Amazon AWS.

The Amazon AWS root certificate is publicly (https://good.sca1a.amazontrust.com/) available from the Amazon AWS website. The current certificate is shown below:

Amazon Trust Services Demo Page

Expected Status: good

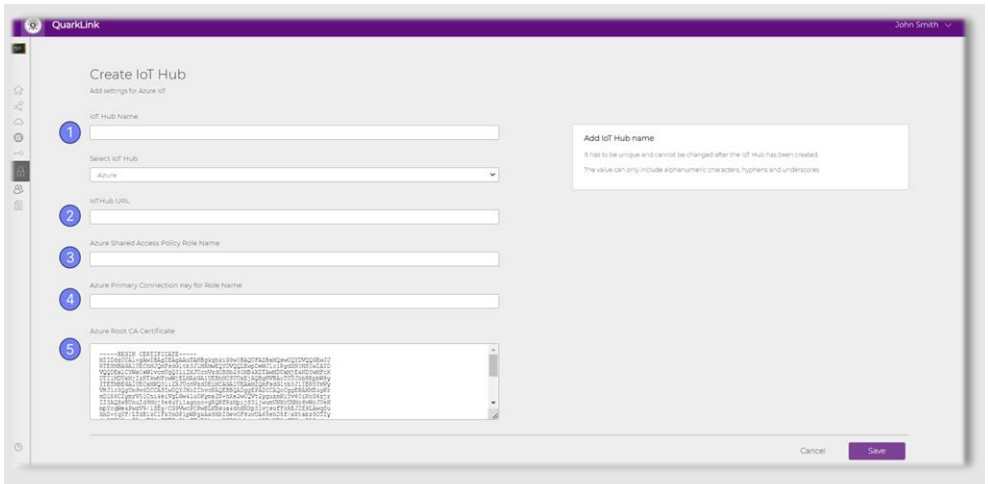
Root: CN=Amazon Root CA 1,O=Amazon,C=US

-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM
9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA
A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----

Once all the required information has been provided, click "***Save***".

12.2 Microsoft Azure

When the Microsoft Azure IoTHub is selected for configuration the screen below will be displayed. The user is required to enter their Azure account information. Details of each required entry are described below:



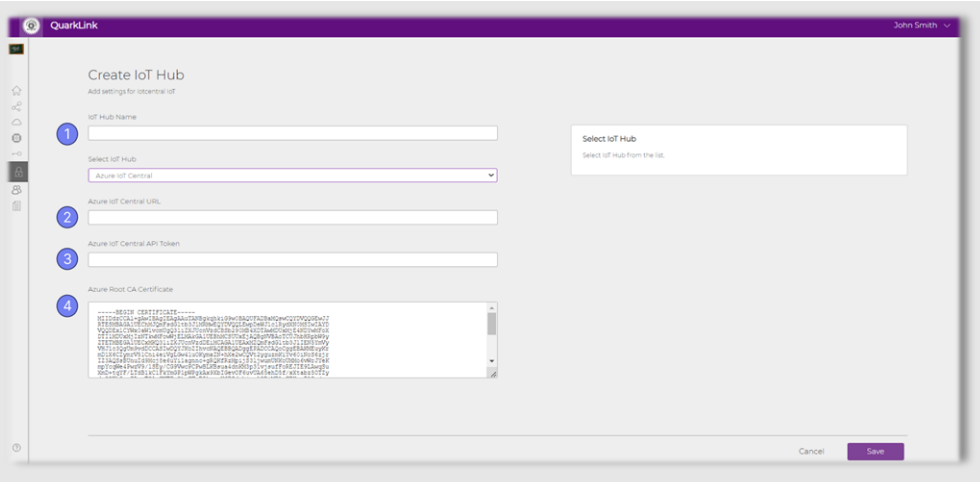
Please refer to Microsoft Azure documentation for further details of the required IoT Hub entries detailed below. Most of the information below will be provided by the users Azure account:

1. **IoT Hub Name** – This is a free text entry and allows the user to create a name that identifies the relevant Azure account with QuarkLink to be used for a ***Policy***.
2. **IoT Hub URL** – The URL to be used for the Azure Hub.
3. **Shared Access Policy Role Name** – The name of the role that lets the user group permissions and grants them to applications using access keys and signed security tokens.
4. **Primary Connection Key** – This is the key given to Azure account users that enables connection to Azure IoT Hub.
5. **Root CA Certificate** – This is populated by QuarkLink automatically and is the Root certificate for Azure IoT Hub. This certificate is used by the QuarkLink during secure communications with Azure IoT hub.

Once all the required information has been provided, click "***Save***".

12.3 Microsoft Azure IoT Central

When the Microsoft Azure IoT Central is selected for configuration the screen below will be displayed. The user is required to enter their Azure account information. Details of each required entry are described below:



Please refer to Microsoft Azure documentation for further details of the required IoT Central Hub entries detailed below. Most of the information below will be provided by the users Azure account:

1. **IoT Hub Name** – This is a free text entry and allows the user to create a name that identifies the relevant Azure account with QuarkLink to be used for a ***Policy***.
2. **Azure IoT Central URL** – When opening an Azure IoT Central account, Azure IoT Central generates a unique **URL** prefix for the account user, based on the

application name. This URL allows access to the user application. This URL must be unique.

3. **Azure IoT Central API Token** - API tokens are meant for service to service communication, without a signed-in user context.
4. **Azure Root CA Certificate** – This is populated by QuarkLink automatically and is the Root certificate for Azure IoT Hub. This certificate is used by the QuarkLink during secure communications with Azure IoT Hub.

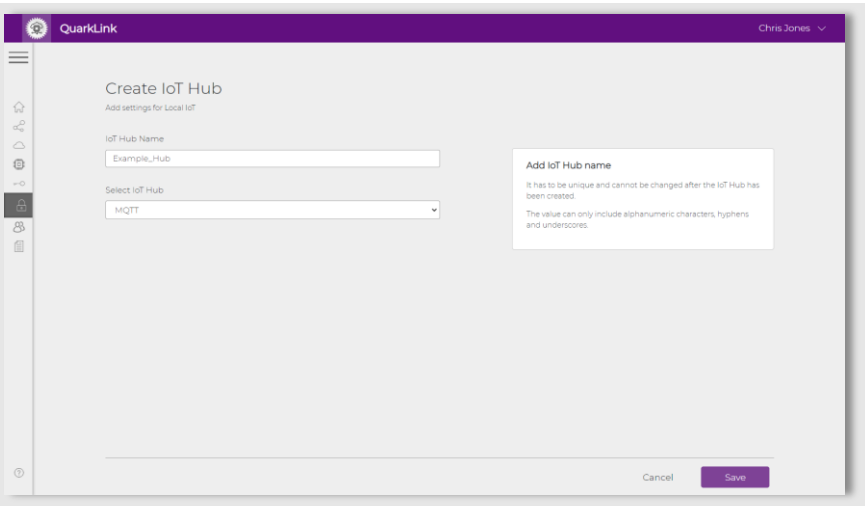
Once all the required information has been provided, click "***Save***".

12.4 MQTT

MQTT is an acronym for the inaccurately named Message Queuing Telemetry Transport which is a machine to machine network protocol. It is designed for communications with remote locations that have devices with resource constraints or limited network bandwidth. The protocol typically runs over TCPIP and can be configured to communicate using TLS encryption.
For customers who do not wish to connect their devices to a cloud service provider, QuarkLink includes an MQTT Broker capability. This broker can act as a post office where connected devices can both subscribe or publish to a "Topic". Multiple clients can subscribe to a topic from a single broker (one to many capability), and a single client can register subscriptions to topics with multiple brokers (many to one). The main advantages of an MQTT broker are:

- It can be configured to eliminate vulnerable and insecure client connections.
- Can easily scale from a single device to thousands
- It can be configured to manage and track all client connection states, including security credentials and certificates.
- It can be configured to reduce network strain without compromising the security (cellular or satellite network).

To enable the QuarkLink MQTT Broker feature, an IoTHub instance must be created. When an MQTT IoTHub is selected there is only a requirement for the IoT Hub name to be entered. An example of the MQTT option dialog is shown below:



Once the MQTT IoT Hub name has been provided, click the "Save" button.

12.4.1 MQTT Broker Communication (Local)

If the MQTT Broker feature has been enabled on the QuarkLink, a device that is associated with a security policy, that includes an IoT Hub of the type MQTT, will be enrolled onto the QuarkLink local MQTT port :

https://<quarklink_instance_name>.quarklink.io:8883

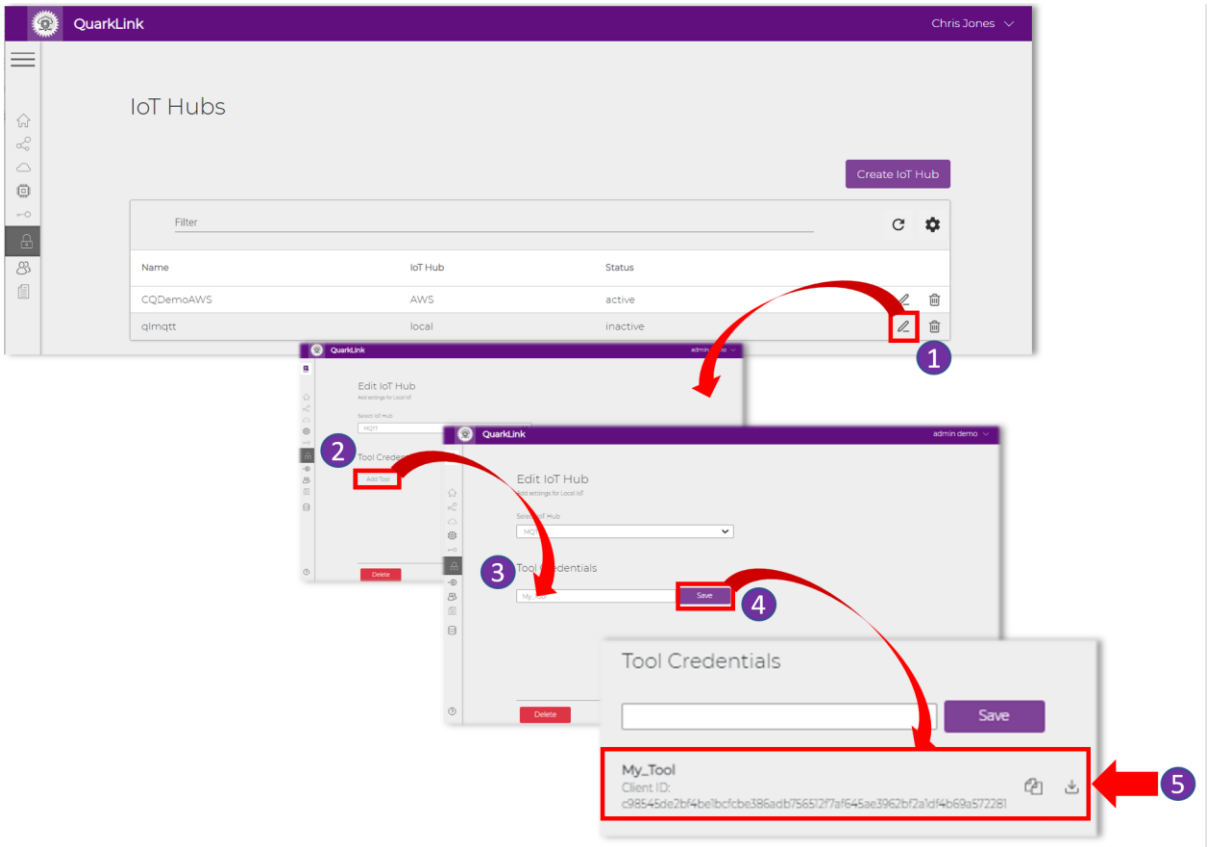
The connected device can publish and subscribe to topics via this URL and port. Topics will be handled by the broker as appropriate supporting one to many and many to one capabilities.

In the case where multiple brokers are required, a separate MQTT Broker instance is created for each defined MQTT IoTHub, each with its own topic space which is not shared by other local MQTT Broker instances.

12.4.2 MQTT Broker Communication (Third Party Tool)

The MQTT Broker feature has been designed to allow users to use third party tools to communicate with the QuarkLink MQTT Broker. Tools such as Node-RED or MQTT Explorer are ideal for communicating with an MQTT connected device securely. However, the correct security credentials are needed by the third party tool to enable the TLS encrypted link. QuarkLink has the capability to create the required credentials for use with the third party tool.

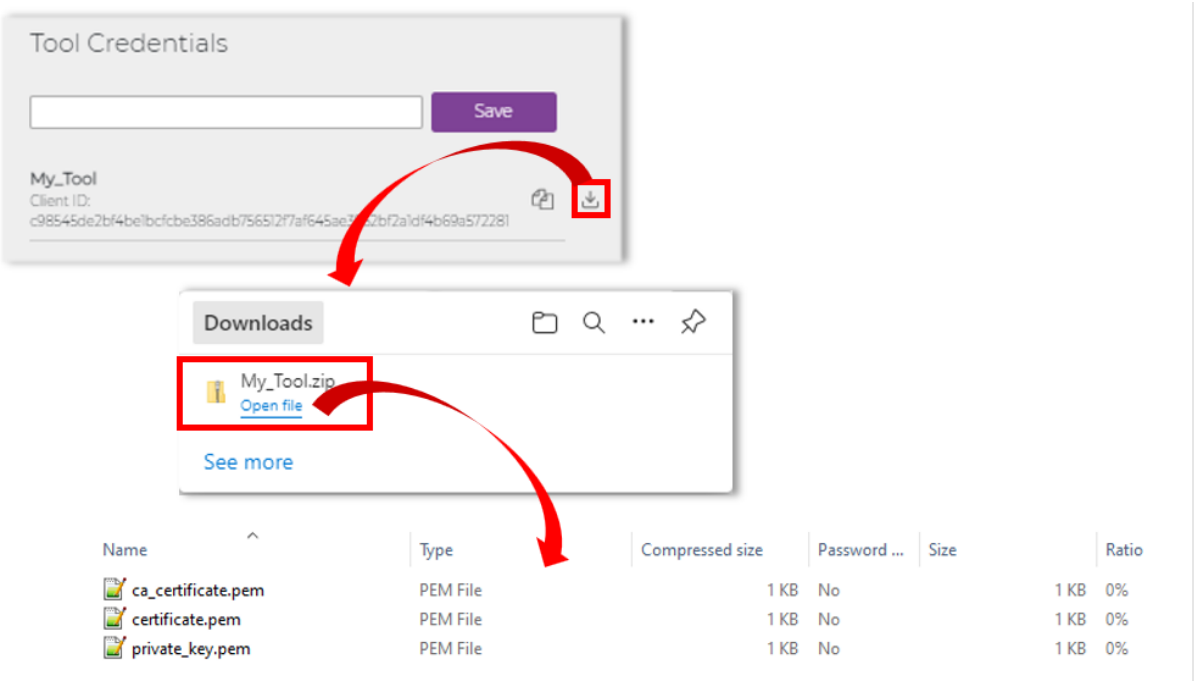
Click on the IoT Hub main menu in the QuarkLink GUI. Create the security credentials for your third party tool by following the sequence shown below :



1. Click in the edit icon for the select IoT Hub of type MQTT. The dialog box shown above will open.
2. Click on the ***Add Tool*** button. The dialog box shown above will open.
3. Type in a free text name for your third party tool instance.
4. Click the ***Save*** button.
5. The security credentials will be generated by the QuarkLink. Icons are available to copy or download the credentials.

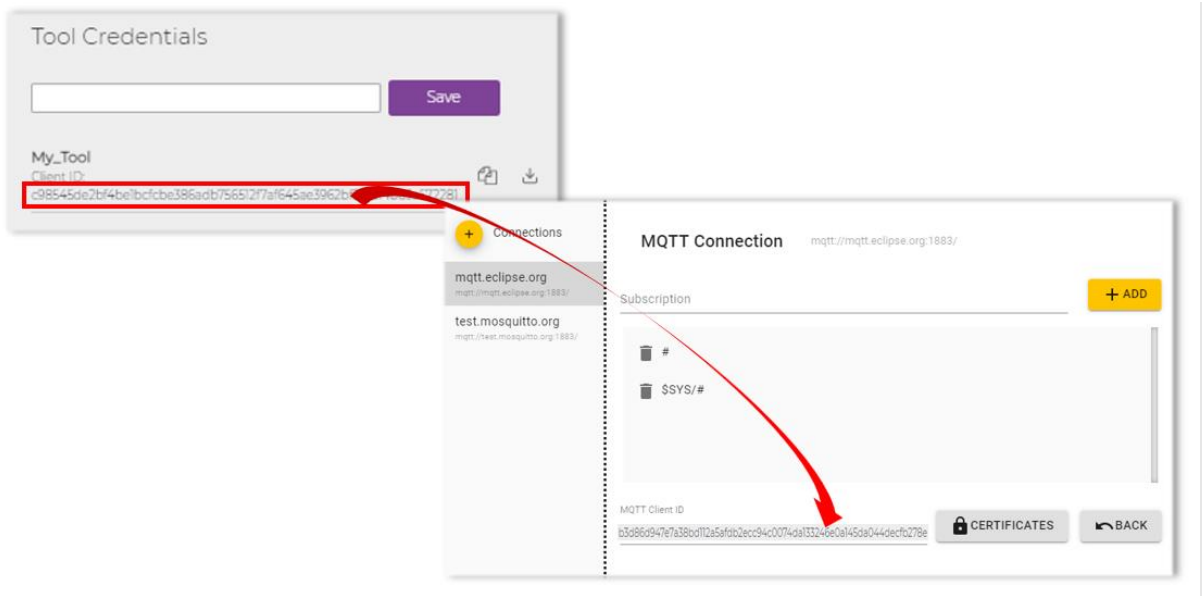
The security credentials for multiple tools can be generated by simply adding additional names into the MQTT Tools free text entry pane and clicking on the ***Save*** button.

To download the credentials, click the  icon. A <tool_name>.zip file will be downloaded to the users PC/Laptop as seen in the diagram below:



The Zip file contains the keys and certificates needed for a third party tool to connect to the QuarkLink MQTT Broker.

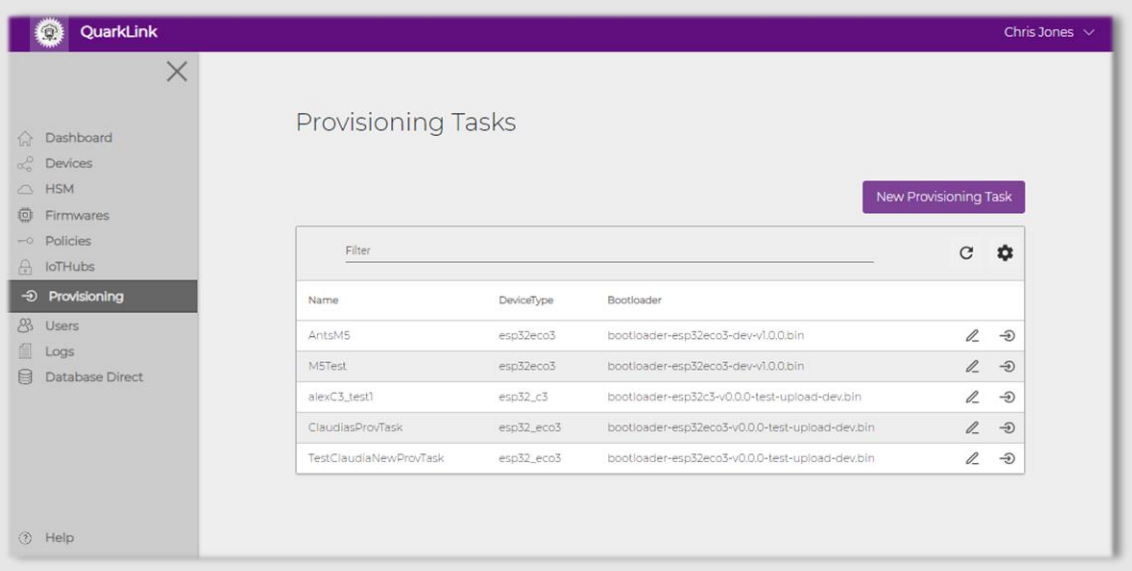
Note : Please ensure that this Zip file is stored securely and is treated as confidential as it contains private cryptographic keys.



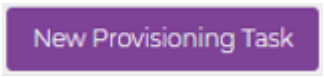
Note : Third party tools that are to be used with the security credentials generated by QuarkLink must also be configured to set the “ClientID” to that as provided by QuarkLink. The diagram above shows an example MQTT application (Tool) that utilises the ClientID generated by QuarkLink.

13 Provisioning

Provisioning tasks will allow the user to provision (program) their IoT device with a bootloader and Initial Enrolment Firmware (IEF) selected. Provisioning is the process of programming the target IoT device with cryptographic keys (IEF) to ensure that firmware updates can be carried out securely using a secure boot process. Once the IEF has been programmed, the users signed firmware can then be programmed as the main application of the IoT device.



To create a new provisioning task, click on the ***Provisioning*** option in the QuarkLink left hand main menu.



Creating a Provisioning Task – To create a new ***Provisioning Task*** to allow the provisioning of a new bare metal IoT device, click on the "***New Provisioning Task***" button. In the dialog box that appears (see below) enter the following information :

The relevant fields can be seen in the diagram below:

1. **Provisioning Task Name –** This will set a unique name for the provisioning task
2. **Device Type –** Displays list of devices that are supported.
3. **Bootloader –** There are two types of bootloader options preloaded into QuarkLink:

   a**. veFuse** – This bootloader type (Virtual eFuse) is used for development and prototyping as the cryptographic security keys are stored in the target hardware volatile memory (SRAM). This ensures that the target device is recoverable once the bootloader has been flashed. It is not recommended that this bootloader type is used for production IoT devices.

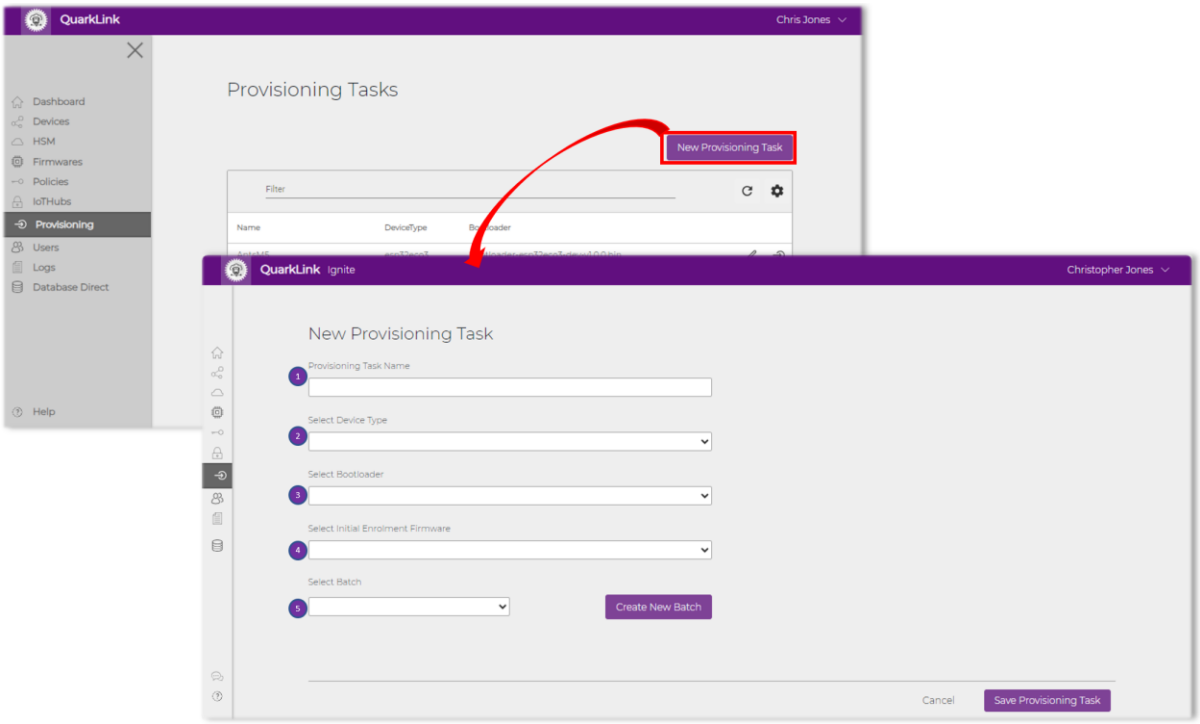
   b. **Release** – Release bootloaders have their cryptographic security keys programmed into OTP (One Time Programmable) memory. The device cannot be reprogrammed with new keys or with a different bootloader once the release version has been provisioned. This bootloader type should be used in production only.

4. **Initial Enrolment Firmware –** The IEF is a firmware image that is used to configure the target hardware. The IEF is run once the bootloader has verified that it has been signed correctly. A more detailed description of the function of the IEF can be found in the *CQ Initial Enrolment Firmware* documentation available from the Crypto Quantique GitHub. There are primarily two types of IEF options:

   a**. veFuse** – Virtual eFuse IEFs include addition debug output information for use during development and should not be used for production.
   b. **Release** – Release IEFs do not have the additional debug output information and are recommended for production release only.

   Additional *veFuse* and *Release* options may be available dependant on the cryptographic hardware utilised to protect the private keys used for encryption and signature verification.

5. **Select Batch –** During provisioning, the QuarkLink will read the DeviceID from the target hardware device and copy it into a Batch. This simplifies configuration of the QuarkLink.



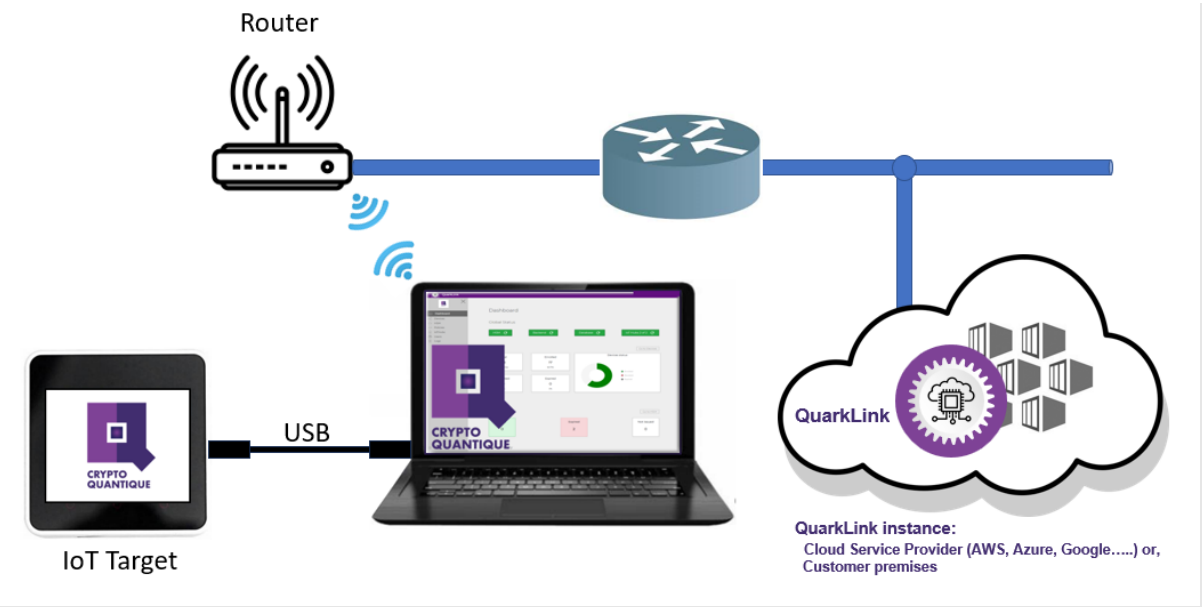
Once all the ***New Provisioning Task*** information has been configured correctly, click "***Save Provisioning Task"*** button.

13.1 Running a Provisioning Task

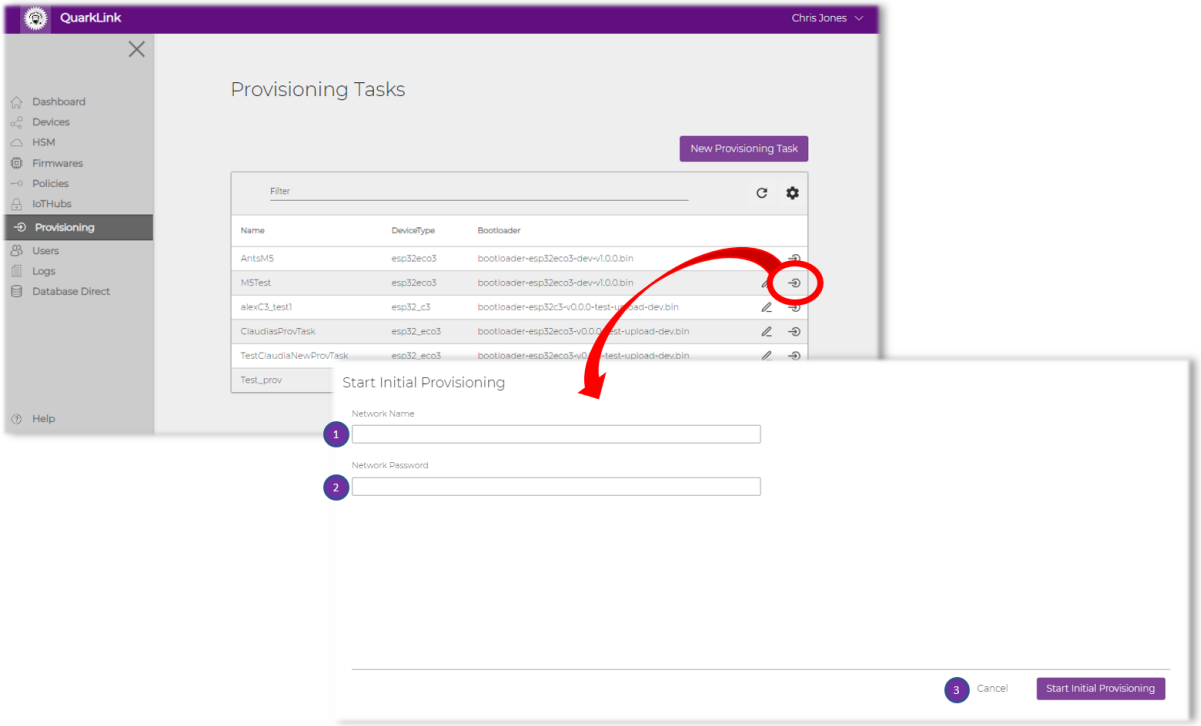
Prior to using the QuarkLink to onboard an IoT device to either a cloud service provider or a database, the target device must be programmed with the correct firmware. QuarkLink includes a feature to allow the provisioning of a target device such that the correct cryptographic keys are securely programmed into a target device along with the correct bootloader that is secure, uses the correct cryptographic keys and ensures secure firmware Over-The-Air updates (FOTA).

To provision a supported IoT device, follow the procedure detailed below:

1. Connect the target device to the PC/Laptop (via a USB cable) that is being used to run the QuarkLink security platform (see below):



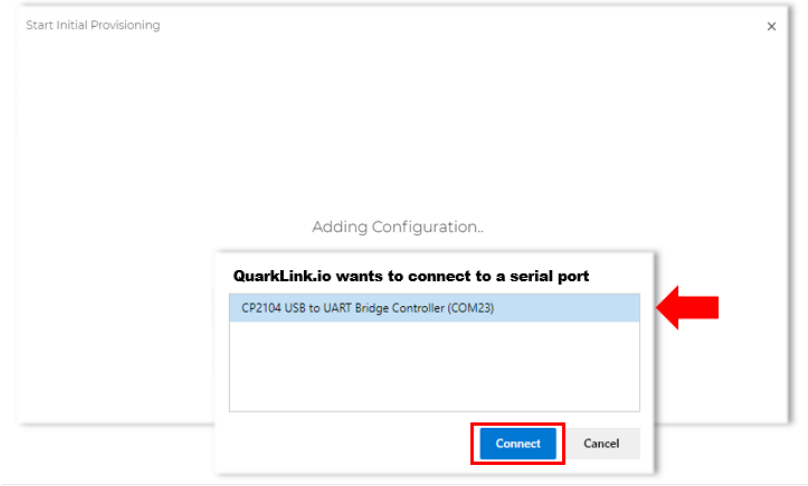
2. Select the ***Provisioning*** menu in the left hand main menu of the QuarkLink and click on the provision icon to start the initial provisioning task (see below) :



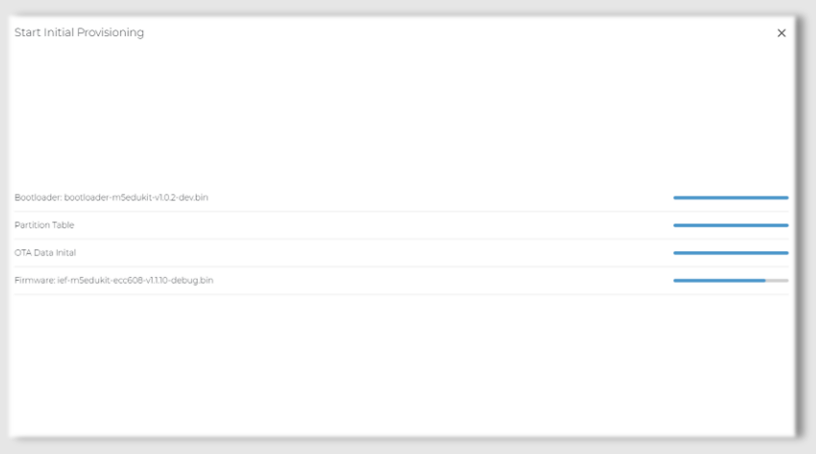
3. Input the required information to start the provisioning task :

1. **Network Name –** This is the name of the WiFi network (SSID) that the user will use to connect the IoT device to the QuarkLink. The WiFi network credentials will be flashed (provisioned) into the target device during provisioning.
2. **Network Password –** This is the password of the WiFi network that the IoT device will use to connect to the QuarkLink.

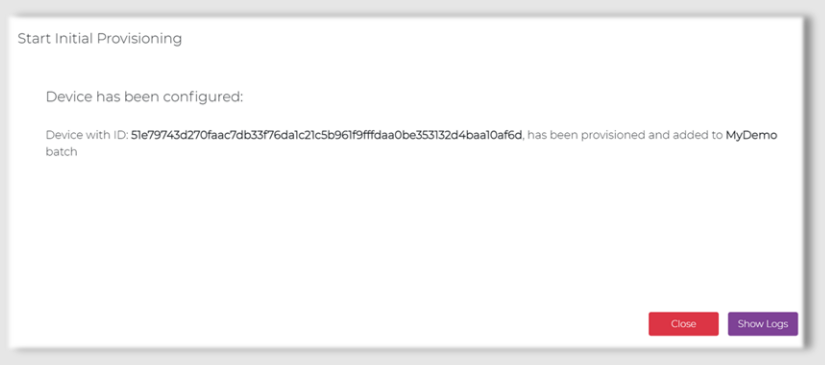
3. Once the information has been input, click the ***Start Initial Provisioning*** button. The QuarkLink will now connect to the IoT device via the USB. The QuarkLink will open a dialog box that lists all the currently connected USB devices. Select the correct COM port for the IoT device that you wish to provision and click on the ***Connect*** button (see below):



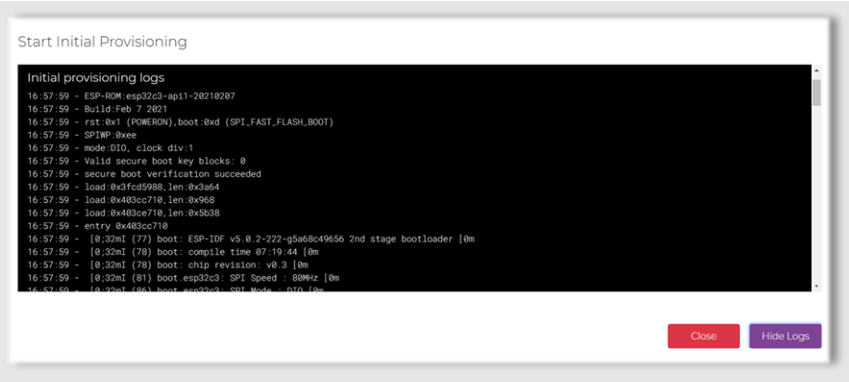
5. The QuarkLink will now provision the IoT devices. The progress of the function will be shown in the dialogue box similar to that shown below :



6. Once programmed, the QuarkLink will display the dialogue box shown below. The device identity ***DeviceID*** with be displayed. It is this device identity that will be programmed into the QuarkLink Batch as configured in step 3 (Note: In the example below the ***DeviceID*** has been loaded into the ***MyDemo*** Batch). It is recommended that the device identity number shown in the dialogue box is copied to a file on the users PC/Laptop for late referral. The ***DeviceID*** does not include any secret information and is not confidential.

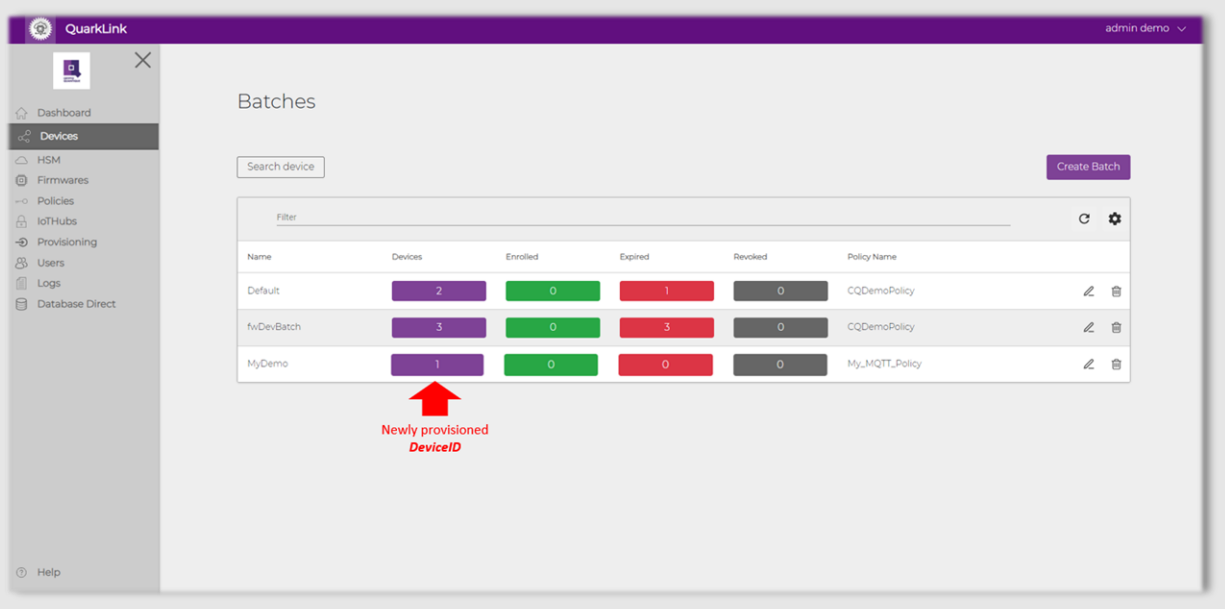


7. To view the process of provisioning the IoT device, the user can click on the ***Show Logs*** button (see below) where a list of the provisioning commands are displayed.



8. To complete the provisioning process, click the ***Close*** button.

9. The QuarkLink ***Dashboard*** will now show the new ***DeviceID*** added to the ***Batch*** (example shown below).



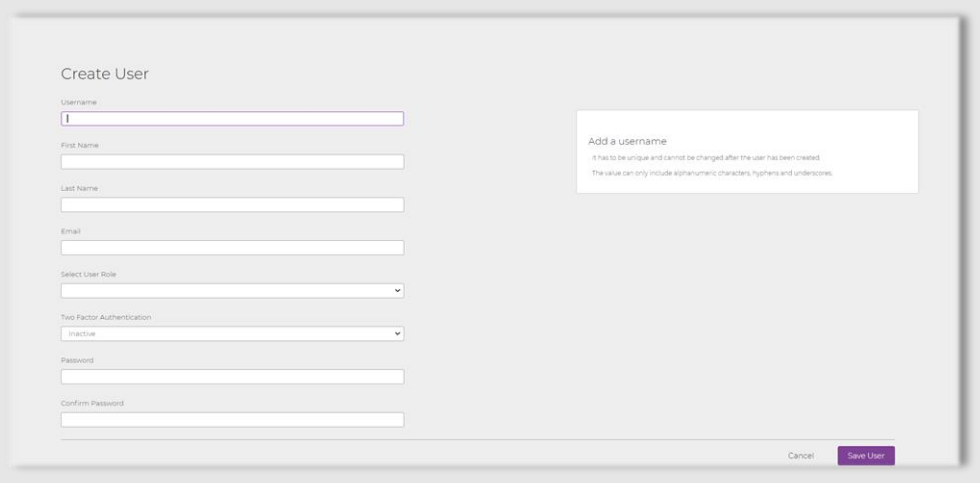
14 Users

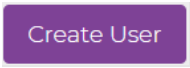
This menu option is used to create users of the QuarkLink instance. There are two types of users available for configuration, they are "***Administrator***" and "***Contributor***". The privileges associated with each user type are described below:

Administrator – Access all areas and functions

Contributor – Restricted access :

Unable to access Users page i.e. cannot create/edit users
Unable to access Firmware page (unable to sign firmware)
Unable to access Settings (no access to IoT Hub information)

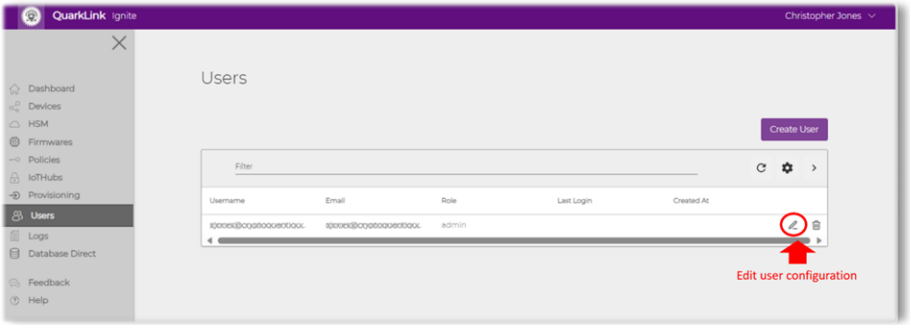




Creating a User – To create a new ***User*** click on the "***Create User***" button. In the dialog box that appears (see above) the user is required to add the information for the new user.

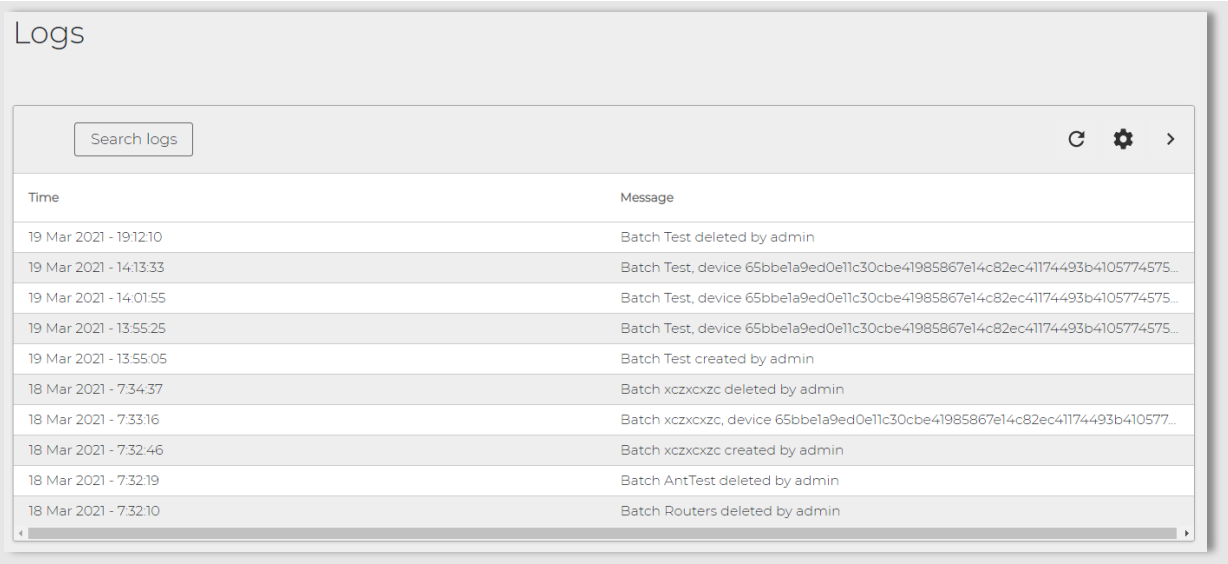
Once all the required information has been provided, click "***Save User***".

Note : Two factor authentication is enabled by default when registering for a new QuarkLink. Please review the Crypto Quantique YouTube website for demonstration videos on how to use the two factor authentication (https://www.youtube.com/@CryptoQuantique).
Two factor authentication can be disabled for individual users (not recommended) by clicking the ***Edit*** icon.



15 Logs

This menu option provides access to the QuarkLink logs. Only the last 10 entry logs are displayed. An example of the logging information is shown below:



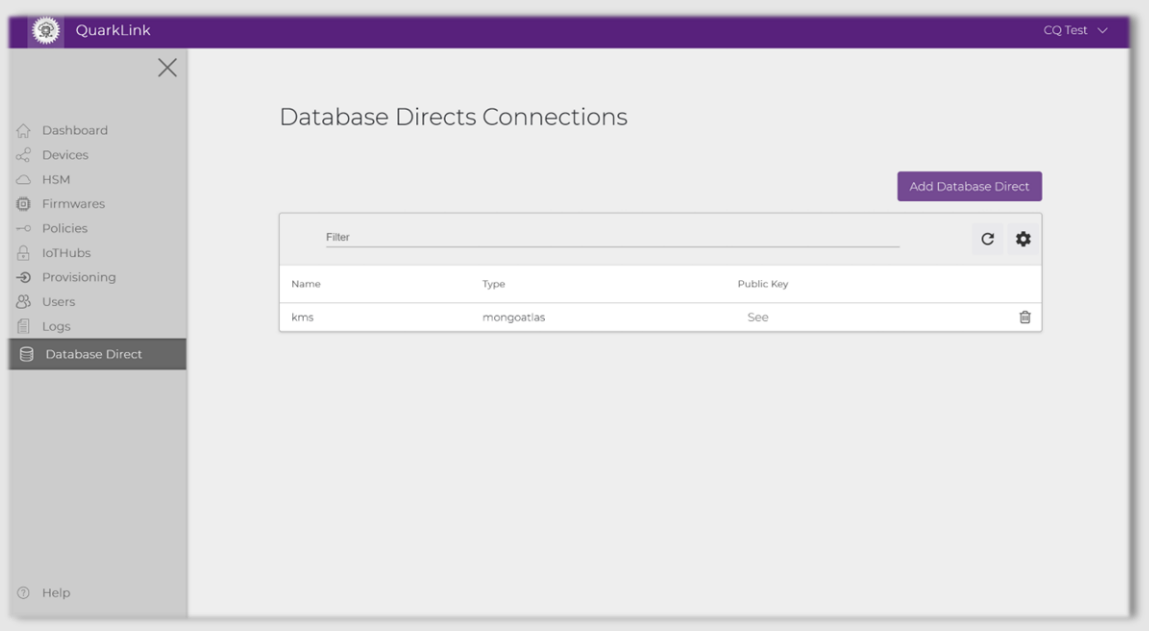
The ***Refresh*** button updates the last activity screen.

The ***Settings*** button allows the user to manage the columns that are required to be displayed in the ***Last Activity*** area (see below).

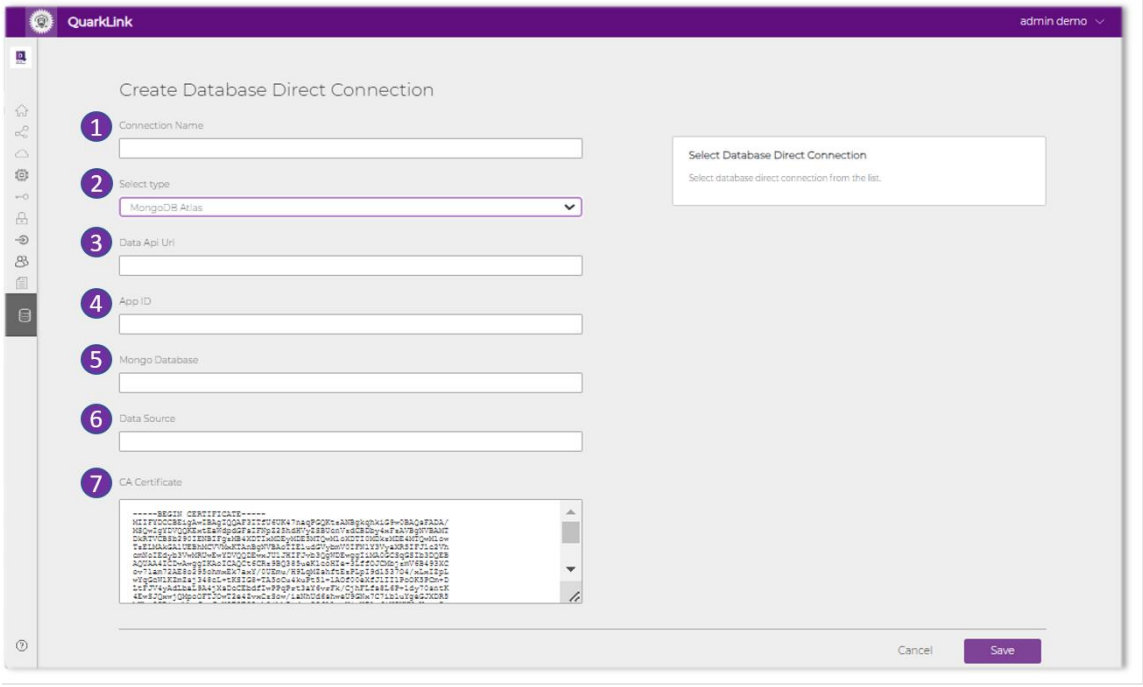
Screen extend – display all message text

16 Database Direct

This menu option shows the database direct connections which have been configured for the instance. Database direct refers to QuarkLink managing connecting an IoT device directly to a database instead of the device connecting to an IoT Hub such as AWS or Azure.



You can create a database direct connection by clicking on "***Add Database Direct***".



The relevant fields descriptions can be seen below:

1. **Connection Name** – The name of the database direct connection. This allows the user to give the connection a unique name to link it to a policy
2. **Select type** – A dropdown to allow you to select the type of database you wish to connect to. For the moment the only option is MongoDB Atlas
3. **App ID** – The ID of the MongoDB Atlas data api connection. This field is automatically completed from the Data API URL field
4. **Data Api Url** – The URL provided by mongo atlas when you create a Data API application – in the format of https://eu-west-2.aws.data.mongodb-api.com/app/data-vwxyz/endpoint/data/v1
5. **Database** – The mongo database the user wishes to write their data to - free text
6. **Data Source** – The name of the mongo instance which data will be written to – this is the mongo cluster name when the mongo database was created.
7. **CA Certificate** – This field is automatically populated by the QuarkLink GUI. The certificate is used by QuarkLink to authenticate the Mongo Atlas database. The CA certificate shown is available from the MongoDB website. For further information regarding this certificate please visit the MongoDB FAQ website (see below).

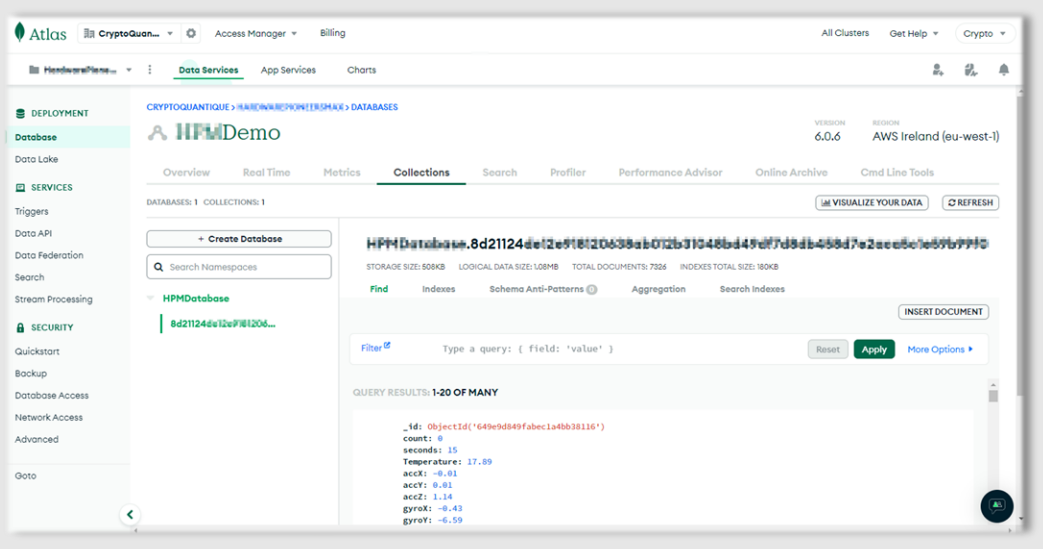
https://www.mongodb.com/docs/atlas/reference/faq/security/

Hard-coded Certificate Authority

If you hard-coded the IdenTrust's root Certificate Authority (DST Root CA X3) or an intermediate certificate as the only trusted Certificate Authority for your application's connection to Atlas, ensure that you add Let's Encrypt's root Certificate Authority (ISRG Root X1) to your certificate store.

Once the information has been added, press the "***Save***" button.

An example Mongo Atlas database screen is shown below from which the information required for the ***Add Database Direct*** dialogue box can be extracted (see above).

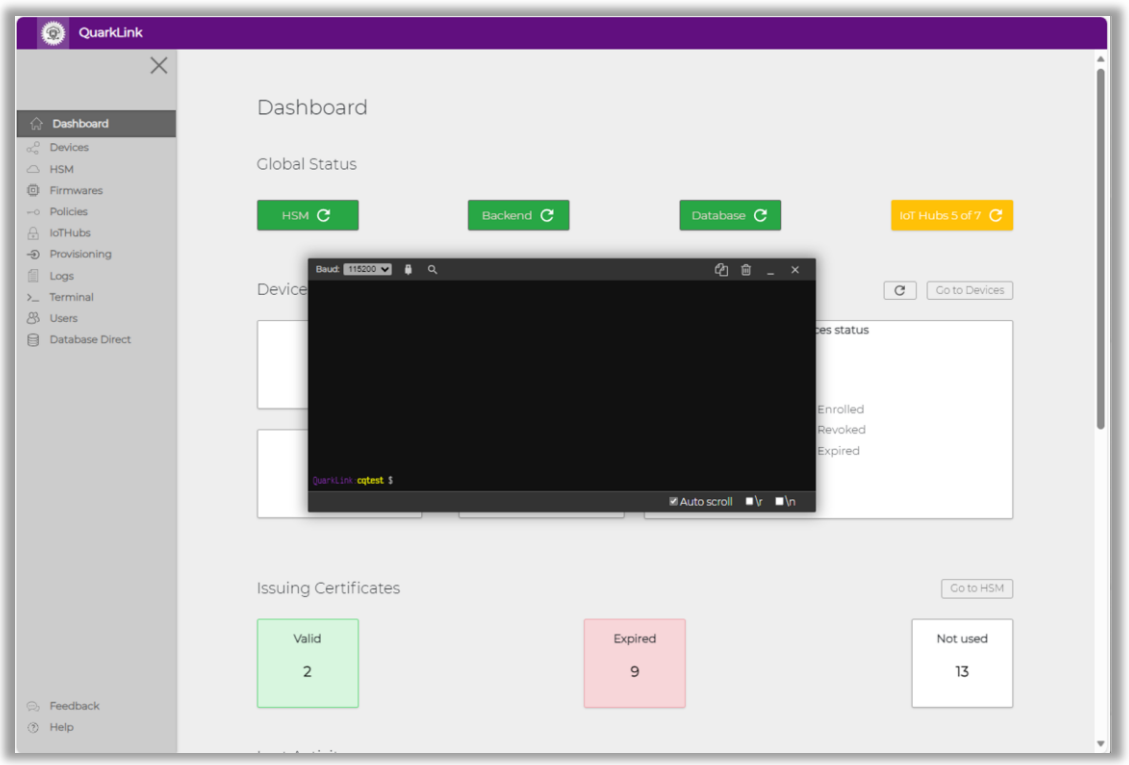


Further details on how to configure Mongo atlas for use with QuarkLink Database Direct can be found in app note ***Mongo Atlas QuarkLink Setup Procedure***.

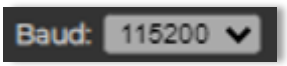
17 Terminal

QuarkLink includes an inbuilt Terminal application that allows you to access the serial port of a connected device within the browser window.

The Terminal window is limited to movement only within the browser window.



The Terminal includes the following controls :



COM Port baud rate. A drop-down menu is available with a selection of baud rates. The baud rate used is typically the default of 115200.

Connect. Click on this icon to open the COM port list and select the required IoT device. The icon will turn green when connected. When connected clicking the icon again will disconnect the COM port. The icon will turn red.

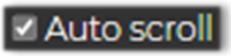
Search the current terminal buffer for the entered text

Copy the terminal buffer to the clipboard

Clear the terminal buffer

Minimise the terminal window within the browser window

Close the terminal application



Click the radio button to automatically scroll the terminal buffer

Click the radio button to add a <carriage return> character

Click the radio button to add a <line feed> character

18 Revision History

CQ QuarkLink User Guide

Rev.	Date	Owner	Description
1.00	1.9.2021	CDJ	Original document
1.10	22.9.2021	CDJ	Added Authenticated Certs and Tokens Updated resources section Corrected errors
1.20	31.10.2022	CDJ	Updated to reflect new Azure IoT Central support
1.30	12.7.2023	CDJ/KB	Updated to reflect Ignite additions
1.31	18.9.2023	CDJ	Updated to reflect GUI changes for firmware signing key
1.40	21.9.2023	KB	Updated to reflect new QL Terminal and Application usage
1.41	13.10.2023	CDJ	Updated Provisioning to reflect DS peripheral update, signing key import

CRYPTO QUANTIQUE

United Kingdom

Unit 304-5,
164-180 Union Street,
London
SE1 0LH

General contact email:
info@cryptoquantique.com