**UWC Gateway to Cloud Communication**

March 2021

[**• Architecture** 2](#_Toc67318309)

[**• Installation and Configuration** 3](#_Toc67318310)

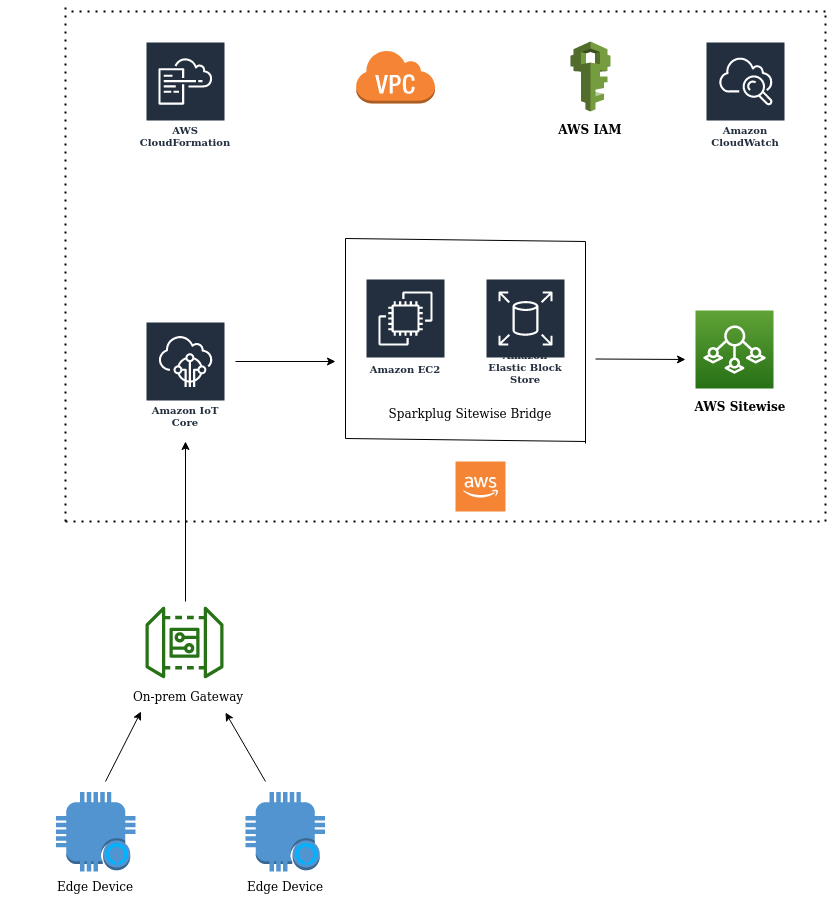
[1. SSB installation and cloud infrastructure provisioning 3](#_Toc67318311)

[2. AWS IoT core broker and SSB configuration 4](#_Toc67318312)

[3. UWC gateway configuration 4](#_Toc67318313)

[**• Monitor Data on Cloud** 5](#_Toc67318314)

**• Architecture**



Edge devices represent PLC devices on the site. They connect to the on-prem Gateway which is also termed as UWC gateway. The gateway securely publishes the sparkplug format data to the cloud using the MQTT communication protocol. The data first reaches AWS IoT core which is an AWS service that provisions cloud connectivity for IoT edge devices. AWS IoT core possess an MQTT broker as one of it's component. The UWC gateway will be communicating with this broker to publish the data. There will be an MQTT client residing in the gateway which will publish the data to the AWS IoT core broker. Post this communication, we have the edge data in the cloud. Find more information about AWS IoT core here https://aws.amazon.com/iot-core/

Sparkplug Sitewise Bridge(SSB) is a service which rapidly connects OT data from Industrial Operations(on-prem data) to AWS IoT SiteWise with minimal configuration and zero coding . For more information on SSB, please hit the link <https://aws.amazon.com/marketplace/pp/Cirrus-Link-Sparkplug-SiteWise-Bridge/B08L8KNCNN>

SSB software will be running in an EC2 instance running in the cloud and it is comprised of an MQTT client which will subscribe to the AWS IoT core broker in order to fetch the on-prem data. Once it gets the required data, it will create and update resources(assets, models, etc) in AWS Sitewise. We will be able to monitor the on-prem data on AWS Sitewise. For more information on Sitewise, please access the link <https://aws.amazon.com/iot-sitewise/>

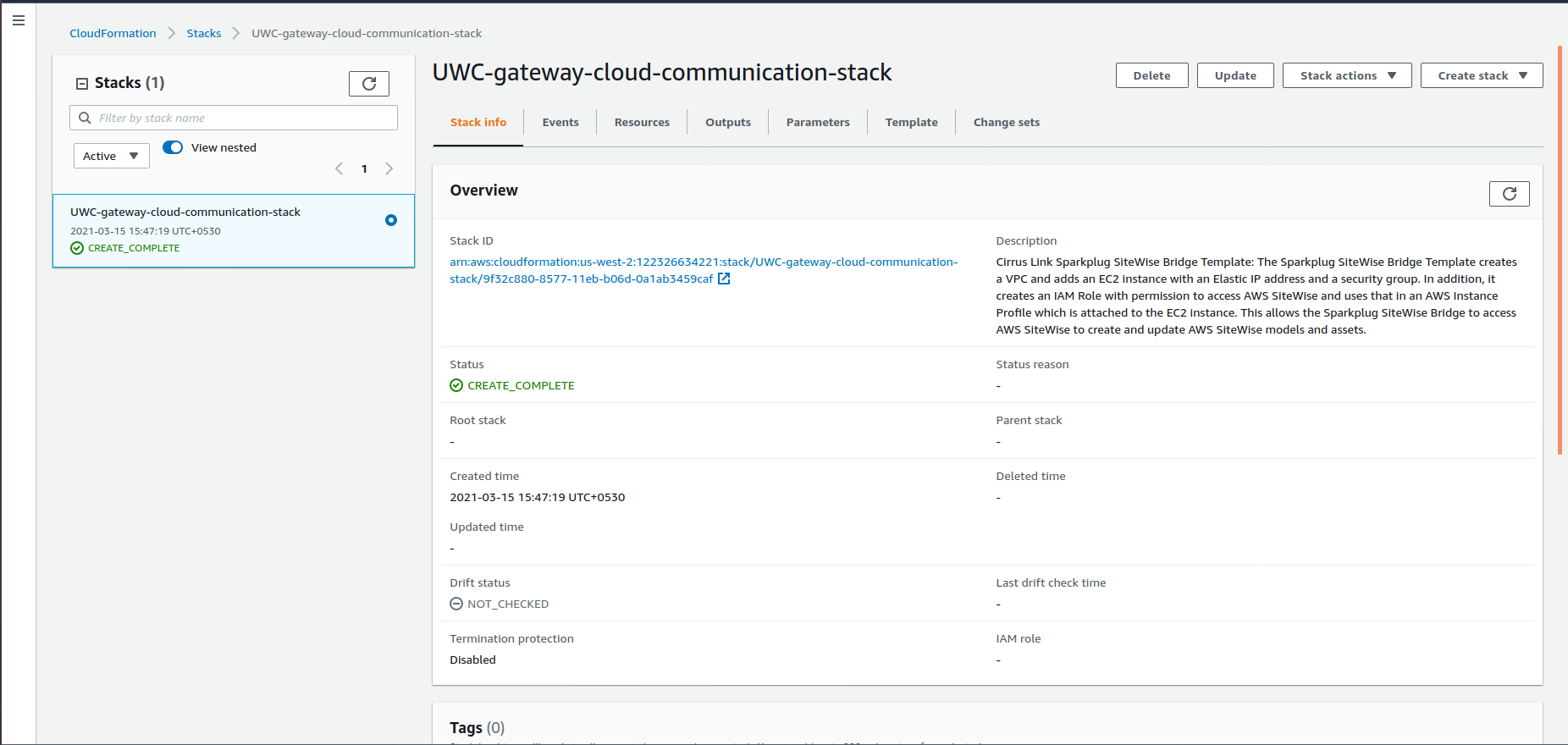
**• Installation and Configuration**

1. SSB installation and cloud infrastructure provisioning

* We need to provision the AWS infrastructure and install SSB in the EC2 instance. Please hit the following link to carry out the procedure -

<https://docs.chariot.io/display/CLD80/IBAS%3A+Installation>

* Kindly note that there will be two different delivery methods for SSB. We have to choose one among them. We are using the 'cloudformation template’ method currently, thus, please opt for the same.
* Once the process is completed, you should be able to see the status as 'CREATE\_COMPLETE'



1. AWS IoT core broker and SSB configuration

* A ‘thing' needs to be created in AWS IoT core which will represent the IoT edge device. In our case, the edge device will be the UWC gateway. Later, SSB needs to be configured so that it can access IoT core to fetch the on-prem data. Please use the following link to carry out the complete procedure - https://docs.chariot.io/display/CLD80/SSB%3A+Quickstart
* Alternate link to get an insight on the creation of a 'thing' in AWS IoT core - <https://docs.aws.amazon.com/iot/latest/developerguide/iot-moisture-create-thing.html>

1. UWC gateway configuration

* The UWC gateway package needs to be deployed on the gateway device. Kindly refer the corresponding user guide in order to achieve the same.
* While deploying the UWC Gateway package, make sure you have configured the appropriate SSL certificates and broker details procured from AWS cloud.
* SSL certificates which were created in STEP 2 during the creation of a ‘thing’ in AWS IoT core must be inputted while running the ‘01\_pre-requisites.sh’ script.

- $sudo ./02\_provision\_UWC.sh --**deployMode**=dev --**recipe**=3 --**isTLS**=yes --**caFile**="/<path>/root-ca.crt" --**crtFile**="/<path> /client**.**crt" --**keyFile**="/<path> client.key" --**brokerAddr**="azeyj7bji4ghe-ats.iot.us-west-2.amazonaws.com" --**brokerPort**=8883 --qos=1

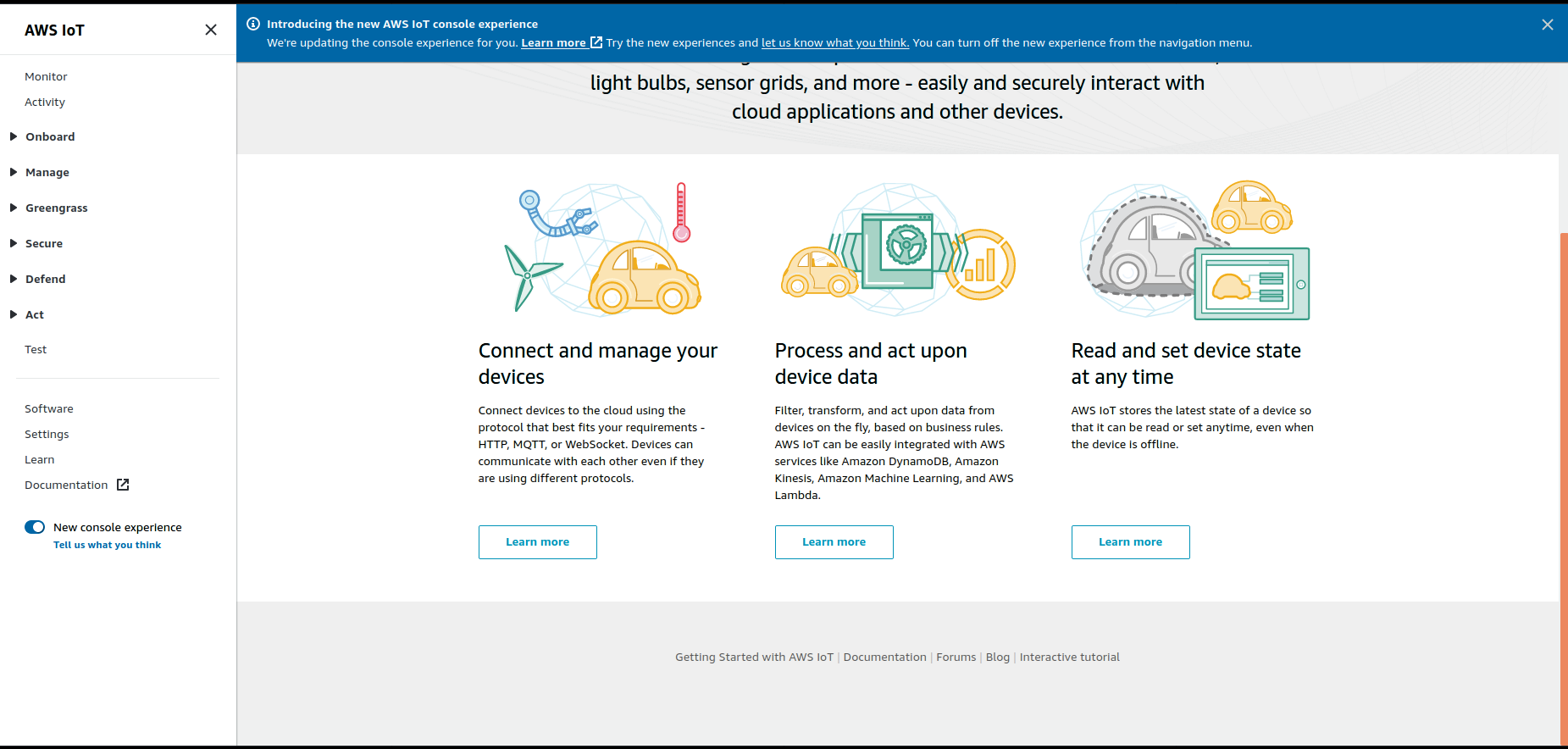
NOTE: The Certificate are required to be generated in .crt format & with naming structure given below,

CA Certificate: root-ca.crt

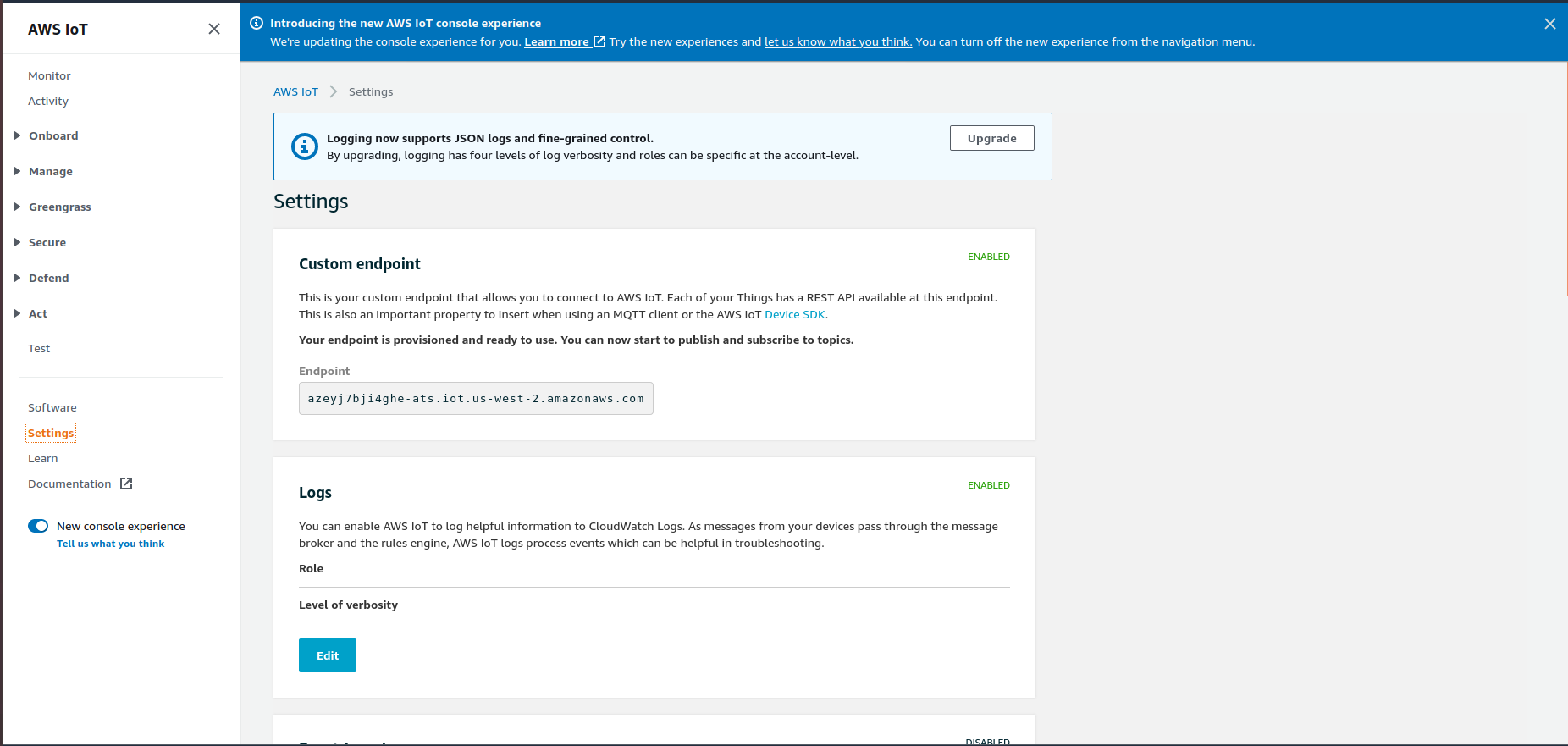
Client Key: client.key

Client certificate: client.crt

* Deploy Mode ‘dev’ or ‘Prod’.
* Select Reciepie as 3 to have Sparkplug Container deployed.
* Make sure the ‘isTLS’ argument is set to 'yes'.
* Configure the ‘caFile’ argument with the path of the CA certificate obtained from AWS IoT core.
* Configure the ‘crtFile’ argument with the path of the client certificate obtained from AWS IoT core.
* Configure the ‘keyFile' argument with the path of the client private key obtained from AWS IoT core
* ‘brokerPort’ should be set to ‘8883’
* ‘brokerAddr' should be set to the custom endpoint of the AWS IoT core. Use the following couple of steps to fetch the custom endpoint.
* Go to the IoT core console. Hit the ‘Settings’ tab in the left pane

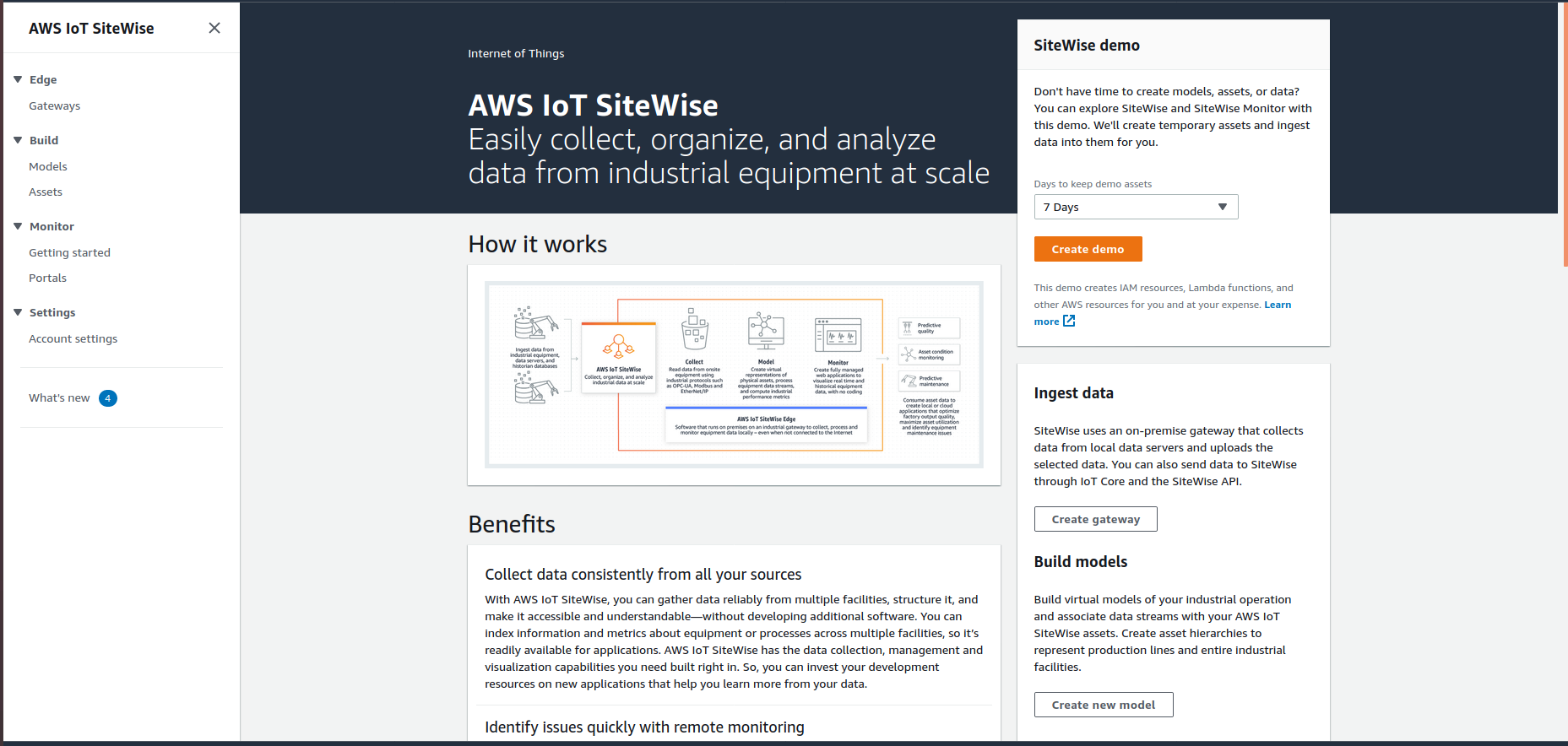


- You can see the custom endpoint which represents the IoT core broker address. This address needs to be configured in the ‘brokerAddr' argument.

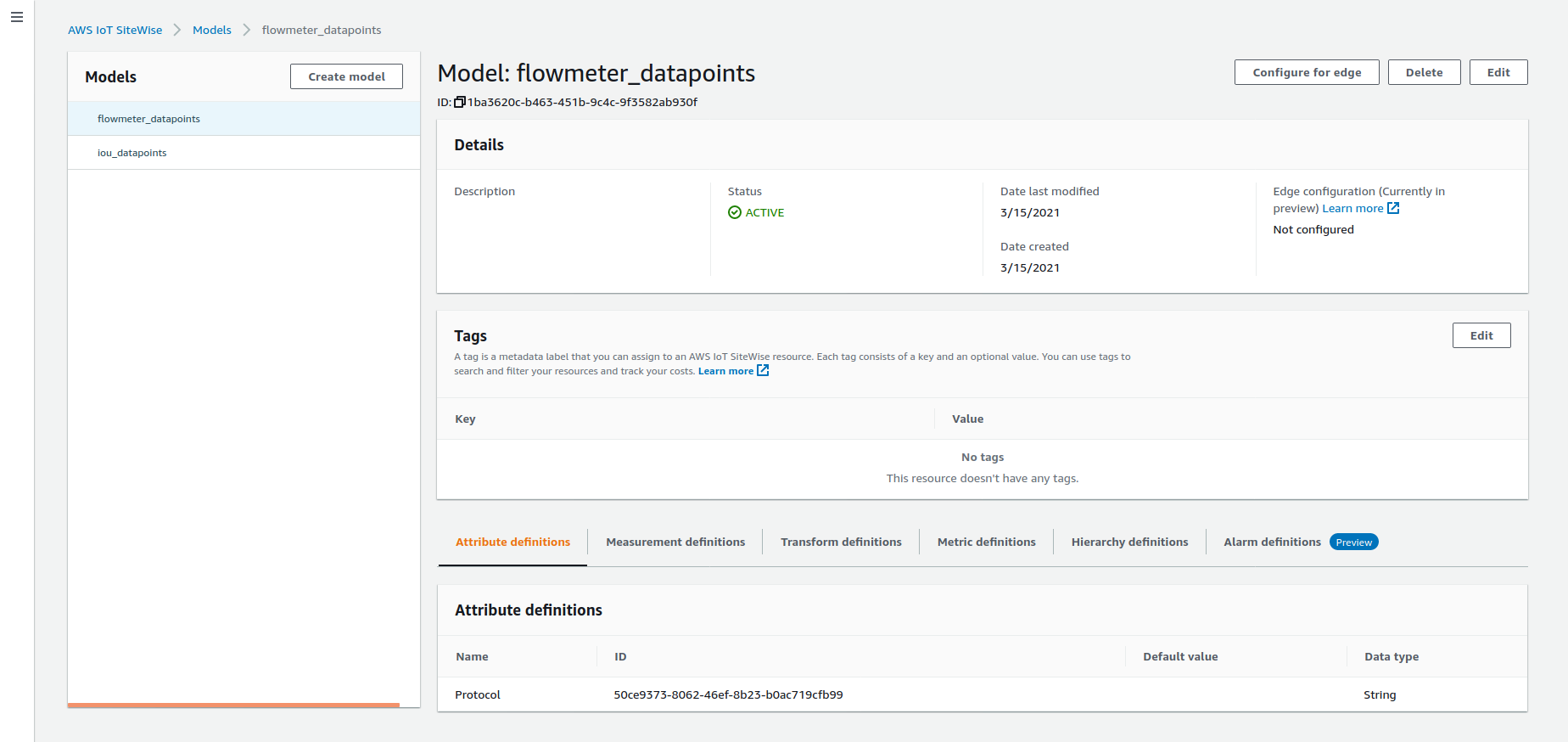


**• Monitor Data on Cloud**

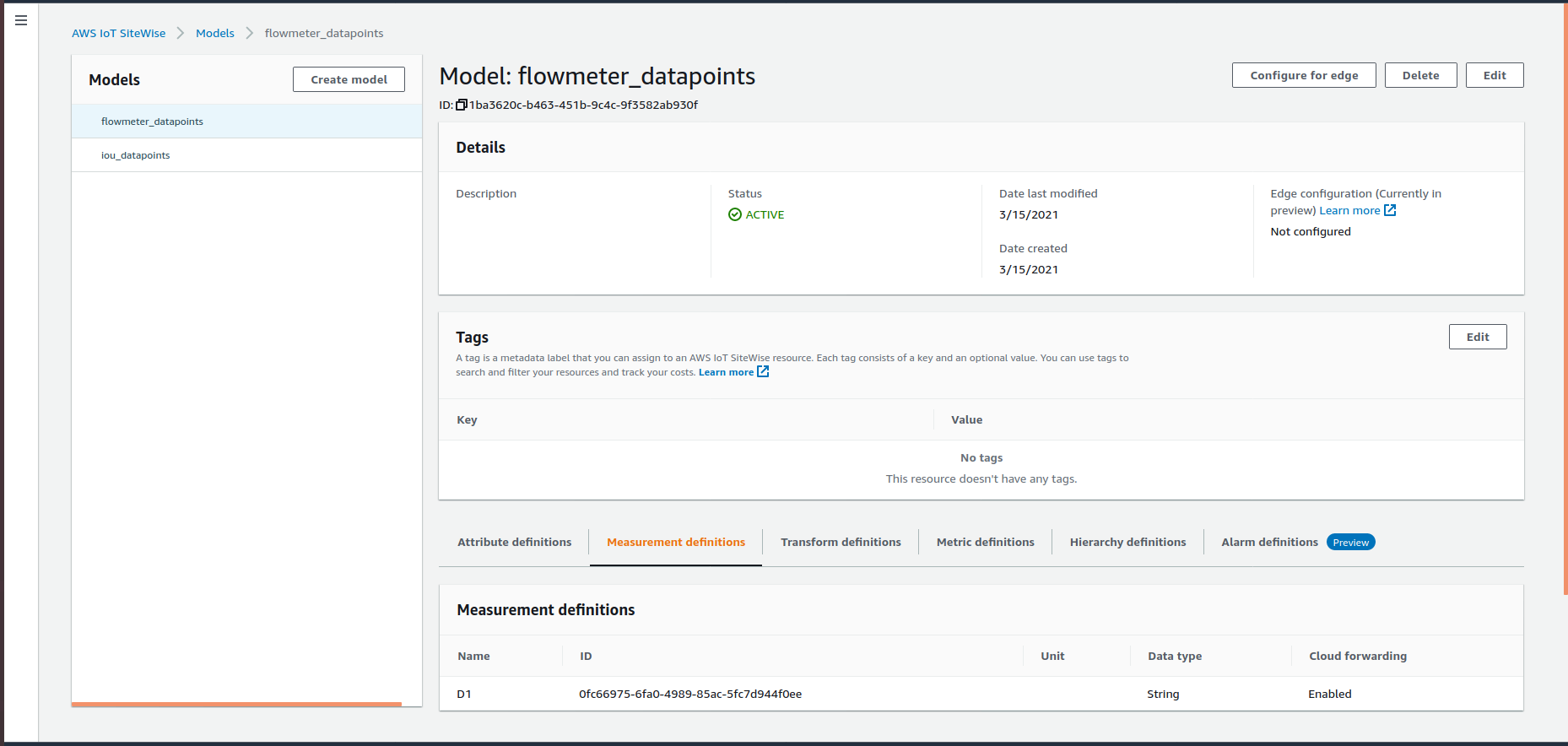
* Once all the above mentioned installations and configurations are done, it is now time to visualize the data.
* The data can be monitored on the AWS Sitewise service.
* Scroll to the AWS Sitewise service in the AWS management console.



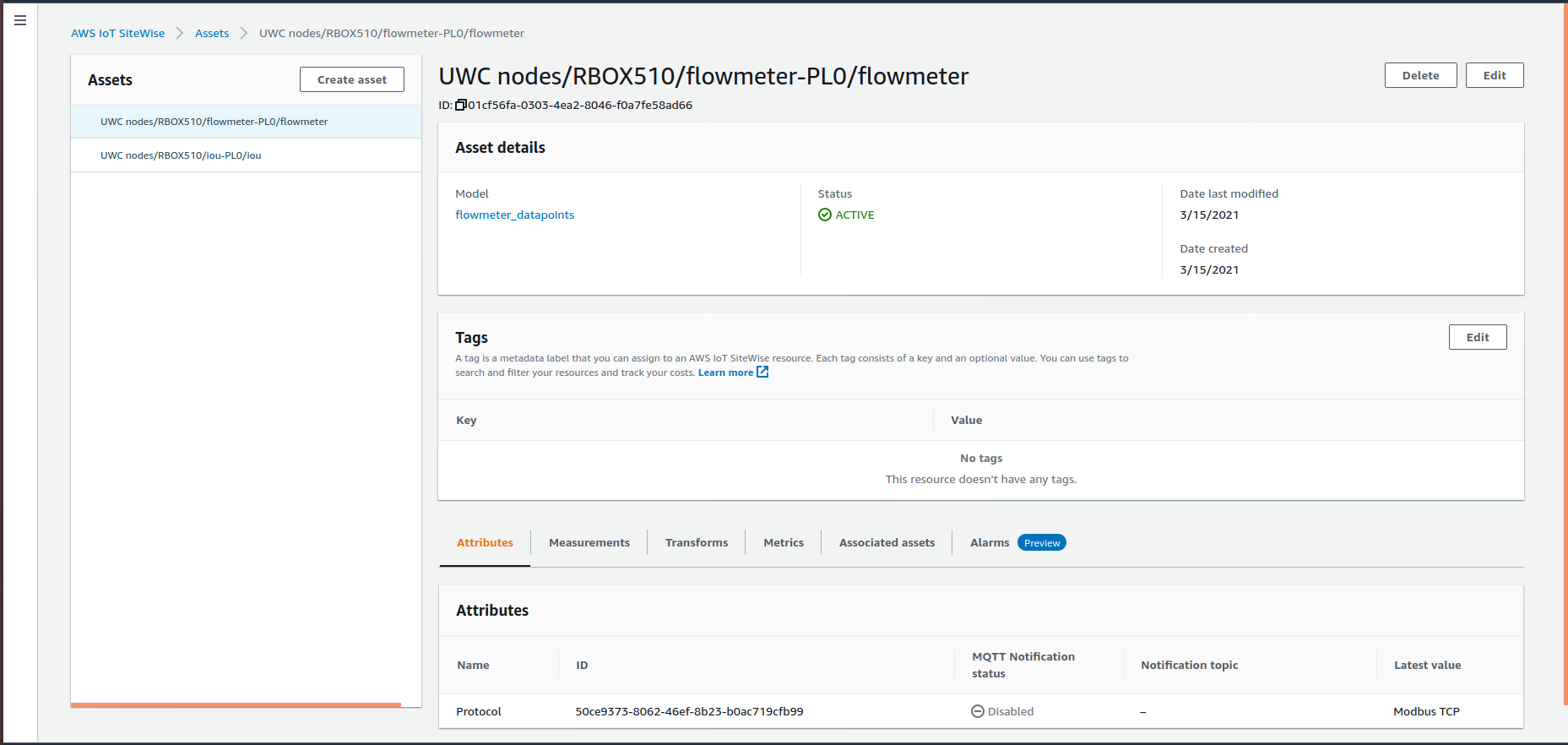
* Go to the ‘Models’ tab. You can see two models. The attribute ‘Protocol’ of a model can be visualized.



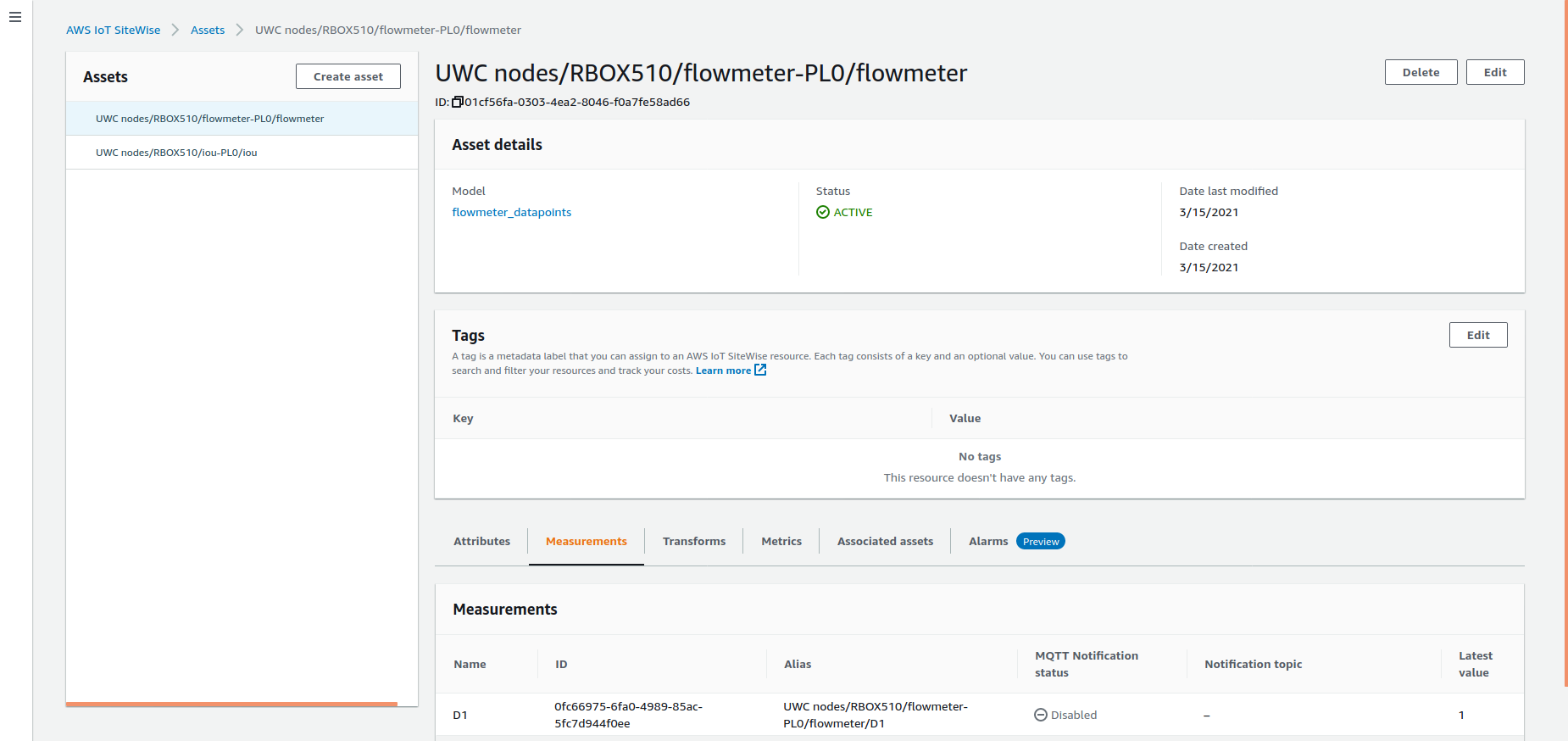
* The ‘measurement' parameter representing a data point ‘D1’ can be seen in a model.



* Go to the ‘Assets’ tab. You can see two assets. The attribute ‘Protocol’ can be visualized with its value defined.



* The ‘measurement' parameter representing a data point ‘D1’ can be seen in the asset with its value defined.



**NOTE: One should delete old Assets & Models from AWS IoT to ensure the updated Assets and Models get reflected. Duplicate Assets and Models will not refreshed.**