Kalki IO Edge for AMI 2.0 using AWS Greengrass

Configuring AMI 2.0 setup

www.kalkitech.com

Version 1.1.0, October 2024

Copyright Notice

Disclaimer

Information in this manual is subject to change without any notice. Information provided in this manual is intended to be accurate and reliable. However, Kalki Communication Technologies assumes no responsibility for its use. Kalki Communication Technologies Pvt. Ltd. makes no commitment to update or keep the current information in this manual.

Contact Information

Kalki Communication Technologies Pvt. Ltd.
The Address, 4th Floor,
17/1, Outer Ring Road,
Opposite to Prestige Cessna
Business Park Kadubeesanahalli,
Bangalore - 560103, INDIA
Telephone: +91- 8067021900
Email: sales@kalkitech.com

Technical Support Contact Information

Online Technical Enquiry Support: www.kalkitech.com/support
E-mail Support: support@kalkitech.com

Table of Contents

Terms and abbreviations

- KIOE: Kalki IO Edge. This is provided along with the amazon machine image (AMI). For more information refer link.
- DLMS Simulator: Used to simulate meter data using DLMS protocol. This is provided along with the amazon machine image (AMI).
- AMI: Amazon Machine Image. Used to create EC2 instances with preconfigured setup.
- AMI 2.0: Advanced Metering Infrastructure 2.0.
- Greengrass: Edge computing software provided by AWS. For more information refer Link.

1. Introduction

This document focuses on setting up the environment for AMI2.0 (DLMS simulator + Kalki IO Edge + Greengrass). The amazon machine image (AMI) consists of DLMS simulator and Kalki IO Edge installed. Later a Cloudformation script provided by the Kalkitech team needs to be run to set up AMI 2.0 with Greengrass.

The AMI 2.0 setup provided by Kalkitech consists of DLMS simulator which simulates the meter data sent to Kalki IO Edge for protocol conversion. The converted data is then sent to Greengrass for edge computing followed by sending to AWS cloud (IoT Core) through MQTT.

Note: The CloudFormation script needs to be followed only if the user wants to set up AMI 2.0 setup provided by Kalkitech. If the user expects a machine image with only the meter simulator and Kalki IO Edge, they may follow steps 2.2 or 2.3 after completing 2.1.

1.1 Pre-requisites: License and files Request

The following steps need to be followed to issue a license and to get access to SCT too, a GUI tool used to install and configure KIOE.

Send an email requesting the license for KIOE to ami2meter@kalkitech.com. Mention if you just want the licensing of KIOE or both license and CloudFormation scripts to setup AMI2.0. The issued license will be a trial license valid for 3 weeks. If the user wishes to get a license for a longer period, mention it in the request mail.

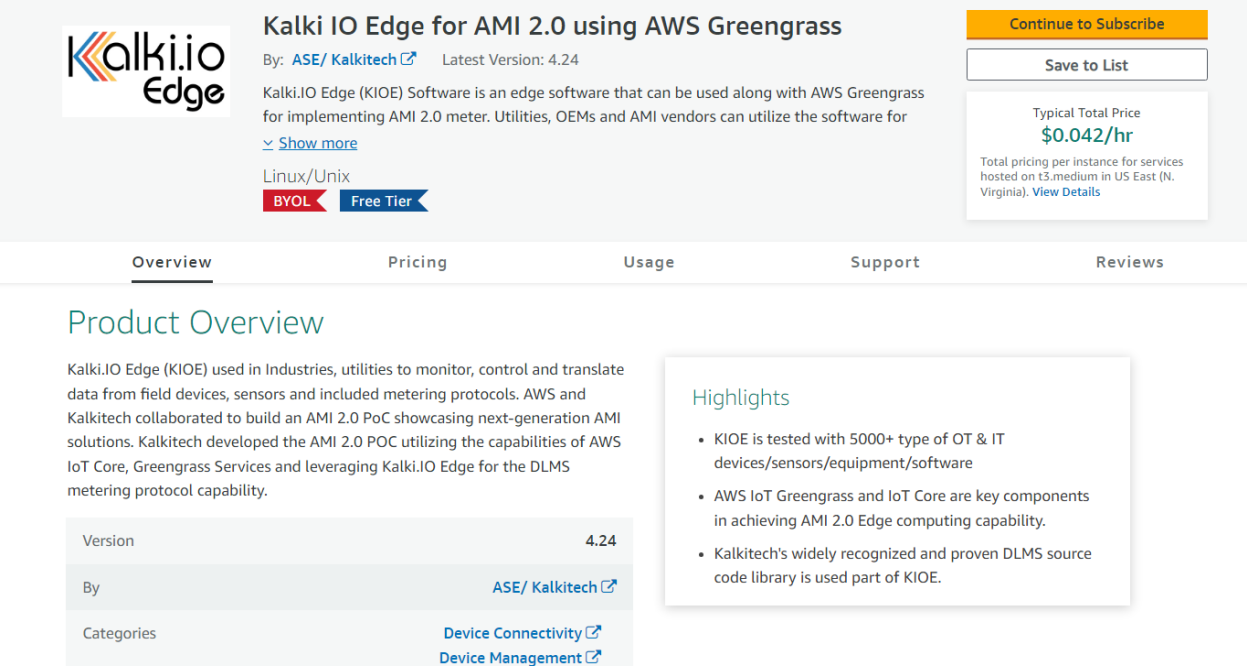
NOTE: Add the message note as “Request for KIOE license request for AMI 2.0 using Greengrass” when sending the request. Usually, it will take 1 business day to issue the license.

2. KIOE AMI 2.0 Setup machine image

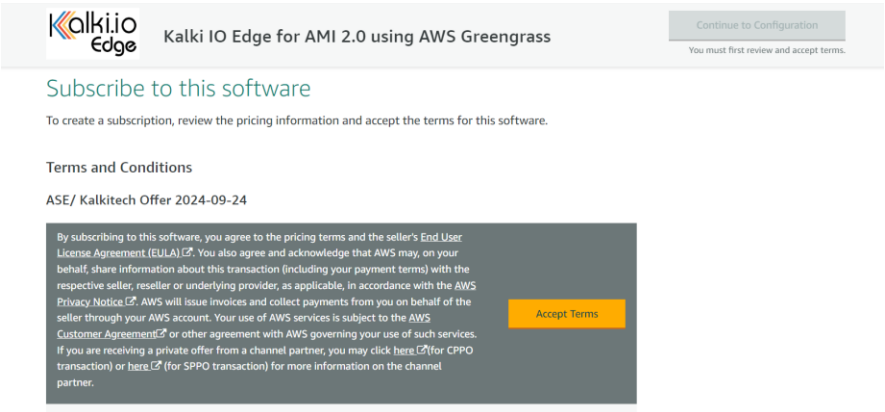
This section will guide you to provision Kalki IO Edge for AMI 2.0 that uses AWS Greengrass on your account. The machine image is available here on AWS Marketplace.

2.1 Kalki IO Edge for AMI 2.0 using AWS Greengrass AMI subscription

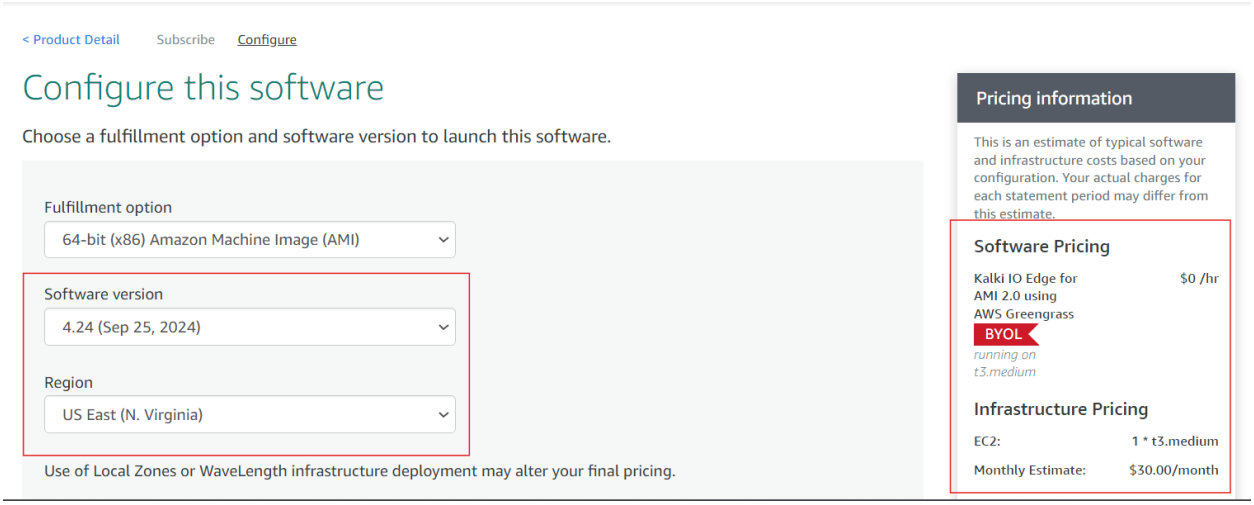
1. Subscribe to KIOE AMI by clicking on Continue to Subscribe. KIOE AMI is free for evaluation, however you may get charged for AWS resources like compute, storage etc.



2. Successful subscription will take you to the Subscription page. Read and Accept EULA for Kalki IO Edge for AMI 2.0 using AWS Greengrass. Note: It may take some time for subscription request to complete.



3. Select KIOE Version and Region of your choice. You can see the corresponding Monthly bill Estimation on right hand side. This Estimation is for t3.medium. Click on Continue to Launch to move to the next step. **Note: Estimation does not include KIOE license price.**



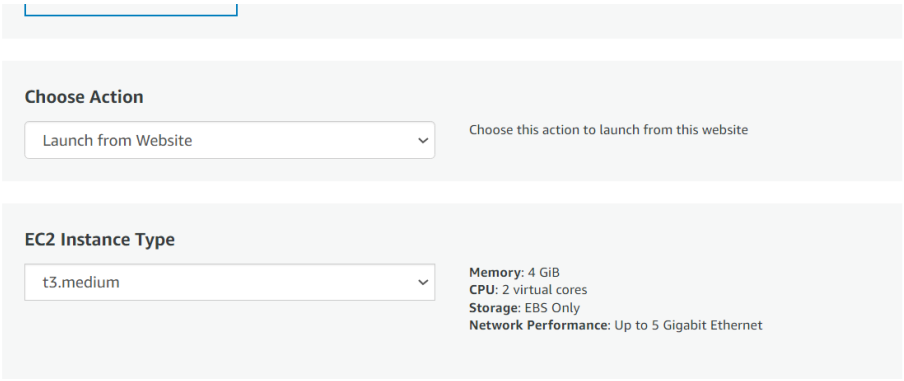
4. AWS provides two methods of launching new instances. **Launch from Website** and **Launch through EC2**. Section 2.2 Explains provisioning AMI 2.0 from Website. Section 2.3 explains **launch through EC2**. If you are new to AWS and provisioning an EC2 instance for the first time, then it is recommended to use **Launch from Website**.

   Note:

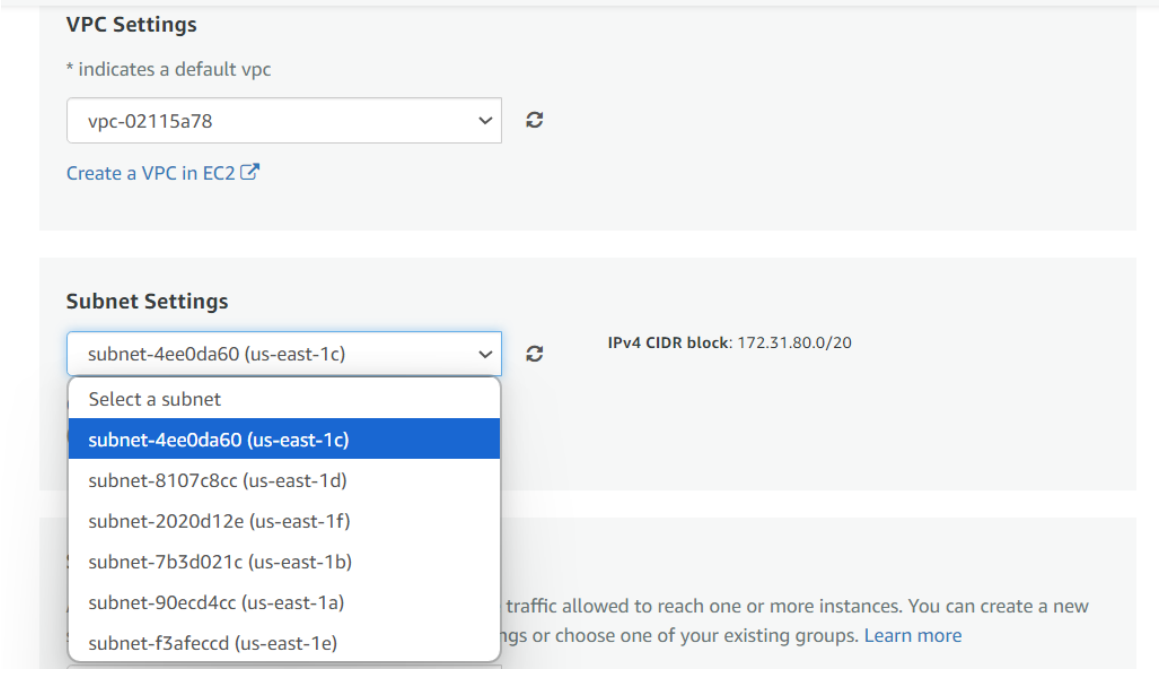
   - If you want to install Greengrass and create Greengrass components on your own, proceed with **Launch through EC2 (Section 2.3)** or **Launch from website (Section 2.2)** option.
   - If you want to install **AMI 2.0 setup provided by Kalkitech (using Cloudformation script)**, proceed with **Section 2.5.**

2.2 AMI 2.0 Provisioning through website

1. Set Launch Action to Launch from Website.

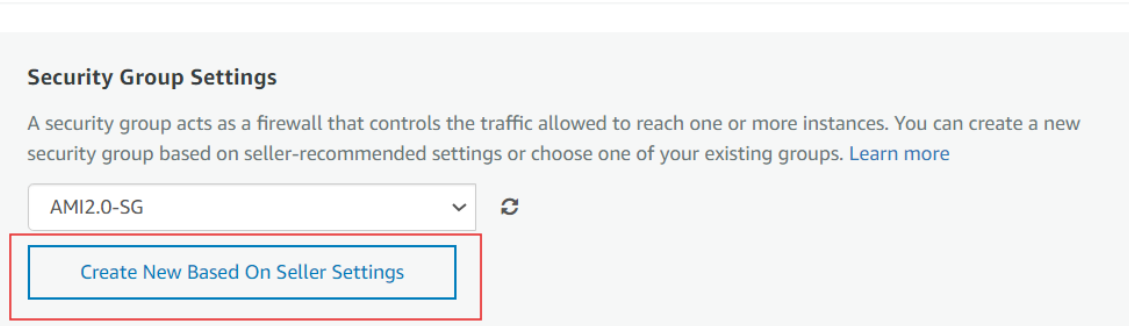


2. This will bring several settings on the same page regarding KIOE AMI launch configuration.
3. Select VPC for launch. This VPC should have at least one subnet with public access. Note: If you have not yet created any VPC in your account then default VPC will have subnets with public network access. This VPC can be selected for AMI 2.0.
4. Select public subnet of your VPC.
   Note: If you are using default VPC then all subnets are public.

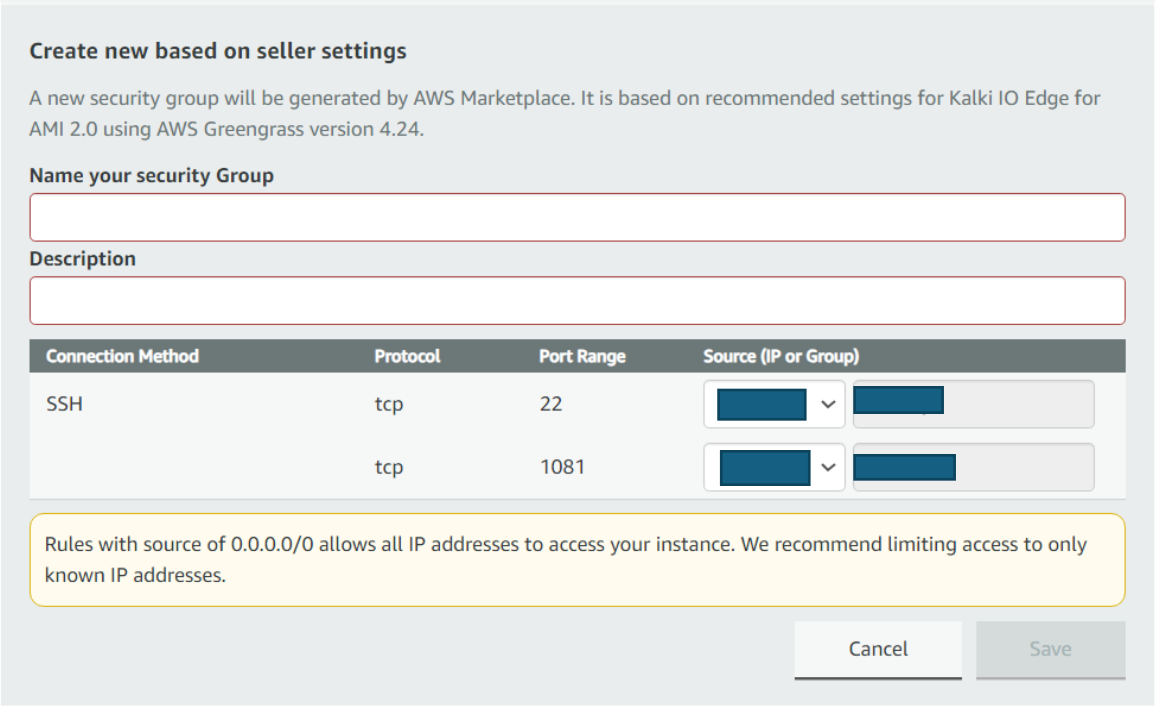


5. KIOE AMIs security group must allow access to SSH TCP port(22) and KIOE Configuration TCP Port (1081). Please limit port 22 and 1081 access to only your IPs/security groups. You can select your existing Security Group which allows these inbound connections by selecting one of available Security Group from Drop down in Select Security Group. You can create a new Security Group with above

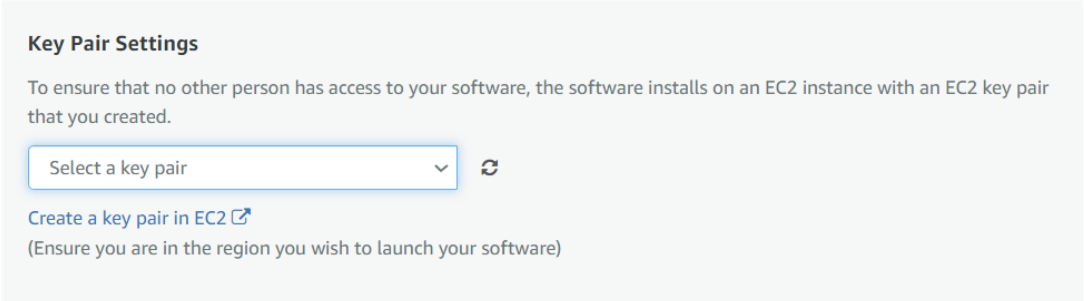
mentioned TCP ports open for inbound TCP connection by clicking on **Create New Based on Seller Settings.**



This will bring a prompt to create a Security Group. Fill Security Group Name and Description and click on save to create Security Group.



6. Select SSH key pair to be used for logging in to your EC2 instance. You can either select an existing SSH key or create a new key.

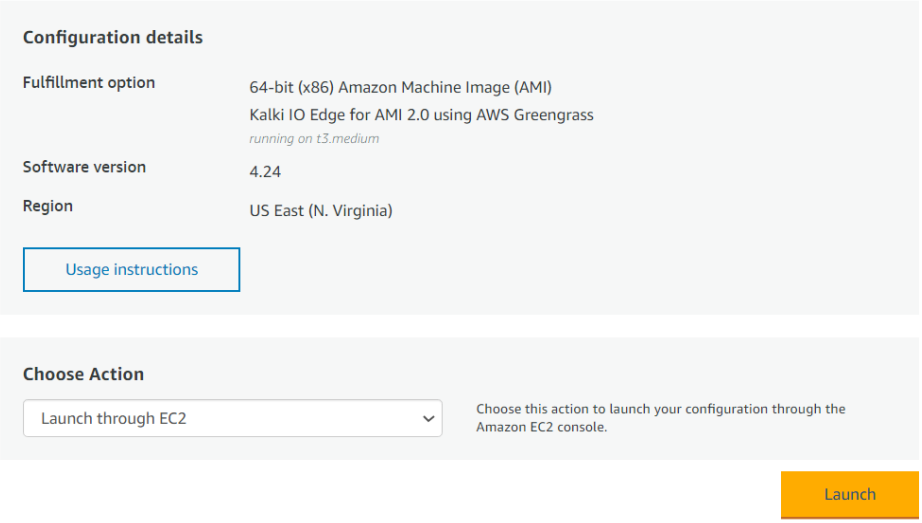


7. Click on the Launch button to launch the KIOE AMI. You should see a similar window stating the successful launch.
8. It will take couple of minutes for KIOE AMI to be provisioned. You can find your instance information in Services -> EC2 -> Running Instances. Now you should go to Section 2.4 and validate KIOE AMI.

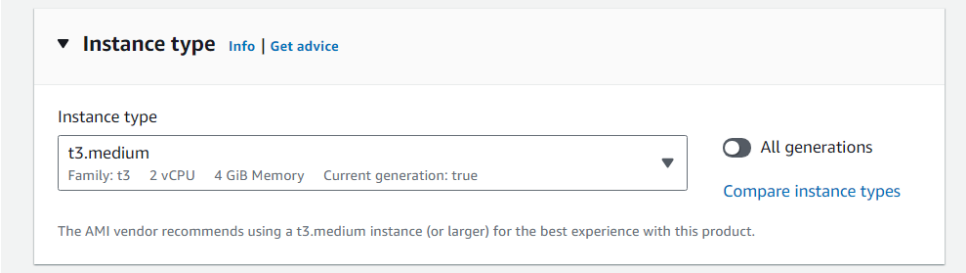
2.3 AMI 2.0 Machine Image Provisioning from EC2 console

It is assumed that you have already subscribed to KIOE AMI. Refer Section 2.1 for Subscription procedure. Follow these steps to install AMI through EC2 launch wizard.

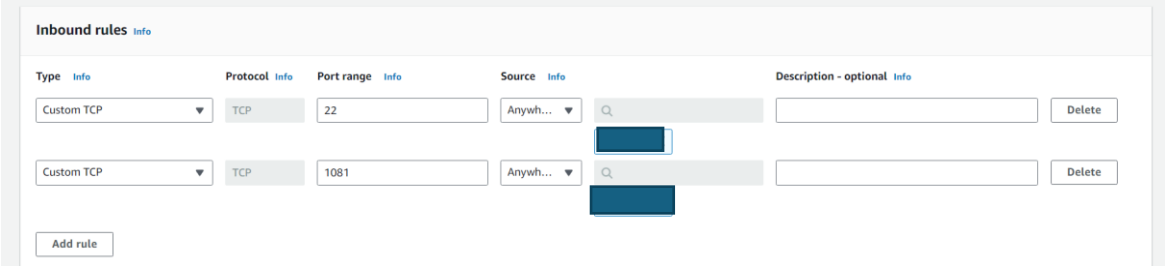
1. Select launch through EC2 and click launch.



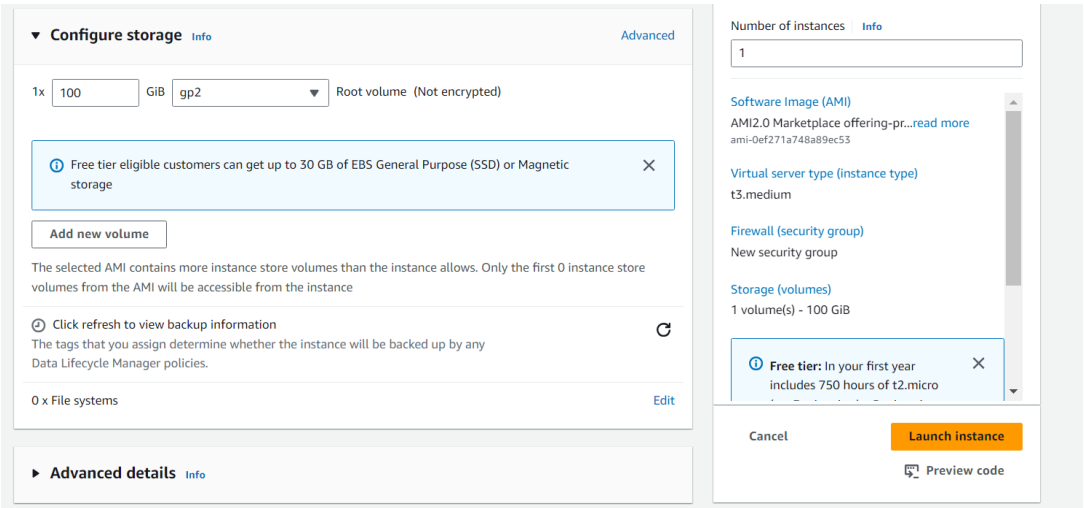
2. Select Instance type. It is recommended to use t3.medium for evaluation. You may choose any of the allowed instance types. For more information on Instance type selection refer Appendix A.



3. Create a key pair for using SSH. Provide storage as required, by default the volume is set to 100GB. It is recommended to provide at least 20GB of storage.
4. KIOE AMIs security group must allow access to SSH TCP port(22) and KIOE Configuration TCP Port (1081). Create a new security group from the EC2 console with inbound port 22 and port 1081 with access to the required IPs.



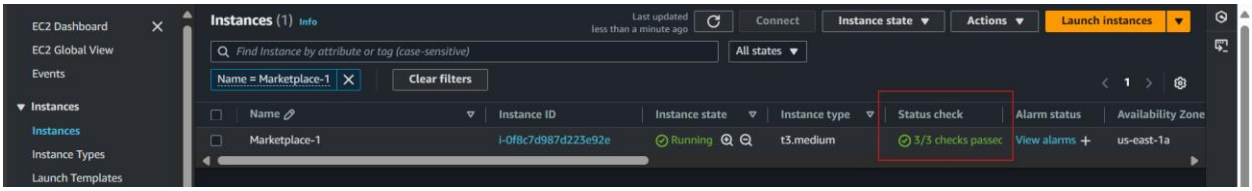
5. Add tags as per your need and click to launch the instance.



2.4 Validating AMI 2.0 instance

These steps will help in validating whether KIOE AMI has been successfully provisioned with required permissions

1. EC2 Instance instantiation Check: Navigate to Services -> EC2, In EC2 dashboard and open Running Instances. Select KIOE EC2 instance.
2. Here you will see a list of EC2 instances. Make Sure that instance State for KIOE is Running and Status Checks have passed. If you have recently provisioned the AMI, then it may take couple of minutes for Status Checks to come as 3/3 Checks Passed. This confirms the successful instantiation of AMI. Note down the Public IPv4 address of instance from the instance details. This IP address will be used for configuring KIOE. (Note:- You may also associate an elastic IP to the instance, but be aware of the potential costs)



2.5 Cloudformation script to install AMI 2.0 setup

These steps are used to set up AMI 2.0 setup environment by Kalkitech using CloudFormation script. Make sure the user has the necessary permissions to run CloudFormation and create all resources meant to be created by the CloudFormation script. If the user does not have permission to create the resources, create an IAM role with necessary permissions for CloudFormation and attach the role to CloudFormation stack while running the script. The steps will cover the following:

1. PreRequisite Script creates IAM role necessary for EC2, security group for KIOE, and creates an SNS topic for notification.
2. Script (1) Installs necessary packages and creates a Greengrass v2 core device.
3. Script (2) creates and attaches necessary IAM policies for Greengrass core device to access S3 folder in which the code of all the Kalki Greengrass components will be stored.
4. Scripts (3a), (3b), (3c) Creates Greengrass components and Script (4) deploys the components into the Core Device.

Note: Make sure that you are running the script in one of the following regions:

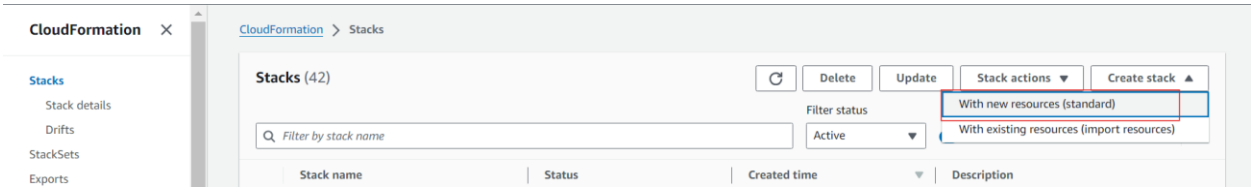
1. us-east-1 (N.Virginia)
2. us-east-2 (Ohio)
3. us-west-2 (Oregon)

2.5.1 Prerequisite Script

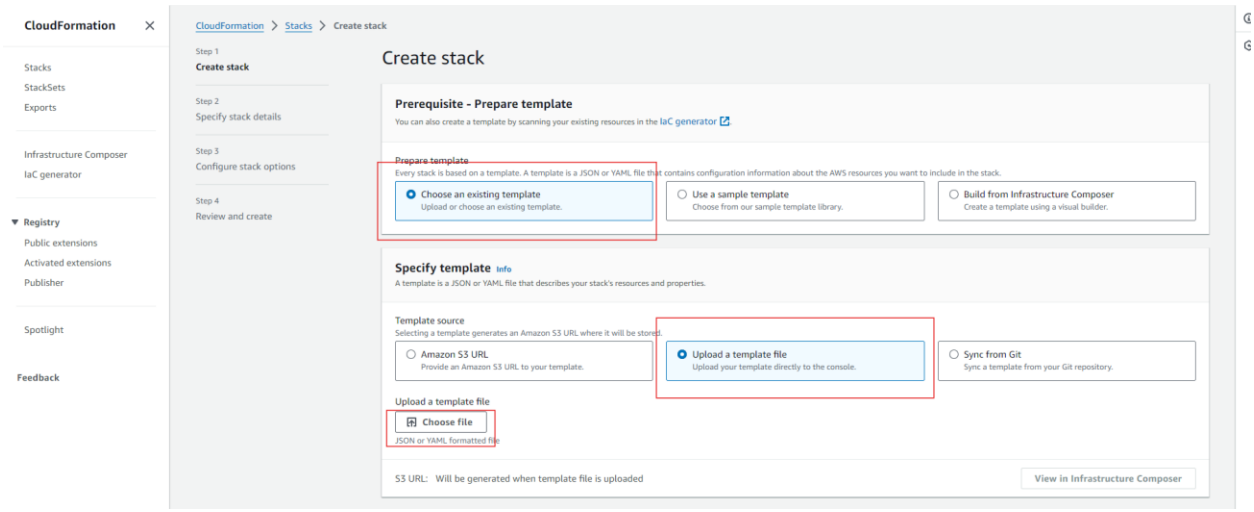
This script needs to be run initially before the scripts for creating Greengrass device are run.

1. Go to CloudFormation console and click create stack

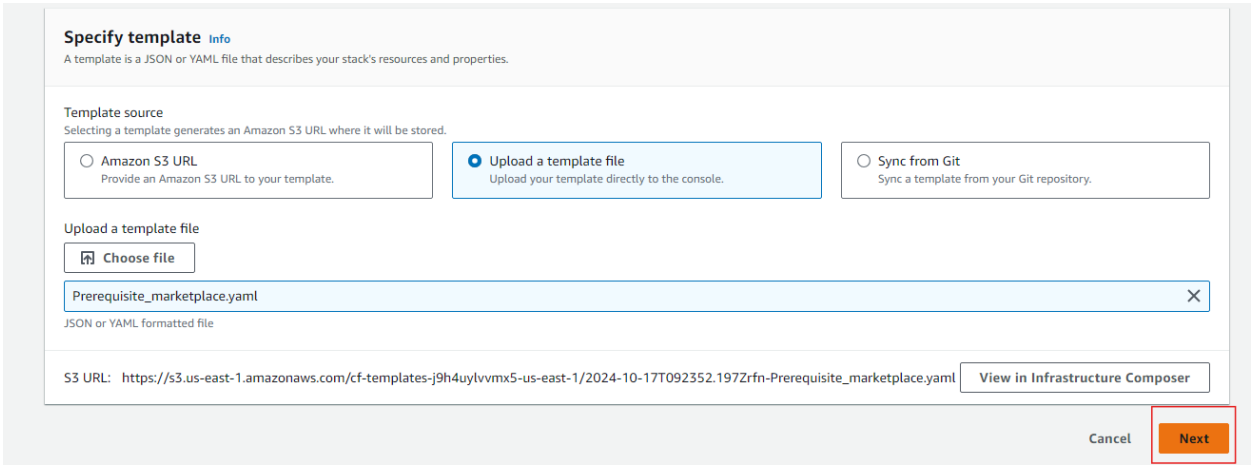
2. Select create new resources



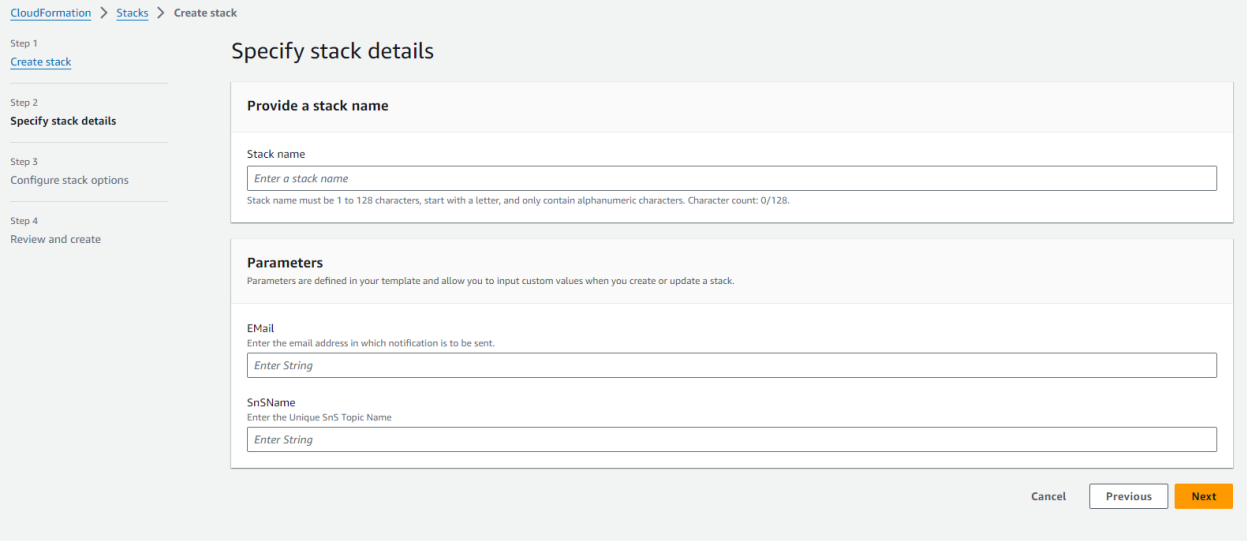
3. Select upload an existing template, upload a template file and attach the Prerequisite CloudFormation script provided. Refer below screenshot.



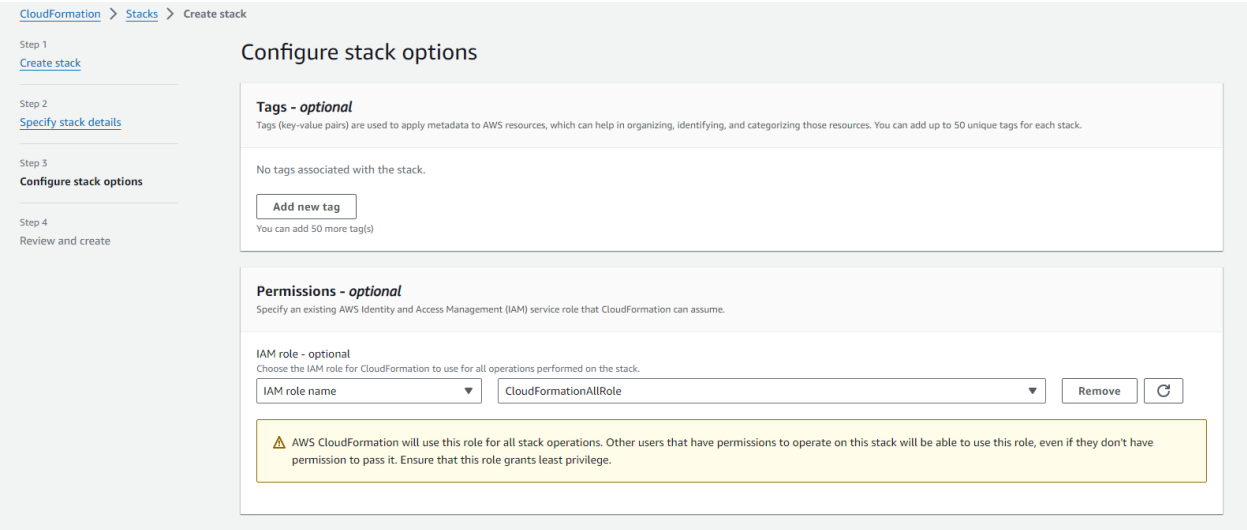
4. After selecting the file click on next.



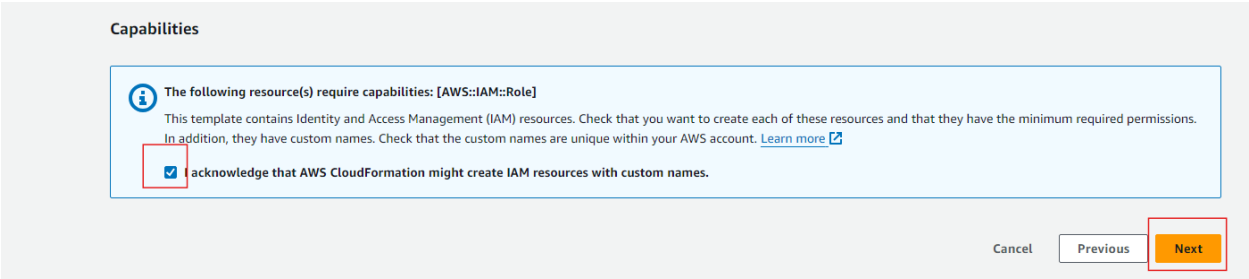
5. Enter a stack name of your choice for the script. For email, give a valid email. This will create an SNS topic to the given email to send notification if power exceeds the threshold value.
6. Provide a unique name for the SNS trigger.



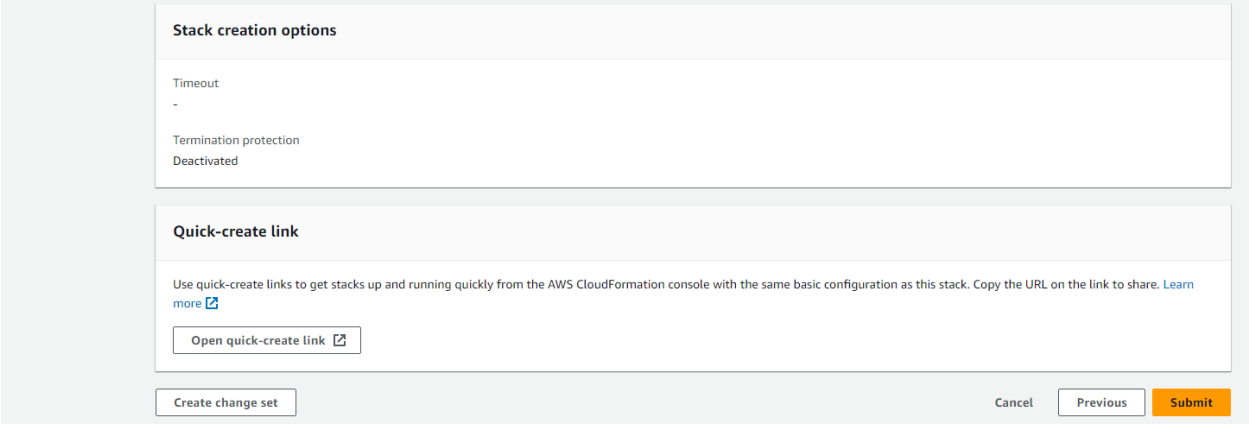
7. Selecting a role for CloudFormation is optional. If the user has the necessary permission to create the resources, you can skip this step.



8. Acknowledge the IAM role creation check box and click on next.



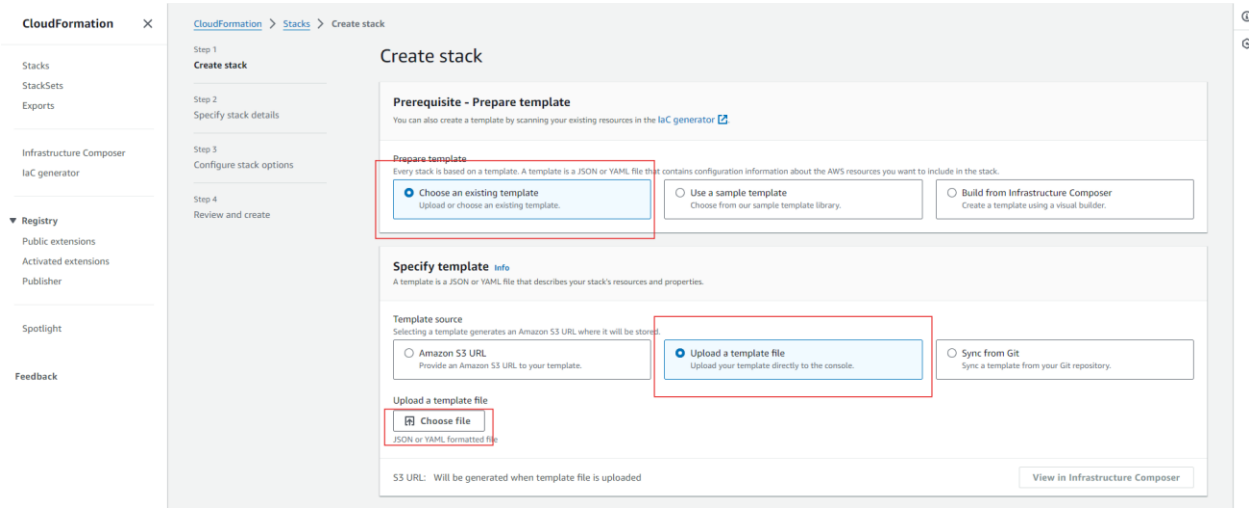
9. Review the options selected and select submit.



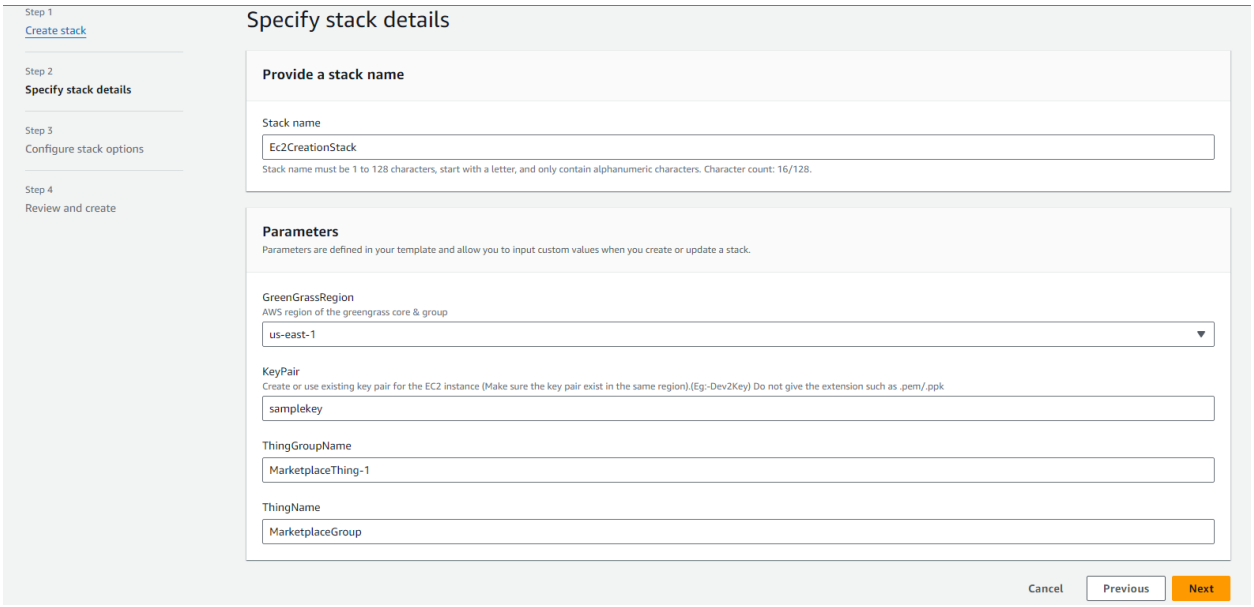
2.5.2 Script (1) To create Greengrass core device

Make sure you have run the prerequisite script before running this script. Create a key pair manually from the EC2 console before running the script as it will ask a Key Pair name as parameter. This key pair will be used to SSH into the instance.

1. Go to CloudFormation console and select create new resources.
2. Select upload an existing template, upload a template file and attach the (1)CloudFormationGreenGrassSetupForAwsEc2Ubuntu-RunForEachThingCreation.yaml script provided.



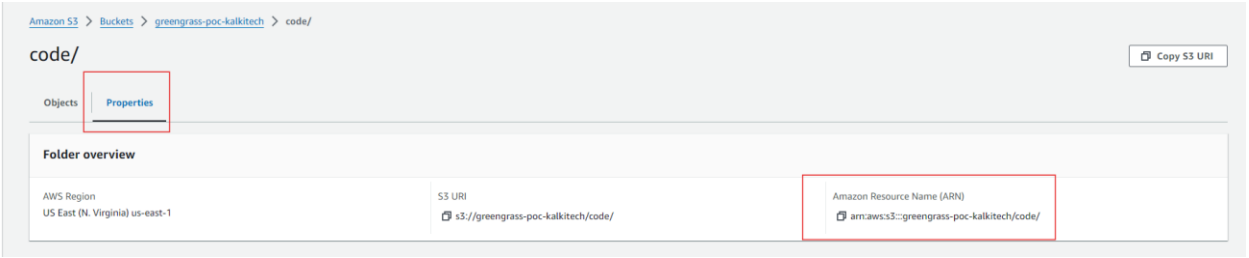
3. After selecting the (1)CloudFormationGreenGrassSetupForAwsEc2Ubuntu-RunForEachThingCreation.yaml click on next.
4. Enter a unique stack name of your choice. For parameters, enter the same region in which you are running the CloudFormation from. Make sure that you use one of the three mentioned regions.
5. Provide the key name which was created from EC2 console as mentioned in the initial step of this section, a thing name and thing group name of your choice and click next.

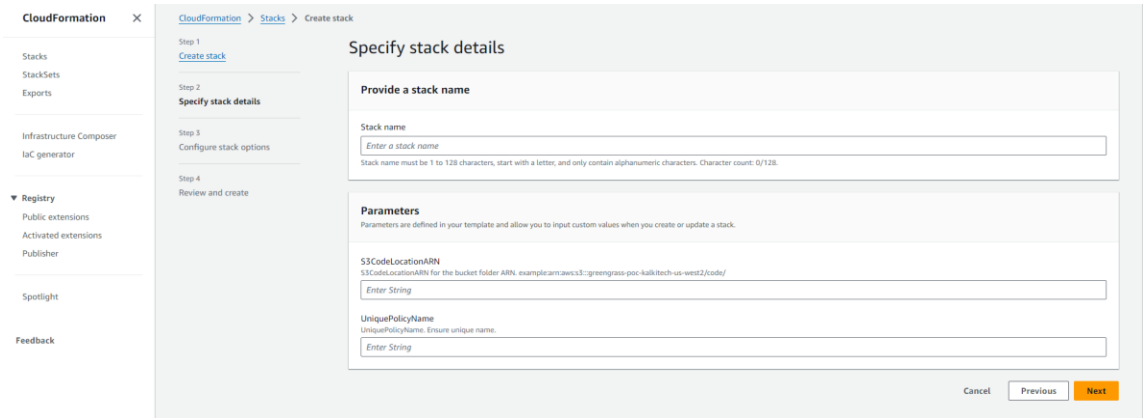


6. Continue till reviewing the inputs and click submit.

2.5.3 Script (2) to create and attach bucket policy

1. Go to CloudFormation console, create a stack with new resources, upload the (2) script and click next.
2. Give a stack name of your choice. For parameters, give the S3 location ARN in which you will store the provided code and a unique name for the same. You can find the ARN of folder under properties section in the folder.



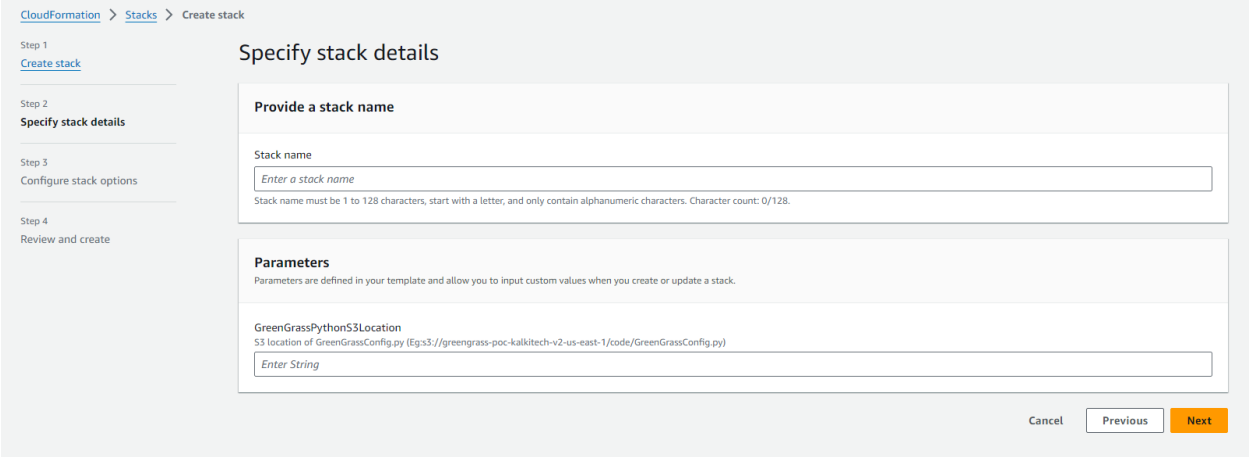


3. After providing the parameters, continue till the end, review the inputs and click on submit.

2.5.4 Script (3a), (3b) (3c) - Used to create components for Greengrass.

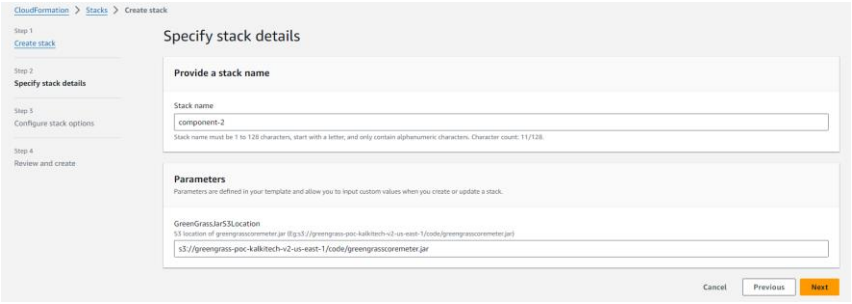
Script (3a) parameters

1. S3 URI of the file GreenGrassConfig.py



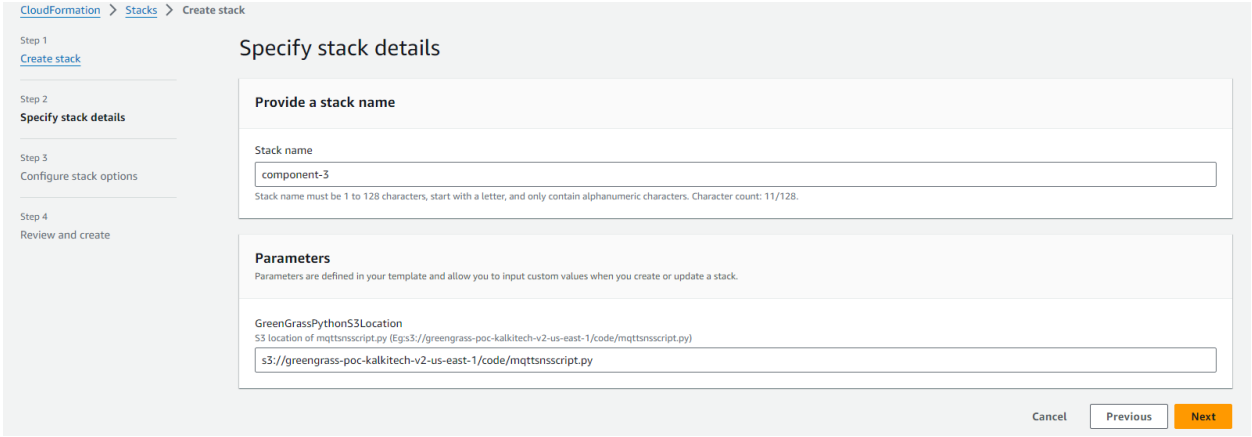
Script (3b) parameters

1. S3 URI of the jar file (greengrasscoremeter.jar)



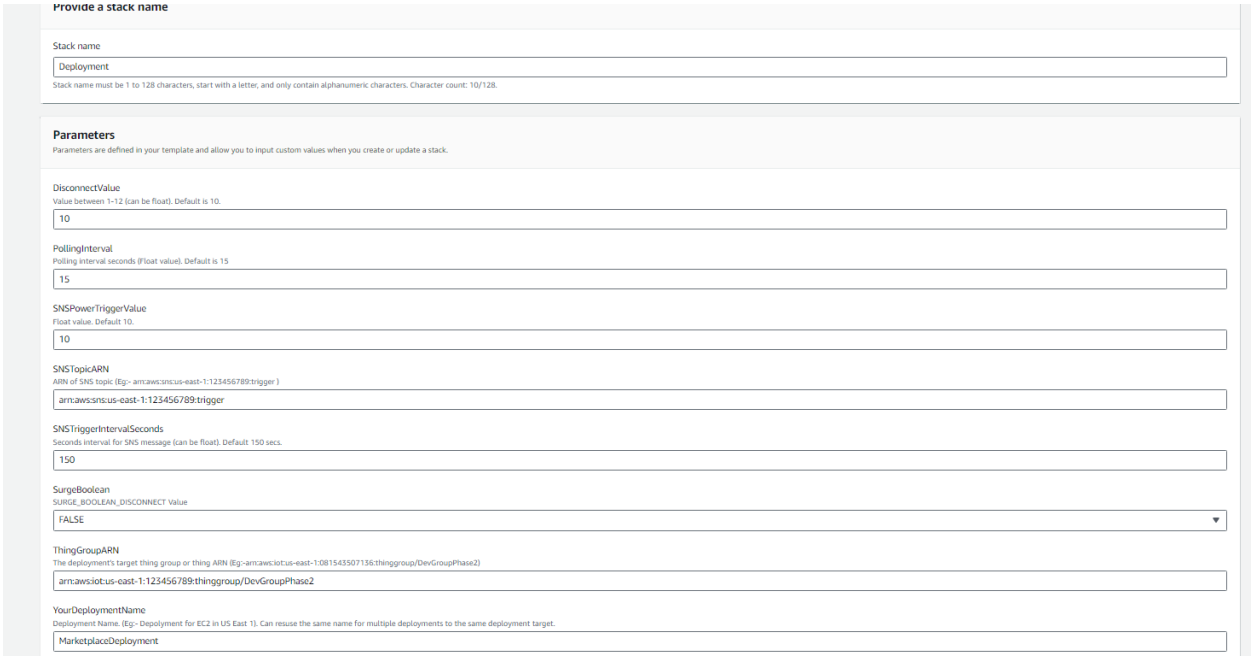
Script (3c) parameters

1. S3 URI of the file mqttsnsscript.py



2.5.5 Script (4) Script for deploying the components

1. There are some default values which are provided, you can change if required.
2. Provide the SNS trigger ARN which was already created in the prerequisite script. This can be found in SNS console.
3. Provide the thing group ARN in which the core device is made. This can be found under AWS IoT console -> All devices -> Thing Groups
4. Provide a Deployment name of your choice.

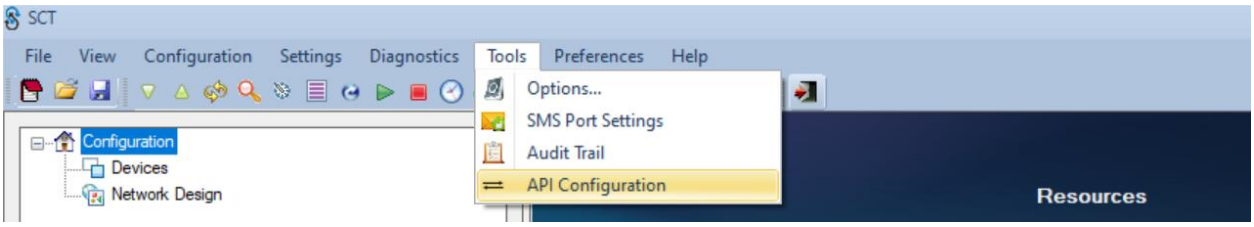


3. Kalki IO Edge Licensing

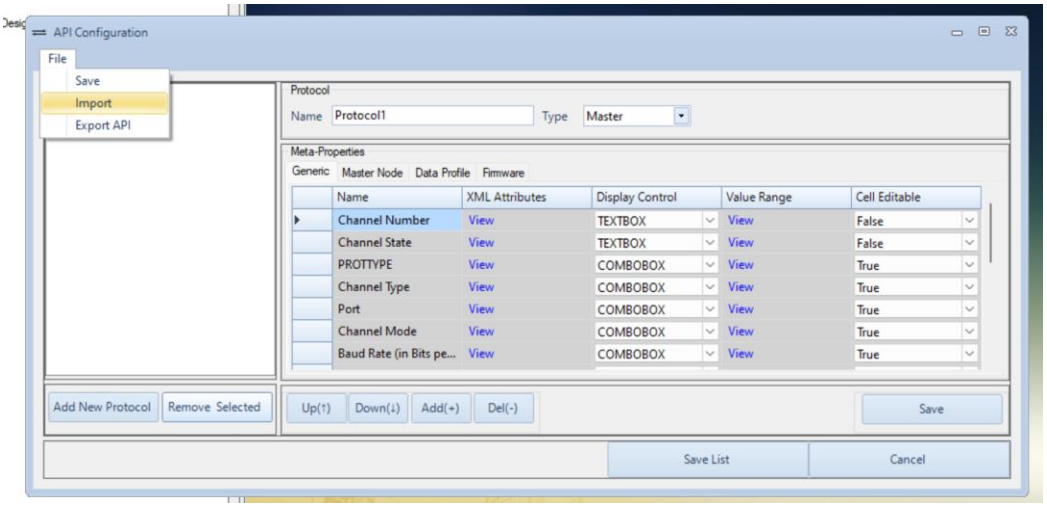
3.1 Import API

This step is a prerequisite for protocol configuration.

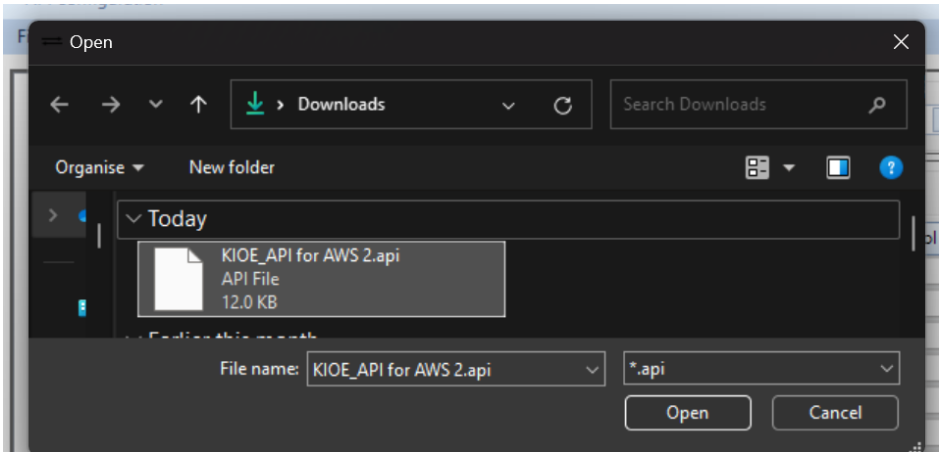
1. Open SCT tool and go to tools and select API configuration.



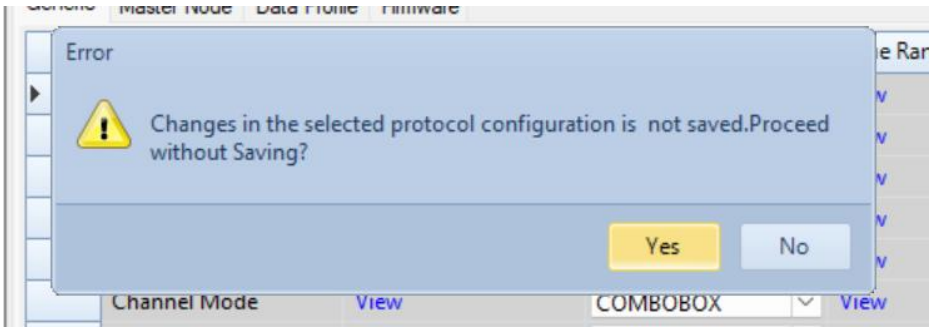
2. Click on file and select import.



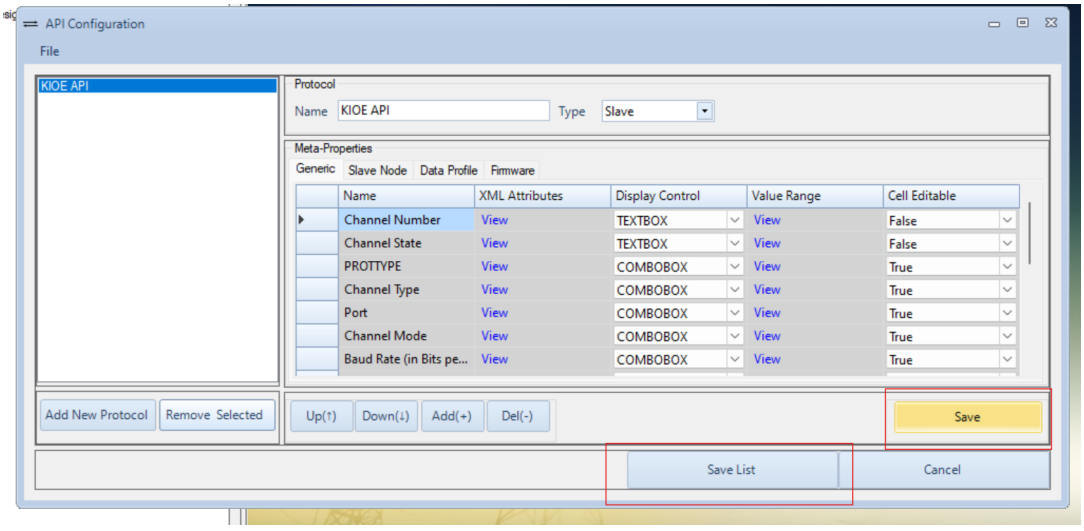
3. Select ‘KIOE_API for AWS’ file which was shared and click open.



4. Click on proceed without saving.



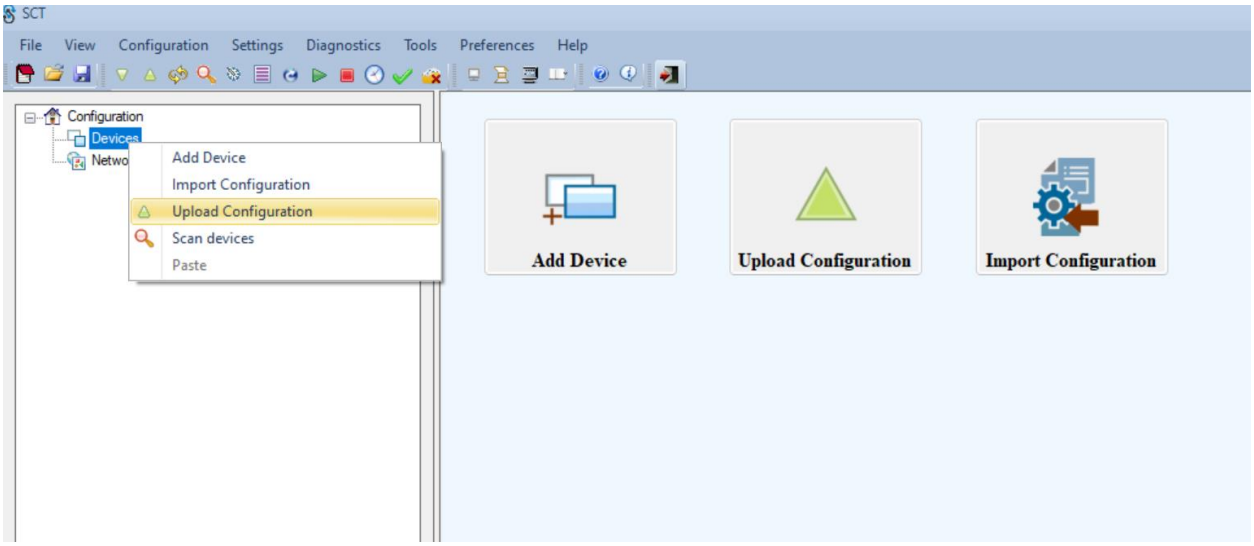
5. Click on save followed by save list.



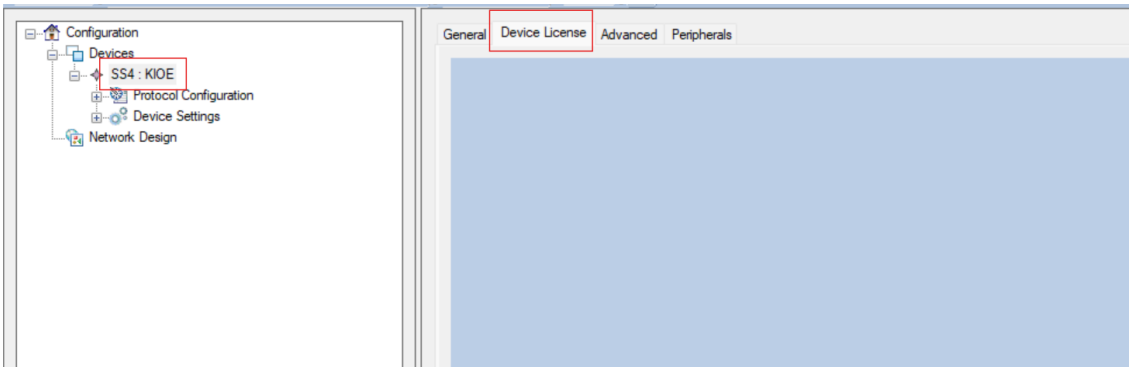
3.2 License Installation

Once license file and SCT tool are received, they can be installed in KIOE host. Follow the steps below to install the license.

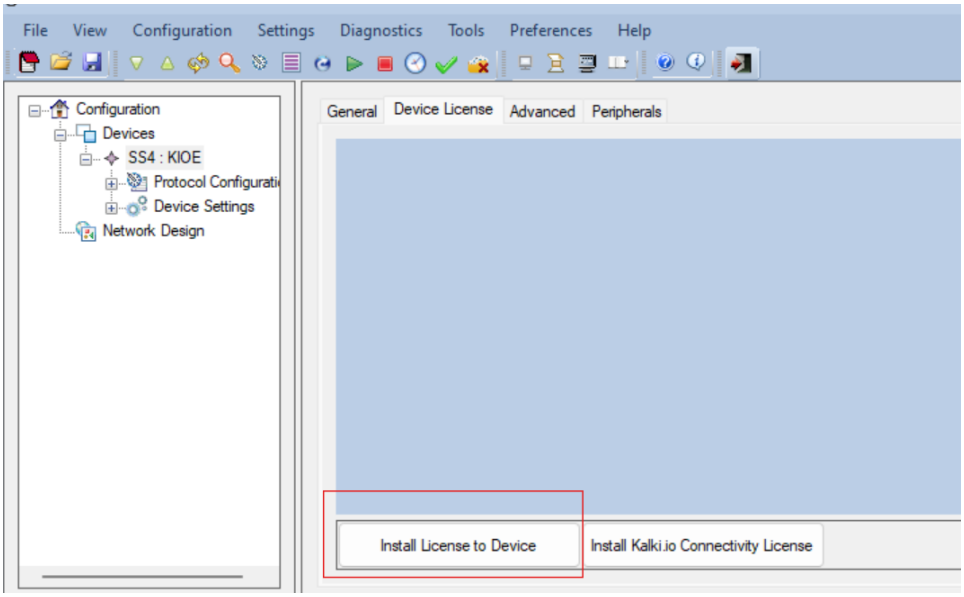
1. Launch SCT from the start menu, right click on Devices and click upload configuration as shown below.



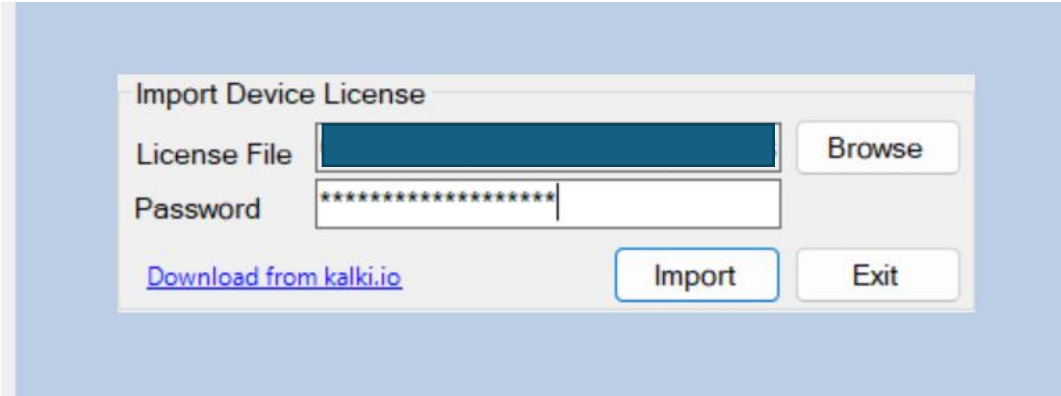
2. Click on SS4:KIOE from the left panel and select Device license in right panel.



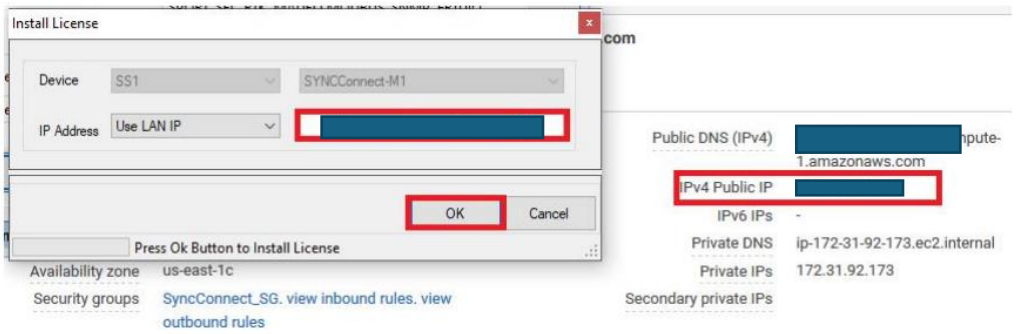
3. Click on Install License to device option as shown below.



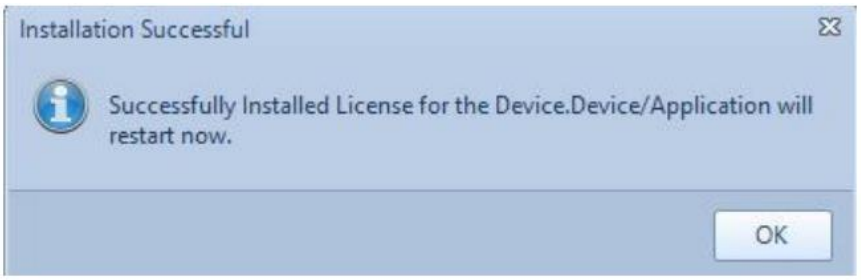
4. Browse the license file provided and enter the password given and click import.



You will see License information. Click on OK to move to the next step at which the following prompt will appear for IP address of KIOE. Provide IP address of KIOE host.



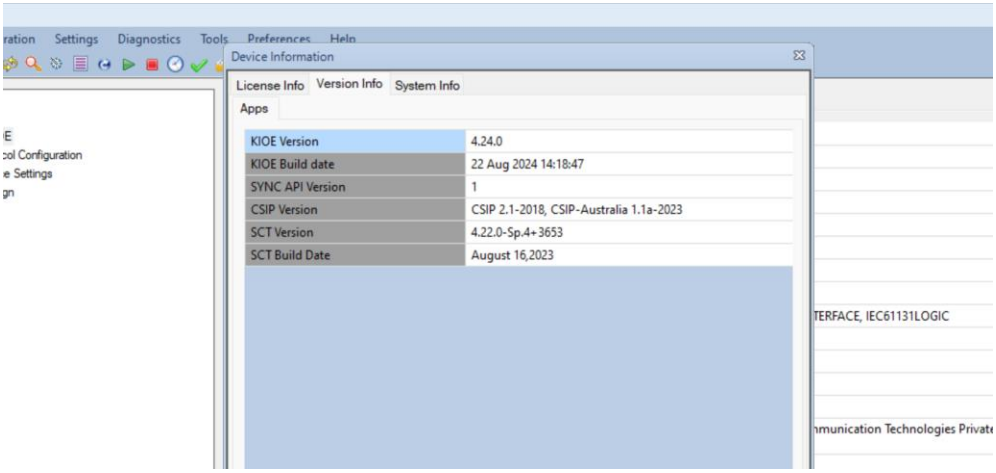
You should get the following prompt. KIOE will restart after this operation. Wait for a minute and move to the next step.



This step verifies **KIOE version**. In SCT toolbar go to **Settings -> Version Information**. Provide IP for KIOE host and press OK.

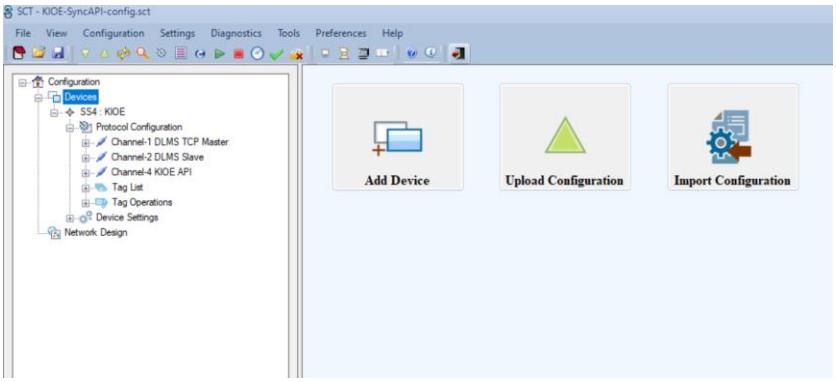
If you get the version for software in Version Info tab, this confirms successful installation of KIOE Licenses.

You can get more information about licensed protocols in **the License Info** tab.

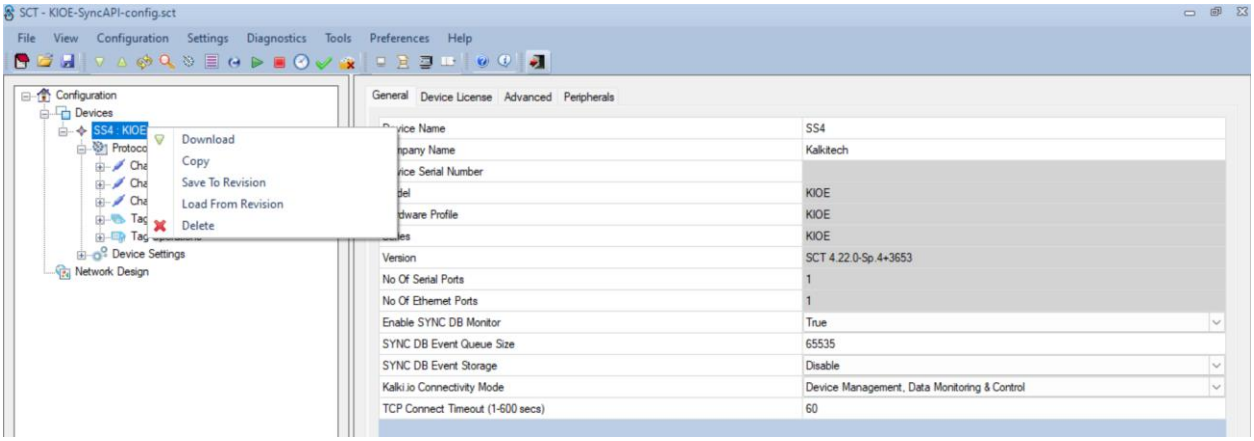


3.3 Protocol configuration

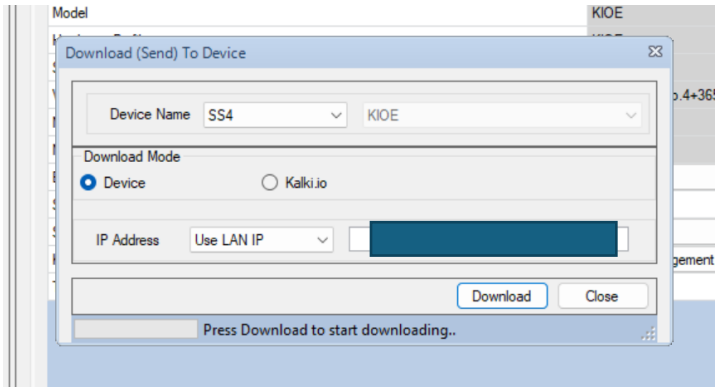
1. Open the shared file named ‘KIOE-SyncAPI-config.sct’.
2. You will see a window like shown below



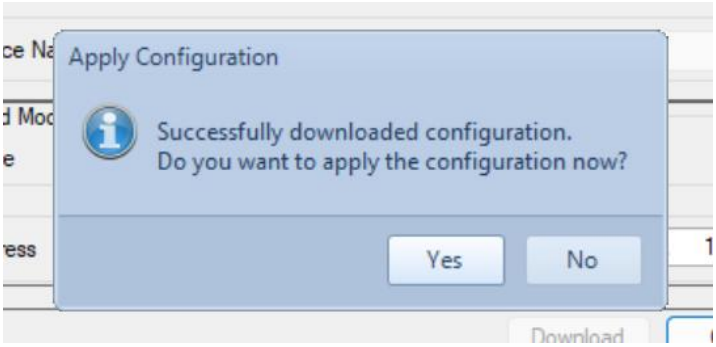
3. Right click on SS4:KIOE and click download



4. Enter the target IP of the core device available from EC2 console details.



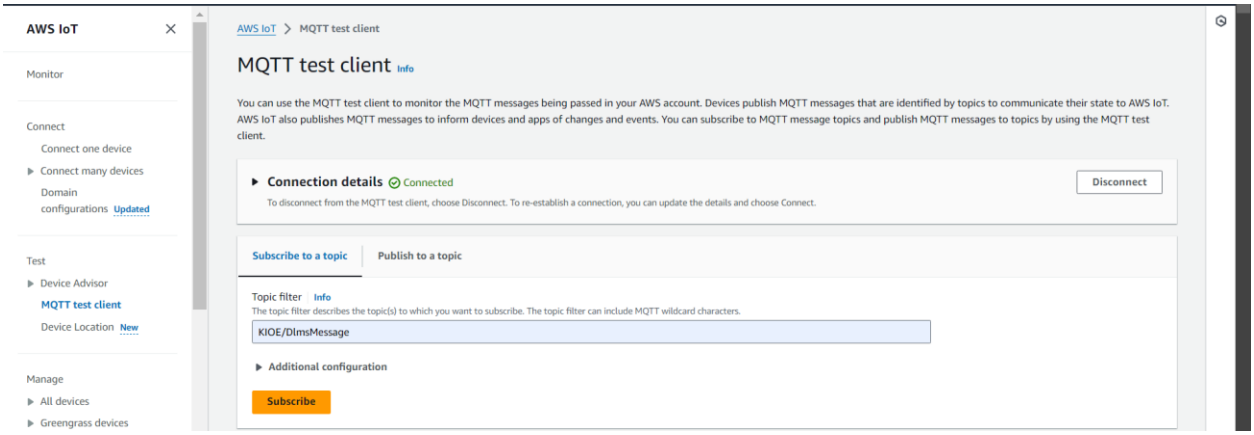
5. Click yes to apply configuration.



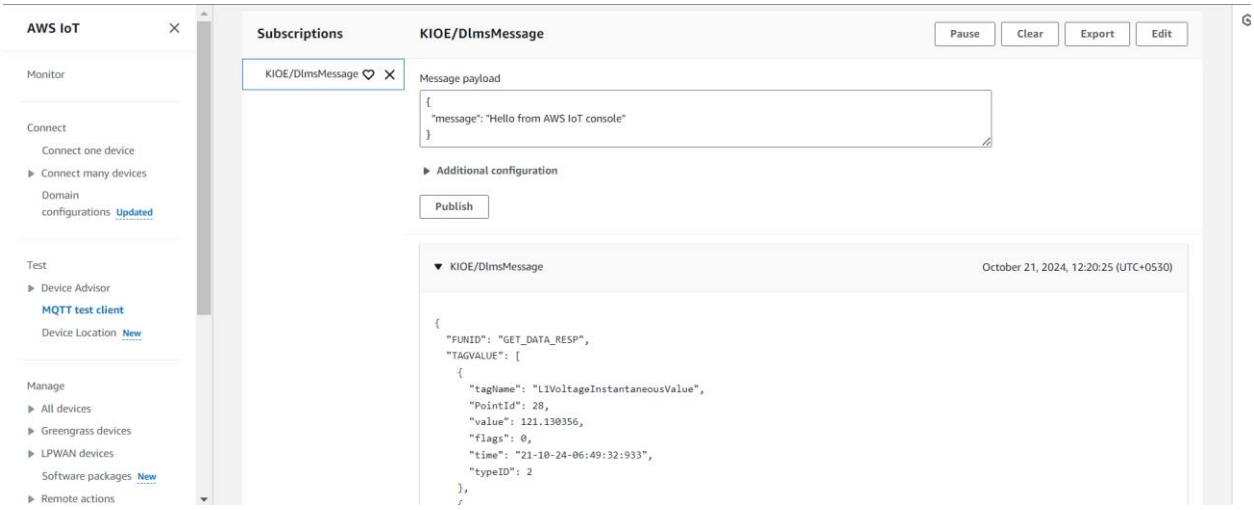
4. Verification of Installation

After successfully following the steps mentioned, you will be able to see the data generated by meter simulator in AWS IoT Cloud.

1. Go to AWS IoT Console
2. Select MQTT Test Client, select subscribe to topic and enter the topic 'KIOE/DlmsMessage'



3. You will receive messages depending on the polling time given during deployment of Greengrass components. By default, the polling time is 10 seconds.



4. You can simulate a surge in power by doing the following steps. This will give a surge in power instantaneous value and an email notification that the power has exceeded the value.
    a. In MQTT Test client, click Publish to topic and enter the topic name 'GGCore/SetUp'
    b. Give the following JSON as message payload

```
{
 "SURGE_BOOLEAN": "TRUE",
 "DISCONNECT_AT_VALUE": "10",
 "TOPIC": "KIOE/MeterDisconnect",
 "IOT_CORE_TOPIC": "KIOE/DlmsMessage",
 "QUEUE_URL": "",
 "POLLING_INTERVAL_SECONDS": "10",
 "TRIGGER_POWERINSTANTANEOUSVALUE": "8",
 "SNSTOPICARN": "arn:aws:sns:us-east-1:123456789:Trigger",
 "SNS_TRIGGER_INTERVAL_SECONDS": "150.0"
}
```


NOTE: Make sure you send the whole JSON even when you want to modify only one parameter.

Parameters:

- SURGE_BOOLEAN: Boolean value used to trigger a surge. TRUE triggers a surge and FALSE reverts the surge.
- DISCONNECT_AT_VALUE: Threshold value of power at which the current should be disconnected.
- TOPIC: Make sure the value is set to KIOE/MeterDisconnect as it is used for internal working of Greengrass components.
- IOT_CORE_TOPIC: Topic to which the message needs to be sent so that it will be visible in IoT Core MQTT Test client.
- QUEUE_URL: Set it as empty as it is out of scope for the use case.
- POLLING_INTERVAL_SECONDS: Polling time of data in seconds. Make sure you set it to 10 or more.
- TRIGGER_POWERINSTANTANEOUSVALUE: Threshold value of power, which if exceeded, an email notifying about the surge in power would be sent.
- SNSTOPICARN: Make sure you give the SNS trigger ARN which was created using the prerequisite step.
- SNS_TRIGGER_INTERVAL_SECONDS: Time interval in seconds in which the SNS alert will be sent.

5. After publishing the message with SURGE_BOOLEAN value as TRUE, the power value would surge. A screenshot of MQTT test client UI with surged power value is shown here.

```
    {
      "tagName": "L1CurrentInstantaneousValue",
      "PointId": 20,
      "value": 4.445415,
      "flags": 0,
      "time": "10-12-24-04:29:18:627",
      "typeID": 2
    },
    {
      "tagName": "PowerInstantaneousValue",
      "PointId": 31,
      "value": 12.015869,
      "flags": 0,
      "time": "10-12-24-04:29:19:674",
      "typeID": 2
    },
    {
      "tagName": "InstantaneousFrequency",
      "PointId": 32,
      "value": 59.564987,
      "flags": 0,
      "time": "10-12-24-04:29:19:674",
      "typeID": 2
    }
  ],
  "AWS_IOT_THING_NAME": "Thing"
}
```


If it crosses the value mentioned in DISCONNECT_AT_VALUE, the system will disconnect and thereby resulting in L1CurrentInstaneousValue and PowerInstaneousValue of 0. (Refer screenshot).

Note: If you want to see a demo of the surge and disconnect mentioned here, the SURGE_BOOLEAN value should be TRUE, and we suggest keeping DISCONNECT_AT_VALUE & POLLING_INTERVAL_SECONDS to reasonable values like 10 & 10 seconds respectively. Note that disconnect due to the surge is almost immediate. Thus, the surge response would only be shown fleetingly since the MQTT test client UI refreshes continuously.

Once disconnected, to reconnect and revert to the normal working, send SURGE_BOOLEAN value as FALSE. A screenshot of mqtt test client UI with normal values for power and current is shown here.


```
    },
    {
      "tagName": "L1CurrentInstantaneousValue",
      "PointId": 20,
      "value": 4.757845,
      "flags": 0,
      "time": "10-12-24-04:31:18:863",
      "typeID": 2
    },
    {
      "tagName": "PowerInstantaneousValue",
      "PointId": 31,
      "value": 2.707062,
      "flags": 0,
      "time": "10-12-24-04:31:20:202",
      "typeID": 2
    },
    {
      "tagName": "InstantaneousFrequency",
      "PointId": 32,
      "value": 60.065071,
      "flags": 0,
      "time": "10-12-24-04:31:20:202",
      "typeID": 2
    }
  ],
  "AWS_IOT_THING_NAME": "Thing"
```

Appendix A

KIOE linearly consumes more resources as the number of tags in operation increases. Following Sections provides Compute resources required for tag counts.

A.1 Tag count to EC2 Instance type

The following table gives tag count to instance type. It is recommended to use t2 instances for most workloads. Same t3 instances can be used for workloads requiring high network throughput.

Serial Number	Tag Count	Instance Type	Instance Type (with ICCP or IEC 61850)
1	0 - 500	t2.micro	t2.small
2	500-1000	t2.small	t2.medium
3	1000-2000	t2.medium	t2.large
4	2000-5000	t2.large	t2.xlarge
5	5000 - 10000	t2.xlarge	t2.2xlarge

A.2 Tag count to ECS Memory and Process and Units

Following table gives CPU and RAM units required in ECS containers

Serial Number	Tag Count	CPU(Units)	RAM(MB)	ICCP and IEC 61850 tags	
				CPU(Units)	RAM(MB)
1	0 - 500	512	512	1024	1024
2	500-1000	512	1536	1024	3048
3	1000-2000	1024	3048	2048	5120
4	2000-5000	2048	5120	3048	7168
5	5000 - 10000	3048	8192	4096	8192