

# Kalki IO Edge for AMI 2.0 using AWS Greengrass

Configuring AMI 2.0 setup

[www.kalkitech.com](http://www.kalkitech.com)

## **Copyright Notice**

© Kalki Communication Technologies Pvt. Ltd. All Rights reserved.

This user manual is a publication of Kalki Communication Technologies Pvt. Ltd. and it is intended for its customers use only. Content of this manual is copyrighted and reproduction in whole or in part, for use other than in support of Kalkitech equipment/software, is prohibited without the specific written permission of Kalkitech.

## **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](mailto:sales@kalkitech.com)

## **Technical Support Contact Information**

Online Technical Enquiry Support: [www.kalkitech.com/support](http://www.kalkitech.com/support)  
E-mail Support: [support@kalkitech.com](mailto:support@kalkitech.com)

# Table of Contents

Table of Contents.....	3
Terms and abbreviations.....	4
1. Introduction .....	5
1.1 Pre-requisites: License and files Request .....	5
2. KIOE AMI 2.0 Setup machine image.....	6
2.1 Kalki IO Edge for AMI 2.0 using AWS Greengrass AMI subscription .....	6
2.2 AMI 2.0 Provisioning through website .....	8
2.3 AMI 2.0 Machine Image Provisioning from EC2 console .....	11
2.4 Validating AMI 2.0 instance .....	13
2.5 Cloudformation script to install AMI 2.0 setup .....	14
2.5.1 Prerequisite Script .....	14
2.5.2 Script (1) To create Greengrass core device .....	17
2.5.3 Script (2) to create and attach bucket policy .....	18
2.5.4 Script (3a), (3b) (3c) - Used to create components for Greengrass.....	19
Script (3a) parameters.....	19
Script (3b) parameters .....	19
Script (3c) parameters.....	19
2.5.5 Script (4) Script for deploying the components .....	20
3. Kalki IO Edge Licensing.....	21
3.1 Import API.....	21
3.2 License Installation .....	22
3.3 Protocol configuration .....	25
4. Verification of Installation .....	27
Appendix A .....	31
A.1 Tag count to EC2 Instance type.....	31
A.2 Tag count to ECS Memory and Process and Units.....	31

## 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](mailto: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.



### Kalki IO Edge for AMI 2.0 using AWS Greengrass

By: [ASE/ Kalkitech](#) Latest Version: 4.24

Kalki.IO Edge (KIOE) Software is an edge software that can be used along with AWS Greengrass for implementing AMI 2.0 meter. Utilities, OEMs and AMI vendors can utilize the software for

[Show more](#)

Linux/Unix

**BYOL** **Free Tier**

[Continue to Subscribe](#)  
[Save to List](#)  
Typical Total Price  
**\$0.042/hr**  
Total pricing per instance for services hosted on t3.medium in US East (N. Virginia). [View Details](#)

[Overview](#) [Pricing](#) [Usage](#) [Support](#) [Reviews](#)

### Product Overview


Kalki.IO Edge (KIOE) used in Industries, utilities to monitor, control and translate data from field devices, sensors and included metering protocols. AWS and Kalkitech collaborated to build an AMI 2.0 PoC showcasing next-generation AMI solutions. Kalkitech developed the AMI 2.0 PoC utilizing the capabilities of AWS IoT Core, Greengrass Services and leveraging Kalki.IO Edge for the DLMS metering protocol capability.

Version	4.24
By	<a href="#">ASE/ Kalkitech</a>
Categories	<a href="#">Device Connectivity</a> <a href="#">Device Management</a>

#### Highlights

- KIOE is tested with 5000+ type of OT & IT devices/sensors/equipment/software
- AWS IoT Greengrass and IoT Core are key components in achieving AMI 2.0 Edge computing capability.
- Kalkitech's widely recognized and proven DLMS source code library is used part of KIOE.

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.



### Kalki IO Edge for AMI 2.0 using AWS Greengrass

[Continue to Configuration](#)  
You must first review and accept terms.

### Subscribe to this software

To create a subscription, review the pricing information and accept the terms for this software.

#### Terms and Conditions

ASE/ Kalkitech Offer 2024-09-24

By subscribing to this software, you agree to the pricing terms and the seller's End User License Agreement (EULA). You also agree and acknowledge that AWS may, on your behalf, share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the AWS Privacy Notice. AWS will issue invoices and collect payments from you on behalf of the seller through your AWS account. Your use of AWS services is subject to the AWS Customer Agreement or other agreement with AWS governing your use of such services. If you are receiving a private offer from a channel partner, you may click [here](#) for CPPO transaction) or [here](#) (for SPPO transaction) for more information on the channel partner.

[Accept Terms](#)

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.**

[< Product Detail](#)   [Subscribe](#)   [Configure](#)

## Configure this software

Choose a fulfillment option and software version to launch this software.

Fulfillment option

64-bit (x86) Amazon Machine Image (AMI) ▾

Software version

4.24 (Sep 25, 2024) ▾

Region

US East (N. Virginia) ▾

Use of Local Zones or WaveLength infrastructure deployment may alter your final pricing.

Pricing information

This is an estimate of typical software and infrastructure costs based on your configuration. Your actual charges for each statement period may differ from this estimate.

Software Pricing

Kalki IO Edge for AMI 2.0 using AWS Greengrass

**BYOL**

running on t3.medium

\$0 /hr

Infrastructure Pricing

EC2: 1 \* t3.medium

Monthly Estimate: \$30.00/month

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.

The screenshot shows two sections of the AWS console. The first section, 'Choose Action', has a dropdown menu set to 'Launch from Website' and a note: 'Choose this action to launch from this website'. The second section, 'EC2 Instance Type', has a dropdown menu set to 't3.medium'. To the right of this dropdown, the instance specifications are listed: 'Memory: 4 GiB', 'CPU: 2 virtual cores', 'Storage: EBS Only', and 'Network Performance: Up to 5 Gigabit Ethernet'.

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.

The screenshot shows the 'VPC Settings' section. It includes a note: '\* indicates a default vpc'. Below this is a dropdown menu showing 'vpc-02115a78' with a refresh icon to its right. At the bottom of the section is a link: 'Create a VPC in EC2' with an external link icon.

The screenshot shows the 'Subnet Settings' section. It includes a dropdown menu with 'subnet-4ee0da60 (us-east-1c)' selected. The dropdown menu is open, showing a list of subnets: 'Select a subnet', 'subnet-4ee0da60 (us-east-1c)', 'subnet-8107c8cc (us-east-1d)', 'subnet-2020d12e (us-east-1f)', 'subnet-7b3d021c (us-east-1b)', 'subnet-90ecd4cc (us-east-1a)', and 'subnet-f3afecdd (us-east-1e)'. To the right of the dropdown is a refresh icon and the text 'IPv4 CIDR block: 172.31.80.0/20'. Below the dropdown, there is a link: 'Learn more'.


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**.

**Security Group Settings**

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. You can create a new security group based on seller-recommended settings or choose one of your existing groups. [Learn more](#)

AMI2.0-SG 

[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.

**Create new based on seller settings**

A new security group will be generated by AWS Marketplace. It is based on recommended settings for Kalki IO Edge for AMI 2.0 using AWS Greengrass version 4.24.

**Name your security Group**

**Description**

Connection Method	Protocol	Port Range	Source (IP or Group)
SSH	tcp	22	<input type="text"/> 
	tcp	1081	<input type="text"/> 



Rules with source of 0.0.0.0/0 allows all IP addresses to access your instance. We recommend limiting access to only known IP addresses.


[Cancel](#) [Save](#)

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.

**Key Pair Settings**

To ensure that no other person has access to your software, the software installs on an EC2 instance with an EC2 key pair that you created.

Select a key pair  

[Create a key pair in EC2](#) 

(Ensure you are in the region you wish to launch your software)

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.

The screenshot shows the 'Configuration details' section of the AWS EC2 console. It lists the fulfillment option as '64-bit (x86) Amazon Machine Image (AMI)' with the specific image 'Kalki IO Edge for AMI 2.0 using AWS Greengrass running on t3.medium'. The software version is '4.24' and the region is 'US East (N. Virginia)'. There is a 'Usage instructions' button. Below this is the 'Choose Action' section with a dropdown menu set to 'Launch through EC2' and a 'Launch' button. A note states: 'Choose this action to launch your configuration through the Amazon EC2 console.'

Configuration details	
Fulfillment option	64-bit (x86) Amazon Machine Image (AMI) Kalki IO Edge for AMI 2.0 using AWS Greengrass <i>running on t3.medium</i>
Software version	4.24
Region	US East (N. Virginia)

[Usage instructions](#)

**Choose Action**

Launch through EC2 ▼ Choose this action to launch your configuration through the Amazon EC2 console.

[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.

The screenshot shows the 'Instance type' selection screen. It features a dropdown menu with 't3.medium' selected, showing details: 'Family: t3', '2 vCPU', '4 GiB Memory', and 'Current generation: true'. There is an 'All generations' toggle switch and a 'Compare instance types' link. A recommendation note at the bottom states: 'The AMI vendor recommends using a t3.medium instance (or larger) for the best experience with this product.'

**▼ Instance type** [Info](#) | [Get advice](#)

Instance type

t3.medium ▼  
Family: t3 2 vCPU 4 GiB Memory Current generation: true

☒ All generations [Compare instance types](#)

The AMI vendor recommends using a t3.medium instance (or larger) for the best experience with this product.

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.

Inbound rules

Type

Info

Protocol

Info

Port range

Info

Source

Info

Description - optional

Info

Custom TCP	TCP	22	Anywh...	<div><div></div><div></div></div>		Delete
Custom TCP	TCP	1081	Anywh...	<div><div></div><div></div></div>		Delete

Add rule

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

▼ Configure storage

Info

Advanced

1x 100 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

×

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

↻

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Edit

► Advanced details

Info

Number of instances

Info

1

Software Image (AMI)

AMI2.0 Marketplace offering-pr...read more

ami-0ef271a748a89ec53

Virtual server type (instance type)

t3.medium

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 100 GiB

Free tier: In your first year includes 750 hours of t2.micro

×

Cancel

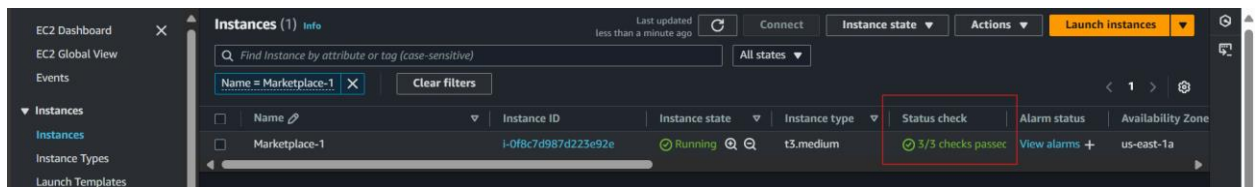
Launch instance

Preview code

## 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.

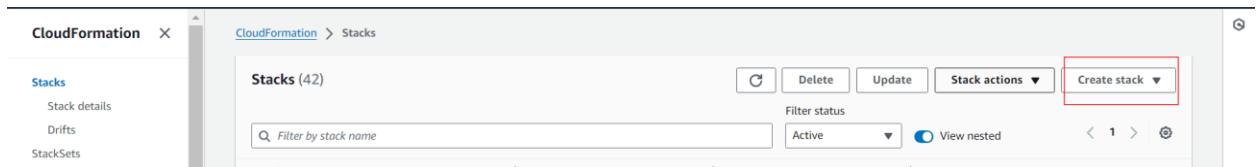
Note: Make sure that you are running the script in one of the following regions:

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

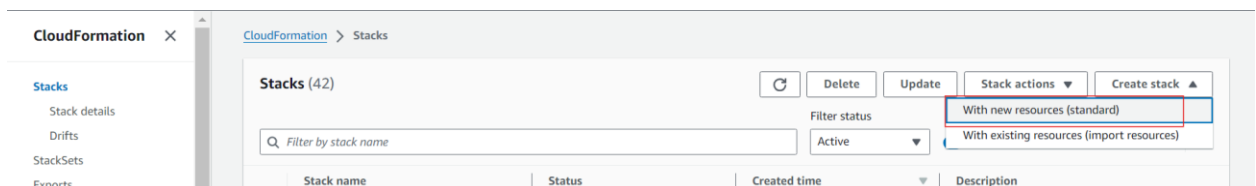
### 2.5.1 Prerequisite Script

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.

CloudFormation X

Stacks StackSets Exports

Infrastructure Composer IaC generator

Registry Public extensions Activated extensions Publisher

Spotlight

Feedback

CloudFormation > Stacks > Create stack

Step 1 Create stack

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review and create

### Create stack

**Prerequisite - Prepare template**

You can also create a template by scanning your existing resources in the [IaC generator](#).

**Prepare template**

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Choose an existing template  
Upload or choose an existing template.

☐ Use a sample template  
Choose from our sample template library.

☐ Build from Infrastructure Composer  
Create a template using a visual builder.

**Specify template** info

A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**

Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL  
Provide an Amazon S3 URL to your template.

☒ Upload a template file  
Upload your template directly to the console.

☐ Sync from Git  
Sync a template from your Git repository.

**Upload a template file**

JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

[View in Infrastructure Composer](#)

4. After selecting the file click on next.

**Specify template** info

A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**

Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL  
Provide an Amazon S3 URL to your template.

☒ Upload a template file  
Upload your template directly to the console.

☐ Sync from Git  
Sync a template from your Git repository.

**Upload a template file**

Prerequisite\_marketplace.yaml

JSON or YAML formatted file

S3 URL: [https://s3.us-east-1.amazonaws.com/cf-templates-j9h4uylvvmx5-us-east-1/2024-10-17T092352.197Zrfn-Prerequisite\\_marketplace.yaml](https://s3.us-east-1.amazonaws.com/cf-templates-j9h4uylvvmx5-us-east-1/2024-10-17T092352.197Zrfn-Prerequisite_marketplace.yaml)

[View in Infrastructure Composer](#)

[Cancel](#) [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.

CloudFormation > Stacks > Create stack

Step 1  
[Create stack](#)

Step 2  
**Specify stack details**

Step 3  
[Configure stack options](#)

Step 4  
[Review and create](#)

### Specify stack details

**Provide a stack name**

Stack name

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 0/128.

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**E-Mail**

Enter the email address in which notification is to be sent.

**SnSName**

Enter the Unique SnS Topic Name

Cancel Previous **Next**

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

CloudFormation > Stacks > Create stack

Step 1  
[Create stack](#)

Step 2  
[Specify stack details](#)

Step 3  
**Configure stack options**

Step 4  
[Review and create](#)

### Configure stack options

**Tags - optional**

Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organizing, identifying, and categorizing those resources. You can add up to 50 unique tags for each stack.

No tags associated with the stack.

You can add 50 more tag(s)

**Permissions - optional**

Specify an existing AWS Identity and Access Management (IAM) service role that CloudFormation can assume.

**IAM role - optional**

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

**⚠ AWS CloudFormation will use this role for all stack operations. Other users that have permissions to operate on this stack will be able to use this role, even if they don't have permission to pass it. Ensure that this role grants least privilege.**

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

**Capabilities**

**i The following resource(s) require capabilities: [AWS::IAM::Role]**

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.

Cancel Previous **Next**

9. Review the options selected and select submit.



### Stack creation options

Timeout

-

Termination protection

Deactivated

### Quick-create link

Use quick-create links to get stacks up and running quickly from the AWS CloudFormation console with the same basic configuration as this stack. Copy the URL on the link to share. [Learn more](#)

Open quick-create link

Create change set

Cancel

Previous

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.

CloudFormation

Stacks

StackSets

Exports

Infrastructure Composer

laC generator

Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

CloudFormation

Stacks

Create stack

Step 1

Create stack

Step 2

Specify stack details

Step 3

Configure stack options

Step 4

Review and create

### Create stack

#### Prerequisite - Prepare template

You can also create a template by scanning your existing resources in the [laC generator](#)

##### Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Choose an existing template
 

Upload or choose an existing template.

☐ Use a sample template
 

Choose from our sample template library.

☐ Build from Infrastructure Composer
 

Create a template using a visual builder.

#### Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

##### Template source

Selecting a template generates an Amazon S3 URL, where it will be stored.

☐ Amazon S3 URL
 

Provide an Amazon S3 URL to your template.

☒ Upload a template file
 

Upload your template directly to the console.

☐ Sync from Git
 

Sync a template from your Git repository.

##### Upload a template file

Choose file
 

JSON or YAML formatted file

S3 URL: Will be generated when template file is uploaded

View in Infrastructure Composer

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.

Step 1

Create stack

Step 2

Specify stack details

Step 3

Configure stack options

Step 4

Review and create

Specify stack details

Provide a stack name

Stack name

Ec2CreationStack

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 16/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

GreenGrassRegion

AWS region of the greengrass core & group

us-east-1

KeyPair

Create or use existing key pair for the EC2 instance (Make sure the key pair exist in the same region).(Eg.-Dev2Key) Do not give the extension such as .pem/.ppk

samplekey

ThingGroupName

MarketplaceThing-1

ThingName

MarketplaceGroup

Cancel

Previous

Next

- Continue till reviewing the inputs and click submit.

### 2.5.3 Script (2) to create and attach bucket policy

- Go to CloudFormation console, create a stack with new resources, upload the (2) script and click next.
- 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.

Amazon S3

Buckets

greengrass-poc-kalkitech

code/

code/

Objects

Properties

Folder overview

<div>AWS Region</div> <div>US East (N. Virginia) us-east-1</div>	<div>S3 URI</div> <div>s3://greengrass-poc-kalkitech/code/</div>	<div>Amazon Resource Name (ARN)</div> <div>arn:aws:s3::greengrass-poc-kalkitech/code/</div>
--	--	---

Copy S3 URI

CloudFormation

Stacks

StackSets

Exports

Infrastructure Composer

laC generator

Registry

Public extensions

Activated extensions

Publisher

Spotlight

Feedback

CloudFormation

Stacks

Create stack

Specify stack details

Provide a stack name

Stack name

Enter a stack name

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 0/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

S3CodeLocationARN

S3CodeLocationArn for the bucket folder ARN. example:arn:aws:s3::greengrass-poc-kalkitech-us-west2/code/

Enter String

UniquePolicyName

UniquePolicyName. Ensure unique name.

Enter String

Cancel

Previous

Next

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

CloudFormation > Stacks > Create stack

Step 1  
[Create stack](#)

Step 2  
**Specify stack details**

Step 3  
[Configure stack options](#)

Step 4  
[Review and create](#)

### Specify stack details

**Provide a stack name**

Stack name  
  
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 0/128.

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

GreenGrassPythonS3Location  
S3 location of GreenGrassConfig.py (Eg:s3://greengrass-poc-kalitech-v2-us-east-1/code/GreenGrassConfig.py)

Cancel Previous Next

### Script (3b) parameters

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

CloudFormation > Stacks > Create stack

Step 1  
[Create stack](#)

Step 2  
**Specify stack details**

Step 3  
[Configure stack options](#)

Step 4  
[Review and create](#)

### Specify stack details

**Provide a stack name**

Stack name  
  
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 11/128.

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

GreenGrassJarS3Location  
S3 location of greengrasscoremeter.jar (Eg:s3://greengrass-poc-kalitech-v2-us-east-1/code/greengrasscoremeter.jar)

Cancel Previous Next

### Script (3c) parameters

1. S3 URI of the file mqttsnsscript.py

CloudFormation > Stacks > Create stack

Step 1  
[Create stack](#)

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review and create

### Specify stack details

**Provide a stack name**

Stack name  
component-3  
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 11/128.

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

GreenGrassPythonS3Location  
S3 location of mqttssnscript.py (Eg:s3://greengrass-poc-kalkitech-v2-us-east-1/code/mqttssnscript.py)  
s3://greengrass-poc-kalkitech-v2-us-east-1/code/mqttssnscript.py

Cancel Previous **Next**

## 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.

**Provide a stack name**

Stack name  
Deployment  
Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 10/128.

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

DisconnectValue  
Value between 1-12 (can be float). Default is 10.  
10

PollingInterval  
Polling interval seconds (float value). Default is 15  
15

SNSPowerTriggerValue  
Float value. Default 10.  
10

SNSTopicARN  
ARN of SNS topic (Eg- amawsnsus-east-1:123456789:trigger )  
amawsnsus-east-1:123456789:trigger

SNSTriggerIntervalSeconds  
Seconds interval for SNS message (can be float). Default 150 secs.  
150

SurgeBoolean  
SURGE, BOOLEAN\_DISCONNECT Value  
FALSE

ThingGroupARN  
The deployment's target thing group or thing ARN (Eg- amawsiotus-east-1:081543507135:thinggroup/DevGroupPhase2)  
amawsiotus-east-1:123456789:thinggroup/DevGroupPhase2

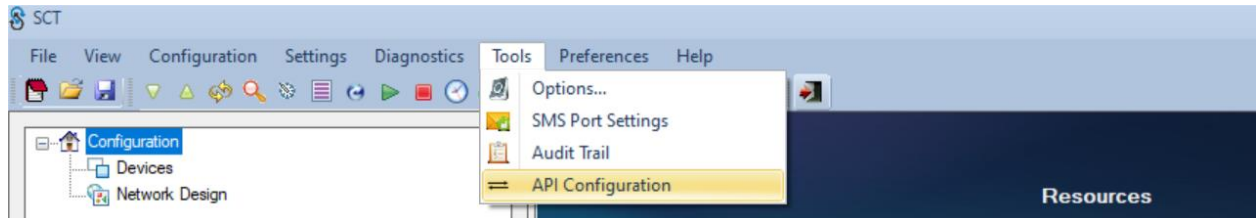
YourDeploymentName  
Deployment Name. (Eg- Deployment for EC2 in US East 1). Can reuse the same name for multiple deployments to the same deployment target.  
MarketplaceDeployment

### 3. Kalki IO Edge Licensing

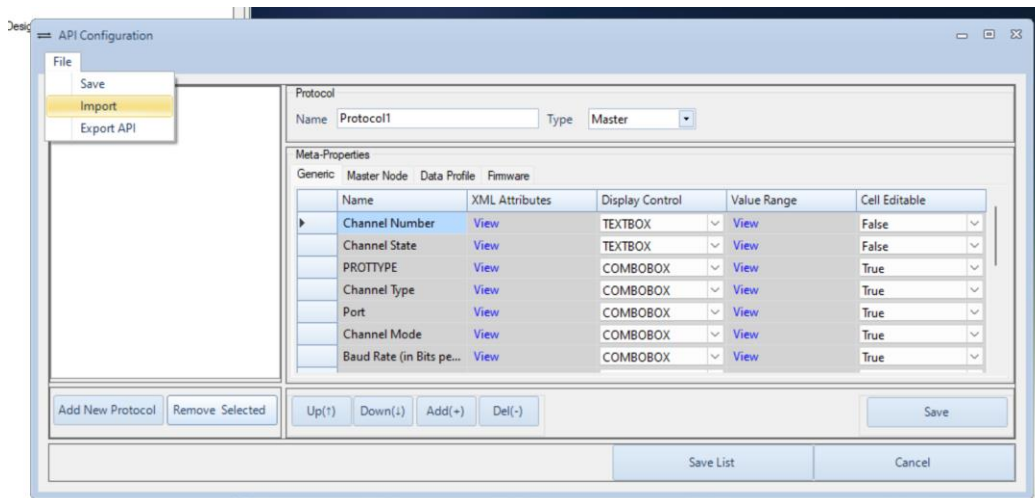
#### 3.1 Import API

This step is a prerequisite for protocol configuration.

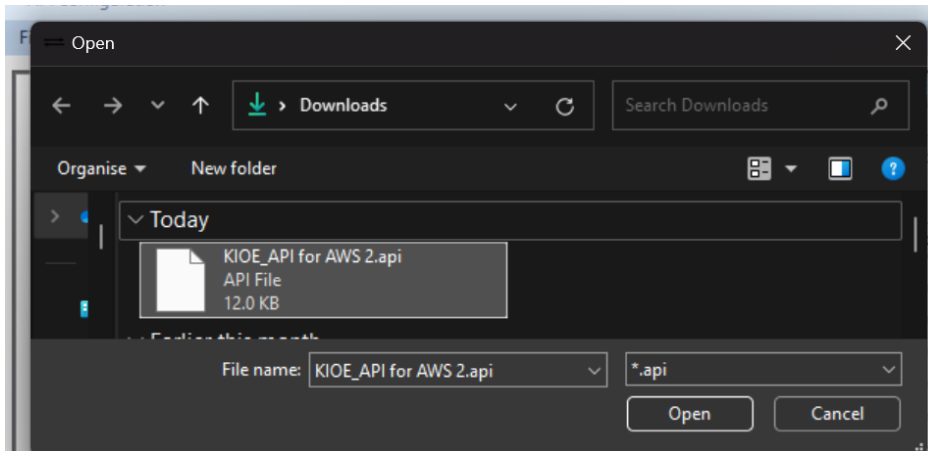
1. Open SCT tool and go to tools and select API configuration.



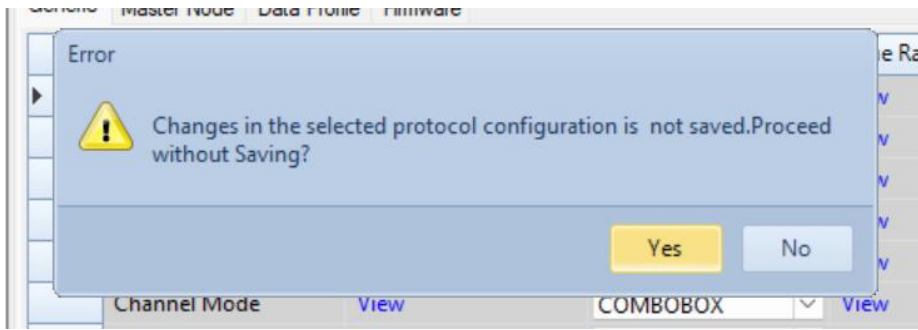
2. Click on file and select import.



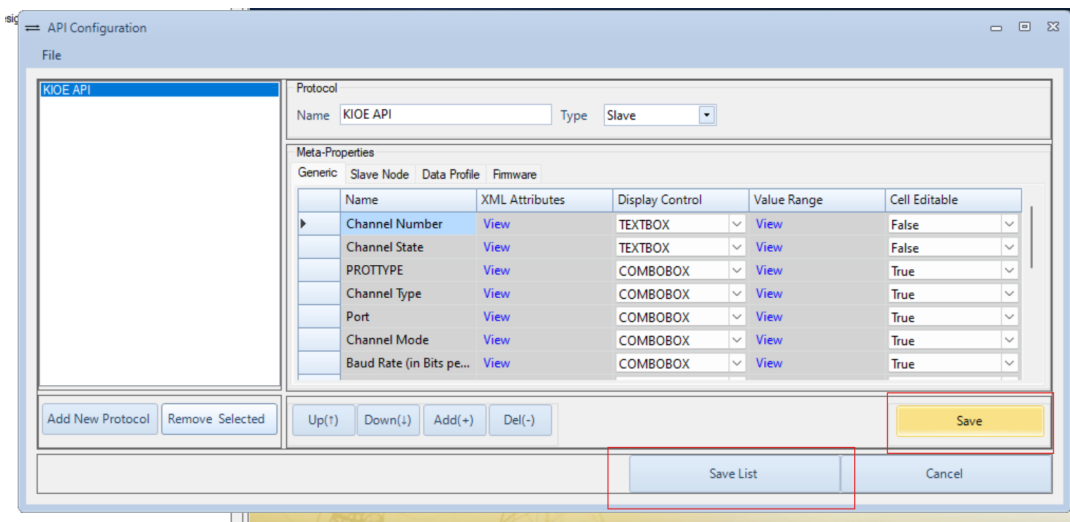
3. Select 'KIOE\_API for AWS' file which was shared and click open.



4. Click on proceed without saving.



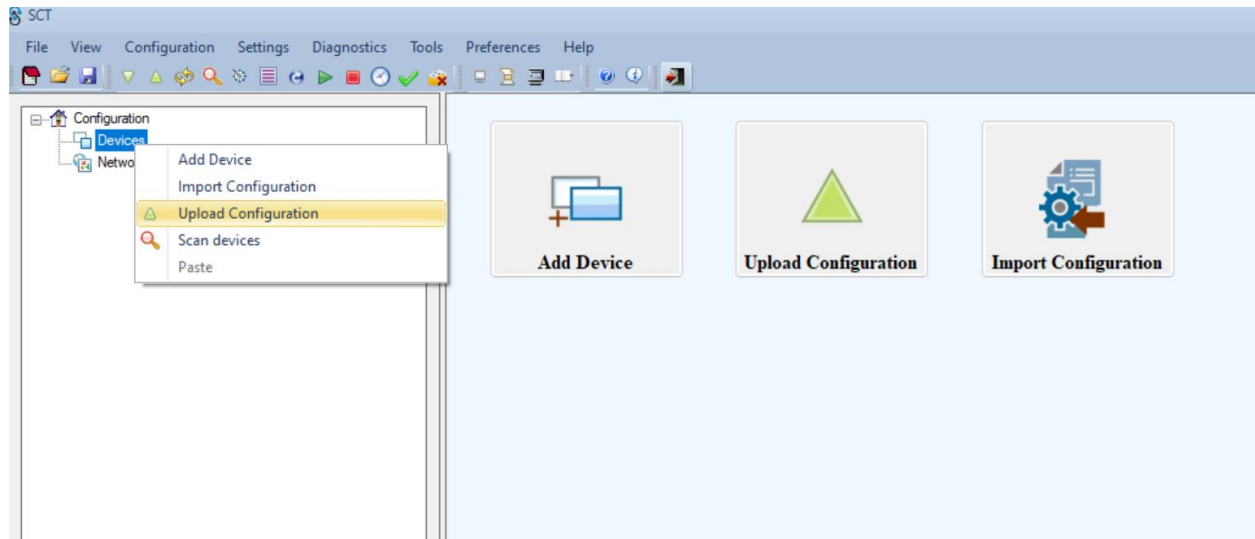
5. Click on save followed by save list.



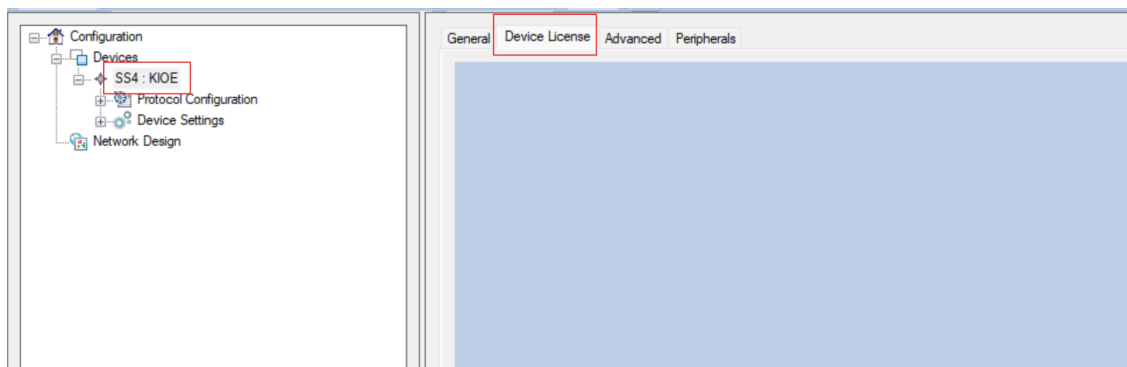
### 3.2 License Installation

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

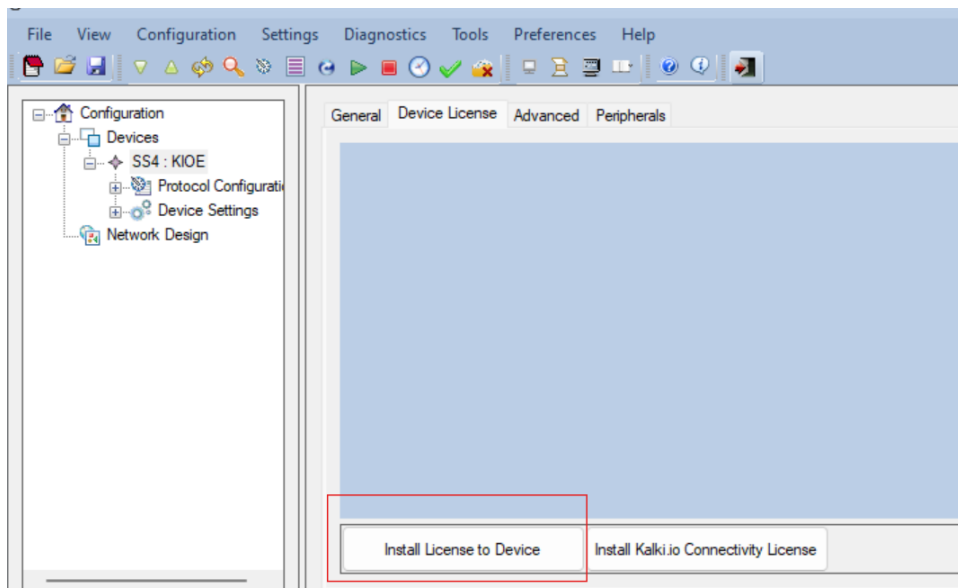
1. Launch SCT from the start menu, right click on Devices and click upload configuration as shown below.



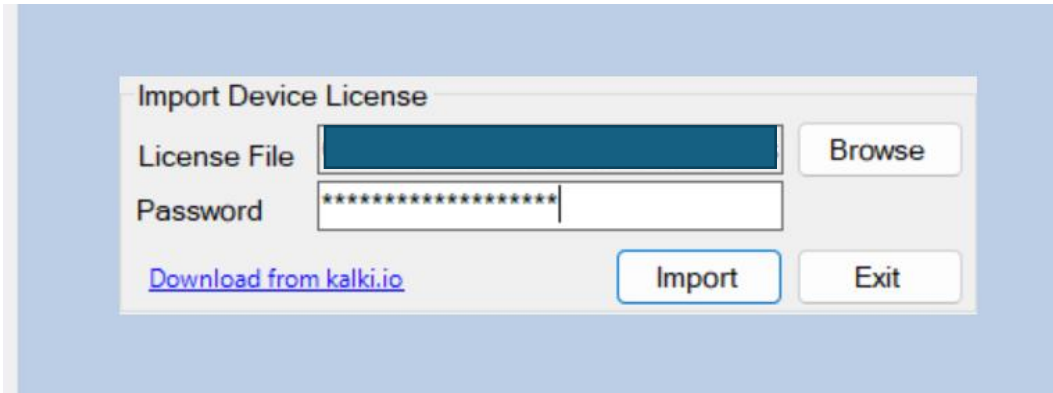
2. Click on SS4:KIOE from the left panel and select Device license in right panel.



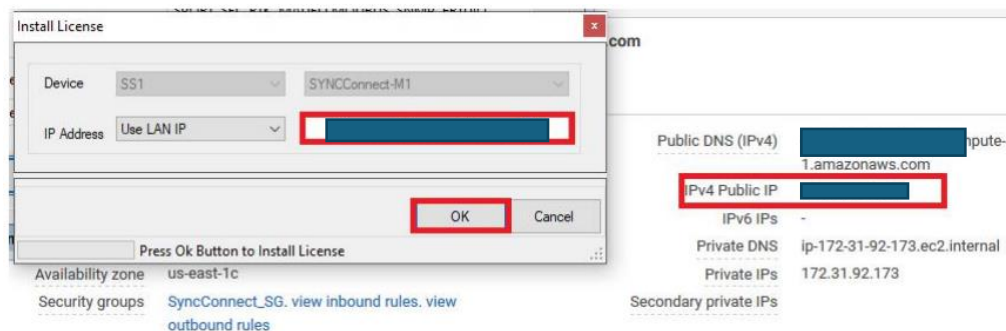
3. Click on Install License to device option as shown below.



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.

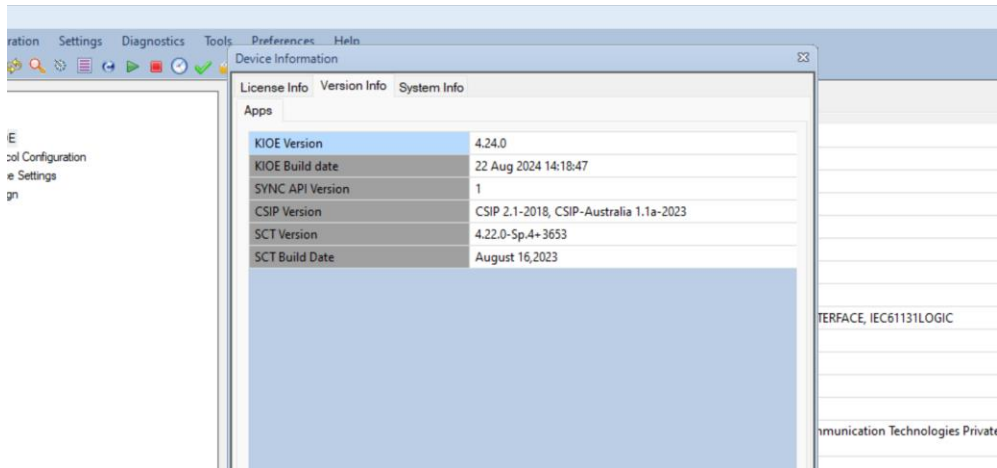


This step verifies **KIOE version**. In SCT toolbar go to **Settings -> Version Information**. Provide IP for KIOE host and press OK.

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

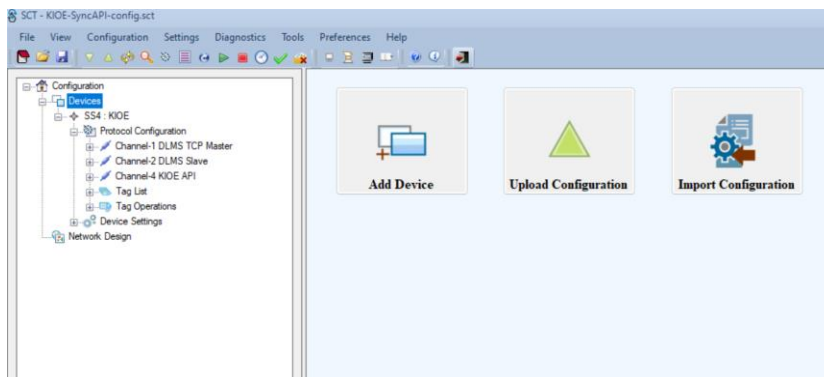


You can get more information about licensed protocols in **the License Info** tab.

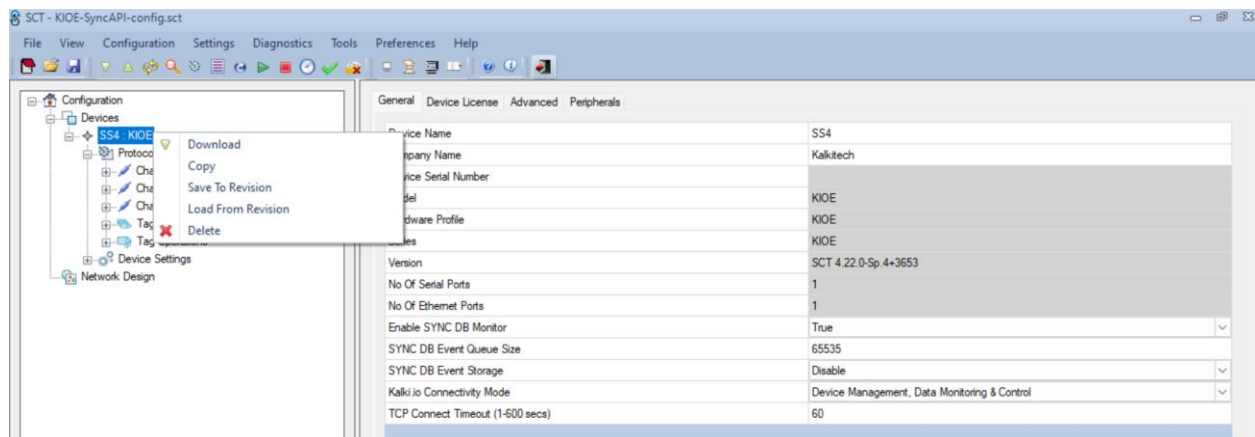


### 3.3 Protocol configuration

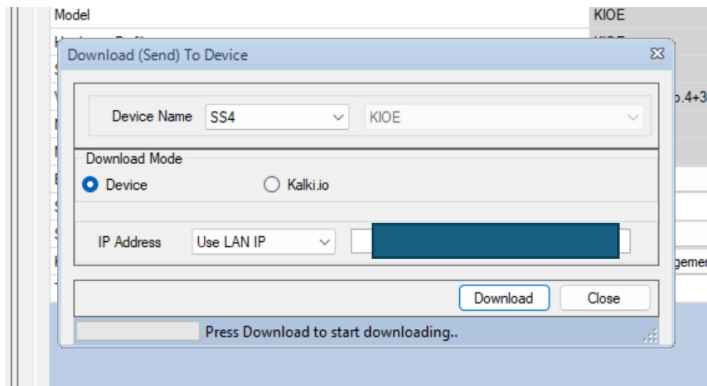
1. Open the shared file named 'KIOE-SyncAPI-config.sct'.
2. You will see a window like shown below



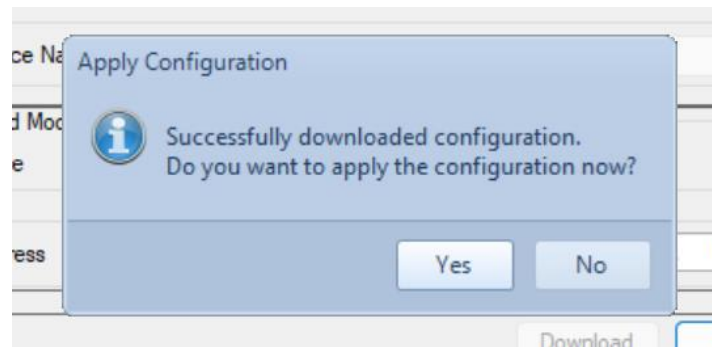
3. Right click on SS4:KIOE and click download



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



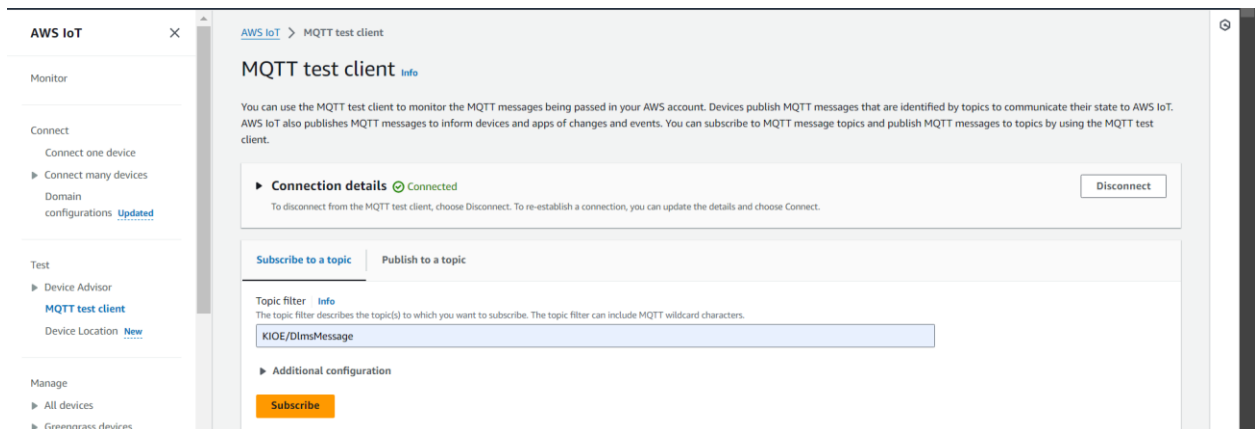
5. Click yes to apply configuration.



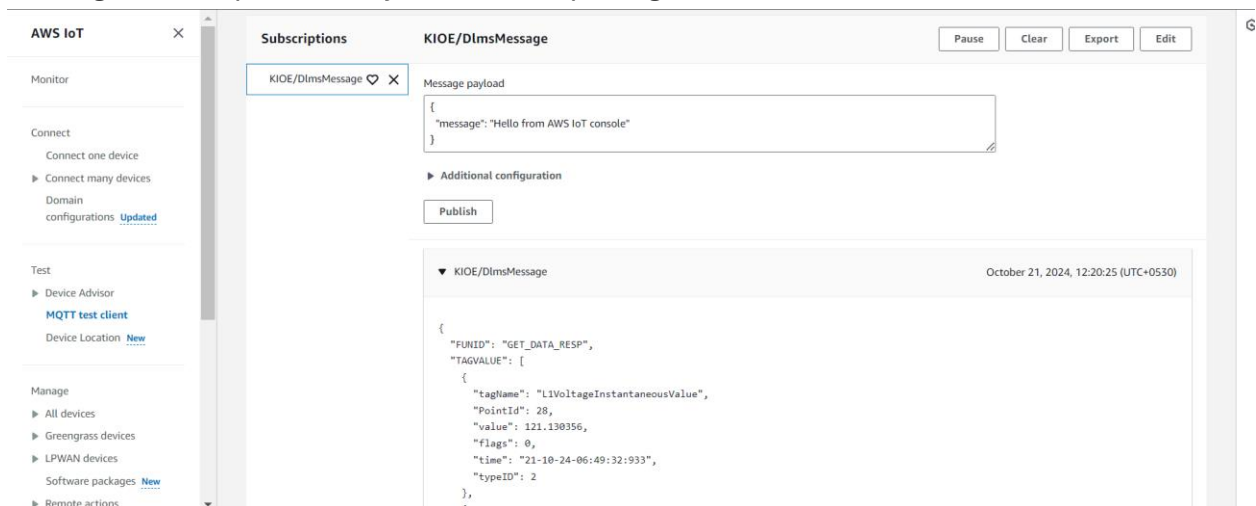
## 4. Verification of Installation

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 L1CurrentInstantaneousValue and PowerInstantaneousValue of 0. (Refer screenshot).

The screenshot shows the AWS IoT console interface. On the left, there is a sidebar with navigation options: Monitor, Connect (with sub-options: Connect one device, Connect many devices, Domain configurations), Test (with sub-options: Device Advisor, MQTT test client, Device Location), and Manage (with sub-options: All devices, Greengrass devices, LPWAN devices, Software packages, Remote actions). The main area displays a JSON payload for a Thing named "MarketplaceThing-4". The payload contains three sensor readings, each with a "value" field highlighted by a red box:

- L1CurrentInstantaneousValue**: "value": 0, "PointId": 20, "time": "21-10-24-07:16:36:085", "typeID": 2
- PowerInstantaneousValue**: "value": 0, "PointId": 31, "time": "21-10-24-07:16:23:690", "typeID": 2
- InstantaneousFrequency**: "value": 60.064896, "PointId": 32, "time": "21-10-24-07:18:04:102", "typeID": 2

**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

