Industrial Machine Connectivity (IMC) Kit

Table of Contents

[Industrial Machine Connectivity QuickStart Getting Started Guide 3](#_Toc43381449)

[Deployment Types 4](#_Toc43381450)

[Virtual: 4](#_Toc43381451)

[Physical 4](#_Toc43381452)

[Physical - Greenfield 4](#_Toc43381453)

[Physical - Brownfield 4](#_Toc43381454)

[Data Flow Options 5](#_Toc43381455)

[Option 1 5](#_Toc43381456)

[Option 2a 5](#_Toc43381457)

[Option 2b 5](#_Toc43381458)

[Virtual 6](#_Toc43381459)

[Pre-Requisites: 7](#_Toc43381460)

[Virtual Option 1 Cirrus Link Module 8](#_Toc43381461)

[CloudFormation stack launch 8](#_Toc43381462)

[Post Deployment Steps 12](#_Toc43381463)

[View SiteWise Portal Data 13](#_Toc43381464)

[Troubleshooting 14](#_Toc43381465)

[Virtual Option 1 Ignition File Export 15](#_Toc43381466)

[CloudFormation stack launch 15](#_Toc43381467)

[Post Deployment Steps 17](#_Toc43381468)

[View SiteWise Portal Data 19](#_Toc43381469)

[Troubleshooting 20](#_Toc43381470)

[Virtual Option 2a Cirrus Link Module 21](#_Toc43381471)

[CloudFormation stack launch 21](#_Toc43381472)

[Post Deployment Steps 23](#_Toc43381473)

[Troubleshooting 24](#_Toc43381474)

[Virtual Option 2a Ignition File Export 25](#_Toc43381475)

[CloudFormation stack launch 25](#_Toc43381476)

[Post Deployment Steps 27](#_Toc43381477)

[Troubleshooting 29](#_Toc43381478)

[Virtual Option 2b Cirrus Link Module 30](#_Toc43381479)

[CloudFormation stack launch 30](#_Toc43381480)

[Post Deployment Steps 32](#_Toc43381481)

[Troubleshooting 33](#_Toc43381482)

[Virtual Option 2b Ignition File Export 34](#_Toc43381483)

[CloudFormation stack launch 34](#_Toc43381484)

[Post Deployment Steps 36](#_Toc43381485)

[Troubleshooting 38](#_Toc43381486)

[Virtual Cleanup 39](#_Toc43381487)

[Virtual FAQs 40](#_Toc43381488)

[Physical – Greenfield 43](#_Toc43381489)

[Pre-Requisites 44](#_Toc43381490)

[Physical Greenfield Option 1 Cirrus Link Module 46](#_Toc43381491)

[CloudFormation stack launch 46](#_Toc43381492)

[Post Deployment Steps 48](#_Toc43381493)

[View SiteWise Portal Data 51](#_Toc43381494)

[Troubleshooting 52](#_Toc43381495)

[Physical Greenfield Option 1 Ignition File Export 53](#_Toc43381496)

[CloudFormation stack launch 53](#_Toc43381497)

[Post Deployment Steps 55](#_Toc43381498)

[View SiteWise Portal Data 58](#_Toc43381499)

[Troubleshooting 59](#_Toc43381500)

[Physical Greenfield Option 2a Cirrus Link Module 60](#_Toc43381501)

[CloudFormation stack launch 60](#_Toc43381502)

[Post Deployment Steps 62](#_Toc43381503)

[Troubleshooting 64](#_Toc43381504)

[Physical Greenfield Option 2a Ignition File Export 65](#_Toc43381505)

[CloudFormation stack launch 65](#_Toc43381506)

[Post Deployment Steps 67](#_Toc43381507)

[Troubleshooting 70](#_Toc43381508)

[Physical Greenfield Option 2b Cirrus Link Module 71](#_Toc43381509)

[CloudFormation stack launch 71](#_Toc43381510)

[Post Deployment Steps 73](#_Toc43381511)

[Troubleshooting 76](#_Toc43381512)

[Physical Greenfield Option 2b Ignition File Export 77](#_Toc43381513)

[CloudFormation stack launch 77](#_Toc43381514)

[Post Deployment Steps 79](#_Toc43381515)

[Troubleshooting 82](#_Toc43381516)

[Physical Greenfield Cleanup 83](#_Toc43381517)

[Physical Greenfield FAQs 85](#_Toc43381518)

[Physical-Brownfield 87](#_Toc43381519)

[Pre-Requisites 88](#_Toc43381520)

[Physical Brownfield Option 1 Ignition/KepServer File Export 90](#_Toc43381521)

[CloudFormation stack launch 90](#_Toc43381522)

[Post Deployment Steps 92](#_Toc43381523)

[View SiteWise Portal Data 94](#_Toc43381524)

[Troubleshooting 95](#_Toc43381525)

[Physical Brownfield Cleanup 96](#_Toc43381526)

[Physical Brownfield FAQs 98](#_Toc43381527)

[Creating AMC Drivers 100](#_Toc43381528)

[Instructions 100](#_Toc43381529)

[Appendix 101](#_Toc43381530)

[Artifacts 101](#_Toc43381531)

[AMC-Approved DynamoDB Format 101](#_Toc43381532)

# Industrial Machine Connectivity QuickStart Getting Started Guide

The Industrial Machine Connectivity (IMC) kit enables connecting industrial assets into AWS cloud services along with visualizing data using AWS IoT SiteWise Monitor and QuickSight dashboards. This kit integrates with AWS IoT Partner edge software such as Inductive Automation’s Ignition Server and PTC’s KEPServerEX. These partner edge software applications handle the industrial protocol translation from the PLCs, other devices, historians and SCADA systems they connect to. The IMC architecture integrates both the data collected by the edge software applications and the asset model hierarchy that they maintain.

A key component of this kit is the Asset Model Converter (AMC). The AMC is a serverless, module-based framework supporting mapping edge-based asset modeling software conventions into AWS IoT SiteWise models and assets.

Currently as part of this kit, we provide sample modules (drivers) for:

* Inductive Automation (Ignition Server)
  + Cirrus Link Module MQTT driver
  + Ignition Project File Export driver
* PTC (KepServer)
  + KepServer Project File Export driver

Based on the driver selected, the AMC will ingest the edge software’s (i.e. Ignition or KEPServerEX) native asset model definition(s) and automatically provision the matching asset hierarchy within AWS IoT SiteWise. This automatic mapping enables application builders, whether they be customer’s own developers, SIs, GSIs or AWS ProServe team, to have immediate access to the customer’s asset hierarchy within a managed service in the AWS Cloud (AWS IoT SiteWise).

The objective of the IMC kit is to accelerate industrial machine connectivity to the AWS cloud so solution builder teams can demonstrate real business value to customers faster and with lower integration cost and effort than is currently possible.

# Deployment Types

The IMC kit can be deployed in 3 configurations:

## Virtual:

The virtual deployment is intended for demonstration, training and evaluation of the Kit’s capabilities. EC2 instances will be launched to simulate edge gateway hardware but in all other respects the experience will mirror that of the real physical deployment. This deployment mode relies on simulated tag values generated by the partner edge software. There are no physical PLCs or sensors that are being connected.

## Physical

Physical deployment of the IMC kit enables users to deploy edge software (i.e. AWS IoT Greengrass and partner edge software) on physical industrial PCs that are ready to connect to physical devices (I.e. PLCs)/historians/SCADA systems on the customers plant floor. The physical deployment has two flavors:

### Physical - Greenfield

AWS IoT Greengrass and the partner edge software will be running on a single industrial PC.

### Physical - Brownfield

AWS IoT Greengrass will run standalone on an industrial PC and will connect to partner edge software application that is already running on the customers premises (i.e. on a VM in the server room of a manufacturing plant). We assume our access to the hardware is limited or non-existent, and our ability to reconfigure the Edge Software Application Server is limited to additive changes only.

-------------------------------------------------------------------------------------------------------------------------------

The type of deployment (Virtual or Physical) determines whether to use physical edge hardware (Physical industrial PCs) or virtual edge hardware (EC2) and how connectivity and security is configured. All other cloud-based resources are largely the same.

# Data Flow Options

In addition to the Virtual/Physical edge hardware distinction, the IMC kit supports three types of data flow architectures. Each type outlines different methods of data ingestion from the edge environment into the AWS Cloud.

## Option 1

**Edge Software Application OPCUA Server -> AWS IoT SiteWise:**

In this variant, we have the AWS Greengrass SiteWise Connector configured to connect to the Edge Software Application OPCUA Server. All telemetry data will flow directly into AWS IoT SiteWise.

## Option 2a

**Edge Software Application Server -> AWS IoT Core:**

In this variant, the Edge Software Application Server has some kind functionality to connect to IoT Core via MQTT. All telemetry data is pushed from the Edge Software Application Server to AWS IoT Core, and from there usually pushed to S3 or a similar data lake for processing.

## Option 2b

**Edge Software Application Server -> AWS Greengrass Core -> IoT Core -> S3:**

Option 2b is almost identical to option 2a, except we instead have the Edge Software Application Server pushing MQTT data messages to AWS Greengrass Core first, and then those messages are forwarded on to AWS IoT Core.

# Virtual

The virtual deployment is intended for demonstration, training and evaluation of the Kit’s capabilities. EC2 instances will be launched to simulate edge gateway hardware but in all other respects the experience will mirror that of the real physical deployment. This deployment mode relies on simulated tag values generated by the partner edge software. There are no physical PLCs or sensors that are being connected.

The virtual deployment has the following sections:

1. Pre-Requisites
2. CloudFormation stack launch
3. Post deployment steps
4. Troubleshooting
5. Virtual Cleanup
6. Virtual FAQ

## Pre-Requisites:

* **AWS account with SSO enabled:** <https://docs.aws.amazon.com/singlesignon/latest/userguide/getting-started.html>
* **EC2 Key Pair:** <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>
* **Create IoT SiteWise Service-linked Role Using the AWS CLI:** 
  + aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com --description "Service-linked role to support IoT SiteWise"
  + <https://docs.aws.amazon.com/iot-sitewise/latest/userguide/using-service-linked-roles.html>
* **QuickStart Bucket Preparation:**
  + Create an S3 bucket and give it a unique name such as “imc-quickstart-bucket-ABC-123”. In that S3 bucket create a folder called “quickstart-IMC”. You then copy all the IMC CloudFormation [artifacts](#_Artifacts_1) into the “quickstart-IMC” folder. The structure will then resemble the structure below:

S3 bucket name: “imc-quickstart-bucket-ABC-123”

S3 bucket Contents:

quickstart-IMC/

functions/

scripts/

templates/

LICENSE.txt

NOTICE.txt

README.md

## Virtual Option 1 Cirrus Link Module

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

* **Step 1: Specify templates**
  + Pre-Requisite:
    - Select “Template is ready”
  + Specify Template:
    - Template Source: Leave as default – “Amazon S3 URL”. Use the “IMC-workload.template.yaml” CloudFormation workload template that’s in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
      * The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
  + Click “Next” to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)
* **Step 2: Specify stack details**
  + Stack Name:
    - Stack Name: Give the stack a unique name such as “IMC-Virtual”
  + Parameters:
    - Edge Deployment Configuration
      * Name for the edge device**:** You may leave as default or, if desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
      * Type of Deployment (Virtual or Physical):Virtual
      * Deployment Flow: Select “Option 1”.
    - Amazon EC2 Configuration
      * SSH Key Name: Select your SSH Key Name (EC2 Key Pair). You will use this SSH key to SSH into the 2 EC2 instances that are running the partner edge software application and AWS IoT Greengrass, respectively.
      * VPC ID: Find your default VPC ID and copy it into this field.
      * Greengrass EC2 Instance Type**:** (default: t3.small) You can select a larger (t3.medium) instance if desired. It is suggested you leave the as the default value.
      * Ignition EC2 Instance Types: (default: t3.large) You can select a larger (t3.xlarge) instance if desired. It is suggested you leave the as the default value.
      * EC2 AMI: Select “ami-085925f297f89fce1” from the dropdown menu. This AMI is used for both EC2 instances (running Ignition and Greengrass).
      * EC2 Subnet: Find the VPC Subnet associated with availability zone us-east-1a in your account and use that value in this field.
    - AWS Quick Start Configuration
      * QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
      * QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section we named the folder “quickstart-IMC/”
      * QuickStart S3 Bucket Region: Leave as default “us-east-1”
      * Select the Asset Model Converter (AMC) Driver: Leave as default “IgnitionCirrusLink”.
      * User Public IP Address: Input your public IP address in the format “x.x.x.x” so that you have access to SSH into the EC2 instances.
  + Click “Next” to proceed to Step 3 “Configure stack options”
* **Step 3: Configure stack options**
  + Accept all defaults
  + Click “Next” to proceed to Step 4 “Review”
* **Step 4: Review**
  + Review and accept the acknowledgements at the bottom of the page
  + Click “Create stack” to launch the CloudFormation stack

**Stack Deployment:**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

* Verify SiteWise Asset Creation
  + After the CloudFormation stack is complete, navigate to the AWS IoT SiteWise console. From there you can watch as the assets and models are created in SiteWise and associated with each other into a single hierarchy.
  + This will take several minutes (up to 10 minutes), but once it's completed you should see an asset structure much like this one:

```

/AWS Smart Factory

/AWS Smart Factory/Smart Factory 1

/AWS Smart Factory/Smart Factory 1/Line 1

/AWS Smart Factory/Smart Factory 1/Line 1/Conveyer

/AWS Smart Factory/Smart Factory 1/Line 1/Hauloff

/AWS Smart Factory/Smart Factory 1/Line 1/Stamping Machine

...

```

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

**Option 1 Steps**

1. Accept SiteWise Certificate in Ignition: To enable the SiteWise to ingest data over OPC UA from Ignition’s OPC UA server, you must accept the certificate presented by the SiteWise connector within Ignition.
   1. Open the Ignition web UI: Navigate to the AWS EC2 Console and find the EC2 instance running Ignition server. Its name should end with /Ignition
   2. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<IginitionServerPublicIP>:8088
      1. Reminder: For AWS Employees, do not be on the corporate VPN for this step.
   3. Once the Ignition Web UI is open, you should see a gear like icon on the left labeled 'Config'. Click that, and it will ask you to log in. The default credentials are:
      1. Username: admin
      2. Password: password
   4. Navigate to "OPC UA -> Security -> Server" and wait for the quarantined certificate to appear (from AWS IoT SiteWise Gateway). You should see a single entry under 'Quarantined Certificates' named something like 'AWS IoT SiteWise Gateway Client'.
      1. Click “Trust” to accept the certificate. At this point, the SiteWise connector will start consuming data over OPC UA from Ignition and this data will be sent up to the AWS IoT SiteWise in the cloud.
2. Update the SiteWise Gateway
   1. Navigate to AWS IoT SiteWise console and select Ingest🡪Gateways
   2. Select the gateway created during the stack launch.
      1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
   3. Click “Edit” in the Source Configuration for Automated Gateway Config” section
   4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
3. Validate Incoming PLC Data
   1. Now that you've trusted the certificate, go back to the AWS IoT SiteWise console.
   2. In the SiteWise console, click the icon on the left side of the page, select build🡪assets
   3. In the asset tree on the left, drill down to an asset (i.e. Hauloff or Conveyor), select it and then select “Measurements” tab for that asset.
   4. Verify that the values in the “Latest value” column are updating. This indicates that the Ignition simulation of those virtual devices and sensors is properly sending data through to the SiteWise connector in Greengrass and up to AWS IoT SiteWise in the AWS cloud.

### View SiteWise Portal Data

**Log in to SiteWise Monitor Portal**

1. For a more visual display of the data, navigate to the SiteWise console, select the icon on the left and select Monitor🡪 Portals.
2. Select the hyperlinked "name" of the Portal most recently added (the topmost on the list).
3. Add yourself as an administrator of the Portal by clicking “Assign Users” in the Portal Administrators section
4. Once you are listed as a Portal Administrator, click the hyperlinked URL in the Portal details section under the “URL” column. This URL should have the format <https://[XXXXX....XXXXXX].app.iotsitewise.aws>.
5. Log in with the credentials (username and password) you just created for your administrator account.

**View Data in SiteWise Monitor Portal**

1. Select “Dashboards” tab on the left-hand side, then select the newly created dashboard hyperlink under the “Name” column of the Dashboards page.
   1. Data should be flowing into the line charts for the asset measurement properties
2. You can also see data for individual assets by navigating to the “Asset Library” tab on the left and selecting an asset from the asset tree. Once an asset is selected, you can view its properties.

### Troubleshooting

**Quarantined certificate in Ignition doesn't show up, or data doesn’t show up for Option 1 deployments**

First, verify that the Ignition trial period (2 hours) has not expired. If that action does not remediate the issue, repeat the process of refreshing the SiteWise Gateway:

1. Navigate to the AWS IoT SiteWise console and select Ingest 🡪 Gateways
2. Select the gateway created during the stack launch:
   1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
3. Click “Edit” in the Source Configuration for Automated Gateway Config section
4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
5. If it hasn’t already been done, look for and accept the quarantined certificate in Ignition.

## Virtual Option 1 Ignition File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

* **Step 1: Specify templates**
  + Pre-Requisite:
    - Select “Template is ready”
  + Specify Template:
    - Template Source: Leave as default – “Amazon S3 URL”. Use the “IMC-workload.template.yaml” CloudFormation workload template that’s in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
      * The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
  + Click “Next” to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)
* **Step 2: Specify stack details**
  + Stack Name:
    - Stack Name: Give the stack a unique name such as “IMC-Virtual”
  + Parameters:
    - Edge Deployment Configuration
      * Name for the edge device**:** You may leave as default or, if desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
      * Type of Deployment (Virtual or Physical):Virtual
      * Deployment Flow: Option 1
    - Amazon EC2 Configuration
      * SSH Key Name: Select your SSH Key Name (EC2 Key Pair). You will use this SSH key to SSH into the 2 EC2 instances that are running the partner edge software application and AWS IoT Greengrass, respectively.
      * VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
      * Greengrass EC2 Instance Type**:** (default: t3.small) You can select a larger (t3.medium) instance if desired. It is suggested you leave the as the default value.
      * Ignition EC2 Instance Types: (default: t3.large) You can select a larger (t3.xlarge) instance if desired. It is suggested you leave the as the default value.
      * EC2 AMI: Select “ami-085925f297f89fce1” from the dropdown menu. This AMI is used for both EC2 instances (running Ignition and Greengrass).
      * EC2 Subnet: Select the VPC Subnet associated with availability zone us-east-1a in your account.
    - AWS Quick Start Configuration
      * QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
      * QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section we named the folder “quickstart-IMC/”
      * QuickStart S3 Bucket Region: Leave as default “us-east-1”
      * Select the Asset Model Converter (AMC) Driver: Select “IgnitionFileExport”.
      * User Public IP Address: Input your public IP address in the format “x.x.x.x” so that you have access to SSH into the EC2 instances.
  + Click “Next” to proceed to Step 3 “Configure stack options”
* **Step 3: Configure stack options**
  + Accept all defaults
  + Click “Next” to proceed to Step 4 “Review”
* **Step 4: Review**
  + Review and accept the acknowledgements at the bottom of the page
  + Click “Create stack” to launch the CloudFormation stack

**Stack Deployment:**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

**Option 1 Steps**

1. Export the JSON file that describes your project hierarchy in Ignition.
   1. Download the Ignition Designer software
      1. Navigate to the EC2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC Kit
         1. The end of the EC2’s instance should contain “Ignition”
      2. In your browser of choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
      3. On the top right corner of the screen, click “Get Designer”
      4. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. A screenshot of a cell phone

      Description automatically generatedIn the Tag Browser, under “Tag Providers” select “default” and click export.
2. Upload the JSON file from step 1 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. Upon dropping the JSON file in this S3 bucket, an S3 event trigger will automatically invoke the AMC Lambda function
   2. Wait for the AMC to complete creating your models and assets in SiteWise
3. Accept SiteWise Certificate in Ignition: To enable the SiteWise to ingest data over OPC UA from Ignition’s OPC UA server, you must accept the certificate presented by the SiteWise connector within Ignition.
   1. Open the Ignition web UI: Navigate to the AWS EC2 Console and find the EC2 instance running Ignition server. It should be named you should see an instance named “Virtual/Option1/Ignition
   2. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<IginitionServerPublicIP>:8088
      1. Reminder: For AWS Employees, do not be on the corporate VPN for this step.
   3. Once the Ignition Web UI is open, you should see a gear like icon on the left labeled 'Config'. Click that, and it will ask you to log in. The default credentials are:
      1. Username: admin
      2. Password: password
   4. Navigate to "OPC UA -> Security -> Server" and wait for the quarantined certificate to appear (from AWS IoT SiteWise Gateway). You should see a single entry under 'Quarantined Certificates' named something like 'AWS IoT SiteWise Gateway Client'.
      1. Click “Trust” to accept the certificate. At this point, the SiteWise connector will start consuming data over OPC UA from Ignition and this data will be sent up to the AWS IoT SiteWise in the cloud.
4. Update the SiteWise Gateway
   1. Navigate to AWS IoT SiteWise console and select Ingest🡪Gateways
   2. Select the gateway created during the stack launch.
      1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
   3. Click “Edit” in the Source Configuration for Automated Gateway Config” section
   4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
5. Validate Incoming PLC Data
   1. Now that you've trusted the certificate, go back to the AWS IoT SiteWise console.
   2. In the SiteWise console, click the icon on the left side of the page, select build🡪assets
   3. In the asset tree on the left, drill down to an asset (i.e. Hauloff or Conveyor), select it and then select “Measurements” tab for that asset.
   4. Verify that the values in the “Latest value” column are updating. This indicates that the Ignition simulation of those virtual devices and sensors is properly sending data through to the SiteWise connector in Greengrass and up to AWS IoT SiteWise in the AWS cloud.

### View SiteWise Portal Data

**Log in to SiteWise Monitor Portal**

1. For a more visual display of the data, navigate to the SiteWise console, select the icon on the left and select Monitor🡪 Portals.
2. Select the hyperlinked "name" of the Portal most recently added (the topmost on the list).
3. Add yourself as an administrator of the Portal by clicking “Assign Users” in the Portal Administrators section
4. Once you are listed as a Portal Administrator, click the hyperlinked URL in the Portal details section under the “URL” column. This URL should have the format <https://[XXXXX....XXXXXX].app.iotsitewise.aws>.
5. Log in with the credentials (username and password) you just created for your administrator account.

**View Data in SiteWise Monitor Portal**

1. Select “Dashboards” tab on the left-hand side, then select the newly created dashboard hyperlink under the “Name” column of the Dashboards page.
   1. Data should be flowing into the line charts for the asset measurement properties
2. You can also see data for individual assets by navigating to the “Asset Library” tab on the left and selecting an asset from the asset tree. Once an asset is selected, you can view its properties.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Quarantined certificate in Ignition doesn't show up, or data doesn’t show up for Option 1 deployments**

First, verify that the Ignition trial period (2 hours) has not expired. If that action does not remediate the issue, repeat the process of refreshing the SiteWise Gateway:

1. Navigate to the AWS IoT SiteWise console and select Ingest 🡪 Gateways
2. Select the gateway created during the stack launch:
   1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
3. Click “Edit” in the Source Configuration for Automated Gateway Config section
4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
5. If it hasn’t already been done, look for and accept the quarantined certificate in Ignition.

## Virtual Option 2a Cirrus Link Module

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

* **Step 1: Specify templates**
  + Pre-Requisite:
    - Select “Template is ready”
  + Specify Template:
    - Template Source: Leave as default – “Amazon S3 URL”. Use the “IMC-workload.template.yaml” CloudFormation workload template that’s in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
      * The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
  + Click “Next” to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)
* **Step 2: Specify stack details**
  + Stack Name:
    - Stack Name: Give the stack a unique name such as “IMC-Virtual”
  + Parameters:
    - Edge Deployment Configuration
      * Name for the edge device**:** You may leave as default or, if desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
      * Type of Deployment (Virtual or Physical):Virtual
      * Deployment Flow: Option 2a
    - Amazon EC2 Configuration
      * SSH Key Name: Select your SSH Key Name (EC2 Key Pair). You will use this SSH key to SSH into the 2 EC2 instances that are running the partner edge software application and AWS IoT Greengrass, respectively.
      * VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
      * Greengrass EC2 Instance Type**:** (default: t3.small) You can select a larger (t3.medium) instance if desired. It is suggested you leave the as the default value.
      * Ignition EC2 Instance Types: (default: t3.large) You can select a larger (t3.xlarge) instance if desired. It is suggested you leave the as the default value.
      * EC2 AMI: Select “ami-085925f297f89fce1” from the dropdown menu. This AMI is used for both EC2 instances (running Ignition and Greengrass).
      * EC2 Subnet: Select the VPC Subnet associated with availability zone us-east-1a in your account.
    - AWS Quick Start Configuration
      * QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
      * QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section we named the folder “quickstart-IMC/”
      * QuickStart S3 Bucket Region: Leave as default “us-east-1”
      * Select the Asset Model Converter (AMC) Driver: Select “CirrusLinkModule”.
      * User Public IP Address: Input your public IP address in the format “x.x.x.x” so that you have access to SSH into the EC2 instances.
  + Click “Next” to proceed to Step 3 “Configure stack options”
* **Step 3: Configure stack options**
  + Accept all defaults
  + Click “Next” to proceed to Step 4 “Review”
* **Step 4: Review**
  + Review and accept the acknowledgements at the bottom of the page
  + Click “Create stack” to launch the CloudFormation stack

**Stack Deployment:**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2a deployments, models and assets are not required to collect data.

**Option 2a Steps**

1. Download the Ignition Designer software
   1. Navigate to the EC2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC Kit
      1. The end of the EC2’s instance should contain “Ignition”
   2. In your browser of choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
   3. On the top right corner of the screen, click “Get Designer”
   4. Install the Ignition Designer for your operating system
2. Open the Ignition Designer and connect to your instance of Ignition
   1. Launch the Designer
   2. Click “Add Designer”
   3. Click “Manually Add Gateway”
   4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
   5. Under the Gateway tile you just added, click “Launch”
   6. Supply the username and password and click “Login”
      1. Username: admin
      2. Password: password
3. Trigger an MQTT node “birth” message
   1. Note for AWS/SME: a birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
   2. This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
      1. \*\* Not required to collect data for this deployment option
4. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
5. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Open the Ignition web UI: Navigate to the AWS EC2 Console and find the EC2 instance running Ignition server. It should be named “Virtual/[Option2a\_or\_Option2b]/Ignition
2. Get the public IP address of that instance, and load a URL like this into your broser of choice: http://<IgnitionServerPublicIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
3. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
4. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[your\_aws\_account\_iot\_endpoint]:8883
   2. Download the .tar.gz file that represents the non-GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Replace the CA Certificate File with “root.ca.pem” from the tarball package
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Private Key File with the “.private” file from the tarball package
   7. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Virtual Option 2a Ignition File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

* **Step 1: Specify templates**
  + Pre-Requisite:
    - Select “Template is ready”
  + Specify Template:
    - Template Source: Leave as default – “Amazon S3 URL”. Use the “IMC-workload.template.yaml” CloudFormation workload template that’s in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
      * The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
  + Click “Next” to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)
* **Step 2: Specify stack details**
  + Stack Name:
    - Stack Name: Give the stack a unique name such as “IMC-Virtual”
  + Parameters:
    - Edge Deployment Configuration
      * Name for the edge device**:** You may leave as default or, if desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
      * Type of Deployment (Virtual or Physical):Virtual
      * Deployment Flow: Option 2a
    - Amazon EC2 Configuration
      * SSH Key Name: Select your SSH Key Name (EC2 Key Pair). You will use this SSH key to SSH into the 2 EC2 instances that are running the partner edge software application and AWS IoT Greengrass, respectively.
      * VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
      * Greengrass EC2 Instance Type**:** (default: t3.small) You can select a larger (t3.medium) instance if desired. It is suggested you leave the as the default value.
      * Ignition EC2 Instance Types: (default: t3.large) You can select a larger (t3.xlarge) instance if desired. It is suggested you leave the as the default value.
      * EC2 AMI: Select “ami-085925f297f89fce1” from the dropdown menu. This AMI is used for both EC2 instances (running Ignition and Greengrass).
      * EC2 Subnet: Select the VPC Subnet associated with availability zone us-east-1a in your account.
    - AWS Quick Start Configuration
      * QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
      * QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section we named the folder “quickstart-IMC/”
      * QuickStart S3 Bucket Region: Leave as default “us-east-1”
      * Select the Asset Model Converter (AMC) Driver: Select “IgnitionFileExport”.
      * User Public IP Address: Input your public IP address in the format “x.x.x.x” so that you have access to SSH into the EC2 instances.
  + Click “Next” to proceed to Step 3 “Configure stack options”
* **Step 3: Configure stack options**
  + Accept all defaults
  + Click “Next” to proceed to Step 4 “Review”
* **Step 4: Review**
  + Review and accept the acknowledgements at the bottom of the page
  + Click “Create stack” to launch the CloudFormation stack

**Stack Deployment:**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2a deployments, models and assets are not required to collect data.

**Option 2a Steps**

1. Export the JSON file that describes your project hierarchy in Ignition.
   1. Download the Ignition Designer software
      1. Navigate to the EC2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC Kit
         1. The end of the EC2’s instance should contain “Ignition”
      2. In your browser of choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
      3. On the top right corner of the screen, click “Get Designer”
      4. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. A screenshot of a cell phone

      Description automatically generatedIn the Tag Browser, under “Tag Providers” select “default” and click export.
2. Upload the JSON file from step 1 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. Upon dropping the JSON file in this S3 bucket, an S3 event trigger will automatically invoke the AMC Lambda function
   2. Wait for the AMC to complete creating your models and assets in SiteWise
3. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
4. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Open the Ignition web UI: Navigate to the AWS EC2 Console and find the EC2 instance running Ignition server. Its name should end with /Ignition
2. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<IgnitionServerPublicIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
3. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
4. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[your\_aws\_account\_iot\_endpoint]:8883
   2. Download the .tar.gz file that represents the non-GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Replace the CA Certificate File with “root.ca.pem” from the tarball package
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Private Key File with the “.private” file from the tarball package
   7. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Virtual Option 2b Cirrus Link Module

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

* **Step 1: Specify templates**
  + Pre-Requisite:
    - Select “Template is ready”
  + Specify Template:
    - Template Source: Leave as default – “Amazon S3 URL”. Use the “IMC-workload.template.yaml” CloudFormation workload template that’s in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
      * The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
  + Click “Next” to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)
* **Step 2: Specify stack details**
  + Stack Name:
    - Stack Name: Give the stack a unique name such as “IMC-Virtual”
  + Parameters:
    - Edge Deployment Configuration
      * Name for the edge device**:** You may leave as default or, if desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
      * Type of Deployment (Virtual or Physical):Virtual
      * Deployment Flow: Option 2b
    - Amazon EC2 Configuration
      * SSH Key Name: Select your SSH Key Name (EC2 Key Pair). You will use this SSH key to SSH into the 2 EC2 instances that are running the partner edge software application and AWS IoT Greengrass, respectively.
      * VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
      * Greengrass EC2 Instance Type**:** (default: t3.small) You can select a larger (t3.medium) instance if desired. It is suggested you leave the as the default value.
      * Ignition EC2 Instance Types: (default: t3.large) You can select a larger (t3.xlarge) instance if desired. It is suggested you leave the as the default value.
      * EC2 AMI: Select “ami-085925f297f89fce1” from the dropdown menu. This AMI is used for both EC2 instances (running Ignition and Greengrass).
      * EC2 Subnet: Select the VPC Subnet associated with availability zone us-east-1a in your account.
    - AWS Quick Start Configuration
      * QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
      * QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section we named the folder “quickstart-IMC/”
      * QuickStart S3 Bucket Region: Leave as default “us-east-1”
      * Select the Asset Model Converter (AMC) Driver: Select “CirrusLinkModule”.
      * User Public IP Address: Input your public IP address in the format “x.x.x.x” so that you have access to SSH into the EC2 instances.
  + Click “Next” to proceed to Step 3 “Configure stack options”
* **Step 3: Configure stack options**
  + Accept all defaults
  + Click “Next” to proceed to Step 4 “Review”
* **Step 4: Review**
  + Review and accept the acknowledgements at the bottom of the page
  + Click “Create stack” to launch the CloudFormation stack

**Stack Deployment:**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2b deployments, models and assets are not required to collect data.

**Option 2b Steps**

1. Download the Ignition Designer software
   1. Navigate to the EC2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC Kit
      1. The end of the EC2’s instance should contain “Ignition”
   2. In your browser of choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
   3. On the top right corner of the screen, click “Get Designer”
   4. Install the Ignition Designer for your operating system
2. Open the Ignition Designer and connect to your instance of Ignition
   1. Launch the Designer
   2. Click “Add Designer”
   3. Click “Manually Add Gateway”
   4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
   5. Under the Gateway tile you just added, click “Launch”
   6. Supply the username and password and click “Login”
      1. Username: admin
      2. Password: password
3. Trigger an MQTT node “birth” message
   1. Note for AWS/SME: a birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
   2. This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
      1. \*\* Not required to collect data for this deployment option
4. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
5. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Open the Ignition web UI: Navigate to the AWS EC2 Console and find the EC2 instance running Ignition server. It should be named “Virtual/[Option2a\_or\_Option2b]/Ignition
2. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<IgnitionServerPublicIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
3. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
4. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[greengrass\_ec2\_private\_ip\_address]:8883
   2. Download the .tar.gz file that represents the GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Retrieve the GreenGrass group CA certificate by running the following command, replacing the sections in red with their correstponding CloudFormation template outputs found in the “NESTED” stack:
      1. wget -O root.ca.pem <https://$RESTAPIID.execute-api.$REGION.amazonaws.com/api/deploygg/$EDGEDEVICEID>
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Certificate File with the “.private” file from the tarball package
   7. Replace the CA Certificate File with the “root.ca.pem” file retrieved from step d.
   8. Check the “show advanced properties” box
   9. Make sure the client ID reads: [name\_for\_edge\_device\_parameter]Device
   10. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Virtual Option 2b Ignition File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

* **Step 1: Specify templates**
  + Pre-Requisite:
    - Select “Template is ready”
  + Specify Template:
    - Template Source: Leave as default – “Amazon S3 URL”. Use the “IMC-workload.template.yaml” CloudFormation workload template that’s in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
      * The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
  + Click “Next” to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)
* **Step 2: Specify stack details**
  + Stack Name:
    - Stack Name: Give the stack a unique name such as “IMC-Virtual”
  + Parameters:
    - Edge Deployment Configuration
      * Name for the edge device**:** You may leave as default or, if desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
      * Type of Deployment (Virtual or Physical):Virtual
      * Deployment Flow: Option 2b
    - Amazon EC2 Configuration
      * SSH Key Name: Select your SSH Key Name (EC2 Key Pair). You will use this SSH key to SSH into the 2 EC2 instances that are running the partner edge software application and AWS IoT Greengrass, respectively.
      * VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
      * Greengrass EC2 Instance Type**:** (default: t3.small) You can select a larger (t3.medium) instance if desired. It is suggested you leave the as the default value.
      * Ignition EC2 Instance Types: (default: t3.large) You can select a larger (t3.xlarge) instance if desired. It is suggested you leave the as the default value.
      * EC2 AMI: Select “ami-085925f297f89fce1” from the dropdown menu. This AMI is used for both EC2 instances (running Ignition and Greengrass).
      * EC2 Subnet: Select the VPC Subnet associated with availability zone us-east-1a in your account.
    - AWS Quick Start Configuration
      * QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
      * QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section we named the folder “quickstart-IMC/”
      * QuickStart S3 Bucket Region: Leave as default “us-east-1”
      * Select the Asset Model Converter (AMC) Driver: Select “IgnitionFileExport”.
      * User Public IP Address: Input your public IP address in the format “x.x.x.x” so that you have access to SSH into the EC2 instances.
  + Click “Next” to proceed to Step 3 “Configure stack options”
* **Step 3: Configure stack options**
  + Accept all defaults
  + Click “Next” to proceed to Step 4 “Review”
* **Step 4: Review**
  + Review and accept the acknowledgements at the bottom of the page
  + Click “Create stack” to launch the CloudFormation stack

**Stack Deployment:**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2b deployments, models and assets are not required to collect data.

**Option 2b Steps**

1. Download the Ignition Designer software
   1. Navigate to the Ec2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC kit
      1. The end of the Ec2’s instance should contain “Ignition”
   2. In your browser choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
      1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
   3. On the top corner of the screen, click “Get Designer”
   4. Install the Ignition Designer for your operating system
2. Export the JSON file that describes your project hierarchy in Ignition.
   1. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   2. A screenshot of a cell phone

      Description automatically generatedIn the Tag Browser, under “Tag Providers” select “default” and click export.
3. Upload the JSON file from step 1 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. Upon dropping the JSON file in this S3 bucket, an S3 event trigger will automatically invoke the AMC Lambda function
   2. Wait for the AMC to complete creating your models and assets in SiteWise
4. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
5. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Open the Ignition web UI: Navigate to the AWS EC2 Console and find the EC2 instance running Ignition server. It should be named “Virtual/[Option2a\_or\_Option2b]/Ignition
2. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<IgnitionServerPublicIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
3. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
4. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[greengrass\_ec2\_private\_ip\_address]:8883
   2. Download the .tar.gz file that represents the GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Retrieve the GreenGrass group CA certificate by running the following command, replacing the sections in red with their corresponding CloudFormation template outputs found in the “NESTED” stack:
      1. wget -O root.ca.pem <https://$RESTAPIID.execute-api.$REGION.amazonaws.com/api/deploygg/$EDGEDEVICEID>
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Certificate File with the “.private” file from the tarball package
   7. Replace the CA Certificate File with the “root.ca.pem” file retrieved from step d.
   8. Check the “show advanced properties” box
   9. Make sure the client ID reads: [name\_for\_edge\_device\_parameter]Device
   10. Hit “Save Changes”, and make sure the connectivity says “1 of 1”.

## Virtual Cleanup

Follow these steps to clean up the IMC CloudFormation stack deployment:

1. Navigate to the CloudFormation console and delete the base stack (not the stack named "NESTED"), in order to clean up the account as much as possible. Most of the resources will be deleted, but the stack deletion will fail due to non-empty S3 buckets and potentially a deployed Greengrass group (for all Virtual options by default, and for all Physical deployments that have been completed on a piece of hardware. The steps required to delete a stack are:
   1. Empty the S3 buckets:
      1. Navigate to the S3 service in the AWS Console.
      2. In the search bar, enter your stack name.
      3. For each bucket that is associated with the stack (naming convention: [name\_of\_stack]-[bucket\_identifier]-unique hash), select the bucket, and click “Empty” under the search bar.
      4. Here are the following “bucket identifiers” that exist for each deployment:
         1. amcincomingresource
         2. amcoutputresource
         3. devicesbucketresource
         4. imcs3bucket
         5. lambdazipsbucket
   2. Force a reset of the GreenGrass group:
      1. Navigate to the GreenGrass console
      2. Select the GreenGrass group with the “Name for the Edge device” parameter name provided to the stack
      3. Under “Actions”, select “Reset Deployments”
      4. Check the box that asks if you want to force the reset
      5. Click “Reset Deployment”
   3. Navigate back to the CloudFormation console and once again delete the base stack.
      1. There is a master and nested stack. you must delete the master stack (the one that does **not** have “NESTED” in a gray box associated with it.
2. Other resources to clean up after stack deletion: SiteWise Portal, SiteWise Gateway, **SiteWise Models and Assets**, QuickSight dataset.

## Virtual FAQs

**Why do the instructions for Options 2a and 2b include creating models and assets in SiteWise if data doesn’t populate via SiteWise?**

* The purpose of the IMC kit, in many respects, is to show the numerous possibilities of automation and expediency surrounding asset modeling between edge-based asset modeling software and SiteWise. Although not required, the creation of the models and assets during these data flow options highlights the AMC’s versatility as a module based solution.

**Can I update a stack to a different deployment type (Physical, Virtual) or dataflow option (Option1, 2a, 2b)?**

* Updates are currently not supported. To achieve a different deployment type or dataflow type, you'll need to [deploy a new stack](#_Virtual_1). See the details on having more than one stack deployed in an account below, and then decide whether or not you’ll want to [delete your original stack](#_Cleanup) before re-creating another.

**Can I deploy multiple times in the same AWS account?**

Yes, you may deploy multiple stacks in the same account – but be wary of the following:

* For Option 1 deployments, SiteWise may receive data from multiple sources if there are, for example, two instances of Ignition that are publishing data onto the topic “/Tag Providers/default/Line1/CNC/Temperature”.
* For Option 2a and 2b deployments, data ingestion pipelines are not deployment specific, and data will flow into AWS IoT on the same MQTT topics. This means if both a "Virtual Option 2a" and "Virtual Option 2b" deployment exists, data from the "Virtual Option 2a" deployment will appear in the "Virtual Option 2b" S3 bucket. To temporarily prevent this, you may disable the IoT Rules associated with the deployment you no longer want to receive data from. Find the IoT Rules associated with a specific deployment by the CloudFormation stack name.
* If SiteWise models and assets exist in the Cloud from a previous deployment, a new deployment will not re-create the model/asset hierarchy. Instead, the Asset Model Converter operates by recognizing any “deltas” in the existing models/assets and the new deployment hierarchy. For example:
  + If the previously deployed hierarchy is identical to the newly deployed, nothing in SiteWise is changed.
  + If the newly deployed hierarchy has an identical structure to the previously deployed hierarchy **with additional nodes that follow the hierarchy pattern,** the new models/assets will be created in SiteWise**:**
    - Previous:
      * Line 1/CNC1/Temperature
    - New:
      * Line 1/CNC1/Temperature
      * Line 1/CNC2/Temperature

If the newly deployed hierarchy has a different hierarchy than the previously deployed hierarchy altogether, the Asset Model Converter will not succeed in creating the new models and assets.

**Can I re-run the AMC to create new models and assets?**

Yes. Follow these steps:

1. If any of the models/assets you want to create share a name with the models/assets created in the first pass of the AMC execution, you’ll need to delete the models and assets in SiteWise.
2. The second action you’ll need to take is clearing out the following DynamoDB tables associated with the initial IMC kit deployment:
   1. [stack-name]-asset-model-table
   2. [stack-name]-asset-table

Depending on your driver of choice, you’ll need to take the follow actions to re-run the AMC:

* IgnitionCirrusLink
  + Initiate a re-birth (NBIRTH,DBIRTH) MQTT message that represents the your project hierarchy.
    - Connect to your instance of Ignition in the Ignition Designer
    - A screenshot of a cell phone

      Description automatically generatedA birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
    - This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
* IgnitionFileExport
  + Obtain your JSON file that represents the new hierarchy and drop it into the following S3 bucket:
    - [name\_of\_stack]-[amcincomingresource]-[hash]
* KepServerFileExport
  + Obtain your JSON file that represents the new hierarchy and drop it into the following S3 bucket:
    - [name\_of\_stack]-[amcincomingresource]-[hash]

# Physical – Greenfield

The physical greenfield deployment is intended to demonstrate the capabilities of the IMC kit in an environment where the end user does not have existing edge-based asset modeling software (such as Ignition or KepServer). It is deployed onto physical hardware. After deployment, the physical hardware will run GreenGrass software and edge-based asset modeling software (Ignition). The IMC kit supports the following OEM devices:

1. **Lenovo**
   1. Model: ThinkCentre M90n IoT
   2. Architecture: Intel® Celeron® 4205U (x86)
   3. URL: <https://www.lenovo.com/us/en/desktops-and-all-in-ones/thinkcentre/m-series-tiny/ThinkCentre-M90n-IoT/p/thinkcentre-m90n-iot>
2. **ADLINK** 
   1. Model: MXE-211
   2. Architecture: Intel Atom® Processor E3900 (x86)
   3. <https://www.adlinktech.com/Products/Industrial_IoT_and_Cloud_solutions/IoTGateway/MXE-210_Series?lang=en>
3. **OnLogic**
   1. Model: Karbon 300 Compact Rugged Computer
   2. Architecture: Intel Atom® E3930 or E3950 processors
   3. URL: <https://onlogic.com/k300/>
4. **Advantech** 
   1. Model: UNO-2372G
   2. Architecture: Intel Atom E3845/Celeron® J1900 Quad-Core Processors
   3. URL: <https://www.advantech.com/products/1-2mlj9a/uno-2372g/mod_f4ff5680-f016-44bd-bff0-e5eddfd82237>
5. **MOXA**
   1. Model: MC-1112-E4-T
   2. Architecture: Intel Atom® Processor E3845 processor
   3. URL: <https://www.moxa.com/en/products/industrial-computing/x86-computers/mc-1100-series/mc-1121-e4-t>

This deployment mode does not come with a configured set of project tags similar to the virtual deployment but does come with a set of device simulations that can be configured to represent a project tag structure similar to the virtual deployment tag structure (or your own structure entirely). This deployment can be configured to work with a physical PLC test harness.

The virtual deployment has the following sections:

1. Pre-Requisites
2. CloudFormation stack launch
3. Post deployment steps
4. Troubleshooting
5. Physical Greenfield Cleanup
6. Physical Greenfield FAQ

## Pre-Requisites

* **AWS account with SSO enabled:** <https://docs.aws.amazon.com/singlesignon/latest/userguide/getting-started.html>
* **EC2 Key Pair:** <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>
* **Create IoT SiteWise Service-linked Role Using the AWS CLI:** 
  + aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com --description "Service-linked role to support IoT SiteWise"
  + <https://docs.aws.amazon.com/iot-sitewise/latest/userguide/using-service-linked-roles.html>
* **Connect to your physical hardware running Ubuntu 18.04, make sure it is connected to the internet, and configure the AWS CLI:**
  + You may achieve this however you please – either via an SSH command or a connection to the device with a keyboard and monitor
  + Ensure you have the ability to connect to the internet
  + Configure the AWS CLI on the hardware to communicate with the AWS account you plan to deploy in: <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>
* **The directory structure of the physical hardware running Ubuntu 18.04 must look like the following:**

home/

ubuntu

* **QuickStart Bucket Preparation:**
  + Create an S3 bucket and give it a unique name such as “imc-quickstart-bucket-ABC-123”. In that S3 bucket create a folder called “quickstart-IMC”. You then copy all the IMC CloudFormation [artifacts](#_Artifacts_1) into the “quickstart-IMC” folder. The structure will then resemble the structure below:

S3 bucket name: “imc-quickstart-bucket-ABC-123”

S3 bucket Contents:

quickstart-IMC/

functions/

scripts/

templates/

LICENSE.txt

NOTICE.txt

README.md

## Physical Greenfield Option 1 Cirrus Link Module

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:
  1. Template Source: Leave as default - “Amazon S3 URL”
  2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
     1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 1”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Leave as default “IgnitionCirrusLink”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

**Option 1**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptGreenfieldOption1] physical-greenfield-option1.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-option1.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsGreenfield1and2a]
   6. Replace [HardwareIP] with the physical device’s private IP address
   7. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-greenfield-option1.sh physicalgreenfieldoption1-devicesbucketresource-1ifk8w3a77621 PhysicalGreenfieldOption1/PhysicalGreenfieldOption1Core.tar.gz 866prvj43g us-east-1 PhysicalGreenfieldOption1 [Hardware-IP] PhysicalGreenfieldOption1/PhysicalGreenfieldOption1Device.tar.gz [a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com](http://a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com/)
2. Represent your data in Ignition by creating a project tag hierarchy. The source of this data can be physical PLCs, or simulated devices in Ignition.
   1. Download the Ignition Designer software
      1. In your browser of choice, visit the following URL: http://<hardwarePrivateIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      2. On the top right corner of the screen, click “Get Designer”
      3. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: http://<reachableIgnitionIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. With the help of an OT professional or IMC Kit contact, represent your PLC data (simulated or real) in a hierarchy
3. Trigger a node “birth” message
   1. Using the steps in 2.b, connect to your instance of Ignition
   2. A screenshot of a cell phone

      Description automatically generatedA birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
   3. This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
4. Accept SiteWise Certificate in Ignition: To enable the SiteWise to ingest data over OPC UA from Ignition’s OPC UA server, you must accept the certificate presented by the SiteWise connector within Ignition.
   1. Get the private IP address of the physical hardware, and load a URL like this into your browser of choice: http://<hardwarePrivateIP>:8088
      1. Reminder: For AWS Employees, do not be on the corporate VPN for this step.
   2. Once the Ignition Web UI is open, you should see a gear like icon on the left labeled 'Config'. Click that, and it may ask you to log in. The default credentials are:
      1. Username: admin
      2. Password: password
   3. Navigate to "OPC UA -> Security -> Server" and wait for the quarantined certificate to appear (from AWS IoT SiteWise Gateway). You should see a single entry under 'Quarantined Certificates' named something like 'AWS IoT SiteWise Gateway Client'.
      1. Click “Trust” to accept the certificate. At this point, the SiteWise connector will start consuming data over OPC UA from Ignition and this data will be sent up to the AWS IoT SiteWise in the cloud.
5. Update the SiteWise Gateway
   1. Navigate to AWS IoT SiteWise console and select Ingest🡪Gateways
   2. Select the gateway created during the stack launch.
      1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
   3. Click “Edit” in the Source Configuration for Automated Gateway Config” section
   4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
6. Validate Incoming PLC Data
   1. Now that you've trusted the certificate, go back to the AWS IoT SiteWise console.
   2. In the SiteWise console, click the icon on the left side of the page, select build🡪assets
   3. In the asset tree on the left, drill down to an asset (i.e. Hauloff or Conveyor), select it and then select “Measurements” tab for that asset.
   4. Verify that the values in the “Latest value” column are updating. This indicates that the Ignition simulation of those virtual devices and sensors is properly sending data through to the SiteWise connector in Greengrass and up to AWS IoT SiteWise in the AWS cloud.

### View SiteWise Portal Data

**Log in to SiteWise Monitor Portal**

1. For a more visual display of the data, navigate to the SiteWise console, select the icon on the left and select Monitor🡪 Portals.
2. Select the hyperlinked "name" of the Portal most recently added (the topmost on the list).
3. Add yourself as an administrator of the Portal by clicking “Assign Users” in the Portal Administrators section
4. Once you are listed as a Portal Administrator, click the hyperlinked URL in the Portal details section under the “URL” column. This URL should have the format <https://[XXXXX....XXXXXX].app.iotsitewise.aws>.
5. Log in with the credentials (username and password) you just created for your administrator account.

**View Data in SiteWise Monitor Portal**

1. Select “Dashboards” tab on the left-hand side, then select the newly created dashboard hyperlink under the “Name” column of the Dashboards page.
   1. Data should be flowing into the line charts for the asset measurement properties
2. You can also see data for individual assets by navigating to the “Asset Library” tab on the left and selecting an asset from the asset tree. Once an asset is selected, you can view its properties.

### Troubleshooting

**Quarantined certificate in Ignition doesn't show up, or data doesn’t show up for Option 1 deployments**

First, verify that the Ignition trial period (2 hours) has not expired. If that action does not remediate the issue, repeat the process of refreshing the SiteWise Gateway:

1. Navigate to the AWS IoT SiteWise console and select Ingest 🡪 Gateways
2. Select the gateway created during the stack launch:
   1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
3. Click “Edit” in the Source Configuration for Automated Gateway Config section
4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
5. If it hasn’t already been done, look for and accept the quarantined certificate in Ignition.

## Physical Greenfield Option 1 Ignition File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:
  1. Template Source: Leave as default - “Amazon S3 URL”
  2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
     1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 1”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Leave as default “IgnitionFileExport”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

**Option 1**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptGreenfieldOption1] physical-greenfield-option1.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-option1.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsGreenfield1and2a]
   6. Replace [HardwareIP] with the physical device’s private IP address
   7. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-greenfield-option1.sh physicalgreenfieldoption1-devicesbucketresource-1ifk8w3a77621 PhysicalGreenfieldOption1/PhysicalGreenfieldOption1Core.tar.gz 866prvj43g us-east-1 PhysicalGreenfieldOption1 [Hardware-IP] PhysicalGreenfieldOption1/PhysicalGreenfieldOption1Device.tar.gz [a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com](http://a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com/)
2. Represent your data in Ignition by creating a project tag hierarchy. The source of this data can be physical PLCs, or simulated devices in Ignition.
   1. Download the Ignition Designer software
      1. In your browser of choice, visit the following URL: http://<hardwarePrivateIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      2. On the top right corner of the screen, click “Get Designer”
      3. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: http://<reachableIgnitionIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. With the help of an OT professional or IMC Kit contact, represent your PLC data (simulated or real) in a hierarchy
3. Export the JSON file that describes your project hierarchy in Ignition.
   1. Follow the steps in 2.b to connect to your instance of Ignition via the Ignition Designer
   2. A screenshot of a cell phone

      Description automatically generatedIn the Tag Browser, under “Tag Providers” select “default” and click export.
4. Upload the JSON file from step 1 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. Upon dropping the JSON file in this S3 bucket, an S3 event trigger will automatically invoke the AMC Lambda function
   2. Wait for the AMC to complete creating your models and assets in SiteWise
5. Accept SiteWise Certificate in Ignition: To enable the SiteWise to ingest data over OPC UA from Ignition’s OPC UA server, you must accept the certificate presented by the SiteWise connector within Ignition.
   1. Get the private IP address of the physical hardware, and load a URL like this into your browser of choice: http://<hardwarePrivateIP>:8088
      1. Reminder: For AWS Employees, do not be on the corporate VPN for this step.
   2. Once the Ignition Web UI is open, you should see a gear like icon on the left labeled 'Config'. Click that, and it may ask you to log in. The default credentials are:
      1. Username: admin
      2. Password: password
   3. Navigate to "OPC UA -> Security -> Server" and wait for the quarantined certificate to appear (from AWS IoT SiteWise Gateway). You should see a single entry under 'Quarantined Certificates' named something like 'AWS IoT SiteWise Gateway Client'.
      1. Click “Trust” to accept the certificate. At this point, the SiteWise connector will start consuming data over OPC UA from Ignition and this data will be sent up to the AWS IoT SiteWise in the cloud.
6. Update the SiteWise Gateway
   1. Navigate to AWS IoT SiteWise console and select Ingest🡪Gateways
   2. Select the gateway created during the stack launch.
      1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
   3. Click “Edit” in the Source Configuration for Automated Gateway Config” section
   4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
7. Validate Incoming PLC Data
   1. Now that you've trusted the certificate, go back to the AWS IoT SiteWise console.
   2. In the SiteWise console, click the icon on the left side of the page, select build🡪assets
   3. In the asset tree on the left, drill down to an asset (i.e. Hauloff or Conveyor), select it and then select “Measurements” tab for that asset.
   4. Verify that the values in the “Latest value” column are updating. This indicates that the Ignition simulation of those virtual devices and sensors is properly sending data through to the SiteWise connector in Greengrass and up to AWS IoT SiteWise in the AWS cloud.

### View SiteWise Portal Data

**Log in to SiteWise Monitor Portal**

1. For a more visual display of the data, navigate to the SiteWise console, select the icon on the left and select Monitor🡪 Portals.
2. Select the hyperlinked "name" of the Portal most recently added (the topmost on the list).
3. Add yourself as an administrator of the Portal by clicking “Assign Users” in the Portal Administrators section
4. Once you are listed as a Portal Administrator, click the hyperlinked URL in the Portal details section under the “URL” column. This URL should have the format <https://[XXXXX....XXXXXX].app.iotsitewise.aws>.
5. Log in with the credentials (username and password) you just created for your administrator account.

**View Data in SiteWise Monitor Portal**

1. Select “Dashboards” tab on the left-hand side, then select the newly created dashboard hyperlink under the “Name” column of the Dashboards page.
   1. Data should be flowing into the line charts for the asset measurement properties
2. You can also see data for individual assets by navigating to the “Asset Library” tab on the left and selecting an asset from the asset tree. Once an asset is selected, you can view its properties.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Quarantined certificate in Ignition doesn't show up, or data doesn’t show up for Option 1 deployments**

First, verify that the Ignition trial period (2 hours) has not expired. If that action does not remediate the issue, repeat the process of refreshing the SiteWise Gateway:

1. Navigate to the AWS IoT SiteWise console and select Ingest 🡪 Gateways
2. Select the gateway created during the stack launch:
   1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
3. Click “Edit” in the Source Configuration for Automated Gateway Config section
4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
5. If it hasn’t already been done, look for and accept the quarantined certificate in Ignition.

## Physical Greenfield Option 2a Cirrus Link Module

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:
  1. Template Source: Leave as default - “Amazon S3 URL”
  2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
     1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 2a”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Leave as default “CirrusLinkModule”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2a deployments, models and assets are not required to collect data.

**Option 2a**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptGreenfieldOption2a] physical-greenfield-option2a.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-option2a.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsGreenfield1and2a]
   6. Replace [HardwareIP] with the physical device’s private IP address
   7. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-greenfield-option2a.sh physicalgreenfieldoption1-devicesbucketresource-1ifk8w3a77621 PhysicalGreenfieldOption2a/PhysicalGreenfieldOption2aCore.tar.gz 866prvj43g us-east-1 PhysicalGreenfieldOption2a [Hardware-IP] PhysicalGreenfieldOption2a/PhysicalGreenfieldOption1Device.tar.gz [a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com](http://a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com/)
2. Represent your data in Ignition by creating a project tag hierarchy. The source of this data can be physical PLCs, or simulated devices in Ignition.
   1. Download the Ignition Designer software
      1. In your browser of choice, visit the following URL: http://<hardwarePrivateIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      2. On the top right corner of the screen, click “Get Designer”
      3. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: http://<reachableIgnitionIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. With the help of an OT professional or IMC Kit contact, represent your PLC data (simulated or real) in a hierarchy
3. Trigger an MQTT node “birth” message
   1. Note for AWS/SME: a birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
   2. This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
      1. \*\* Not required to collect data for this deployment option
4. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
5. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<hardwarePrivateIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
2. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
3. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[your\_aws\_account\_iot\_endpoint]:8883
   2. Download the .tar.gz file that represents the non-GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Replace the CA Certificate File with “root.ca.pem” from the tarball package
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Private Key File with the “.private” file from the tarball package
   7. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Physical Greenfield Option 2a Ignition File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:
  1. Template Source: Leave as default - “Amazon S3 URL”
  2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
     1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 2a”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Leave as default “IgnitionFileExport”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2a deployments, models and assets are not required to collect data.

**Option 2a**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptGreenfieldOption2a] physical-greenfield-option2a.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-option2a.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsGreenfield1and2a]
   6. Replace [HardwareIP] with the physical device’s private IP address
   7. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-greenfield-option2a.sh physicalgreenfieldoption1-devicesbucketresource-1ifk8w3a77621 PhysicalGreenfieldOption2a/PhysicalGreenfieldOption2aCore.tar.gz 866prvj43g us-east-1 PhysicalGreenfieldOption2a [Hardware-IP] PhysicalGreenfieldOption2a/PhysicalGreenfieldOption1Device.tar.gz [a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com](http://a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com/)
2. Represent your data in Ignition by creating a project tag hierarchy. The source of this data can be physical PLCs, or simulated devices in Ignition.
   1. Download the Ignition Designer software
      1. In your browser of choice, visit the following URL: http://<hardwarePrivateIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      2. On the top right corner of the screen, click “Get Designer”
      3. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: http://<reachableIgnitionIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. With the help of an OT professional or IMC Kit contact, represent your PLC data (simulated or real) in a hierarchy
3. Export the JSON file that describes your project hierarchy in Ignition.
   1. Download the Ignition Designer software
      1. Navigate to the EC2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC Kit
         1. The end of the EC2’s instance should contain “Ignition”
      2. In your browser of choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
      3. On the top right corner of the screen, click “Get Designer”
      4. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. A screenshot of a cell phone

      Description automatically generatedIn the Tag Browser, under “Tag Providers” select “default” and click export.
4. Upload the JSON file from step 1 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
         1. \*\* Not required to collect data for this deployment option
   2. Wait for the AMC to complete creating your models and assets in SiteWise
5. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
6. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<hardwarePrivateIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
2. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
3. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[your\_aws\_account\_iot\_endpoint]:8883
   2. Download the .tar.gz file that represents the non-GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Replace the CA Certificate File with “root.ca.pem” from the tarball package
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Private Key File with the “.private” file from the tarball package
   7. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Physical Greenfield Option 2b Cirrus Link Module

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:

1. Template Source: Leave as default - “Amazon S3 URL”
2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
   * 1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml

* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 2b”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Leave as default “CirrusLinkModule”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2b deployments, models and assets are not required to collect data.

**Option 2b**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptGreenfieldOption2b] physical-greenfield-option2b.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-option2b.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsGreenfield2b]
   6. Replace [HardwareIP] with the physical device’s private IP address
   7. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-greenfield-option2b.sh physicalgreenfieldoption2b-devicesbucketresource-1ifk8w3a77621 PhysicalGreenfieldOption2b/PhysicalGreenfieldOption2bCore.tar.gz 866prvj43g us-east-1 PhysicalGreenfieldOption2b [Hardware-IP] PhysicalGreenfieldOption2b/PhysicalGreenfieldOption2bDevice.tar.gz [a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com](http://a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com/)
2. Represent your data in Ignition by creating a project tag hierarchy. The source of this data can be physical PLCs, or simulated devices in Ignition.
   1. Download the Ignition Designer software
      1. In your browser of choice, visit the following URL: http://<hardwarePrivateIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      2. On the top right corner of the screen, click “Get Designer”
      3. Install the Ignition Designer for your operating system
   2. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: http://<reachableIgnitionIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   3. With the help of an OT professional or IMC Kit contact, represent your PLC data (simulated or real) in a hierarchy
3. Trigger a node “birth” message
   1. Using the steps in 2.b, connect to your instance of Ignition
   2. A screenshot of a cell phone

      Description automatically generatedA birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
   3. This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
      1. \*\* Not required to collect data for this deployment option
4. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
5. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<hardwarePrivateIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
2. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
3. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[greengrass\_ec2\_private\_ip\_address]:8883
   2. Download the .tar.gz file that represents the GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Retrieve the GreenGrass group CA certificate by running the following command, replacing the sections in red with their correstponding CloudFormation template outputs found in the “NESTED” stack:
      1. wget -O root.ca.pem <https://$RESTAPIID.execute-api.$REGION.amazonaws.com/api/deploygg/$EDGEDEVICEID>
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Certificate File with the “.private” file from the tarball package
   7. Replace the CA Certificate File with the “root.ca.pem” file retrieved from step d.
   8. Check the “show advanced properties” box
   9. Make sure the client ID reads: [name\_for\_edge\_device\_parameter]Device
4. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Physical Greenfield Option 2b Ignition File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:

1. Template Source: Leave as default - “Amazon S3 URL”
2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
   * 1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml

* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 2b”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Leave as default “IgnitionFileExport”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

\*\*Note – for Option 2b deployments, models and assets are not required to collect data.

**Option 2b**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptGreenfieldOption2b] physical-greenfield-option2b.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-option2b.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsGreenfield2b]
   6. Replace [HardwareIP] with the physical device’s private IP address
   7. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-greenfield-option2b.sh physicalgreenfieldoption2b-devicesbucketresource-1ifk8w3a77621 PhysicalGreenfieldOption2b/PhysicalGreenfieldOption2bCore.tar.gz 866prvj43g us-east-1 PhysicalGreenfieldOption2b [Hardware-IP] PhysicalGreenfieldOption2b/PhysicalGreenfieldOption2bDevice.tar.gz [a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com](http://a1zs31l2dkuqc-ats.iot.us-east-1.amazonaws.com/)
2. Download the Ignition Designer software
   1. Navigate to the Ec2 console and retrieve the public IP address of the EC2 instance running Ignition for the IMC kit
      1. The end of the Ec2’s instance should contain “Ignition”
   2. In your browser choice, visit the following URL: http://<ignition\_ec2\_IP>:8088
      1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
   3. On the top corner of the screen, click “Get Designer”
   4. Install the Ignition Designer for your operating system
3. Represent your data in Ignition by creating a project tag hierarchy. The source of this data can be physical PLCs, or simulated devices in Ignition.
   1. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: http://<reachableIgnitionIP>:8088
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   2. With the help of an OT professional or IMC Kit contact, represent your PLC data (simulated or real) in a hierarchy
4. Export the JSON file that describes your project hierarchy in Ignition.
   1. Open the Ignition Designer and connect to your instance of Ignition
      1. Launch the Designer
      2. Click “Add Designer”
      3. Click “Manually Add Gateway”
      4. Add a Gateway URL in the following format: <http://[ignition_ec2_public_ip]:8088>
         1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
      5. Under the Gateway tile you just added, click “Launch”
      6. Supply the username and password and click “Login”
         1. Username: admin
         2. Password: password
   2. A screenshot of a cell phone

      Description automatically generatedIn the Tag Browser, under “Tag Providers” select “default” and click export.
5. Upload the JSON file from step 1 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. Upon dropping the JSON file in this S3 bucket, an S3 event trigger will automatically invoke the AMC Lambda function
   2. Wait for the AMC to complete creating your models and assets in SiteWise
6. Validate Incoming PLC Data in IoT Core
   1. Navigate to the AWS IoT Core console.
   2. Select “Test” from the navbar on the left.
   3. Subscribe to the topic: “spBv1.0/AWS Smart Factory/DDATA/#”
   4. Verify that messages are coming in from this topic.
7. Validate Incoming PLC Data in S3
   1. Navigate to the S3 console.
   2. Search for the bucket: “[stack\_name\_here]-imcs3bucket-[hash]
   3. Click into the bucket and confirm that a directory inside the bucket called “mqtt” exists.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Data via the MQTT Transmission module doesn’t show up in IoT Cloud**

1. Get the public IP address of that instance, and load a URL like this into your browser of choice: http://<hardwarePrivateIP>:8088
   1. Reminder: For AWS Employees, do not be on the corporate VPN for the step.
2. Open the Ignition Web UI is open, you should see a gear like icon on the left labeled ‘Config’. Click that, and it will ask you to log in. The default credentials are:
   1. Username: admin
   2. Password: password
3. Navigate to “MQTT Transmission -> Settings -> Server” and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. Make sure the URL is in the format: ssl://[greengrass\_ec2\_private\_ip\_address]:8883
   2. Download the .tar.gz file that represents the GreenGrass IoT thing from the following S3 bucket location:
      1. Bucket Name: [stack\_name]-devicesbucketresource-hash
      2. Key Name: [name\_for\_edge\_device\_parameter]/[name\_for\_edge\_device\_parameter]Device.tar.gz
   3. Expand the tarball
   4. Retrieve the GreenGrass group CA certificate by running the following command, replacing the sections in red with their correstponding CloudFormation template outputs found in the “NESTED” stack:
      1. wget -O root.ca.pem <https://$RESTAPIID.execute-api.$REGION.amazonaws.com/api/deploygg/$EDGEDEVICEID>
   5. Replace the Client Certificate File with the “.pem” file from the tarball package
   6. Replace the Client Certificate File with the “.private” file from the tarball package
   7. Replace the CA Certificate File with the “root.ca.pem” file retrieved from step d.
   8. Check the “show advanced properties” box
   9. Make sure the client ID reads: [name\_for\_edge\_device\_parameter]Device
4. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

## Physical Greenfield Cleanup

**Follow these steps to clean up the IMC CloudFormation stack deployment:**

Cloud

1. Navigate to the CloudFormation console and delete the base stack (not the stack named "NESTED"), in order to clean up the account as much as possible. Most of the resources will be deleted, but the stack deletion will fail due to non-empty S3 buckets and potentially a deployed Greengrass group (for all Virtual options by default, and for all Physical deployments that have been completed on a piece of hardware. The steps required to delete a stack are:
   1. Empty the S3 buckets:
      1. Navigate to the S3 service in the AWS Console.
      2. In the search bar, enter your stack name.
      3. For each bucket that is associated with the stack (naming convention: [name\_of\_stack]-[bucket\_identifier]-unique hash), select the bucket, and click “Empty” under the search bar.
      4. Here are the following “bucket identifiers” that exist for each deployment:
         1. amcincomingresource
         2. amcoutputresource
         3. devicesbucketresource
         4. imcs3bucket
         5. lambdazipsbucket
   2. Force a reset of the GreenGrass group:
      1. Navigate to the GreenGrass console
      2. Select the GreenGrass group with the “Name for the Edge device” parameter name provided to the stack
      3. Under “Actions”, select “Reset Deployments”
      4. Check the box that asks if you want to force the reset
      5. Click “Reset Deployment”
   3. Navigate back to the CloudFormation console and once again delete the base stack.
      1. There is a master and nested stack. you must delete the master stack (the one that does **not** have “NESTED” in a gray box associated with it.
2. Other resources to clean up after stack deletion (if desired, for cleanliness): SiteWise Portal, SiteWise Gateway, SiteWise Models and Assets, QuickSight dataset.

Edge Hardware

1. Navigate to a terminal on the edge hardware. Become the root user using the “sudo su” command.
2. Stop and remove Ignition from hardware:
   1. cd /[path\_to\_Ignition\_download]/Ignition-AWS-Kit-MQTT-v4
   2. ./remove.sh
   3. cd ..
   4. rm device.tar.gz group.tar.gz opcclient.der Ignition-AWS-Kit-MQTT-v4.zip physical-greenfield-option<insert\_option\_here>.sh
   5. rm -rf Ignition-AWS-Kit-MQTT-v4
3. Stop and remove GreenGrass:
   1. apt remove aws-iot-greengrass-core
   2. rm -rf /greengrass
   3. rm -rf /var/sitewise

## Physical Greenfield FAQs

**Can I update a stack to a different deployment type (Physical, Virtual) or dataflow option (Option1, 2a, 2b)?**

* Updates are currently not supported. To achieve a different deployment type or dataflow type, you'll need to [deploy a new stack.](#_Physical_–_Greenfield) See the details on having more than one stack deployed in an account below, and then decide whether or not you’ll want to [delete your original stack](#_Physical_Greenfield_Cleanup_1) before re-creating another.

**Can I deploy multiple times in the same AWS account?**

Yes, you may deploy multiple stacks in the same account – but be wary of the following:

* For Option 1 deployments, SiteWise may receive data from multiple sources if there are, for example, two instances of Ignition that are publishing data onto the topic “/Tag Providers/default/Line1/CNC/Temperature”.
* If SiteWise models and assets exist in the Cloud from a previous deployment, a new deployment will not re-create the model/asset hierarchy. Instead, the Asset Model Converter operates by recognizing any “deltas” in the existing models/assets and the new deployment hierarchy. For example:
  + If the previously deployed hierarchy is identical to the newly deployed, nothing in SiteWise is changed.
  + If the newly deployed hierarchy has an identical structure to the previously deployed hierarchy **with additional nodes that follow the hierarchy pattern,** the new models/assets will be created in SiteWise**:**
    - Previous:
      * Line 1/CNC1/Temperature
    - New:
      * Line 1/CNC1/Temperature
      * Line 1/CNC2/Temperature
* If the newly deployed hierarchy has a different hierarchy than the previously deployed hierarchy altogether, the Asset Model Converter will not succeed in creating the new models and assets.

**Does any hardening of the OS occur during deployment?**

Yes. Fail2ban is installed and enabled on the device, which bans IPs that show the following malicious signs:

* Too many password failures
* Seeking for exploits

Additionally, shared memory on the device is secured via /etc/fstab.

**Can I re-run the AMC to create new models and assets?**

Yes. Follow these steps:

1. If any of the models/assets you want to create share a name with the models/assets created in the first pass of the AMC execution, you’ll need to delete the models and assets in SiteWise.
2. The second action you’ll need to take is clearing out the following DynamoDB tables associated with the initial IMC kit deployment:
   1. [stack-name]-asset-model-table
   2. [stack-name]-asset-table

Depending on your driver of choice, you’ll need to take the follow actions to re-run the AMC:

* IgnitionCirrusLink
  + Initiate a re-birth (NBIRTH,DBIRTH) MQTT message that represents the your project hierarchy.
    - Connect to your instance of Ignition in the Ignition Designer
    - A screenshot of a cell phone

      Description automatically generatedA birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
    - This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
* IgnitionFileExport
  + Obtain your JSON file that represents the new hierarchy and drop it into the following S3 bucket:
    - [name\_of\_stack]-[amcincomingresource]-[hash]
* KepServerFileExport
  + Obtain your JSON file that represents the new hierarchy and drop it into the following S3 bucket:
    - [name\_of\_stack]-[amcincomingresource]-[hash]

# Physical-Brownfield

The physical brownfield deployment is intended to demonstrate the capabilities of the IMC kit in an environment where the end user has an existing edge-based asset modeling software (such as Ignition or KepServer). It is deployed onto physical hardware. After deployment, the physical hardware will run GreenGrass software and connect into the edge-based asset modeling software. The IMC kit supports the following OEM devices:

1. **Lenovo**
   1. Model: ThinkCentre M90n IoT
   2. Architecture: Intel® Celeron® 4205U (x86)
   3. URL: <https://www.lenovo.com/us/en/desktops-and-all-in-ones/thinkcentre/m-series-tiny/ThinkCentre-M90n-IoT/p/thinkcentre-m90n-iot>
2. **ADLINK** 
   1. Model: MXE-211
   2. Architecture: Intel Atom® Processor E3900 (x86)
   3. <https://www.adlinktech.com/Products/Industrial_IoT_and_Cloud_solutions/IoTGateway/MXE-210_Series?lang=en>
3. **OnLogic**
   1. Model: Karbon 300 Compact Rugged Computer
   2. Architecture: Intel Atom® E3930 or E3950 processors
   3. URL: <https://onlogic.com/k300/>
4. **Advantech** 
   1. Model: UNO-2372G
   2. Architecture: Intel Atom E3845/Celeron® J1900 Quad-Core Processors
   3. URL: <https://www.advantech.com/products/1-2mlj9a/uno-2372g/mod_f4ff5680-f016-44bd-bff0-e5eddfd82237>
5. **MOXA**
   1. Model: MC-1112-E4-T
   2. Architecture: Intel Atom® Processor E3845 processor
   3. URL: <https://www.moxa.com/en/products/industrial-computing/x86-computers/mc-1100-series/mc-1121-e4-t>

This deployment mode does not come with a configured set of project tags similar to the virtual deployment but does come with a set of device simulations that can be configured to represent a project tag structure similar to the virtual deployment tag structure (or your own structure entirely). This deployment can be configured to work with a physical PLC test harness.

The virtual deployment has the following sections:

1. Pre-Requisites
2. CloudFormation stack launch
3. Post deployment steps
4. Troubleshooting
5. Physical Brownfield Cleanup
6. Physical Brownfield FAQ

## Pre-Requisites

* **AWS account with SSO enabled:** <https://docs.aws.amazon.com/singlesignon/latest/userguide/getting-started.html>
* **EC2 Key Pair:** <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>
* **Create IoT SiteWise Service-linked Role Using the AWS CLI:** 
  + aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com --description "Service-linked role to support IoT SiteWise"
  + <https://docs.aws.amazon.com/iot-sitewise/latest/userguide/using-service-linked-roles.html>
* **Connect to your physical hardware running Ubuntu 18.04, make sure it is connected to the internet, and configure the AWS CLI:**
  + You may achieve this however you please – either via an SSH command or a connection to the device with a keyboard and monitor
  + Ensure you have the ability to connect to the internet
  + Configure the AWS CLI on the hardware to communicate with the AWS account you plan to deploy in: <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>
* **The directory structure of the physical hardware running Ubuntu 18.04 must look like the following:**

home/

ubuntu

* **QuickStart Bucket Preparation:**
  + Create an S3 bucket and give it a unique name such as “imc-quickstart-bucket-ABC-123”. In that S3 bucket create a folder called “quickstart-IMC”. You then copy all the IMC CloudFormation [artifacts](#_Artifacts_1) into the “quickstart-IMC” folder. The structure will then resemble the structure below:

S3 bucket name: “imc-quickstart-bucket-ABC-123”

S3 bucket Contents:

quickstart-IMC/

functions/

scripts/

templates/

LICENSE.txt

NOTICE.txt

README.md

## Physical Brownfield Option 1 Ignition/KepServer File Export

Open up the AWS Console and navigate to the CloudFormation console. Create a new stack and select “with new resources (standard)”).

### CloudFormation stack launch

**Step 1: Specify templates**

* Prerequisite - Prepare template: Leave as default - “Template is ready”
* Specify Template:
  1. Template Source: Leave as default - “Amazon S3 URL”
  2. Amazon S3 URL: Use the “IMC-workload.template.yaml” CloudFormation workload template that's in the S3 bucket we just created (i.e. “imc-quickstart-bucket-ABC-123”)
     1. The URL should look like this: https://<BUCKETNAME>.s3.amazonaws.com/quickstart-IMC/templates/IMC-workload.template.yaml
* Click Next to proceed to Step 2 of the CloudFormation stack launch (Specify stack details)

**Step 2: Specify stack details**

* Stack Name:
  + Stack Name: Give the stack a unique name such as “IMC-Physical”
* Parameters:
  + Edge Deployment Configuration
    - Name for the edge device**:** You may leave as default or, If desired, you may specify a new name for the edge device. This name will be the name of the Greengrass group that gets created with this stack.
    - Type of Deployment (Virtual or Physical):Physical
    - Deployment Flow: Select “Option 1”
  + Amazon EC2 Configuration
    - SSH Key Name: For physical deployments, the EC2 Key Pair won’t be used, but one needs to be selected for the stack to deploy properly. If you haven’t, navigate to the EC2 service in the AWS console and create an EC2 key pair. Select the key pair when deploying the stack.
    - VPC ID: Select the VPC to launch this stack in. You can select your account’s default VPC.
    - Greengrass EC2 Instance Type**:** (default: t3.small) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - Ignition EC2 Instance Types: (default: t3.large) For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Leave as the default value.
    - EC2 AMI: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during creation of the stack. Select “ami-085925f297f89fce1” from the dropdown menu.
    - EC2 Subnet: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Fill this field in with any non-blank string value.
  + AWS Quick Start Configuration
    - QuickStart S3 Bucket Name: Use the name of the bucket you created previously in the Pre-Requisites section. We used the example S3 bucket name of “imc-quickstart-bucket-ABC-123”
    - QuickStart S3 Key Prefix: (default: “quickstart-IMC/”) Use the name of the root folder in the S3 bucket you created. In the Pre-Requisites section, we named the folder “quickstart-IMC/”
    - QuickStart S3 Bucket Region: Leave as default “us-east-1”
    - Select the Asset Model Converter (AMC) Driver: Select “IgnitionFileExport” or “KepwareFileExport”.
    - User Public IP Address: For physical deployments, an EC2 instance won’t be created, but this field must be filled in during the creation of the stack. Input any IP address in the format “x.x.x.x”.
  + Click “Next” to proceed to Step 3 “Configure stack options”

**Step 3: Configure stack options**

* You can accept all defaults
* Click “Next” to proceed to Step 4 “Review”

**Step 4: Review**

* Review and accept the acknowledgements at the bottom of the page
* Click “Create stack” to launch the CloudFormation stack.

**Stack Deployment**

Stack deployment will take approximately 5-10 minutes. You can track the progress of the stack launch by viewing the “Events” tab of the stack.

### Post Deployment Steps

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational.

**Option 1**

1. Retrieve and run the bootup script for the physical hardware device
   1. Open a terminal on the physical hardware
   2. Use the command line to become the root user in your terminal session:
      1. sudo su
   3. Use the command line to retrieve the deployment script from your stack’s S3 bucket, replacing the values in brackets with the corresponding output values from your CloudFormation stack (found on the AWS CloudFormation console, in the stack labeled “NESTED” in the output tab):
      1. aws s3api get-object --bucket [DependenciesBucket] --key [BootupScriptBrownfieldAllOptions] physical-brownfield-all-options.sh
      2. \*\*You must have the AWS CLI configured to point to the AWS account you’re using for the IMC kit.
   4. Use the command line to make the file executable:
      1. chmod +x physical-greenfield-all-options.sh
   5. Retrieve (copy to your clipboard) the command from the “NESTED” CloudFormation output: [FullScriptParamsBrownField]
   6. Use the command line to run the deployment script, which should resemble something like the following (but filled in with your stack-specific values):
      1. ./physical-brownfield-all-options.sh physicalbrownfieldoption1-devicesbucketresource-1ifk8w3a77621 PhysicalBrownfieldOption1/PhysicalBrownfieldOption1Core.tar.gz 866prvj43g us-east-1 PhysicalBrownfieldOption1
2. Ensure that your instance of edge-based asset modeling software is set up with the desired hierarchy of assets.
3. Export the JSON file that describes your project hierarchy in either Ignition or KepServer.
4. Upload the JSON file from step 3 into the following S3 bucket created during deployment to trigger the AMC and creation of models and assets in SiteWise:
   1. [name\_of\_stack]-[amcincomingresource]-[hash]
      1. Upon dropping the JSON file in this S3 bucket, an S3 event trigger will automatically invoke the AMC Lambda function
   2. Wait for the AMC to complete creating your models and assets in SiteWise
5. Update the SiteWise Gateway
   1. Navigate to AWS IoT SiteWise console and select Ingest🡪Gateways
   2. Select the gateway created during the stack launch.
      1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
   3. Edit the "Source Configuration for Automated Gateway Config” section to point to the edge-based asset modeling software.
      1. Ignition format: opc.tcp://<reachable\_IP>:62541
      2. KepServer format: opc.tcp://<reachable\_IP>:49320
   4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.
6. Accept SiteWise Certificate in your edge-based asset modeling software: To enable the SiteWise to ingest data over OPC UA from the OPC UA server, you must accept the certificate presented by the SiteWise connector.
7. Validate Incoming PLC Data
   1. Now that you've trusted the certificate, go back to the AWS IoT SiteWise console.
   2. In the SiteWise console, click the icon on the left side of the page, select build🡪assets
   3. In the asset tree on the left, drill down to an asset (i.e. Hauloff or Conveyor), select it and then select “Measurements” tab for that asset.
   4. Verify that the values in the “Latest value” column are updating. This indicates that the Ignition simulation of those virtual devices and sensors is properly sending data through to the SiteWise connector in Greengrass and up to AWS IoT SiteWise in the AWS cloud.

### View SiteWise Portal Data

**Log in to SiteWise Monitor Portal**

1. For a more visual display of the data, navigate to the SiteWise console, select the icon on the left and select Monitor🡪 Portals.
2. Select the hyperlinked "name" of the Portal most recently added (the topmost on the list).
3. Add yourself as an administrator of the Portal by clicking “Assign Users” in the Portal Administrators section
4. Once you are listed as a Portal Administrator, click the hyperlinked URL in the Portal details section under the “URL” column. This URL should have the format <https://[XXXXX....XXXXXX].app.iotsitewise.aws>.
5. Log in with the credentials (username and password) you just created for your administrator account.

**View Data in SiteWise Monitor Portal**

1. Select “Dashboards” tab on the left-hand side, then select the newly created dashboard hyperlink under the “Name” column of the Dashboards page.
   1. Data should be flowing into the line charts for the asset measurement properties
2. You can also see data for individual assets by navigating to the “Asset Library” tab on the left and selecting an asset from the asset tree. Once an asset is selected, you can view its properties.

### Troubleshooting

**Models and assets weren’t created in SiteWise**

Check the Lambda function responsible for creating the models and assets in SiteWise for errors:

1. In the AWS lambda console, navigate to the function named [name\_of\_stack]-AssetModelIngestionLambdaResource-[hash]
2. Hit the “Monitoring” tab
3. Click “View logs in CloudWatch”
4. Click into the most recent Log Stream and find the error message

**Quarantined certificate in Ignition/KepServer doesn't show up, or data doesn’t show up for Option 1 deployments**

If using Ignition, verify that the Ignition trial period (2 hours) has not expired. If that action does not remediate the issue, repeat the process of refreshing the SiteWise Gateway:

1. Navigate to the AWS IoT SiteWise console and select Ingest 🡪 Gateways
2. Select the gateway created during the stack launch:
   1. Naming convention: [name\_of\_stack]\_Automated\_Gateway
3. Click “Edit” in the Source Configuration for Automated Gateway Config section
4. Click “Save” at the bottom. No changes are necessary. This action simply activates the SiteWise gateway to ensure data flows from the OPC UA server.

If it hasn’t already been done, look for and accept the quarantined certificate in Ignition

If using KepServer for Windows, make sure that your default firewalls have been turned off (they prevent the SiteWise Gateway certificates from showing up).

## Physical Brownfield Cleanup

**Follow these steps to clean up the IMC CloudFormation stack deployment:**

Cloud

1. Navigate to the CloudFormation console and delete the base stack (not the stack named "NESTED"), in order to clean up the account as much as possible. Most of the resources will be deleted, but the stack deletion will fail due to non-empty S3 buckets and potentially a deployed Greengrass group (for all Virtual options by default, and for all Physical deployments that have been completed on a piece of hardware. The steps required to delete a stack are:
   1. Empty the S3 buckets:
      1. Navigate to the S3 service in the AWS Console.
      2. In the search bar, enter your stack name.
      3. For each bucket that is associated with the stack (naming convention: [name\_of\_stack]-[bucket\_identifier]-unique hash), select the bucket, and click “Empty” under the search bar.
      4. Here are the following “bucket identifiers” that exist for each deployment:
         1. amcincomingresource
         2. amcoutputresource
         3. devicesbucketresource
         4. imcs3bucket
         5. lambdazipsbucket
   2. Force a reset of the GreenGrass group:
      1. Navigate to the GreenGrass console
      2. Select the GreenGrass group with the “Name for the Edge device” parameter name provided to the stack
      3. Under “Actions”, select “Reset Deployments”
      4. Check the box that asks if you want to force the reset
      5. Click “Reset Deployment”
   3. Navigate back to the CloudFormation console and once again delete the base stack.
      1. There is a master and nested stack. you must delete the master stack (the one that does **not** have “NESTED” in a gray box associated with it.
2. Other resources to clean up after stack deletion (if desired, for cleanliness): SiteWise Portal, SiteWise Gateway, SiteWise Models and Assets, QuickSight dataset.

Edge Hardware

1. Navigate to a terminal on the edge hardware. Become the root user using the “sudo su” command.
2. Stop and remove GreenGrass:
   1. apt remove aws-iot-greengrass-core
   2. rm -rf /greengrass
   3. rm -rf /var/sitewise

## Physical Brownfield FAQs

**Can I update a stack to a different deployment type (Physical, Virtual) or dataflow option (Option1, 2a, 2b)?**

* Updates are currently not supported. To achieve a different deployment type or dataflow type, you'll need to [deploy a new stack.](#_Physical_–_Greenfield) See the details on having more than one stack deployed in an account below, and then decide whether or not you’ll want to [delete your original stack](#_Physical_Greenfield_Cleanup_1) before re-creating another.

**Can I deploy multiple times in the same AWS account?**

Yes, you may deploy multiple stacks in the same account – but be wary of the following:

* For Option 1 deployments, SiteWise may receive data from multiple sources if there are, for example, two instances of Ignition that are publishing data onto the topic “/Tag Providers/default/Line1/CNC/Temperature”.
* If SiteWise models and assets exist in the Cloud from a previous deployment, a new deployment will not re-create the model/asset hierarchy. Instead, the Asset Model Converter operates by recognizing any “deltas” in the existing models/assets and the new deployment hierarchy. For example:
  + If the previously deployed hierarchy is identical to the newly deployed, nothing in SiteWise is changed.
  + If the newly deployed hierarchy has an identical structure to the previously deployed hierarchy **with additional nodes that follow the hierarchy pattern,** the new models/assets will be created in SiteWise**:**
    - Previous:
      * Line 1/CNC1/Temperature
    - New:
      * Line 1/CNC1/Temperature
      * Line 1/CNC2/Temperature
* If the newly deployed hierarchy has a different hierarchy than the previously deployed hierarchy altogether, the Asset Model Converter will not succeed in creating the new models and assets.

**Does any hardening of the OS occur during deployment?**

Yes. Fail2ban is installed and enabled on the device, which bans IPs that show the following malicious signs:

* Too many password failures
* Seeking for exploits

Additionally, shared memory on the device is secured via /etc/fstab.

**Can I re-run the AMC to create new models and assets?**

Yes. Follow these steps:

1. If any of the models/assets you want to create share a name with the models/assets created in the first pass of the AMC execution, you’ll need to delete the models and assets in SiteWise.
2. The second action you’ll need to take is clearing out the following DynamoDB tables associated with the initial IMC kit deployment:
   1. [stack-name]-asset-model-table
   2. [stack-name]-asset-table

Depending on your driver of choice, you’ll need to take the follow actions to re-run the AMC:

* IgnitionCirrusLink
  + Initiate a re-birth (NBIRTH,DBIRTH) MQTT message that represents the your project hierarchy.
    - Connect to your instance of Ignition in the Ignition Designer
    - A screenshot of a cell phone

      Description automatically generatedA birth message is triggered by launching the Ignition Designer, navigating to the tag browser 🡪 tag providers 🡪 MQTT Transmission 🡪 Transmission Control 🡪 clicking the “Refresh” button.
    - This action triggers the IMC kit’s AMC, which creates the models and assets that represent the Ignition hierarchy in SiteWise.
* IgnitionFileExport
  + Obtain your JSON file that represents the new hierarchy and drop it into the following S3 bucket:
    - [name\_of\_stack]-[amcincomingresource]-[hash]
* KepServerFileExport
  + Obtain your JSON file that represents the new hierarchy and drop it into the following S3 bucket:
    - [name\_of\_stack]-[amcincomingresource]-[hash]

# Creating AMC Drivers

## Instructions

1. Write the driver that interprets the incoming hierarchy data from your edge-based asset modeling software and converts it into the AMC-approved format ([see the format here](#_AMC-Approved_DynamoDB_Format)) and puts it into DynamoDB
   1. Refer to the template file for guidance while writing your driver:
      1. /functions/source/AssetModelConverter/drivers/example\_driver\_template.py
   2. \*\* Highly recommended – also refer to the existing drivers:
      1. /functions/source/AssetModelConverter/drivers/igniitonCirrusLinkDriver.py
      2. /functions/source/AssetModelConverter/drivers/ignitionFileDriver.py
      3. /functions/source/AssetModelConverter/drivers/kepserver\_file\_driver.py
2. Edit the entry point file for the AMC (/functions/source/AssetModelConverter/assetModelConverter.py) to use your new driver:
   1. Import your driver
      1. From drivers.[name\_of\_file] import [name\_of\_driver\_class]
   2. Add your driver to the ‘driverTable’ list
      1. ‘[name\_of\_driver]’: [name\_of\_driver\_class]
3. Replace the AssetModelConverter zip file with its new contents:
   1. Zip up the contents of /functions/source/AssetModelConverter/
   2. Name the zip file above “AssetModelConverter.zip”
   3. Replace the old “AssetModelConverter.zip” file (/functions/packages/AssetModelConverter/AssetModelConverter.zip) with the new “AssetModelConverter.zip” file you created in ‘b’ above.
4. Edit the CloudFormation template to include your driver’s name:
   1. /templates/IMC-workload.template.yaml
      1. Add an item to the list of AMCDrivers (parameter section)
         1. - [name\_of\_driver\_here]

# Appendix

## Artifacts

**The following directories and files are necessary for running an IMC kit deployment:**

functions/

scripts/

templates/

LICENSE.txt

NOTICE.txt

README.md

**quickstart**-IMC: The root directory in the S3 bucket, where the rest of the folders live.

**functions**: Contains zipped lambda code that is used for various pieces of the IMC kit.

**scripts**: Contains the scripts that are run on physical hardware if running a physical deployment.

**templates**: Contains the various CloudFormation templates that will be deployed depending on the deployment options selected during stack creation.

## AMC-Approved DynamoDB Format

**Asset Model Table ([name-of-stack]-asset-model-table):**

assetModelEntry = {  
    “assetModelName”: type<string>, # Name of the asset model  
    “parent”: type<string>, # name of the parent asset model, if any  
    “assetModelProperties”: type<list<modelProperty>>, # list of sitewise assetModelProperties as ‘modelProperty’ listed below.  
    “assetModelHierarchies”: type<list>, # sitewise assetModelHierarchies, leave blank []  
    “change”: type<string>, # Should be ‘YES’, indicates in DynamoDB that the record is new or updated.  
}  
    modelProperty = {  
        ‘name’: type<string>, # Name of the property  
        ‘dataType’: type<string>, # Sitewise data type of the property  
        ‘type’: {  
            ‘measurement’: {} # Don’t change this or populate it with anything, used to identify property type in sitewise  
        }  
    }

**Asset Table ([name-of-stack]-asset-table):**

assetEntry = {  
    ‘assetName’: type<string>, # name of the asset  
    ‘modelName’: type<string>, # model name this asset is an instance of  
    ‘change’: type<string>, # Should be ‘YES’, indicates in DynamoDB that the record is new or updated.  
    ‘tags’: type<list<tagEntry>>, # List of tagEntry struct, as specified below  
}    tagEntry = {  
        ‘tagName’: type<string>, # name of the tag  
        ‘tagPath’: type<string>, # Full property alias path for the tag  
    }