Table of Contents

[Industrial Machine Connectivity QuickStart Getting Started Guide 2](#_Toc42788327)

[Deployment Types 3](#_Toc42788328)

[Data Flow Options 4](#_Toc42788329)

[Option 1 4](#_Toc42788330)

[Option 2a 4](#_Toc42788331)

[Option 2b 4](#_Toc42788332)

[Deployment Guide 5](#_Toc42788333)

[Type 1: Virtual 6](#_Toc42788334)

[Pre-Requisites: 7](#_Toc42788335)

[CloudFormation Stack Launch 8](#_Toc42788336)

[IMC Kit Configuration 11](#_Toc42788337)

[IMC Kit Operation 13](#_Toc42788338)

[View SiteWise Portal Data 13](#_Toc42788339)

[Cleanup 14](#_Toc42788340)

[FAQs 15](#_Toc42788341)

[Troubleshooting 16](#_Toc42788342)

[Physical - Greenfield 17](#_Toc42788343)

[Physical - Brownfield 18](#_Toc42788344)

[Appendix 19](#_Toc42788345)

[Artifacts 19](#_Toc42788346)

# 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, as well as 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 ISV edge-based asset modeling conventions into AWS IoT SiteWise.

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

• Inductive Automation (Ignition Server)

• PTC (KepServer)

The AMC will ingest the edge software’s (i.e. Ignition or KEPServerEX) native asset model definition file(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:

1. Virtual:
   1. **Type 1**: 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.
2. Physical: Physical deployment of the IMC kit enables users to deploy edge software (i.e. AWS IoT Greengrass and partner edge software) on real physical industrial PCs that are read to connect to physical devices (I.e. PLCs)/historians/SCADA systems on the customers plant floor. The physical deployment has 2 versions:
   1. **Type 2**: Physical - Greenfield: AWS IoT Greengrass and the partner edge software will be running on a single industrial PC.
   2. **Type 3**: 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 ISV 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.

In addition to the Virtual/Physical edge hardware distinction, the IMC kit supports 2 types of deployment types:

# Data Flow Options

Once you decide which of the 3 deployment types to launch, you will determine which data flow option to configure. There are 3 data flow options that provide flexibility to the user on how to move data from the customer’s edge environment to AWS IoT services in the cloud.

## Option 1

**Edge ISV OPCUA Server -> AWS IoT SiteWise:**

In this case, we have the AWS Greengrass SiteWise Connector configure to connect to the Edge ISV OPCUA Server. All telemetry data will flow directly into AWS IoT SiteWise.

## Option 2a

**Edge ISV Server -> AWS IoT Core:**

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

## Option 2b

**Edge ISV Server -> AWS Greengrass Core -> IoT Core -> S3:**

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

# Deployment Guide

Navigate to the section for your deployment type

Deployment Configurations:

1. Type 1: Virtual
2. Type 2: Physical - Greenfield
3. Type 3: Physical - Brownfield

# Type 1: 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. IMC Kit configuration
4. IMC Kit operation
5. Cleanup
6. FAQ
7. Troubleshooting

## 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:** 
  + 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:** Next 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

## CloudFormation Stack Launch

Now we'll 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

1. Prerequisite - Prepare template: Leave as default - “Template is ready”
2. 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/templates/IMC-workload.template.yaml>
3. 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: Choose the desired data flow option. The options are:
      * Option 1: Via OPC UA through SiteWise connector to AWS IoT SiteWise in the AWS cloud
      * Option 2(a): Via MQTT from partner edge software to AWS IoT Core in the AWS cloud
      * Option 2(b): via MQTT to AWS IoT Greengrass core and then to AWS IoT Core in the AWS cloud.
  + 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**: Leave as default “IgnitionCirrusLink”. For the virtual deployment you may also choose “IgnitionFileExport” which will require you to manually export the tag definition file from Ignition Server and upload to an S3 bucket. To use this AMC driver option, follow the steps detailed in the IMC Kit Configuration section.
    - Us**er 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

* 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.

* 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

...

```

## IMC Kit Configuration

Once the CloudFormation stack is completed, follow the steps to configure the IMC Kit to make it operational. You only need to follow the steps for the specific data flow option you chose during the CloudFormation stack launch:

* Option 1
* Option 2(a)
* Option 2(b)

Option 1

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. 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
         1. You have the option to update login credentials once you log in.
   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.

Option 2a

1. 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.
2. 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.

Option 2b

1. 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.
2. 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.

## IMC Kit Operation

### 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.
3. Troubleshooting:
   1. If data is not flowing in the dashboard, 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 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.

## Cleanup

\*\*TODO\*\* Need to add additional details to this section

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 (if desired, for cleanliness): SiteWise Portal, SiteWise Gateway, SiteWise Models and Assets, QuickSight dataset.

## 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.](#_Deployment_Guide) 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.

## Troubleshooting

Quarantined certificate in Ignition doesn't show up for Option 1 deployments

- Navigate to the "Gateways" in the IoT SiteWise console, find the Gateway associated with your deployment (compare to the Greengrass Group ID if required), hit "Edit" then hit "Save". Look out for the Quarantined certificate in the Ignition console.

Data does not show up on IoT topic for Option 2a deployments

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/[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://<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
        1. You have the option to update login credentials once you log in.
4. Navigate to "MQTT Transmission -> Settings -> Server" and confirm that the connectivity shows 1 of 1. If it doesn’t, click edit and:
   1. 2a Deployments:
      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\_the\_edge\_device\_parameter]/[name\_for\_the\_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”.
   2. 2b Deployments:
      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\_the\_edge\_device\_parameter]/[name\_for\_the\_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 Private Key File with the “.private” file from the tarball package
      7. Replace the CA Certificate File with the “root.ca.pem” file retrieved from step iv.
      8. Check the “show advanced properties” box.
      9. Make sure the client ID reads: [name\_for\_the\_edge\_device\_parameter]Device.
      10. Hit “Save Changes”, and make sure that the connectivity says “1 of 1”.

# Physical - Greenfield

TODO: Insert instruction for Physical - Greenfield deployment

# Physical - Brownfield

TODO: Insert instruction for Physical - Greenfield deployment

# Appendix

## Artifacts

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

S3 bucket Contents:

quickstart-IMC/

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.