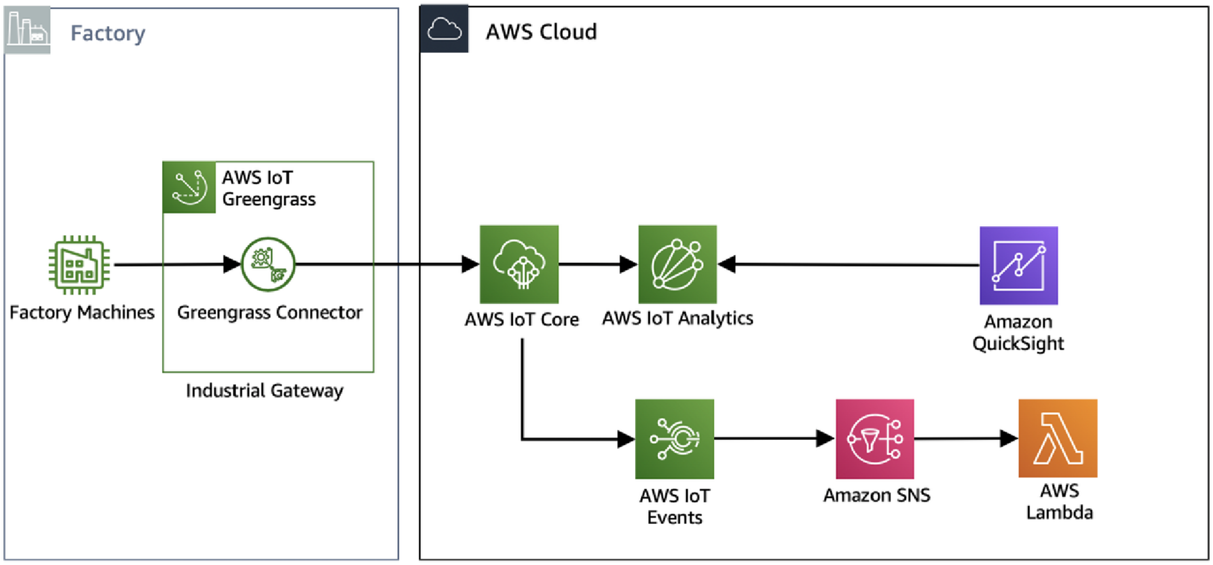
MFG401 - Builders Session - Asset condition monitoring with AWS industrial IoT services

Introduction

In this builders session you will go through the fundamental steps to see how to build an asset condition monitoring solution in-line with the below architecture.



For this session you will deploy a CloudFormation template that will create a Drill Machine simulator on a EC2 instance that will simulate the Factory Machine within the architecture.

Instructions

BEFORE YOU START

This builders session is set at level 400 and therefore provides the high-level steps to build the architecture with the AWS builder providing you guidance through out the session.

You will need a laptop that is running Linux, Windows or macOS with a AWS account to complete the session and AWS will provide you AWS credits to cover the cost of services used during the session.

STEP 1. DEPLOYING THE CLOUDFORMATION TEMPLATE

- Log into your AWS Console with a IAM user that has an administrator role.
- Launch one of the following CloudFormation templates per your region of choice (suggest Oregon).
  - Oregon
  - Ireland
  - N. Virginia

- Leave all the Parameters as default
- Click the tick box at the bottom of the page to acknowledge the IAM capabilities and create the stack.

> *Note: It will take about 5-10 minutes for the CloudFormation template to successfully deploy*

CONFIGURING GREENGRASS

- Via the AWS Console go into IoT Core > Greengrass and “Create a Group”
- If promted “Grant permission”
- Click “Use easy creation” and name the group for example “Drill”.
- Create the group and core and download the security resources and the current Greengrass Core software.

> *Note: For this session as Greengrass is running on a EC2 instance you will need the x86_64 distribution.*

- Click Finish within the AWS Console window.
- To make the session easier for yourself open a new browser window and go to Cloud9 within the AWS Console
- Click on Open IDE
- Select the folder “my-greengrass” and upload the downloaded zip files from the previous step
- Run the following commands within the terminal to extract and install the Greengrass software and the security certificates.

```
sudo tar -xzvf ~/environment/my-greengrass/greengrass-linux-x86-*.tar.gz -C /
```

```
sudo tar -xzvf ~/environment/my-greengrass/*-setup.tar.gz -C /greengrass
```

- Using yum install Java 8.

```
sudo yum install java-1.8.0-openjdk
```

- Switch /usr/bin/java to link to the java8 version.

```
sudo ln -sf /usr/bin/java8 /usr/bin/java
```

- Start Greengrass with the following command

```
sudo /greengrass/ggc/core/greengrassd start
```

- Switch back to the IoT Core window
- For the Greengrass group add a Role via the Settings menu.

> *The role will start with a name like IIoTWS-GGGatewayRole-**

- Add a local log type for Lamba and Greengrass System with a level of Debug.
- Deploy the Group via the Action menu, use “Automatic selection“

> *Note: If the deploy fails run the following commands in the terminal window.*

```
gg_service_arn=$(aws iam get-role --role-name Greengrass_ServiceRole --query Role.Arn)
aws greengrass associate-service-role-to-account –role-arn ${gg_service_arn//\"/}
```

VIEWING THE DRILL DATA

- Go back to the Cloud9 terminal window and run the following command.

```
opcua-commander -e opc.tcp://192.168.128.20:4840
```

- Using the arrow keys browse down the following tree

> `Objects→ Server→ PLC1→ MAIN→ Drill`.

- Select “Status” and press “M”, this will allow you to view the values being generated from the Drill.
- Press “Q” to quit out of the OPC-UA browser.
- Go back to the IoT window and within the Drill Greengrass group select “Lambdas” and “Add Lambda”
- Use and existing Lambda which starts with “IIoTWS-OPCUA”
- Select the “latest-version” and click “Finish”
- Edit the Lambda configuration and ensure the “Memory limit” is 160MB and the “Lambda lifecycle” is “long-lived”.
- From the Subscriptions menu add a new subscription with the source being the IIoTWS-OPCUA lambda and the target being the IoT Cloud.

> *Note: Set the topic filter to #*

- Now redeploy the Group via the “Actions” menu
- From the AWS IoT menu, select “Test”
- Subscribe to “#” in the topic.

You should see message with values similar to that from the OPC-UA Browser.

EXPOSING ALL DATA FIELDS

- Go back to the browser window with the Cloud9 terminal.
- Open the `index.js` file and add 3 more subscriptions for the **Spindle Speed, the Motor Speed and the Pressure**

> *Note: index.js is within the Lambda_Functions > IIoTWS-OPCUA folder.*

```
{
    name: 'Drill.DrillState',
    nodeId: 'ns=5;s=MAIN.Drill.DrillState',
},
{
    name: 'Drill.Pressure',
    nodeId: 'ns=5;s=MAIN.Drill.Pressure',
},
{
    name: 'Drill.SpindleSpeed',
    nodeId: 'ns=5;s=MAIN.Drill.SpindleSpeed',
},
{
    name: 'Drill.MotorSpeed',
    nodeId: 'ns=5;s=MAIN.Drill.MotorSpeed',
}
```

- Make sure you save the file changes
- Run the following commands below to update the functions version.

```
cd ~/environment/Lambda_Functions
./upload-publish-and-version-lambda-function.sh *OPCUA*
```

- In the IoT console window in the Greengrass group, deploy the group via the action menu.
- Via the “Test” menu and Subscribing to # you will be able to see all the data values.

PRE-PROCESSING THE DATA AT THE EDGE

The next section can be done within AWS IoT Events but for simplicity of the session we will do event processing via a Lambda function at the Edge.

- From IoT Core, add a new Lambda to the Greengrass Group
- Select the existing Lambda which name starts with IIoTWS-CombineEvents
- Select the alias “latest-version” and set the memory to 160MB and the Lambda to run as long lived.
- Add two new subscriptions to the Greengrass group as per the below.

Source	Target	Topic
IIoTWS-OPCUA	IIOTWS-CombineEvents	`opcua/server/node/#`
IIOTWS-CombineEvents	IoT Cloud	#

- Deploy the Greengrass group

> *Note: If you now subscribe to the topic* `data/aggregated you will see a combined data set.`

ANALYISING THE DATA

- In the AWS console, go to IoT Analytics.
- Use the Quick Create option, using the Resource prefix as “Drill” and the Topic as “`data/aggregated"`
- From the menu go to Data sets, select the “drill_dataset”
- Add a schedule of 15 mins and retain data set content as Indefinitely
- From the Action menu, run now

After a couple of seconds you should see a preview of the data in the Result preview menu.

- Create a new data set with a new SQL
- Name it drill_quicksight, select “drill_datastore” as the data store
- Add the below query statement

```
SELECT substr(workstart,1,19) as ts, * FROM drill_datastore
```

- Leave the Data selection window as “None” and set a Frequency of 15 minutes.
- Run the new data set and you should notice a new column called “ts” in the preview.

VIEWING THE DATA IN QUICKSIGHT

- From the AWS Console open **“QuickSight”**

> *Note: If you haven't used QuickSight before you will see a screen asking to Sign Up*

- Select **“New analysis”**
- Click **“New data set”**
- Select **“AWS IoT Analytics”** (if you don't see this item you need to enable IoT Analytics data sets in the setting)
- Select the AWS IoT Analytics data set that you just created and click **“Create data source”**

After the data set has been created click **“Edit/Preview data”**.

> *NOTE: if you get an error message that QuickSight cannot connect to the data set, go back to AWS IoT Analytics, select the data set and refresh it.*

- Expand the Fields pane on the left side, and deselect all the label fields, such as `spindlelow, durationtooshort, spindlehigh, motorlow, pressurehigh, hasanomaly, durationtoolong, motorhigh`. This is to reduce clutter in the field selection pane for the visualisation. Notice also that the `ts` field is of type `date` while `workstart` is of type `string`.
- Now click on **“Save & visualize”** on the top bar.
- On the left hand side you have a panel with the available fields, and on the bottom the available visual types. The default selection is to let QuickSight automatically determine the visualisation type based on the selected fields, but we suggest to explicitly pick the Line Chart visualisation for displaying time series data.
- Select the **ts** and **avgpressure** fields by clicking on them. You should have a graph showing a line. In the field wells section click on the down arrow on the **ts** field and change the Aggregate value to **Minute**. Then the **avgpressure** field change the aggregation to **Average**.
- Add the **avgpressure** field again twice to the Value field well and select **Min** and **Max** aggregate respectively.

Conclusion

Congratulations you are now able to analysis and visualise the output from an industrial devices in the Cloud and see monitor parameters of a machine.

The next step would be to start building history of the machines parameters and use the data to train a machine learning model to identify significant changes in the parameter or anomalies which could be an identifier for a developing fault.