# Solution accelerator for Industrial IoT on Azure

Often times I have seen customers and partners getting stuck at how to use Microsoft's OPC modules and how to use OPC UA data upstream for different purposes. This project is a solution accelerator that provides guidance on how to get started with the OPC Publisher module at the edge and data visualization. This article makes some important assumptions:

* The initial version of this deployment provides default security settings. This is not recommended for production environments as security is something that should be discussed and planned in advance.
* The IoT edge runtime is deployed on an Ubuntu 18.04 virtual machine. Similar results can be achieved with Windows or other Linux distributions as well.

#! Note: This solution aims to serve as a starting point to get familiar with some of the core components of the Industrial IoT platform and other Azure services that are usually paired with them in production scenarios. For a complete deployment of the Azure Industrial IoT Platform, please refer to the official repository [here](https://github.com/azure/industrial-IoT/).

## Prerequisites

In order to successfully deploy this solution, you will need a couple of things first:

* \*PowerShell\*. This deployment script is written in PowerShell. If you are using a Linux environment, follow these [instructions](https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-core-on-linux?view=powershell-7) to install PowerShell on Linux.
* \*Azure CLI\*. Follow these [instructions](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest) to install or update to the latest version.
* \*Webhook endpoint\*. This solution uses Azure Event Grid to send notifications and alerts, so you will need a webhook endpoint to receive such events. There are several services that provide this functionality for free. You can obtain a webhook URL by going to <https://webhook.site/> and copying **your unique URL**.

[webhook image]

!# Node: If you are interested in understanding how Azure Event Grid works and create your own webhook, you can look at this Azure Event Grid viewer [sample solution](https://github.com/Azure-Samples/azure-event-grid-viewer/tree/master/).

## Architecture

[architecture image]

## Getting started

Clone the repository:

```

git clone <https://github.com/marvin-garcia/IoTGateway.git>

cd IoTGateway

```

Start the deployment:

```

. ./deploy.ps1 -webhook\_url <webhook url> -deploy\_time\_series\_insights $true

```

!# Note: If you don’t want to include Time Series Insights in the deployment, set \*deploy\_time\_series\_insights\* to $false

## Next Steps

Once the solution has been successfully deployed, you may want to spend some time understanding each component.

[link] OPC Simulator

[link] OPC Publisher

[link] Edge Stream Analytics job

[link] IoT Edge Deployment

[link] Cloud Stream Analytics job

[link] Notification & Alerting

[link] Real-time data visualization through Time Series Insights

[link] Data visualization through Azure Data Explorer

### OPC Simulator

This project uses an Ubuntu 18.04 virtual machine to simulate three OPC PLC servers, you can find the official documentation for OPC PLC server [here](https://docs.microsoft.com/en-us/samples/azure-samples/iot-edge-opc-plc/azure-iot-sample-opc-ua-server/). In this implementation the three OPC PLC servers will be deployed with the default nodes:

* Alternating Boolean
* Random signed 32-bit integer
* Random unsigned 32-bit integer
* Sine wave with a spike anomaly
* Sine wave with a dip anomaly
* Value showing a positive trend
* Value showing a negative trend

### IoT Edge device

This project uses an Ubuntu 18.04 virtual machine to run the [Azure IoT Edge runtime](https://docs.microsoft.com/en-us/azure/iot-edge/about-iot-edge) and three IoT edge modules on top of it: [OPC Publisher](https://docs.microsoft.com/en-us/azure/iot-accelerators/overview-opc-publisher), edge Stream Analytics job and OPC translator.

#### OPC Publisher

OPC Publisher is a reference implementation that connects to existing OPC UA servers and publishes JSON encoded telemetry data in OPC UA Pub/Sub format, to Azure IoT Hub.