

# KNI IE Installation Guide

This blueprint is part of the Kubernetes-Native Infrastructure for Edge family. All blueprints in this family follow the same installation guide, so please see the KNI family's [User Documentation](#) for deployment pre-requisites and deployment procedures for each target platform (e.g. GCP).

## KNI IE-specific Installation Steps

As described in the [KNI IE Architecture document](#), this blueprint contains of *two clusters*, the central management hub and the edge factory site(s). At a high level, you therefore need to perform the following steps:

1. Provide the deployment pre-requisites (cloud API keys, pull secrets, etc.) for your target platform according to the generic [User Documentation](#).
2. Adapt the blueprint to your own environment:
  - a. Create Quay container image repos to host the blueprints' auto-built images.
  - b. Fork the two GitOps helper repos on GitHub into your own org.
  - c. git clone the two blueprints and two helper repos.
  - d. Replace Akraino URLs with your own and update TLS certs.
  - e. Commit these changes to your repos.
3. Deploy the management hub according to the generic [User Documentation](#).
4. Deploy the factory site according to the generic [User Documentation](#).

The following instructions detail the second step of adapting the blueprint to your own environment.

## Creating Quay image repos

Create the Quay image repositories to host the container images built as part of the GitOps workflow. You can leave these repositories empty since the initial pipeline runs will build the initial images, push them to your Quay repositories, and tag them:

- <https://quay.io/organization/akraino/iot-frontend>
- <https://quay.io/organization/akraino/iot-consumer>
- <https://quay.io/organization/akraino/iot-software-sensor>
- <https://quay.io/organization/akraino/iot-anomaly-detection>

Store your own Quay org name in MYQUAYORG variable:

```
$ export MYQUAYORG=your_quay_org_name_here
```

Create a robot account with write permissions to the four repos and download the access token to ~/.kni/dockerconfig.json.

## Forking the GitHub helper repos

[Fork](#) the following repos on GitHub into your own GitHub org:

- <https://github.com/akraino-kni/manuela-gitops>
- <https://github.com/akraino-kni/manuela-dev>

Store your own GitHub org name in MYGITHUBORG variable:

```
$ export MYGITHUBORG=your_github_org_name_here
```

Create a GitHub personal access token with "repo" permissions for these repos and store them under ~/.kni/githubsecret.json.

## Git Cloning all repos

Execute the following commands to clone the two forked repositories and the two Akraino KNI IE blueprint repos:

```
$ git clone git@github.com:$MYGITHUBORG/manuela-gitops
$ git clone git@github.com:$MYGITHUBORG/manuela-dev
$ git clone https://gerrit.akraino.org/r/kni/blueprint-management-hub
$ git clone https://gerrit.akraino.org/r/kni/blueprint-ie
```

## Replacing URLs and rebuilding TLS certs

Export the following variables, filling in your own cluster names and domains as well as the GCP project ID and region:

```
$ export MGMT_HUB_NAME=your_mgmt_hub_cluster_name
$ export MGMT_HUB_DOMAIN=your_mgmt_hub_cluster_domain
$ export MGMT_HUB_PROJECT_ID=your_mgmt_hub_project_id
$ export MGMT_HUB_REGION=your_mgmt_hub_region
$ export EDGE_SITE_NAME=your_edge_cluster_name
$ export EDGE_SITE_DOMAIN=your_edge_cluster_domain
$ export EDGE_PROJECT_ID=your_edge_project_id
$ export EDGE_REGION=your_edge_region
```

Replace URLs in all repos with your own:

```
$ find $r -not \( -path $r/.git -prune \) -type f -exec sed -i \
-e "s/github.com\/akraino-kni/github.com\/$MYGITHUBORG/g" \
-e "s/quay.io\/akraino-kni/quay.io\/$MYQUAYORG/g" \
-e "s/edge-mgmt-hub.gcp.devcluster.openshift.com/$MGMT_HUB_NAME.$MGMT_HUB_DOMAIN/g" \
-e "s/staging-edge.gcp.devcluster.openshift.com/$EDGE_SITE_NAME.$EDGE_SITE_DOMAIN/g" \
{} \;
done
```

Generate new TLS certificates matching your environment:

```
$ openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem -subj "/C=DE/OU=Manuela
/CN=*.apps.$MGMT_HUB_NAME.$MGMT_HUB_DOMAIN"
$ cat <<EOF >manuela-gitops/config/instances/manuela-data-lake/central-kafka-cluster/kafka-tls-certificate-and-
key.yaml
apiVersion: v1
kind: Secret
metadata:
name: kafka-tls-certificate-and-key
data:
tls.crt: $(base64 -w0 <certificate.pem)
tls.key: $(base64 -w0 <key.pem)
EOF
$ cat <<EOF >manuela-gitops/config/instances/manuela-data-lake/factory-mirror-maker/kafka-tls-certificate.yaml
apiVersion: v1
kind: Secret
metadata:
name: kafka-tls-certificate
data:
tls.crt: $(base64 -w0 <certificate.pem)
EOF
```

Change name and domain of your clusters:

```
$ pushd blueprint-management-hub >/dev/null
$ sed -i -e "s|projectID:.*|projectID: $MGMT_HUB_PROJECT_ID|g" profiles/production.gcp/00_install-config/install-
config.patch.yaml
$ sed -i -e "s|region:.*|region: $MGMT_HUB_REGION|g" profiles/production.gcp/00_install-config/install-config.
patch.yaml
$ git mv sites/edge-mgmt-hub.gcp.devcluster.openshift.com sites/$MGMT_HUB_NAME.$MGMT_HUB_DOMAIN
$ sed -i -e "s|gcp.devcluster.openshift.com|$MGMT_HUB_NAME.$MGMT_HUB_DOMAIN|g" sites/$MGMT_HUB_NAME.$MGMT_HUB_DOMAIN/00_install-
config/install-config.patch.yaml
$ sed -i -e "s|edge-mgmt-hub|$MGMT_HUB_NAME|g" sites/$MGMT_HUB_NAME.$MGMT_HUB_DOMAIN/00_install-config/install-
config.name.patch.yaml
$ popd >/dev/null

$ pushd blueprint-industrial-edge >/dev/null
$ sed -i -e "s|projectID:.*|projectID: $EDGE_PROJECT_ID|g" profiles/production.gcp/00_install-config/install-
config.patch.yaml
$ sed -i -e "s|region:.*|region: $EDGE_REGION|g" profiles/production.gcp/00_install-config/install-config.patch.
yaml
$ git mv sites/staging-edge.gcp.devcluster.openshift.com sites/$EDGE_SITE_NAME.$EDGE_SITE_DOMAIN
$ sed -i -e "s|gcp.devcluster.openshift.com|$EDGE_SITE_NAME.$EDGE_SITE_DOMAIN|g" sites/$EDGE_SITE_NAME.$EDGE_SITE_DOMAIN
/00_install-config/install-config.patch.yaml
$ sed -i -e "s|staging-edge|$EDGE_SITE_NAME|g" sites/$EDGE_SITE_NAME.$EDGE_SITE_DOMAIN/00_install-config/install-
config.name.patch.yaml
$ popd >/dev/null
```

## Committing and pushing changes

Push the changes back to GitHub:

```
$ export REPOS=("blueprint-management-hub" "blueprint-industrial-edge" "manuela-gitops" "manuela-dev")
$ for r in ${REPOS[@]}; do
pushd $r >/dev/null
git add .
git commit -m "Customize URLs and update certificates"
git push origin master
popd >/dev/null
done
```