

IEC Type 2 Installation Guide for R5

- [Introduction](#)
- [Blueprint System Requirements](#)
- [Pre-Installation Requirements](#)

Introduction

This document covers the installation requirements for Integrated Edge Cloud(IEC) Type 2 R5 blueprint. Blueprint can be installed in two different ways

- a) Using terraform command line utility : A multi-node cluster is provisioned by applying the terraform template. EdgeX Foundry can be installed manually using the deployment specification repository [here](#).
- b) Using a Platform approach : Blueprint (terraform template) can be uploaded to [gopaddle](#). The blueprint can then be used to launch multiple Kubernetes environments from an interactive GUI based approach. Once the cluster is ready, EdgeX Foundry can be installed in the Kubernetes environment by choosing the template from the gopaddle catalog. The north bound APIs to interact with gopaddle can be found [here](#).

Blueprint System Requirements

Installing the blueprint brings up a 3 node cluster with 1 master and 2 worker nodes. Node sizes should a minimum of t4g.medium. A pre-existing VPC and a sub-net is required prior to the installation process. Host machine for the cluster requires Ubuntu 18.04.

Item	Capacity
Number of nodes	3
Node Size	t4g.medium - 2vCPUs - 4 GiB Memory
Disks in Storidge HA Clustering mode	<div>NOT YET SUPPORTED</div> 3 Disks per node - 100 GB each.
VPC	Pre-existing VPC
Subnet	Public (for now). Will switch to private subnet with Gateway configuration in future releases.
Amazon Machine Image (AMI)	Ubuntu Server 18.04 LTS

How to use this document

The following sections describe the prerequisites for planning an IEC deployment. Once the pre-requisites are met, installation steps provided should be followed in order to obtain an IEC compliant Kubernetes cluster.

Pre-Installation Requirements

1. Download the terraform template to the client machine from the gerrit repo | <https://gerrit.akraino.org/r/c/iec/+/4273>.
2. If Follow the instructions [here](#) to install terraform on the client machine from where the blueprint install is to be executed.



Supported Client OS

Ubuntu 18.04

3. AWS IAM User Access Keys - Create an AWS IAM User by following the steps [here](#). Enable **Programmatic Access** and choose **Attach existing policies directly**. Select **AmazonEC2FullAccess** to grant full access to EC2 services.



Security Consideration

In the future releases, access policies will be scope to specific operations instead of a complete EC2 access.

4. Generate an AWS Private Key file as described [here](#). Private Key file is required to access the EC2 instances during the installation process. Place the private key file in the root directory of the template folder.

5. Initialize the environment variables to configure the AWS specific inputs. Choose a region, an AMI, a pre-existing VPC and sub-net. Here is an example of how these environment variables can be initialized. TF_LOG_PATH specifies the file path where the terraform execution logs will be redirected. TF_LOG can be set to TRACE, DEBUG, INFO, WARN, or ERROR.

```
export TF_VAR_aws_region="us-east-2"
export TF_VAR_aws_ami="ami-026141f3d5c6d2d0c"
export TF_VAR_aws_instance="t4g.medium"
export TF_VAR_vpc_id="vpc-561e9f3e"
export TF_VAR_aws_subnet_id="subnet-d64dcabe"
export TF_VAR_access_key="<aws-access-key>"
export TF_VAR_secret_key="<aws-secret-key>"
export TF_LOG="TRACE"
export TF_LOG_PATH="tf.log"
```

5. Apply terraform

To create a master node with microk8s, run the following commands.

```
terraform init
terraform plan
terraform apply
```

Once the worked nodes are created, they will be connected to the master automatically. A multi-node k8s cluster will be provisioned with a calico CNI.