

IEC Type 2 Test Document for R5

Introduction

This document covers both Integrated Edge Cloud(IEC) Type 1 & 2.

[Integrated Edge Cloud \(IEC\)](#) focuses on the multi-architecture based solution on the Edge and Cloud computing areas. In this document, we will give an overview of IEC testing done in CI/CD.

This document is a successor of [IEC Test Document R1](#).

Testing of the IEC platform is done in CI/CD using the validation labs described [here](#).

Once the terraform is done provisioning the Kubernetes environment, the test.sh script is run to validate the cluster set-up. If the setup was successful, the script must return the list of nodes. During the terraform apply process, the config files are moved from the remote machines, both from master and worker, to the local machine where terraform processes takes place. The test.sh script then moves the master node's config file to ~/.kube/config in the local machine. Then the 'kubectl get nodes -A' is executed to validate the cluster.

CI/CD process

Pre-Requisites :

The instructions mentioned here are to set up a private Jenkins lab for the CI/CD process.

1. Terraform must be installed as a pre-requisites in the Jenkins containers. Execute the following steps to install terraform.

```
wget https://releases.hashicorp.com/terraform/0.14.9/terraform_0.14.9_linux_amd64.zip
unzip terraform_0.14.9_linux_amd64.zip
sudo mv terraform /usr/local/bin/
```

2. Ensure the Installation by executing the following command.

```
terraform --help
```

3. Jenkins Image used: docker pull Jenkins. Bring up Jenkins with default installations.

Setup :

A Jenkins freestyle job with some additional plugins is to be set up for the CI/CD process. The freestyle job waits for events from the Gerrit-Event trigger plugin and then initializes the terraform process. The terraform process validates the template and provisions the infrastructure through terraform commands. Once the terraform is done provisioning the cluster, a post-build task is set up to push the build log to the nexus server. The logs are from terraform init, plan and apply commands which is also stored in a tf.log file. The following are the steps to set up the jenkins job.

1. Add Gerrit credentials under Manage Jenkins > Manage credentials > Add new credentials. Enter your Gerrit user name and password.
2. Install Gerrit trigger plugin & post-build task plugin from > Manage Jenkins > Manage plugins > Available plugins.

To Configure the Gerrit - trigger plugin, Follow the instructions below :

- a. Create a folder ~/.ssh inside Jenkins container
- b. Generate ssh key using **ssh-keygen -m PEM**
- c. Register the id_rsa.pud key in the Gerrit account under settings > ssh keys > add new keys
- d. Once the key is registered with the Gerrit account, test the connectivity with the following command.

```
ssh -p 29418 <UserName>@<HostName>
```

3. Create a new freestyle job and configure the job with your Gerrit account in the Source Code Management section. Enter the repository URL and select the credentials that were added in step 1.
4. Add the following shell script under the 'Build' section. These instructions are executed in order while building the job. The script mainly exports TF_VAR for terraform purposes, installs LFtools, and initializes the terraform init, plan and apply processes.

```

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=""
export TF_VAR_aws_subnet_id=""
export TF_VAR_access_key=""
export TF_VAR_secret_key=""
export TF_LOG="TRACE"
export TF_LOG_PATH="./tf.log"
sudo apt-get update
sudo apt-get -y install python3-pip
curl --proto 'https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
export CRYPTOGRAPHY_DONT_BUILD_RUST=1
apt-get install build-essential libssl-dev libffi-dev
pip3 install lftools
cd /var/jenkins_home/workspace/gerrit-akraino/src/foundation/microk8s
terraform init
terraform plan
terraform apply -auto-approve

```

5. Add the following to the post-build task. The post-build script is for pushing the current build's log to the nexus server. Configure the SILO, JENKINS_HOSTNAME, JOB_NAME etc accordingly.

```

echo "post build tasks"
cat /var/jenkins_home/workspace/gerrit-akraino/src/foundation/microk8s/tf.log
echo $BUILD_NUMBER
NEXUS_URL=https://nexus.akraino.org
SILO=gopaddle
JENKINS_HOSTNAME=<hostName/IP:PORT>
JOB_NAME=gerrit-akraino
BUILD_URL="${JENKINS_HOSTNAME}/job/${JOB_NAME}/${BUILD_NUMBER}/"
NEXUS_PATH="${SILO}/job/${JOB_NAME}/${BUILD_NUMBER}"
lftools deploy logs $NEXUS_URL $NEXUS_PATH $BUILD_URL
echo "Logs uploaded to $NEXUS_URL/content/sites/logs/$NEXUS_PATH"

```

6. Once configuring the job is done, build it. It might take some time to complete. To check if the source code is pulled from gerrit, go to /var/jenkins_home/workspace/<jobName>/src/foundation/microk8s.