

Introduction

This document is a guide on how to setup Root Level Azure objects for the KYOCERA PLATFORM via Terraform.

The files within this sub-directory contains the necessary terraform config to create/update the Root Level Azure objects.

Any changes to any of the Root Level objects should be done here (the terraform config files) instead of manually changing the properties via Azure Portal.

NOTE: It has initially been decided that we will only use one App Service Plan and one SQL Database server for "KYOCERA PLATFORM" subscription unless there is a special circumstance. We will also create one Root Level Storage Account for any root level storage needs.

Set Environment Variables Before You Run Terraform

Log in to Azure and list your subscriptions. Check which subscription you are currently connected to and if necessary, change set it to the correct subscription.

```
# Log in to AZ
az login

# Dump available AZ Subscription accounts
az account list --output table

# Set you account to the subscription where you want to create these objects
az account set --subscription 8015bba0-aba8-410e-9023-33f279d17af2 #devtest
az account set --subscription 11b09e5a-5eca-45cf-a49a-63542a69cf18 #payg
az account set --subscription 7baed697-6342-4c6f-bdde-3392c3e7a6c7 #kplat

AZLOCATION=westus2
SUBSCRIPTION_INDICATOR=devtest

AZLOCATION=westus2
SUBSCRIPTION_INDICATOR=payg

AZLOCATION=westus3
SUBSCRIPTION_INDICATOR=kplat

KVNAME='kv-kii-tfstate-'$SUBSCRIPTION_INDICATOR
```

Terraform environmental variables

Terraform will rely on environmental variables for running with a Service Principle context. To do so, we need to set environmental variables in whatever system Terraform will run. For example, if we are running Terraform

on the local machine, we will have to set the environmental variables locally so that Terraform will have the proper access to run. We will now set environmental variables by getting the values from the Key Vault.

```
echo $KVNAME
echo "Length is ${KVNAME}. Valid length is 3-24; Alphanumerics and hyphens."

export ARM_ACCESS_KEY=$(az keyvault secret show --vault-name $KVNAME --name
StorageAccountKey --query value -o tsv)
export ARM_TENANT_ID=$(az keyvault secret show --vault-name $KVNAME --name
TenantId --query value -o tsv)
export ARM_SUBSCRIPTION_ID=$(az keyvault secret show --vault-name $KVNAME --name
SubscriptionId --query value -o tsv)
export ARM_CLIENT_ID=$(az keyvault secret show --vault-name $KVNAME --name
AppRegistrationClientId --query value -o tsv)
export ARM_CLIENT_SECRET=$(az keyvault secret show --vault-name $KVNAME --name
AppRegistrationClientSecret --query value -o tsv)

printenv | grep ^ARM*
```

Terraform Guide

1. Start terminal session and sure you are in the `/RootLevelObjects` directory.
2. ~~Connect to Azure with az login.~~ There's no need to log in to AZ as long as we are setting the environment variables for Terraform. Terraform will use a service principal to connect to Azure.
3. Choose and set current subscription.
 - `az account set --subscription "KYOCERA Platform"`
 - Note: The "provider" property of the terraform config files will overrides the subscription based on value selected (check "00_main.tf")
4. Confirm current subscription.
 - `az account show --output table`
5. Run `terraform init`
6. Run `terraform validate`
7. Run `terraform plan -out AzureAppPlanKYOPlatform.tflplan`
 - The output file "az-app-plan.tflplan" should be tracked in git
8. Run `terraform apply AzureAppPlanKYOPlatform.tflplan`
9. Check Azure Portal for the newly created objects.
10. To remove everything and start over: `terraform destroy`
 - When prompted. type: yes
11. **[IMPORTANT]** If the terraform config and output plan has been updated, make sure commit and push the new changes!