ADJUST ASSIGNMENT

Building the Infrastructure:

Tools Used: Terraform, Docker, Kubernetes(AKS), Azure Devops

Requirements: Azure and Azure Devops Account and SPN for authentication and Storage account for storing backend remote file of Terraform.

Getting Started:

Quick Steps to recreate:

Infra and Application Deployment:

1. Clone the repo from git in Azure Devops.
2. Create a build pipeline in Azure Devops as 'app-build-deploy.yaml'.
3. Pass the below variables in variables section of the pipeline:

Variables	Values
az_subscription_id	Your Subscription id
az_tenant_id	Your Tenant ID
provider_client_secret	SPN App Secret
provider_client_id	SPN Client ID
region	Region of your choice (eg: 'eastus')
Project	Project Name (eg: 'adjust')
dockerid	Username in docker ('kanika02' in my case)
ui_image	Image name you wish to provide Eg: rubyapp_adjust
Username	Docker username
password	Docker Password
image_tag	Ver1

4. Run the pipeline, you'll see the resources will get created on Azure.
5. Check the output of the pipeline, you'll get the ip of the service running in kubernetes
6. You can take the ip and curl from your machine to test the application.

Infrastructure Strategy:

This section explains about the infrastructure which includes a Kubernetes cluster deployed on Azure and application inside that cluster through terraform.

Terraform files:

- **Provider.tf** - We have taken azurerm as terraform provider and backend state is stored in the storage account. SPN, subscription and tenant are defined as variables.
- **Main.tf -** All the variables including modules variables are defined in this file.

MODULE: COMMON

- **variables.tf –** all variables to be used in this module are defined here.
- **commonResources.tf -** In this file, we have created a Resource group with custom name using variables like region, project. For this Cluster, we have defined two nodes and utilized kubenet networking settings. However, we can also take Azure CNI (based on the requirements).

YAML:

- We have then defined the ui-yaml which will deploy the ruby application.
- The service is exposed as load balancer.
- In deployment, we have created 2 replicas for high availability, liveliness probe and readiness probe to check the health of the container and restart it if the state is not healthy.

Reference Image: docker.io/kanika02/rubyapp_adjust:ver1

PIPELINES (app-build-deploy.yaml)

Pipeline for deploying the Infra and Application:

We have defined two jobs in this pipeline as mentioned:

1. **Build :**

- In first task, we are creating the docker image from mentioned dockerfile and pushing it to the docker hub.
- In the second task, we are taking the image tag from above generated task and passing that output to next task.

2. **Deploy:**
   In Deploy Job, we have are deploying our infra. Below are the three tasks we have used for this:

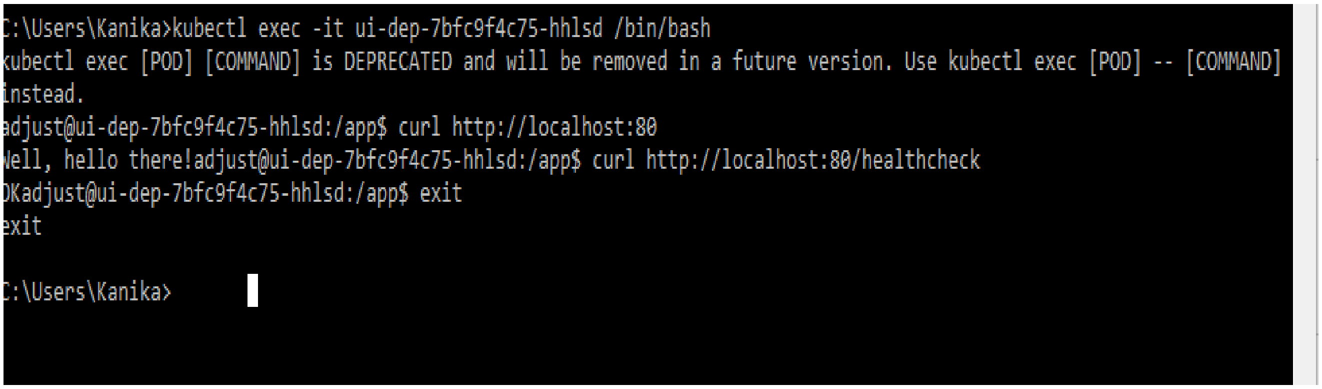
- **Terraform init**- to initialize the working directory.

- **Terraform plan:** we are passing the variables as arguments in task and defining variables in the variable section of the pipeline.
- **Terraform Apply**: we are passing the variables as arguments in task and defining the values in variable section of the pipeline.

Make note of the output 'applicationendpoint' from the above pipeline logs. You can verify the application by using curl command.

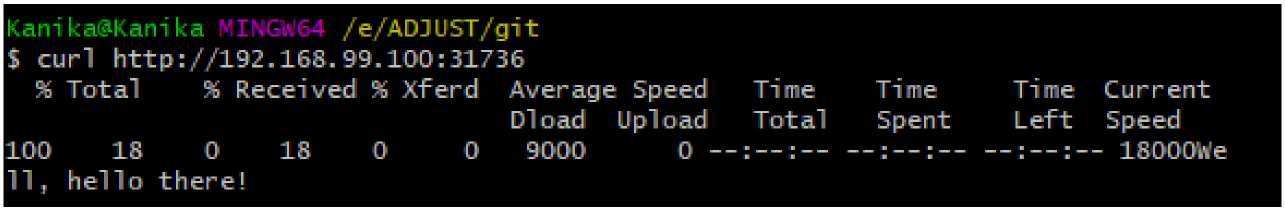
Output verification:

Once the Application is deployed, you can verify its output by checking the url using curl command as shown in the screenshot below:



```
C:\Users\Kanika>kubectl exec -it ui-dep-7bfc9f4c75-hhlsd /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND]
instead.
adjust@ui-dep-7bfc9f4c75-hhlsd:/app$ curl http://localhost:80
Well, hello there!adjust@ui-dep-7bfc9f4c75-hhlsd:/app$ curl http://localhost:80/healthcheck
OKadjust@ui-dep-7bfc9f4c75-hhlsd:/app$ exit
exit

C:\Users\Kanika>
```



```
Kanika@Kanika MINGW64 /e/ADJUST/git
$ curl http://192.168.99.100:31736
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    18    0    18    0     0   9000      0 --:--:-- --:--:-- --:--:-- 18000We
ll, hello there!
```