Background: You are working for a fintech company that is developing a complex .NET core 8 application called FinSmart. The application is a financial analytics platform that provides real-time data processing, analytics, and reporting. The architecture consists of multiple microservices, each responsible for different functionalities such as user authentication, transaction processing, data analytics, and reporting. Each microservice has its own database.

Requirements:

1. CI/CD with GitLab:

o Set up a GitLab CI/CD pipeline to automate the build, test, and deployment of the .NET application. o Ensure the pipeline includes stages for code quality checks, unit testing, integration testing, and deployment to a Kubernetes cluster.

>> Created a sample repository with a single code/service, and add the required code checks.

Link is below:

URL: <https://gitlab.com/tlogic1/dotnet1.git>

Code Quality:

include:

- template: Jobs/Code-Quality.gitlab-ci.yml

Unit Test:

add [artifacts:reports:junit](https://docs.gitlab.com/ee/ci/yaml/artifacts_reports.html#artifactsreportsjunit) in .gitlab-ci.yml, and specify the paths of the generated test reports

2. Azure Kubernetes Service (AKS):

o Deploy the microservices on an Azure Kubernetes Service (AKS) cluster.

o Ensure high availability and scalability of the services.

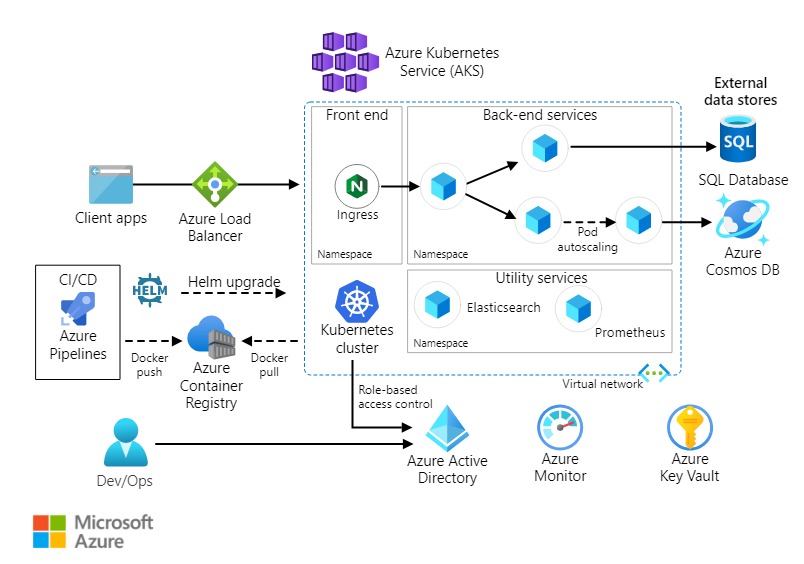
o Implement monitoring and logging for the AKS cluster using Azure Monitor and Log Analytics.

>> We will have separate yaml files for various microservices.

1. First will deploy the AKS cluster (make sure to enable monitoring),
2. Build the docker/container images, deployment/services yaml files and they do the deployment via the Helm for all these microservices.

>> image build >>>>>

Using helm chart will deploy them providing the parameters as required. Here is high level diagram:



3. Azure Landing Zone:

o Design and implement an Azure Landing Zone to establish a secure and scalable foundation for the AKS deployment.

o The Landing Zone should include network architecture, identity and access management, security, and governance policies.

>> Using the Terraform we can deploy it as per the requirement and included security and governance.

Adding the code here.

Here is the high level diagram, what we will be deploying (may change as per client requirements).

A diagram of a cloud

Description automatically generated

4. Resource Group Export/Import:

o Develop a process to export the resource group containing the AKS cluster and associated resources from the current Azure subscription to a new subscription in a different Azure account.

o Ensure that all dependencies and configurations are correctly transferred and the application remains functional in the new subscription

>> As we deployed it via the Terraform, we can tweak the code for the new subscription and deploy.

Configuration we can take a backup and restore there.

Option2: can do it via the ARM template. Download the template and deploy in the target subscription. Other non-aks resources can be moved using move option.