Azure DevOps

Contents

[1.0 Problem statement 2](#_Toc43025802)

[2.0 Skills to develop the project 2](#_Toc43025803)

[3.0 Architecture Diagram for the Problem Statement 3](#_Toc43025804)

[4.0 Use Case 4](#_Toc43025805)

[5.0 Milestone and duration 4](#_Toc43025806)

[6.0 Implementation Notes 5](#_Toc43025807)

[7.0 Evaluation rubrics 6](#_Toc43025808)

# Problem statement

The objective of the document is to provide associate with a case study for implementing DevOps concepts.

Customer is embarking on cloud transformation/migration journey and as part the transformation it would like meet following objectives

* Manage application code base in a Git repository
* Deploying Mongo express web interface on Kubernetes cluster
* Use ARM templates and configure the Azure infrastructure using Azure Devops

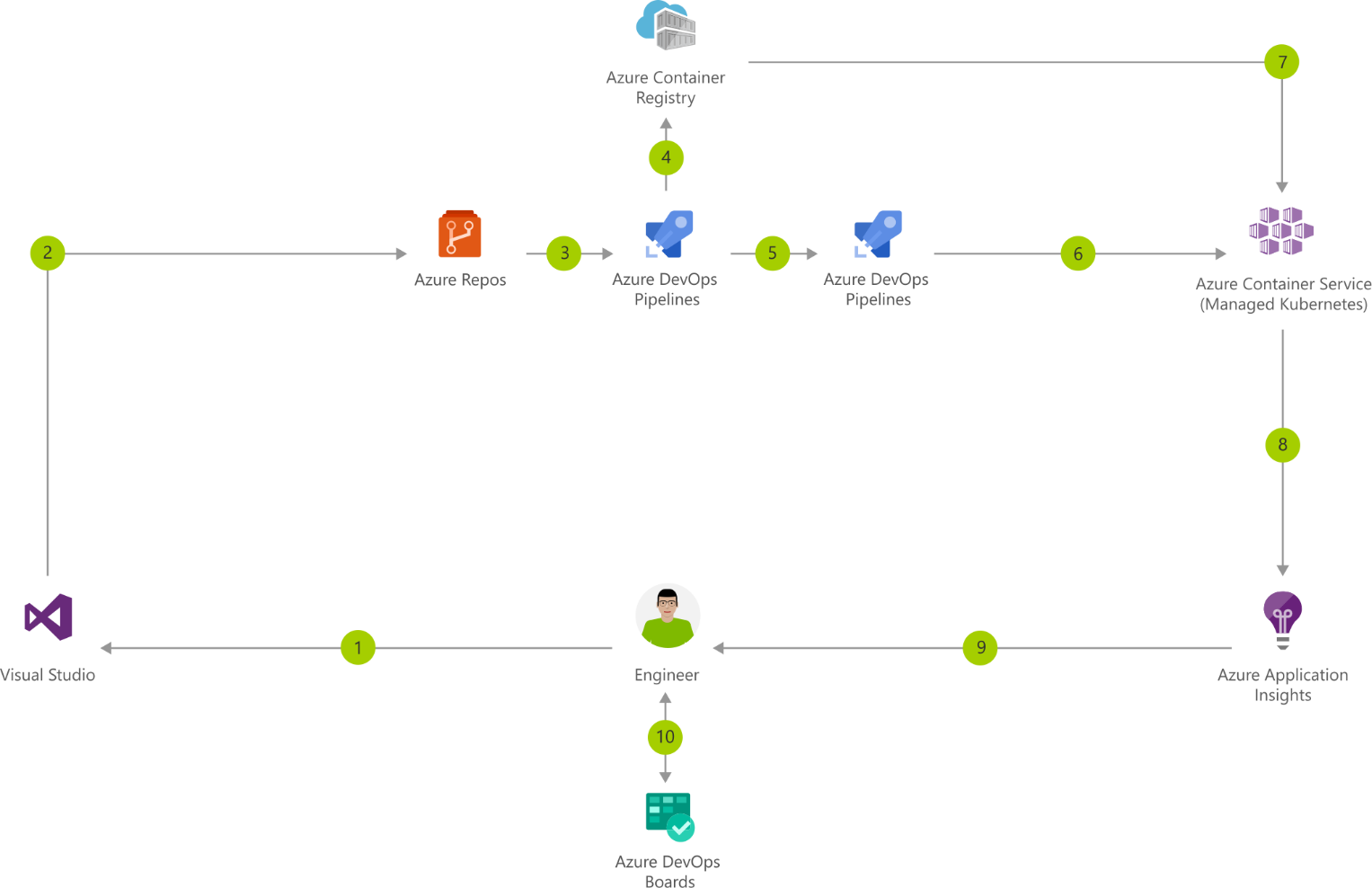
# Skills to develop the project

Associate will perform the DevOps activities using the below skills

Below are the skill details.

|  |  |
| --- | --- |
| **Tower Name** | **Skill Names** |
| DevOps | Git  Docker  Kubernetes(AKS)  Mongo Express  HELM  ARM Templates  Azure DevOps |

# Architecture Diagram for the Problem Statement



# Use Case

* Automate build and deployment for Mongo Express into Azure Kubernetes services

# Milestone and duration

|  |  |
| --- | --- |
| Milestone | Topic |
| Milestone -1 | * Basic Git commands to create repo, branches, get code, commit, merge * Use ARM templates and configure the Azure infrastructure using Azure Devops * All Azure resources should be tagged inside a Resource Group * Use ARM templates to configure the following resources   + Virtual Network   + Subnet   + Kubernetes Cluster   + use Azure CNI for the Cluster * Use Helm charts to create the artifacts for the application pods and services to be deployed. Docker image has to be pushed to the Azure Container registry |
| Milestone -2 | * Create two pods one for hosting the Mongo express app and the other one for Mongo DB * Two services to be created for the pods to connect, restrict the service to internal while connecting from the DB to the Mongo Express. * Other service to have the user interface of the Mongo express. This should be exposed as the external service on port 8084. * Configure Config Maps and Secret to configure the username, Password and connection string * Once established the pods and services, confirm the setup using the kubectl commands |

# Implementation Notes

|  |  |
| --- | --- |
| Milestone -1 | * Develop the relevant scripting for performing CI/CD process using any one of the scripting languages. |
| Milestone -2 | Perform CI/CD pipeline by using tools using Git, Azure DevOps and Kubernetes (AKS) shown below     * **GIT**   Code repository used - GIT   * **Azure DevOps**   Create Azure Container Registry   * **Kubernetes**   Create application cluster that includes deployment of service, pod, replication controller and replica set.   * **Helm**   Use Helm to create artifacts which needs to be pushed to ECR  Guidelines for AKS services:  AKS:  1. Azure Pipelines app. If so, select Approve and install.  2. Repo should have the Dockerfile and once changes are made, the images should be pushed to Azure Container Registry.  3. Manifests should be deployed to Azure Kubernetes Service cluster |
|  |  |

# Evaluation rubrics

|  |  |
| --- | --- |
| DevOps scripting | * Should have developed the relevant scripting for performing CI/CD process using any one of the scripting languages. |
| GIT | * Code repository - GIT |
| Azure DevOps | * Azure DevOps Build Pipeline * Azure DevOps Release Pipeline * Automate Azure DevOps Release Pipeline |
| Kubernetes | * Creation of AKS application cluster which includes deployment of service, pod, replication controller and replica set. |
| Screen shots | * Verify the logs and screenshots |

Create ARM Template

Commit to Repos

Create the pipeline

Create the agent in the default agent pool