**AZURE DEVOPS ASSESSMENT**

**NAME :** KAILASH S

**DATE :** 12-01-24

**QUESTION:**

Explain process of Azure Dev ops CICD pipeline & Explain azure SQL server pool integration with azure synapse

**SOLUTION:**

**1.**

**AZURE DEVOPS:**

Azure DevOps is a set of development tools and services provided by Microsoft to support the entire DevOps lifecycle. It encompasses a range of features for source code management, build automation, release management, and more. The main components of Azure DevOps are:

* Azure boards
* Azure repos
* Azure pipelines
* Azure test plans
* Azure artifacts

**AZURE CI/CD:**

Azure CI/CD refers to the practice of Continuous Integration and Continuous Deployment, which are key components of the software development lifecycle. CI/CD helps automate and streamline the process of building, testing, and deploying applications, resulting in faster and more reliable software releases.

**CONTINUOS INTEGRATION:**

Continuous Integration (CI) is a software development practice that involves automatically integrating code changes from multiple contributors into a shared repository multiple times a day. The primary goal of CI is to detect and address integration issues early in the development process, ensuring that the software remains in a working state at all times.

**CONTINUOS DEPLOYMENT:**

Continuous Deployment (CD) is an extension of Continuous Integration (CI) in the DevOps process. While CI focuses on integrating code changes and running automated tests, CD takes it a step further by automatically deploying those changes to production or staging environments if they pass all tests. The primary goal of Continuous Deployment is to deliver new features, enhancements, or bug fixes to end-users as quickly and reliably as possible.

**Prerequisites**:

* An Azure DevOps account.
* A repository containing application code, hosted on a version control system like Git.

**Steps to Create an Azure CICD Pipeline**:

**STEP 1: Access Azure DevOps**: Log in to your Azure DevOps account.

**STEP 2: Create a New Pipeline:** Navigate to your project. Go to “Pipelines” from the left sidebar. Click on the “New Pipeline” button.

**STEP 3: Select a Repository**: Choose the repository that contains your application code. Select the appropriate source control system (e.g., Git).

**STEP 4: Configure Pipeline Settings**: Choose a template that matches your application’s technology stack. If a template isn’t available, you can choose the “Starter pipeline” option and define your pipeline using YAML or the visual designer.

**STEP 5: Define Build and Test Stages**: Configure the build stage to compile your code, restore dependencies, and generate build artefacts. Add steps to run tests, code analysis, and any other required quality checks.

**STEP 6: Define Deployment Stages**: Set up deployment stages for each target environment. Configure deployment tasks to deploy your application to the respective environments using Azure resources such as Azure App Service, Azure Kubernetes Service etc.

**STEP 7: Configure Triggers:** Define triggers to automatically start the pipeline when changes are pushed to specific branches, pull requests are created, or on a schedule.

**STEP 8: Add Variables and Secrets**: Define variables to store your pipeline’s configuration settings. Securely store sensitive information, such as API keys or connection strings, in the pipeline’s secret store.

**STEP 9: Review and Save**: Review the pipeline configuration to ensure everything is set up correctly. Save and commit the pipeline configuration to your repository.

**STEP 10: Run and Monitor the Pipeline**: Trigger the pipeline manually or let it be triggered automatically based on the defined triggers. Monitor the pipeline’s progress and review logs to identify any issues.

**STEP 11: Customise and Iterate**: As your application and requirements evolve, you can customize and iterate on your pipeline by adding more stages, tasks, or adjustments to the configuration.

**2.**

**AZURE SYNAPSE:**

Azure synapse analytics is a unified analytics platform that brings together enterprise data warehousing and big data analytics. It gives the freedom to query data on your terms, using either server-less or provisioned resources at scale. Synapse provides a unified platform forprocessing, managing, and serving data for immediate business intelligence and data prediction needs by integrating with power BI and azure machine learning.

**SQL SERVER POOL:**

Azure Synapse SQL Server Pool is a cloud-based data warehouse service provided by Microsoft Azure. It is designed for handling large volumes of data and performing complex analytical queries.

**PREREQUISITES:**

* Azure subscription
* Azure synapse
* Azure data lake storage

**SYNAPSE AND SERVERPOOL INTEGRATION:**

**Step-1:**

**Creating synapse workspace**

* Open azure portal and launch azure synapse workspace.
* Search azure synapse analytics
* Select add to create workspace
* Select the storage account
* Select review and create

**Step-2:**

**Configure storage account with workspace**

* Open azure storage account and select access control.
* Select add> role assignment
* Assign azure roles

**Step-3:**

* Launch synapse studio
* Navigate to management hub
* Open SQL pools

**Step-4:**

* Select new in SQL pool window
* Give pool name and select performance level
* Select review and create

Once the dedicated SQL pool is created, it will be available in the workspace for loading data, processing streams, reading from the lake, etc.

**Deleting SQL pool:**

* Go to SQL pools window
* Select the pool to be deleted
* Click delete and it will be deleted.