Kavya Gundala

12-01-2024

1. **Explain process of Azure DevOps cicd pipeline**

**DevOps:**

* DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams continuously.
* A DevOps team includes developers and IT operations working collaboratively throughout the product lifecycle, in order to increase the speed and quality of software deployment.

**CI/CD pipelines:**

* **Continuous integration (CI)** focuses on the early stages of a software development pipeline where the code is built and undergoes initial testing.
* Multiple developers work on the same codebase simultaneously and make frequent commits to the code repository. Build frequency can be daily or even several times per day at some points in the project's lifecycle.
* These small, frequent builds enable easy and low-risk experimentation, as well as the ability to easily roll back or abandon undesirable outcomes.
* CI ends when a build successfully completes initial testing and is ready to move to more comprehensive testing.
* **Continuous delivery (CD)** picks up where CI leaves off. It focuses on the later stages of a pipeline, where a completed build is thoroughly tested, validated and delivered for deployment. Continuous delivery can -- but does not necessarily -- deploy a successfully tested and validated build.
* CD likewise relies heavily on tools and automation to take a build through advanced testing, including functional, user acceptance, configuration and load testing.
* These validate that the build meets requirements and is ready for use in a production environment. Again, small incremental iterations ensure that any problems revealed in testing are identified and remediated quickly and less expensively than traditional software development approaches.

**The steps to create a CI/CD pipeline:**

Sign in to the organization ([https://dev.azure.com/{yourorganization}](https://dev.azure.com/%7byourorganization%7d)) and select the project.

**Create a CI Pipeline:**

In project, go to "Pipelines" in the left menu and select "Create Pipeline."

**Select Source:**

Choose the source where code resides (Azure Repos Git, GitHub, Bitbucket, etc.).

**Configure Pipeline:**

Choose an appropriate template or start with an empty job. Define build steps such as setting up the execution environment, installing dependencies, and running unit tests.

**Save and Run:**

Save the pipeline configuration and trigger a manual build to ensure the CI pipeline is working.

**Set Up Artifact (Build Output):**

**Configure Artifact:**

In your CI pipeline, publish artifacts that are the output of your build, which may include scripts or compiled binaries.

**Create a CD Pipeline:**

**Navigate to Pipelines:**

In your project, go to "Pipelines" again and select "Create Release Pipeline."

**Select Template:**

Choose a template suitable for deploying data engineering artifacts or start with an empty job.

**Configure Stages:**

Define stages representing different environments (Dev, Test, Production).

Specify deployment targets, which could include databases, data warehouses, or data processing platforms.

**Add Artifacts:**

Add the artifacts produced by the CI pipeline as the input for the CD pipeline.

**Configure Deployment Tasks:**

Define deployment tasks to execute your data engineering scripts or deploy data artifacts.

Account for any necessary data schema updates or migrations.

**Set Environment Variables:**

Configure environment-specific variables like connection strings or credentials.

**Configure Approvals and Triggers:**

Set up approvals for promoting releases to higher environments. Configure triggers for automatic deployments based on certain conditions.

**Deploy:**

Save the release pipeline configuration and create a release to deploy your data engineering artifacts.

**Monitor and Improve:**

**Monitor Pipelines:**

Monitor the CI/CD pipelines to ensure successful builds and deployments.

1. **Explain azure sql server pool integration with azure synapse.**

* Azure Synapse Analytics, previously known as Azure SQL Data Warehouse, is an analytics service provided by Microsoft Azure. It is a cloud-based, integrated analytics service that brings together big data and data warehousing. Azure Synapse Analytics is designed to handle large volumes of data for both analytical and transactional workloads, providing a unified platform for data storage, processing, and analysis.
* It gives you the freedom to query data on your terms, using either server-less or provisioned resources at scale.

**Creating Azure SQL server pool integration with Azure Synapse:**

1. Open the Azure portal, and at the top search for Synapse.

2. In the search results, under Services, select Azure Synapse Analytics.

3. Select Add to create a workspace.

4. In the Basics tab, give the workspace a unique name. We’ll use mysworkspace in this document

5. We need an ADLSGEN2 account to create a workspace.

6. Your Azure Synapse workspace will use this storage account as the “primary” storage account and the container to store workspace data.

The workspace stores data in Apache Spark tables. It stores Spark application logs under a folder called /synapse/workspacename.

7. Select Review + create to  Create. Your workspace is ready in a few minutes.

**Open Synapse Studio:**

Open your Synapse workspace in the Azure portal. On the top of the Overview section, select Launch Synapse Studio.

**Prepare an existing storage account for use with Azure Synapse Analytics:**

1. Open the Azure portal.

2. Navigate to an existing ADLSGEN2 storage account

3. Select Access control (IAM).

4. Select Add and Add role assignment to open the Add role assignment page.

5. Assign the Owner and Storage Blob Data Owner role. Assign access to USER. Members are your user name.

6. On the left pane, select Containers and create a container.

7. You can give the container any name. In this document, we’ll name the container users.

8. Accept the default setting Public access level, and then select Create.

**Configure access to the storage account from your workspace:**

1. Open the Azure portal and the primary storage account chosen for your workspace.

2. Select Access control (IAM).

3. Select Add and Add role assignment to open the Add role assignment page.

4. Assign the Storage Blob Data Contributor role. Assign access to MANAGEDIDENTITY. And members are myworkspace.

5. Select Save.

**Create a dedicated SQL pool in Synapse Studio:**

1. Navigate to the Synapse workspace to create SQL pool.
2. A list of workspaces will be opened, select a workspace.
3. Launch the Synapse Studio.
4. On the Synapse studio, navigate to Management Hub.
5. Select SQL pools to see available current list of pools.
6. Create New and the SQL pool create wizard will appear.
7. Give a SQL pool name and its Performance level.
8. Select Review+Create.
9. Once the SQL Server pool is created it will be available in the workspace for loading data, processing streams, reading data from lake, etc.