**Manage and Modularize Tasks and Templates in Azure DevOps Pipelines**

**1. Introduction**

Azure DevOps Pipelines provide a robust CI/CD automation framework for deploying applications efficiently. Managing and modularizing tasks and templates enhances reusability, maintainability, and scalability.

**2. Managing and Modularizing Tasks**

**a. Using YAML Templates**

Templates in YAML pipelines allow you to reuse task configurations across multiple pipelines.

**Task Grouping with Templates**

* **Step Templates**: Encapsulate a set of steps in a reusable YAML file.
* **Job Templates**: Define reusable job configurations.
* **Stage Templates**: Allow the reuse of entire stages.

**Example: Reusable Task Template**

# templates/build-template.yml

parameters:

- name: buildAgent

type: string

tasks:

- task: UseDotNet@2

inputs:

packageType: 'sdk'

version: '6.x'

- script: dotnet build

# azure-pipelines.yml

stages:

- stage: Build

jobs:

- job: BuildJob

steps:

- template: templates/build-template.yml

parameters:

buildAgent: 'ubuntu-latest'

**b. Managing Complex Pipelines with Task Groups**

Task groups allow encapsulating multiple tasks in a single reusable entity.

**Steps to Create a Task Group:**

1. Navigate to **Azure DevOps** -> **Pipelines** -> **Task Groups**.
2. Create a new task group and select tasks to be grouped.
3. Save and reuse the group across multiple pipelines.

**3. Create and Manage Variable Groups**

Variable groups in Azure DevOps centralize the management of variables across multiple pipelines.

**Creating a Variable Group**

1. Navigate to **Azure DevOps** -> **Library** -> **Variable Groups**.
2. Click **Add Variable Group** and enter a name.
3. Add variables (normal or secret) and assign values.
4. Link the variable group to the pipeline using YAML or Classic UI.

**Example: Using a Variable Group in YAML**

variables:

- group: MyVariableGroup

jobs:

- job: Deploy

steps:

- script: echo $(MySecretVar)

**4. Explore Variables in Release Pipelines**

Azure DevOps provides various types of variables for controlling pipeline behavior.

**a. Predefined Variables**

Azure DevOps offers system-defined variables available during runtime. Example:

steps:

- script: echo $(Build.BuildNumber)

**b. Release Pipeline Variables**

Defined at the pipeline level and accessible across all stages.

Example:

variables:

environmentName: 'production'

**c. Stage Variables**

Defined at the stage level, affecting only that stage.

Example:

stages:

- stage: Deploy

variables:

stageVar: 'DeployStage'

jobs:

- job: DeployJob

steps:

- script: echo $(stageVar)

**d. Normal and Secret Variables**

* **Normal Variables**: Stored in plain text, accessible in logs.
* **Secret Variables**: Stored securely and not exposed in logs.

Example:

variables:

normalVar: 'HelloWorld'

secretVar: '$(MySecret)' # Secret variable stored in Azure DevOps