

In Azure Pipelines, you can share variables across pipelines using **variable groups**, or by outputting variables in one pipeline and consuming them in another pipeline. Here's a step-by-step example to demonstrate both approaches.

Approach 1: Using Variable Groups(static variables)

Example:

You have a **Variable Group** named AppConfig that contains:

- APP_NAME: MyApp
- APP_ENV: production
- APP_VERSION: 1.0.0
- APP_PORT: 8080

Pipeline YAML Example:

```
trigger:
- main

variables:
- group: AppConfig # Reference the variable group named 'AppConfig'

steps:
- script: |
    echo "Application Name: $(APP_NAME)"
    echo "Environment: $(APP_ENV)"
    echo "Version: $(APP_VERSION)"
    echo "Port: $(APP_PORT)"
  displayName: "Print Application Configuration"
```

Approach 2: Output Variables from One Pipeline to Another(Dynamic variables)

Scenario:

- **Pipeline 1** builds an application and outputs the build's version, artifact name, and environment details.
- Use the `task.setvariable` logging command to define variables and store them in the **pipeline1** run's context.
- **Pipeline 2** deploys the application using the output from **Pipeline 1**.
- Use the `resources.pipelines` feature to link the **Pipeline 1** and access its output variables.

Pipeline 1: Build Pipeline (*build-pipeline.yml*)

```
trigger:
- main

pool:
  vmImage: 'ubuntu-latest'

steps:
- script: |
    echo "##vso[task.setvariable variable=BUILD_ARTIFACT_NAME;isOutput=true]my-app-artifact"
    echo "##vso[task.setvariable variable=BUILD_VERSION;isOutput=true]1.0.0"
    echo "##vso[task.setvariable variable=DEPLOY_ENV;isOutput=true]production"
  name: OutputVariables
  displayName: "Set and Output Build Variables"
```

Pipeline 2: Deploy Pipeline (*deploy-pipeline.yml*)

```
trigger:
- main

resources:
  pipelines:
  - pipeline: buildPipeline # Reference Pipeline 1
    source: build-pipeline # Name of Pipeline 1 YAML file
    trigger: true

steps:
- script: |
    echo "Deploying Artifact:
${{ resources.pipelines['buildPipeline'].outputs['OutputVariables.BUILD_ARTIFACT_NAME'] }}"
    echo "Version:
${{ resources.pipelines['buildPipeline'].outputs['OutputVariables.BUILD_VERSION'] }}"
    echo "Environment:
${{ resources.pipelines['buildPipeline'].outputs['OutputVariables.DEPLOY_ENV'] }}"
    displayName: "Deploy Application"
```

Explanation of Variables

1. **APP_NAME**: The name of your application (MyApp).
2. **APP_ENV**: The deployment environment (production, staging, development).
3. **APP_VERSION**: The version of the application being built or deployed.
4. **APP_PORT**: The port where the application will run.
5. **BUILD_ARTIFACT_NAME**: The name of the built artifact (e.g., my-app-artifact).
6. **BUILD_VERSION**: The version of the build being deployed.
7. **DEPLOY_ENV**: The environment to which the application is deployed.

Key Notes:

- **Variable Groups** are easier for static variables shared across pipelines.
- **Output Variables** are better for dynamic data generated at runtime.