Agile Devops Tools – Azure Release Management

[**✅ Task 6: Release Management in Pipelines – Delivery** 1](#_Toc205558348)

[**1. 🎯 Purpose** 1](#_Toc205558349)

[**2. 📘 Theory** 1](#_Toc205558350)

[**3. 🧱 Prerequisites** 2](#_Toc205558351)

[**4. 💻 Step-by-Step Code / UI Walkthrough** 2](#_Toc205558352)

[**5. 📂 Project Snapshot** 3](#_Toc205558353)

[**6. 📝 Summary** 3](#_Toc205558354)

**✅ Task 6: Release Management in Pipelines – Delivery**

**1. 🎯 Purpose**

Release Management helps you **deploy builds to different environments** (Dev, QA, UAT, Prod) with **approval, schedules, and rollback options**. It ensures **controlled, reliable delivery** of software in a CI/CD pipeline.

**2. 📘 Theory**

**What is Release Management in Azure DevOps?**

Release Management is a process that takes **artifacts from a build pipeline** and **deploys them to various environments** with defined approvals and conditions.

**Key Concepts:**

* **Release Pipeline**: A pipeline that defines **deployment steps**, **approvals**, and **environment-specific configurations**.
* **Stages/Environments**: E.g., Dev → QA → UAT → Prod.
* **Approvals**: Manual intervention before deployment to sensitive environments.
* **Artifacts**: Outputs from a build pipeline (e.g., zip, DLLs, executables).

**Use Cases:**

* Automatically deploy code to dev after every build.
* Manually approve UAT deployment by a tester or product owner.
* Schedule Production deployment on weekends at midnight.

**Difference: Build vs Release Pipelines**

| **Build Pipeline** | **Release Pipeline** |
| --- | --- |
| CI (Continuous Integration) | CD (Continuous Delivery/Deployment) |
| Compile, test, package | Deploy to environments |
| Triggered on code commit | Triggered on build artifact creation |

**3. 🧱 Prerequisites**

* Existing build pipeline that produces an artifact.
* Azure DevOps project and permissions to create release pipelines.
* Deployment target (e.g., Azure App Service or local IIS).

**4. 💻 Step-by-Step Code / UI Walkthrough**

**✅ Step 1: Create a Build Pipeline (If not already)**

Build pipeline should output a zip or deployable artifact.

**✅ Step 2: Go to "Pipelines" → Releases → Click "New Pipeline"**

**✅ Step 3: Select the Artifact**

* Click **Add an artifact**
* Choose your **build pipeline** as the source
* Select the latest version or trigger on new build

**✅ Step 4: Add a Stage**

* Click **Add a stage** → Select **Azure App Service deployment**, IIS deployment, or Empty job
* Name the stage (e.g., Dev, QA, Production)

**✅ Step 5: Configure Deployment Tasks**

* In the stage → Add task (e.g., Azure App Service Deploy)
* Provide App Service name, resource group, package location (from artifact)

**✅ Step 6: Add Approvals and Conditions**

* Click on the stage → **Pre-deployment conditions**
* Add required **approvers** before deployment starts

**✅ Step 7: Enable Triggers**

* Enable **"After release creation"** or **"After build artifact"** to automate release creation

**✅ Step 8: Create and Deploy Release**

* Click **Create Release** → Select stages
* Click **Deploy** → Observe logs and status

**5. 📂 Project Snapshot**

Azure DevOps

│

├── Pipelines

│ ├── Build Pipelines

│ │ └── Build-App-Pipeline (Produces Artifact.zip)

│ └── Release Pipelines

│ └── Release-App

│ ├── Artifact: Build-App-Pipeline

│ ├── Stage: Dev (Auto deploy)

│ ├── Stage: QA (Manual approval)

│ └── Stage: Production (Scheduled release)

**6. 📝 Summary**

Release Management in Azure DevOps enables **automated and controlled deployment** of your applications to multiple environments. With **stages, approvals, and artifacts**, it brings safety and repeatability to your **CD (Continuous Delivery)** process.