TASK-6

YAML

Azure DevOps provides two primary ways to build pipelines: the Classic Editor and YAML. Each method offers distinct workflows and advantages.

**🔹 What is YAML?**

**YAML** stands for "**YAML Ain’t Markup Language**" (a recursive acronym). It's a **human-readable data serialization format**, used primarily for:

* Configuration files (e.g., in Docker, Kubernetes, GitHub Actions)
* Data exchange between languages with different data structures
* Writing structured data in a readable format

🔹 Key Features of YAML

| **Feature** | **Description** |
| --- | --- |
| **Human-readable** | Clean and easy to write/read for humans |
| **Language-agnostic** | Works across many programming languages |
| **Hierarchical** | Uses indentation to represent data structure (like Python) |
| **Supports complex types** | Lists, dictionaries, strings, numbers, booleans, nulls |
| **Comment support** | Allows inline comments with # |

🔹 Basic Syntax

1. **Key-Value Pairs**

name: Alice

age: 30

1. **Lists:**

fruits:

- Apple

- Banana

- Orange

3. **Dictionaries (Nested)**

person:

name: Alice

age: 30

address:

city: New York

zip: 10001

**4.Multiple Documents**

---

name: Document1

value: A

---

name: Document2

value: B

**🔹 Indentation Rules**

* **Always use spaces**, not tabs.
* Indentation defines **structure**.
* Misaligned indentation causes parsing errors.

**🔹 Common Mistakes**

* Mixing tabs and spaces
* Improper indentation
* Forgetting dashes in lists
* Using colons inside unquoted strings (:, - are special)

**🔹 What Are YAML Scalars?**

In YAML, a **scalar** is a **single, primitive value**. It is the most basic kind of data — like a string, number, boolean, or null.

Scalars are the building blocks for more complex structures like lists and maps.

**🔹 Types of Scalar Values**

**1. Strings**

name: Alice

quote: "Hello, world!"

note: 'Single-quoted text'

message: This is a plain string

**2. Numbers**

age: 30

price: 99.99

**3. Booleans**

isActive: true

isAdmin: false

**4. Null**

value: null

optional: ~

**🔹 2. YAML List (aka Sequence)**

A **list** is an ordered collection of values.

fruits:

- Apple

- Banana

- Mango

**🔹 3. YAML Mapping (aka Dictionary or Key-Value pairs)**

A **mapping** is a collection of **key-value pairs**.

**✅ Syntax:**

person:

name: John

age: 30

city: New York

**🔹 4. YAML Multiline Scalar (Block Scalars)**

YAML supports two styles for multiline strings:

**▶ Literal Style (|)**

Preserves newlines **exactly as written**.

description: |

Line one

Line two

Line three

**🔹 YAML Hierarchy in CI/CD Pipelines**

The **hierarchy in YAML files** for defining **CI/CD pipelines**, specifically with terms like:

* **Stages**
* **Jobs**
* **Steps**
* **Tasks**

**✅ Full Hierarchy Overview**:

pipeline

├── stages

│ ├── stage

│ │ ├── jobs

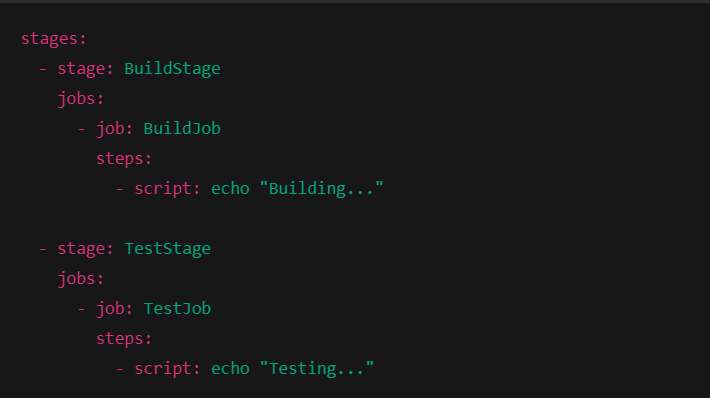
│ │ │ ├── job

│ │ │ │ ├── steps

│ │ │ │ │ ├── task or script

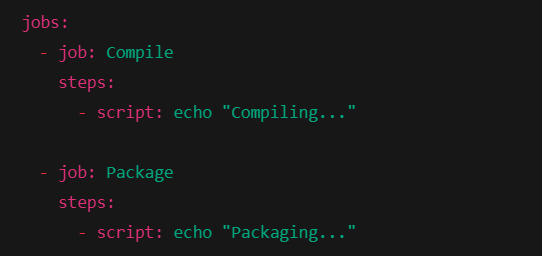
**🔸 1. Stages**

* Top-level division of a pipeline.
* Used to separate parts of the process (e.g., build, test, deploy).



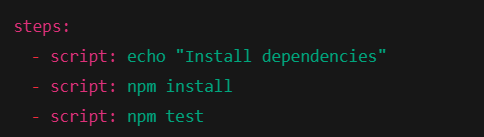
**🔸 2. Jobs**

* A stage contains one or more **jobs**.
* Each job runs on an agent (VM/runner).
* Jobs run **in parallel** by default unless dependencies are set.



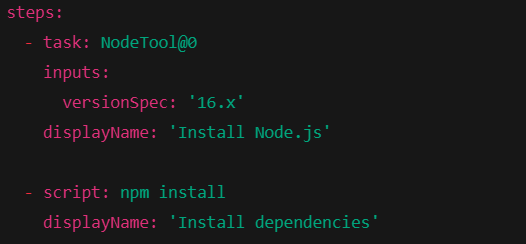
**🔸 3. Steps**

* A job contains one or more **steps**.
* Steps run **sequentially** within a job.



**🔸 4. Tasks**

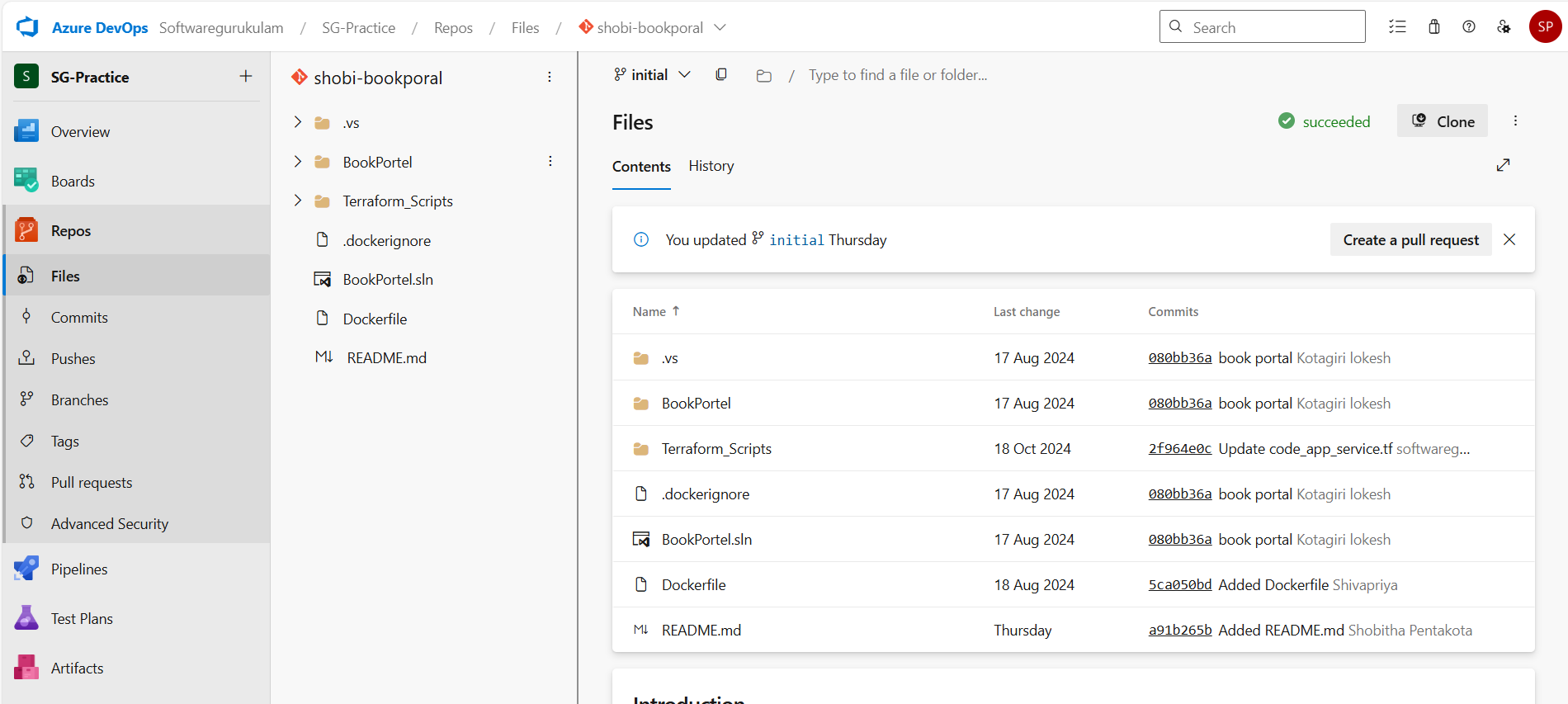
* A task is a reusable, pre-defined step (like NodeTool, PublishBuildArtifacts, etc.)
* Steps can be either **script** or **task**

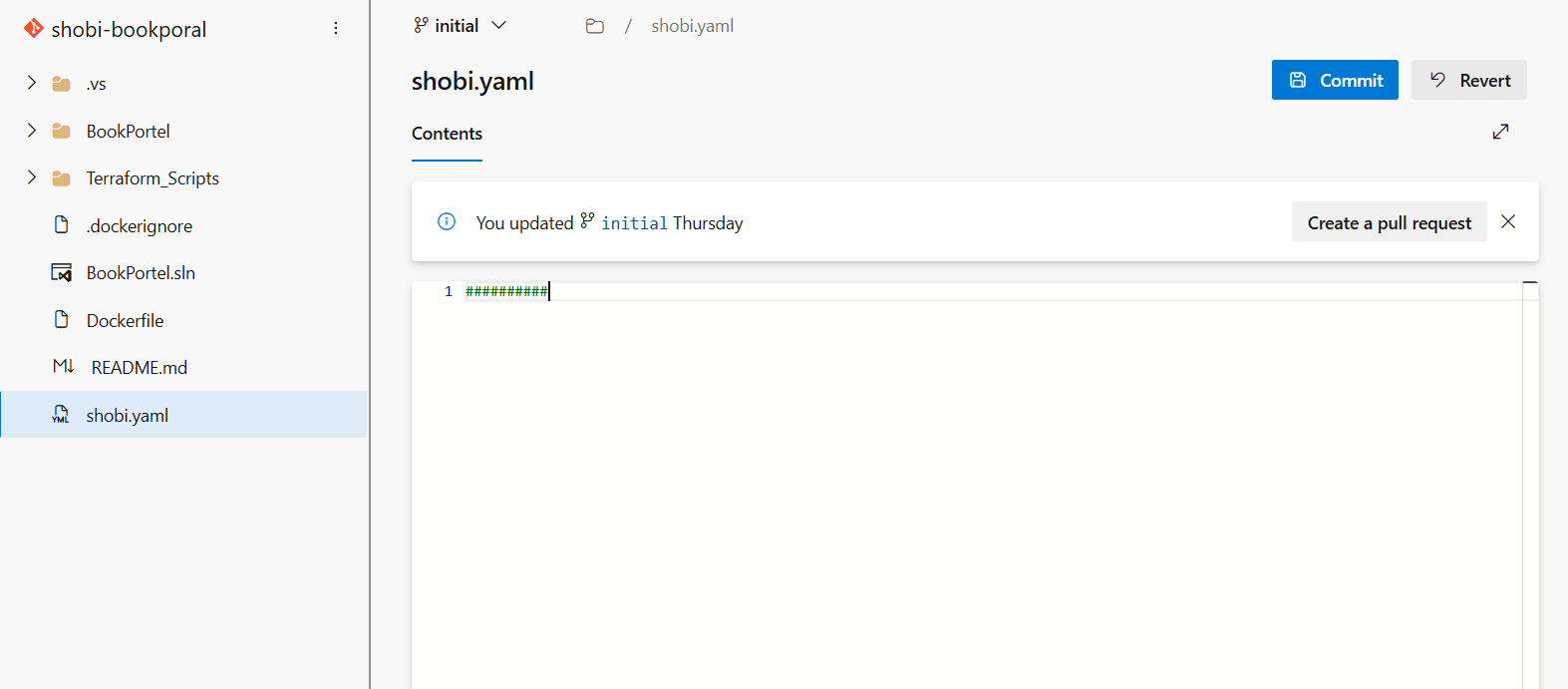


**🚀 step-by-step guide to building and deploying a .NET application using YAML in Azure DevOps:**

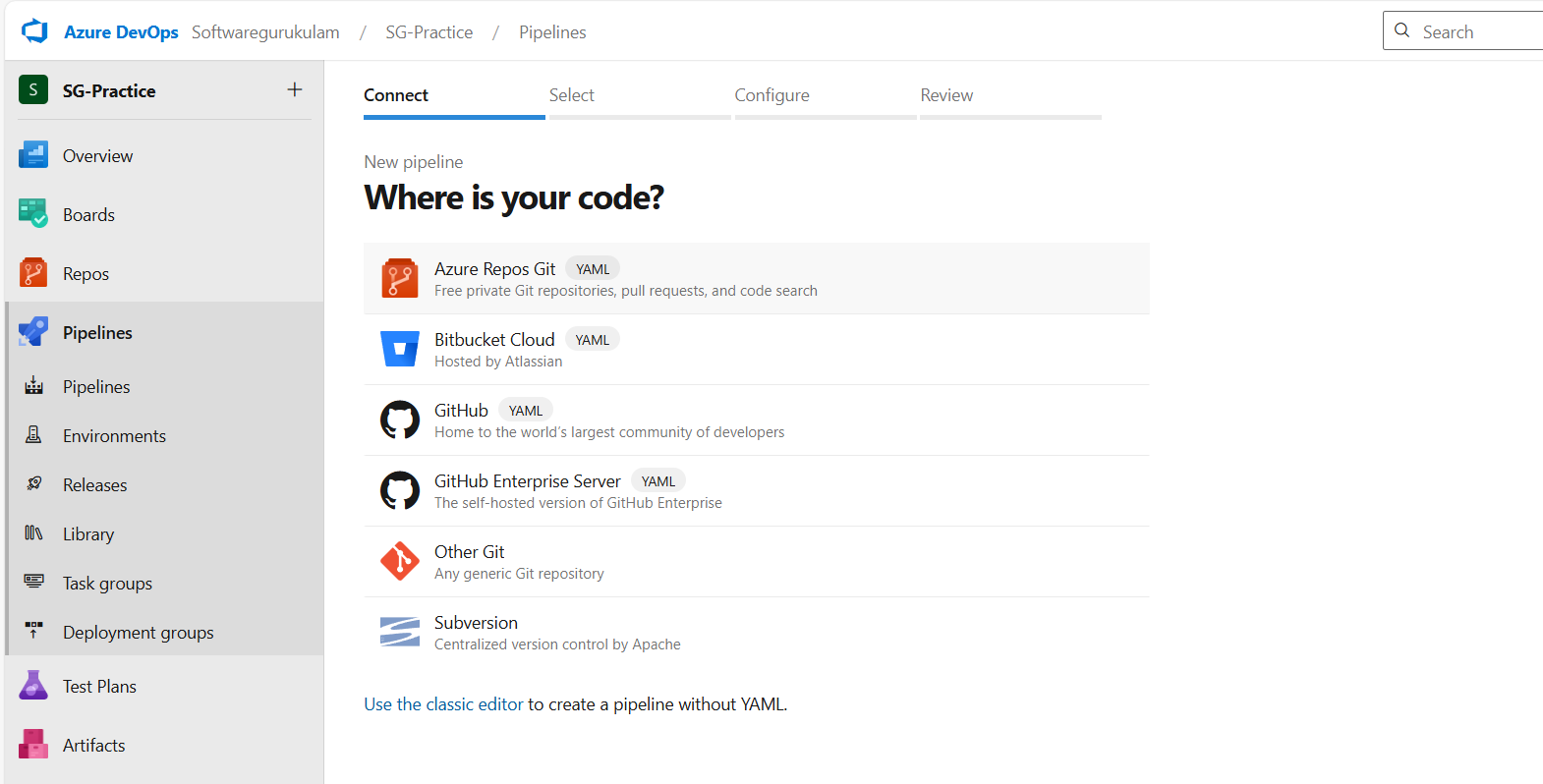
**🔸 Step 1: Create a new YAML Pipeline**

* select your repo …..create shobi.yaml file and do commit**.**

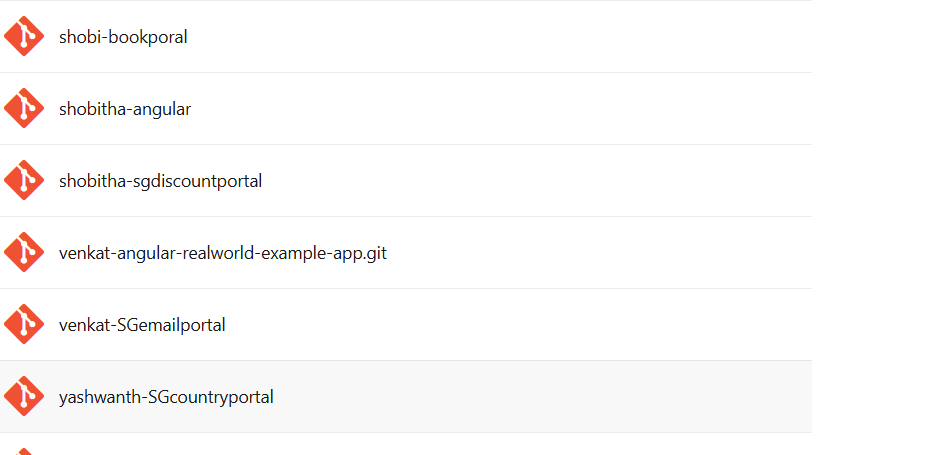
****

****

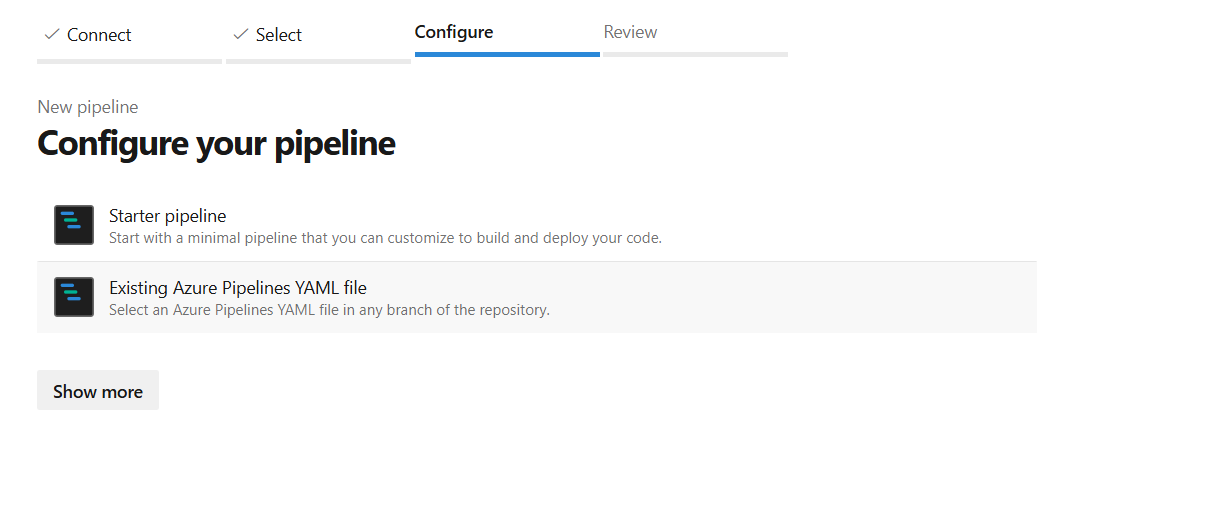
1. Go to Azure DevOps → Pipelines
2. Click "New Pipeline"
3. Choose "Azure Repos Git" (or GitHub)



1. Choose "Azure Repos Git" (or GitHub)
2. Select your repository



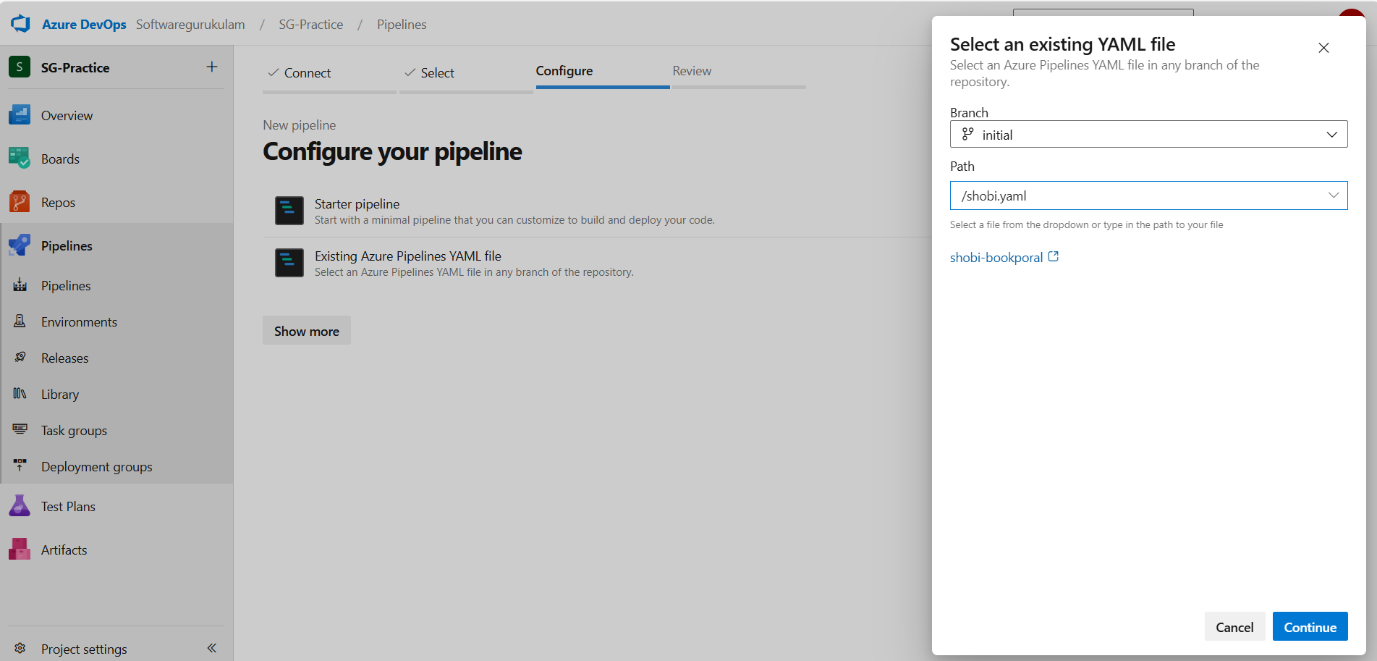
1. Choose "YAML"
2. Select "Starter pipeline" or click "Existing YAML file" if you already created one



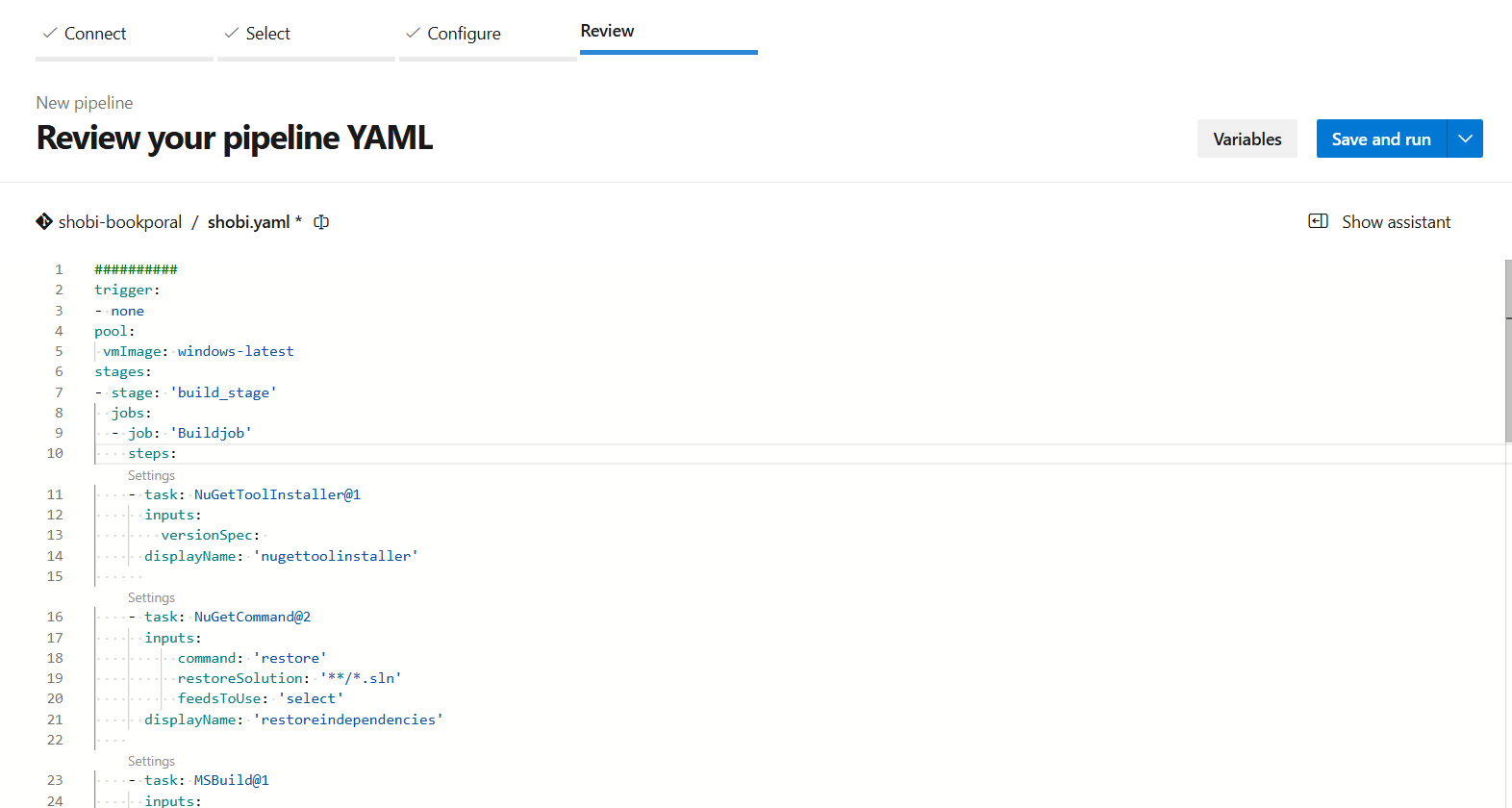
**🔸 Step 2:**

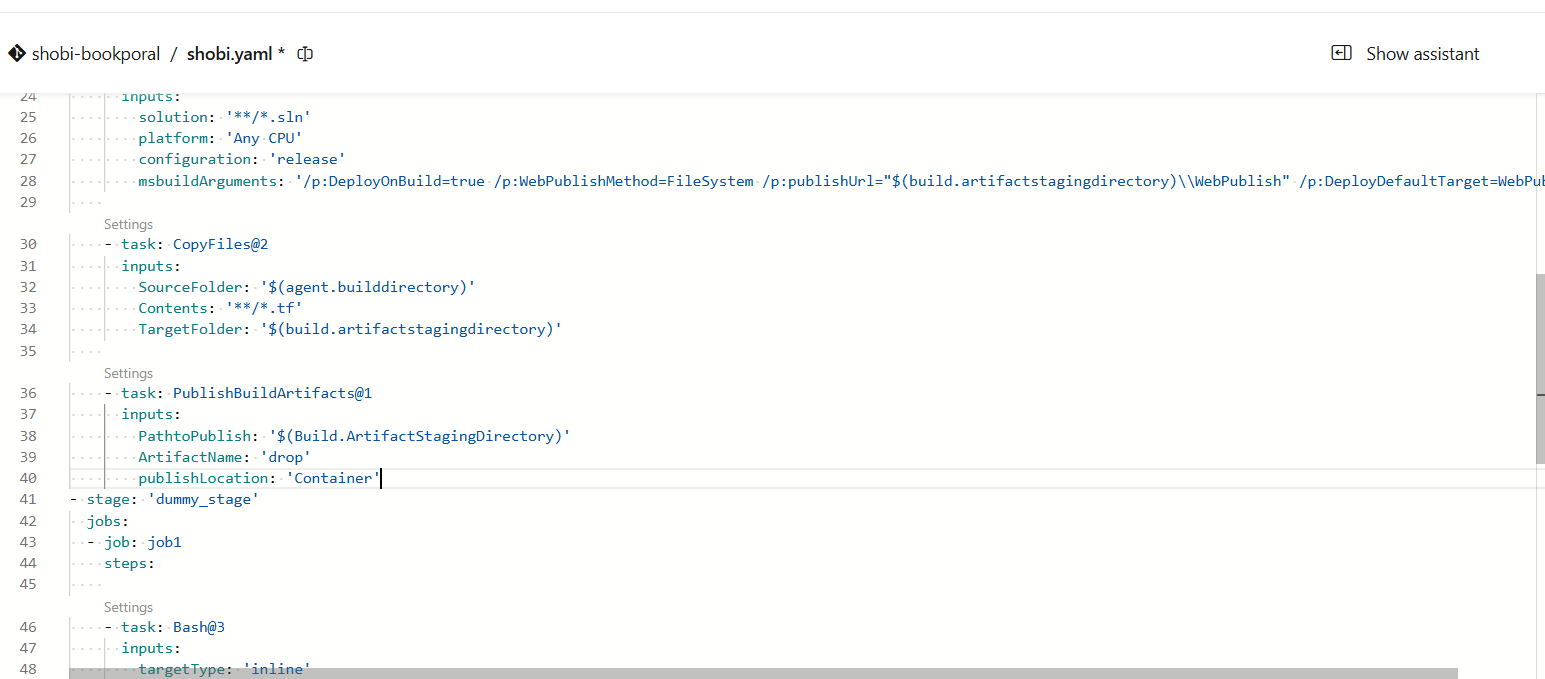
**1)**After selecting existing yaml file.

* Select branch and path.



* Click continue
* Wirte yaml script.

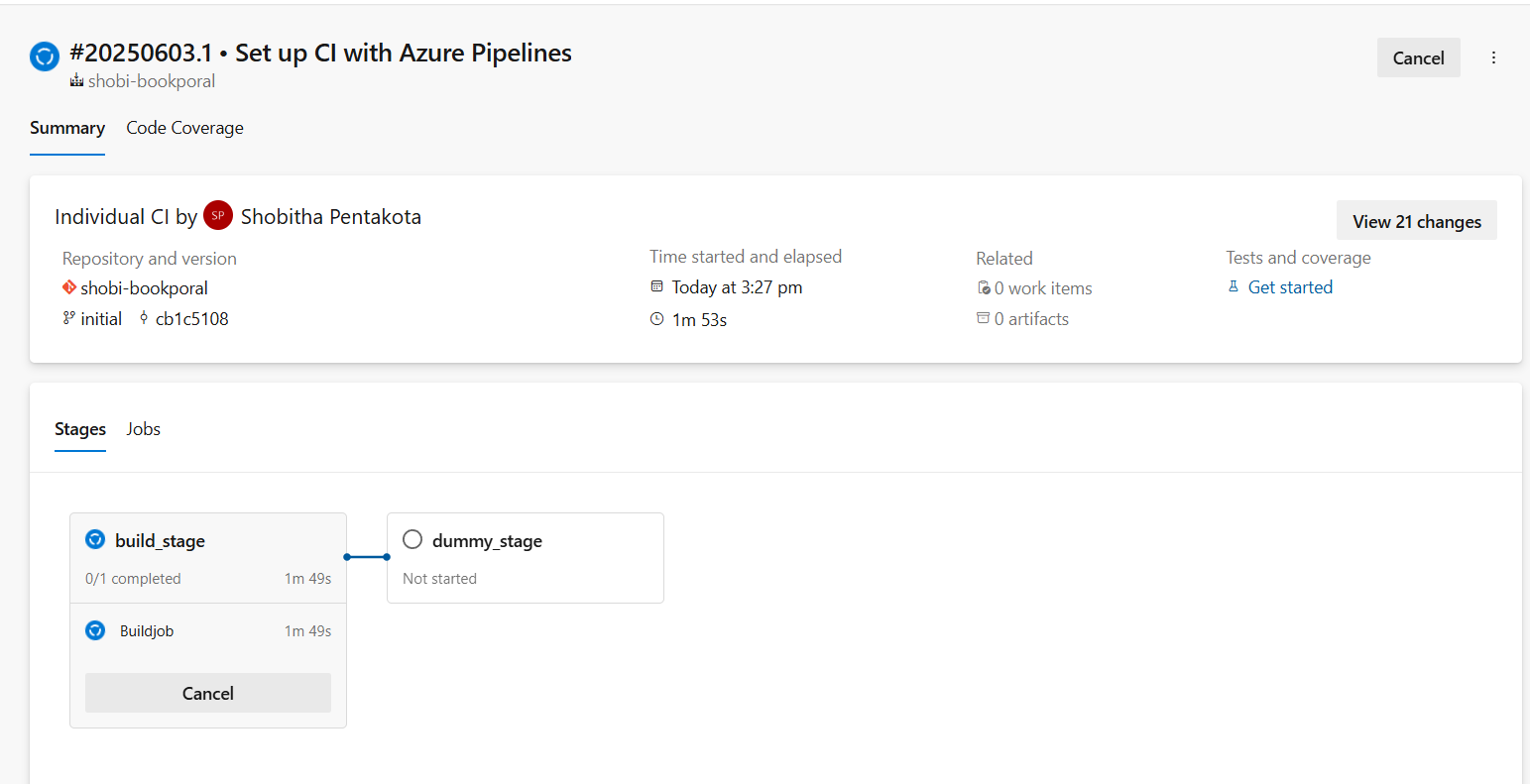


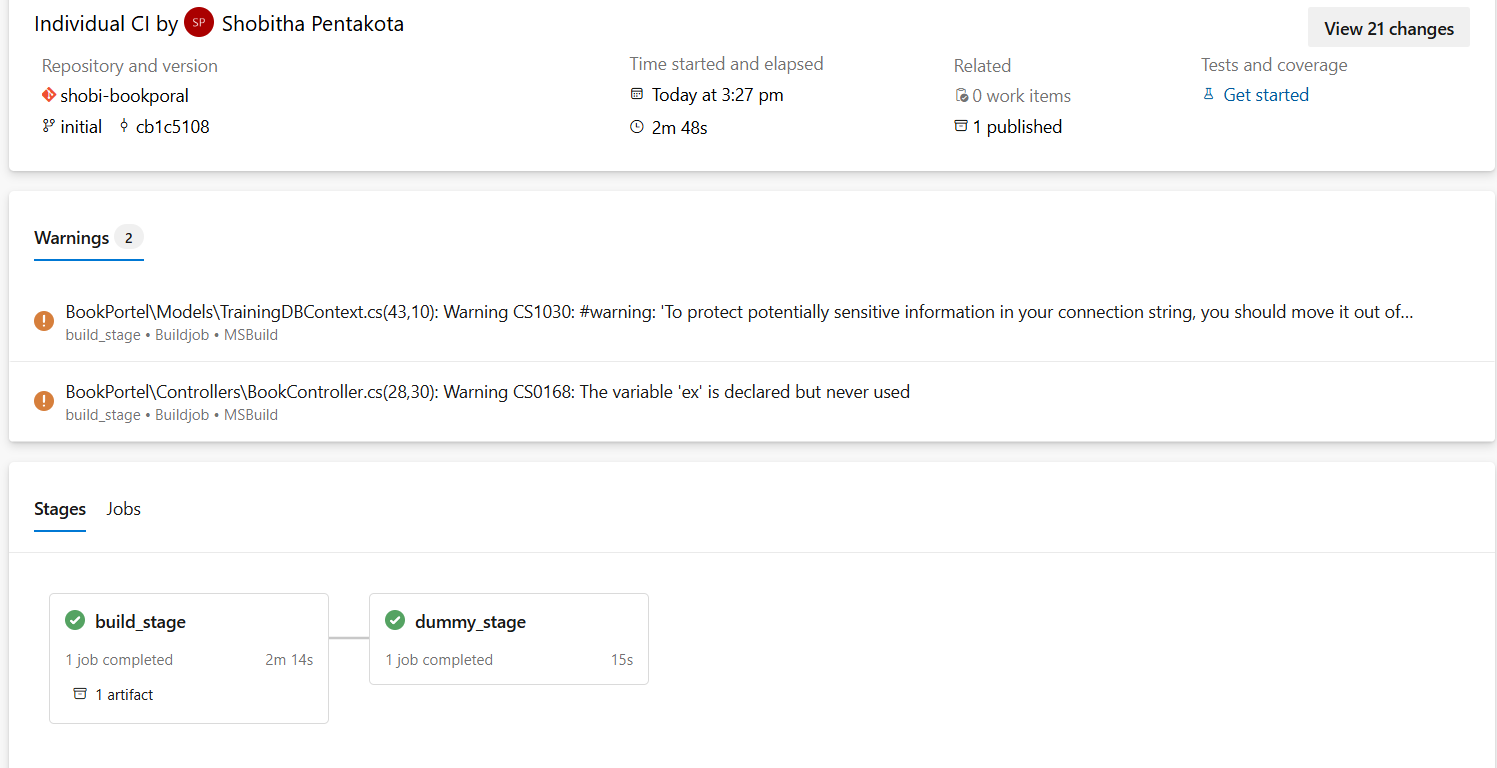




**🔸 Step 3: Save the pipeline**

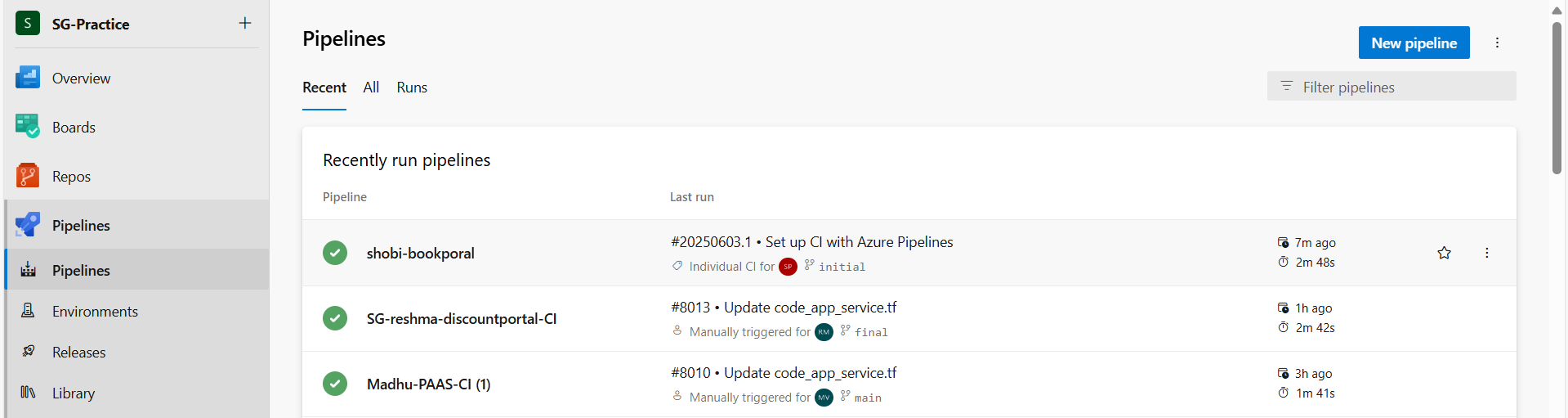
1. Click **Save and run**
2. Confirm the branch (usually main)
3. The pipeline will now start running automatically



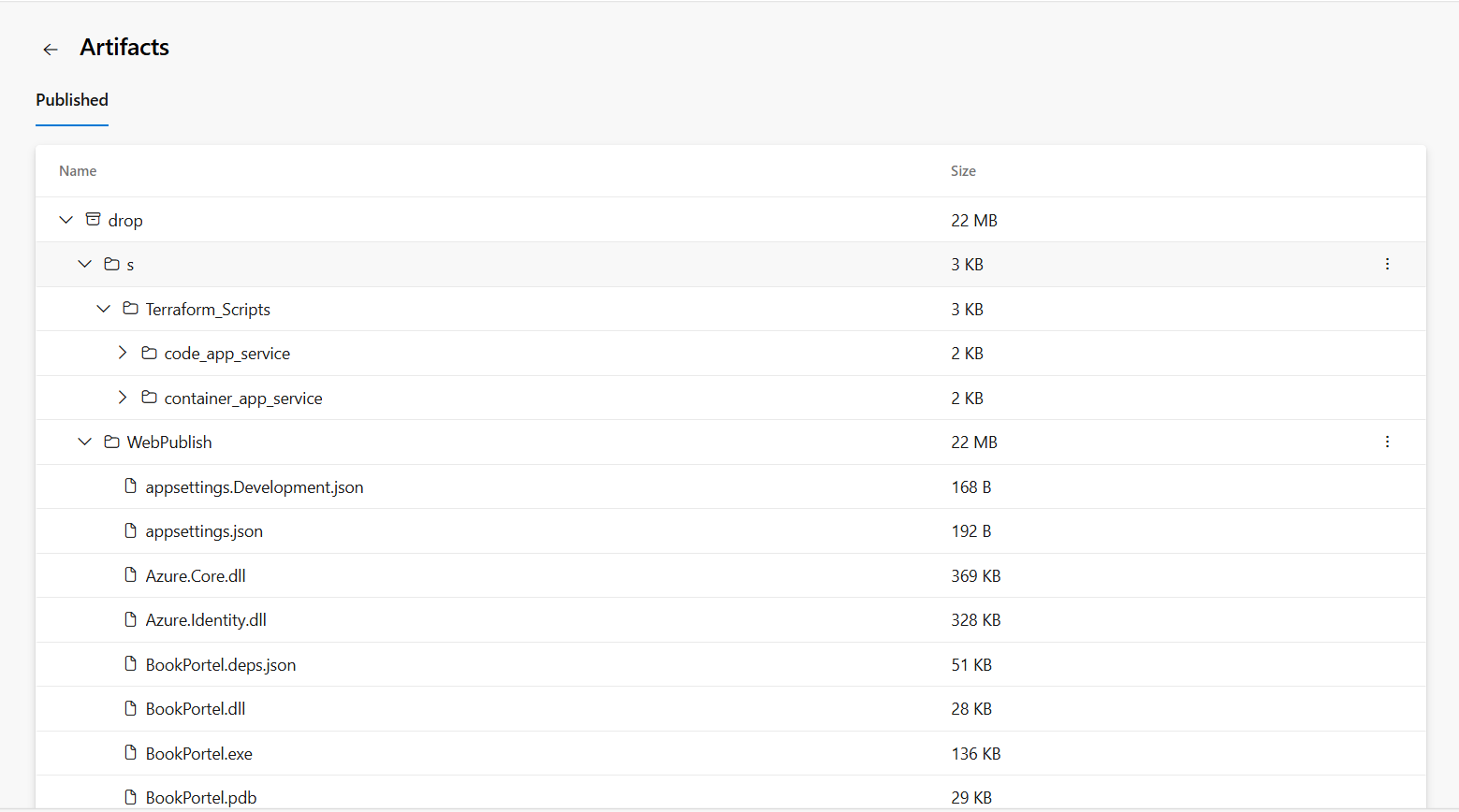


**🔸 Step 4: Monitor the Pipeline**

* Go to **Pipelines > [Your pipeline name]**
* View the **logs**, **build summary**, and **test results.**



🔸 Add Artifact Publishing (For Deployments):we already added artifacts.



* We deploy our artifacts in dummy stage….

