

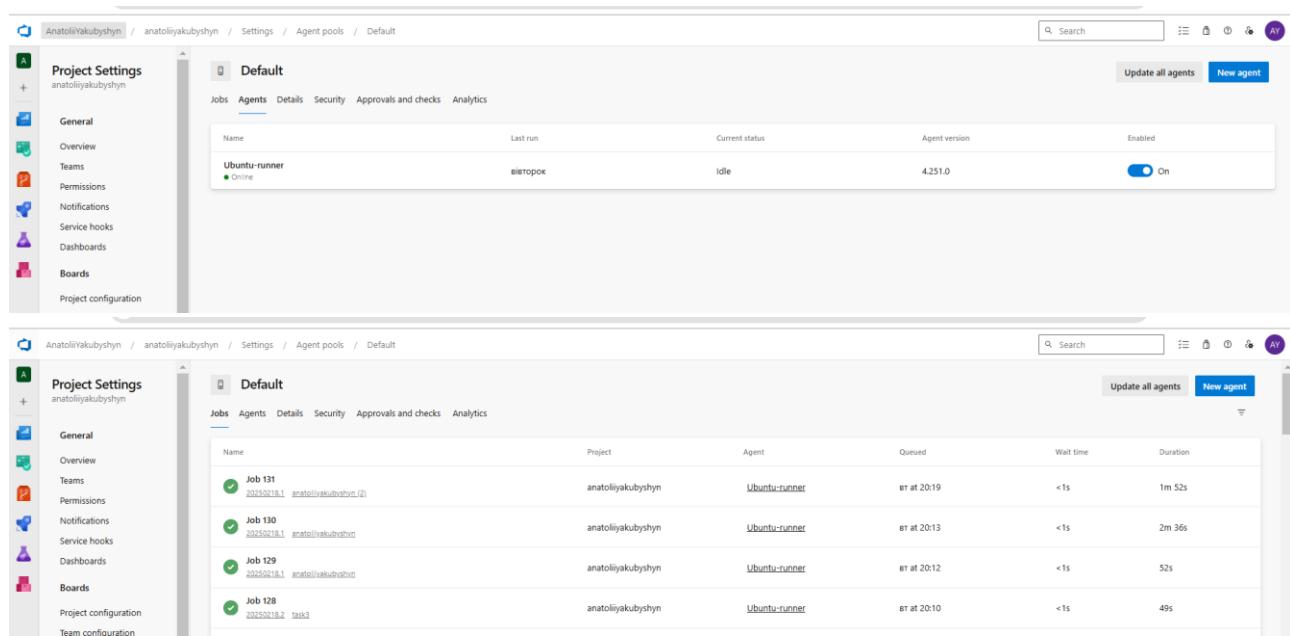
Anatolii Yakubshyn

Practical Task 1: Create and Initialize a New Repository with a README

PreRequirements:

We have created a Terraform pipeline that requires a remote state on a storage account—this might be useful for you. Please complete the following steps:

1. Create a virtual machine (Linux).
2. Set up a storage account for the state.
3. Install a self-hosted agent (Linux). pool: name: MyLinux demands: - agent.name -equals myAgent
4. Review the pull request I created with the pipeline.
5. Merge it and update the Agent name and variable group. 6. Ensure the pipeline runs successfully.



The image shows two screenshots of the Azure DevOps interface. The top screenshot displays the 'Agent pools' section for the 'Default' pool. It lists a single agent named 'Ubuntu-runner' which is online. The bottom screenshot shows the 'Jobs' section of the pipeline, listing four jobs: Job 131, Job 130, Job 129, and Job 128, all of which are queued and waiting for an agent.

Requirements:

Log in to your Azure DevOps account and navigate to your project.



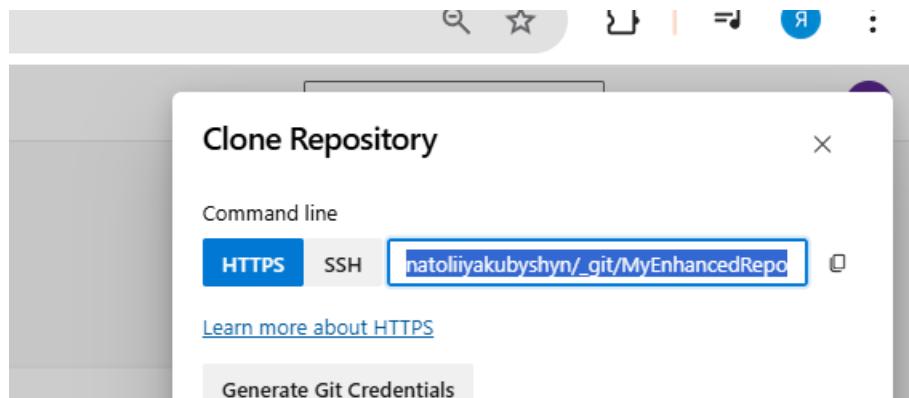
The image shows the 'Overview' page of a project named 'anatoliyakubshyn'. The left sidebar includes options like 'Summary', 'Dashboards', 'Wiki', 'Boards', 'Repos', and 'Pipelines'. The main area features sections for 'About this project' (with a placeholder for a project description) and 'Project stats' (showing 6 pull requests opened and 113 commits by 3 authors over the last 7 days). A central illustration of a person running on clouds is present.

Create a new Git repository named *MyEnhancedRepo*.

The screenshot shows the Azure DevOps interface for creating a new repository. In the top navigation bar, the URL is dev.azure.com/AnatoliYakubshyn/anatoliyakubshyn/_git/task3. The left sidebar shows the user's profile and navigation links for Overview, Boards, and Repos. The main area displays a list of existing repositories: task3, anatoliyakubshyn_task2, task2, and a New repository option. A modal window titled "Create a repository" is open, showing the configuration for the new repository. The "Repository type" dropdown is set to "Git". The "Repository name" field contains "MyEnhancedRepo". The "Add a README" checkbox is checked. The ".gitignore" dropdown is set to "None". A note at the bottom states, "Your repository will be initialized with a main branch." The background shows the previous repositories listed.

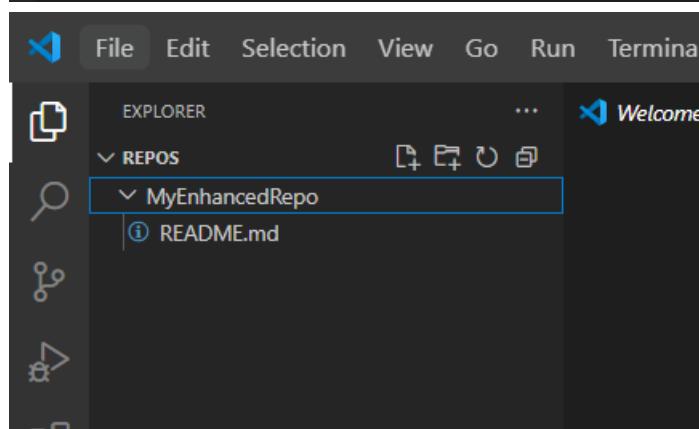
The screenshot shows the Azure DevOps interface after the repository has been created. The URL is dev.azure.com/AnatoliYakubshyn/anatoliyakubshyn/_git/MyEnhancedRepo. The left sidebar shows the user's profile and navigation links for Overview, Boards, and Repos. The main area displays the details of the newly created repository, "MyEnhancedRepo". It shows a single file, "README.md", with a commit history: "Added README.md Anatoli Yakubshyn". Below the file list, there are sections for "Introduction" and "Getting Started". The "Introduction" section has a placeholder: "TODO: Give a short introduction of your project. Let this section explain the objectives or the motivation behind this project." The "Getting Started" section has a placeholder: "TODO: Guide users through getting your code up and running on their own system. In this section you can talk about: 1. Installation process 2. Software dependencies". The background shows the repository structure and details.

Clone the repository to your local machine using Git.



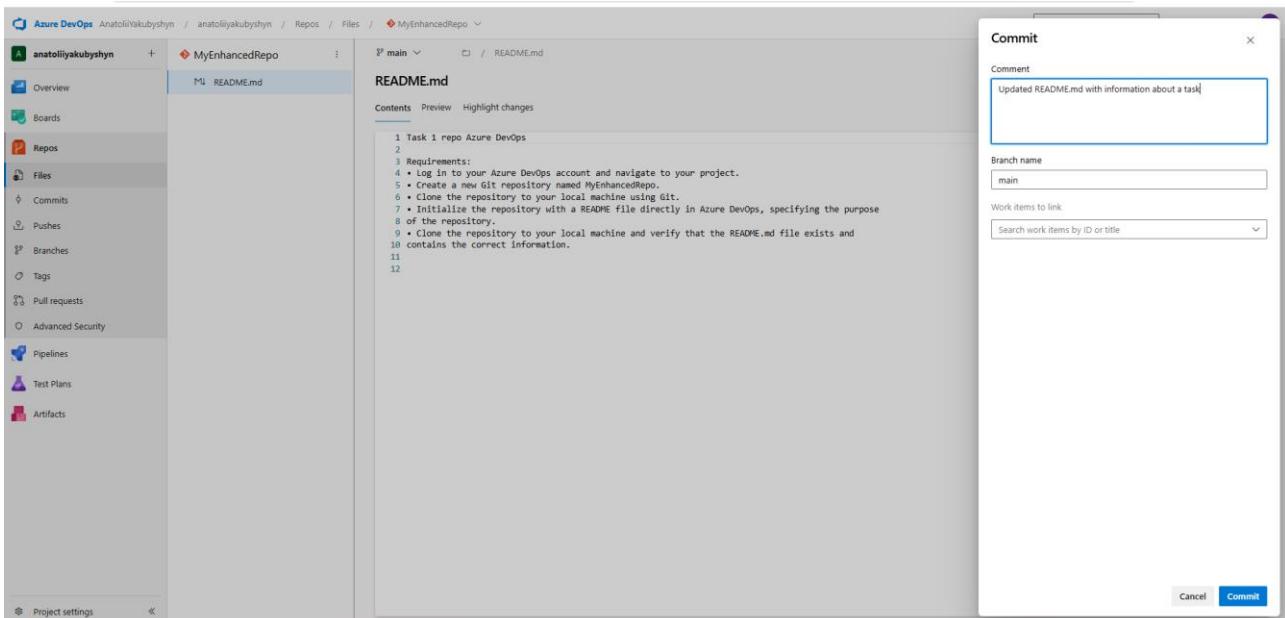
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

● PS D:\Azure\task7\repos> git clone https://AnatoliiYakubshyn@dev.azure.com/AnatoliiYakubshyn/anatoliyakubshyn/_git/MyEnhancedRepo
Cloning into 'MyEnhancedRepo'...
remote: Azure Repos
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
remote: Found 3 objects to send. (19 ms)
Unpacking objects: 100% (3/3), 763 bytes | 3.00 KiB/s, done.
○ PS D:\Azure\task7\repos>
```

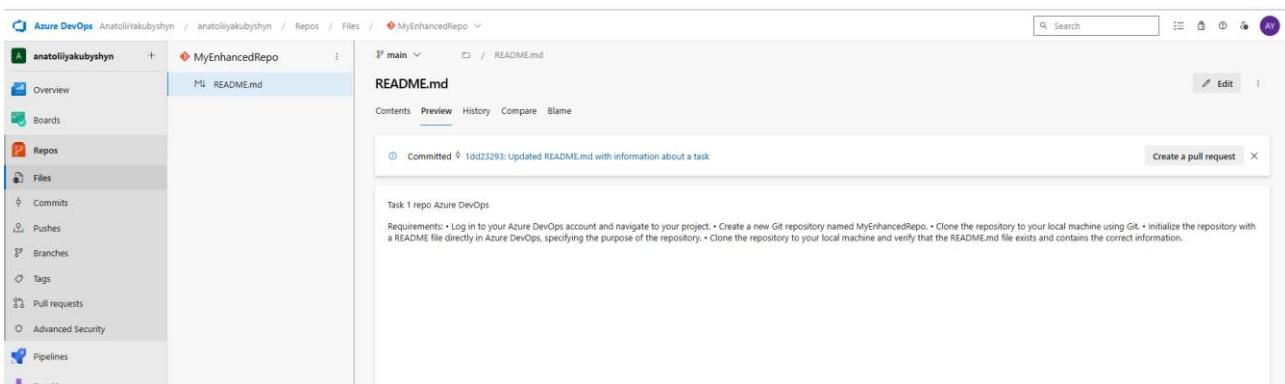


Initialize the repository with a README file directly in Azure DevOps, specifying the purpose of the repository.

```
1 Task 1 repo Azure DevOps
2 Requirements:
3 • Log in to your Azure DevOps account and navigate to your project.
4 • Create a new Git repository named MyEnhancedRepo.
5 • Clone the repository to your local machine using Git.
6 • Initialize the repository with a README file directly in Azure DevOps, specifying the purpose
7 of the repository.
8 • Clone the repository to your local machine and verify that the README.md file exists and
9 contains the correct information.
10
11
12
```

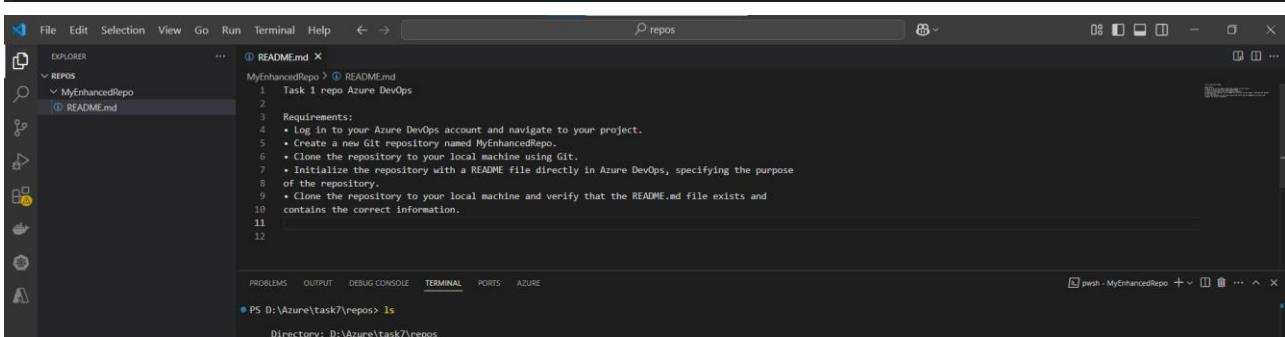


click commit



Clone the repository to your local machine and verify that the README.md file exists and contains the correct information.

```
● PS D:\Azure\task7\repos> cd MyEnhancedRepo
● PS D:\Azure\task7\repos\MyEnhancedRepo> git pull origin main
remote: Azure Repos
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
remote: Found 3 objects to send. (18 ms)
Unpacking objects: 100% (3/3), 540 bytes | 3.00 KiB/s, done.
From https://dev.azure.com/AnatoliiYakubshyn/anatoliiyakubshyn/_git/MyEnhancedRepo
 * branch      main      -> FETCH_HEAD
   b85b09e..1dd2329  main      -> origin/main
Updating b85b09e..1dd2329
Fast-forward
 README.md | 27 ++++++-----
 1 file changed, 9 insertions(+), 18 deletions(-)
● PS D:\Azure\task7\repos\MyEnhancedRepo>
```

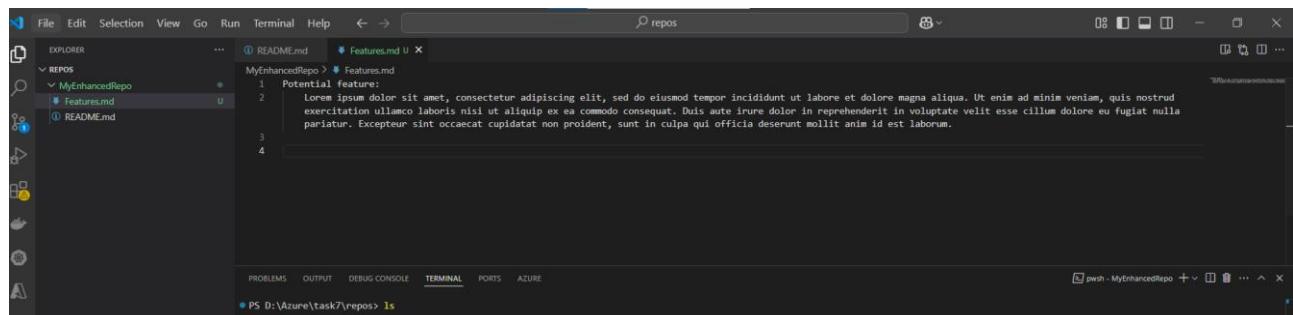


Practical Task 2: Implement Branching and Merging

- Create a new branch named feature-enhancement from the main branch in your local repository

```
1 file changed, 9 insertions(+), 18 deletions(-)
● PS D:\Azure\task7\repos\MyEnhancedRepo> git checkout -b feature-enhancement
Switched to a new branch 'feature-enhancement'
○ PS D:\Azure\task7\repos\MyEnhancedRepo>
```

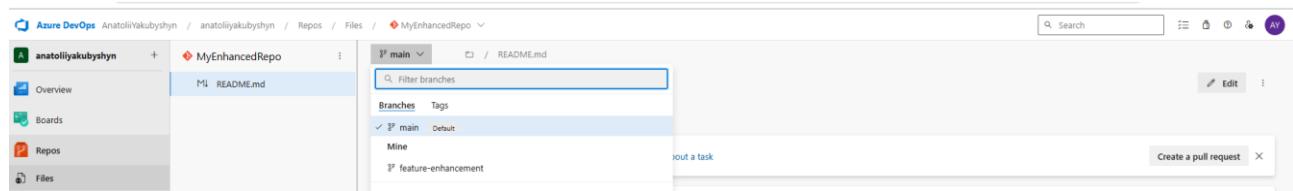
Add a new file named FEATURES.md to the feature-enhancement branch, detailing potential features for the project.



Commit the changes and push the feature-enhancement branch to the remote repository in Azure Repos

```
1 file changed, 9 insertions(+), 18 deletions(-)
● PS D:\Azure\task7\repos\MyEnhancedRepo> git checkout -b feature-enhancement
Switched to a new branch 'feature-enhancement'
● PS D:\Azure\task7\repos\MyEnhancedRepo> git add -A
● PS D:\Azure\task7\repos\MyEnhancedRepo> git status
On branch feature-enhancement
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  Features.md

● PS D:\Azure\task7\repos\MyEnhancedRepo> git commit -m "created Features.md and filled with dummy text"
[feature-enhancement f38f6e0] created Features.md and filled with dummy text
 1 file changed, 3 insertions(+)
   create mode 100644 Features.md
● PS D:\Azure\task7\repos\MyEnhancedRepo> git branch
 * feature-enhancement
   main
● PS D:\Azure\task7\repos\MyEnhancedRepo> git push origin feature-enhancement
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 579 bytes | 579.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (3/3) (4 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (43 ms)
remote: Storing index... done (23 ms)
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
To https://dev.azure.com/AnatoliyYakubshyn/anatoliyyakubshyn/_git/MyEnhancedRepo
 * [new branch]      feature-enhancement -> feature-enhancement
○ PS D:\Azure\task7\repos\MyEnhancedRepo>
```



- Create a pull request from feature-enhancement to main, including a description of the changes and why they are necessary.

Azure DevOps Anatoliyakubshyn / anatoliyakubshyn

anatoliyakubshyn

MyEnhancedRep

README.md

Overview Boards Repos Files Commits Pushes Branches Tags Pull requests

Click Pull requests

Anatoliyakubshyn / anatoliyakubshyn / Repos / Pull requests / MyEnhancedRepo

New pull request

feature-enhancement into main

Title: created Features.md and filled with dummy text

Description: These changes needed to implement the task's 2 subtask: "Create a pull request from feature-enhancement to main, including a description of the changes and why they are necessary."

183/4000

Markdown supported. Drag & drop, paste, or select files to insert. Link work items.

Reviewers: Add required reviewers

Work items to link: Search work items by ID or title

Tags:

Create

Click create

Anatoliyakubshyn / anatoliyakubshyn / Repos / Pull requests / MyEnhancedRepo

Anatoliyakubshyn proposes to merge feature-enhancement into main

No merge conflicts Last checked just now

Description: These changes needed to implement the task's 3 subtask: "Create a pull request from feature-enhancement to main, including a description of the changes and why they are necessary."

Reviewers: Required: No required reviewers, Optional: No optional reviewers

Approve Complete

Review the pull request and merge it into the main branch after resolving any conflicts that may arise.

Show everything (3) 

Anatolii Yakubshyn approved the pull request Just now

Anatolii Yakubshyn joined as a reviewer Just now

Add a comment... 

Work item 

🔍 ⭐ 📁 🔍 ⏮

Complete pull request 

Merge type  Merge (no fast forward)

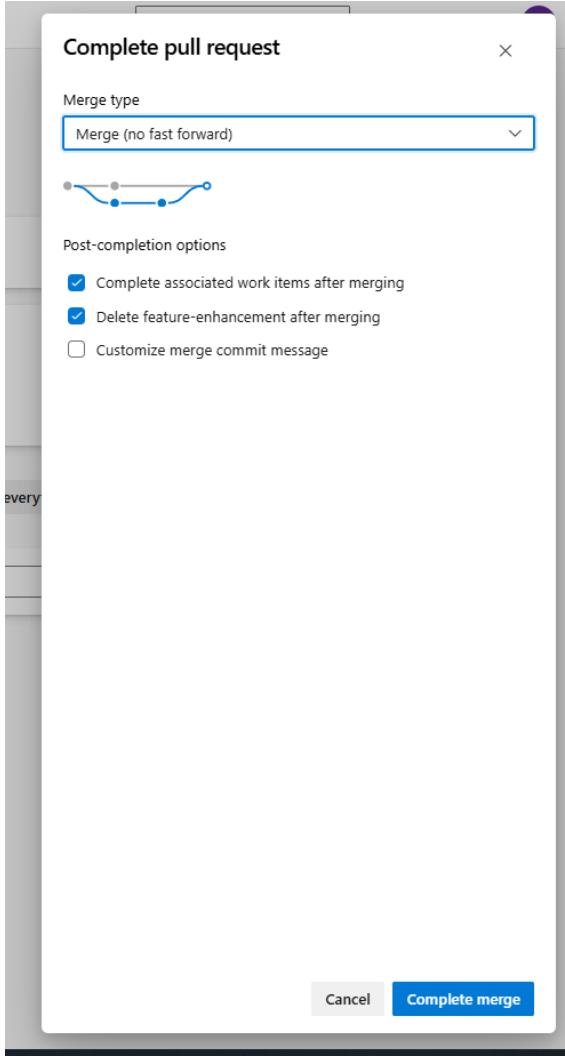


Post-completion options

Complete associated work items after merging

Delete feature-enhancement after merging

Customize merge commit message



The screenshot shows the Azure DevOps repository page for 'MyEnhancedRepo'. A pull request titled 'Anatoli Yakubshyn proposes to merge feature-enhancement into main' is shown as 'Completed'. The status bar indicates 'Anatoli Yakubshyn completed this pull request just now'. The 'Reviewers' section shows 'Required' with one user listed. The sidebar on the left shows 'Overview', 'Boards', 'Repos', and 'Files' sections.

The screenshot shows the Azure DevOps file editor for 'Features.md' in the 'main' branch. The file content is as follows:

```
1 Potential feature:  
2 | Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercit  
3 |  
4 |
```

Practical Task 3: Set Up Branch Policies and Code Review

Configure branch policies for the main branch in Azure Repos to require at least one reviewer for pull requests and to enforce a minimum number of successful builds (you can simulate this by using a mock build status if needed).

The screenshot shows the Azure Repos settings interface for a project named 'anatoliyakubshyn'. The left sidebar is open, showing various settings like General, Boards, Pipelines, and Repos. The 'Repositories' section is selected. The main area displays 'Repository Policies' and 'Branch Policies'.

Repository Policies:

- Commit author email validation:** Off (Blocks pushes with a commit author email that does not match the following patterns.)
- File path validation:** Off (Blocks pushes from introducing file paths that match the following patterns.)
- Case enforcement:** Off (Blocks pushes that change name casing on files, folders, branches, and tags.)
- Reserved names:** Off (Blocks pushes that introduce file, folder, or branch names that include punctuation, reserved names or incompatible characters.)
- Maximum path length:** Off (Blocks pushes that introduce paths or exceed the specified length.)
- Maximum file size:** Off (Blocks pushes that contain new or updated files larger than this limit.)
- Check for credentials and other secrets:** Off (Blocks pushes that contain credentials or other secrets.)

Branch Policies:

Protect important branch namespaces across all repositories in this project.

Add branch protection dialog:

Branches to protect:

Protect the default branch of each repository
 Protect current and future branches matching a specified pattern

main

Example patterns: "main", "master", "releases/*", "*"

Matches 3 total branches in 3 repos

main
anatoliyakubshyn_task2

main
MyEnhancedRepo

main
task3

Create

🔍 ☆ ✖

Add build policy

Build pipeline *

MyEnhancedRepo (8)

Path filter (optional)

Trigger

Automatic (whenever the source branch is updated)

Manual

Policy requirement

Required

Build must succeed in order to complete pull requests.

Optional

Build failure will not block completion of pull requests.

Build expiration

Immediately when \varnothing branches named "refs/heads/main" is updated

After hours if \varnothing branches named "refs/heads/main" has been updated

Never

Display name

Build

Save Cancel



- Create a new branch named bugfix-issue from the main branch and make changes to the FEATURES.md file to address a hypothetical bug (e.g., update a feature description). • Commit the changes and push the bugfix-issue branch to the remote repository.

```
● PS D:\Azure\task7\repos\MyEnhancedRepo> git checkout -b bugfix-issue
Switched to a new branch 'bugfix-issue'
PS D:\Azure\task7\repos\MyEnhancedRepo>
Branch Policies: repos
View Go Run Terminal Help ← → 🔍 repos 🌐
... README.md Features.md
MyEnhancedRepo > Features.md
1 Potential feature:
2 Updated description
3 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
4
5

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE
git - MyEnhancedRepo + ⚡ 🌐 ... ▾
● PS D:\Azure\task7\repos\MyEnhancedRepo> git checkout -b bugfix-issue
Switched to a new branch 'bugfix-issue'
● PS D:\Azure\task7\repos\MyEnhancedRepo> git status
On branch bugfix-issue
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   Features.md

no changes added to commit (use "git add" and/or "git commit -a")
● PS D:\Azure\task7\repos\MyEnhancedRepo> git add .
● PS D:\Azure\task7\repos\MyEnhancedRepo> git commit -m "updated feature description"
[bugfix-issue b26d7da] updated feature description
 1 file changed, 1 insertion(+)
● PS D:\Azure\task7\repos\MyEnhancedRepo> git push bugfix-issue
fatal: 'bugfix-issue' does not appear to be a valid repository
fatal: Could not read from remote repository.

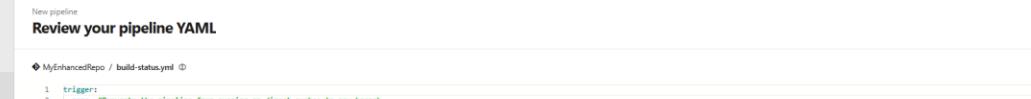
Please make sure you have the correct access rights
and the repository exists.
● PS D:\Azure\task7\repos\MyEnhancedRepo> git push origin bugfix-issue
Enumerating objects: 5, done.
Counting objects: 100%, (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 329.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (3/3) (4 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... (2/2) (22 ms)
remote: Writing objects... (3/3) (22 ms)
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
To https://dev.azure.com/AnatoliyAkubsyn/AnatoliyAkubsyn/_git/MyEnhancedRepo
 * [new branch]  bugfix-issue -> bugfix-issue

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE
git - build-status.yml + build-statusyml 🌐
... README.md Features.md ! build-statusyml
! build-statusyml > [ ] stages > ( ) o > [ ] steps
1 trigger:
2 | none #Prevents the pipeline from running on direct pushes to any branch
3
4 pr:
5 branches:
6 | include:
7 | | - main
8
9 pool:
10 name: default
11 demands:
12 | - agent.name -equals Ubuntu-runner
13
14 stages:
15 - stage: Build
16 | job: myName: Build
17 | steps:
18 | - script: curl https://www.google.com/
19 | | display-name: mock-build
20
21
22
```

created mock PR build

Create a pull request from bugfix-issue to main, ensuring that the pull request meets the branch policies set earlier

The screenshot shows the 'New pull request' creation interface in Azure DevOps. The left sidebar is visible with 'anatoliyakubshyn' selected under 'Anatoliyakubshyn'. The main area shows a 'bugfix-issue' branch being merged into the 'main' branch. The 'Overview' tab is selected. The 'Title' field contains 'Bug fix'. The 'Description' field has the placeholder 'Describe the code that is being reviewed'. Below the description is a rich text editor toolbar. The 'Reviewers' section includes a search bar for users and groups. The 'Work items to link' section has a search bar for work items by ID or title. The 'Tags' section is empty. At the bottom right is a blue 'Create' button.



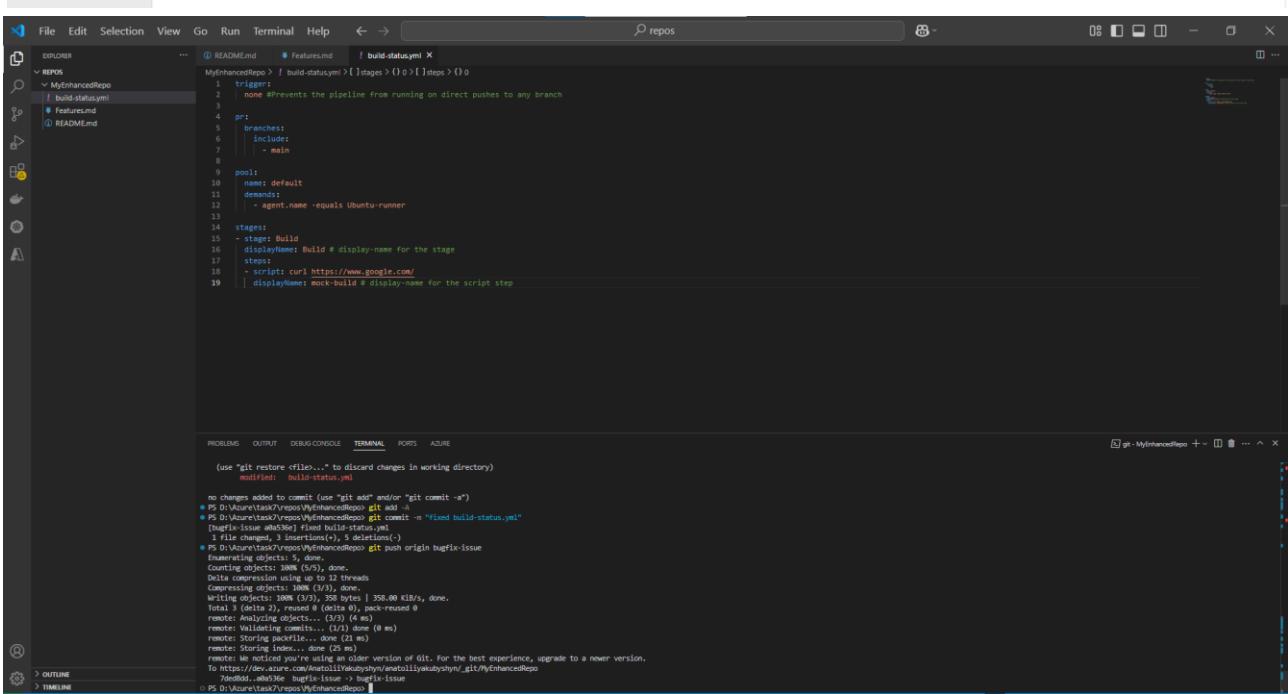
The screenshot shows the Azure DevOps Pipeline Review screen. The pipeline name is "MyEnhancedRepo / build-status.yml". The review tab is selected. The pipeline configuration is as follows:

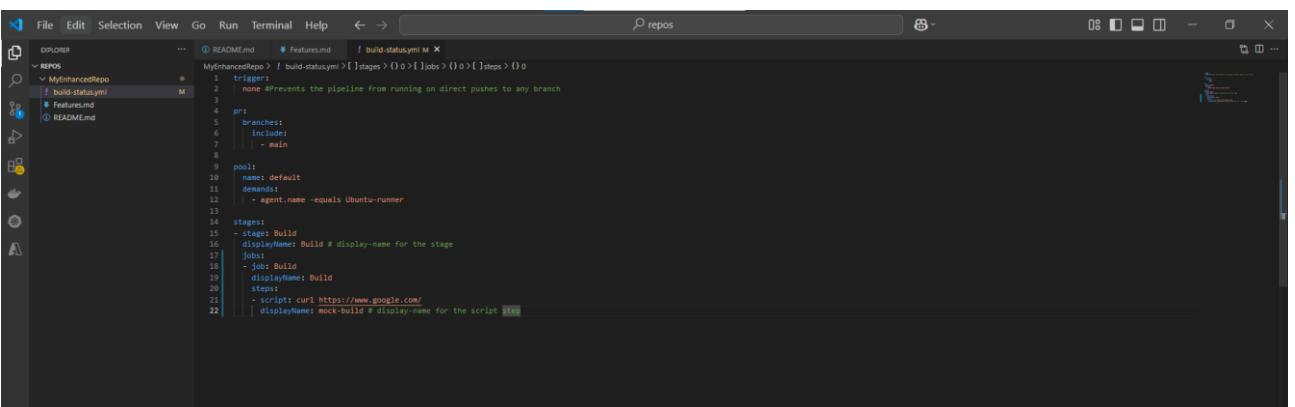
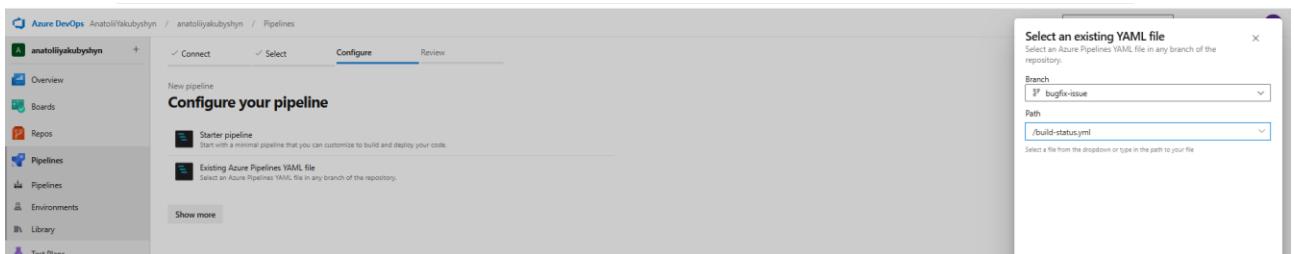
```
trigger:
  - none #Prevents the pipeline from running on direct pushes to any branch

pr:
  branches:
    include:
      - main

pool:
  default:
  demands:
    - agent.name -equals Ubuntu-runner

stages:
  stage: Build
  - displayName: Build
  steps:
    - script: curl https://www.google.com/
      displayName: Mock-Build
```





Azure DevOps Pipelines interface showing a build log for job #20250221.4. The log details the execution of various steps: Initialize job, Checkout MyEnhancedRepo@refs..., mock-build, Post-job: Checkout MyEnhanced..., Finalize Job, and Report build status. The build was started just now and completed successfully.

Azure DevOps Repos interface showing a pull request titled "Bug fix" from "AnatoliYakubshyn" to "MyEnhancedRepo". The pull request has 8 actions and 1 review. It includes sections for Overview, Files, Updates, Commits, and Description. The pull request is currently in the "Build Validation" stage, which is enabled and set to "Build Required". The status checks section shows "Required check succeeded" and "Build succeeded". The reviewers section lists "AnatoliYakubshyn" as approved. The pull request is awaiting review by at least one other person.

- Invite at least one team member or colleague to review the pull request and provide feedback. •

Address any comments or requested changes from the reviewer and update the pull request accordingly. •

Merge the pull request into the main branch once it has been approved.

Azure DevOps Repos interface showing the same pull request for "bugfix-issue" into "main". The pull request now has 8 reviews and 1 approval. The reviewers section shows "Dmytro Slotvinskyi" as having "No review yet" and "AnatoliYakubshyn" as approved. The pull request is now ready for merging.

Invite at least one team member or colleague to review the pull request and provide feedback.

- Address any comments or requested changes from the reviewer and update the pull request accordingly.
- Merge the pull request into the main branch once it has been approved.

The screenshot shows a pull request in the Azure DevOps interface. The pull request has been completed by Dmytro Stovinskiy and approved by Anatoli Yakubshyn. Anatoli pushed a commit. A build status check failed, and the log shows a script error. Dmytro responded asking for logic explanation, and Anatoli provided a more complex script. Dmytro approved the pull request. The URL in the browser bar is dev.azure.com/AnatoliYakubshyn/anatoliyakubshyn/_git/MyEnhancedRepo/pullrequest/8.

Practical Task 4: Create a Simple Build Pipeline in Azure DevOps

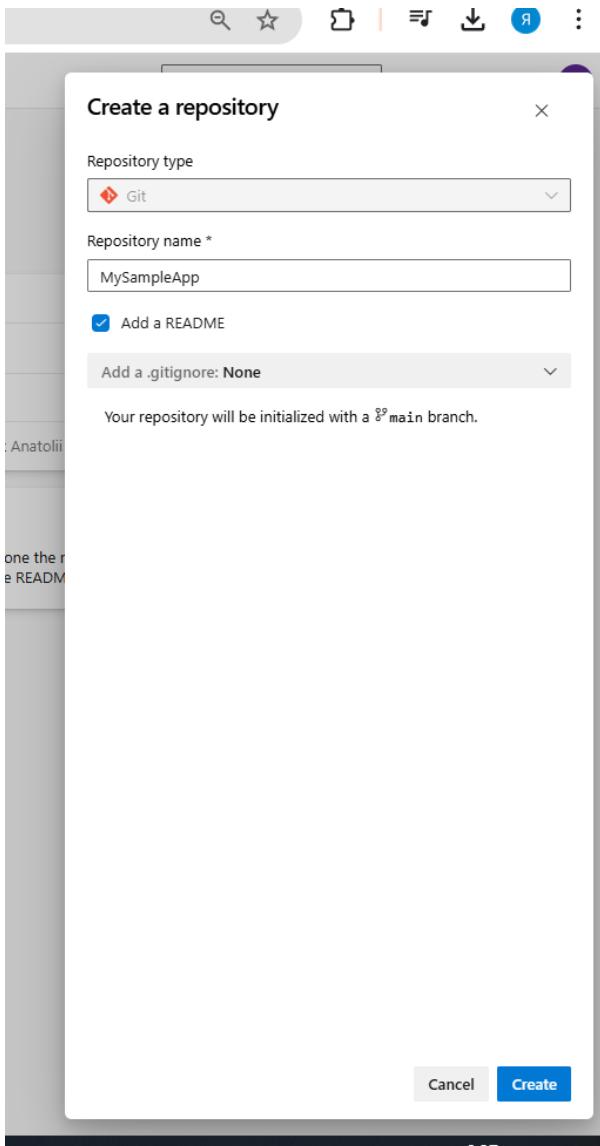
1. Log in to Azure DevOps: a. Access your Azure DevOps organization and navigate to your project.

The screenshot shows the project overview page for 'anatoliyakubshyn'. It includes sections for 'About this project' (with a 'Add Project Description' button), 'Project stats' (showing 7 pull requests opened, 125 commits by 3 authors, and 25% builds succeeded), and a summary of recent activity.

2. Create a Git Repository:

a. In Azure Repos, create a new Git repository named MySampleApp.

The screenshot shows the 'Create New Repository' interface. A new repository named 'MySampleApp' is being created under the 'anatoliyakubshyn' organization. The repository structure includes a 'MyEnhancedRepo' folder containing 'build-status.yml', 'Features.md', and 'README.md' files.



Clone the repository to your local machine and add a simple application (e.g., a basic .NET Core or Node.js application).

```
PS D:\Azure\task7\repos> git clone https://AnatoliiYakubyshyn@dev.azure.com/AnatoliiYakubyshyn/anatoliiyakubyshyn/_git/MySampleApp
Cloning into 'MySampleApp'...
remote: Azure Repos
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
remote: Found 3 objects to send. (24 ms)
Unpacking objects: 100% (3/3), 763 bytes | 3.00 KiB/s, done.
PS D:\Azure\task7\repos>
```

Node.js app (React) – Calculator

MySampleApp	•
public	•
src	•
JS App.js	U
# App.module.css	U
JS App.test.js	U
# index.css	U
JS index.js	U
JS reportWebVitals.js	U
JS setupTests.js	U
diamond .gitignore	U
{} package-lock.json	U
{} package.json	U

The screenshot shows the Azure DevOps interface for a repository named 'MySampleApp'. The left sidebar lists the repository's structure: 'public', 'src', '.gitignore', 'package.json', and 'README.md'. The main area displays the contents of these files. A 'Getting Started with Create React App' section provides instructions for running the application.

3. Create a Build Pipeline:

a. Navigate to Azure Pipelines and create a new pipeline.

The screenshot shows the Azure DevOps Pipelines page. The 'Pipelines' tab is selected. It displays a list of recently run pipelines, with a 'Pipeline' and 'Last run' column.

Click new pipeline

The screenshot shows the 'New pipeline' creation page. The 'Select' tab is active. It asks 'Where is your code?' and lists options for connecting to code repositories: 'Azure Repos Git' (YAML), 'Bitbucket Cloud' (YAML), 'GitHub' (YAML), and 'GitHub Enterprise Server' (YAML).

Azure repos git

b. Select the Azure Repos Git repository as the source.

The screenshot shows the 'Select a repository' step in the pipeline creation process. The 'Select' tab is active. It shows a list of repositories: 'anatoliyakubshyn_task2', 'MyEnhancedRepo', 'MySampleApp', and 'task3'. The repository 'MySampleApp' is highlighted.

MySampleApp

c. Use the YAML pipeline editor to define the pipeline, including the following steps:

The screenshot shows the Azure DevOps interface for defining a new pipeline. The top section, "Configure your pipeline", shows a Node.js icon with the text "Build a general Node.js project with npm.". The bottom section, "Review your pipeline YAML", displays the following YAML code:

```
1 # Node.js
2 # Build a general Node.js project with npm.
3 # Add steps that analyze code, save build artifacts, deploy, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/javascript
5
6 trigger:
7 - main
8
9 pool:
10 | vmImage: ubuntu-latest
11
12 steps:
13 - task: NodeTool@0
14   inputs:
15     versionSpec: '16.x'
16     displayName: 'Install Node.js'
17
18 - script: npm install
19   displayName: 'npm install'
20 - script: npm test
21   displayName: 'npm test'
22 - script: npm build
23   displayName: 'npm build'
```

Below this, the "Edit" button is visible, indicating the code can be modified directly.

(fixed build script)

```
},
  "scripts": [
    "start": "react-scripts start",
    "predeploy": "npm run build",
    "deploy": "gh-pages -d build",
    "build": "react-scripts build",
    "test": "react-scripts test --watchAll=false",
    "eject": "react-scripts eject"
  ],
  "eslintConfig": {
```

changed test

```

166   > 
167   </button>
168   </div>
169   </main>
170   </div>
171   </div>
172
173   4 | test('renders learn react link', () => {
174     5 |   render();
175     > 6 |   const linkElement = screen.getByText(/learn react/i);
176     |
177     7 |   expect(linkElement).toBeInTheDocument();
178     8 | });
179
180
181   at Object.getComputedStyle (node_modules/@testing-library/react/node_modules/@testing-library/react-dom/dist/config.js:37:19)
182   at node_modules/@testing-library/react/node_modules/@testing-library/dom/dist/query-helpers.js:76:38
183   at node_modules/@testing-library/react/node_modules/@testing-library/react-dom/dist/query-helpers.js:51:17
184   at getByText (node_modules/@testing-library/react/node_modules/@testing-library/dom/dist/query-helpers.js:95:19)
185   at Object.<anonymous> (src/App.test.js:6:30)
186   at TestScheduler.scheduleTests (node_modules/@jest/core/build/testscheduler.js:333:13)
187   at runTest (node_modules/@jest/core/build/runtest.js:464:19)
188   at _runTest (node_modules/@jest/core/build/cli/index.js:51:26:7)
189   at runCLI (node_modules/@jest/core/build/cli/index.js:173:3)
190
191 Test Suites: 1 failed, 1 total
192 Tests:       1 failed, 1 total
193 Snapshots:  0 total
194 Time:        2.084 s
195 Ran all test suites.
196
197 [error] bash exited with code '1'.
198 Finishing: npm test

```

Fix test->

```

1 import { render, screen } from '@testing-library/react';
2 import App from './App';
3
4 test('renders learn react link', () => {
5   render(<App />);
6   const linkElement = screen.getByText(/learn react/i);
7   expect(linkElement).toBeInTheDocument();
8 });

```

4. Run the Pipeline: a. Save and run the pipeline. b. Verify that the build pipeline completes successfully and check the build logs for any errors.

```

← Jobs in run #20250222.7
MySampleApp

Jobs
Job      54s
Initialize job 3s
Checkout MySampleApp@main t... 45s
Install Node.js 2s
npm install 21s
npm test 6s
npm run build 11s
Post-job: Checkout MySampleApp... 45s

npm test
51   6 |   const linkElement = screen.getByText(/>/);
52   7 |   expect(linkElement).toBeInTheDocument();
53   8 | });
54
55   at printWarning (node_modules/react-dom/cjs/react-dom.development.js:66:30)
56   at error (node_modules/react-dom/cjs/react-dom.development.js:68:7)
57   at validateProperty1 (node_modules/react-dom/cjs/react-dom.development.js:1740:19)
58   at validateProperty2 (node_modules/react-dom/cjs/react-dom.development.js:1703:21)
59   at validateProperties1 (node_modules/react-dom/cjs/react-dom.development.js:1827:3)
60   at validateProperties2 (node_modules/react-dom/cjs/react-dom.development.js:1841:5)
61   at setInitialProperties (node_modules/react-dom/cjs/react-dom.development.js:980:5)
62   at finalizeInitialChildren (node_modules/react-dom/cjs/react-dom.development.js:1895:3)
63   at completeRoot (node_modules/react-dom/cjs/react-dom.development.js:2222:17)
64   at completeUnitOfWork (node_modules/react-dom/cjs/react-dom.development.js:2663:16)
65   at performUnitOfWork (node_modules/react-dom/cjs/react-dom.development.js:2669:5)
66   at workLoopSync (node_modules/react-dom/cjs/react-dom.development.js:2649:7)
67   at flushSyncWorkQueue (node_modules/react-dom/cjs/react-dom.development.js:2649:7)
68   at performSyncWorkQueue (node_modules/react-dom/cjs/react-dom.development.js:2877:74)
69   at flushActQueue (node_modules/react/cjs/react.development.js:2667:24)
70   at act (node_modules/react/cjs/react.development.js:2521:1)
71   at actWithTransition (node_modules/react-dom/cjs/react-dom-test-util.development.js:1740:18)
72   at node_modules/@testing-library/react/dist/internal/comput.js:1:2
73   at renderRoot (node_modules/@testing-library/react/dist/pure.js:159:28)
74   at render (node_modules/@testing-library/react/dist/pure.js:124:18)
75   at Object.<anonymous> (src/App.test.js:5:19)
76
77   at TestScheduler.scheduleTests (node_modules/@jest/core/build/testscheduler.js:333:13)
78   at runTest (node_modules/@jest/core/build/runtest.js:464:19)
79   at _runTest (node_modules/@jest/core/build/cli/index.js:51:26:7)
80   at runCLI (node_modules/@jest/core/build/cli/index.js:173:3)
81
82 PASS src/App.test.js
83   ✓ renders learn react link (97 ms)
84
85 Test Suites: 1 passed, 1 total
86 Tests:       1 passed, 1 total
87 Snapshots:  0 total
88 Time:        3.5 s
89 Ran all test suites.
90
91 Finishing: npm test

```

Pipeline	Last run	Time
MySampleApp	#20250222.7 • fixed test	4m ago 1m 5s

Practical Task 5: Set Up Continuous Deployment (CD) to Azure Web App

Presetup:

The screenshot shows the 'Organization Settings' page in Azure DevOps. On the left, there's a sidebar with sections like General, Security, Boards, and Pipelines. The Pipelines section has a switch labeled 'Disable creation of classic release pipelines' which is currently set to 'Off'. This switch is circled in red.

Setting	Status	Description
Disable anonymous access to badges	On	Anonymous users can access the status badge API for all pipelines unless this option is enabled (does not apply to public projects).
Limit variables that can be set at queue time	On	You can set any variables at queue time unless this option is enabled. With this option enabled, only those variables that are explicitly marked as "Settable at queue time" can be set. Learn more
Limit job authorization scope to current project for non-release pipelines	On	Non-Release Pipelines can run with collection scoped access tokens unless this option is enabled. With this option enabled, you can reduce the scope of access for all non-release pipelines to the current project.
Limit job authorization scope to current project for release pipelines	On	Release pipelines can run with collection scoped access tokens unless this option is enabled. With this option enabled, you can reduce the scope of access for all release pipelines to the current project.
Protect access to repositories in YAML pipelines	On	Apply checks and approvals when accessing repositories from YAML pipelines. Also, generate a job access token that is scoped to repositories that are explicitly referenced in the YAML pipeline.
Disable stage chooser	Off	With this enabled, users will not be able to select stages to skip from the Queue Pipeline panel.
Disable creation of classic build pipelines	On	No classic build pipelines can be created / imported. Existing ones will continue to work.
Disable creation of classic release pipelines	Off (circled)	No classic release pipelines, task groups, and deployment groups can be created / imported. Existing ones will continue to work.

I decided to create a separate repo with java jar. Because my first app produces static files but I need package.

2. Create an Azure Web App: a. In the Azure portal, manually create an Azure Web App or use an ARM template to create it.

Create Web App ...

[Basics](#) [Database](#) [Deployment](#) [Networking](#) [Monitor + secure](#) [Tags](#) [Review + create](#)

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. [Learn more](#)

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource Group *

Instance Details

Name

Secure unique default hostname on. [More about this update](#)

Publish * Code Container

Runtime stack *

Java web server stack *

Operating System * Linux Windows

Region *
(i) Not finding your App Service Plan? Try a different region or select your App Service Environment.

Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

Linux Plan (Poland Central) *

Pricing plan

Zone redundancy

An App Service plan can be deployed as a zone redundant service in the regions that support it. This is a deployment time only decision. You can't make an App Service plan zone redundant after it has been deployed. [Learn more](#)

Zone redundancy Enabled: Your App Service plan and the apps in it will be zone redundant. The minimum App Service plan instance count will be three. Disabled: Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.

[Review + create](#)[< Previous](#)[Next : Database >](#)

☰ Microsoft Azure Search resources, se...[Home](#) > [AnatoliYakubshyn](#) > [Marketplace](#) > [Web App](#) >

Create Web App

[Basics](#) [Database](#) [Deployment](#) [Networking](#) [Monitor + secure](#) [Tags](#) [Review + create](#)

Summary



Free sku

Estimated price - Free

Basic authentication for this app is currently disabled and may impact deployments. Click to learn more.

Details

Subscription	9a6ae428-d8c3-44fe-bdf2-4e08593901a0
Resource Group	AnatoliYakubshyn
Name	Yakubshyn
Secure unique default hostname	Enabled
Publish	Code
Runtime stack	Java 11
Java web server stack	Java SE (Embedded Web Server)

App Service Plan (New)

Name	Yakubshyn
Operating System	Linux
Region	Poland Central
SKU	Free
ACU	Shared infrastructure
Memory	1 GB memory

Monitor + secure

Application Insights	Not enabled
----------------------	-------------

Deployment

Basic authentication	Disabled
Continuous deployment	Not enabled / Set up after app creation

[Create](#)[< Previous](#)[Next >](#)[Download a template for automation](#)

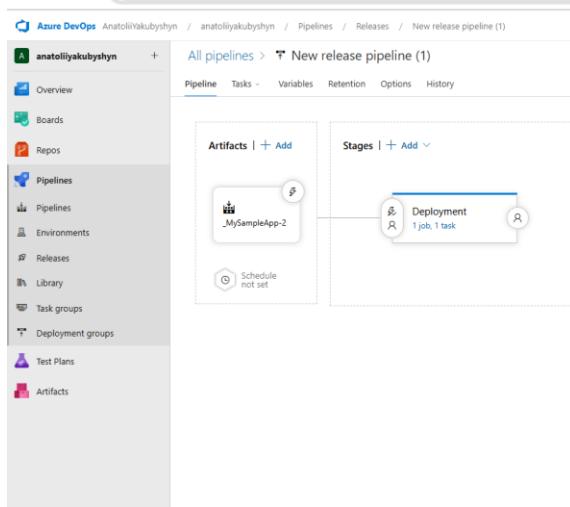
1. Extend the Build Pipeline: a. Open the existing build pipeline created in Task 4. b. Add a new stage for deployment after the build stage.

Contents History Compare Blame

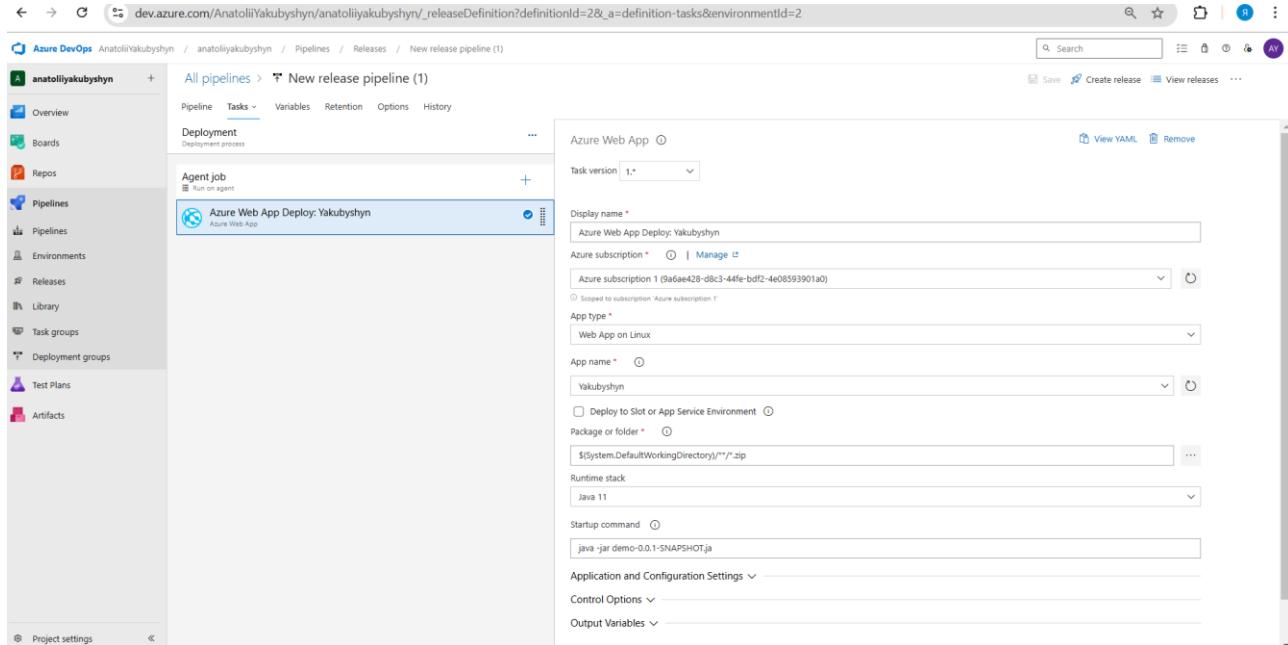
```
1 trigger:
2 - main
3
4 pool:
5   name: default
6   demands:
7     - agent.name -equals Ubuntu-runner
8
9 stages:
10 - stage: Build
11   displayName: Build and Test
12   jobs:
13     - job: Package
14       steps:
15         - script:
16           - sudo apt update
17           - sudo apt install -y openjdk-11-jdk
18           - java -version
19           - displayName: 'Install Java 11'
20
21         - script:
22           - sudo apt install -y maven
23           - mvn -v
24           - displayName: 'Install Maven'
25
26         - script: mvn package
27           - displayName: 'Run mvn package'
28
29         - script: sudo apt-get -y install zip
30           - displayName: 'Install zip'
31
32       - task: ArchiveFiles@2
33         inputs:
34           - rootFolderOrFile: 'target/demo-0.0.1-SNAPSHOT.jar'
35           - archiveType: 'zip'
36           - archiveFile: '${Build.ArtifactStagingDirectory}/build.zip'
37
38     - task: PublishBuildArtifacts@0
39       displayName: "Publish Build Artifacts"
40       inputs:
41         - pathToPublish: '${Build.ArtifactStagingDirectory}/build.zip'
42         - artifactName: 'build'
43
44   - stage: Deploy
45     dependsOn: Build
46     displayName: Deploy
47     condition: succeeded()
48     jobs:
49       - job: Deploy
50         steps:
51           - task: DownloadBuildArtifacts@0
52             displayName: "Download Build Artifacts"
53             inputs:
54               - buildType: "current"
55               - artifactName: "build"
56               - downloadPath: "${System.ArtifactsDirectory}"
57
58       - task: AzureRMWebAppDeployment@5
59         inputs:
60           - connectionType: 'AzureRM'
61           - azureSubscription: 'Azure subscription 1 (9a8ae428-d8c3-44fe-bdf2-4e08593981a8)'
62           - appType: 'webAppLinux'
63           - webAppName: 'Yakubbyshyn'
64           - packageForLinux: '${Pipeline.workspace}/**/build/build.zip'
65           - runtimeStack: 'JAVA:11-javaw'
66           - startupCommand: 'java -jar demo-0.0.1-SNAPSHOT.jar --server.port=80'
67           - deploymentTypeLinux: 'oneDeploy'
68
```

3. Configure Release Pipeline:

- Create a release pipeline in Azure DevOps that triggers automatically after a successful build.
- Add an Azure Web App deployment task to the release pipeline, specifying the Azure subscription and the Web App name.



The screenshot shows the Azure DevOps Pipelines interface. On the left, the sidebar is visible with 'Pipelines' selected. In the center, a 'New release pipeline (1)' is displayed. The pipeline has one stage named 'Deployment' which contains one task named 'Deployment'. An artifact named '_MySampleApp-2' is connected to this stage. The stage is currently set to 'not set'.



The screenshot shows the configuration for the 'Deployment' task under the 'Azure Web App' section. The task version is set to 1.*. The display name is 'Azure Web App Deploy: Yakubshyn'. The Azure subscription is set to 'Azure subscription 1 (9a0aae428-d8c3-44fe-bdf2-4e08593901a0)'. The app type is 'Web App on Linux'. Other settings include the app name 'Yakubshyn', package or folder '\$(System.DefaultWorkingDirectory)/**/.zip', runtime stack 'Java 11', and startup command 'java -jar demo-0.0.1-SNAPSHOT.jar'.

4. Deploy the Application: a. Save and run the release pipeline. b. Verify that the application is successfully deployed and accessible via the web.

Azure DevOps Pipelines interface showing a successful deployment process. The pipeline has completed with 4 tasks: Initialize job, Download artifact, Azure Web App Deploy, and Finalize Job, all succeeded. The deployment process took 1m 33s.

Hello World!

Fix:

Jar was uploaded inside this /home/site/.....

Fixed in release pipeline and just in pipeline and started to work.

Configuration settings for the Azure Web App Deploy task. The startup command is set to "java -jar /home/site/www/root/demo-0.0.1-SNAPSHOT.jar".

Practical Task 6: Implement CI/CD with GitHub and Azure Pipelines

Requirements:

1. Create a GitHub Repository:

- Navigate to GitHub and create a new repository named MySampleApp.

2. Push Application Code:

- Push the application code from Task 4 to the new GitHub repository.

3. Set Up Azure Pipeline:

- In Azure DevOps, create a new pipeline that uses the GitHub repository as the source.

- Authenticate with GitHub and select the repository.

4. Define the Pipeline:

- Use the YAML pipeline editor to define the pipeline with steps to:

- Build the application.
- Run tests.
- Deploy to the Azure Web App (similar to Task 5).

- Enable GitHub Integration:
 - Configure the pipeline to trigger automatically on every push to the main branch.
 - Verify that the pipeline triggers successfully and the application is deployed to Azure.

Actually, I have done all of this steps in task 5. In task5 I did it in two ways: I have added deployment stage + using azure release pipelines. Both ways worked.

The screenshot shows the Azure DevOps interface for a pipeline run. At the top, the URL is yshyn / anatoliyakubshyn / Pipelines / MySampleApp-2 / 20250222.20. The title of the run is "#20250222.20 • Update azure-pipelines.yml for Azure Pipelines". A note says "This run is being retained as one of 3 recent runs by main (Branch)". Below the title, there's a summary section with tabs for Summary, Releases, Code Coverage, and Associated pipelines. The Summary tab is selected, showing it was manually run by Anatolii Yakubshyn. It details the repository (MySampleApp-2, main branch), the start time (yesterday at 19:36), and the duration (3m 3s). It also shows 0 work items and 1 published artifact. Below this, there are tabs for Stages and Jobs. The Stages tab shows a flow from "Build and Test" to "Deploy". The "Build and Test" stage completed 1 job in 39s and produced 1 artifact. The "Deploy" stage completed 1 job in 45s. There are also tabs for Tests and coverage, and a "View change" button.

But in task 5 (I used another code then in task 4 because I had problems with deployment of static content that react produces (I suppose for this is needed another service like Blob storage))

```

1 trigger:
2   - main
3
4 pool:
5   vmImage: default
6   dependsOn:
7     - agent.name -equals ubuntu-runner
8
9 stages:
10 - stage: Build
11   displayName: Build and Test
12   jobs:
13     - job: Package
14       steps:
15         - script: |
16           sudo apt update
17           sudo apt install -y openjdk-11-jdk
18           displayname: 'Install Java 11'
19
20         - script: |
21           sudo apt install -y maven
22           mvn -v
23           displayname: 'Install Maven'
24
25         - script: mvn package
26           displayname: 'Run mvn package'
27
28         - script: sudo apt-get -y install zip
29           displayname: 'Install zip'
30
31       - task: ArchiveFiles@2
32         inputs:
33           rootFolderOrFile: 'target/demo-0.0.1-SNAPSHOT.jar'
34           archiveType: 'zip'
35           archiveFile: '$(Build.ArtifactStagingDirectory)/build.zip'
36
37       - task: PublishBuildArtifacts@0
38         displayName: 'Publish build Artifacts'
39         inputs:
40           pathToPublish: '$(Build.ArtifactStagingDirectory)/build.zip'
41           artifactName: 'build'
42
43 - stage: Deploy
44   dependsOn: Build
45   displayName: Deploy
46   condition: succeeded()
47   jobs:
48     - job: Deploy
49       steps:
50         - task: DownloadBuildArtifacts@0
51           displayName: 'Download Build Artifacts'
52           inputs:
53             connectionType: 'AzureRM'
54             azureSubscription: 'Azure subscription 1 (98ae428-d8c1-4afe-bdf2-4e08593901a8)'
55             artifactName: 'current'
56             downloadPath: '$(System.ArtifactsDirectory)'
57
58       - task: AzureWebAppDeployment@0
59         inputs:
60           connectionType: 'AzureRM'
61           azureSubscription: 'Azure subscription 1 (98ae428-d8c1-4afe-bdf2-4e08593901a8)'
62           appType: 'webapp'
63           packagePath: '$(System.ArtifactsDirectory)/build.zip'
64           packageForLinux: '$(Pipeline.Workspace)/**/build/build.zip'
65           runInIIS: true
66           startCommand: 'java -jar /site/wwwroot/demo-0.0.1-SNAPSHOT.jar --server.port=80'
67           deploymentType: 'linux'

```

All pipeline runs

Description	Stages	Created
Update azure-pipelines.yml for Azure Pipelines #20250222.20 on MySampleApp-2		Yesterday 3m 3s

Practical Task 7: Use Azure Artifacts to Manage Dependencies

Requirements:

1. Create an Azure Artifacts Feed:

a. In Azure DevOps, navigate to Artifacts and create a new feed named MyDependenciesFeed.

The screenshot shows the Azure DevOps interface for managing artifacts. On the left, there's a sidebar with links for Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. The 'Artifacts' link is currently selected. The main area has a header with 'AnatoliYakubshyn' and a search bar. Below the header, there are buttons for 'Connect to Feed', '+ Create Feed', 'Search Upstream Sources', and 'Recycle Bin'. A large central area features a blue cloud icon with an arrow pointing up, labeled 'Connect to the feed to get started'. Below this, there's a 'Connect to Feed' button and a 'Search Upstream Sources' button. At the bottom, there's a link 'Learn more about Azure Artifacts'.

Create Feed

Create new feed

X

Feeds host your packages and let you control permissions.

Name *

MyDependenciesFeed

Visibility

Members of your Microsoft Entra tenant

- Any member of your Microsoft Entra tenant can view the packages in this feed

Members of AnatoliiYakubshyn

- Any member of your organization can view the packages in this feed

Specific people

- Only users you grant access to can view the packages in this feed

Upstream sources

- Include packages from common public sources

For example: nuget.org, npmjs.com

Scope

Project: anatoliyyakubshyn (Recommended)

- The feed will be scoped to the anatoliyyakubshyn project.

- Organization

Cancel

Create

2. Publish Dependencies:

a. Modify the build pipeline to include a task that publishes the application's dependencies

(e.g., npm packages or NuGet packages) to the Azure Artifacts feed.

b. Use commands like npm publish or dotnet nuget push in the pipeline.

3. Restore Dependencies: a. Update the build pipeline to

The screenshot shows the Azure DevOps interface for connecting a Maven repository. On the left, under 'Artifacts', 'Maven' is selected. In the center, the 'Connect to feed' section is open, showing configuration for 'MyDependenciesFeed'. It includes sections for 'Project setup' (with XML code for repositories and distribution management), 'Restore packages' (with command 'mvn install'), and 'Publish packages' (with command 'mvn deploy'). A note at the bottom says 'Finally, generate a [Personal Access Token](#) with Publishing read & write scopes and paste it into the <password> tag.'

```
<description>Demo project for Spring Boot</description>
<repositories>
    <repository>
        <id>MyDependenciesFeed</id>
        <url>
            https://pkgs.dev.azure.com/AnatoliiYakubshyn/anatoliyyakubshyn/_packaging/MyDependenciesFeed/maven/v1
        </url>
        <releases>
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled>
        </snapshots>
    </repository>
</repositories>
<distributionManagement>
    <repository>
        <id>MyDependenciesFeed</id>
        <url>https://pkgs.dev.azure.com/AnatoliiYakubshyn/anatoliyyakubshyn/_packaging/MyDependenciesFeed/maven/v1</url>
        <releases>
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled>
        </snapshots>
    </repository>
</distributionManagement>
</project>
```

Create a new personal access token

Name

Organization

Expiration (UTC)

Scopes
Authorize the scope of access associated with this token

Scopes Full access Custom defined

Code
Source code, repositories, pull requests, and notifications

Read Read & write Read, write, & manage Full Status

Build
Artifacts, definitions, requests, queue a build, and update build properties

Read Read & execute

Release
Read, update, and delete releases, release pipelines, and stages

Read Read, write, & execute Read, write, execute, & manage

Test Management
Read, create, and update test plans, cases, and results

Read Read & write

Packaging
Create, read, update, and delete feeds and packages

Read Read & write Read, write, & manage

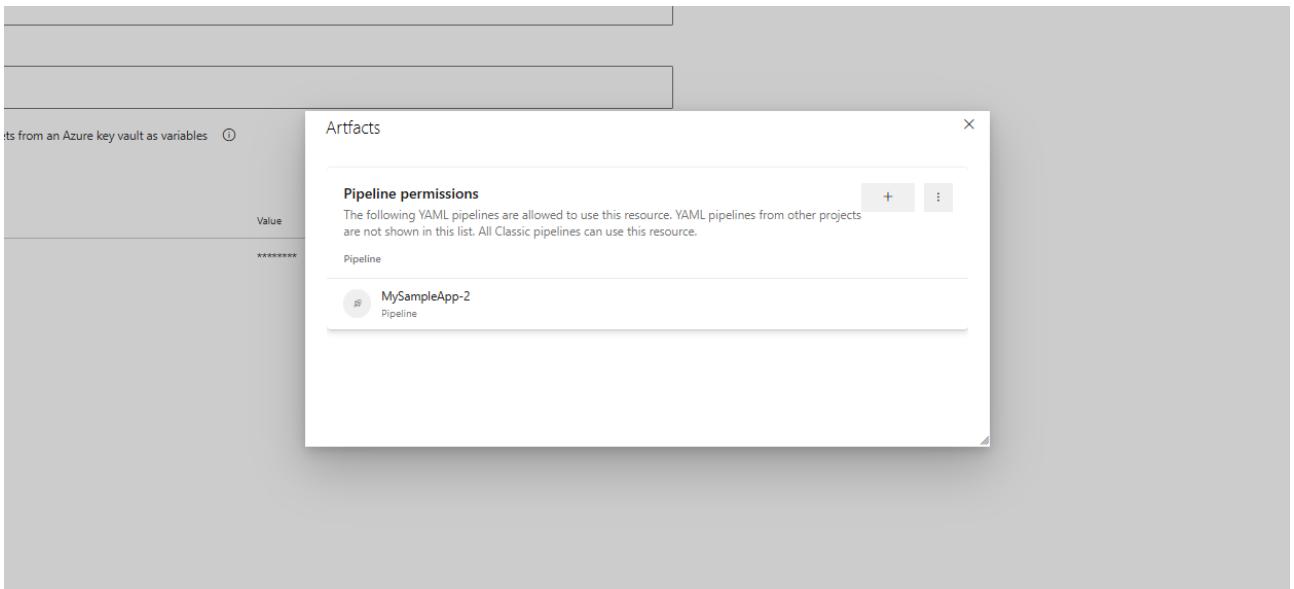
[Show all scopes \(30 more\)](#)

Create **Cancel**

```
1 trigger:
2   - main
3
4 pool:
5   name: default
6   demands:
7     - agent.name -equals Ubuntu-runner
8
9 variables:
10   - group: "Artifacts"
11
12 stages:
13   - stage: Build
14     displayName: Build and Test
15     jobs:
16       - job: Publish_Dependencies
17         steps:
18           - script: |
19             sudo apt update
20             sudo apt install -y openjdk-11-jdk
21             java -version
22             displayName: 'Install Java 11'
23
24           - script: |
25             sudo apt install -y maven
26             mvn -v
27             displayName: 'Install Maven'
28           - script: |
29             echo "<settings xmlns=\"http://maven.apache.org/SETTINGS/1.0.0\""
30               xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\""
31               xsi:schemaLocation=\"http://maven.apache.org/SETTINGS/1.0.0
32                               https://maven.apache.org/xsd/settings-1.0.0.xsd\">
33               <servers>
34                 <server>
35                   <id>MyDependenciesFeed</id>
36                   <username>AnatoliiYakubyshyn</username>
37                   <password>$(PAT)</password>
38                 </server>
39               </servers>
40             </settings>" > ~/.m2/settings.xml
41             displayName: 'Generate settings.xml'
42
43           - script:
44             mvn deploy
```

```
43     - script:
44         mvn deploy
45     - job: Package
46         dependsOn: Publish_Dependencies
47         steps:
48             - script: |
49                 sudo apt update
50                 sudo apt install -y openjdk-11-jdk
51                 java -version
52             displayName: 'Install Java 11'
53
54             - script: |
55                 sudo apt install -y maven
56                 mvn -v
57             displayName: 'Install Maven'
58             - script: |
59                 echo "<settings xmlns=\"http://maven.apache.org/SETTINGS/1.0.0\""
60                 xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\""
61                 xsi:schemaLocation=\"\"http://maven.apache.org/SETTINGS/1.0.0
62                                     https://maven.apache.org/xsd/settings-1.0.0.xsd\"">
63                     <servers>
64                         <server>
65                             <id>MyDependenciesFeed</id>
66                             <username>AnatoliiYakubyshyn</username>
67                             <password>$(PAT)</password>
68                         </server>
69                     </servers>
70                 </settings>" > ~/.m2/settings.xml
71             displayName: 'Generate settings.xml'
72
73             - script: mvn install
74
75             - script: mvn package
```

Adjustment



The screenshot shows the 'Variables' tab of the pipeline configuration for 'MySampleApp-2'. The pipeline variables section is visible, showing a table with columns 'Name' and 'Value'. A row under 'Artifacts (1)' is expanded, showing a single item. Navigation tabs include 'YAML', 'Variables', 'Triggers', 'History', 'Save & queue', 'Discard', 'Summary', 'Queue', and '...'. A sidebar on the left lists 'Pipeline variables', 'Variable groups', and 'Predefined variables'. A search bar and other UI elements are at the top right.

Verify Dependency Management:

iYakubshyn / anatoliyakubshyn / Pipelines / MySampleApp-2 / 20

Jobs in run #20250223.4
MySampleApp-2

Build and Test

- > **Publish_Dependencies** 30s
- ✓ **Package** 46s
 - ✓ Initialize job 2s
 - ✓ Checkout MySampleApp-2@main... 3s
 - ✓ Install Java 11 4s
 - ✓ Install Maven 2s
 - ✓ Generate settings.xml 2s
 - ✓ CmdLine 9s
 - ✓ Run mvn package 8s
 - ✓ Install zip 3s
 - ✓ ArchiveFiles 3s
 - ✓ Publish Build Artifacts 6s
 - ✓ Post-job: Checkout MySampleA... <1s
 - ✓ Finalize Job <1s

MyDependencies...

Type	Package	Views	Source	Push date	Description	Download...	Users
M	com.example.demo		This feed	4m ago	Demo project for Spring Boot	0	0

Practical Task 8: Deploy Infrastructure as Code (IaC) with Bicep

1. Create a Bicep File:

- Create a Bicep file named `main.bicep` that defines the infrastructure for your application, including an Azure Web App and a SQL Database.

```
...  & main.bicep X
  & main.bicep > {} sqlDatabase
  1   @description('The name of the web app')
  2   param webAppName string
  3
  4   @description('The location for the resources')
  5   param location string = resourceGroup().location
  6
  7   @description('The name of the SQL server')
  8   param sqlServerName string
  9
 10  @description('The SQL admin username')
 11  param sqlAdminUser string
 12
 13  @secure()
 14  @description('The SQL admin password')
 15  param sqlAdminPassword string
 16
 17  // SQL Server
 18  <resource sqlServer 'Microsoft.Sql/servers@2022-08-01-preview' = {
 19    name: sqlServerName
 20    location: location
 21    properties: {
 22      administratorLogin: sqlAdminUser
 23      administratorLoginPassword: sqlAdminPassword
 24    }
 25  }
 26
 27  // SQL Database
 28  <resource sqlDatabase 'Microsoft.Sql/servers/databases@2022-08-01-preview' = [
 29    parent: sqlServer
 30    name: 'mydatabase'
 31    location: location
 32    properties: {
 33      collation: 'SQL_Latin1_General_CI_AS'
 34    }
 35  ]
 36
 37  // App Service Plan
 38  <resource appServicePlan 'Microsoft.Web/serverfarms@2021-02-01' = {
 39    name: '${webAppName}-plan'
 40    location: location
 41    sku: {
 42      name: 'B1'
 43      tier: 'Basic'
 44    }
 45  }
 46
 47  // Web App
 48  <resource webApp 'Microsoft.Web/sites@2021-02-01' = {
 49    name: webAppName
 50    location: location
 51    properties: {
 52      serverFarmId: appServicePlan.id
 53    }
 54  }
```

2. Create a New Pipeline:

- Set up a new pipeline in Azure DevOps that includes a stage for deploying the Bicep file.

3. Deploy Using Azure CLI:

- Use the Azure CLI task in the pipeline to deploy the Bicep file.

- The command should look like: `az deployment group create --resource-group --template-file main.bicep`

The screenshot shows the Azure DevOps Pipeline Library interface. A new pipeline group named 'SqlDatabaseCreds' is being created. The 'Variables' section contains a single variable 'adminPassword' with the value '*****'. The 'Description' field is empty. The 'Variables' section has a note: 'Link secrets from an Azure key vault as variables' with a checkbox.

```
task8-bicep / azure-pipelines.yml * ⓘ Show assistant

1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your code.
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7   - main
8
9 variables:
10   - group: SqlDatabaseCreds
11
12 pool:
13   name: default
14   demands:
15     - agent.name -equals Ubuntu-runner
16
17 stages:
18   - stage: DeployInfrastructure
19     displayName: 'Deploy Infrastructure using Bicep'
20   - stage: Deploy
21     job: Deploy
22       displayName: 'Deploy Bicep'
23   - stage: CleanUp
24     job: CleanUp
25       task: AzureCLI@2
26         inputs:
27           azureSubscription: $(azureSubscription)
28           type: KeyVaultSecret
29           scriptLocation: inlineScript
30           inlineScript: |
31             az deployment group create --resource-group $(resourceGroupName) --template-file main.bicep --parameters webappName="yakubsyn-web-app" sqlServerName="yakubsyn-sql-server" sqlAdminUser=$(adminUsername) sqlAdminPassword=$(adminPassword) location=$(location)
32
33
```

4. Run the Pipeline: a. Save and run the pipeline to ensure that the infrastructure is provisioned correctly in Azure. b. Verify that the resources are created in the Azure portal.

The screenshot shows the Azure DevOps Pipeline Review screen. The pipeline is named 'task8-bicep / azure-pipelines.yml'. The 'Review' tab is selected. The pipeline configuration is identical to the one shown in the previous screenshot.

The screenshot shows the Azure DevOps Pipeline Variables screen. The 'Variables' tab is selected. A variable group named 'SqlDatabaseCreds (\$)' is listed under 'Variable groups'. It contains a single variable 'adminPassword' with the value '*****'.

Link secrets from an Azure key vault as variables [?](#)

Variables

Name ↑	Value
adminPassword	*****
adminUsername	*****
azureSubscription	Azure subscription 1 (9a6ae428-d8c3-44fe-bdf2-4e08593901a0)
location	uksouth
resourceGroupName	AnatoliYakubshyn

[+ Add](#)

changed location to uksouth and :

shyn / anatoliyakubshyn / Pipelines / task8-bicep / 20250223.6

[← Jobs in run #20250223.6](#)

[Deploy Infrastructure using Bicep](#)

[Deploy Bicep](#) 2m 24s

- Initialize job 3s
- Checkout task8-bicep@main to s 3s
- Deploy Bicep File 2m 16s
- Post-job: Checkout task8-bicep... <1s
- Finalize Job <1s
- Report build status <1s

[View raw log](#)

Deploy Bicep

```

1 Pool: Default
2 Agent: Ubuntu-runner
3 Started: Today at 12:38
4 Duration: 2m 24s
5
6 ▶ Job preparation parameters

```

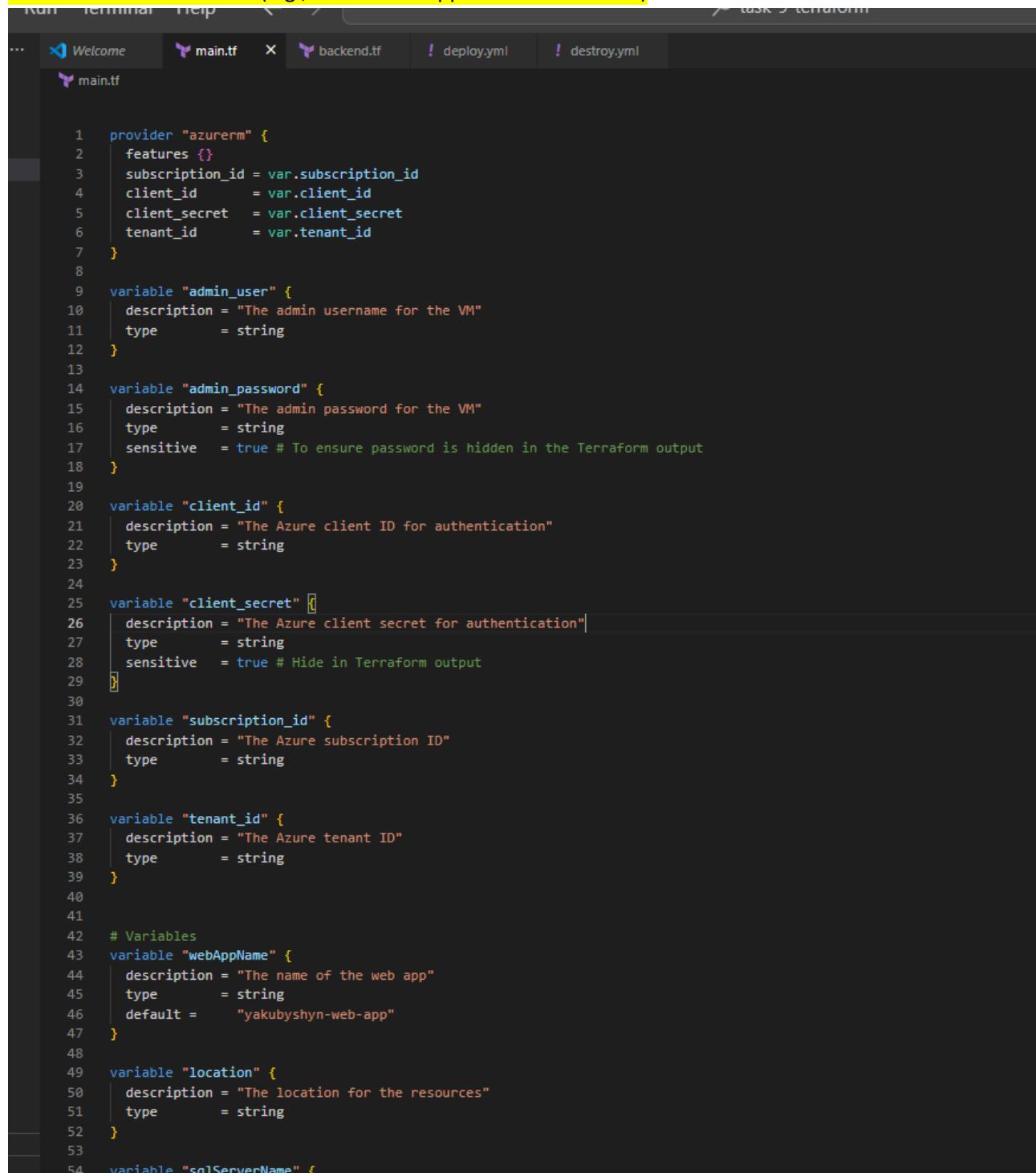
Resources created successfully

<input type="checkbox"/> Name ↗	Type ↗	Location ↗	<input type="checkbox"/> No grouping	<input type="checkbox"/> List view
<input type="checkbox"/> mydatabase (yakubshyn-sql-server/mydatabase)	SQL database	UK South
<input type="checkbox"/> Ubuntu-runner	Virtual machine	Poland Central
<input type="checkbox"/> Ubuntu-runner-ip	Public IP address	Poland Central
<input type="checkbox"/> Ubuntu-runner-ing	Network security group	Poland Central
<input type="checkbox"/> Ubuntu-runner-vnet	Virtual network	Poland Central
<input type="checkbox"/> ubuntu-runner345_s1	Network interface	Poland Central
<input type="checkbox"/> Ubuntu-runner_key	SSH key	Poland Central
<input type="checkbox"/> Ubuntu-runner_ODisk_1_90229d04434e4aa8ba48975c152f60b1	Disk	Poland Central
<input type="checkbox"/> yakubshyn-sql-server	SQL server	UK South
<input type="checkbox"/> yakubshyn-web-app	App Service	UK South
<input type="checkbox"/> yakubshyn-web-app-plan	App Service plan	UK South
<input type="checkbox"/> yakubshynterraform	Storage account	Poland Central

Practical Task 9: Implement Infrastructure as Code (IaC) with Terraform

Requirements: 1.

Create a Terraform Configuration: a. Create a Terraform configuration file (main.tf) that defines the same infrastructure as in Task 8 (e.g., Azure Web App and SQL Database)



```
1 provider "azurerm" {
2   features {}
3   subscription_id = var.subscription_id
4   client_id       = var.client_id
5   client_secret   = var.client_secret
6   tenant_id       = var.tenant_id
7 }
8
9 variable "admin_user" {
10  description = "The admin username for the VM"
11  type        = string
12 }
13
14 variable "admin_password" {
15  description = "The admin password for the VM"
16  type        = string
17  sensitive   = true # To ensure password is hidden in the Terraform output
18 }
19
20 variable "client_id" {
21  description = "The Azure client ID for authentication"
22  type        = string
23 }
24
25 variable "client_secret" [
26  description = "The Azure client secret for authentication"
27  type        = string
28  sensitive   = true # Hide in Terraform output
29 ]
30
31 variable "subscription_id" {
32  description = "The Azure subscription ID"
33  type        = string
34 }
35
36 variable "tenant_id" {
37  description = "The Azure tenant ID"
38  type        = string
39 }
40
41
42 # Variables
43 variable "webAppName" {
44  description = "The name of the web app"
45  type        = string
46  default     = "yakubyshyn-web-app"
47 }
48
49 variable "location" {
50  description = "The location for the resources"
51  type        = string
52 }
53
54 variable "sqlServerName" {
```

```
...  Welcome   main.tf  X  backend.tf  !  deploy.yml  !  destroy.yml
     main.tf

49  variable "location" {
50    description = "The location for the resources"
51    type        = string
52  }
53
54  variable "sqlServerName" {
55    description = "The name of the SQL server"
56    type        = string
57    default    = "yakubshyn-sql-server"
58  }
59
60  variable "sqlAdminUser" {
61    description = "The SQL admin username"
62    type        = string
63    sensitive   = true
64  }
65
66  variable "sqlAdminPassword" {
67    description = "The SQL admin password"
68    type        = string
69    sensitive   = true
70  }
71
72
73  resource "azurerm_mssql_server" "sql_server" {
74    name          = var.sqlServerName
75    location      = var.location
76    resource_group_name = "AnatoliiYakubyshyn"
77    version       = "12.0"
78    administrator_login = var.sqlAdminUser
79    administrator_login_password = var.sqlAdminPassword
80  }
81
82  resource "azurerm_mssql_database" "sql_database" {
83    name          = "mydatabase"
84    server_id     = azurerm_mssql_server.sql_server.id # Correct way to reference the server
85    collation     = "SQL_Latin1_General_CI_AS"
86  }
87
88  resource "azurerm_app_service_plan" "app_service_plan" {
89    name          = "${var.webAppName}-plan"
90    location      = var.location
91    resource_group_name = "AnatoliiYakubyshyn"
92    sku {
93      tier = "Basic"
94      size = "B1"
95    }
96  }
97
98  resource "azurerm_linux_web_app" "web_app" {
99    name          = var.webAppName
100   location      = var.location
101   resource_group_name = "AnatoliiYakubyshyn"
102   service_plan_id = azurerm_app_service_plan.app_service_plan.id
103   site_config {
```

A screenshot of a terminal window titled "task-9-terraform". The window shows several tabs: "Welcome", "main.tf", "backend.tf" (which is currently selected), "deploy.yml", "destroy.yml", and "backend.tf". The "backend.tf" tab contains the following Terraform code:

```
1  terraform {  
2      backend "azurerm" {  
3          resource_group_name = "AnatoliiYakubyshyn" # Static value  
4          storage_account_name = "yakubyshynterraform" # Default value to be overridden  
5          container_name = "tfstate" # Static value  
6          key = "terraform.tfstate" # Static value  
7      }  
8  }
```

2. Set Up a New Azure Pipeline: a. Create a new Azure Pipeline that runs Terraform commands. b. Use the following steps in the pipeline: terraform init to initialize the Terraform working directory. terraform plan to create an execution plan. terraform apply -auto-approve to apply the changes

3. Configure Triggers: a. Set up the pipeline to trigger on changes to the Terraform files in the repository.

A screenshot of an Azure Pipeline YAML configuration file named "deploy.yml". The file defines a single job named "main" which runs on an Ubuntu runner. It includes variables for a variable group, stages for initializing Terraform, installing the Azure CLI, installing Terraform, setting environment variables for Azure SP authentication, initializing Terraform with a backend config override, and finally running a Terraform plan command.

```
1  trigger:  
2  - main  
3  
4  pool:  
5      name: default  
6      demands:  
7          - agent.name -equals Ubuntu-runner  
8  
9  variables:  
10 | - group: "New variable group 13-Feb" # Must include variable: storageAccountName and other secrets  
11  
12  stages:  
13  - stage: Plan  
14      displayName: 'Terraform Plan'  
15      jobs:  
16          - job: TerraformPlan  
17              displayName: 'Terraform Plan'  
18              steps:  
19                  # Install unzip if not present  
20                  - script:  
21                      sudo apt-get update && sudo apt-get install -y unzip  
22                      displayName: 'Install unzip'  
23                      workingDirectory: $(System.DefaultWorkingDirectory)  
24  
25          - script:  
26              # Install Azure CLI  
27              echo "Installing Azure CLI..."  
28              curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash  
29  
30          # Install Terraform  
31          TERRAFORM_VERSION="1.3.0"  
32          echo "Downloading Terraform version ${TERRAFORM_VERSION}..."  
33          curl -fSL "https://releases.hashicorp.com/terraform/${TERRAFORM_VERSION}/terraform_${TERRAFORM_VERSION}_linux_amd64.zip" -o terraform.zip  
34          unzip terraform.zip  
35          sudo mv terraform /usr/local/bin/  
36          rm terraform.zip  
37          terraform version  
38          az --version # Verify Azure CLI installation  
39          displayName: 'Install Terraform and Azure CLI'  
40          workingDirectory: $(System.DefaultWorkingDirectory)  
41  
42          # Set environment variables for Azure SP authentication  
43          - script:  
44              echo "#vso[task.setvariable variable=ARM_CLIENT_ID]$(client_id)"  
45              echo "#vso[task.setvariable variable=ARM_CLIENT_SECRET]$(client_secret)"  
46              echo "#vso[task.setvariable variable=ARM_SUBSCRIPTION_ID]$(subscription_id)"  
47              echo "#vso[task.setvariable variable=ARM_TENANT_ID]$(tenant_id)"  
48              echo "#vso[task.setvariable variable=LOCATION]$(location)"  
49              echo "#vso[task.setvariable variable=ADMIN_USERNAME]$(adminUsername)"  
50              echo "#vso[task.setvariable variable=ADMIN_PASSWORD]$(adminPassword)"  
51  
52          displayName: 'Set environment variables for Azure Service Principal'  
53  
54          # Initialize Terraform with backend config override  
55          - script:  
56              terraform init -reconfigure  
57              displayName: 'Terraform Init with Backend Config'  
58              workingDirectory: $(System.DefaultWorkingDirectory)  
59  
60          # Run Terraform plan and output the plan file  
61          - script:  
62              terraform plan -out=tfplan  
63              ...var="admin_user=$(admin_user)\"  
64              ...var="admin_password=$(admin_password)\"  
65              ...var="location=$(location)\"
```

```

10   steps:
11     workingDirectory: $(System.DefaultWorkingDirectory)
12
13     # Run Terraform plan and output the plan file
14     - script: |
15       terraform plan -out=tfplan \
16       --var="admin_user=${admin_user}" \
17       --var="admin_password=${admin_password}" \
18       --var="client_id=${ARM_CLIENT_ID}" \
19       --var="client_secret=${ARM_CLIENT_SECRET}" \
20       --var="subscription_id=${ARM_SUBSCRIPTION_ID}" \
21       --var="tenant_id=${ARM_TENANT_ID}" \
22       --var="location=${LOCATION}" \
23       --var="sqlAdminUser=${ADMIN_USERNAME}" \
24       --var="sqlAdminPassword=${ADMIN_PASSWORD}"
25
26     displayName: 'Terraform Plan'
27     workingDirectory: $(System.DefaultWorkingDirectory)
28
29   # List files to verify the tfplan exists
30   - script: |
31     | ls -al $(System.DefaultWorkingDirectory)
32     displayName: 'Verify tfplan exists'
33     workingDirectory: $(System.DefaultWorkingDirectory)
34
35   # Publish the tfplan artifact for use in the Deploy stage
36   - publish: $(System.DefaultWorkingDirectory)/tfplan
37     artifact: tfplan
38     displayName: 'Publish tfplan artifact'
39
40   - stage: Deploy
41     displayName: 'Terraform Apply'
42     dependsOn: Plan
43     condition: succeeded()
44   jobs:
45     - job: TerraformApply
46       displayName: 'Terraform Apply'
47       steps:
48         # Install unzip on the deploy agent
49         - script: |
50           sudo apt-get update && sudo apt-get install -y unzip
51           displayName: 'Install unzip'
52           workingDirectory: $(System.DefaultWorkingDirectory)
53
54     # Re-install Terraform on the deploy agent
55     - script: |
56       # Install Azure CLI
57       echo "Installing Azure CLI..."
58       curl -fsSL https://aka.ms/InstallAzureCLIDeb | sudo bash
59
60       # Install Terraform
61       TERRAFORM_VERSION="1.3.0"
62       echo "Downloading Terraform version ${TERRAFORM_VERSION}..."
63       curl -fsSL "https://releases.hashicorp.com/terraform/${TERRAFORM_VERSION}/terraform_${TERRAFORM_VERSION}_linux_amd64.zip" -o terraform.zip
64       unzip terraform.zip
65       sudo mv terraform /usr/local/bin/
66       rm terraform.zip
67       terraform version
68       az --version # Verify Azure CLI installation
69     displayName: 'Install Terraform'
70     workingDirectory: $(System.DefaultWorkingDirectory)
71
72   - stage: Deploy
73     displayName: 'Terraform Apply'
74     dependsOn: Plan
75     condition: succeeded()
76   jobs:
77     - job: TerraformApply
78       displayName: 'Terraform Apply'
79       steps:
80         # Install unzip on the deploy agent
81         - script: |
82           sudo apt-get update && sudo apt-get install -y unzip
83           displayName: 'Install unzip'
84           workingDirectory: $(System.DefaultWorkingDirectory)
85
86   # Publish the tfplan artifact for use in the Deploy stage
87   - publish: $(System.DefaultWorkingDirectory)/tfplan
88     artifact: tfplan
89     displayName: 'Publish tfplan artifact'
90
91   - stage: Deploy
92     displayName: 'Terraform Apply'
93     dependsOn: Plan
94     condition: succeeded()
95   jobs:
96     - job: TerraformApply
97       displayName: 'Terraform Apply'
98       steps:
99         # Install unzip on the deploy agent
100        - script: |
101          sudo apt-get update && sudo apt-get install -y unzip
102          displayName: 'Install unzip'
103          workingDirectory: $(System.DefaultWorkingDirectory)
104
105   # Re-install Terraform on the deploy agent
106   - script: |
107     # Install Azure CLI
108     echo "Installing Azure CLI..."
109     curl -sl https://aka.ms/installazuredel | sudo bash
110
111     # Install Terraform
112     TERRAFORM_VERSION="1.3.0"
113     echo "Downloading Terraform version ${TERRAFORM_VERSION}..."
114     curl -fsSL "https://releases.hashicorp.com/terraform/${TERRAFORM_VERSION}/terraform_${TERRAFORM_VERSION}_linux_amd64.zip" -o terraform.zip
115     unzip terraform.zip
116     sudo mv terraform /usr/local/bin/
117     rm terraform.zip
118     terraform version
119     az --version # Verify Azure CLI installation
120   displayName: 'Install Terraform'
121   workingDirectory: $(System.DefaultWorkingDirectory)
122
123

```

```

10   steps:
11     workingDirectory: $(System.DefaultWorkingDirectory)
12
13     # Run Terraform plan and output the plan file
14     - script: |
15       terraform plan -out=tfplan \
16       --var="admin_user=${admin_user}" \
17       --var="admin_password=${admin_password}" \
18       --var="client_id=${ARM_CLIENT_ID}" \
19       --var="client_secret=${ARM_CLIENT_SECRET}" \
20       --var="subscription_id=${ARM_SUBSCRIPTION_ID}" \
21       --var="tenant_id=${ARM_TENANT_ID}" \
22       --var="location=${LOCATION}" \
23       --var="sqlAdminUser=${ADMIN_USERNAME}" \
24       --var="sqlAdminPassword=${ADMIN_PASSWORD}"
25
26     displayName: 'Terraform Plan'
27     workingDirectory: $(System.DefaultWorkingDirectory)
28
29   # List files to verify the tfplan exists
30   - script: |
31     | ls -al $(System.DefaultWorkingDirectory)
32     displayName: 'Verify tfplan exists'
33     workingDirectory: $(System.DefaultWorkingDirectory)
34
35   # Publish the tfplan artifact for use in the Deploy stage
36   - publish: $(System.DefaultWorkingDirectory)/tfplan
37     artifact: tfplan
38     displayName: 'Publish tfplan artifact'
39
40   - stage: Deploy
41     displayName: 'Terraform Apply'
42     dependsOn: Plan
43     condition: succeeded()
44   jobs:
45     - job: TerraformApply
46       displayName: 'Terraform Apply'
47       steps:
48         # Install unzip on the deploy agent
49         - script: |
50           sudo apt-get update && sudo apt-get install -y unzip
51           displayName: 'Install unzip'
52           workingDirectory: $(System.DefaultWorkingDirectory)
53
54     # Re-install Terraform on the deploy agent
55     - script: |
56       # Install Azure CLI
57       echo "Installing Azure CLI..."
58       curl -fsSL https://aka.ms/InstallAzureCLIDeb | sudo bash
59
60       # Install Terraform
61       TERRAFORM_VERSION="1.3.0"
62       echo "Downloading Terraform version ${TERRAFORM_VERSION}..."
63       curl -fsSL "https://releases.hashicorp.com/terraform/${TERRAFORM_VERSION}/terraform_${TERRAFORM_VERSION}_linux_amd64.zip" -o terraform.zip
64       unzip terraform.zip
65       sudo mv terraform /usr/local/bin/
66       rm terraform.zip
67       terraform version
68       az --version # Verify Azure CLI installation
69     displayName: 'Install Terraform'
70     workingDirectory: $(System.DefaultWorkingDirectory)
71
72   - stage: Deploy
73     displayName: 'Terraform Apply'
74     dependsOn: Plan
75     condition: succeeded()
76   jobs:
77     - job: TerraformApply
78       displayName: 'Terraform Apply'
79       steps:
80         # Install unzip on the deploy agent
81         - script: |
82           sudo apt-get update && sudo apt-get install -y unzip
83           displayName: 'Install unzip'
84           workingDirectory: $(System.DefaultWorkingDirectory)
85
86   # Publish the tfplan artifact for use in the Deploy stage
87   - publish: $(System.DefaultWorkingDirectory)/tfplan
88     artifact: tfplan
89     displayName: 'Publish tfplan artifact'
90
91   - stage: Deploy
92     displayName: 'Terraform Apply'
93     dependsOn: Plan
94     condition: succeeded()
95   jobs:
96     - job: TerraformApply
97       displayName: 'Terraform Apply'
98       steps:
99         # Install unzip on the deploy agent
100        - script: |
101          sudo apt-get update && sudo apt-get install -y unzip
102          displayName: 'Install unzip'
103          workingDirectory: $(System.DefaultWorkingDirectory)
104
105   # Re-install Terraform on the deploy agent
106   - script: |
107     # Install Azure CLI
108     echo "Installing Azure CLI..."
109     curl -sl https://aka.ms/installazuredel | sudo bash
110
111     # Install Terraform
112     TERRAFORM_VERSION="1.3.0"
113     echo "Downloading Terraform version ${TERRAFORM_VERSION}..."
114     curl -fsSL "https://releases.hashicorp.com/terraform/${TERRAFORM_VERSION}/terraform_${TERRAFORM_VERSION}_linux_amd64.zip" -o terraform.zip
115     unzip terraform.zip
116     sudo mv terraform /usr/local/bin/
117     rm terraform.zip
118     terraform version
119     az --version # Verify Azure CLI installation
120   displayName: 'Install Terraform'
121   workingDirectory: $(System.DefaultWorkingDirectory)
122
123

```

```
GO Run Terminal Help task 3 terraform
Welcome main.tf backend.tf deploy.yml destroy.yml

1 trigger: none # Disable automatic triggering
2
3 pr: none # Disable PR triggers if you want manual-only execution
4
5 pool:
6   name: default
7   demands:
8     - agent.name -equals Ubuntu-runner
9
10 variables:
11   - group: "New variable group 13-Feb" # Must include variable: storageAccountName and other secrets
12
13 stages:
14   - stage: Destroy
15     displayName: 'Terraform Destroy'
16     jobs:
17       - job: TerraformDestroy
18         displayName: 'Terraform Destroy'
19         steps:
20           # Install unzip on the deploy agent
21           - script: |
22             sudo apt-get update && sudo apt-get install -y unzip
23             displayName: 'Install unzip'
24             workingDirectory: $(System.DefaultWorkingDirectory)
25
26           # Re-install Terraform on the deploy agent
27           - script: |
28             # Install Azure CLI
29             echo "Installing Azure CLI..."
30             curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
31
32             # Install Terraform
33             TERRAFORM_VERSION="1.3.0"
34             echo "Downloading Terraform version ${TERRAFORM_VERSION}..."
35             curl -fsSL "https://releases.hashicorp.com/terraform/${TERRAFORM_VERSION}/terraform_${TERRAFORM_VERSION}_linux_amd64.zip" -o terraform.zip
36             unzip terraform.zip
37             sudo mv terraform /usr/local/bin/
38             rm terraform.zip
39             terraform version
40             az --version # Verify Azure CLI installation
41             displayName: 'Install Terraform'
42             workingDirectory: $(System.DefaultWorkingDirectory)
43
44           # Set environment variables for Azure SP authentication
45           - script: |
46             echo "#vso[task.setvariable variable=ARM_CLIENT_ID]${client_id}"
47             echo "#vso[task.setvariable variable=ARM_CLIENT_SECRET]${client_secret}"
48             echo "#vso[task.setvariable variable=ARM_SUBSCRIPTION_ID]${subscription_id}"
49             echo "#vso[task.setvariable variable=ARM_TENANT_ID]${tenant_id}"
50             echo "#vso[task.setvariable variable=LOCATION]${location}"
51             echo "#vso[task.setvariable variable=ADMIN_USERNAME]${adminUsername}"
52             echo "#vso[task.setvariable variable=ADMIN_PASSWORD]${adminPassword}"
53             displayName: 'Set environment variables for Azure Service Principal'
54
55           # Reinitialize Terraform with the same backend override
56           - script: |
57             | terraform init -reconfigure
58             displayName: 'Terraform Init with Backend Config'
59             workingDirectory: $(System.DefaultWorkingDirectory)
60
61           # Destroy
62           - script: |
63             | terraform destroy -auto-approve \
64             - script: |
65               | terraform plan -out=tfplan \
```

```

46      echo "##vso[task.setvariable variable=ARM_CLIENT_ID]${client_id}"
47      echo "##vso[task.setvariable variable=ARM_CLIENT_SECRET]${client_secret}"
48      echo "##vso[task.setvariable variable=ARM_SUBSCRIPTION_ID]${subscription_id}"
49      echo "##vso[task.setvariable variable=ARM_TENANT_ID]${tenant_id}"
50      echo "##vso[task.setvariable variable=LOCATION]${location}"
51      echo "##vso[task.setvariable variable=ADMIN_USERNAME]${adminUsername}"
52      echo "##vso[task.setvariable variable=ADMIN_PASSWORD]${adminPassword}"
53      displayName: 'Set environment variables for Azure Service Principal'
54
55      # Reinitialize Terraform with the same backend override
56      - script: |
57          | terraform init -reconfigure
58          | displayName: 'Terraform Init with Backend Config'
59          | workingDirectory: ${System.DefaultWorkingDirectory}
60
61      # Destroy
62      - script: |
63          | terraform destroy -auto-approve \
64          | -var="admin_user=${admin_user}" \
65          | -var="admin_password=${admin_password}" \
66          | -var="client_id=${ARM_CLIENT_ID}" \
67          | -var="client_secret=${ARM_CLIENT_SECRET}" \
68          | -var="subscription_id=${ARM_SUBSCRIPTION_ID}" \
69          | -var="tenant_id=${ARM_TENANT_ID}" \
70          | -var="location=${LOCATION}" \
71          | -var="sqlAdminUser=${ADMIN_USERNAME}" \
72          | -var="sqlAdminPassword=${ADMIN_PASSWORD}"
73          | displayName: 'Terraform Destroy'
74          | workingDirectory: ${System.DefaultWorkingDirectory}
75

```

4. Run the Pipeline: a. Save and run the pipeline to provision the infrastructure in Azure. b. Verify that the resources are created as expected.

#20250223.16 • added backend

This run is being retained as one of 3 recent runs by main (Branch).

Triggered by Anatoliy Yakubshyn

Repository and version

- task-9-terraform
- main · b1b6ae69

Time started and elapsed

- Today at 14:08
- 4m 37s

Related

- 0 work items
- 1 published

Tests and coverage

- Get started

Stages

- Terraform Plan
- Terraform Apply

1 job completed 46s 1 job completed 3m 38s

Microsoft Azure

Home > AnatoliiYakubyshyn

Resource group

Search Create Manage view Delete resource group Refresh Export to CSV Open query Assi

Overview

Essentials

Subscription (move) : Azure subscription 1
Subscription ID : 9a6ae428-d8c3-44fe-bdf2-4e08593901a0
Tags (edit) : Add tags

Resources Recommendations (13)

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 12 of 12 records. Show hidden types

	Name
<input type="checkbox"/>	mydatabase (yakubyshyn-sql-server/mydatabase)
<input type="checkbox"/>	Ubuntu-runner
<input type="checkbox"/>	Ubuntu-runner-ip
<input type="checkbox"/>	Ubuntu-runner-nsg
<input type="checkbox"/>	Ubuntu-runner-vnet
<input type="checkbox"/>	ubuntu-runner345_z1
<input type="checkbox"/>	Ubuntu-runner_key
<input type="checkbox"/>	Ubuntu-runner_OsDisk_1_90229d04434e4aa8ba48975c152f60b1
<input type="checkbox"/>	yakubyshyn-sql-server
<input type="checkbox"/>	yakubyshyn-web-app
<input type="checkbox"/>	yakubyshyn-web-app-plan
<input type="checkbox"/>	yakubyshynterraform

/ anatoliyakubshyn / Pipelines / task-9-terraform (13) / 20250223.3

Jobs in run #20250223.3

task-9-terraform (13)

Terraform Destroy

Terraform Destroy 1m 24s

- Initialize job 2s
- Checkout task-9-terraform@main... 3s
- Install unzip 4s
- Install Terraform 9s
- Set environment variables for Azu... 1s
- Terraform init with Backend Config 8s
- Terraform Destroy 52s
- Post-job: Checkout task-9-terra... <1s
- Finalize Job <1s
- Report build status <1s

Terraform Destroy

```

1 Pool: Default
2 Queue: Just now [manage_parallel_jobs]
3 Agent: Ubuntu-runner
4 Started: Just now
5 Duration: 1m 24s
6
7 The agent request is already running or has already completed.
8 * Job preparation parameters
9 $> Job: Terraform Destroy
10 $> Starting: Terraform Destroy
11 $> Async Command Start: DetectDockerContainer
12 $> Async Command End: DetectDockerContainer
13 $> Async Command Start: DetectDockerContainer
14 $> Async Command End: DetectDockerContainer
15 $> Async Command End: DetectDockerContainer
16 $> Finishing: Terraform Destroy

```

```

16
17 Terraform used the selected providers to generate the following execution
18 plan. Resource actions are indicated with the following symbols:
19 - destroy
20
21 Terraform will perform the following actions:
22
23 # azurerm_app_service_plan.app_service_plan will be destroyed
24 - resource "azurerm_app_service_plan" "app_service_plan" {
25   - id          = "/subscriptions/9a6ae428-d8c1-44fe-bdf2-4e08593901a0/resourceGroups/Anatoliyakubshyn/providers/Microsoft.Web/serverFarms/yakubshyn-web-app-plan" -> null
26   - is_xenon    = false -> null
27   - kind        = "Windows" -> null
28   - location    = "uksouth" -> null
29   - maximum_elastic_worker_count = 1 -> null
30   - maximum_number_of_workers     = 3 -> null
31   - name         = "yakubshyn-web-app-plan" -> null
32   - per_site_scaling           = false -> null
33   - reserved                 = false -> null
34   - resource_group_name       = "Anatoliyakubshyn" -> null
35   - tags                     = {} -> null
36   - zone_redundant           = false -> null
37
38   - sku {
39     - capacity = 1 -> null
40     - size     = "B1" -> null
41     - tier     = "Basic" -> null
42   }
43 }
44
45 # azurerm_linux_web_app.web_app will be destroyed
46 - resource "azurerm_linux_web_app" "web_app" {
47   - app_settings           = {} -> null
48   - client_affinity_enabled = false -> null
49   - client_certificate_enabled = false -> null
50   - client_certificate_mode = "Required" -> null
51   - custom_domain_verification_id = "yakubshyn-web-app.azurewebsites.net" -> null
52   - default_hostname        = true -> null
53   - enabled                = true -> null
54   - ftp_publish_basic_authentication_enabled = false -> null
55   - https_only              = false -> null
56   - id                      = "/subscriptions/9a6ae428-d8c1-44fe-bdf2-4e08593901a0/resourceGroups/Anatoliyakubshyn/providers/Microsoft.Web/sites/yakubshyn-web-app" -> null
57   - key_vault_reference_identity_id = "SystemAssigned" -> null
58   - kind                    = "app" -> null
59   - location                = "uksouth" -> null
60   - name                    = "yakubshyn-web-app" -> null
61   - outbound_ip_address_list = [
62     "10.108.102.154"
63   ]
64 }

```

Home >

Anatoliyakubshyn

Search Create Manage view Delete resource group Refresh Export to CSV Open query Assign tags Move Delete Export template Open in mobile

Deployments : 1 Failed 12 Succeeded JSON View

Subscription (1) : Azure subscription 1
Subscription ID : 9a6ae428-d8c1-44fe-bdf2-4e08593901a0
Location : UK South

Overview Activity log Access control (IAM) Tags Resource visualizer Events Settings

- Deployments
- Policies
- Properties
- Locks
- Cost Management
 - Cost analysis
 - Cost alerts (preview)
 - Budgets
 - Adviser recommendations

Monitoring Automation Help

Resources Recommendations (13)

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 8 of 8 records. Show hidden types

Type	Location	Actions
Virtual machine	Poland Central	
Public IP address	Poland Central	
Network security group	Poland Central	
Virtual network	Poland Central	
Network interface	Poland Central	
SSH key	Poland Central	
Disk	Poland Central	
Storage account	Poland Central	

Practical Task 10: Integrate Azure Test Plans with CI/CD Pipelines

Requirements: 1. Create a Test Plan:

a. Navigate to Azure Test Plans and create a new test plan that includes both manual and automated tests for your application.

The screenshot shows the Azure DevOps Organization Settings page. On the left, the 'Billing' section is selected, displaying usage statistics for MS Hosted CI/CD and Self-Hosted CI/CD. Below this, the 'Boards, Repos and Test Plans' section shows the current access level is 'Free'. It lists 'Basic users' at 5 and 'Basic + Test Plans' which has a trial period of 'Trial expires in 30 days'. At the bottom, the 'Settings' section shows the 'Access level' is set to 'Basic + Test Plans'.

The screenshot shows the Azure DevOps User settings page. A modal window titled 'Change access level' is open for the user 'Anatoliy Yakubshyn', showing the current access level is 'Basic + Test Plans' and was last accessed on 23.02.2025. Below this, the main page shows the 'Test Plans' section where a new test plan named 'HelloWorldSpring' is being created. The test plan details include 'Area Path' set to 'anatoliyakubshyn' and 'Iteration' set to 'anatoliyakubshyn'. The 'Create' button is visible at the bottom right of the form.

The screenshot shows the Azure DevOps Test Plan interface. A new test case is being created with the title "Validate Content of page by '/' path". The test case details include:

- Owner: Anatolii Yakubshyn
- Comments: 0
- Add Tag
- State: Design
- Area: anatoliyyakubshyn
- Reason: New
- Iteration: anatoliyyakubshyn

The "Steps" section contains one step: "1. navigate to a site with path '/' (home) page". The "Expected result" field shows "Page contains text: Hello World!". The "Attachments" section is empty.

The "Deployment" section provides instructions to track releases associated with this work item. The "Development" section includes an "Add link" button and a note about linking to Azure Repos commit, pull request, or branch.

The screenshot shows the IntelliJ IDEA code editor with the following files open:

- README.md
- pom.xml (demo)
- DemoApplicationTests.java (selected)
- SpringBootTest.class
- Test.class
- DemoApplic...

The code in `DemoApplicationTests.java` is:

```
1 package com.example.demo;
2
3 > import ...
4
5 > *
6
7 @SpringBootTest
8 class DemoApplicationTests {
9
10     > *
11     @Test
12     void contextLoads() {
13         assert new HelloWorldController().helloWorld().equals("Hello world!");
14     }
15 }
16
```

created an automated test

2. Modify the Build Pipeline:

a. Update the build pipeline to include a testing stage that runs automated tests using a testing framework (e.g., NUnit for .NET or Jest for Node.js)

```
parameters:
  - name: goal
steps:
  - script: |
    sudo apt update
    sudo apt install -y openjdk-11-jdk
    java -version
    displayName: 'Install Java 11'

  - script: |
    sudo apt install -y maven
    mvn -v
    displayName: 'Install Maven'

  - script: |
    echo "<settings xmlns='http://maven.apache.org/SETTINGS/1.0.0'
      xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
      xsi:schemaLocation='http://maven.apache.org/SETTINGS/1.0.0
      https://maven.apache.org/xsd/settings-1.0.0.xsd'>
<servers>
<server>
<id>MyDependenciesFeed</id>
<username>Anatoliyukubushyn</username>
<password>${PAT}</password>
</server>
</servers>
</settings>" > ~/.m2/settings.xml
    displayName: 'Generate settings.xml'

  - script: |
    mvn ${goal}
    displayName: "Execute mvn goal"
```

```
trigger:
- main

pool:
  name: default
  demands:
  - agent.name -equals Ubuntu-runner

variables:
- group: "Artifacts"

stages:
- stage: Test
  displayName: Test
  jobs:
  - job: Test
    steps:
    - validate:
        template: templates/execute-mvn-goal.yaml
        parameters:
          goal: test
      settings:
        task: PublishTestResults@2
        inputs:
          testResultsFormat: 'JUnit'
          testResultsFiles: '**/TEST-*.xml'
          testRunTitle: 'test-run'
          failTaskOnFailedTests: true
        condition: always()

- stage: Package
  displayName: Package
  dependsOn: Test
  jobs:
  - job: Publish_Dependencies
    steps:
    - validate:
        template: templates/execute-mvn-goal.yaml
        parameters:
          goal: deploy
      job: Package
      dependsOn: Publish_Dependencies
      steps:
      - validate:
          template: templates/execute-mvn-goal.yaml
          parameters:
            goal: install

      - script: mvn package
        displayName: 'Run mvn package'

      - script: sudo apt-get -y install zip
        displayName: 'Install zip'

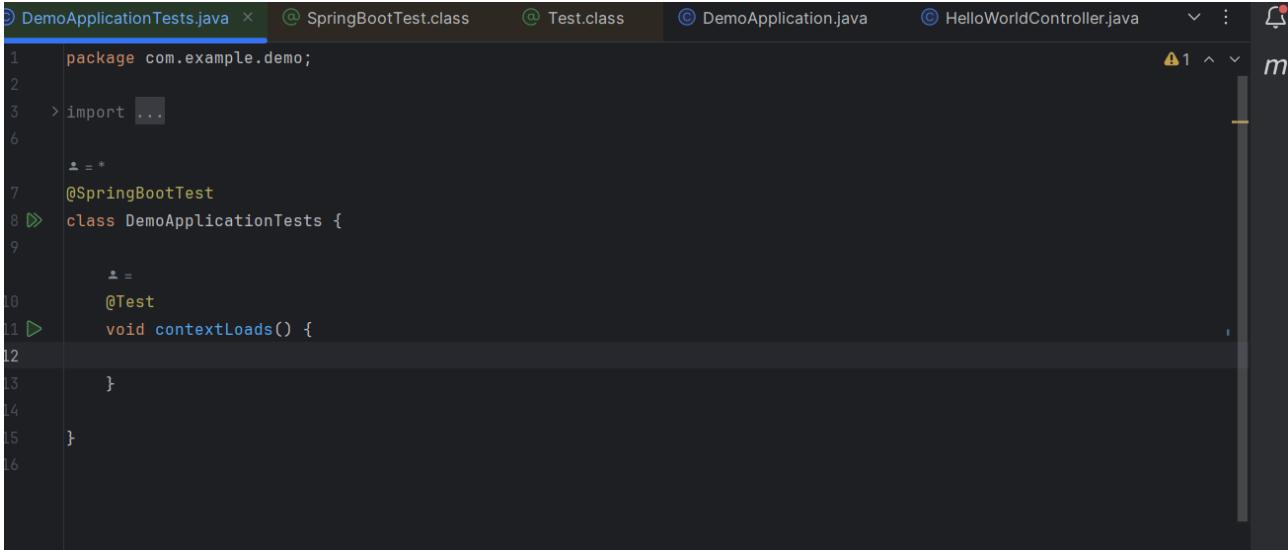
    settings:
```

```
    stages:
    - stage: Test
      displayName: Test
    jobs:
      - job: Test
        steps:
          - template: templates/execute-mvn-goal.yaml
            parameters:
              goal: test
      - task: PublishTestResults@2
        inputs:
          testResultsFormat: 'JUnit'
          testResultsFiles: '**/TEST-*.xml'
          testRunTitle: 'test-run'
          testRunSystem: 'AzureTestPlans'
        condition: always()
```

Added test Run System

3. Configure Release Pipeline:

- Modify the release pipeline to deploy the application only if all tests pass.
- Use conditions in the release pipeline to check for successful test results.



The screenshot shows a Java code editor with several tabs at the top: DemoApplicationTests.java (selected), SpringBootTest.class, Test.class, DemoApplication.java, and HelloWorldController.java. The code in DemoApplicationTests.java is as follows:

```
1 package com.example.demo;
2
3 > import ...
4
5     ...
6
7 @SpringBootTest
8 class DemoApplicationTests {
9
10     ...
11     @Test
12     void contextLoads() {
13
14     }
15 }
```

Changed back to default test

3. Configure Release Pipeline:

- Modify the release pipeline to deploy the application only if all tests pass.
- Use conditions in the release pipeline to check for successful test results.

The screenshot shows the Azure DevOps Release Pipeline configuration interface. At the top, there are buttons for 'Save', 'Create release', 'View releases', and more. Below this, a section titled 'Define gates to evaluate before the deployment' is shown. A 'Deployment gates' section contains a single entry named 'Check Test Results', which is enabled. The configuration for this gate includes:

- The delay before evaluation:** Set to 5 Minutes.
- Deployment gates:** A 'Check Test Results' task is selected.
- Invoke REST API:** Task version 1.*
 - Display name:** Check Test Results
 - Connection type:** Azure Resource Manager
 - Azure subscription:** Azure subscription 1 (9abae428-d8c3-44fe-bdf2-4e08593901)
 - Method:** GET
 - Headers:** A JSON object containing:

```
{"Content-Type": "application/json", "PlanUrl": "$(system.CollectionUri)", "ProjectId": "$(system.TeamProjectId)", "HubName": "$(system.HostType)", "PlanId": "$(system.PlanId)", "JobId": "$(system.JobId)", "TimelineId": "$(system.TimelineId)", "TaskInstanceId": "$(system.TaskInstanceId)", "AuthToken": "$(system.AccessToken)"} 
```
 - URL suffix and parameters:** https://dev.azure.com/AnatoliYakubышын/anatoliyyakubышын/_apis/test/runs?api-version=6.0
- Advanced:** Completion event is set to 'ApiResponse'. Success criteria is defined as 'eq(jsonPath("\$.value[-1].state'), 'Completed')'
- Evaluation options:** Deployment queue settings are defined to define behavior when multiple releases are queued for deployment.

Set trigger gate

4. Run the Pipeline:

- a. Trigger the pipeline and verify that it runs the tests and reports the results back to Azure Test Plans.

ubышин / anatoliyakubышин / Pipelines / MySampleApp-2 / 20250223

Jobs in run #20250223.8
MySampleApp-2

Test

Job: Test (33s)

- Initialize job (2s)
- Checkout MySampleApp-2@main... (2s)
- Install Java 11 (4s)
- Install Maven (2s)
- Generate settings.xml (2s)
- Execute mvn goal (8s)
- PublishTestResults (8s)
- Post-job: Checkout MySampleA... (<1s)
- Finalize Job (<1s)

Package

Publish_Dependencies

Package

- b. Ensure that any failed tests block the deployment process.

HelloWorldSpring (ID: 2)

Define Execute Chart

Test Points (2 items)

Title	Outcome	Order	Test Case Id
Validate Content of page by '/' path	Failed	1	3

Recent test runs

State	Run I...	Title	Completed Date	Build Number	Failed	Pass Rate
Needs investigat...	7	HelloWorldSpring (Manual)	23.02.2025 17:37:36		1	0%

URL suffix and parameters (i)

```
https://dev.azure.com/AnatoliiYakubyshyn/anatoliyyakubyshyn/_apis/test/runs  
?api-version=6.0
```

Advanced ^

Completion event * (i)

ApiResponse

Success criteria (i)

```
eq(jsonPath('$.value[-1].totalTests'), jsonpath('$.value[-1].passedTests'))
```

Output Variables ▾

Evaluation options ▾

Updated deployment gate

Save Create release View releases ...

Gates ▾

Define gates to evaluate before the deployment.

[Learn more](#)

The delay before evaluation (i)

5 Minutes

Deployment gates (i)

+ Add

Check Test Results Enabled

Evaluation options ▾

The time between re-evaluation of gates (i)

5 Minutes

Minimum duration for steady results after a successful gates evaluation (i)

0 Minutes

The timeout after which gates fail (i)

6 Minutes

Gates and approvals (i)

Before gates, ask for approvals

On successful gates, ask for approvals

Ignore gates outcome and ask for approvals

Updated eval options

Then -> create release

yakubshyn / Settings / Service connections

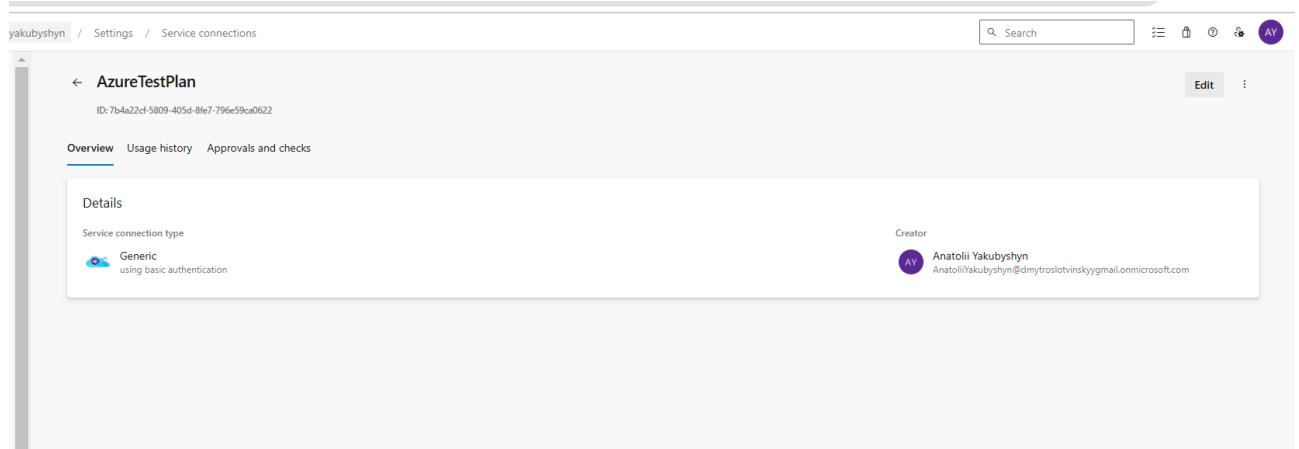
← AzureTestPlan
ID: 7baa22cf-5809-405d-8fc7-796e59ca0622

Overview Usage history Approvals and checks

Details

Service connection type
Generic using basic authentication

Creator
AY Anatolii Yakubshyn Anatolii.Yakubshyn@dmyslotvinskoy@gmail.onmicrosoft.com



added url in generic service connection

Deployment gates + Add ▾

Check Test Results Enabled

Invoke REST API ⓘ

Task version 1.*

Display name * Check Test Results

Connection type * ⓘ Generic

Generic service connection * ⓘ | Manage ↗ AzureTestPlan

Method * ⓘ GET

Headers ⓘ

```
{ "Content-Type": "application/json", "PlanUrl": "${system.CollectionUri}", "ProjectId": "${system.TeamProjectId}", "HubName": "${system.HostType}", "PlanId": "${system.PlanId}", "JobId": "${system.JobId}", "TimelineId": "${system.TimelineId}", "TaskInstanceId": "${system.TaskInstanceId}", "AuthToken": "${system.AccessToken}" }
```

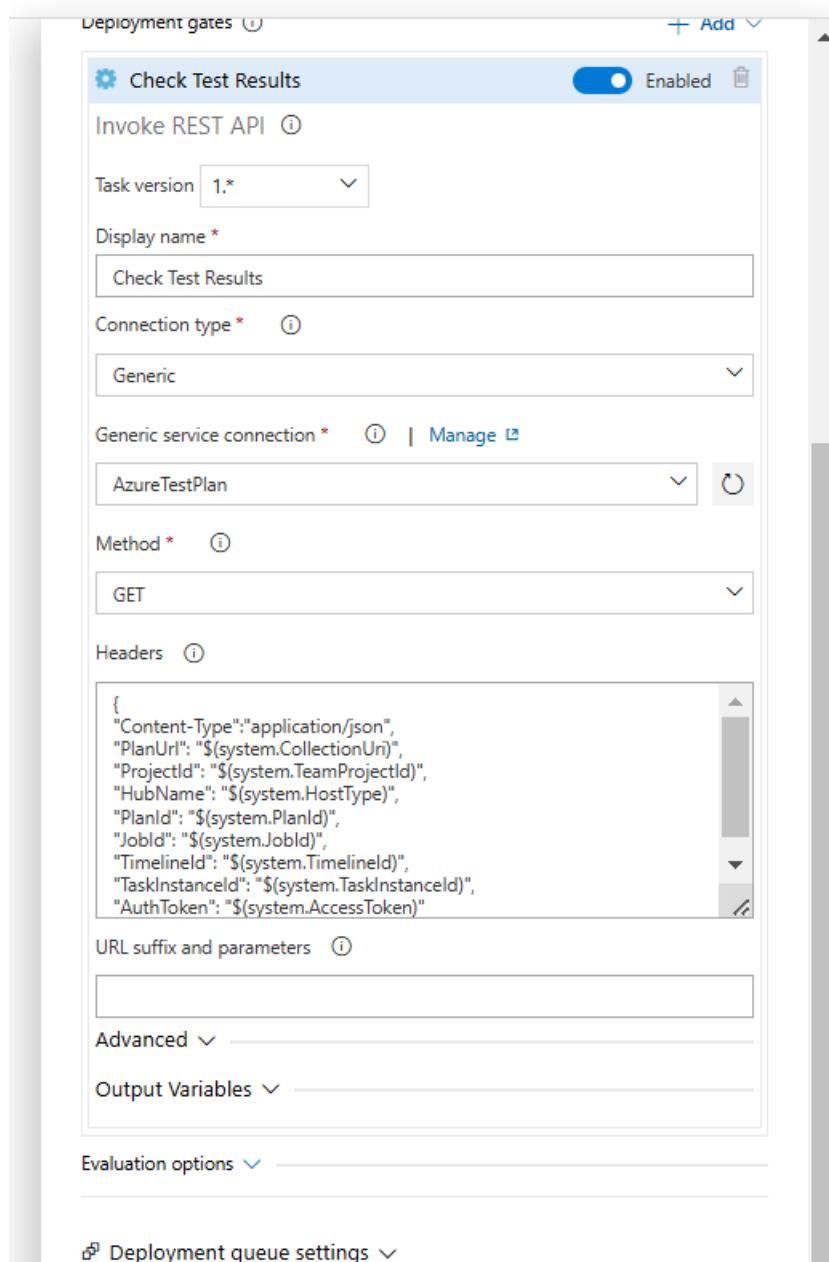
URL suffix and parameters

Advanced

Output Variables

Evaluation options

Deployment queue settings



Completion event *

ApiResponse

Success criteria ⓘ

```
eq(jsonPath('.value[-1].totalTests'), jsonpath('.value[-1].passedTests'))
```

Output Variables ▾

Evaluation options ▾

Check if all test passed in the last test run

/ anatoliyakubshyn / Pipelines / Releases / New release pipeline (1) / Release-12

New release pipeline (1) > Release-12 > Deployment ⓘ Not deployed

← Pipeline Tasks Variables Logs Tests Deploy Cancel Refresh Download all logs Edit ...

Deployment process Not deployed Started: 23.02.2025, 20:18:55
1m 7s

Pre-deployment gates Canceled

Canceled at 23.02.2025, 20:20 Deployment was canceled by Anatolii Yakubshyn

Deployment gates \ samples 20:18

 Check Test Results ⓘ Invoke a REST API as a part of your pipeline. 

Blocks the deployment process

Completion event *

ApiResponse

Success criteria ⓘ

```
eq(variables['value.state'][-1], 'Completed')
```

Output Variables ▾

Finally fixed success criteria

If really all tests pass

Recent test runs						
Test runs			Filter			
State	Run I...	Title	Completed Date	Build Number	Failed	Pass Rate
Needs investigation	9	HelloWorldSpring (Manual)	23.02.2025 18:57:08		1	0%
Completed	8	test-run	23.02.2025 18:23:01	20250223.9	0	100%
Needs investigation	7	HelloWorldSpring (Manual)	23.02.2025 17:37:36		1	0%
Completed	6	test-run	23.02.2025 17:35:53	20250223.8	0	100%
Completed	5	test-run	23.02.2025 16:54:55	20250223.7	0	100%
Completed	4	test-run	23.02.2025 16:48:42	20250223.6	1	0%

Check again