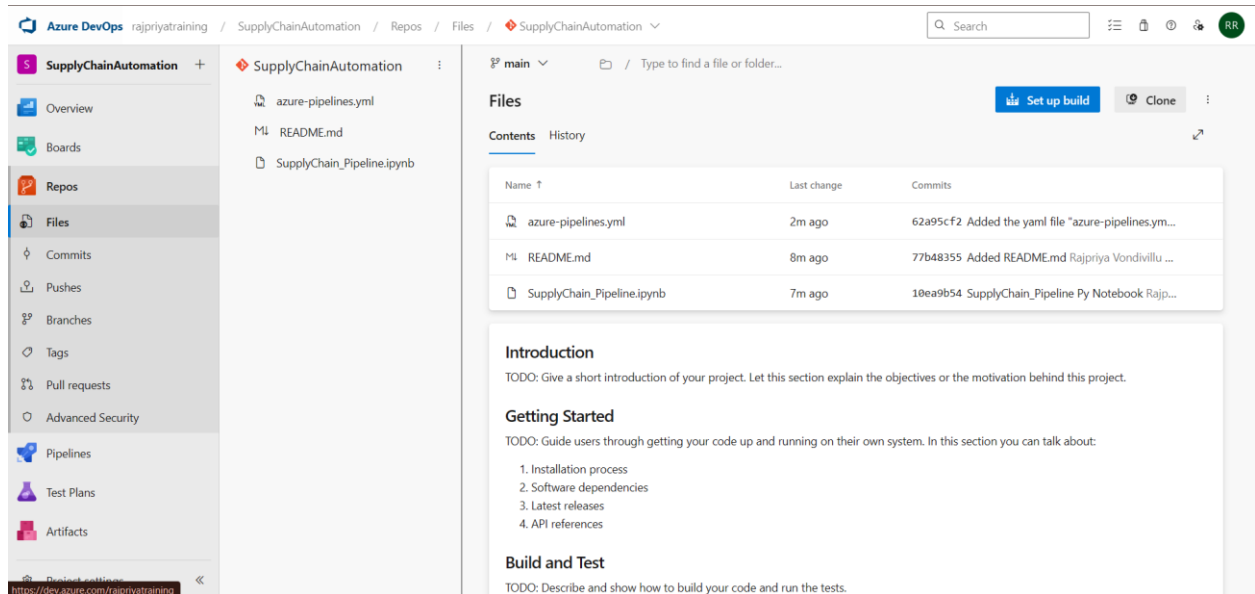


Pictorial Document for DevOps Pipeline Process – WEEK_5

Step 1:

Create New Project -> SupplyChainAutomation -> Click on to the Three dots on the top right corner and add your .ipynb notebook -> “SupplyChain_Pipeline.ipynb” and yaml file “azure-pipelines.yml” file.



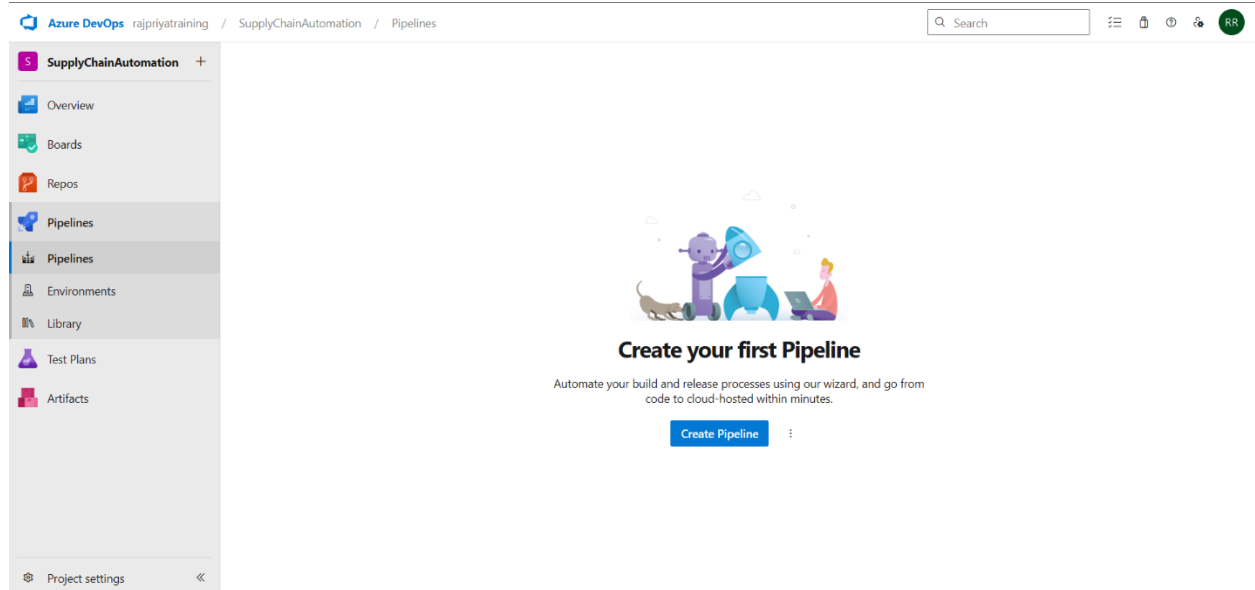
The screenshot shows the Azure DevOps interface for a project named 'SupplyChainAutomation'. The left sidebar contains a navigation menu with options like Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, Test Plans, and Artifacts. The 'Files' tab is selected, showing a list of files in the repository:

Name	Last change	Commits
azure-pipelines.yml	2m ago	62a95cf2 Added the yaml file "azure-pipelines.y...
README.md	8m ago	77b48355 Added README.md Rajpriya Vondivilu ...
SupplyChain_Pipeline.ipynb	7m ago	10ea9b54 SupplyChain_Pipeline Py Notebook Rajp...

Below the file list, there is a section titled 'Introduction' with a TODO: Give a short introduction of your project. Let this section explain the objectives or the motivation behind this project. Another section titled 'Getting Started' has a 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, 3. Latest releases, 4. API references. A third section titled 'Build and Test' has a TODO: Describe and show how to build your code and run the tests.

Step 2:

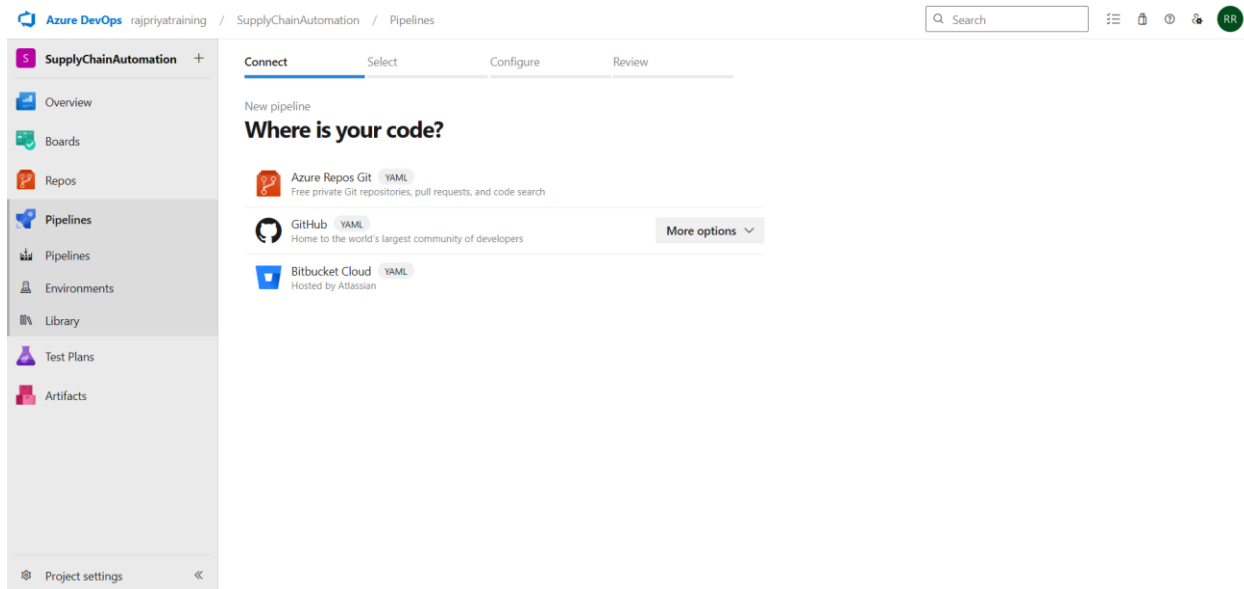
Go to Pipelines -> Pipelines -> Create Pipeline.



The screenshot shows the Azure DevOps interface for the 'Pipelines' section. The left sidebar contains a navigation menu with options like Overview, Boards, Repos, Pipelines, Environments, Library, Test Plans, and Artifacts. The 'Pipelines' tab is selected, showing a 'Create your first Pipeline' wizard. The wizard includes an illustration of a person working on a laptop, a dog, and a robot. The text reads: 'Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.' Below the text is a 'Create Pipeline' button.

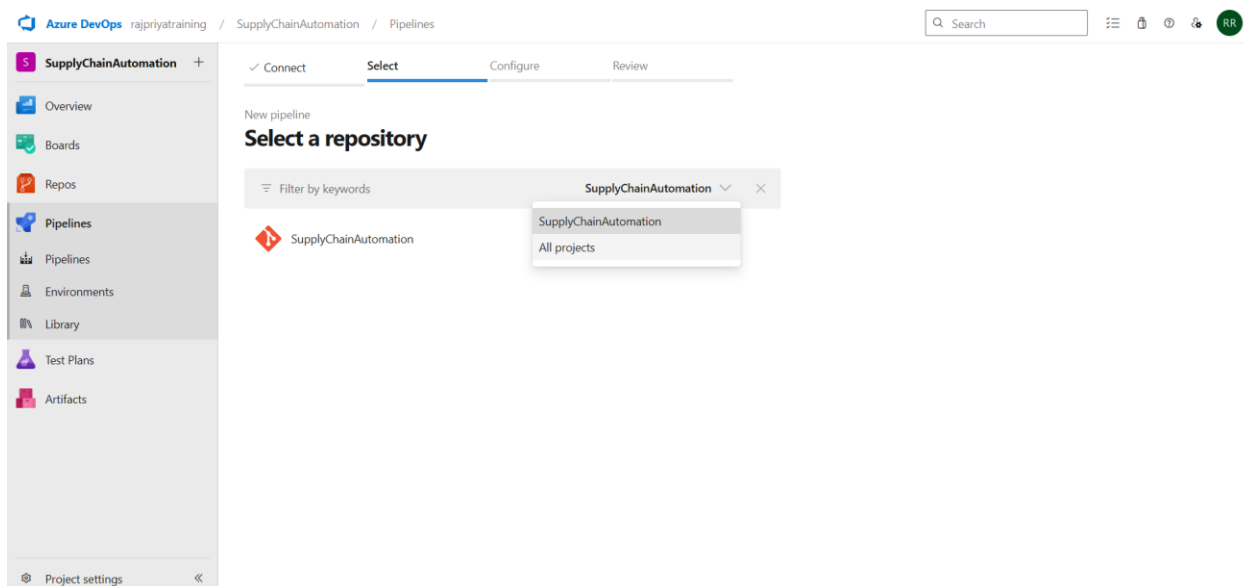
Step 3:

Select Azure Repos Git.



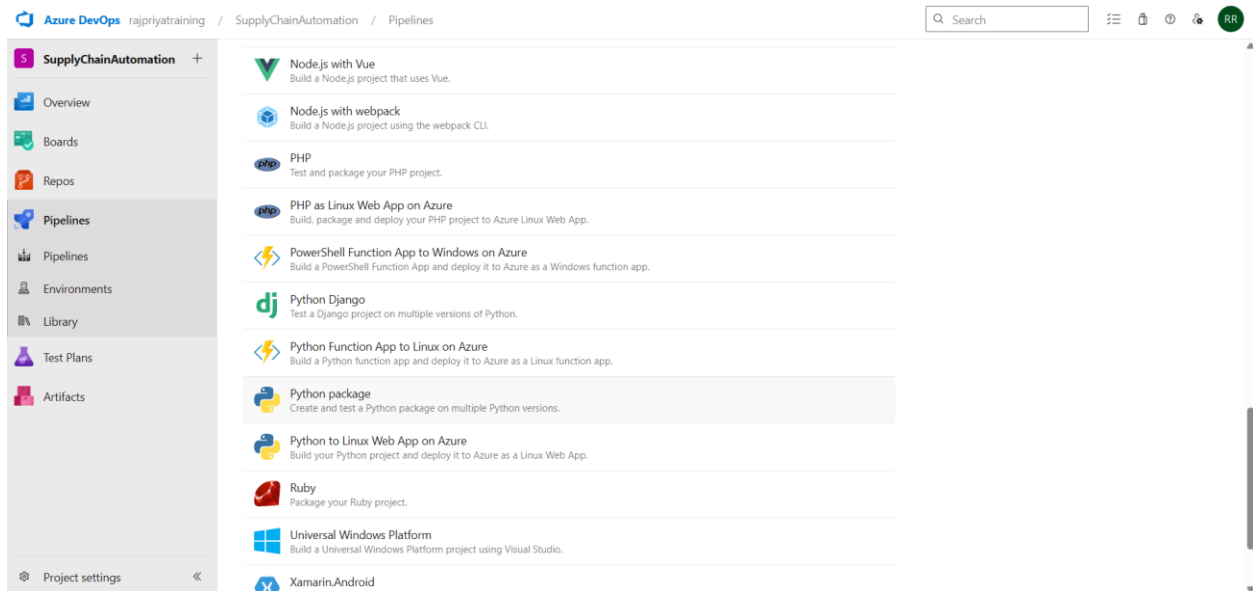
Step 4:

Select a repository -> SupplyChainAutomation.



Step 5:

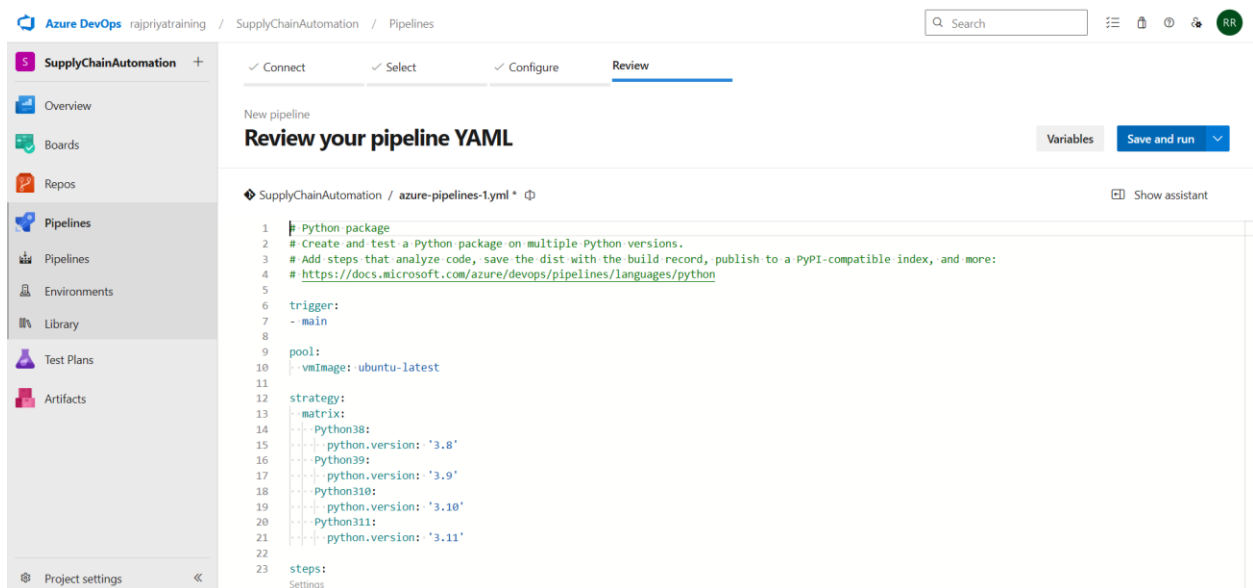
Select Python Package.



The screenshot shows the Azure DevOps interface for the 'SupplyChainAutomation' project. The left sidebar contains navigation links: Overview, Boards, Repos, Pipelines (selected), Environments, Library, Test Plans, and Artifacts. The main area displays a list of pipeline templates. The 'Python package' template is highlighted, with a description: 'Create and test a Python package on multiple Python versions.' Other templates include Node.js with Vue, Node.js with webpack, PHP, PHP as Linux Web App on Azure, PowerShell Function App to Windows on Azure, Python Django, Python Function App to Linux on Azure, Python to Linux Web App on Azure, Ruby, Universal Windows Platform, and Xamarin.Android.

Step 6:

Click on Save and Run.



The screenshot shows the 'Review your pipeline YAML' screen in Azure DevOps. The left sidebar is the same as in Step 5. The top navigation bar shows the project path: 'SupplyChainAutomation / Pipelines'. The main area has tabs for 'Connect', 'Select', 'Configure', and 'Review' (selected). Below the tabs, the title 'Review your pipeline YAML' is displayed. The pipeline configuration for 'azure-pipelines-1.yml' is shown, including the trigger, pool, strategy, and steps sections. The 'Steps' section is currently empty, with a 'Settings' link below it.

```
1 Python package
2 # Create and test a Python package on multiple Python versions.
3 # Add steps that analyze code, save the dist with the build record, publish to a PyPI-compatible index, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/python
5
6 trigger:
7   - main
8
9 pool:
10  vmImage: ubuntu-latest
11
12 strategy:
13   matrix:
14     Python38:
15       python.version: '3.8'
16     Python39:
17       python.version: '3.9'
18     Python310:
19       python.version: '3.10'
20     Python311:
21       python.version: '3.11'
22
23 steps:
24   Settings
```

Step 7:

Again, click on Save and Run.

The screenshot shows the 'Review your pipeline YAML' dialog in Azure DevOps. The 'Review' tab is active, displaying the pipeline YAML code. The 'Save and run' modal is open on the right, showing the commit message 'Set up CI with Azure Pipelines' and the option to 'Commit directly to the main branch' selected. The 'Save and run' button is visible at the bottom right of the modal.

```
1 # Python package
2 # Create and test a Python package on multiple Python versions.
3 # Add steps that analyze code, save the dist with the build record, publish to a PyPI-comp
4 # https://docs.microsoft.com/azure/devops/pipelines/languages/python
5
6 trigger:
7   - main
8
9 pool:
10  vmImage: ubuntu-latest
11
12 strategy:
13   matrix:
14     Python38:
15       python.version: '3.8'
16     Python39:
17       python.version: '3.9'
18     Python310:
19       python.version: '3.10'
20     Python311:
21       python.version: '3.11'
22
23 steps:
24   Settings
```

Step 8:

Now we can see the summary of the pipeline and it is scheduled to run with the configured agent.

The screenshot shows the pipeline summary page for the pipeline named '#20250822.2 • Set up CI with Azure Pipelines'. The 'Summary' tab is active, showing the pipeline was manually run by Rajpriya Vondivillu Raja. The pipeline is currently 'Queued'.

Name	Status	Duration
Job	Queued	