**What is DevOps and why it is needed**

The software development cycle gets started from planning. Once the planning is ready programmers start to work on the code and then QAs test the code. If the test is passed, the code will be deployed on the server. And there are operations to be performed and monitoring of the server (does the server need to be scaled up or scaled down or do you need to increase more capacity for the server?).

In a big organization, these tasks are managed by different teams. The business team or the Product Owner does the planning; the development team code, build and test and the deployment team deals with the Deployment, Operations and Monitoring. With Agile is brought into the picture, the business team and the development team start to work together. However, the Operations team still works separately and often times there is a gap between Development and Operations. Hence DevOps comes into the picture. As the name applies, DevOps is a combination of Development and Operations. It is a practice of collaboration between development and operations so they can work together to smooth out any conflicts between these two teams so they can focus on better quality of the product and fast delivery.

**What is Azure DevOps?**

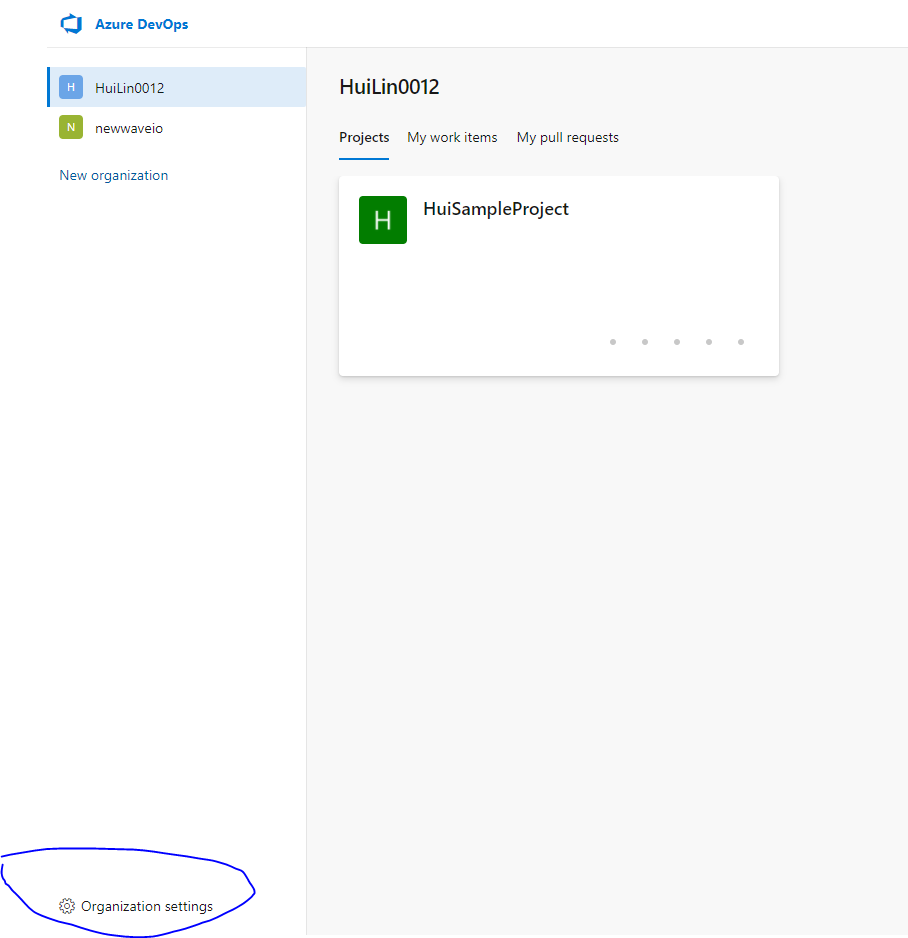
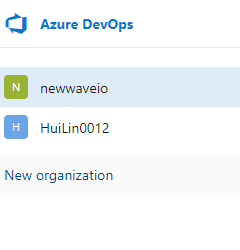
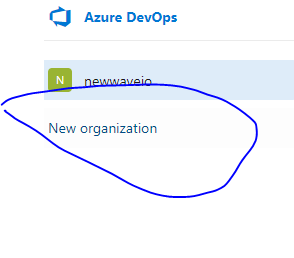
Azure DevOps is a website that provides a set of services to integrate project management, source control and deployment. There are five components in Azure DevOps: Boards, Repos, Pipelines, Test Plans and Artifacts.

* Boards. Boards is where project management takes place. Scrum and Kanban boards are integrated into Azure Boards for you to create and track tasks and test cases for each sprint
* Repos. Repos has a set of tools for version control management
* Pipelines. Azure Pipelines combines continuous integration (CI) and continuous delivery (CD) to constantly and consistently test and build your code and release it to any target.
* Test Plans. Test Plans provides a combined option of planned manual testing, acceptance testing, exploratory testing, and gathering feedback from stakeholders.
* Artifacts. Azure Artifact is a collection of dependency of projects and jar files. They are the output of the pipeline run and made available as the source for deployment/release on any server.
* Azure DevOps Server. Azure DevOps provides two types of server configuration: Cloud (Services) and On-Premises (Server).

**Azure DevOps Overview Walkthrough**

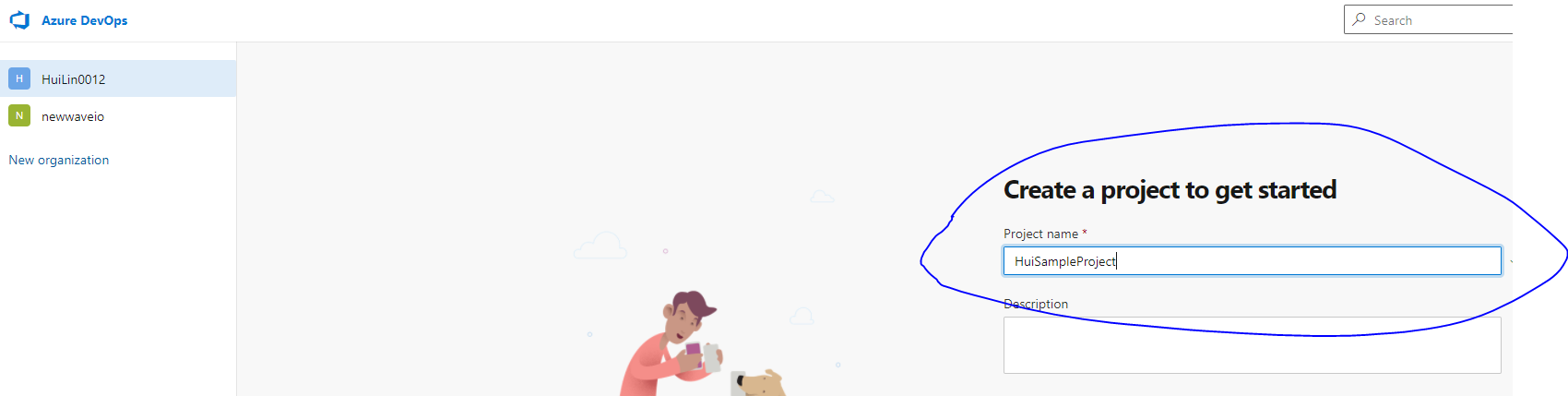
1. How to create an organization.

After you sign up and log in to your Azure DevOps portal, the first thing needs doing is to create an organization. In the screenshot below I created a new organization named HuiLin0012. You are also able to change the settings for an organization.



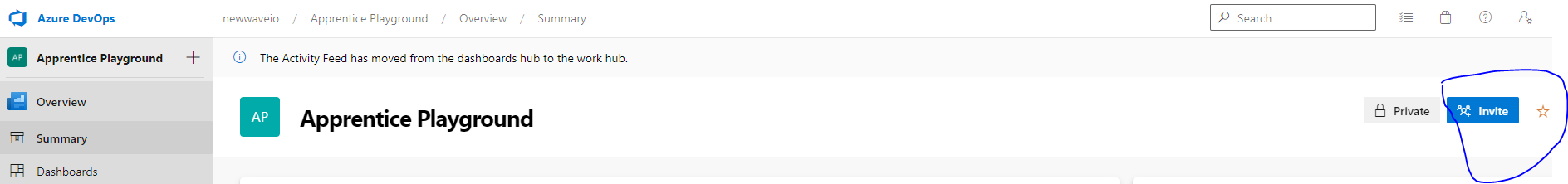
1. How to create a project

After signing into an organization, you will be asked to create a project and you can also set up the settings for the project.



1. How to add team members

In the summary page of the project, click on the Invite button to add your team members to the project. Alternatively, members can also be added from Project Setting > General >Teams > Add menu.

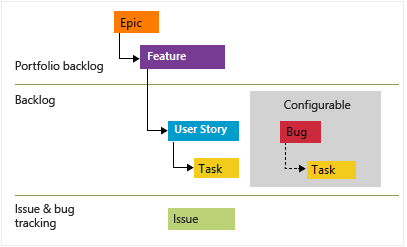


**Azure Boards Walkthrough**

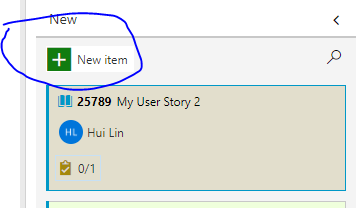
A work process determines the work item types and workflow available in Azure Boards. Azure DevOps provides four types of work processes: Basic, Agile, Scrum and CMMI. This Walkthrough uses the work process of Agile.

1. Work items

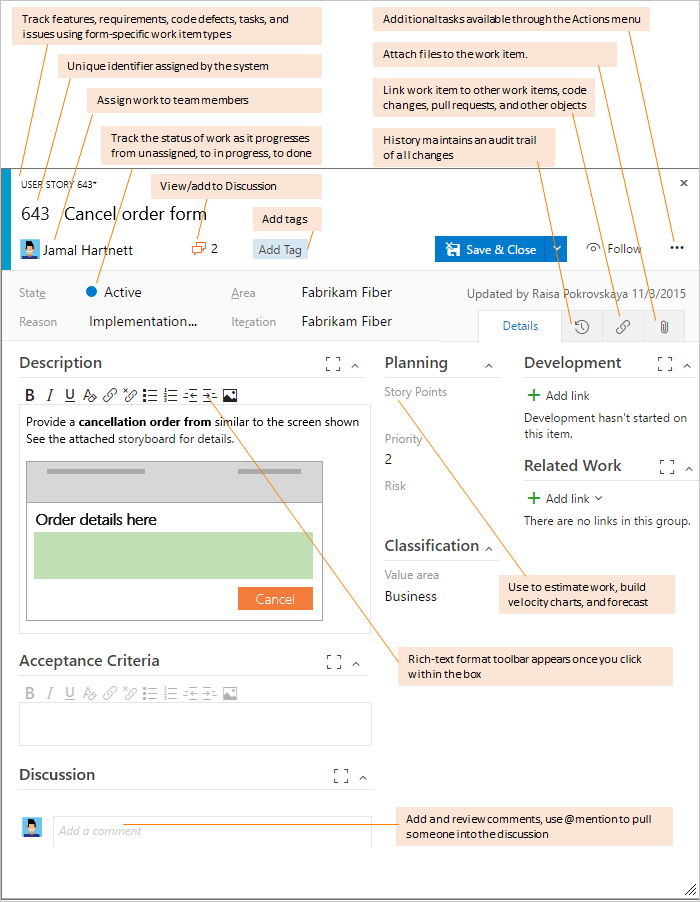
* A work item is a unit of work. It includes Bug, Epic, Feature, User Story, Issue, Task and Test Case.
* Work item hierarchy in Agile: Epic > Feature > User Story > Task. On top of everything is Epic, which is a business initiative to be accomplished. For example, to create an e-Commerce website is an Epic. Under one Epic we can get multiple features. A feature typically represents a shippable component of software. In the e-Commerce Website example, a feature can be creating a shopping cart or a catalog. Inside one feature we can create user stories, which are the implementation of each feature. And for each user story we can create multiple tasks, which is the smallest unit of work you can track either in hours or days.
* For project based on the Agile process, it is commended using User Stories, Bugs, and Features.



* To create a work item from the Boards homepage, click the green plus sign.

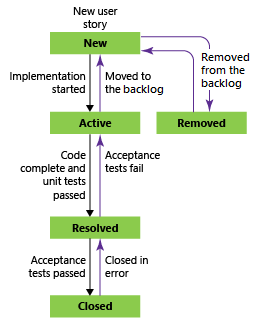


* Options available in a work item: Assign team members, Update State, Add discussion, a tag or description for the work item, Set up Story Point (the estimate of the amount of work required to complete a work item), Create Priority (with 1 highest and 4 lowest). You can also attach a file, a link or view the change history of a work item. If you click on Follow, you will receive all updates to a work item in your email.
* As work progresses from not started to completed, you update the State field of a work item from New, Active, Resolved, and Closed.



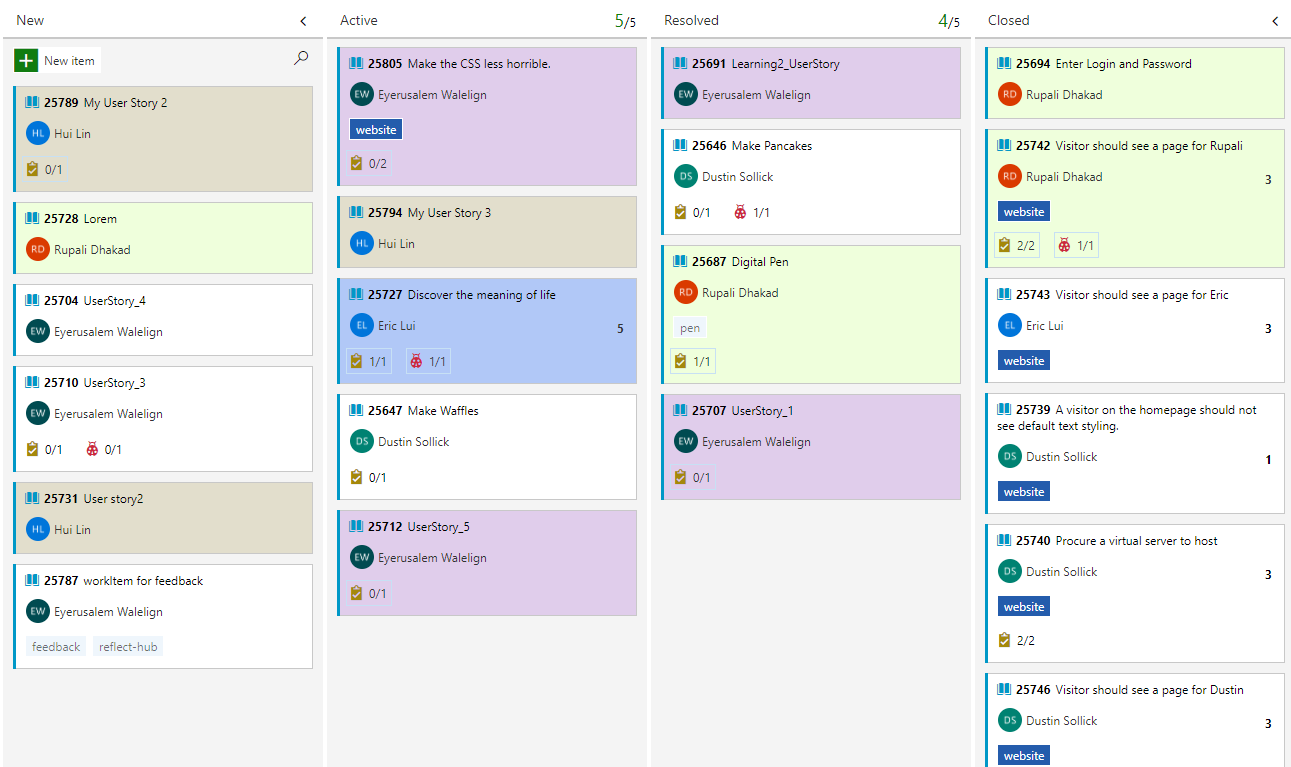
* Workflow of a work item

1. When a new user story is created, it can be either removed from backlog or starts its implementation.
2. Once it starts its implementation, the state is updated to active from new. From there it can be moved to the backlog to get coded and started testing.
3. Once the test case is passed, the state changes to resolved and the story moves on to the Acceptance tests, which is the system testing to evaluate the system's compliance with the business requirements and evaluate whether it is acceptable for delivery.
4. If the Acceptance Tests fail, the state changes back to Active and developers work on the issues. If the Acceptance tests pass, the story will be closed. If in the Closed state, errors are found the story can be moved back to Resolved state.

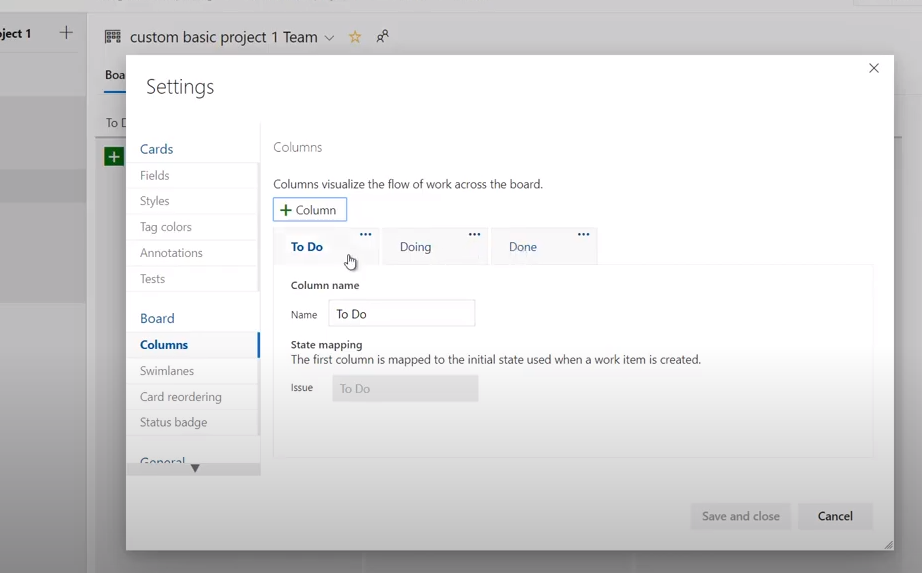


1. Boards

* Azure Boards is designed for tracking needs. It is a Kanban board that provides a clear picture of work done and doing by team members so you can track the progress of each team member or the progress of the sprint or the progress of product development.
* It has four columns each of that has multiple work items. It has multiple options to customize the settings to better satisfy your tracking needs.

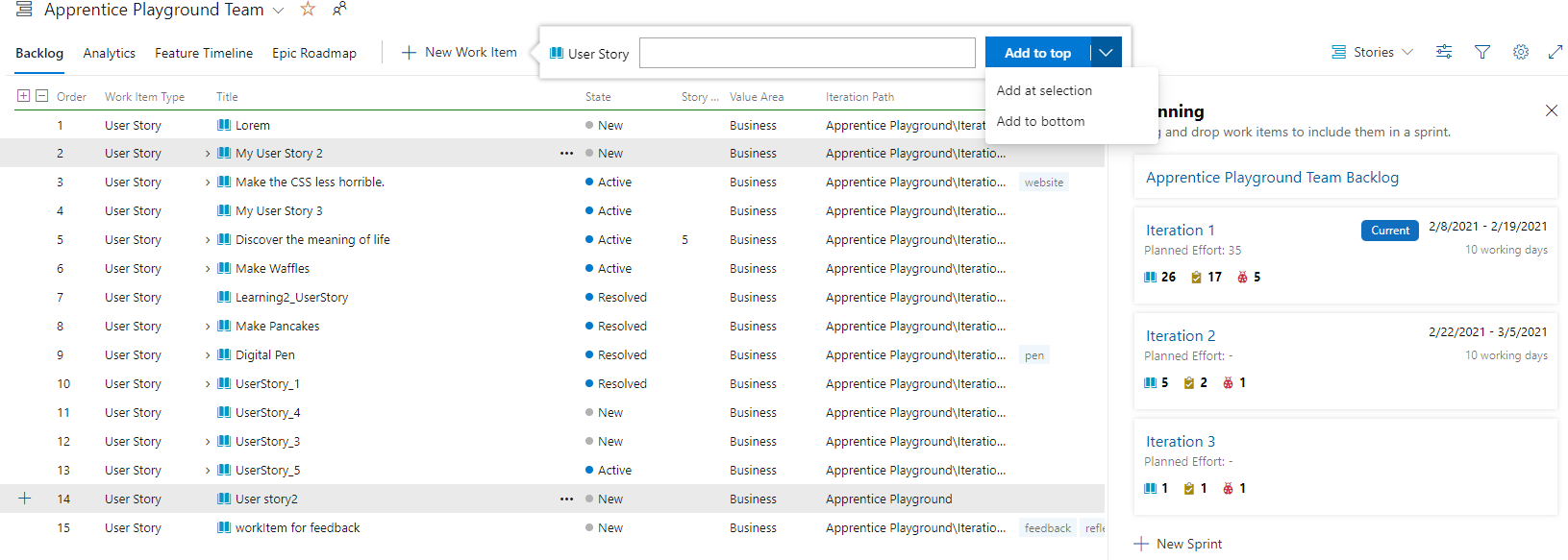


* By drag and drop to a new column, the state of each work item is changed automatically.
* The Settings option allows you to change the setting for Work Items, Boards layout and General.

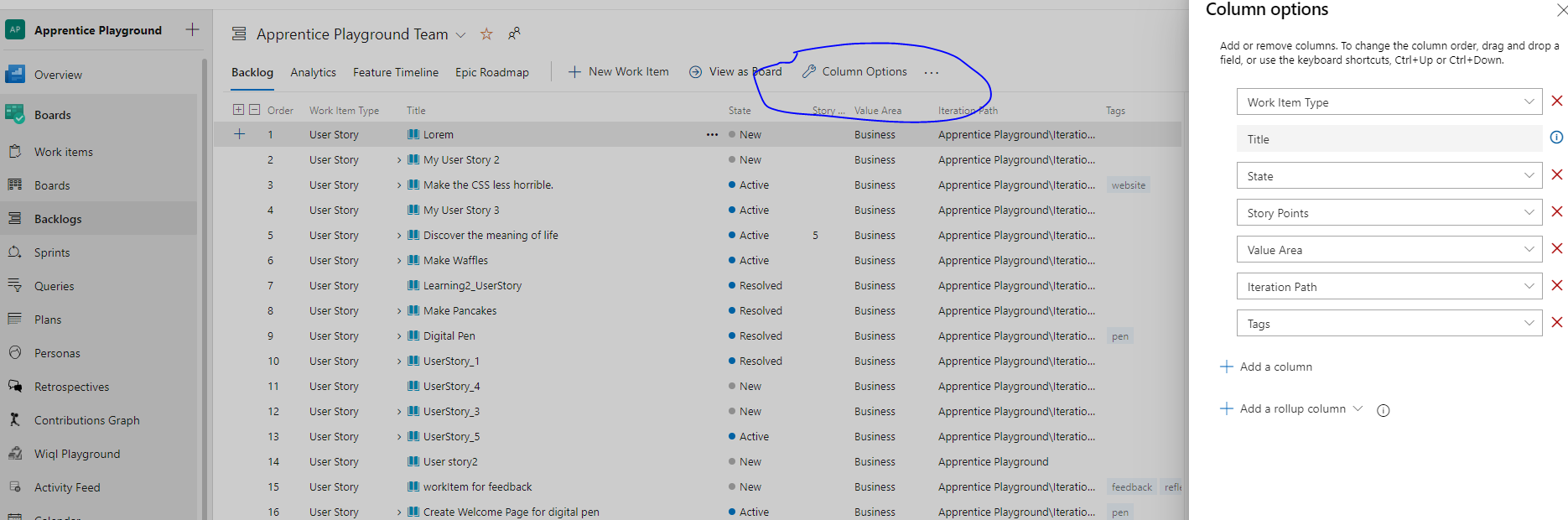


1. Backlog

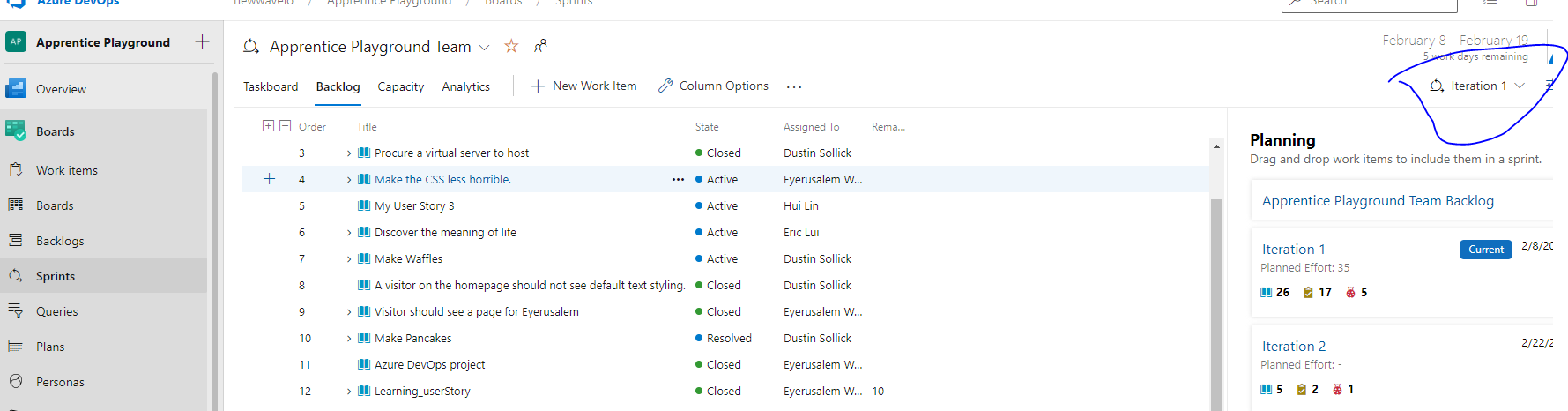
* Backlogs are automatically created when you create a project or add a team.
* Backlog page has two sections. On the left, it displays a list of selected work items in the order of priority, where you are also able to create new work items, updating and maintaining existing work items. For example, you are able to add multiple tasks or bugs to a specific work item, set up the priority for a new work item or change an existing work item’s priority by drag and drop.

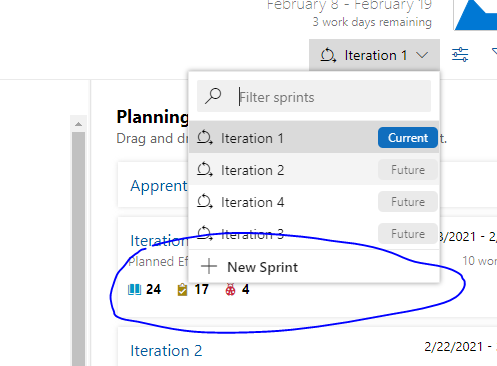


* You can select Column Options to change column setting. You can add, delete and change the order of a column. You can also sort the data in a column.



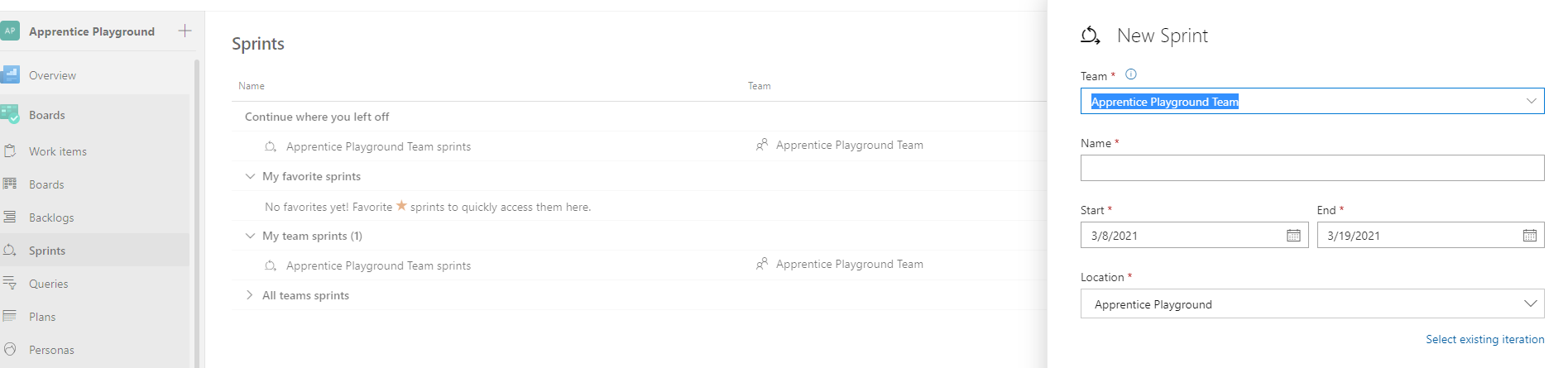
* On the right, there is the planning section. If you click on the current iteration, i.e. Iteration 1, you will be brought to Sprint page and the backlog of Iteration 1 is available for you to view. You are able to update the work items by drag and drop the work item to a particular iteration. There is also an option to add a new sprint.





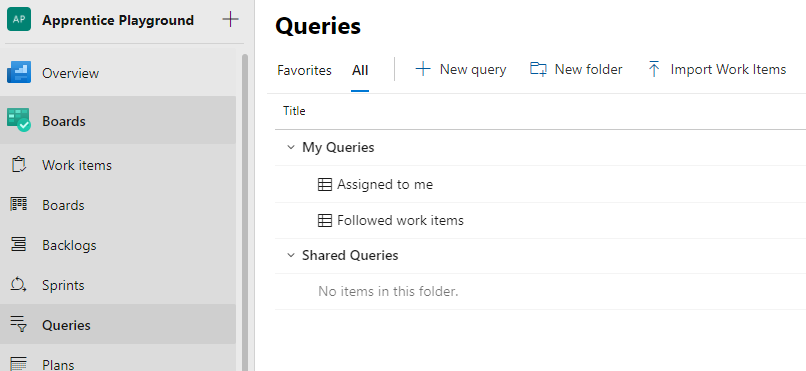
1. Sprints

* Sprints has a filtered backlog and task board based on each sprint. These tools are useful for implementing Scrum practices. You can schedule and plan sprints, update your task board, and monitor your sprint flow in Sprints.



1. Queries

* A query is used to filter work items
* My Queries are either assigned to the user or followed by the user. Shared Queries are Queries created by team members.
* Available to add a query to Favorites, edit a query in All and send a query in email



1. Plans

* Delivery Plans show the scheduled work items by sprint of selected teams against a calendar view

1. Personas

* Personas provides multiple types of roles in a team that will be affected by the work items.

1. Retrospectives

* Retrospectives is where you keep and share feedback among your team. It could be toward a project or a work item.

1. Contributions Graph

* Contributions Graph shows your activities toward each repository.

1. Wiql Playground

* It lets you run a query using Work Item Query Language

1. Activity Feed

* The Activity Feed gives a summary of all recent changes of work items, commits, pull requests and builds

1. Calendar

* Calendar displays the state date and end date of each sprint. You can also add items to the calendar.

1. Product Vision

* It is the vision shared by the product team.

1. Estimate

* Team can vote to decide how much effort is needed for a work item done.

1. Tags

* Tags is used to filter a work item, an affected role or for a query. It is also the place where you manage all your team project work item tags

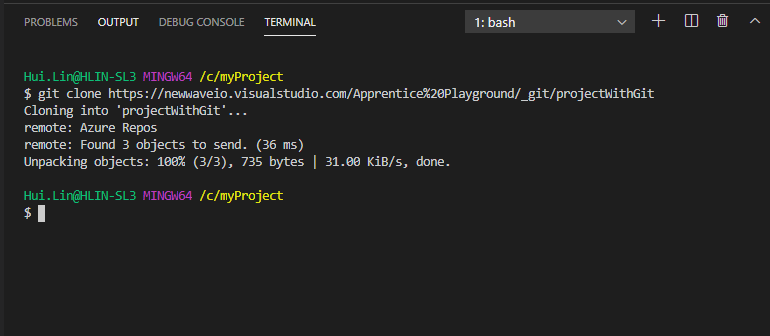
**Azure Repos Walkthrough**

1. Repos

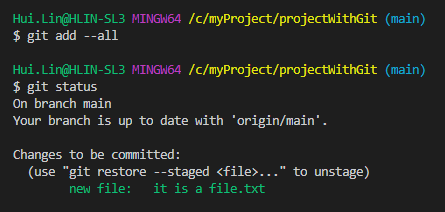
* Repos is a version control tool used to manage code.
* Repos is used to store the code of a project, to track and manage each and every change in code made by the team.
* Two types of version control in Azure DevOps: Git and TFVC (Team Foundation Version Control). TFVC is a centralized version control system while Git is a distributed version control system. The difference lies in the fact that in TFVC team members directly connect to the server. In Git, team members create a local repository, which is an entire replica to the server repository.

1. How to Clone Azure DevOps repository to your local repository and synchronize the files between two repositories

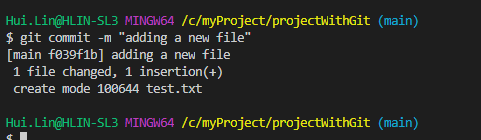
* I create a new repository called projectWithGit in Azure DevOps. Then I clone the Repository to my local machine with the url and the command git clone <url> to my Visio studio terminal.

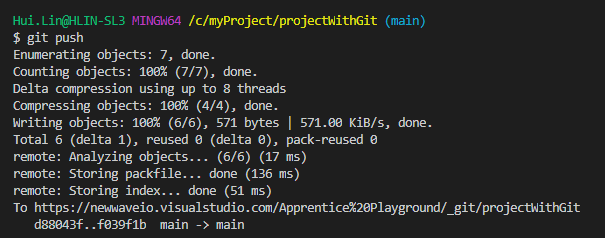


* Add files to my local repository



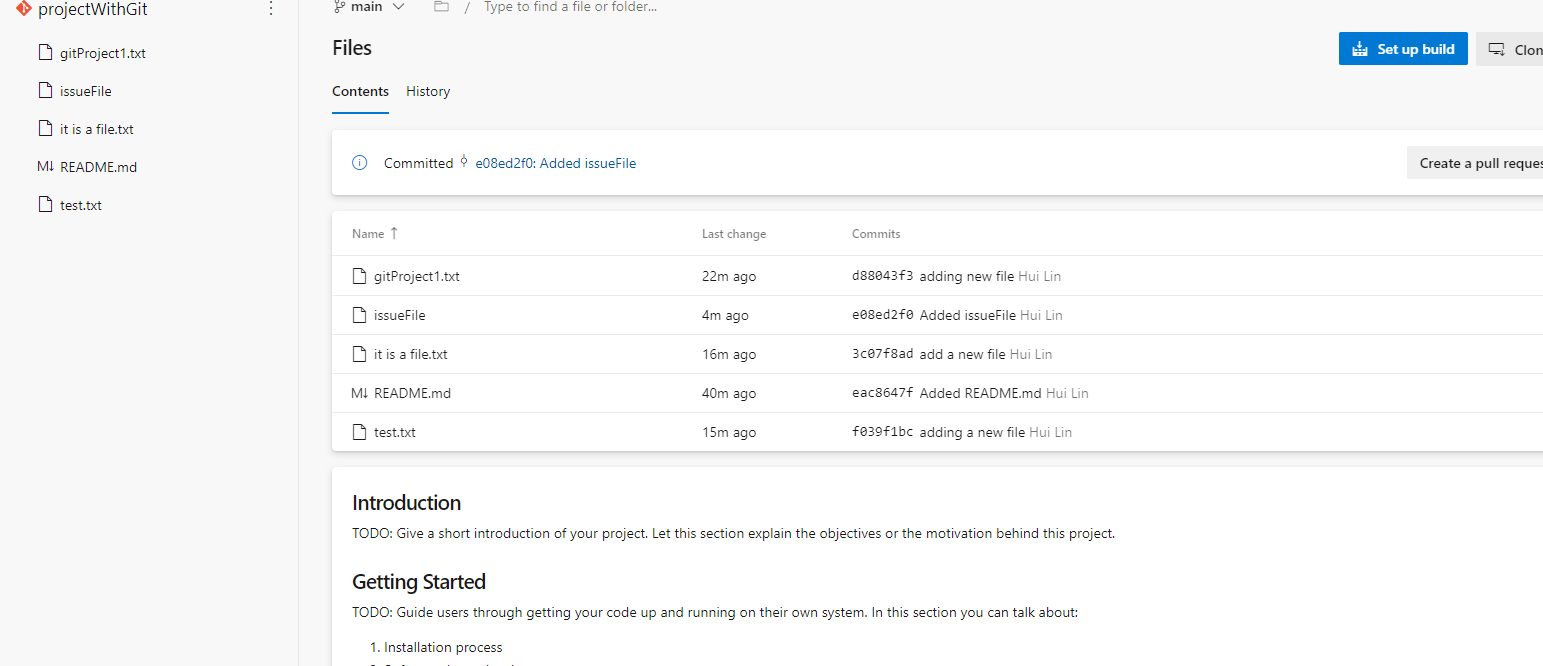
* Commit and push the file to Repos in Azure DevOps

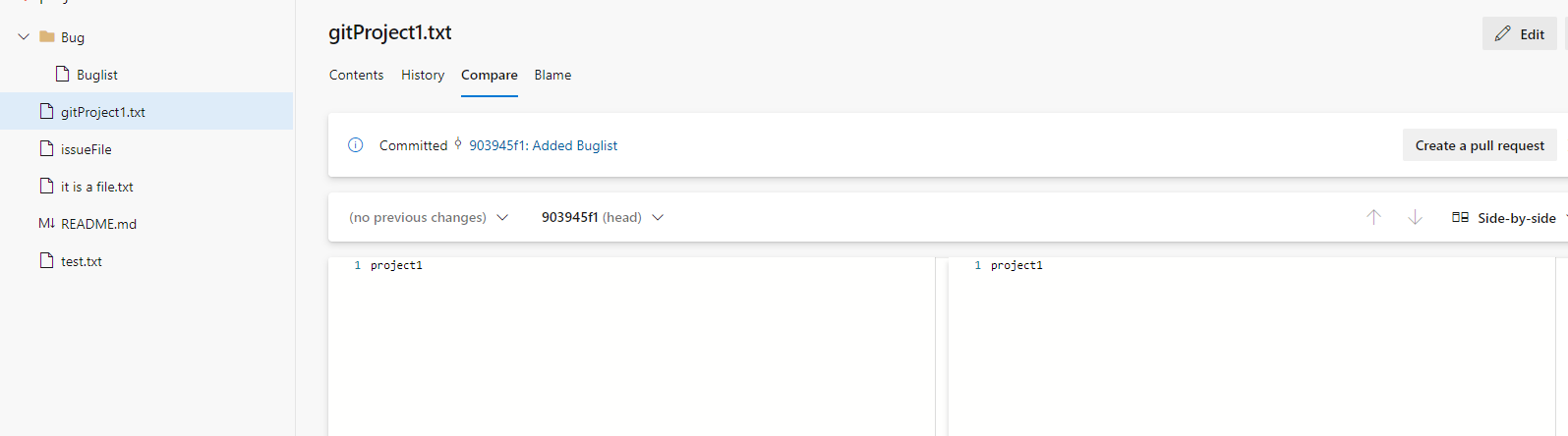




1. Files

* In the Repos Files page, you are able to view the newly added files.
* You can add a folder (must add a new file in the new folder) and by clicking on a file view its contents. You are also able to view a file’s change history, compare two versions of the file and find out who made the last change in Blame. You can also work directly on Azure DevOps and edit a file and commit it to the Repo.

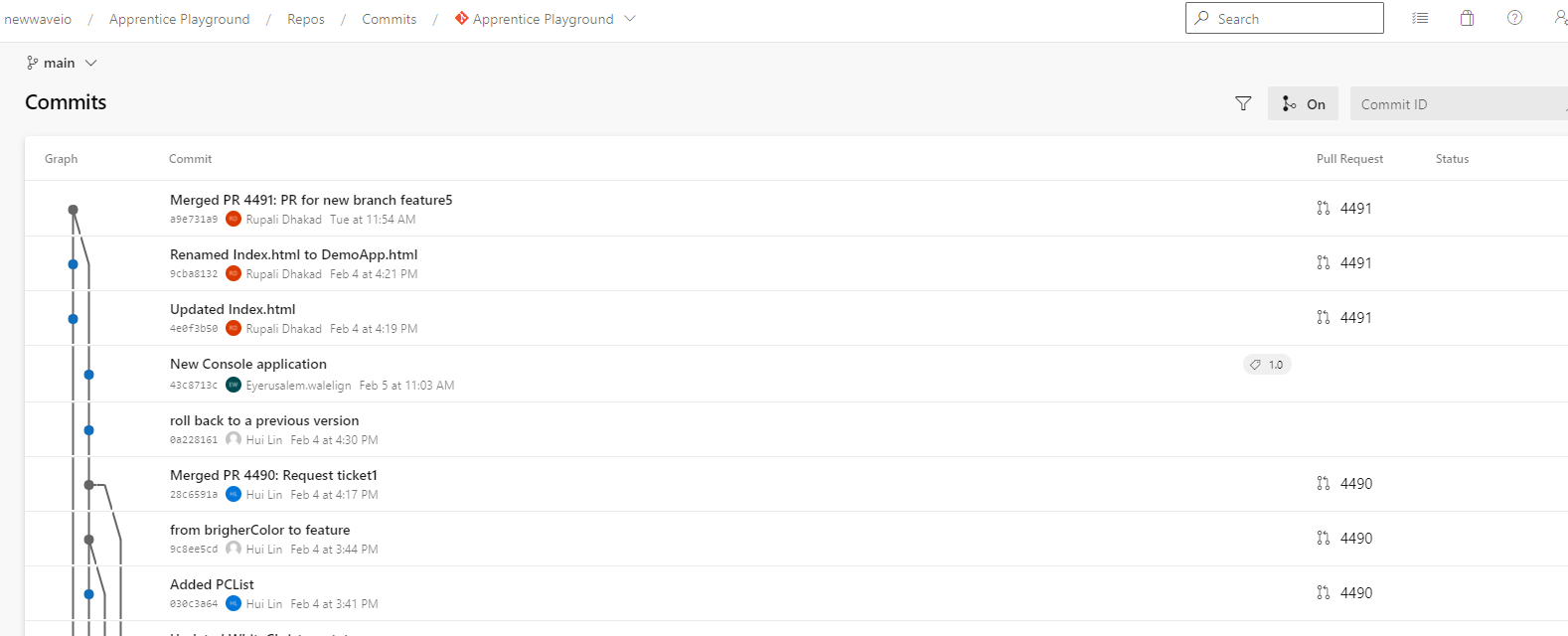




* You are able to upload a file to the folder in the server repo or download a folder or the whole repo with a Zip file.

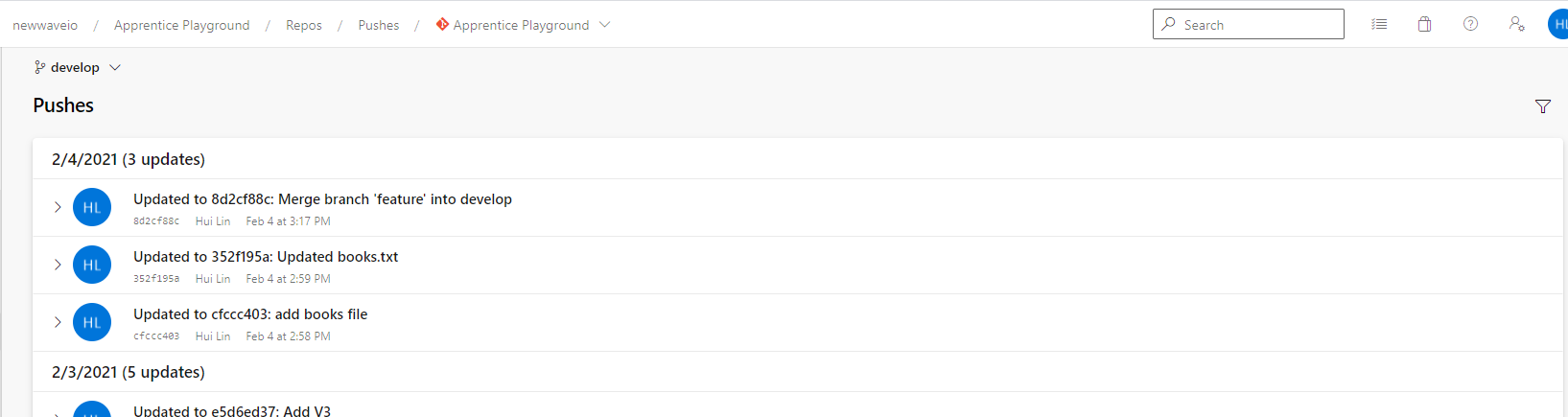
1. Commits

* Commits has a commits graph and lists all commits made by the team in a Repo. You can filter the commits list with a team member’s name, commit start date and end date. You can also hide the graph from the view.

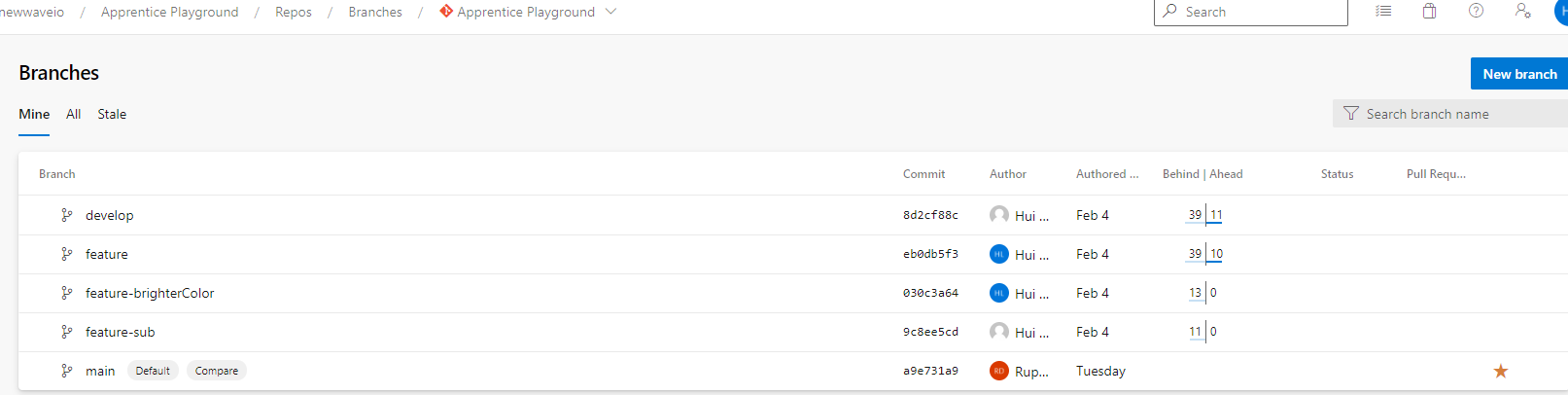


1. Pushes

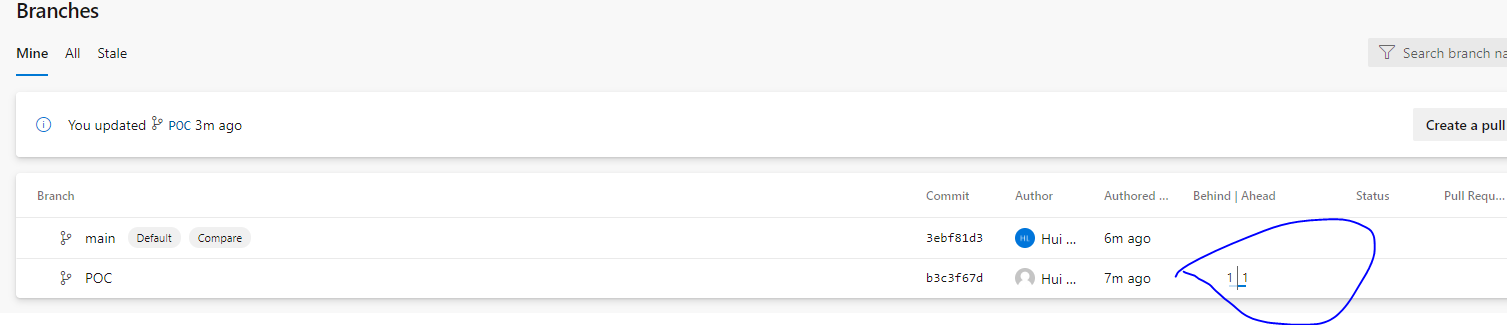
* Pushes displays all pushes done to a particular branch in a particular Repo. There is a filter feature available to filter with a team member’s name, a push’s start date and end date.



1. Branches

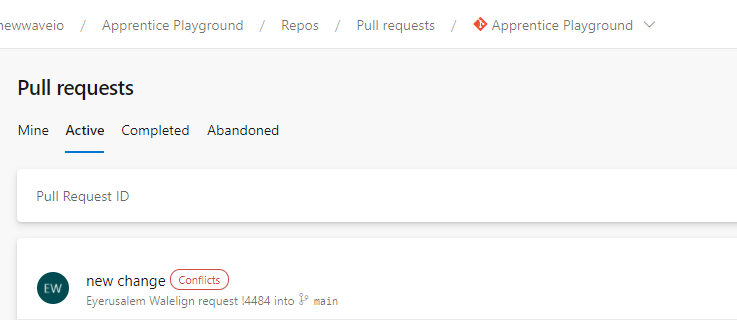
* Branches has a snapshot of each of your branches and the team’s branches. The page shows if the branches are synchronized between your local repository and the server repository.

Suppose I made a commit in my local POC branch and pushed it to the server repository. And I made a commit in my remote main branch. Now my POC branch is both behind and ahead of its parent main branch in the server repository.



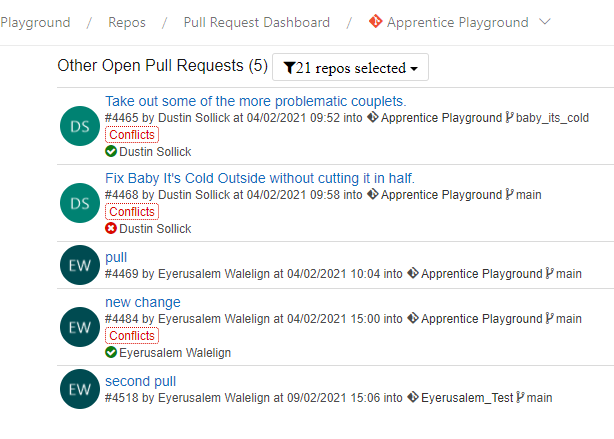
1. Pull Requests

* Pull Requests shows all pull requests made for a particular Repo.



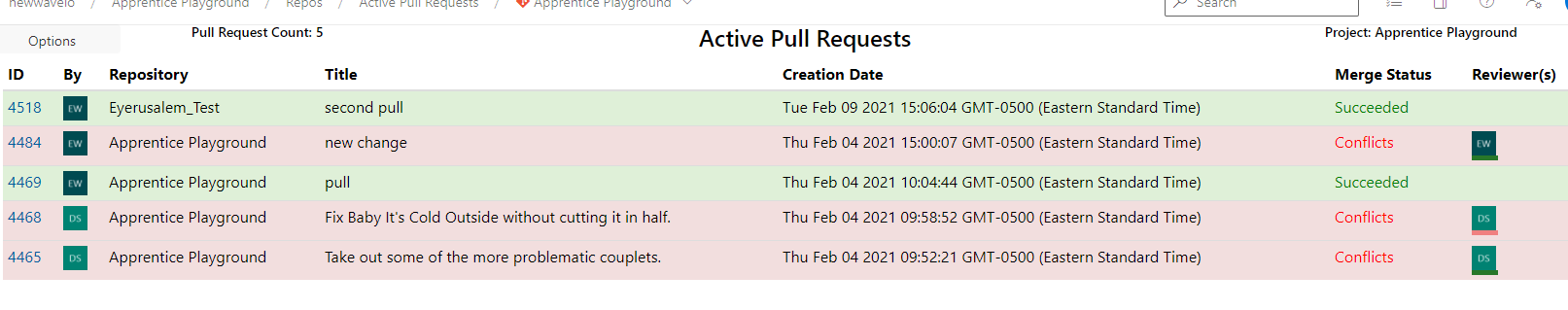
1. Pull Request Dashboard

* This page shows all open pull requests for all repos or a selected repo.



1. Active Pull Request

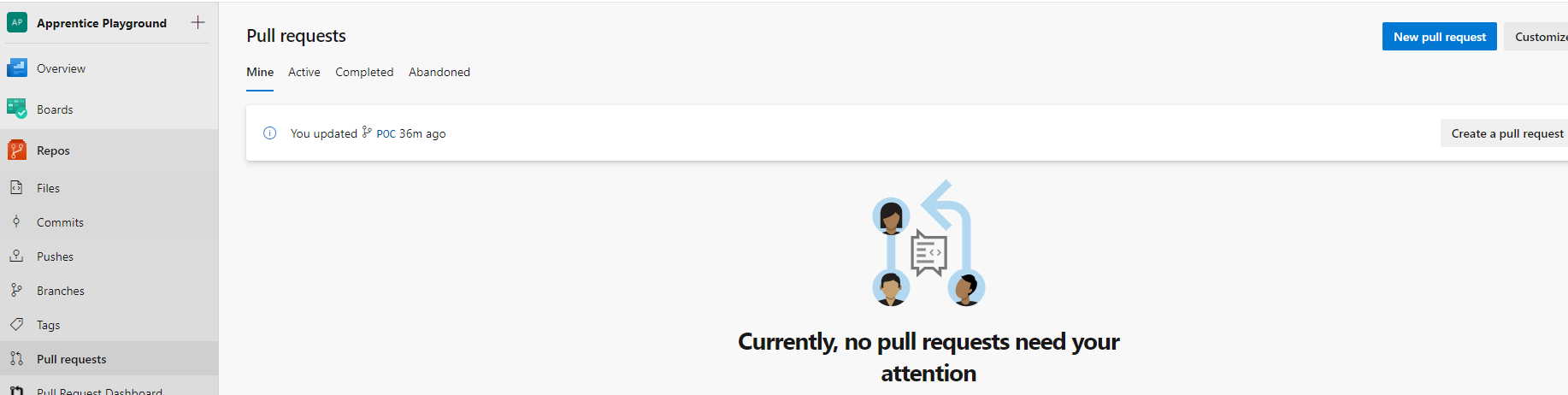
* Active Pull Request displays all pull requests that are waiting for approval.



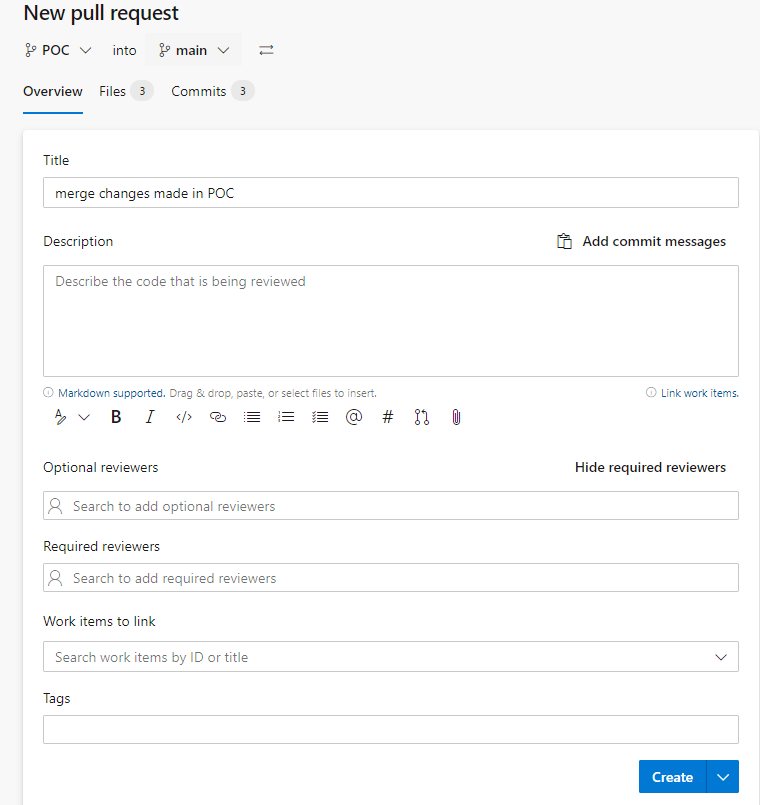
1. How to create pull request in Azure DevOps

* A pull request is used to merge a branch to another branch with extra features.
* Add reviewer to see if there is anything to be improved

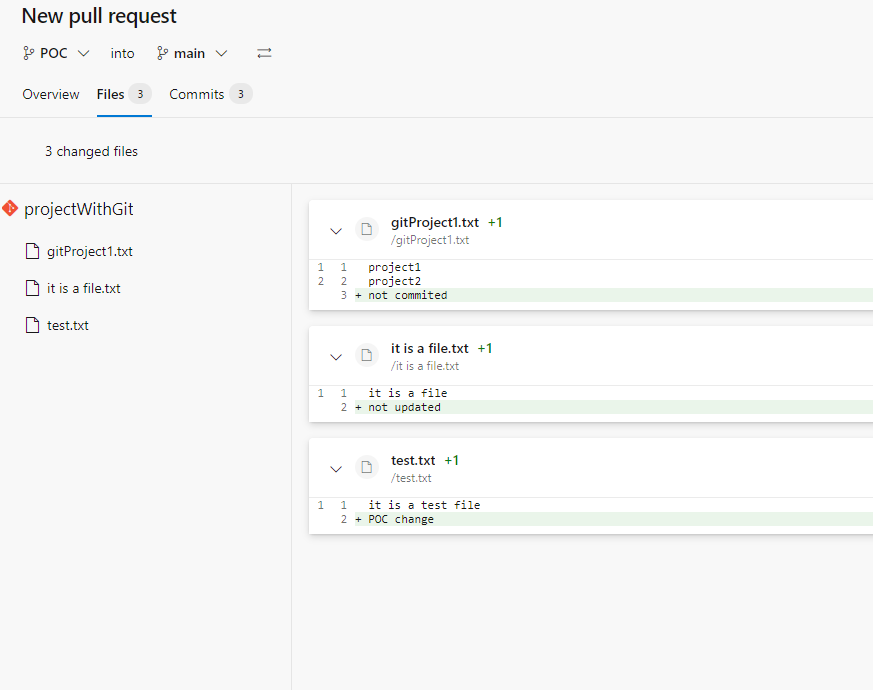
To pull a request, go to Pull requests in Repos. You can view requests of Mine (all requests created or assigned by me), Active, Completed and Abandoned (marked abandoned by reviewer). Right now there is no request there.



Click New Pull Request button, in the New Pull Request screen choose the source and target branches, select reviewers, add work items or tags if necessary.

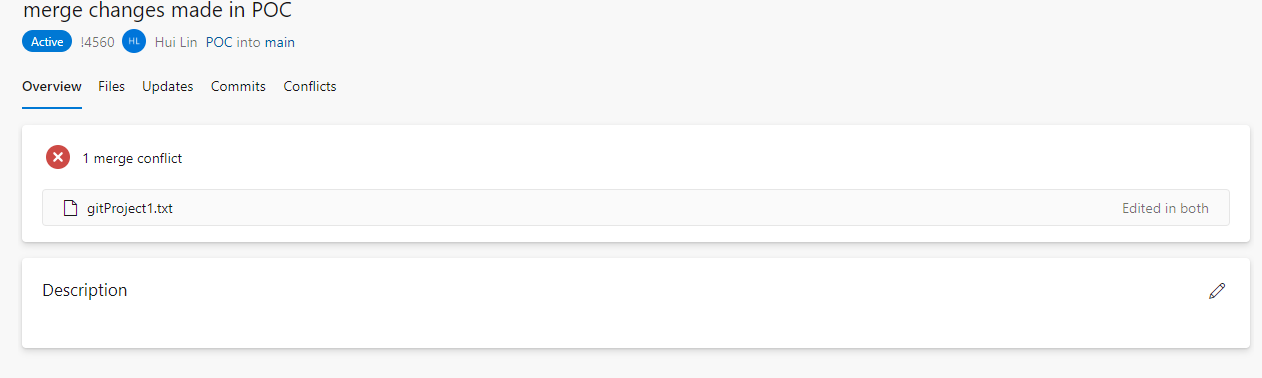


When checking Files menu, you can see the changes in files that will be pulled to the target main branch are marked green with a plus sign in the front of the row, which means these lines will be added to the main branch. You can verify the changes and click Create button to pull. The pull request becomes active once created.

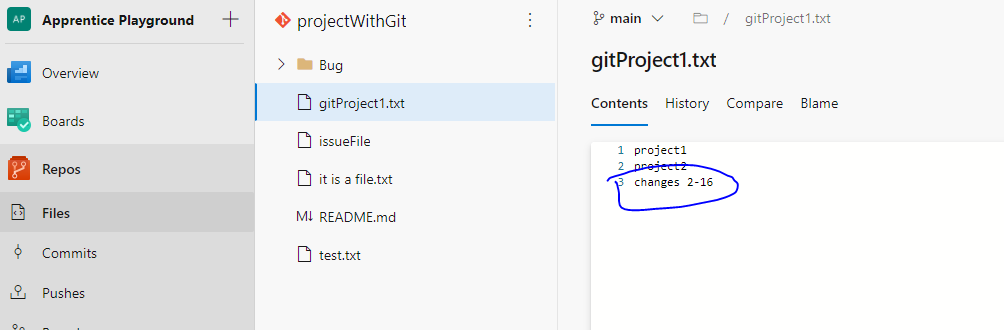


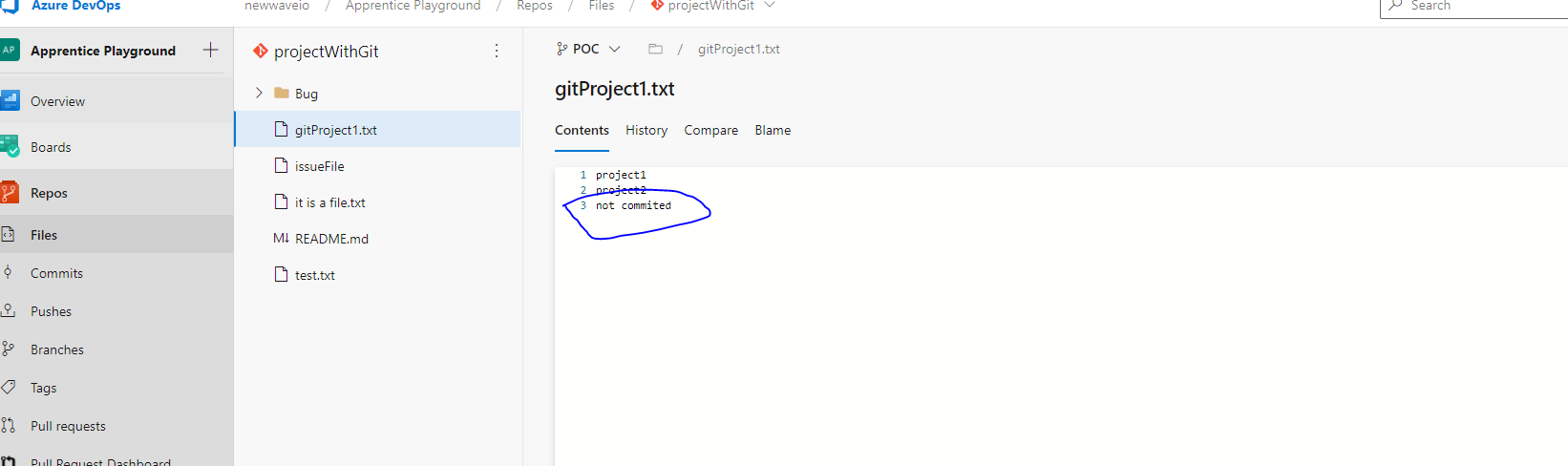
1. How to solve conflict in a pull request

If a file has changes at the same place in both source branch and target branches, you may get a conflict when pull. The screenshot below shows the pull from POC to main has a conflict and it gives me a description that the file gifProject1 has been edited in both branches, which means the file was changed at the same place in both branches.



I go back to Repos > Files and open the file gitProject1 in the main branch and the file in the POC branch. I see both files are changed in the same place; that caused the merge conflict.

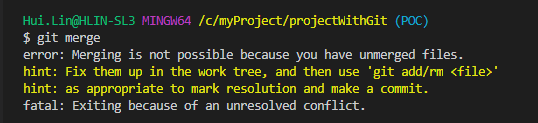


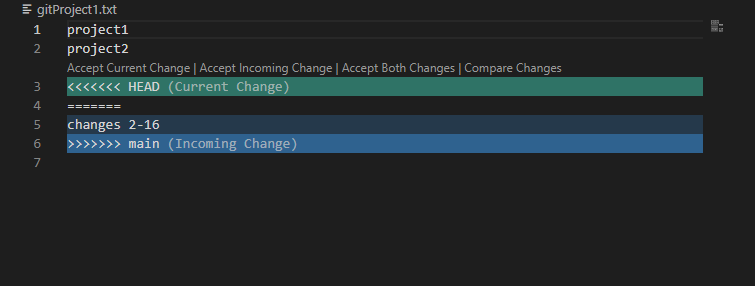


Resolution

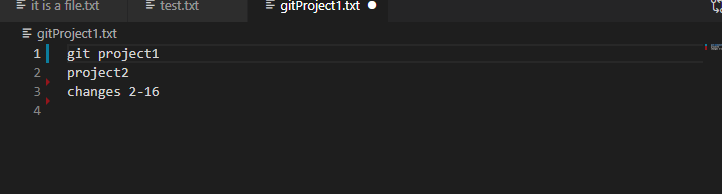
1. Checkout main branch in your local repository
2. Pull the main branch from the server repository
3. Checkout POC branch your local repository
4. Merge main to POC in your local repository
5. Select how you wish to resolve the conflict (Accept incoming change, Accept current change, Accept both changes)
6. Then push POC branch to the remote repository.

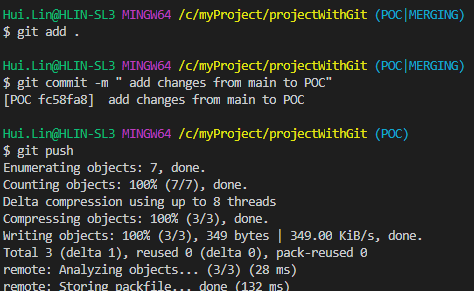




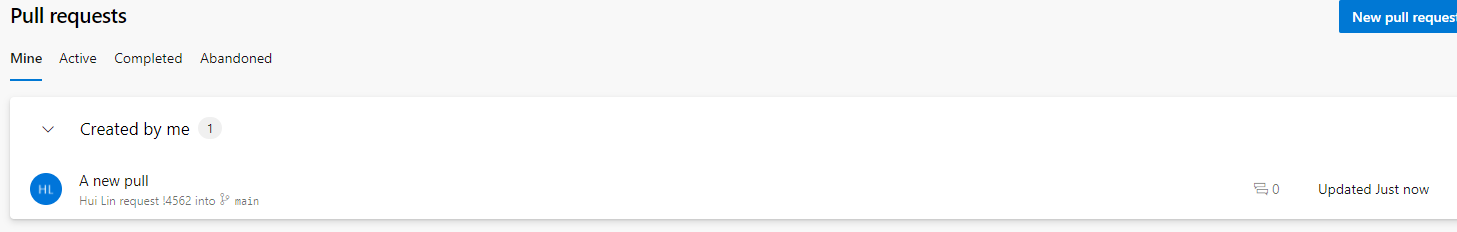


I selected Accepting Incoming Change

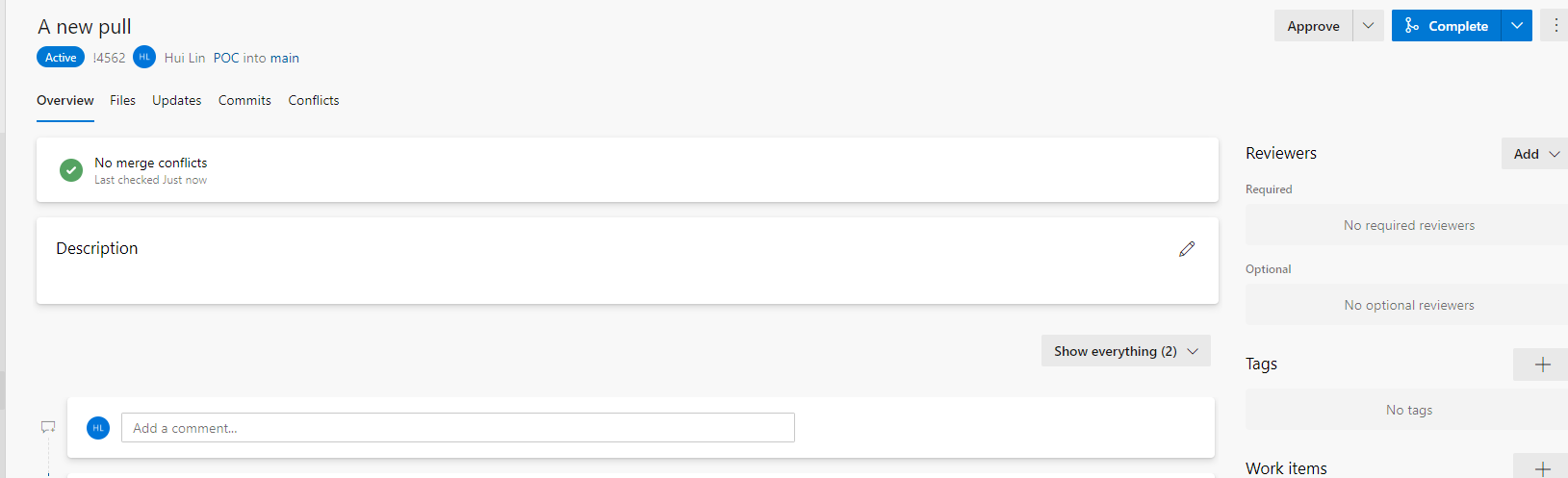




Back to Pull Requests page in Azure DevOps, the conflict is gone.



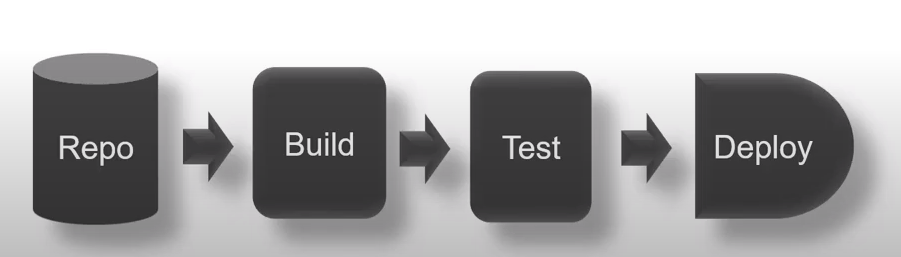
Now I can select the complete button



**Azure DevOps Pipelines Walkthrough**

1. Pipelines

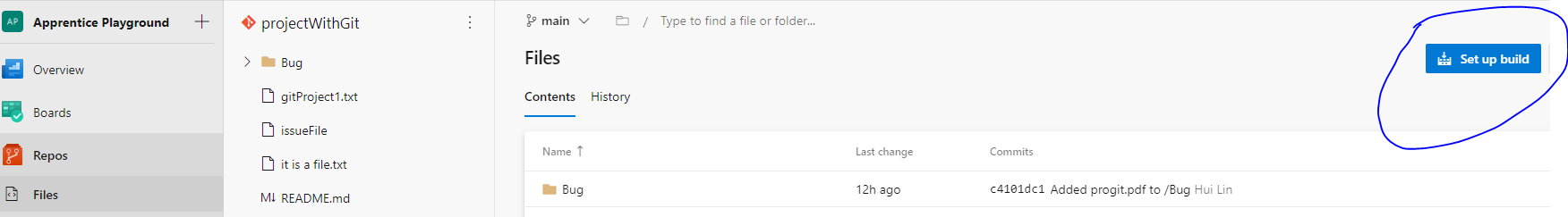
* Pipeline in DevOps is a set of process that is used to make your project code available to users. It can be done automatically or triggered manually.
* Pipeline is designed based on the project need. You can increase or decrease number of steps while designing your Pipeline.
* Process in Pipelines

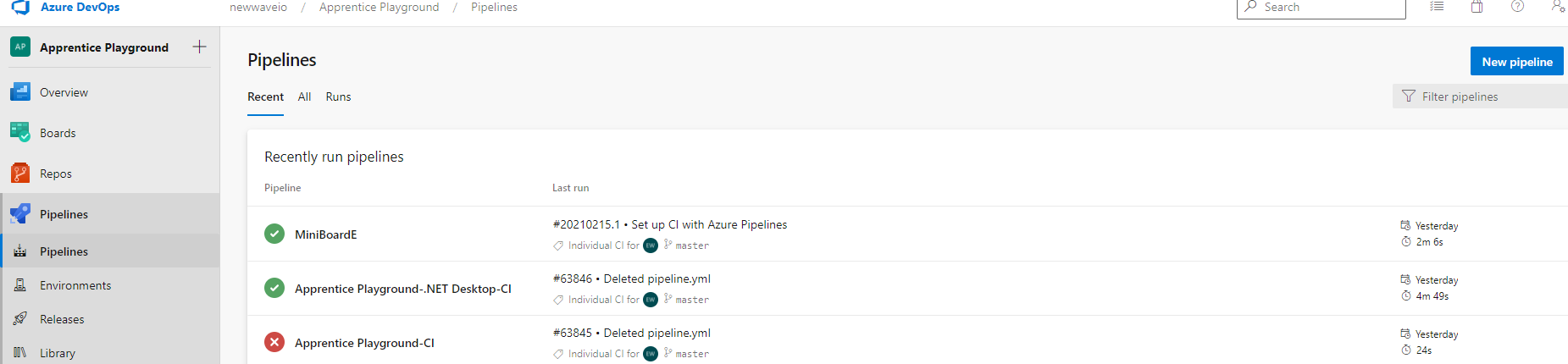


* Azure DevOps works with any programming languages and platforms. Azure DevOps can be used to deploy your code on multiple targets (any cloud target, any on-premises servers, VM)
* CI (Continuous Integration) CI is used to automate tests and build for project. It is used to find bugs or other build issues in early phase.
* CD (Continuous Delivery) CD is the automated (or semi-automated) process by which committed code is released to production.
* CI and CD can be triggered manually or automatically (on each commit, on a fixed interval or at a particular time)

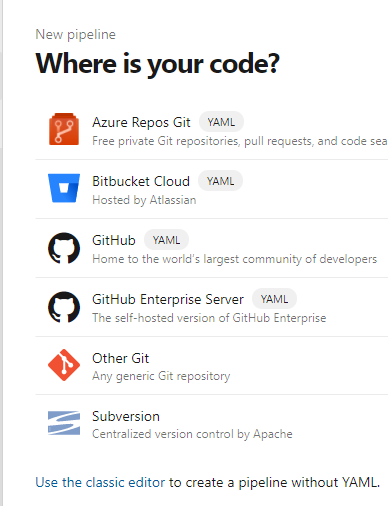
1. How to create a new build (CI) pipeline.

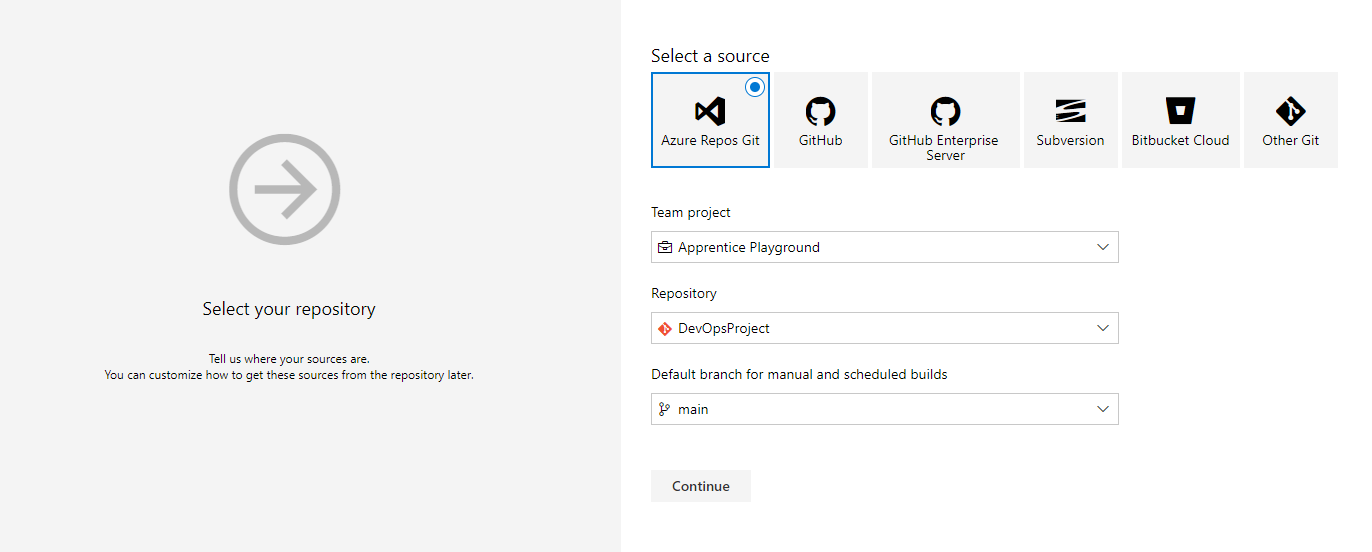
* One way is to go to Repos and click on Set up Build. By default the repository listed on this page will be used as the source for build. Alternatively, go to Pipelines > pipelines to click on the New Pipelines for a new build.



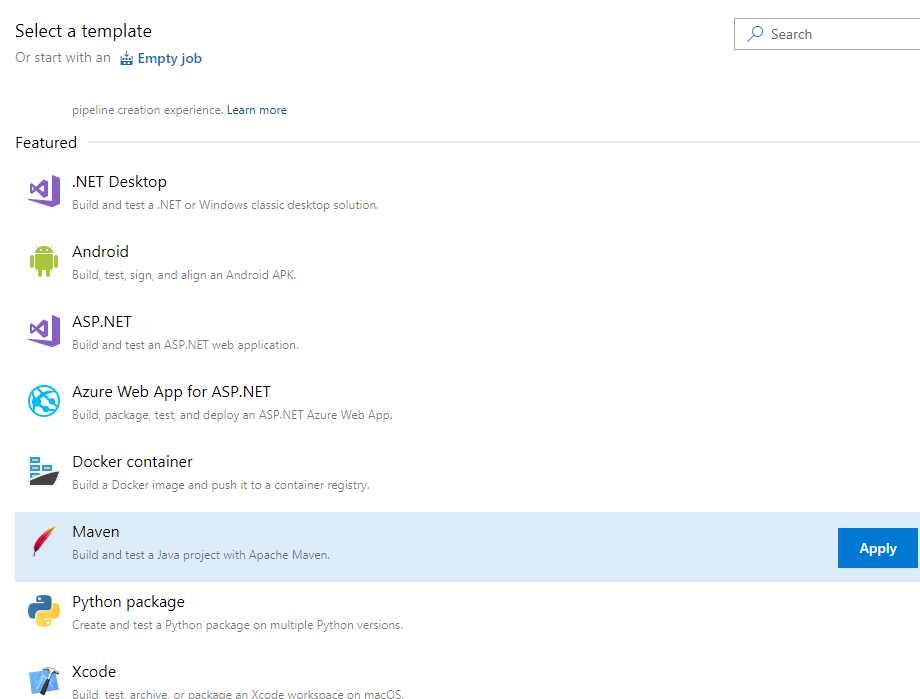


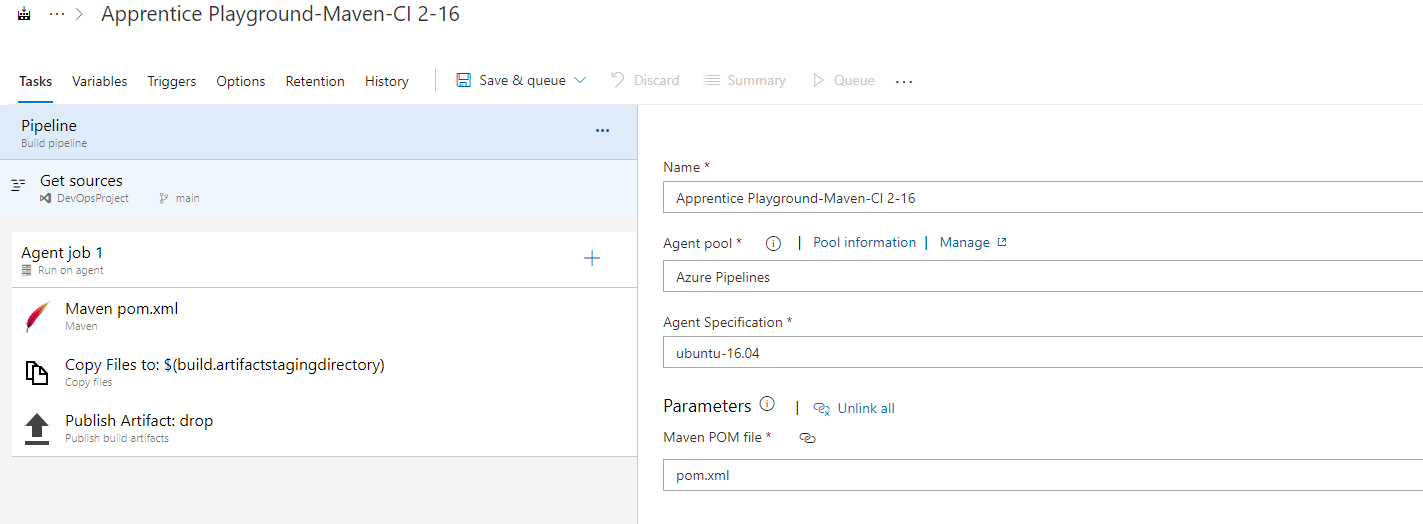
You will then be prompted to select where your code is stored. I selected Classic Editor so I can create a pipeline without YAML.



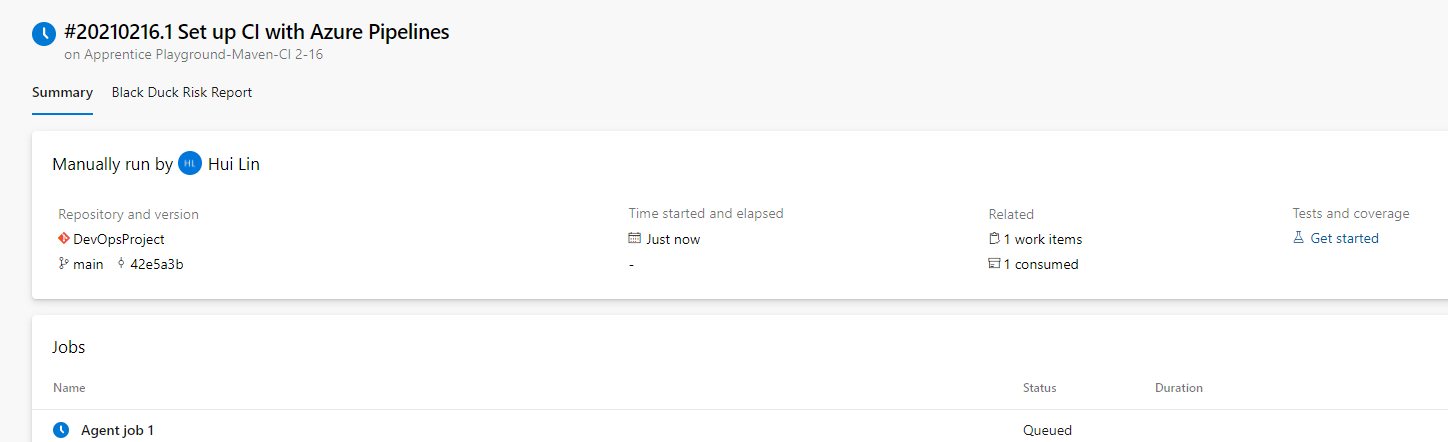


Then select a template

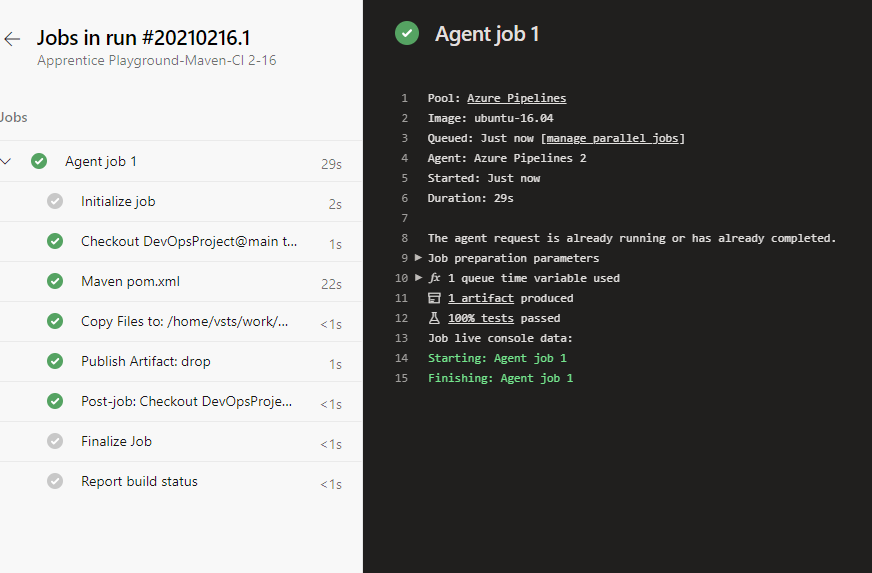




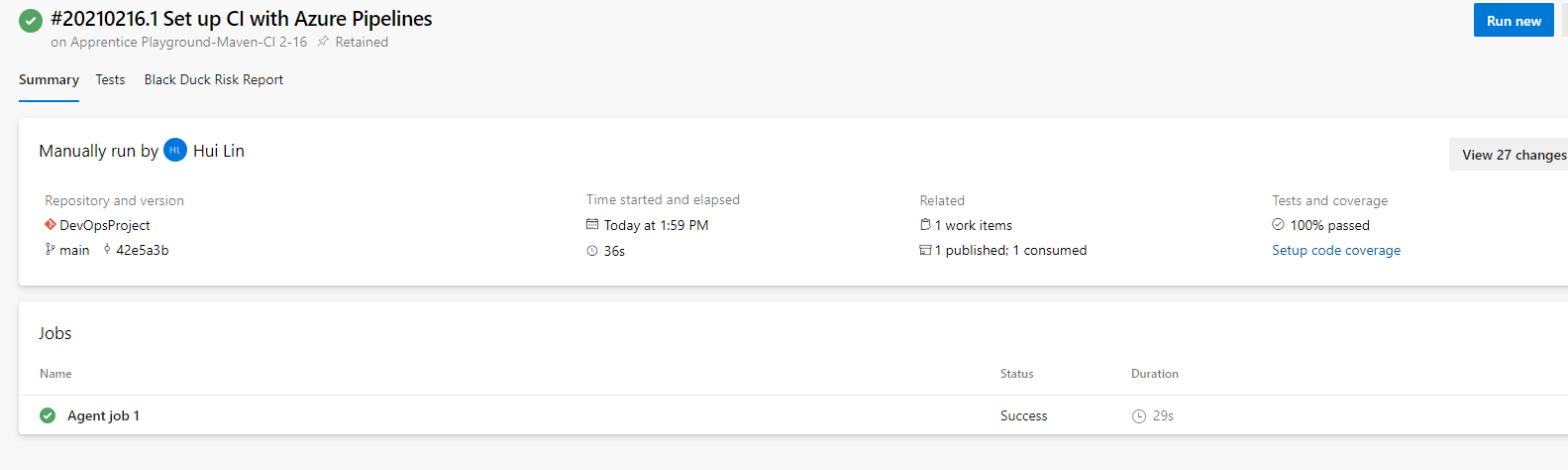
Make sure every field is filled out correctly for the new pipeline then click Save and Queue. Next in the following summary page you will see the status of your pipeline (queued, running).



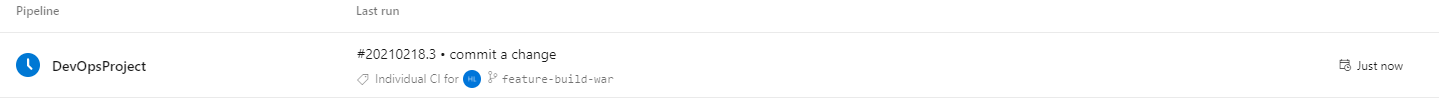
The build pipeline will return information in real-time as each job and task in the pipeline runs.



Once the build has completed, you are greeted with green checkmarks as you can see below.



If I make a change in my local repository and push it to the server repository, immediately pipeline will be showing a changed committed and it will rerun automatically. So this is how CI works.



1. Terms for Azure DevOps Pipelines

* Agent

1. When your build or deployment runs, the system begins one or more jobs. An agent is an installable software that runs one job at a time. In Azure DevOps we have two types of agents available: Microsoft – hosted agent (installed by Microsoft automatically) and Self – hosted agent.
2. With Microsoft-hosted agent when you run a pipeline a new virtual machine defined in the yaml file is created for you. The agents working on the virtual machines get the code from the repository and build it for the further task.

* Environment

An environment is the place where we deploy the application. For example, VMs, Container.

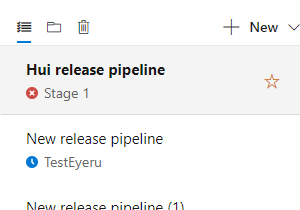
* Stage

A stage is used to mark separate concerns. For example, create build for QA, Production etc. Each stage contains one or more jobs. Stage is very important in adding an extra layer of validation before your code goes to production.

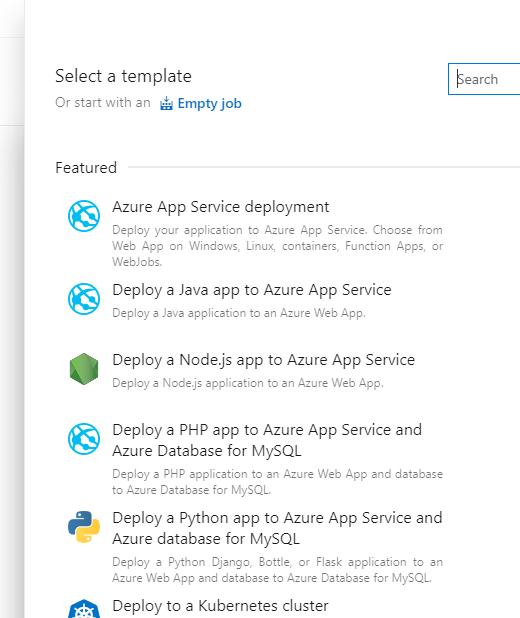
1. How to build a release pipeline

A release pipeline takes a build artifact, a result of the build process and deploys that to one or more environments.

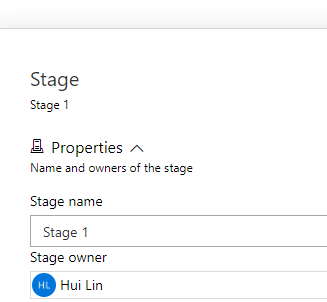
Go to Release and click the New option to create a new release pipeline.



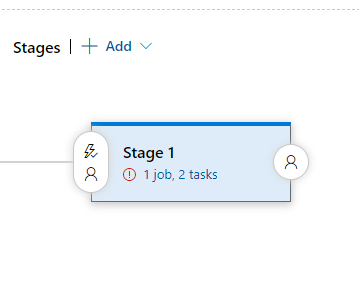
From the template list on the right, select your template.



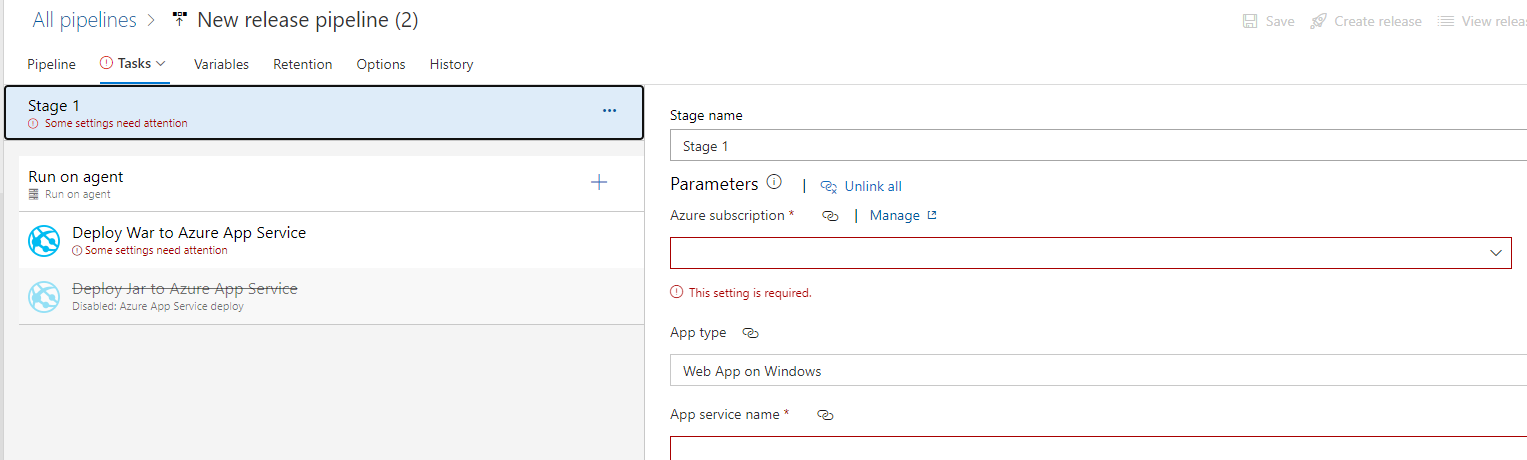
Provide a description for the Stage Name. The stage will contain release tasks.



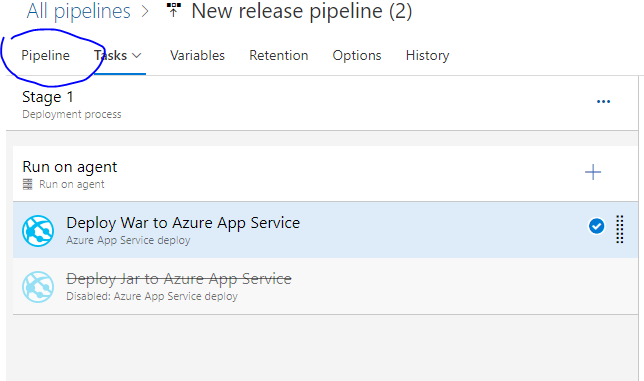
In the Stages field, select 1 job, 2 tasks. This field is where you will eventually provide the settings of the selected template you will use for the actual deployment.

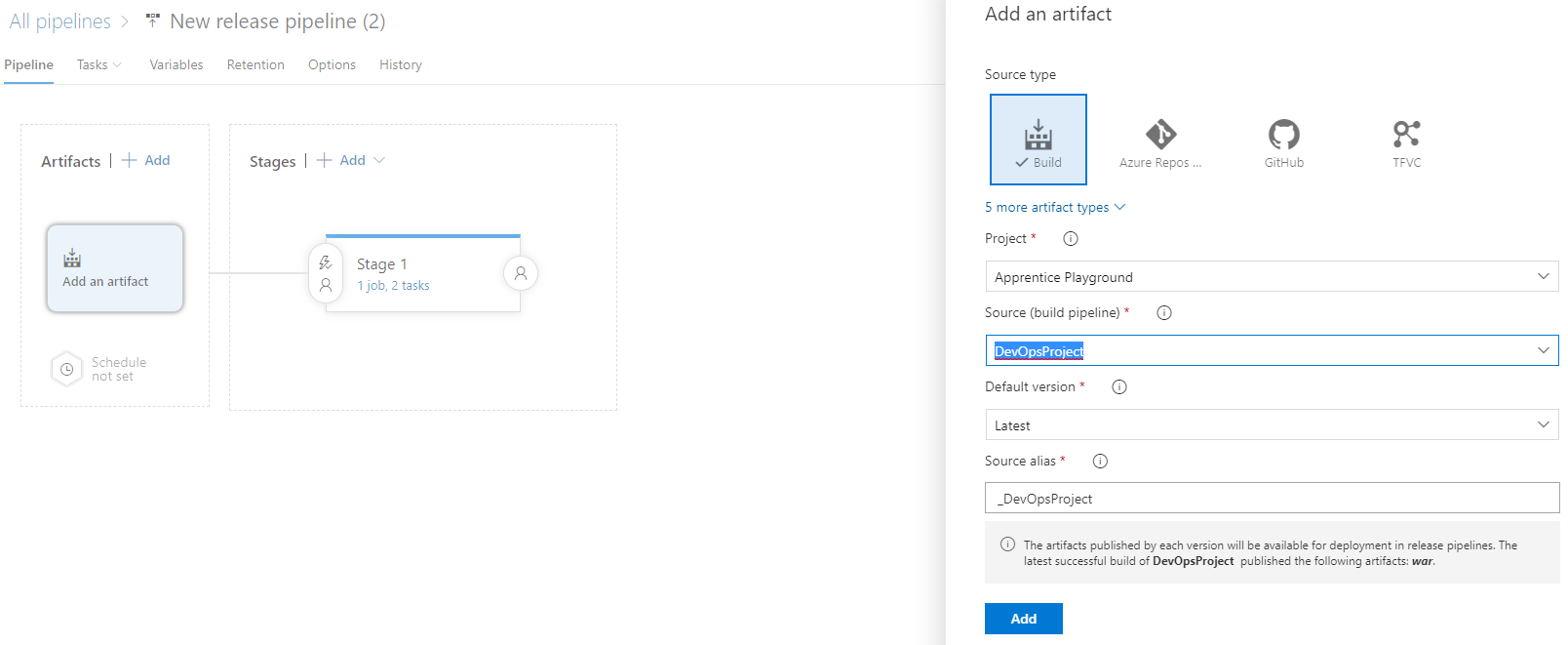


Fill out the parameters and save.



Click Pipeline to select artifacts.

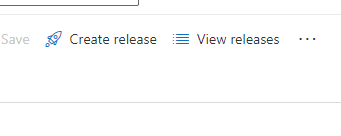




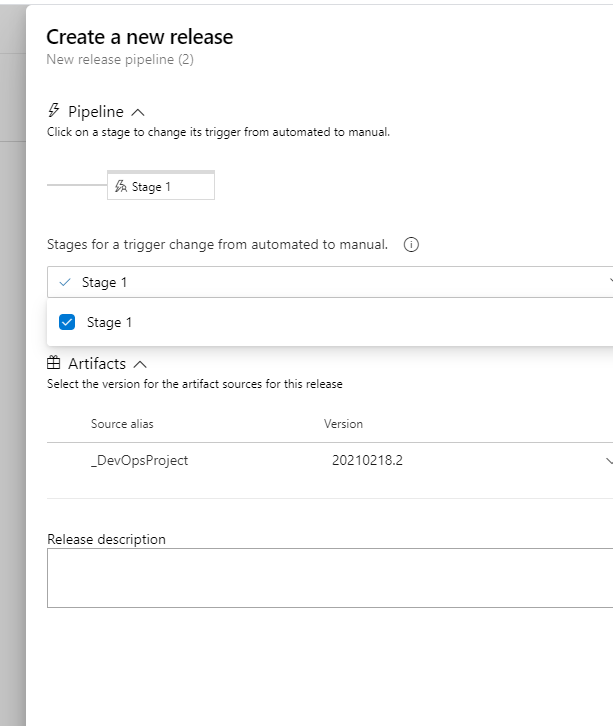
Click on the Save button in the upper right corner of the screen to save the release pipeline.

1. How to create a release

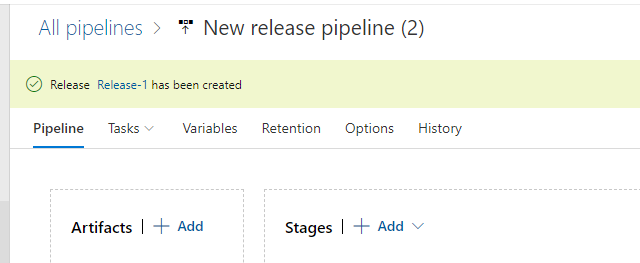
Once the release pipeline is created, you will then begin creating releases. A release is a particular run of the pipeline.



Accept the defaults and click on Create.

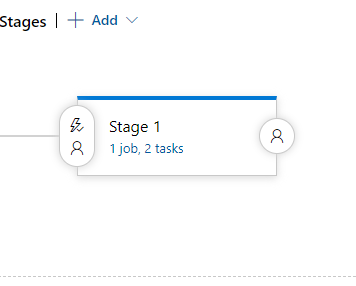
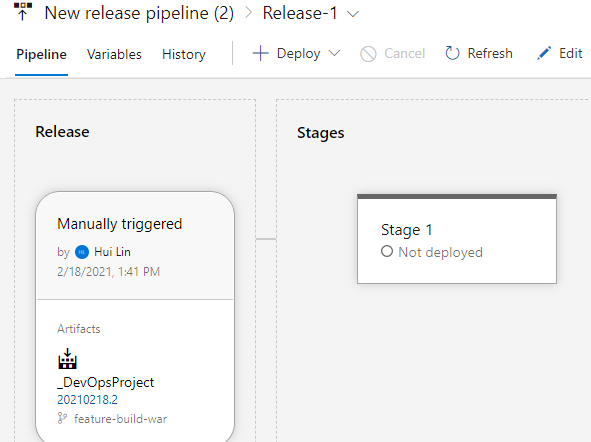


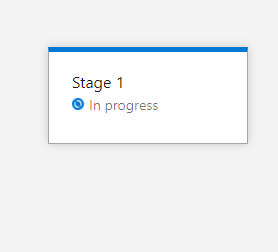
 After a few seconds, you will receive a notification (the green bar) that the release has been created.

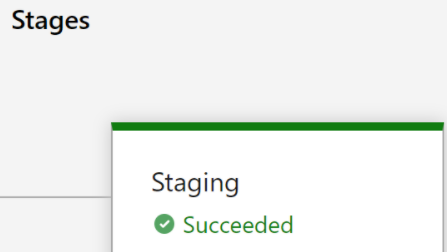


1. How to deploy release or deploy code to an environment.

In the newly created Release, select the Deploy stage and confirm Deploy







1. Library

* Library is a collection of build and release assets for a project.

1. Task Groups

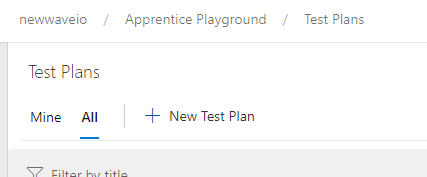
* Task groups area way to standardize and centrally manage deployment steps for all your applications. When you include a task group in your build definitions, and then make a change centrally to the task group, the change is automatically reflected in all the definitions that use the task group. There is no need to change each one individually.

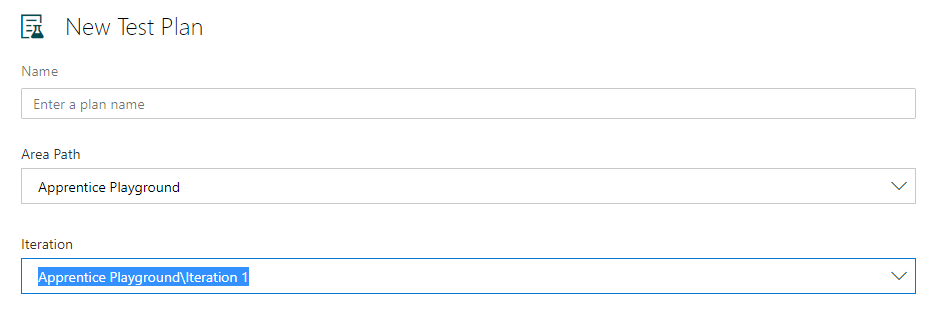
**Azure DevOps Test Plans**

1. Test Plans

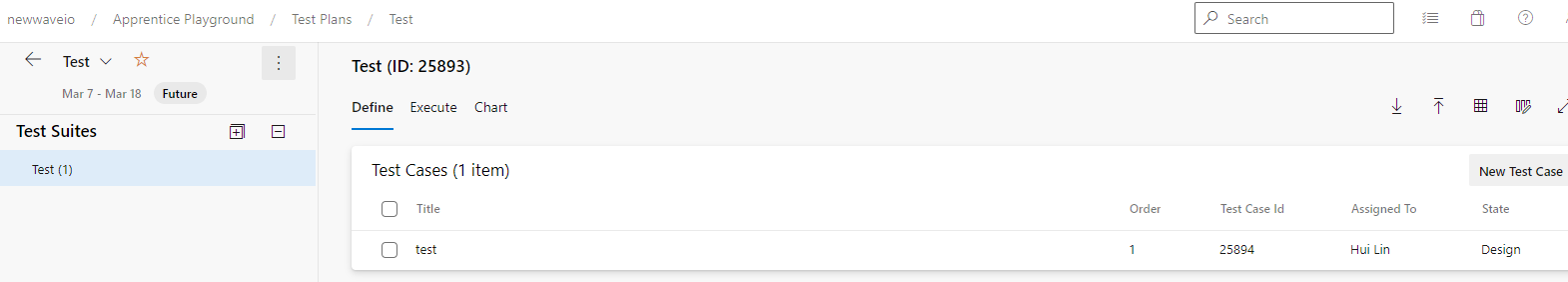
* Azure Test Plans provides three main types of test management artifacts: test plans, test suites, and test cases. These elements are stored in your work repository as special types of work items.
* Test plans group test suites and individual test cases together. Test plans include static test suites, requirement-based suites, and query-based suites.
* Test suites group test cases into separate testing scenarios. Grouping test cases makes it easier to see which scenarios are complete.
* Test cases validate individual parts of your code or app deployment.

1. How to create a test plan

* In the Test Plans page, choose New Test Plan to create a test plan. In New Test Plan page, enter a name for the test plan, select the area path and iteration, then choose Create. 

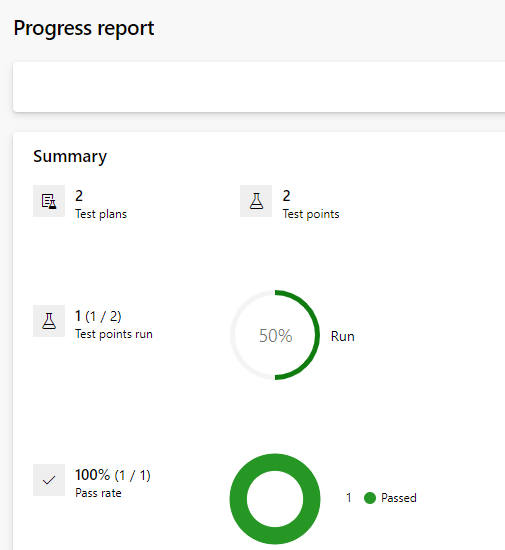


* Once a test plan is created, you can create new test cases (that can be linked to a work item) and group them together into test suits.



1. Progress Report

* You can use Progress Report to track the progress of more than one test plan or test suite.



**Azure Artifacts**

An artifact is a collection of files or packages which are created by a build run. Once you publish an application you get a collection of files that are required to run your application. They are called artifacts.