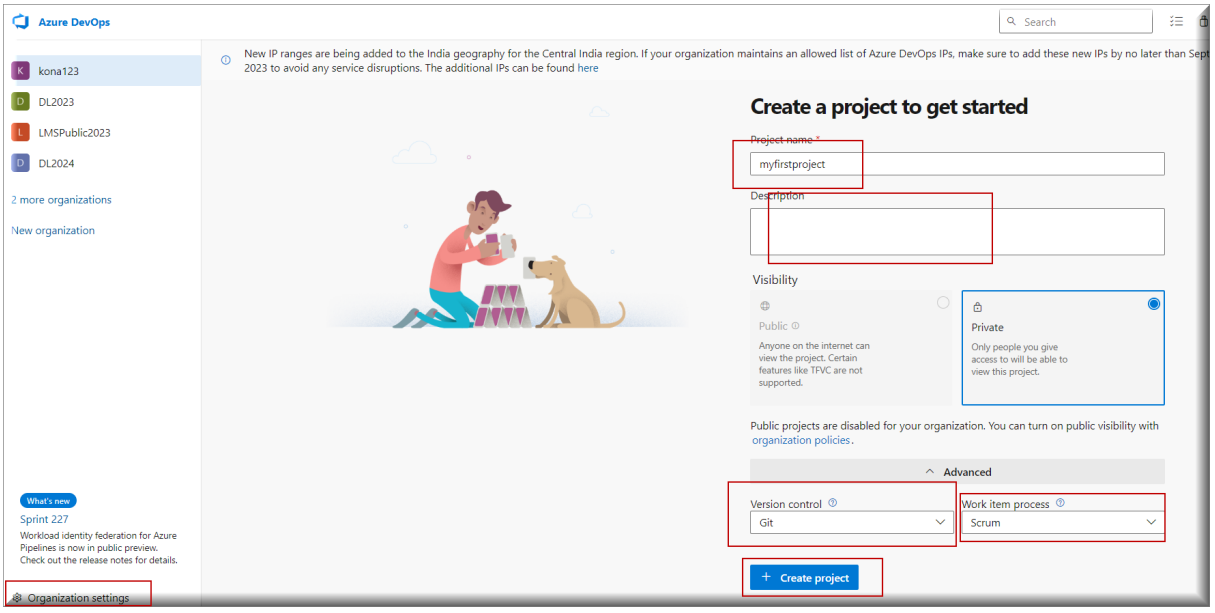
Creating Your First Project

Click on the **New project** button on the top right-hand corner to navigate to the new project creation page.



Provide a suitable name and a description to your project and make it either **Public** or **Private**. Here, you have the possibility to select your version control system for versioning your project resources. By default, it is set to **Git**, which is a distributed version control system. If you prefer a centralized version control system, then you can use the **TFVC** option.

In addition to that, you can select which work item process you prefer to choose. There are 4 main options to choose from.

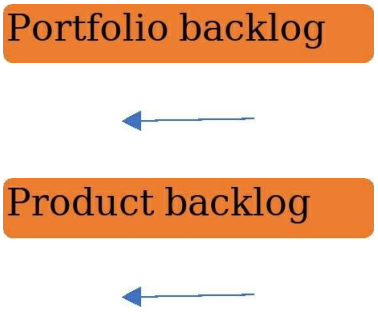
1. Basic
2. Agile
3. Scrum
4. CMMI

Let us have a brief look into these different work item processes.

Basic

This is the simplest model you can choose out of the four. It has only 3 work item types (WITs)

1. Epic



2. Issue
3. Task

These 3 work item types help us to organize our work in a hierarchical way**. Task** is the smallest unit. **Issue** is the parent of **Task**, and **Epic** is the parent of **Issue**. Epic comes under portfolio backlog which lets you to organize your work starting from a high-level business perspective. Following are some examples for **Epics**.

Improve the user friendliness and user experience

-

Convert the paper-based quiz into a web-based application

-

Issues on the other hand, focus more on implementing the Epics on a feature basis. An issue can be considered as a shippable feature of the product.

Following are some examples of Issues.

Add a login functionality Create a notification feature

-
-

Add new icons to improve look and feel Design a new colour scheme

-
-

Issues can be further divided into small tasks. Usually, these tasks should not take more than one day to complete. So, the whole purpose of each task is to implement/fix a given issue.

Agile

This is good for teams using Agile planning. Here, the development and test activities are tracked separately.

Following are the main WITs associated with Agile process.

1. Epic
2. Feature
3. User Story
4. Bug
5. Task

Epic and **Feature** are on the top level, and **User Story** and **Bug** can be managed separately. You can create Tasks for both User Story and Bug. Like the Basic process, you can group your work items according to your needs.

Here, you have more flexibility to organize your work than the Basic process. However, the concepts are basically the same described under the Basic process.

Scrum

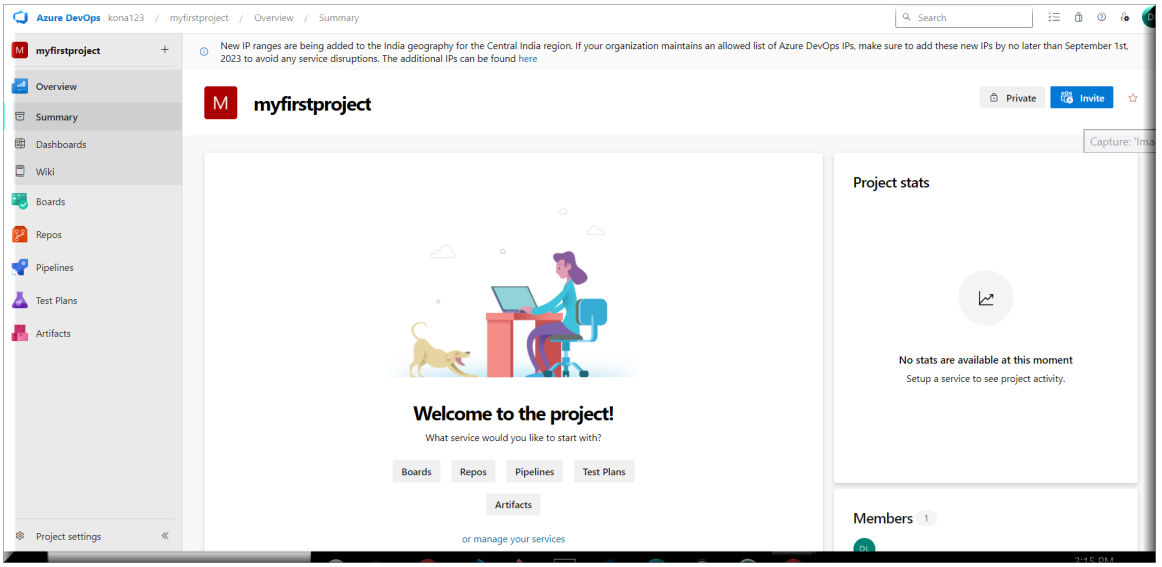
If your team is supposed to practice Scrum, then this is the most suitable type for you. It is quite like Agile where **User Story** is replaced by **Product Backlog Item** (PBI), and **Issue** is replaced by **Impediment**.

CMMI (Capability Maturity Model Integration)

This process can be used if your team follows a more formal approach that requires a framework for improving the process and decision making. It is possible to track requirements, change requests, risks and reviews.

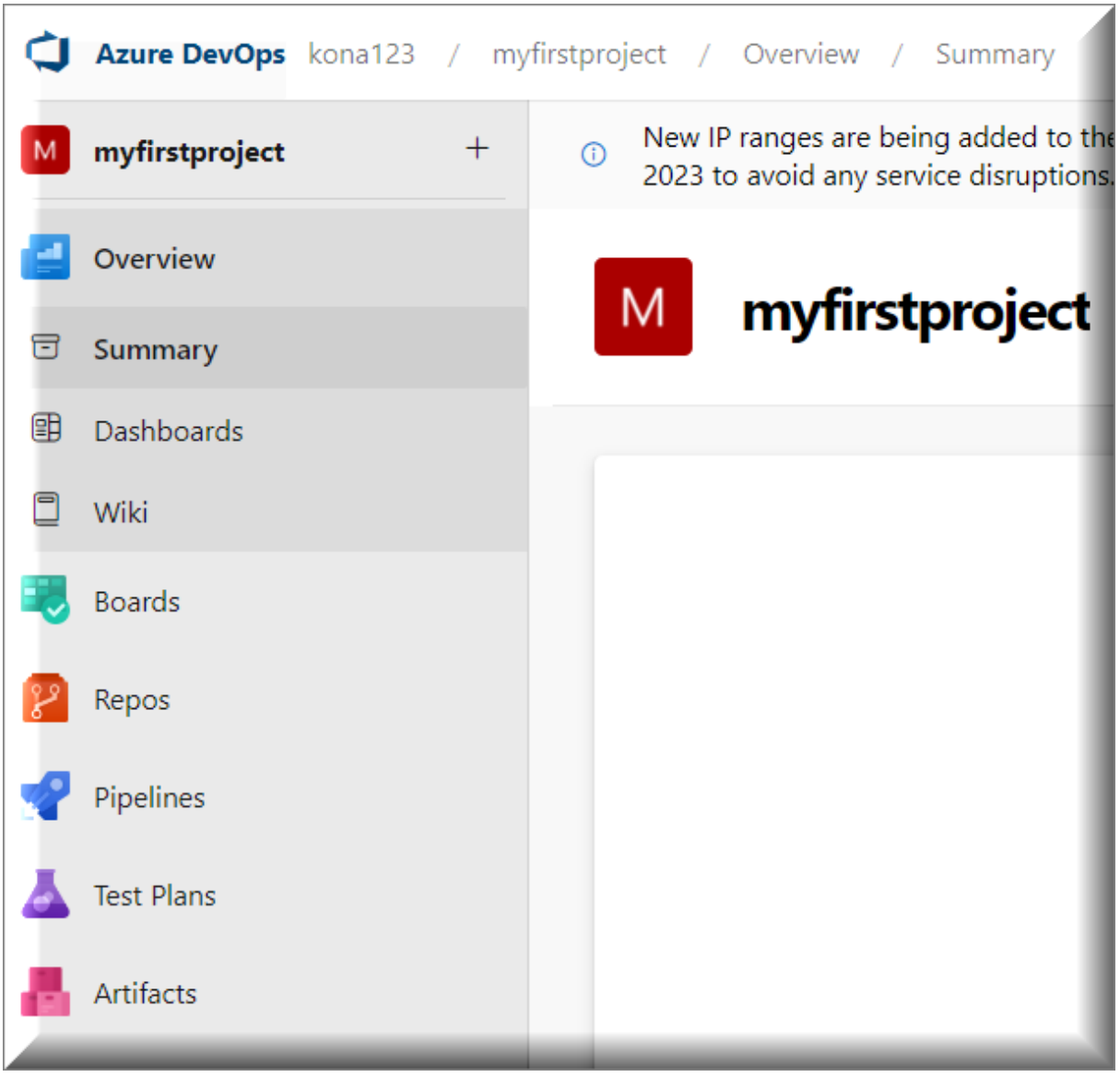
Now, we have some understanding about the work item process. So, in this project, I am going to select the simplest process - **Basic**. Click on the **Create Project** /**Save** button to create the new project.

Now you have created your first Azure DevOps project for your organization. The project summary page is shown in Figure 18.



Project Settings

In this section, I will highlight some of the most important settings under the project section. To access the settings page, navigate to your project and click on the **Project Settings** link on the bottom left-hand corner.

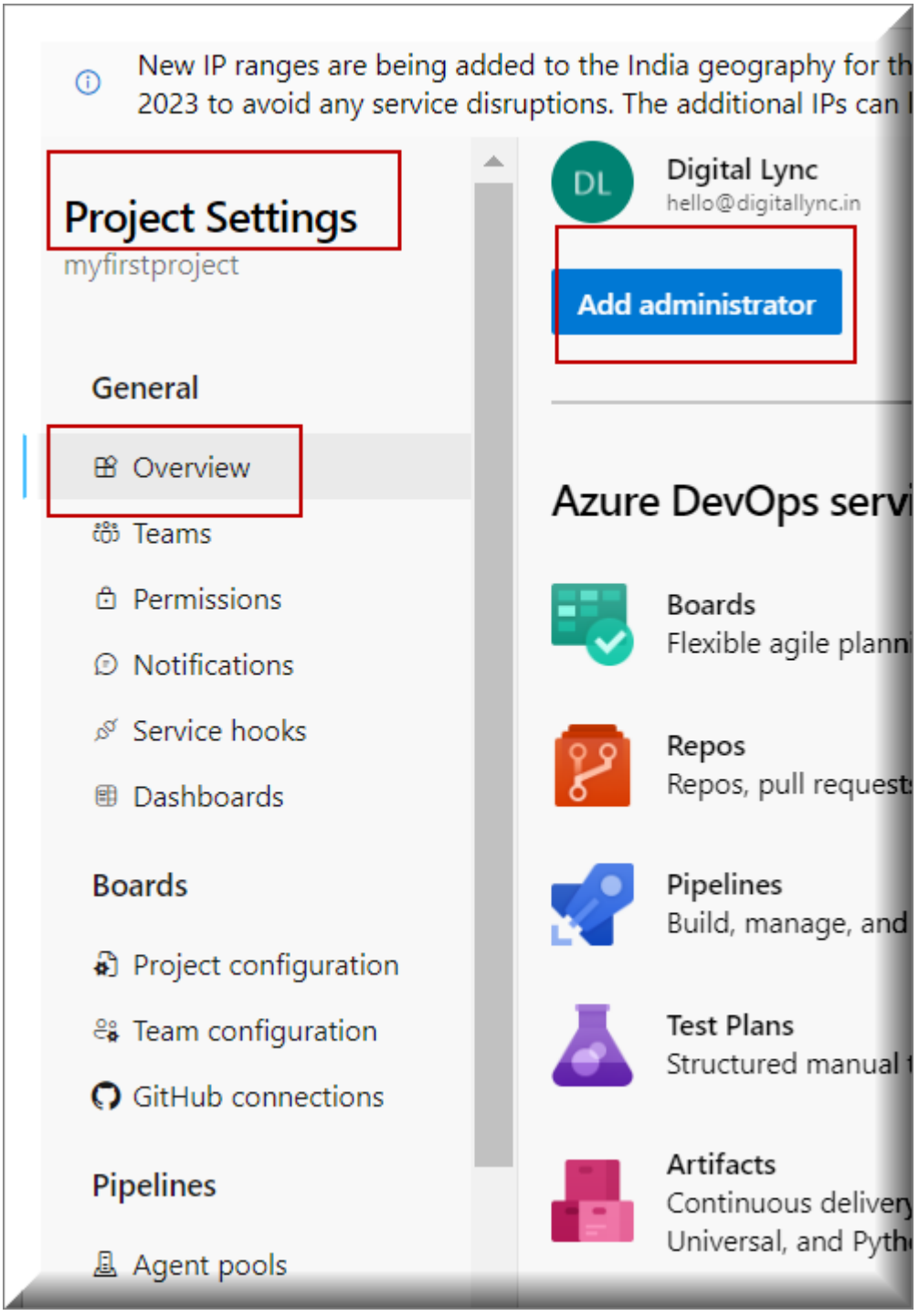


General → Overview

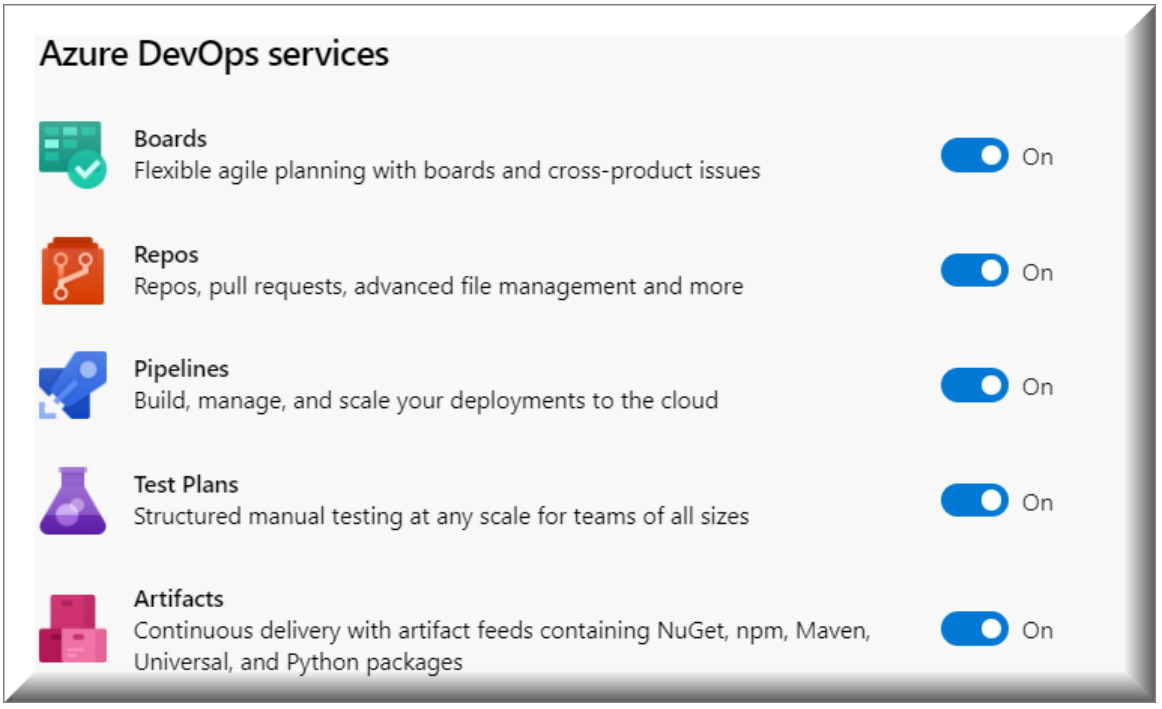
Here, you can rename your project, change the work item process and visibility of the project etc. However, be careful when changing the work item process, as it can be a breaking change and you have to manually fix states of your tasks.

In addition to that, you can add more administrators to the project by clicking

on the **Add administrator** button.



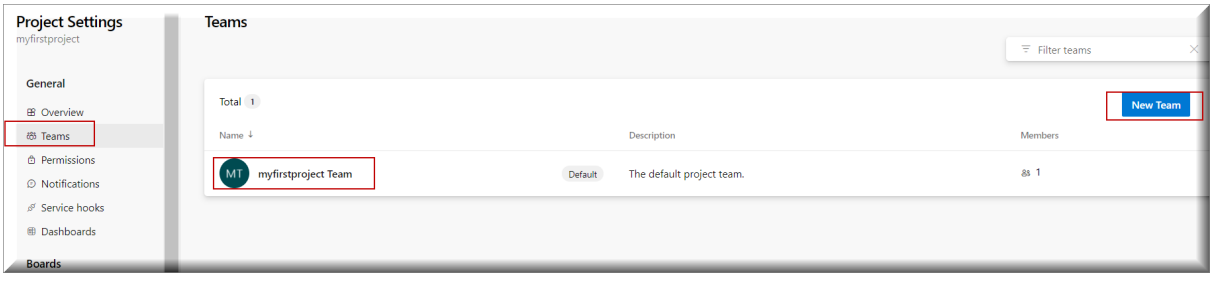
Another nice feature that you can change here is the Azure DevOps services.



Here, you can turn on/off different Azure DevOps services. For example, if you only need a place to store your source code, then you can only turn on ***Repos*** and turn off all the other features.

General →Teams

You might have several teams working on the same project. For example, a developer team, sales team or a support team. That can be done under this section. In addition to that, you can add users to different teams.

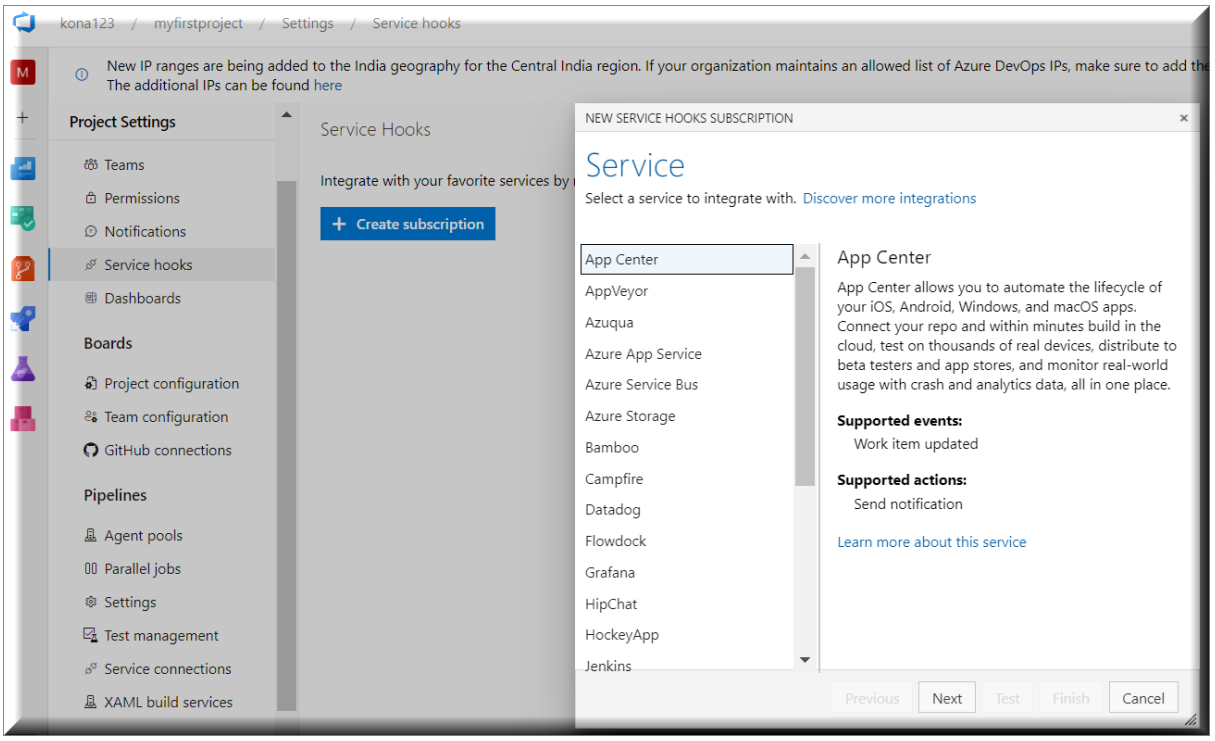


General → Permissions

This section is more or less the same as the permission in the organization settings. But in this case, these permissions you set here will only be applicable to the current project.

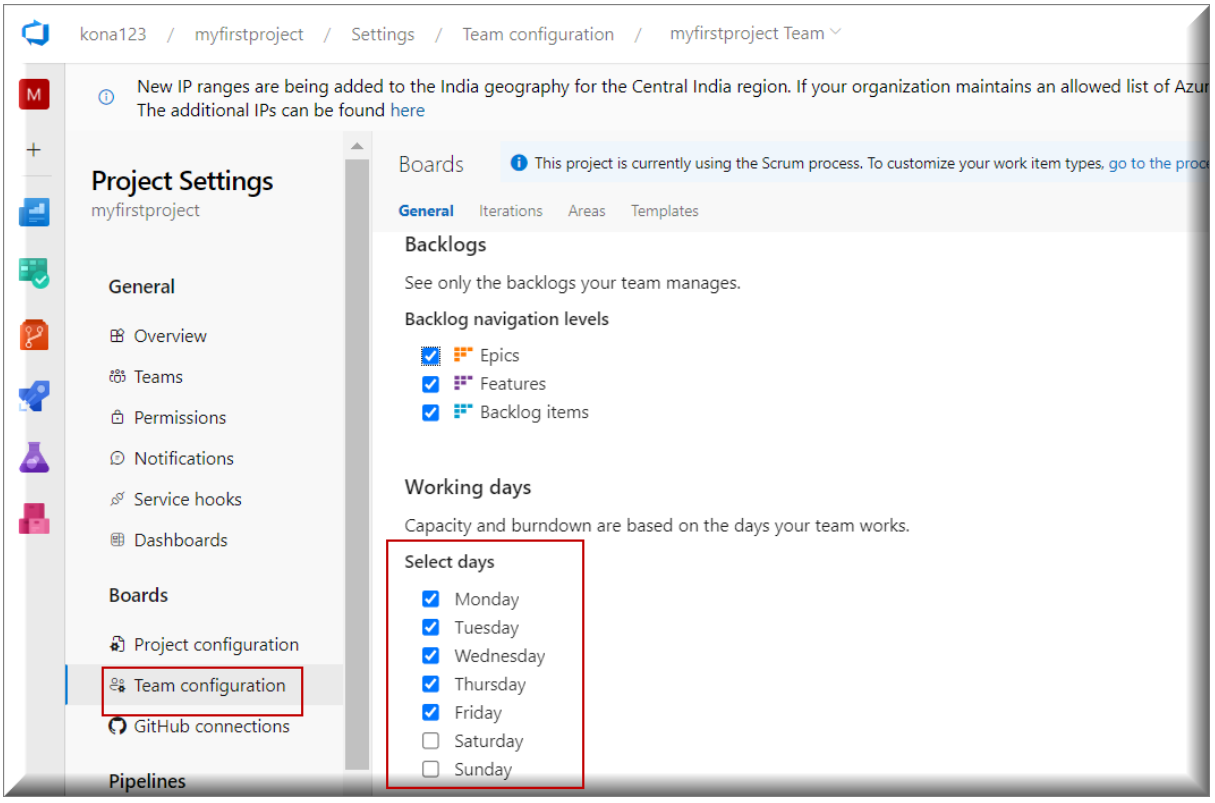
General → Service hooks

Here, you can integrate with third party software/applications. For example, say your support team uses Zendesk as their software to register tickets considering customer issues, then you can integrate those tickets with Azure DevOps using a service hook.



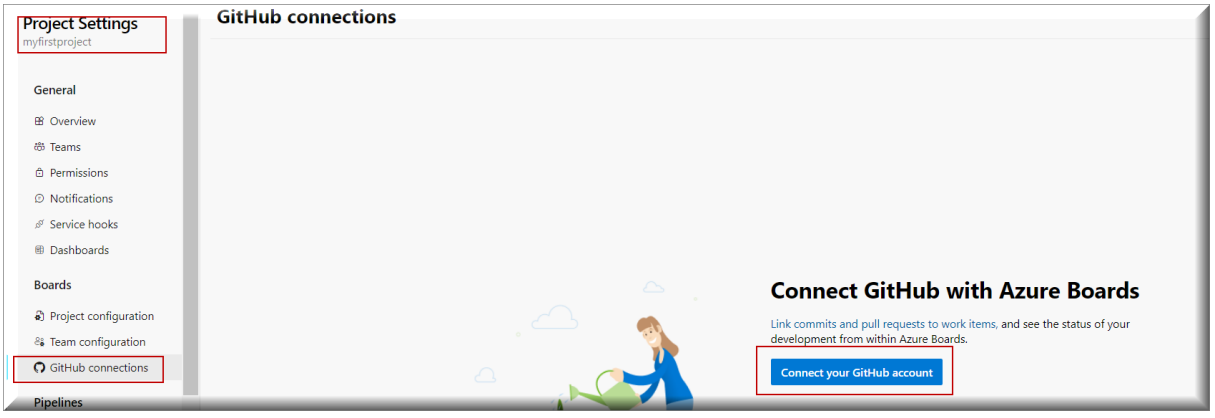
Boards → Team configuration

Here you can set which working days, the team is working on this project.



Boards → GitHub connections

To connect your GitHub account, click on the button **Connect your GitHub account**. After you have given authorization to GitHub to access the project, then you can select which GitHub project you want to associate with this project.

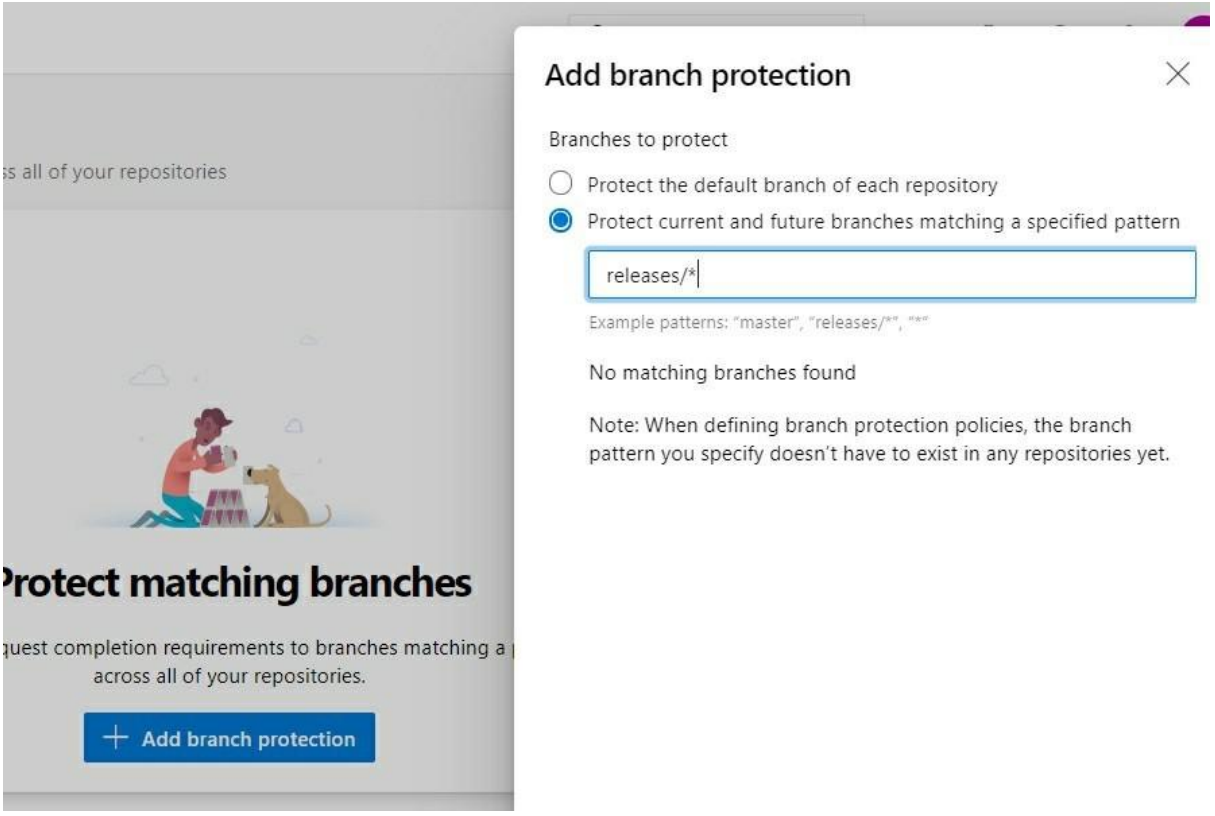


Repos → Repositories

Here you can either create a new repository or edit security, options and policies of an existing repository.

Repos → Cross-repo policies

If you want to protect certain branches of your source code, then you can use this settings page. Click on the **Add branch protection** button to load the modal dialog to add branch protection. For example, if you want to protect the default branch (usually the master branch), you select the first option. If your release branches are located in the **releases** folder, then you can use the second option to protect all your release.



After you have done that, you get more options to protect your branches. For example, you can add a minimum number of reviewers to approve a certain pull request to that branch. In addition to that, you might always want every commit to the branch be linked to an associated work item. Likewise, you can

apply many policies according to your need.