**Azure**

Microsoft Azure, commonly referred to as Azure, is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centres.

**DevOps**

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

**Azure DevOps**

provides developer services for support teams to plan work, collaborate on code development, and build and deploy applications.

**What is ci/cd**

===================

A CI/CD pipeline is used to automate the process of continuous integration and continuous deployment.

The pipeline facilitates the software delivery process via stages like Build, Test, Merge, and Deploy.

In simple words, a pipeline may sound like an overhead, but it isn’t.

Instead, it’s a runnable specification of steps that reduce developers’ manual work by

delivering a new version of a software productively and saves time.

**Continuous Integration (CI)** is a development practice that requires developers to integrate code into a shared repository several times a day.

Each check-in is then verified by an automated build, allowing teams to detect problems early.

What is **continuous delivery** in DevOps?

AWS notes that continuous delivery is a DevOps software development practice

where “code changes are automatically built, tested, and prepared for a release to production.

**Stages of a CI/CD Pipeline:**

===========================

Source Stage

Build Stage

Test Stage

Deploy Stage

**Stages of a CI/CD Pipeline:**

1. **Source Stage –** In most cases, when a change is attempted to the central repository, a pipeline run is triggered. These triggers are set by the [CI/CD pipeline tool](https://www.lambdatest.com/blog/27-best-ci-cd-tools/) in the source stage.
2. **Build Stage –** The combination of source code and its dependencies when building into a runnable instance corporate to the end-user application. The built-in application languages like Java need compilation too, which is done in the build stage. If docker images are to be constructed, that can also be facilitated in this stage. Failing this stage marks a potential error in the code or its dependencies.
3. **Test Stage –** This stage corresponds to automated tests running to validate our code and its behaviour accordingly. This stage acts as a sieve that prevents the bugs from reaching the end-user. There can be multiple stages, from smoke tests to end-to-end integration tests. Failure at this stage will expose errors in the code.
4. **Deploy Stage –** Once we have a runnable code, the deployment is processed with all predefined tests passed. There are a lot of stages like “Beta,” “Staging,” etc., for the product team. A “Production” stage for the end-users is also present.

Remember, the stages mentioned above are the basic stages, and different steps can be added to make the CI/CD process more automated. To bring a new life to these stages, we have Azure DevOps CI/CD.

**Azure**

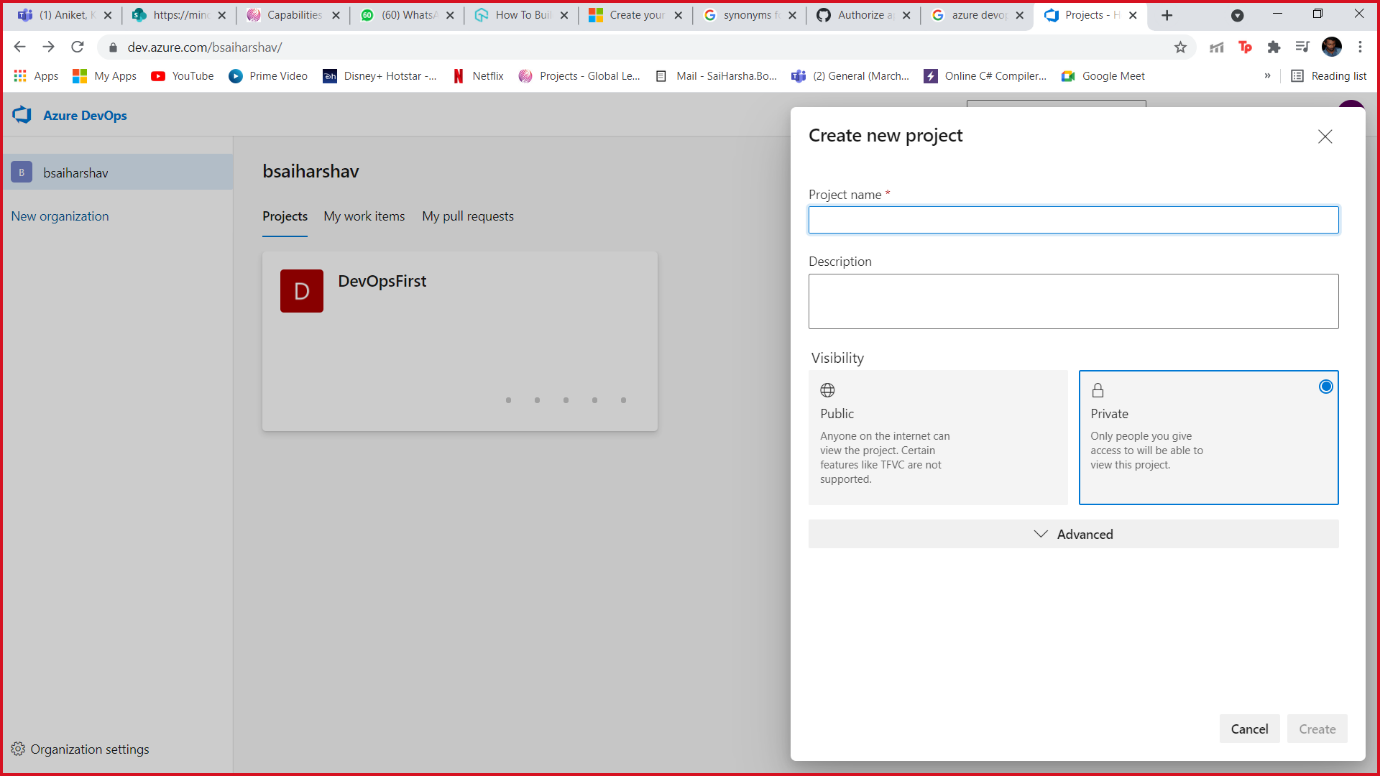
Microsoft Azure, commonly referred to as Azure, is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centres.

**DevOps**

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

**Steps**

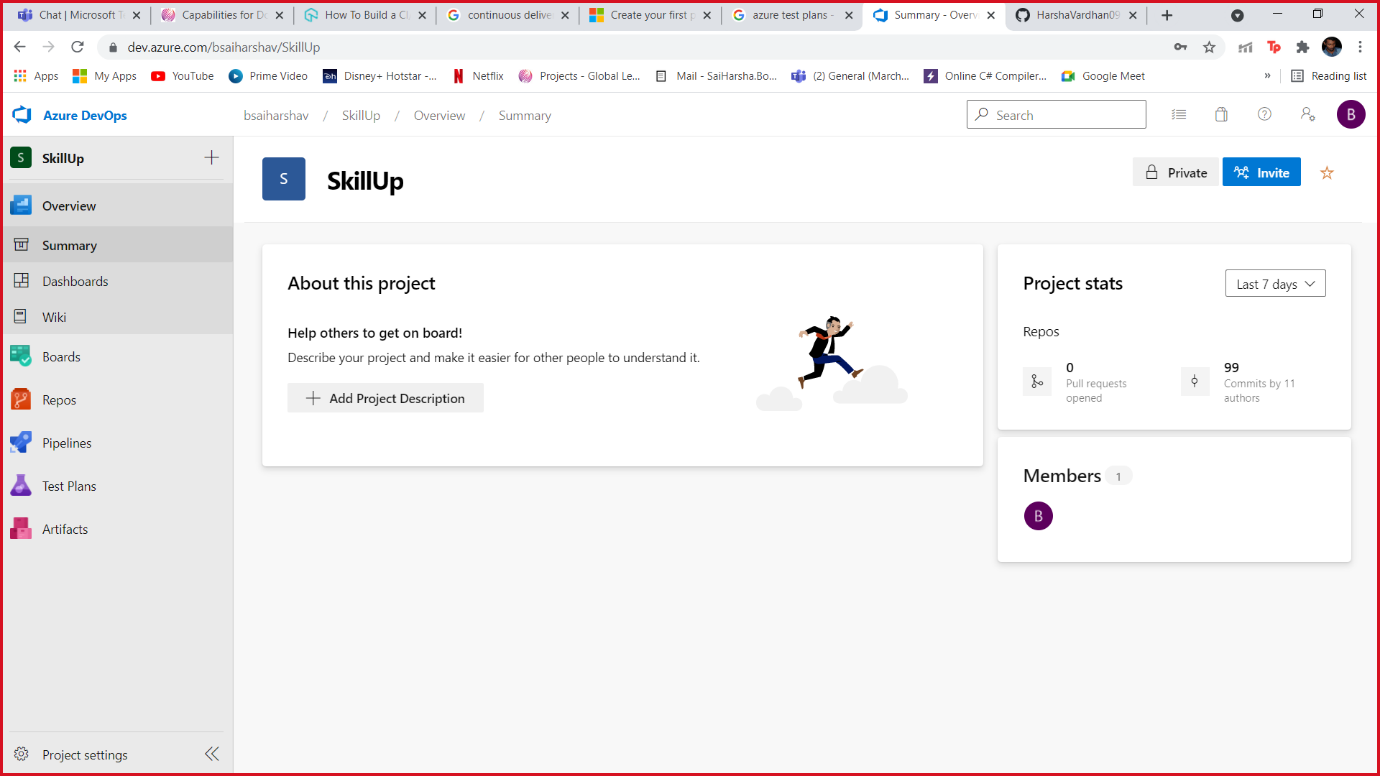
1. Create Project. So first go to <https://dev.azure.com> . Then Create new project and click on private project.

****

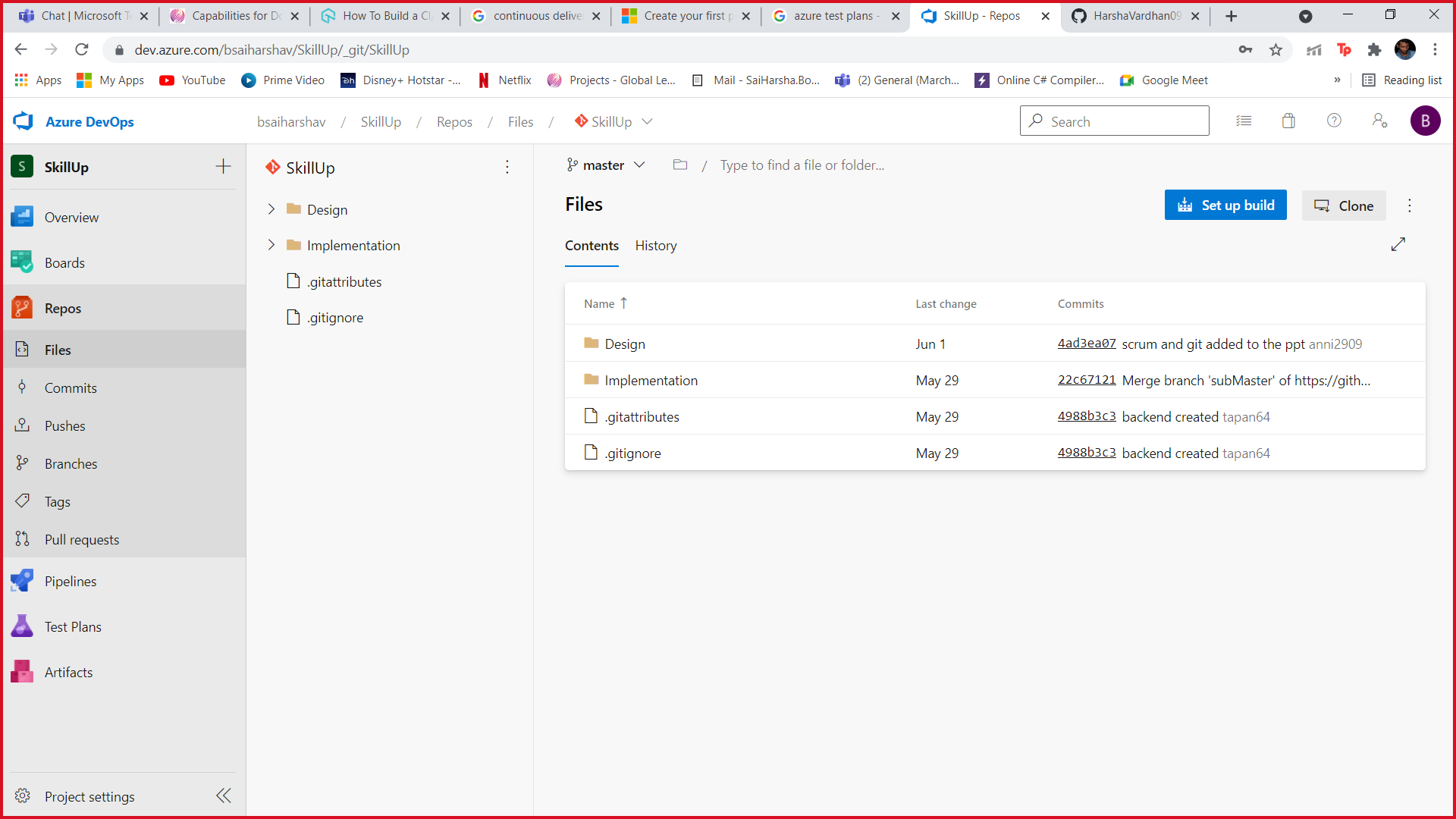
1. Overview Gives full detail description of project. Boards is similar to Scrum.

Repos are used control version. We can create Azure repo using Visual studio or we can directly import it from github, bitbucket, gitlab, etc

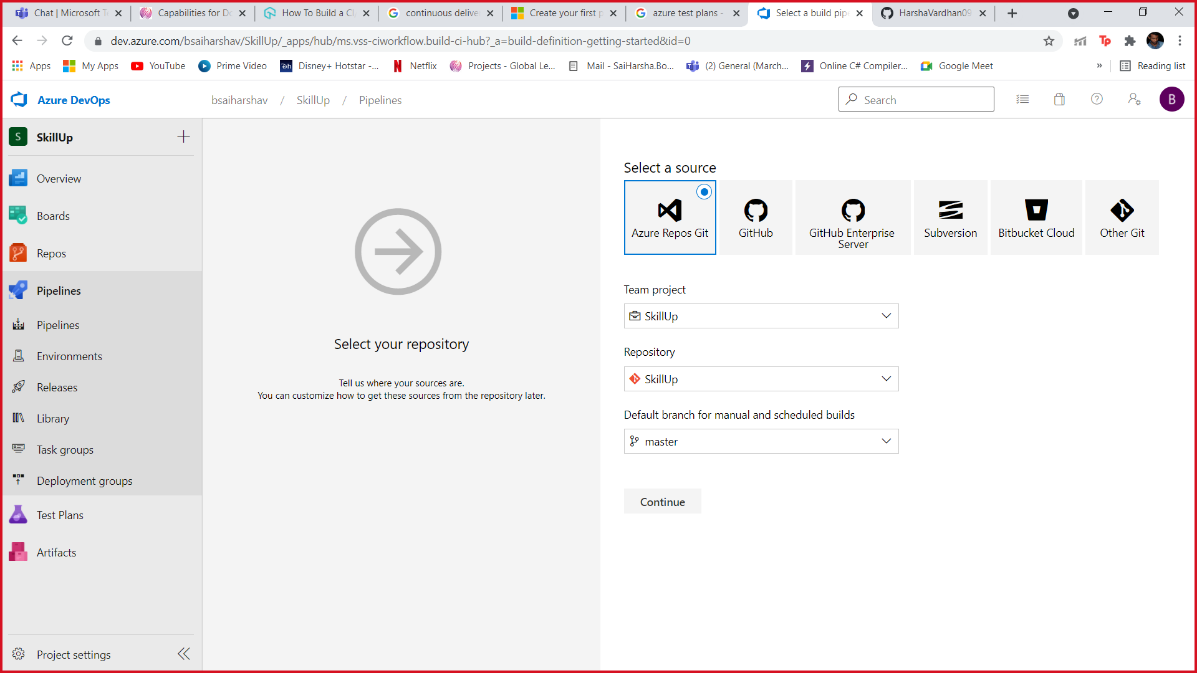
Here we will use direct import from github.



1. Add repository from github.

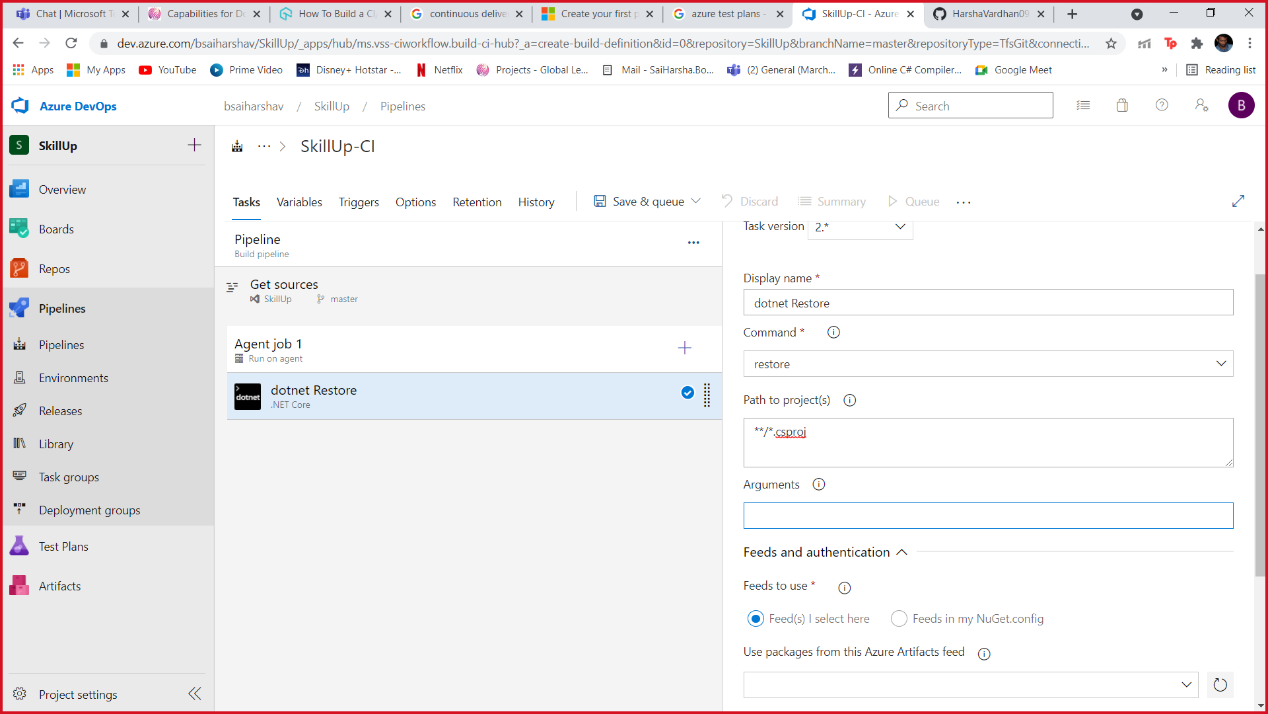


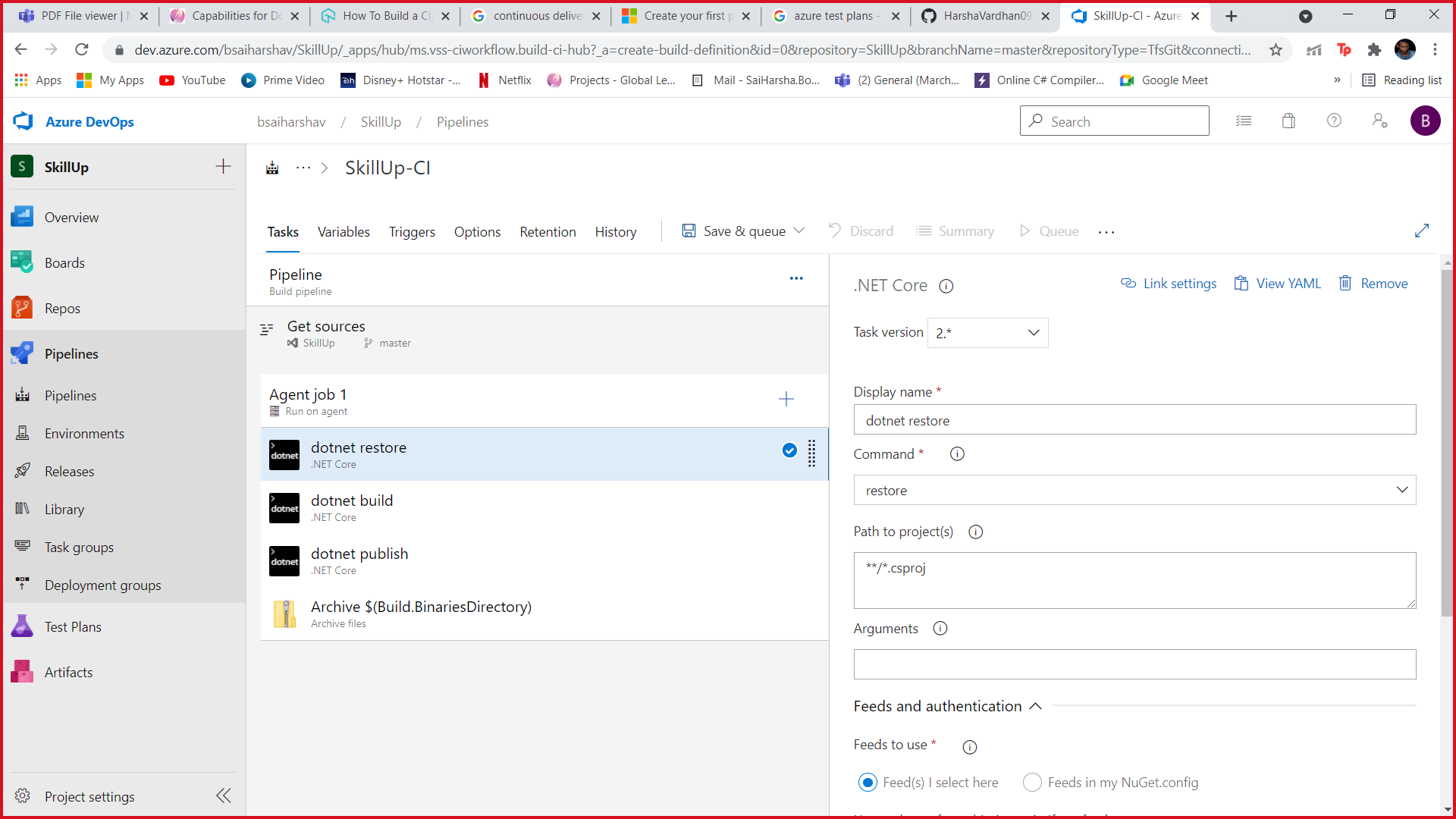
1. After that we have to automate build so for that we have to create Pipeline for our project. So, click on create Pipeline then go to Use the classic editor. Then Select Azure repo which we created earlier.



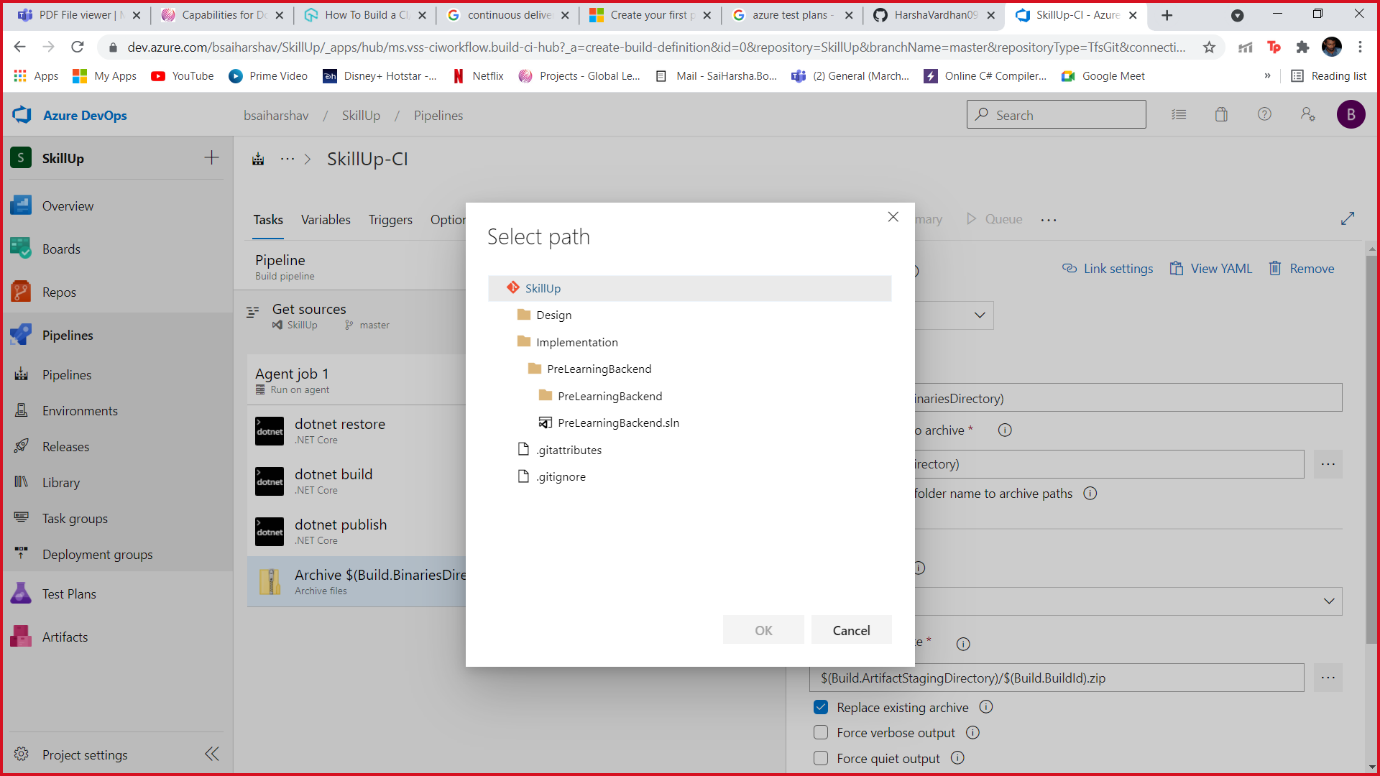
1. Then Click Continue. Then select Empty Load. Then Click to add agent load in that select dotnet core and add it. Edit it to restore and select it.

Again, click on add and Select Dot net Core and add build. Repeat the same procedure again and add publish.

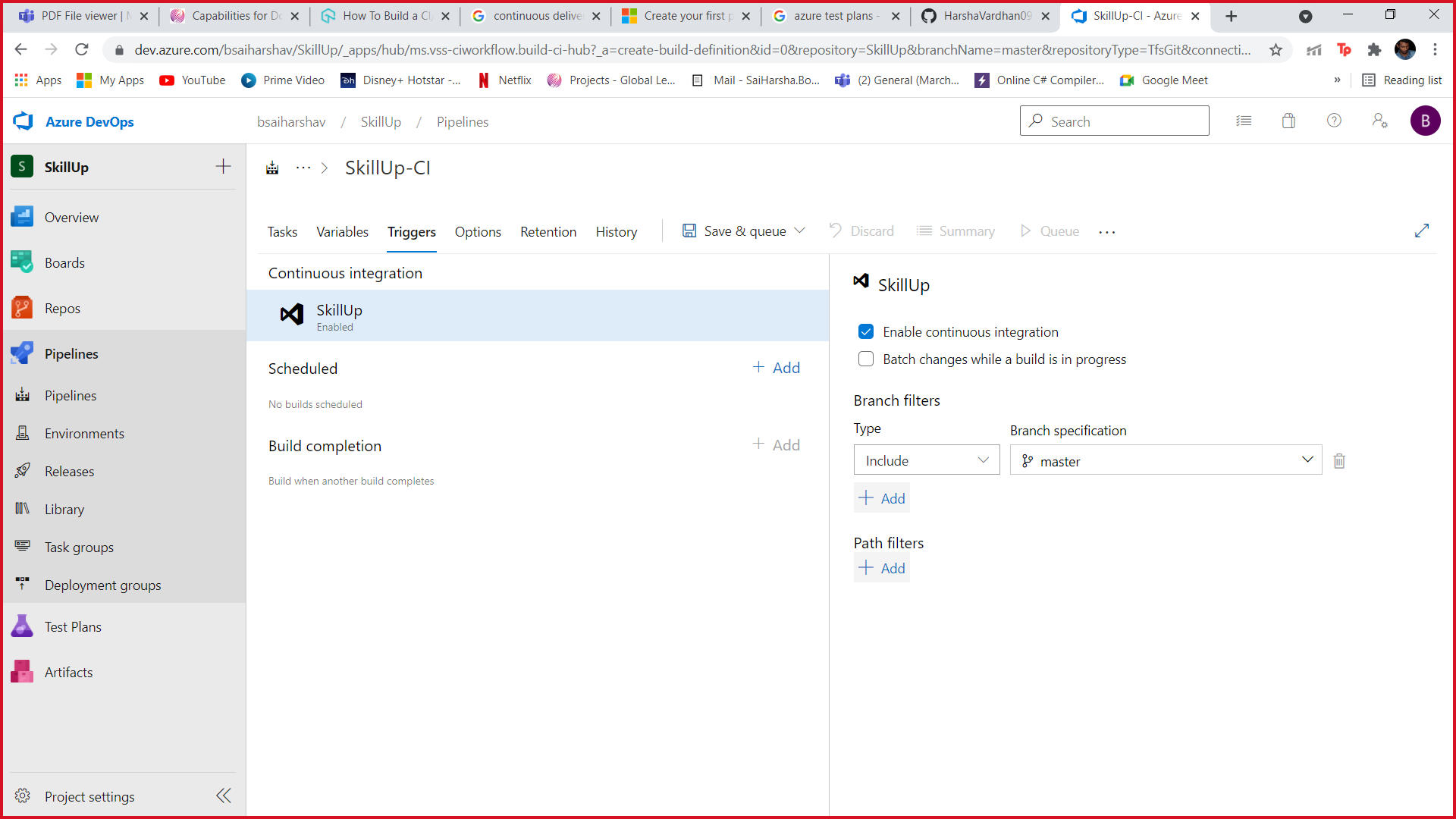




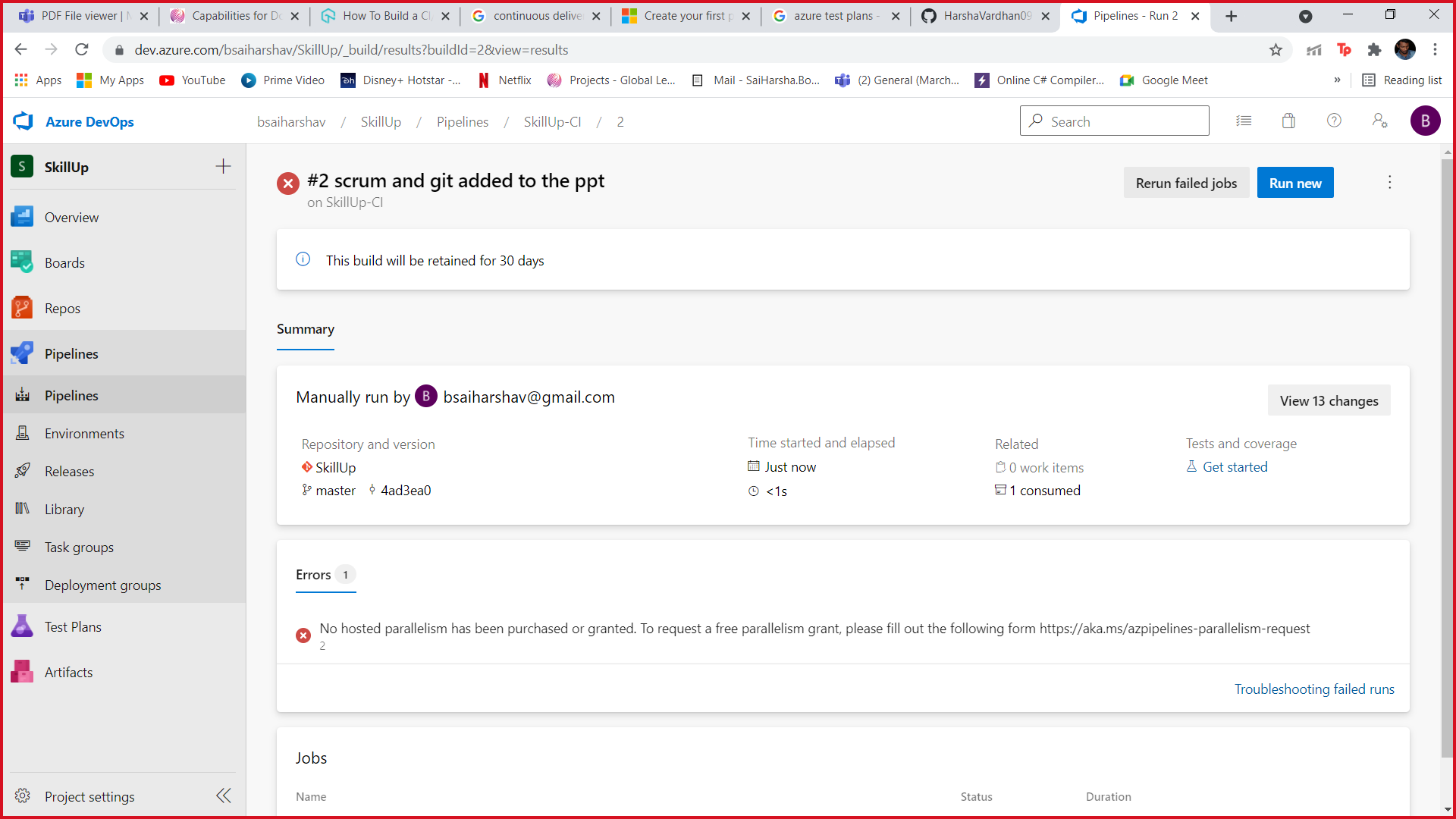
1. Then Search Archive and add it. And deselect publish checkbox. Then add Publish path.



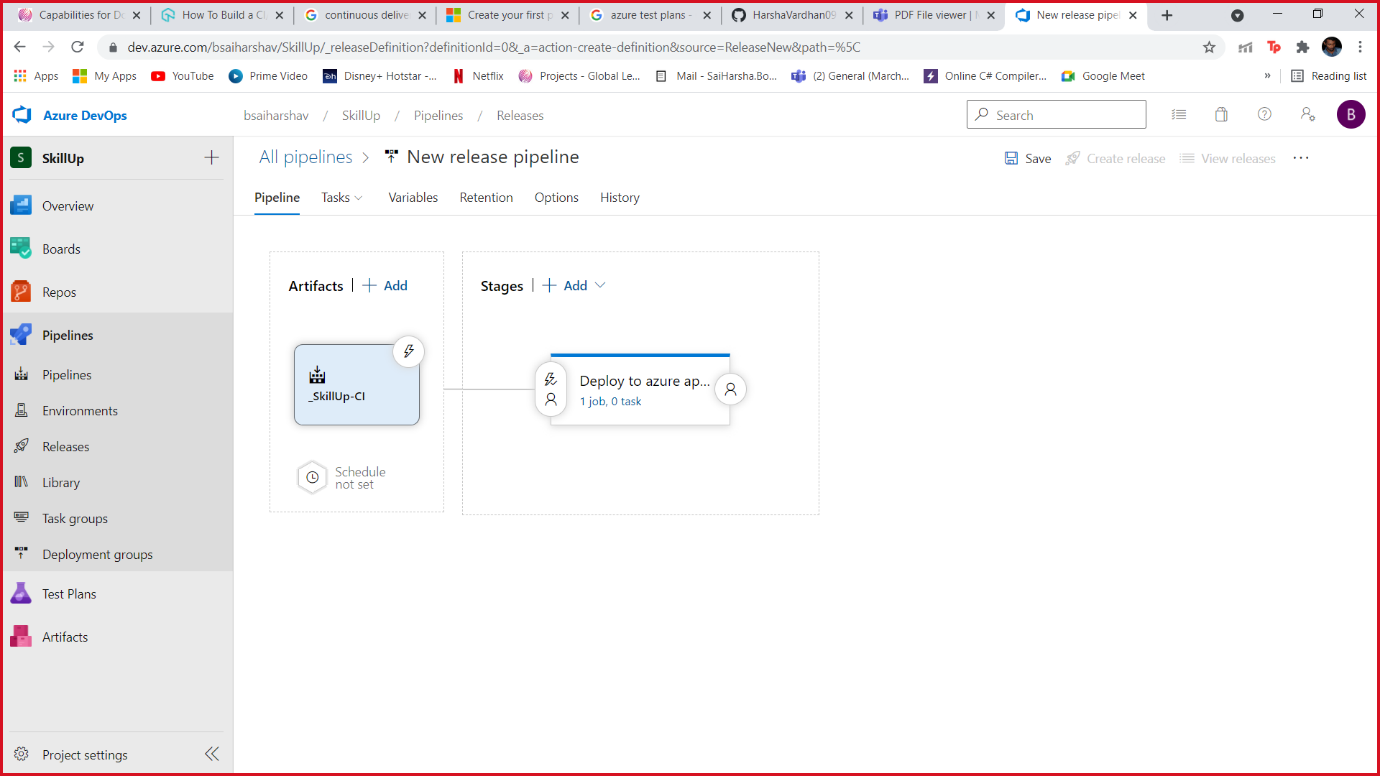
1. After that enable continuous integration by click on Triggers.

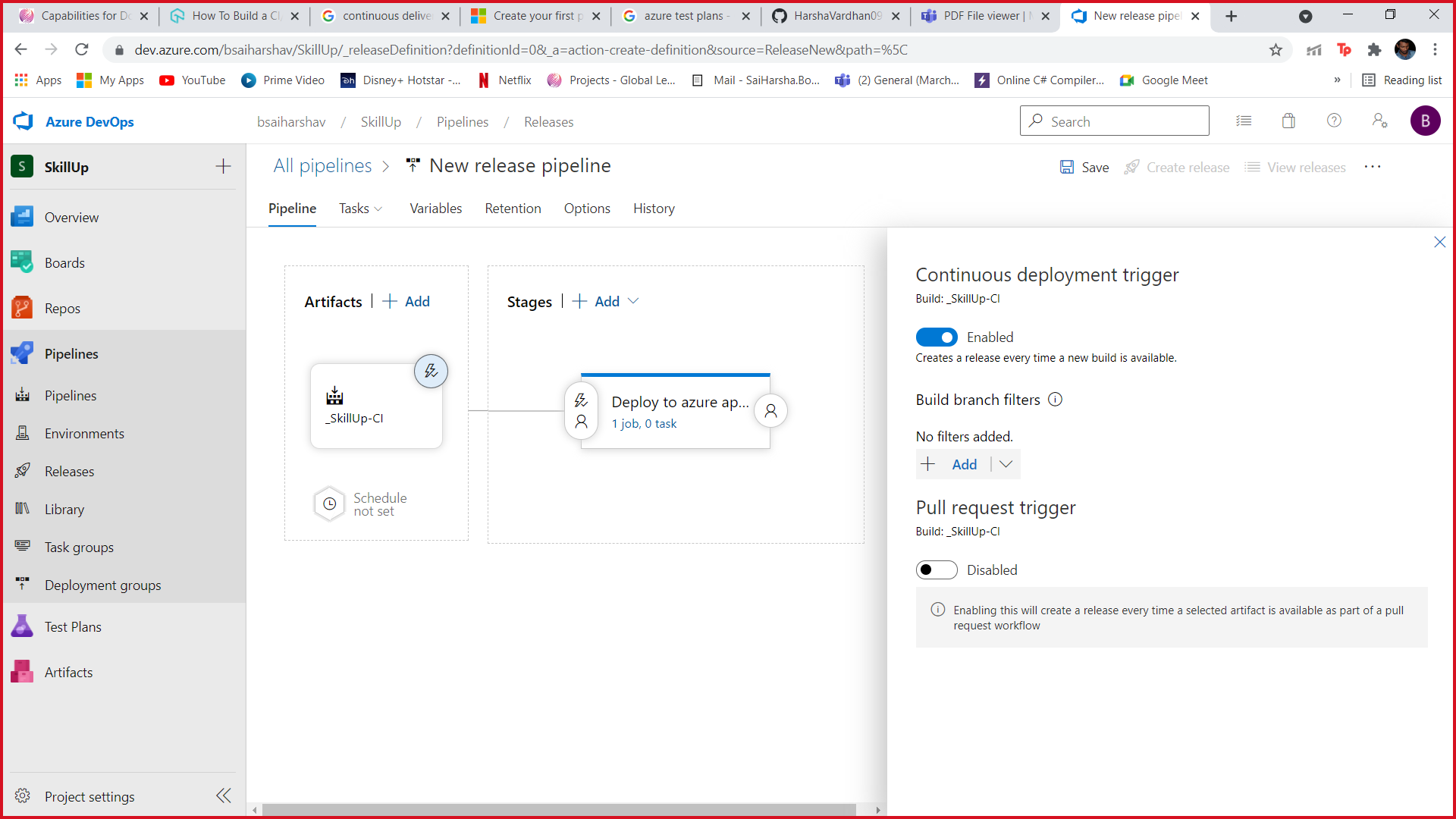
****

1. Then Click save and queue to create pipeline. After build of pipeline then go to release and publish the pipeline using azure app service.

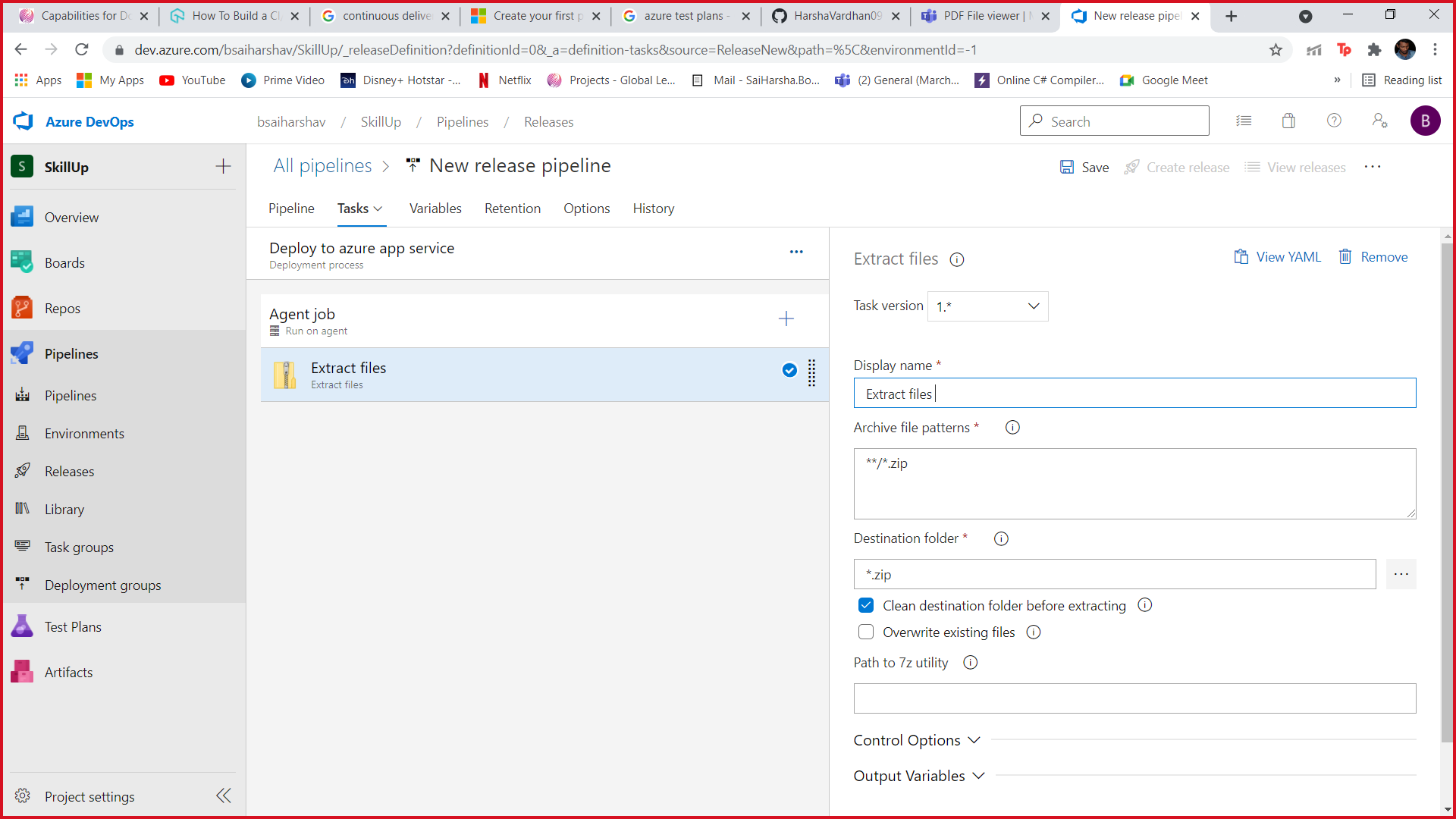


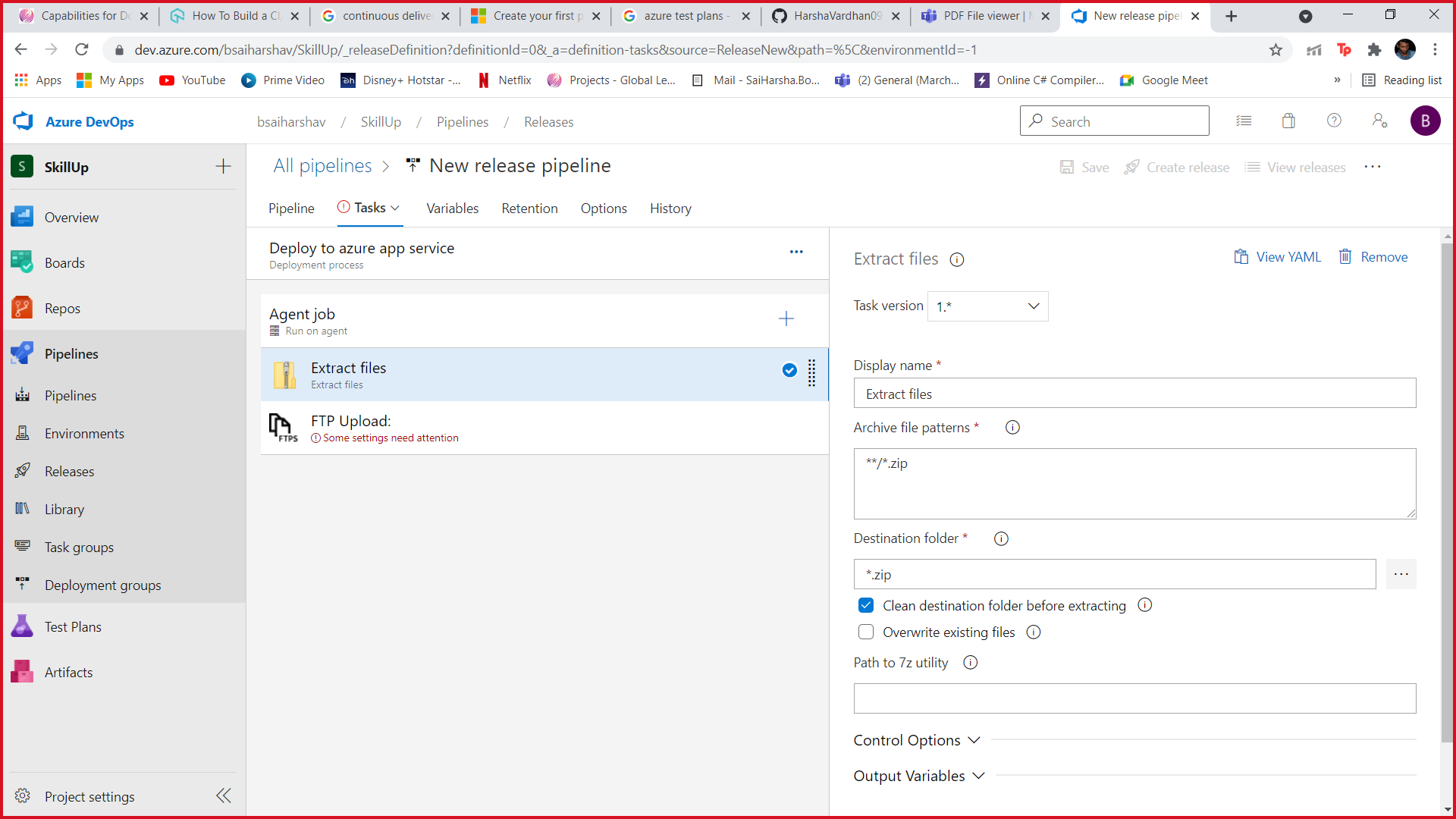
1. Add a Stage to deploy and also add Artifacts





1. Chick on Task in stages





1. Before Releasing we can write Test Plans in test plan section.