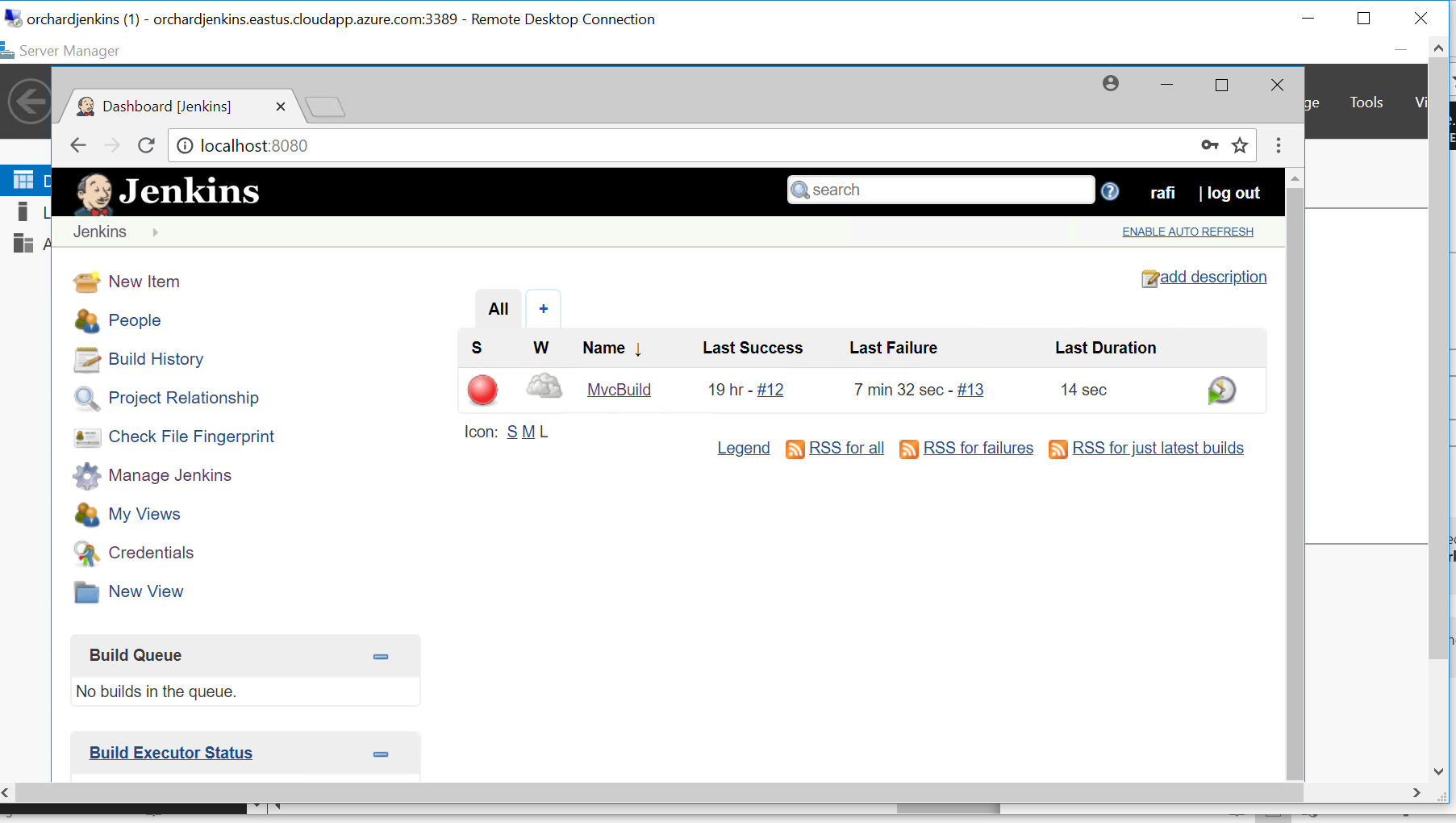
Setting Jenkins server for Microsoft environment with Azure app service integration

*Setup Jenkin Server*

* Install the Jenkins server
* Create a windows VM in Azure portal and connect to the same using RDP
* Download Jenkins installer for Windows from <https://jenkins.io/download/>
* Run the installer, on Successful installation Jenkin server can be accessed using <http://localhost:8080> on the VM



* Install necessary tools for Microsoft Environment
* To build the .Net applications we need MSbuild tool on Jenkins Server
* Get the installer from <https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2017>
* Run the installer on Jenkin Server VM.
* After successful installation you must be able to see the Msbuild.exe under this location

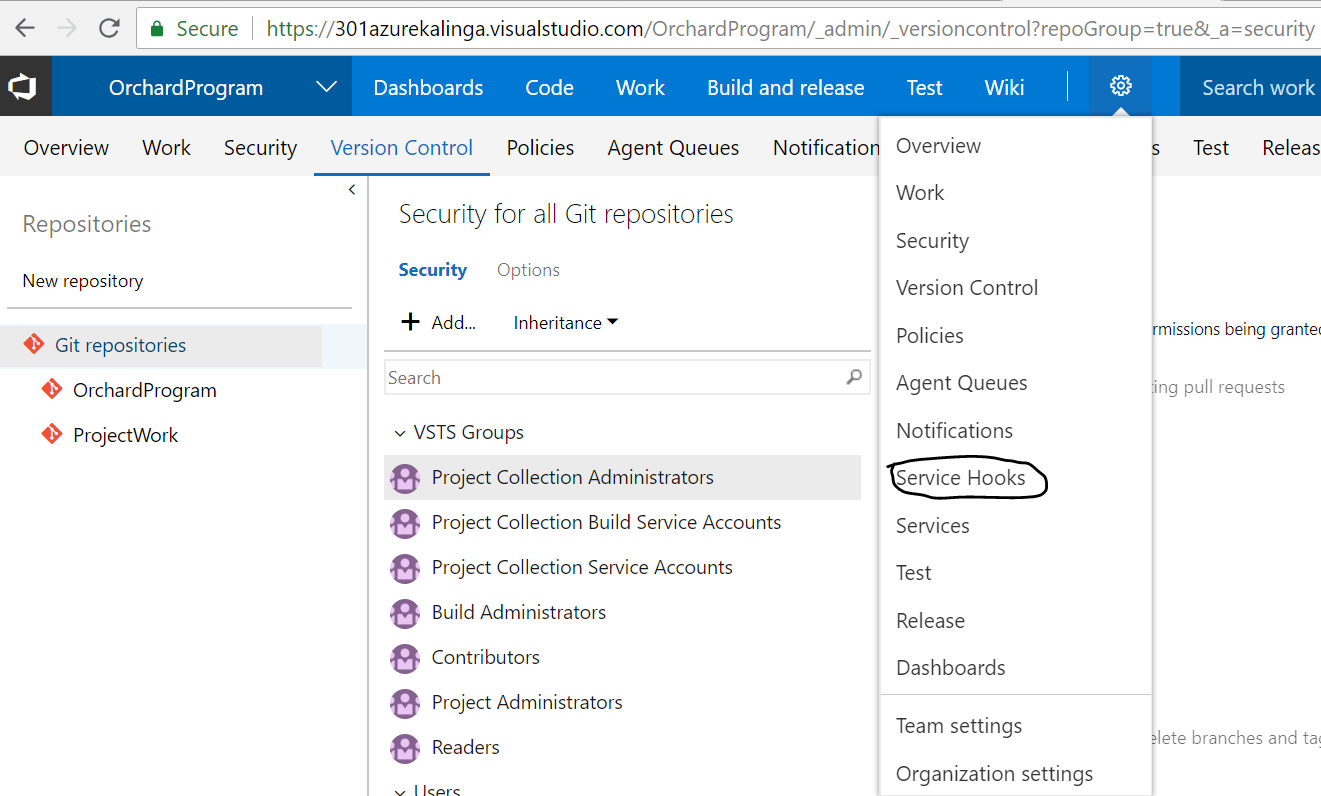
C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\MSBuild\15.0\Bin\

* To resolve the Nuget dependency we need the Nuget.exe on Jenkins server
* Get the Nuget.exe from <https://docs.microsoft.com/en-us/nuget/install-nuget-client-tools>
* Place it in a custom folder on Jenkins folder (ex: c:/Nuget/Nuget.exe)
* Configure tools in Jenkin Server

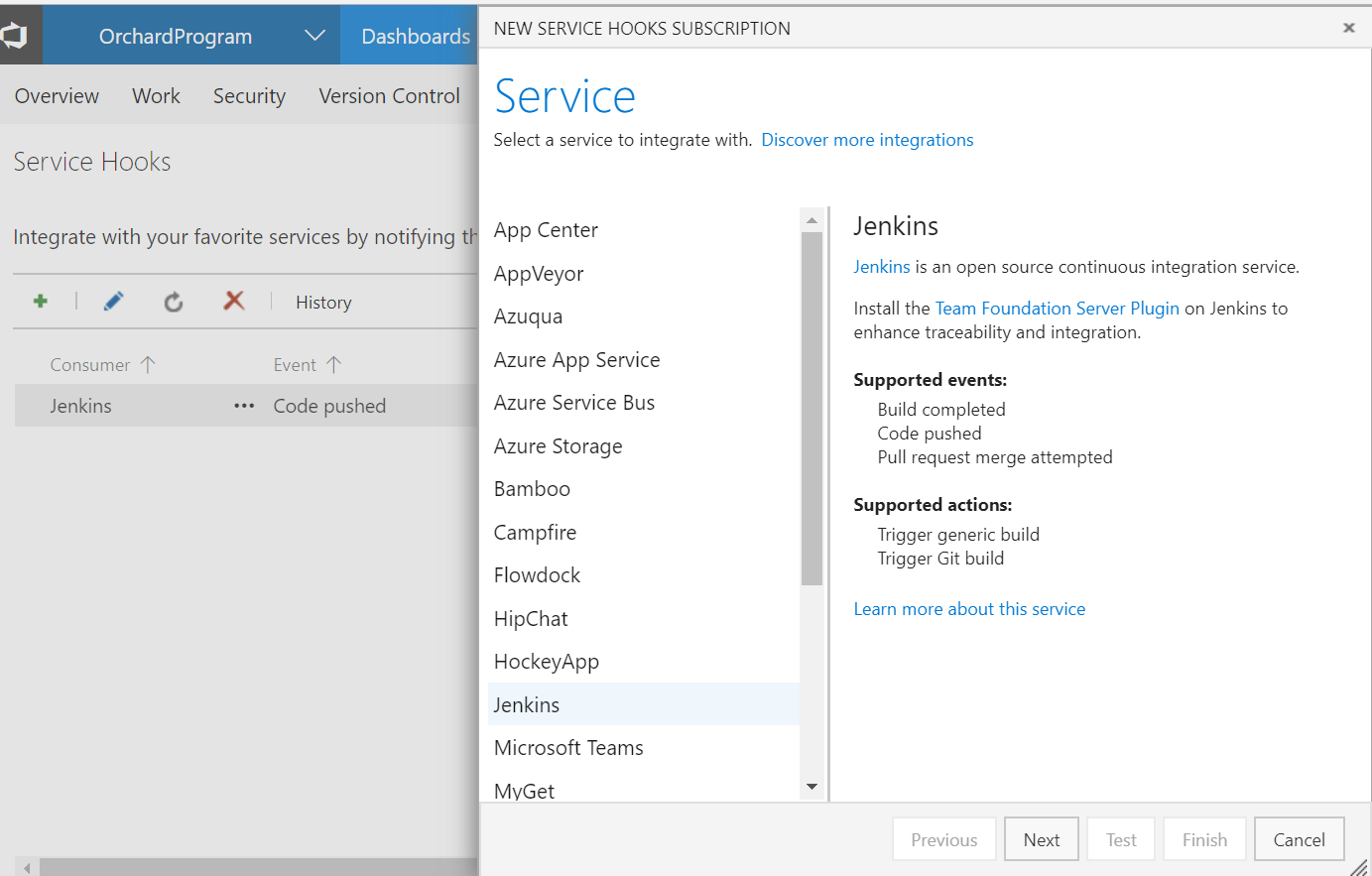
*VSTS integration with Jenkins*

We are using Git for our code repository in VSTS, Jenkins CI pipeline fetches the code from VSTS Git repository and build the code. If we want to trigger the Jenkins build automatically when there is a new code push to the VSTS Git repository, we need to create a service hook in VSTS

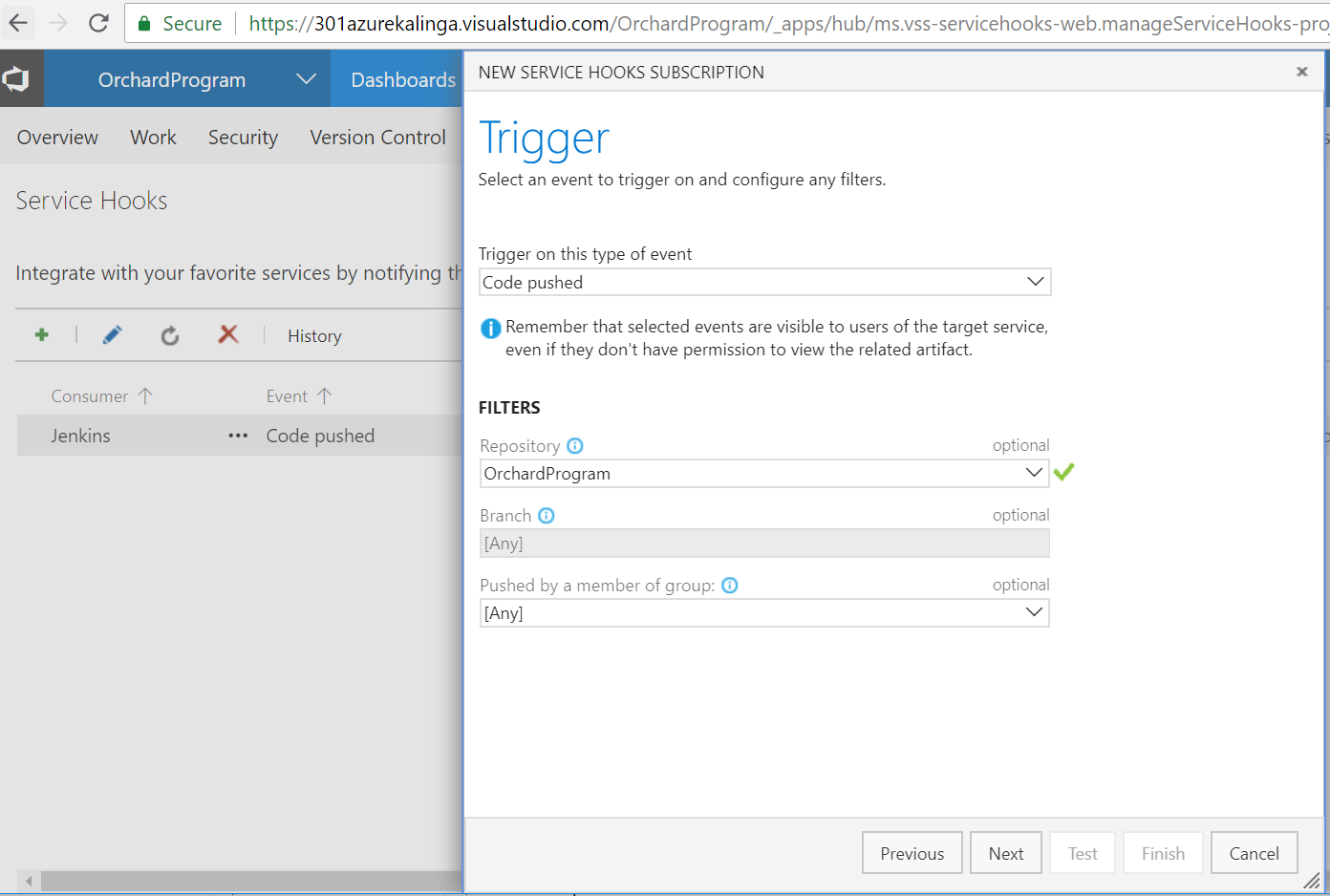
* Go to your VSTS and under settings tab click Service Hooks



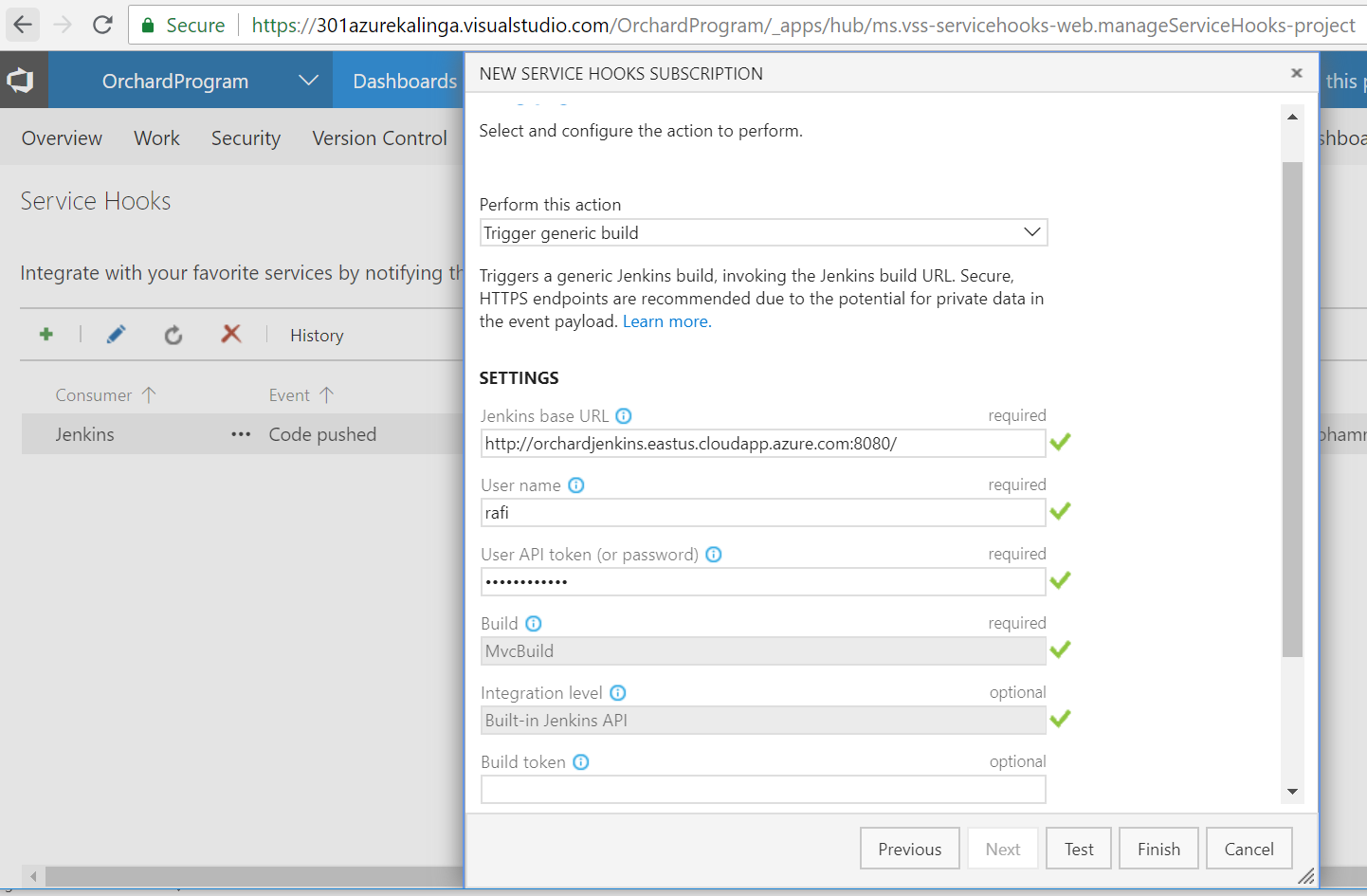
* Create a new subscription



Choose Trigger event type as “Code pushed” and your repository



Configure Jenkin server details

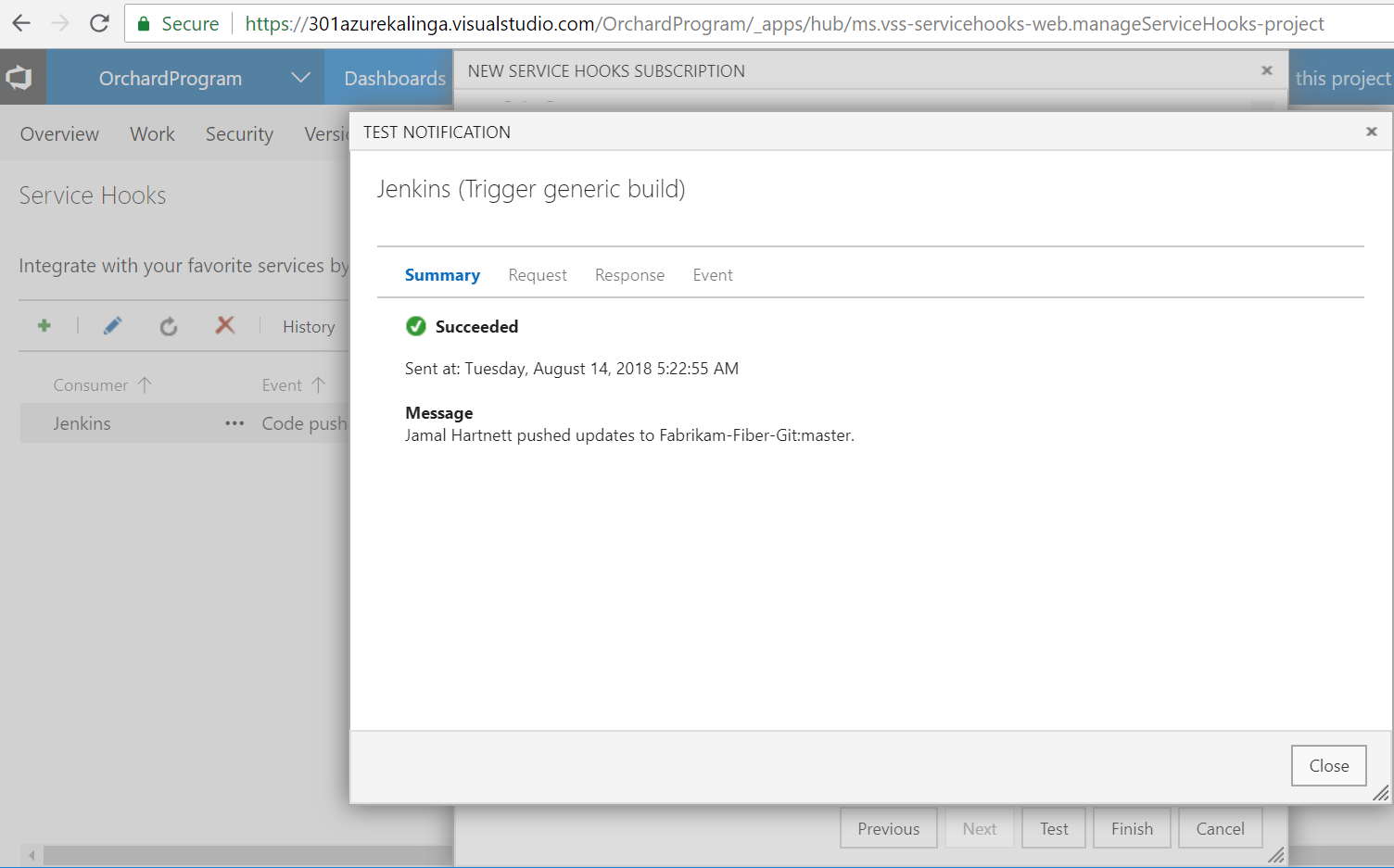


Jenkins Base URL: this is the URL of the Jenkins server

User Name: Jenkins User Name

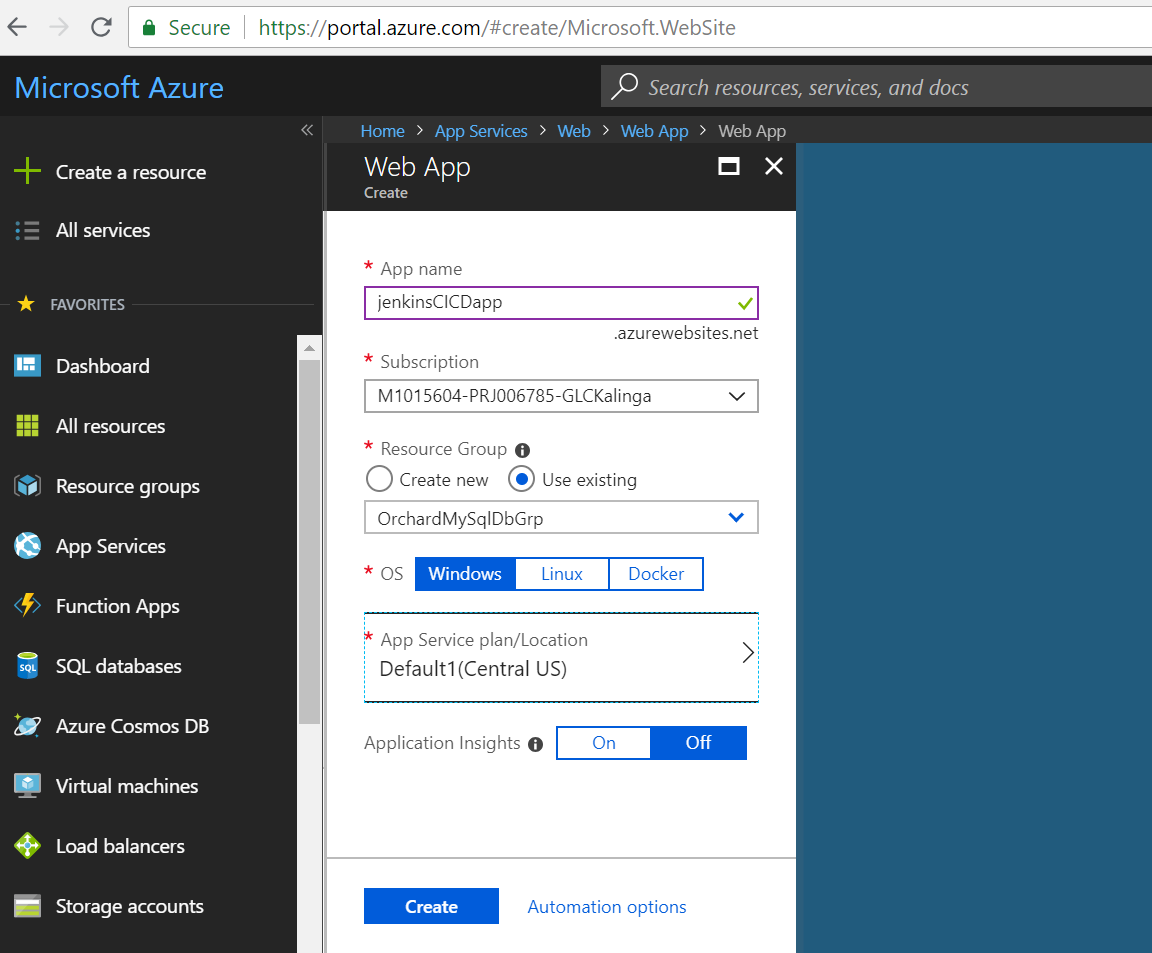
User API token (or password): Password of the User

Test the connectivity by clicking Test button, on success, should see below

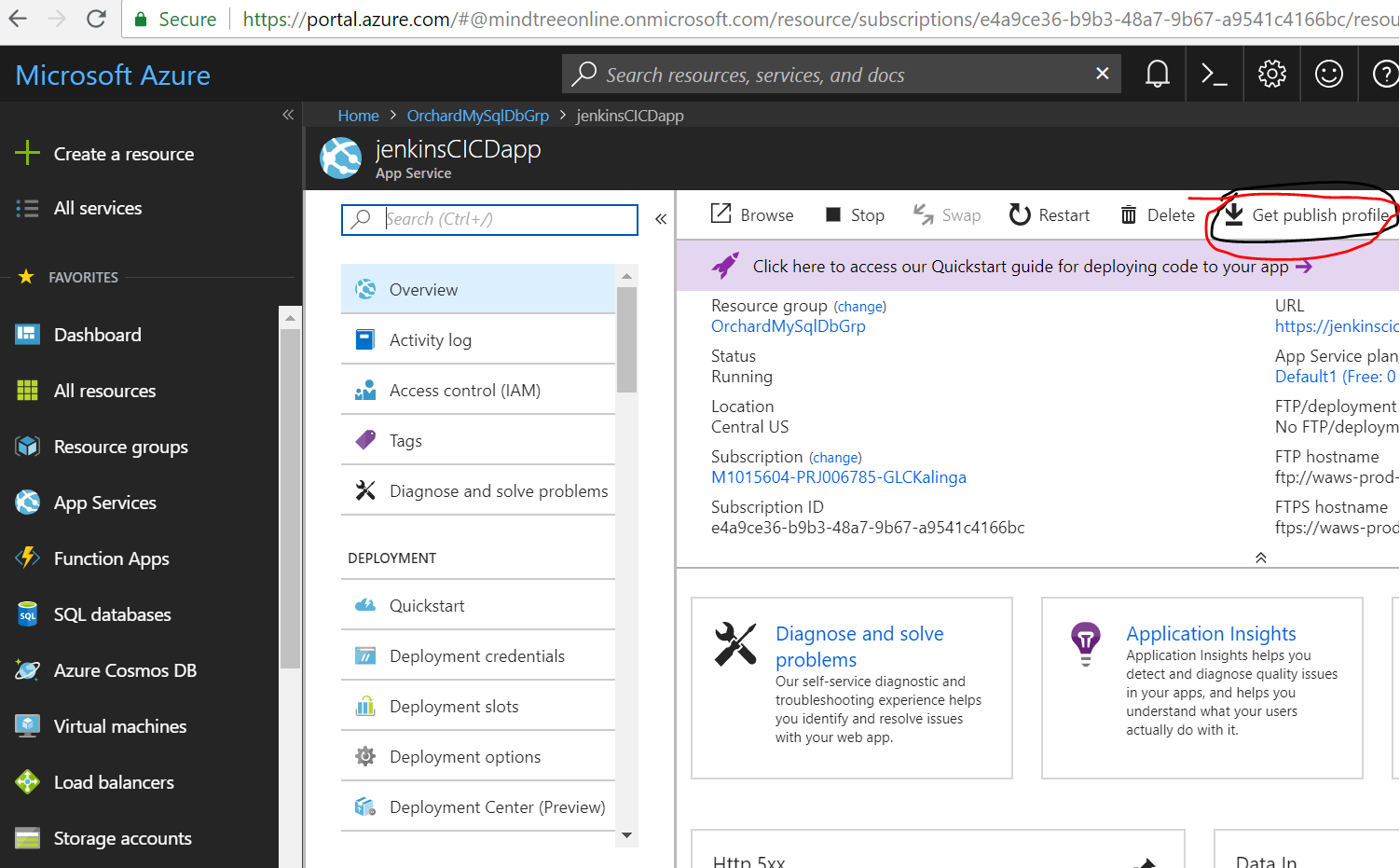


*Deploy .net application to AppServices*

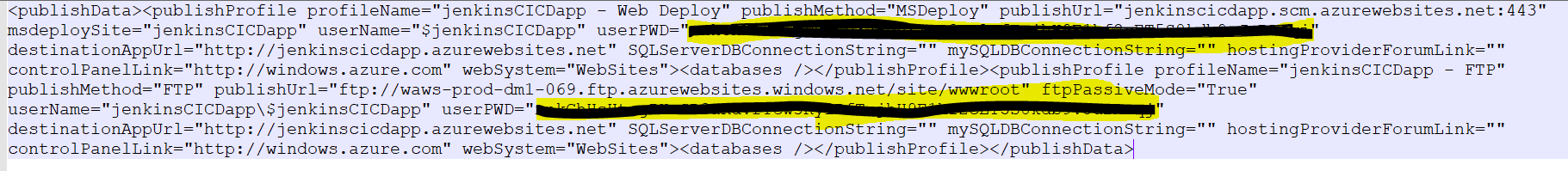
* Choose Create an App Service Web App



* Download the publish profile of the web app we created

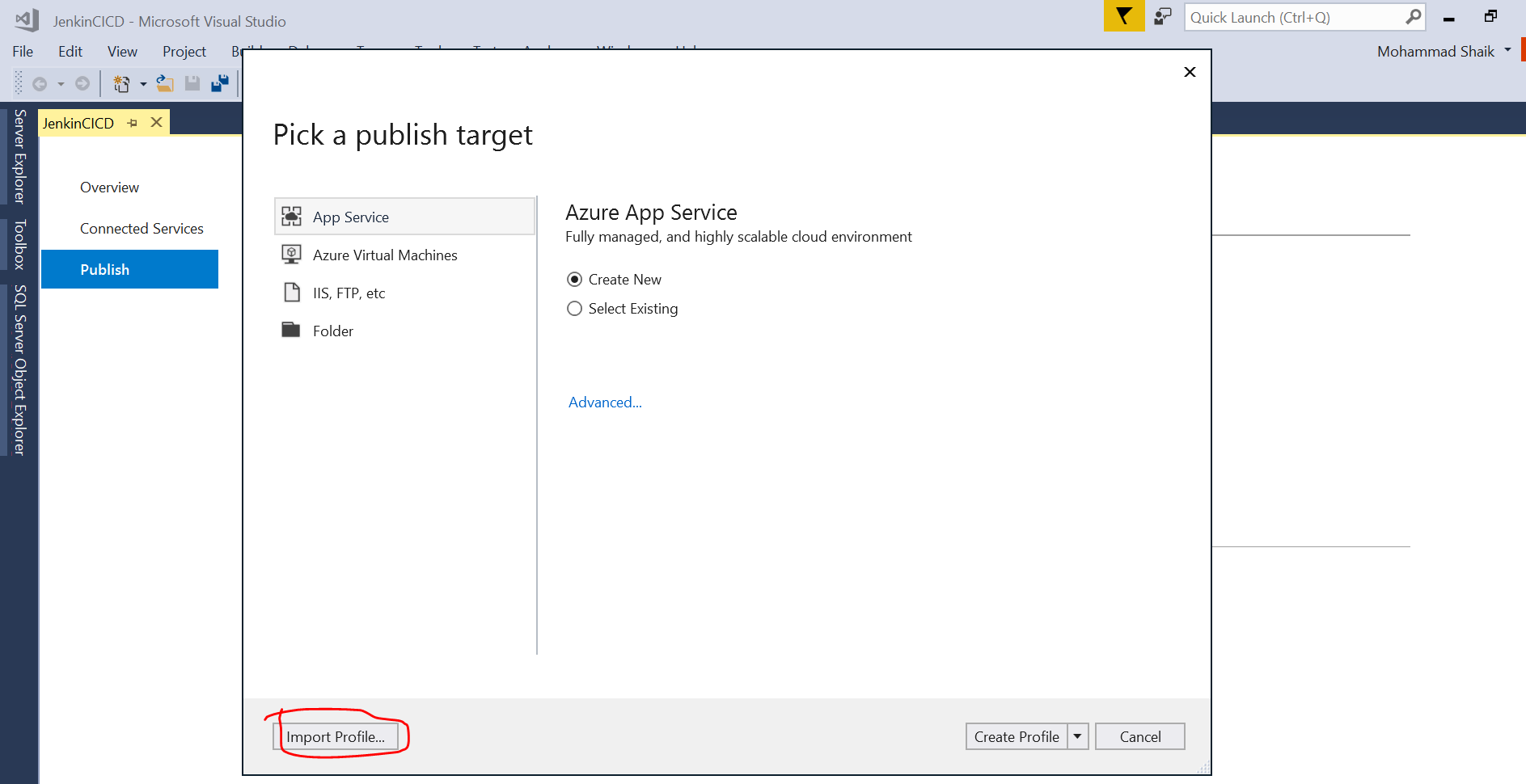


* Grab the User Password from the publish profile (We are going to use this password in the deployment process from Jenkins)

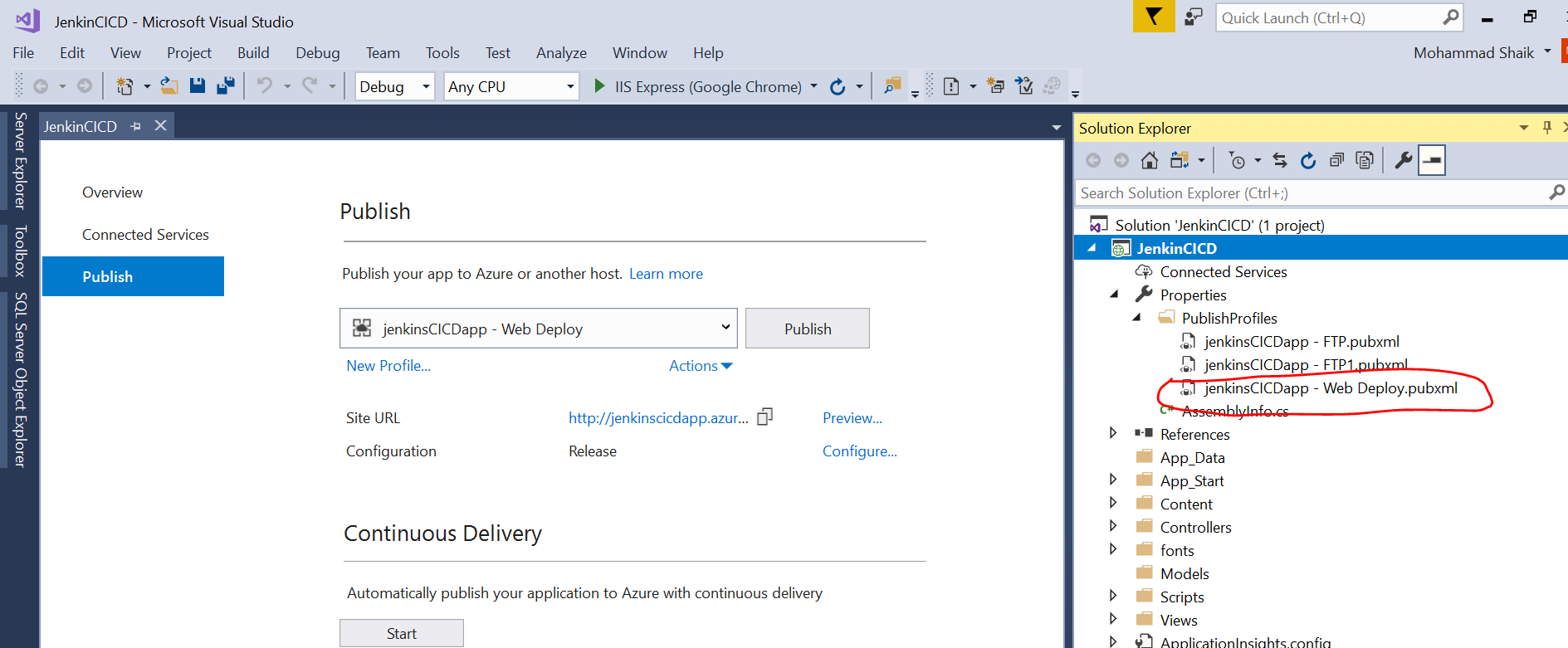


* Also create a .pubxml file from .publishsettings (publish profile ) file from visual studio as shown below

Go to create new publish profile window , and import the .publishsettings file from above step

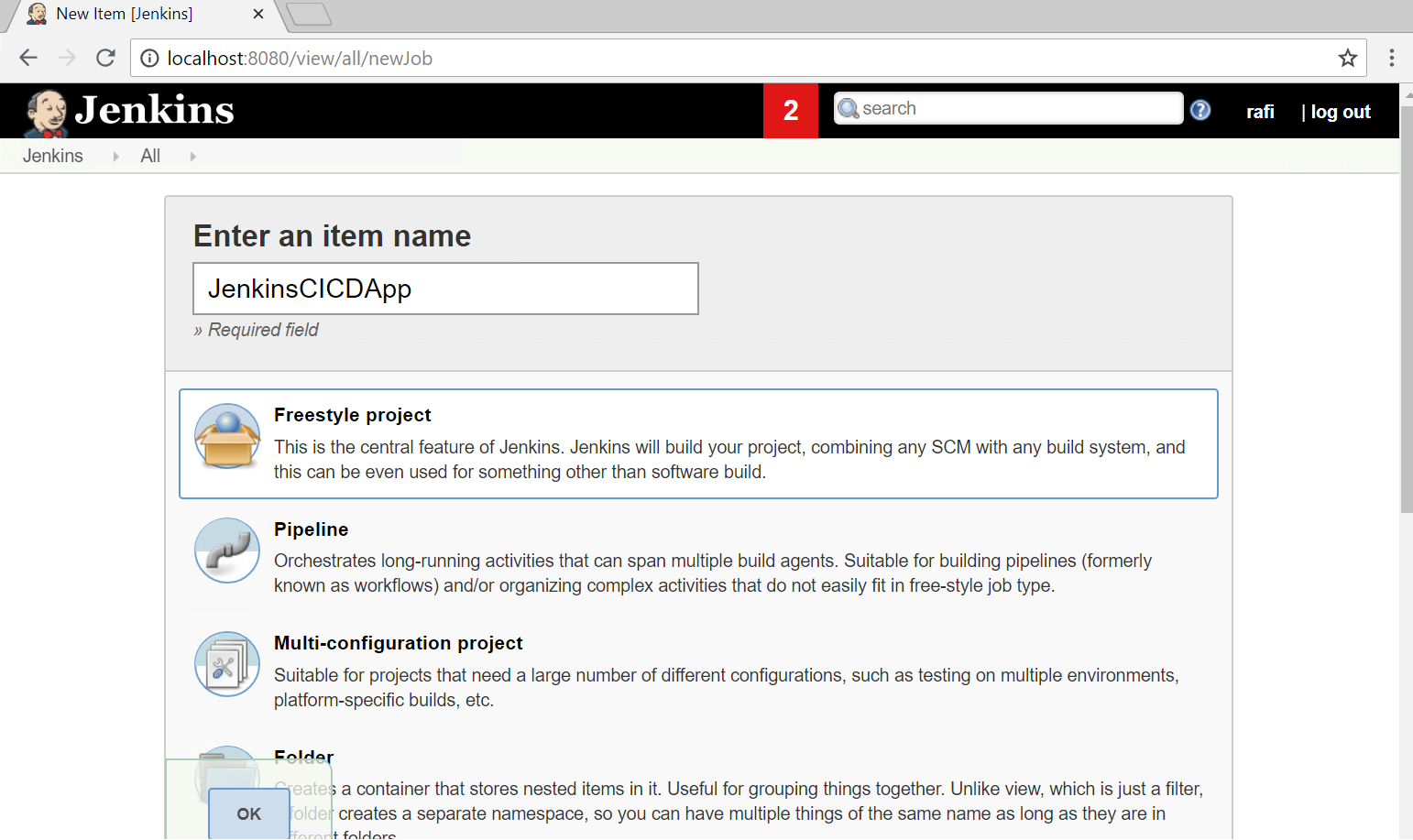


After import it creates .pubxml file in the solution

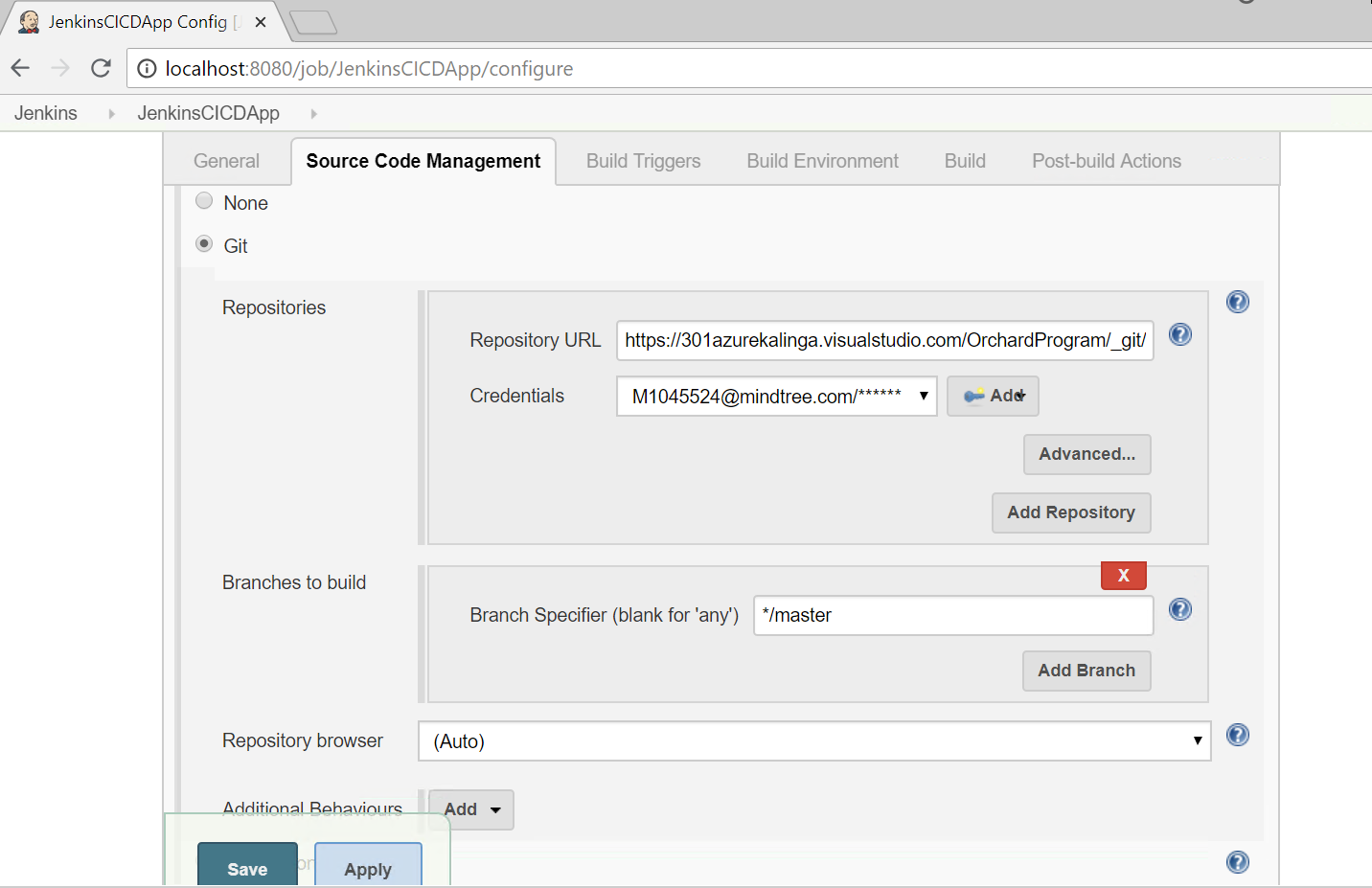


Copy this file to some custome folder on the jenkins server (VM), example location “c:\AzurePublicProfiles\” (we are going to use this in the deployment process from the Jenkins)

* Create a Free Style Project in Jenkins

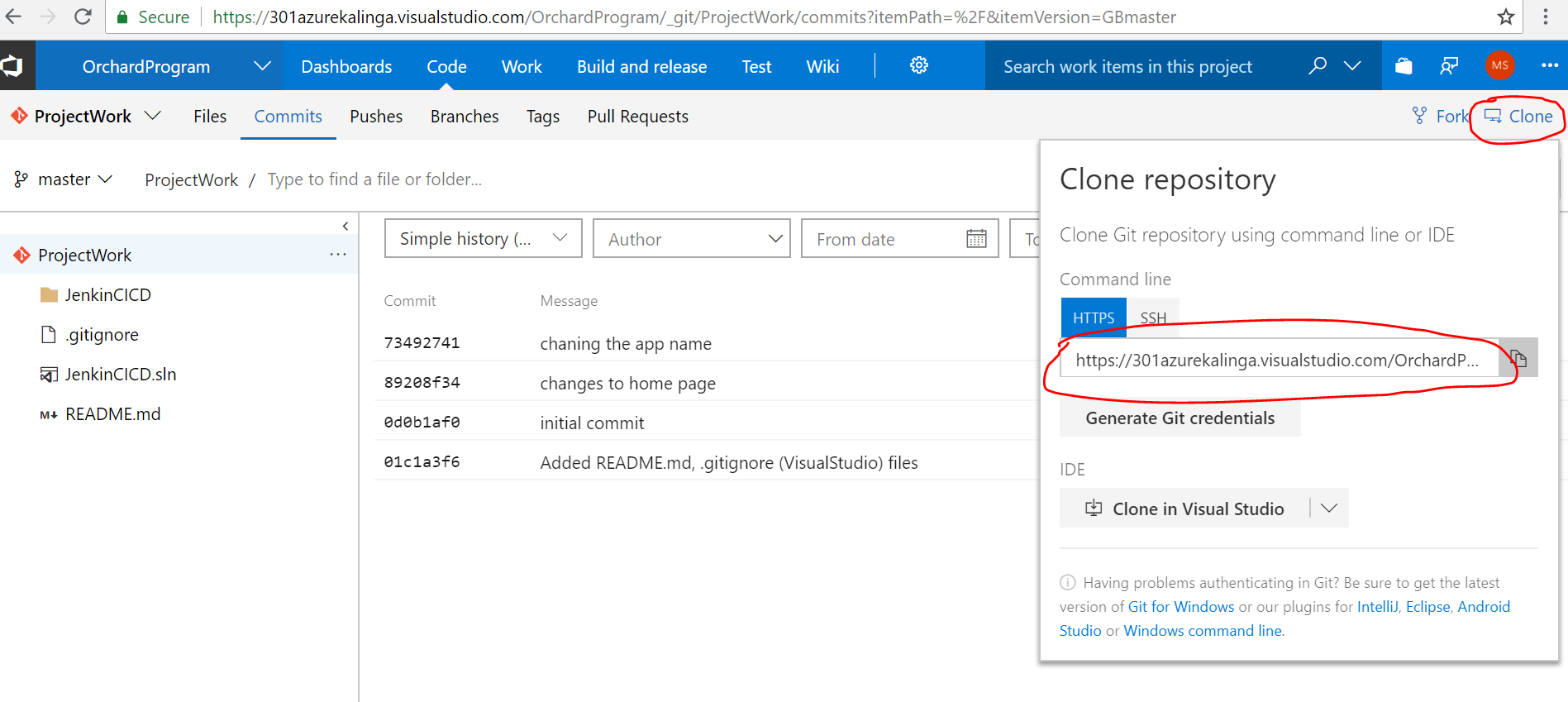


* Configure source code management under freestyle project that we created in the above step

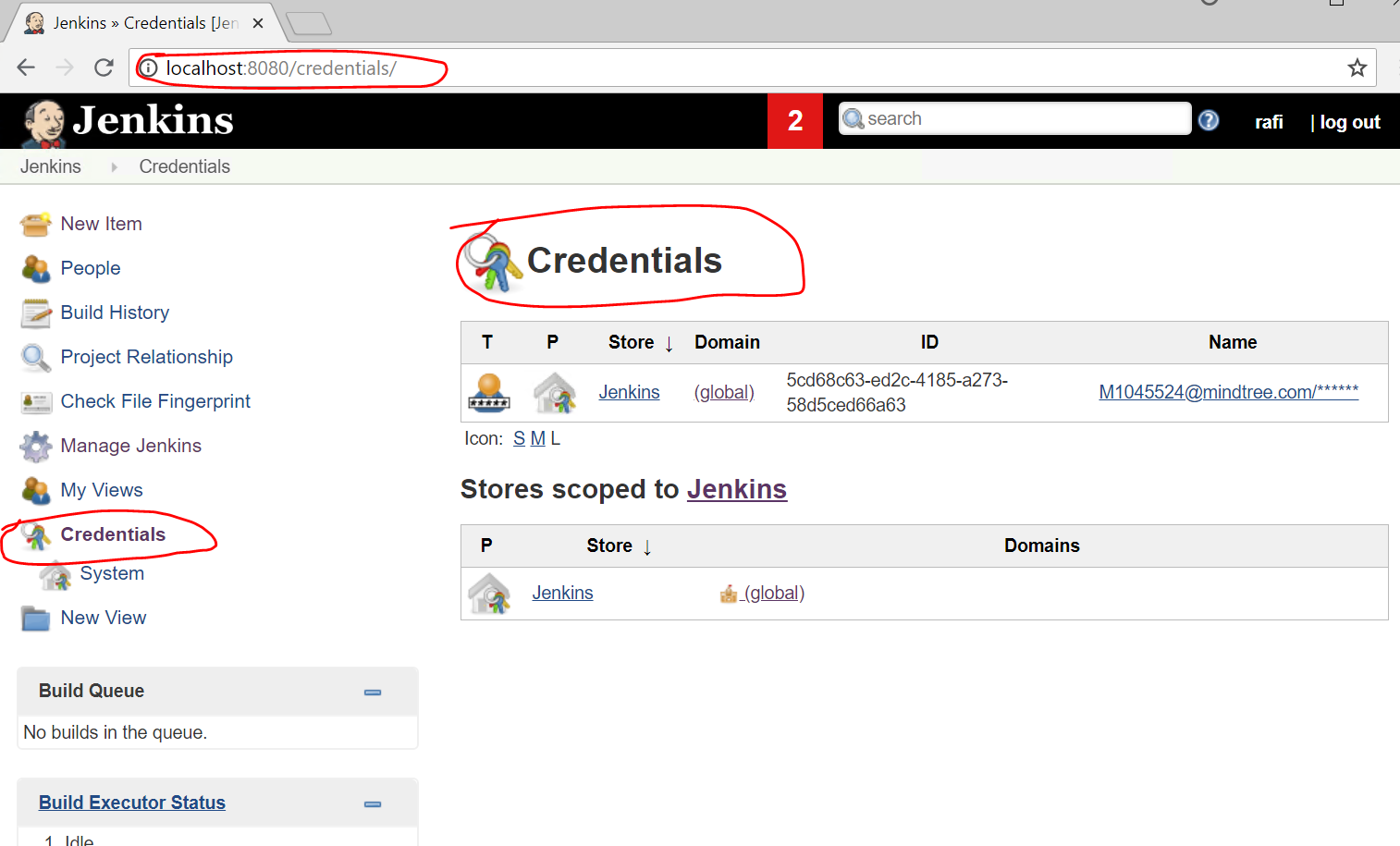


Repository URL : your VSTS git repository URL (ex : https://301azurekalinga.visualstudio.com/OrchardProgram/\_git/ProjectWork)

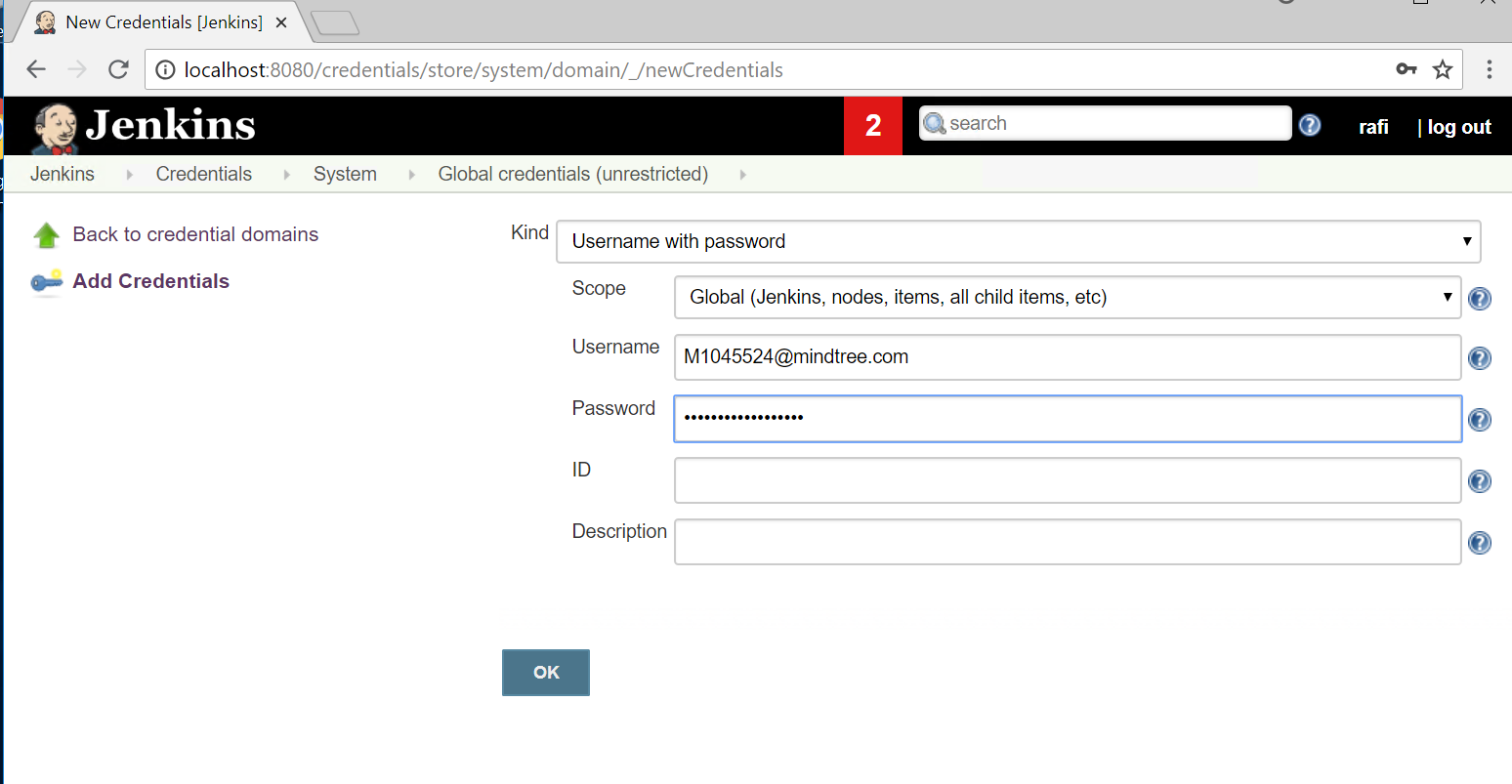
You can get it from below VSTS screen



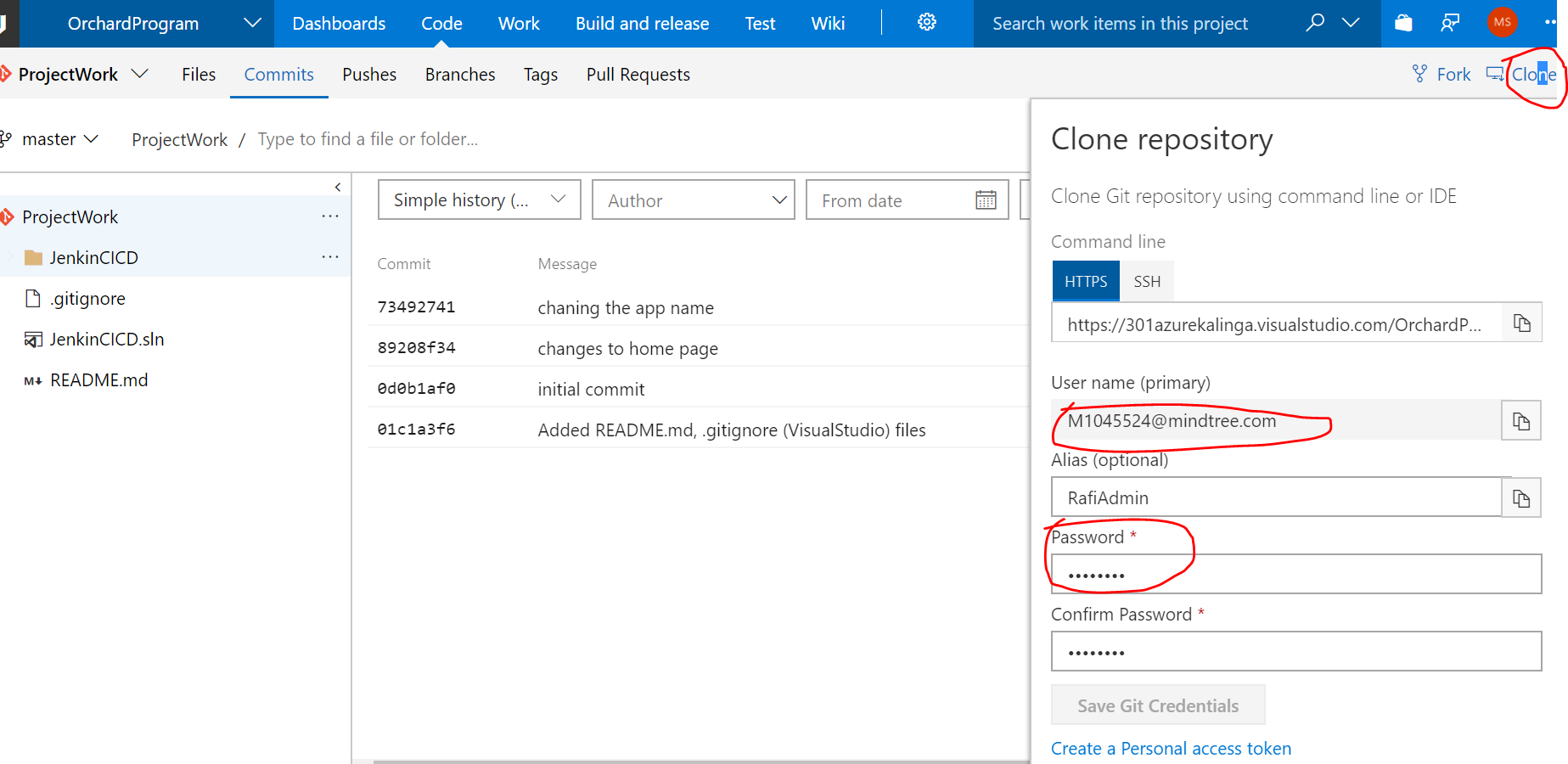
Credentials: you should have created it already in the Jenkins Credentials section



You can create a new one from below jenkin screen



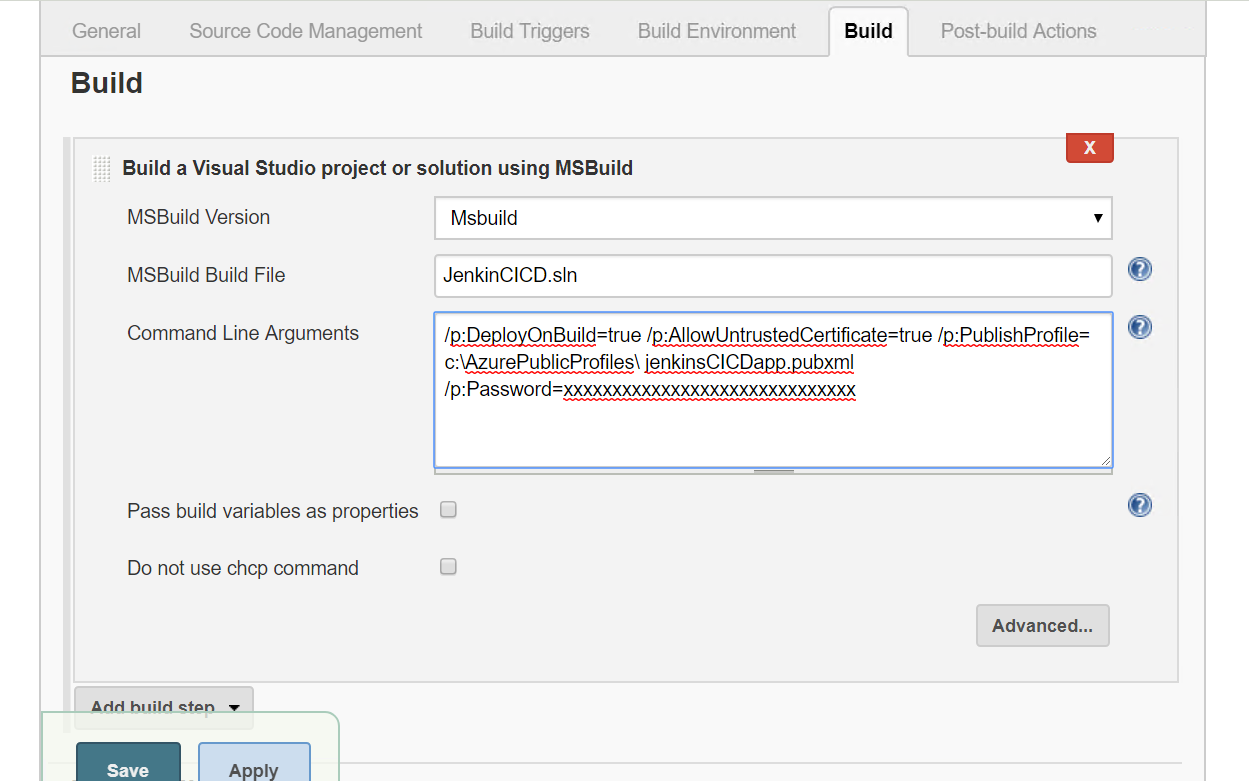
And the user name and the password in the above screen have to be the git credentials you created for your repository in VSTS as shown below



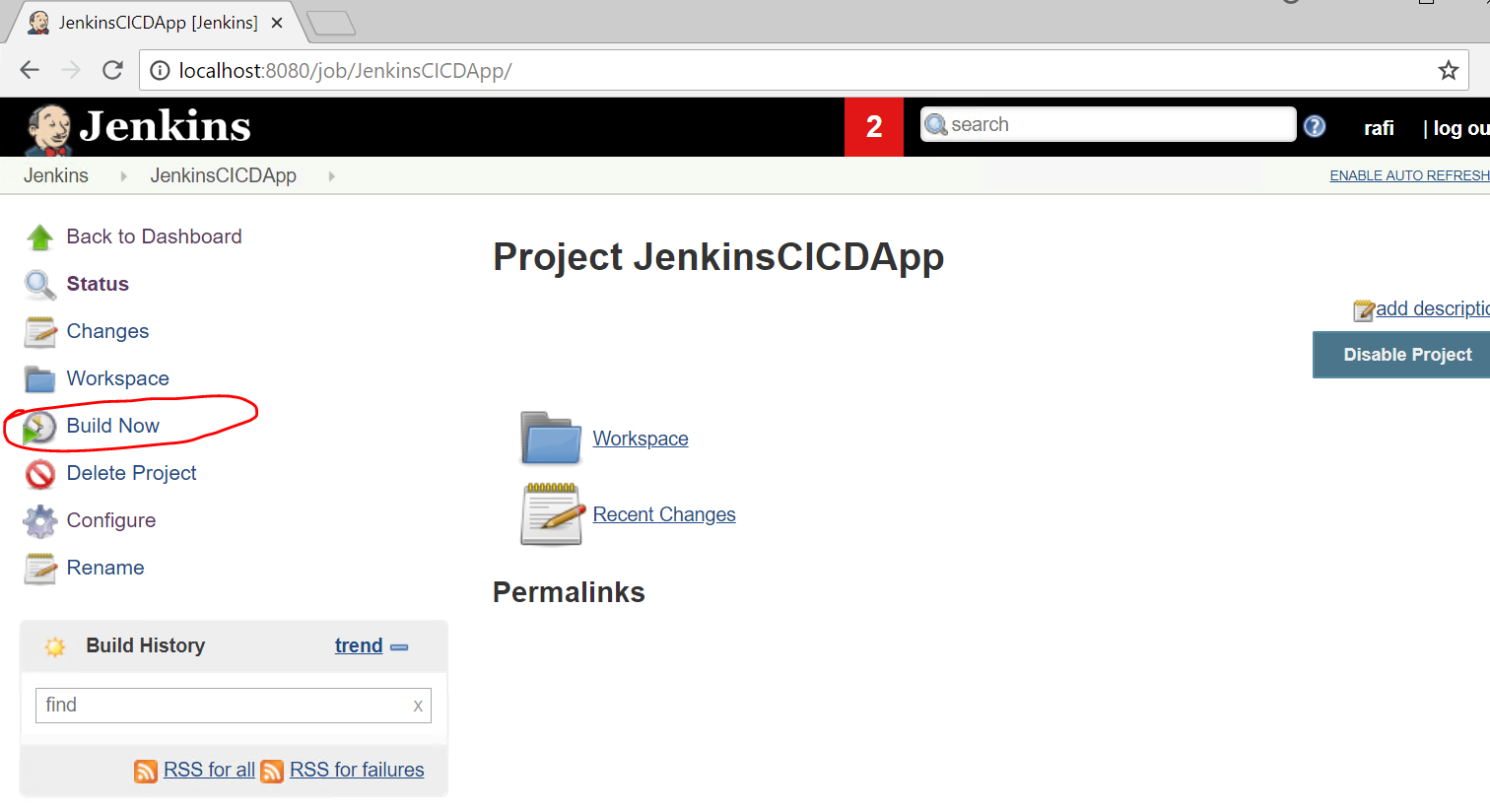
With this, we have done with source control management configuration.

* Now Go to Build Section, Add a build step “Build a Visual Studio Project or solution using MSBuild”. Fill in the “MsBuild Build file” for your sln file location
* Fill in the command line arguments for publishing the application to Azure App Services as below and save the project

/p:DeployOnBuild=true /p:AllowUntrustedCertificate=true /p:PublishProfile= c:\AzurePublicProfiles\ jenkinsCICDapp.pubxml /p:Password=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx



You can now run the build manually from the below screen



Or just push the new changes to the VSTS Git repository and it will aquatically trigger the build