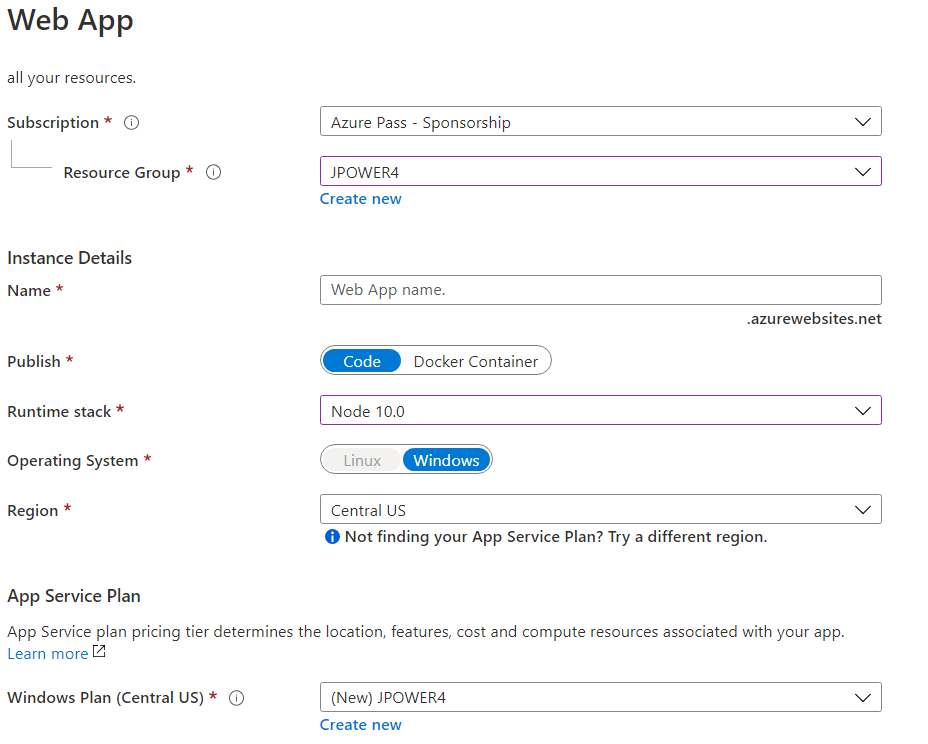
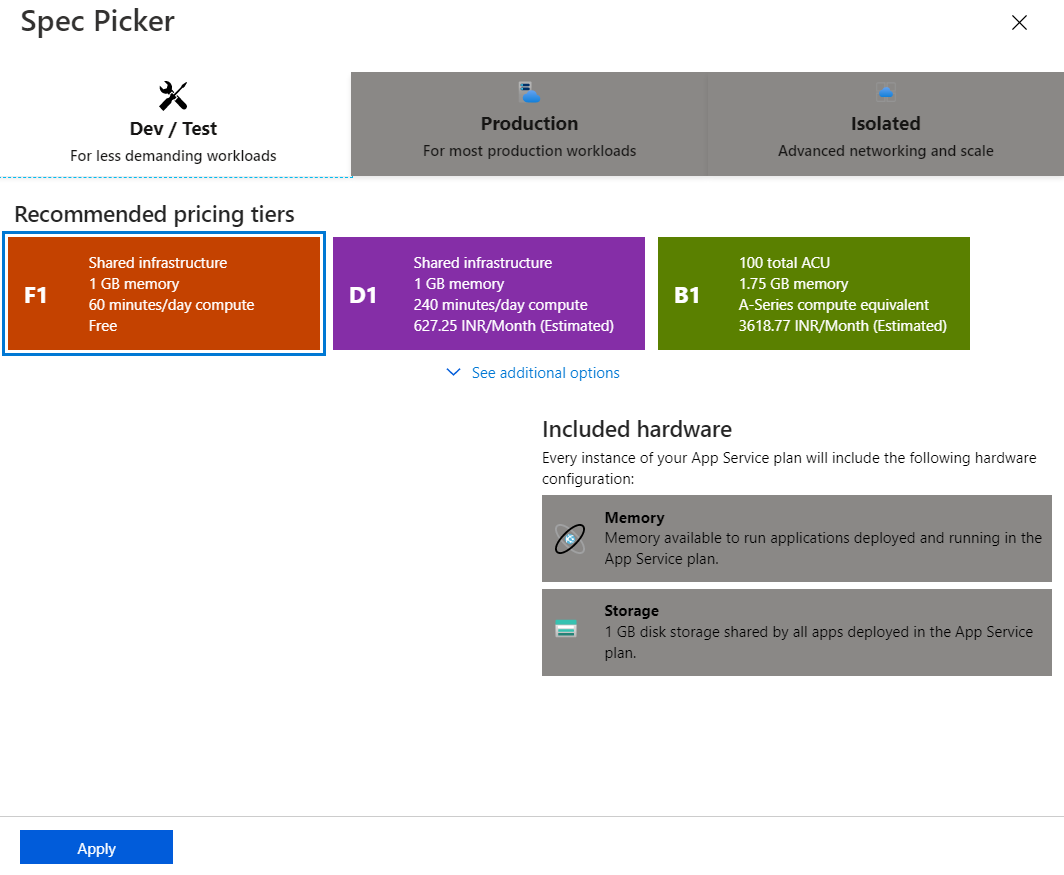
**Deploy Microsoft Teams App in Azure**

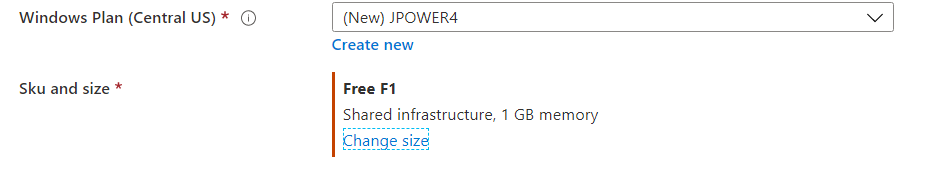
* Step 1: Create a Web App in Azure
* Step 2: Create a repository in GitHub
* Step 3: Create an App using ‘yo teams’
* Step 4: Build the App
* Step 5: Deploying to Azure using Git
* Step 6: Deploy the package to Teams to test the app

**Step 1: Create a Web App in Azure**

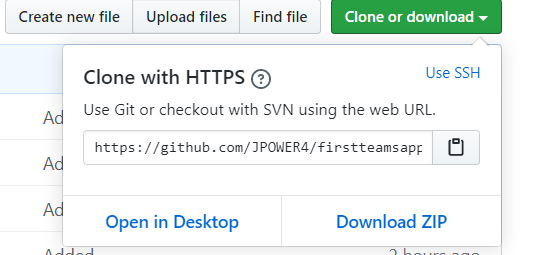
1. Login to Azure
2. Create a resource
3. Select Web App
4. Select Subscription
5. Select or create Resource group
6. Name of the app Ex: <https://firstteamsapptoazure.azurewebsites.net/>
7. Select Code to Publish
8. Runtime stack – Select Node 10.0
9. Operating System – Windows
10. Region – select the default
11. Create new Windows Plan
12. Sku and size
    1. Select F1 under dev / Test
13. Click Review and Create
14. Click Create
15. Now Web App - <https://firstteamsapptoazure.azurewebsites.net/> is ready
16. Refer image below







**Step 2: Create a repository in GitHub**

1. Login github (if you don’t have a account please create one, it is free)
2. Create a new repository **firstteamsapptoAzure**
3. Open the repository
4. Click Clone or download button arrow
5. Click Open in Desktop
   1. 
6. Install github desktop
7. Then Select the folder where you want to clone the project
8. C:\Jenkins\MSTeamsDevelopment\
   1. It will create folder firstteamsapptoAzure

**Step 3: Create an App using ‘yo teams’**

1. Open Command Prompt and create new project using yo teams
2. For this example, I created a personal tab app
3. Then implement your functionality in the code
4. Build and test using gulp ngrok-serve

**Step 4: Build the app**

1. **Building the app**

The application is built using the build Gulp task.

**npm i -g gulp gulp-cli**

**gulp build**

1. **Building the manifest**

To create the Microsoft Teams Apps manifest, run the manifest Gulp task. This will generate and validate the package and finally create the package (a zip file) in the package folder. The manifest will be validated against the schema and dynamically populated with values from the .env file.

**gulp manifest**

1. **Push the code to git hub using GitHub desktop**

**Step 5: Deploying to Azure using Git**

If you want to deploy to Azure using Git follow these steps.

This will automatically deploy your files to Azure, download the npm packages, build the solution and start the web server using Express.

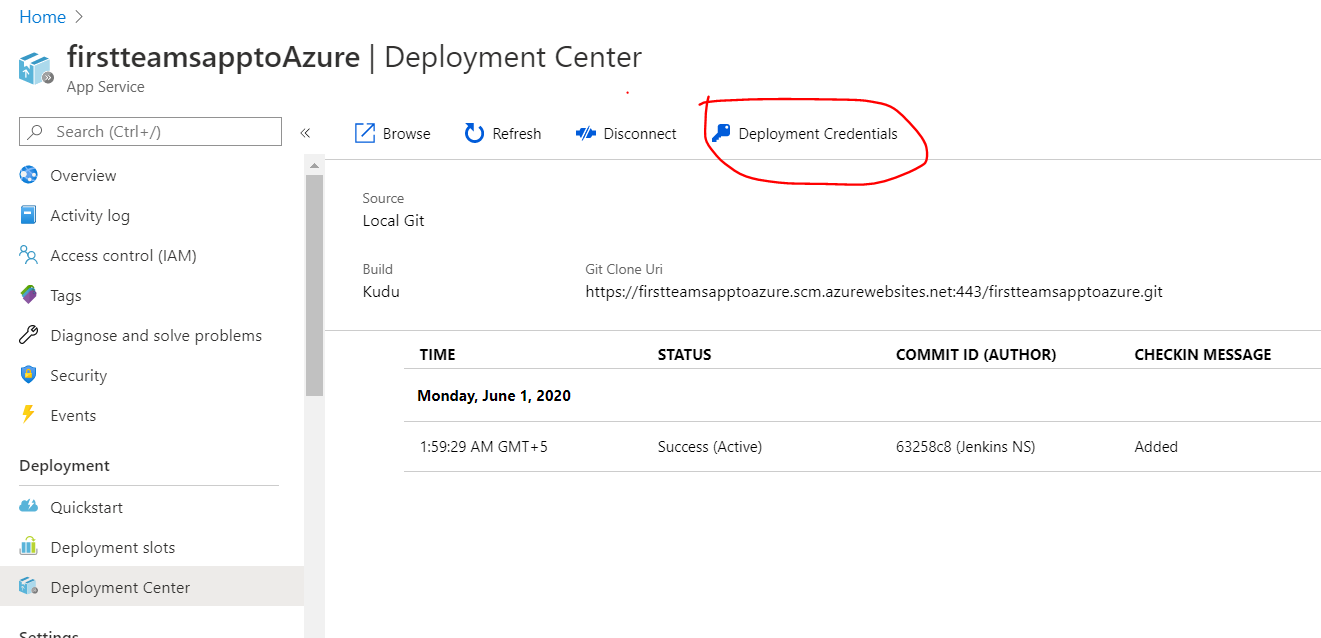
1. Log into [the Azure Portal](https://portal.azure.com/)
2. Add the following keys in the ***Configuration* -> *Application Settings***;
   * Name = WEBSITE\_NODE\_DEFAULT\_VERSION, Value = 8.10.0

and

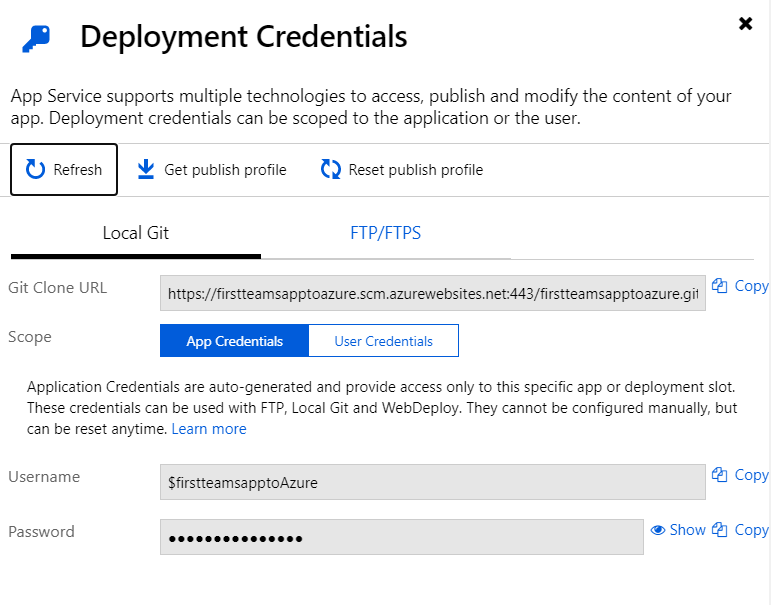
* + Name = SCM\_COMMAND\_IDLE\_TIMEOUT, Value = 1800

Click Save.

1. Go to ***Deployment Center***
2. Choose ***Local Git*** as source and ***App Service build service*** as the Build Provider
3. Click on ***Deployment Credentials*** and store the **App Credentials** securely



* + Copy and paste with your notepad
    1. Username and password



**Open Command Prompt – your project folder**

1. In your tab folder initialize a Git repository using **git init**
2. Build the solution using **gulp build** to make sure you don't have any errors
3. Commit all your files using **git add -A && git commit -m "Initial commit"**
4. Run the following command to set up the remote repository:

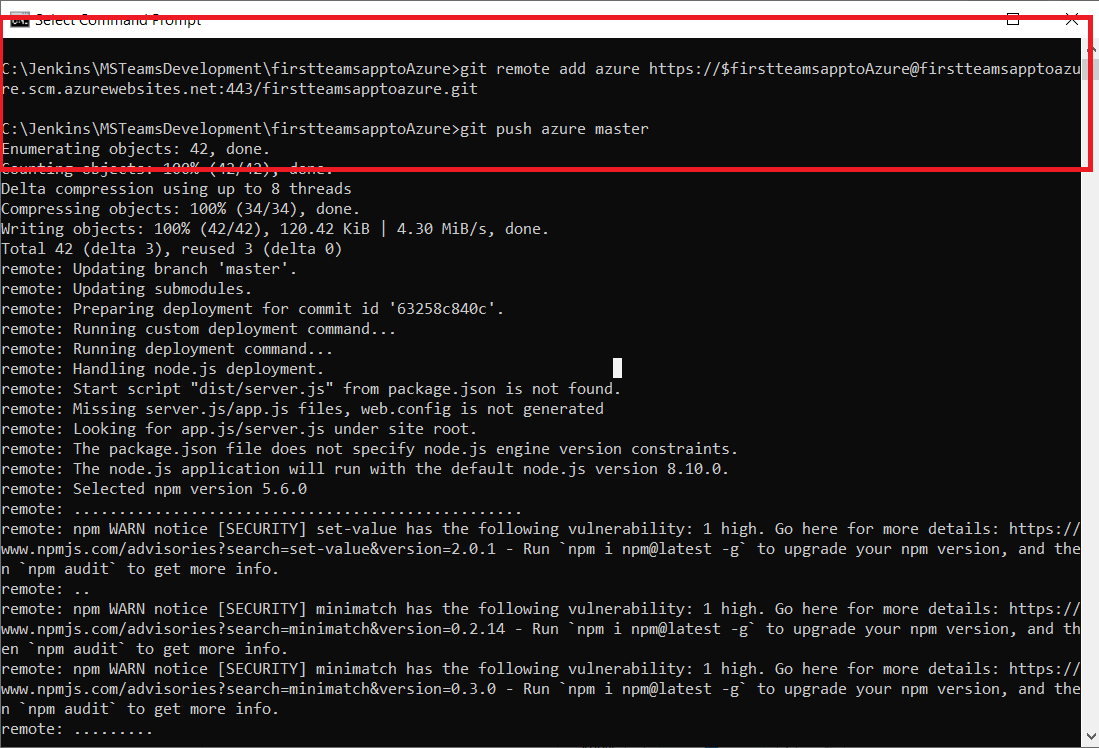
***git remote add azure https://<username>@firstteamsapptoazure.scm.azurewebsites.net:443/firstteamsapptoazure.git.***

Ex: ***git remote add azure https://$firstteamsapptoAzure@firstteamsapptoazure.scm.azurewebsites.net:443/firstteamsapptoazure.git***

You need to replace with the username of the App Credentials you retrieved in *Deployment Credentials*. You can also copy the URL from *Options* in the Azure Web App.

1. To push your code use to Azure, use the following command:

***git push azure master***

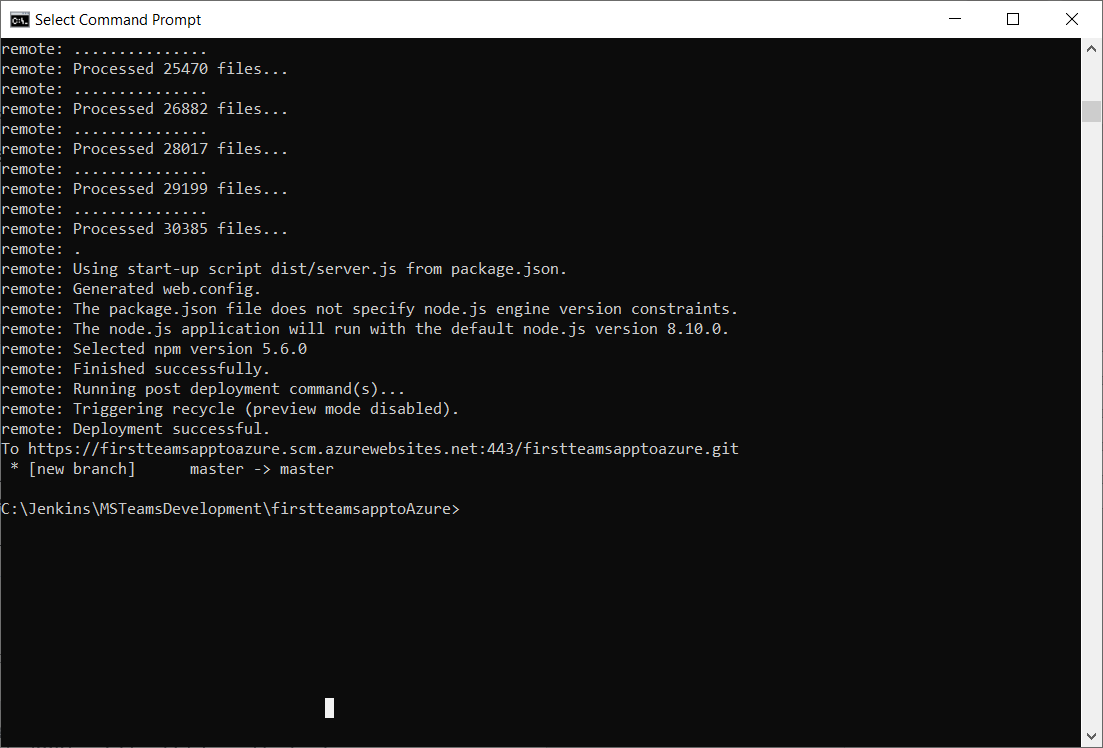
****

1. you will be asked for your credentials the first time, insert the Password for the App Credential. Example:

***$firstteamsapptoAzure***

***oYb6ACqCf9QJCvbvNWk8x033434ZQvpjQt9ys4LGbFCNbA075oiyXuGLxT***

Note that you should update the Azure Web Site application setting before pushing the code as the settings are needed when building the application.

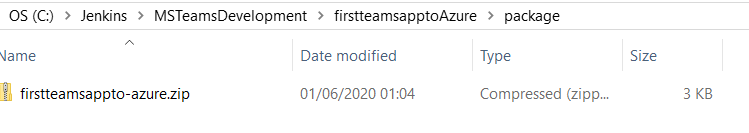


1. Wait until the deployment is completed and navigate to <https://firstteamsapptoazure.azurewebsites.net/privacy.html> to test that the web application is running
2. Done
3. **Repeat 11 for every commit you do and want to deploy**

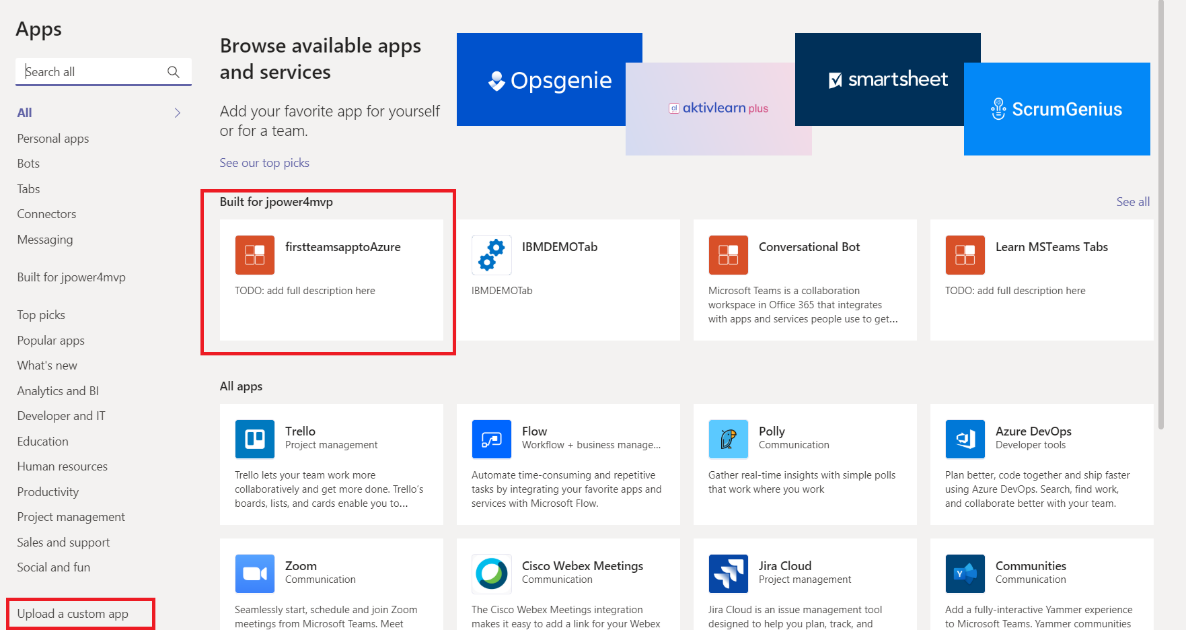
NOTE: The .env file is excluded from source control and will not be pushed to the web site so you need to ensure that all the settings present in the .env file are added as application settings to your Azure Web site (except the PORT variable which is used for local debugging).

**Step 6: Deploy the package to Teams to test the app**

1. Login to Teams
2. Click … , then click more apps
3. Click 🡪 Upload a custom app
4. Select the zip file under package folder



1. Deploy the app



1. Click and add the app
2. Then test it

