React – Node.js application

In this section, you will create a simple static website that can be deployed to the cloud. This tutorial uses [create-react-app](https://github.com/facebook/create-react-app), a React utility CLI, to quickly scaffold out a simple React app from the terminal.

Install create-react-app tool

[React](https://reactjs.org/) is a popular framework for building web applications, so we will use it as an example. You can scaffold (create) a new React application using the [create-react-app](https://github.com/facebook/create-react-app) tool. The create-react-app tool is shipped as an npm module and can be installed by using npm.

npm install -g create-react-app

The -g switch installs the create-react-app globally on your machine so you can run it from anywhere.

Create a new application

Next, scaffold a new React app called react-app by running:

create-react-app react-app

And build the application by switching to the new folder and running npm run build.

cd react-app

npm run build

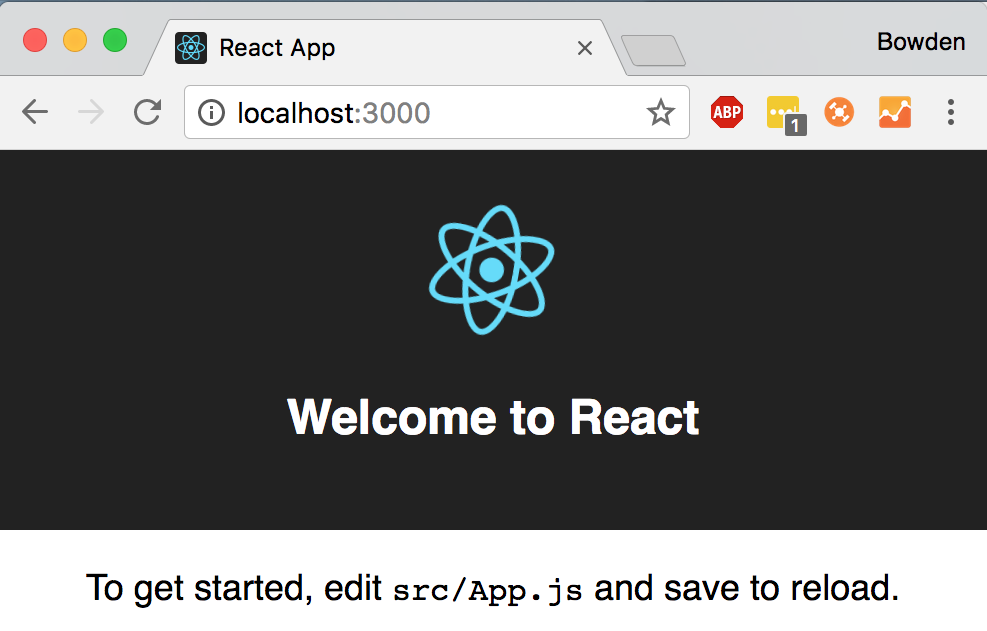
You should now have a build folder in your project folder. This contains the .html, .css, and .js files we will be deploying to Azure Storage.

## Run the application

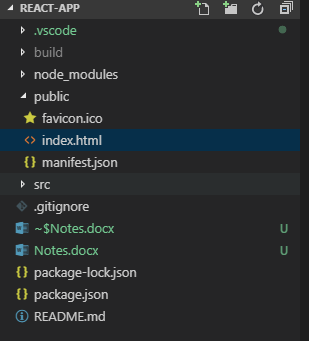
Finally, let's ensure that the application runs. From the terminal, start the application using the npm startcommand.

npm start

Now, open your browser and navigate to [http://localhost:3000](http://localhost:3000/), where you should see something like this:

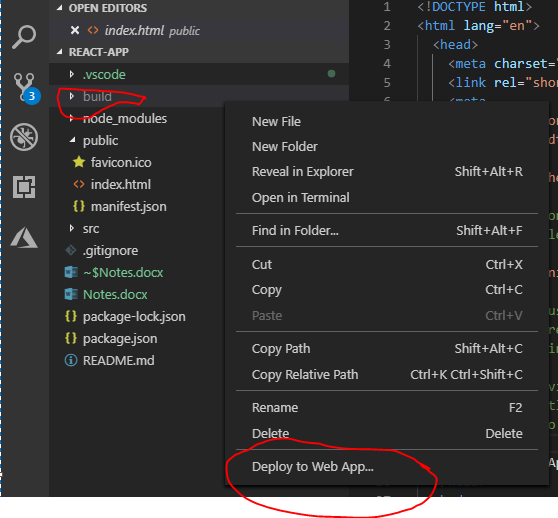


Here, I have made couple of Components for practice, and looks like this.



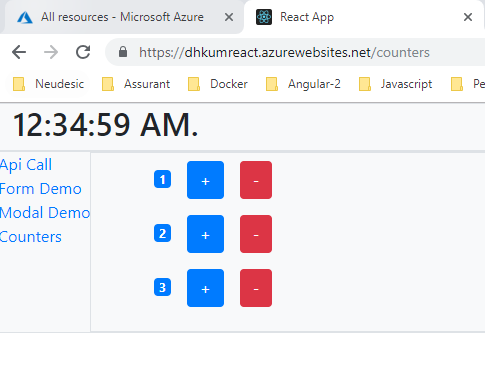
## Deploy your website

1. Open your application code in VS Code, right-click on your **build** directory, and choose **Deploy to Static Website...**.



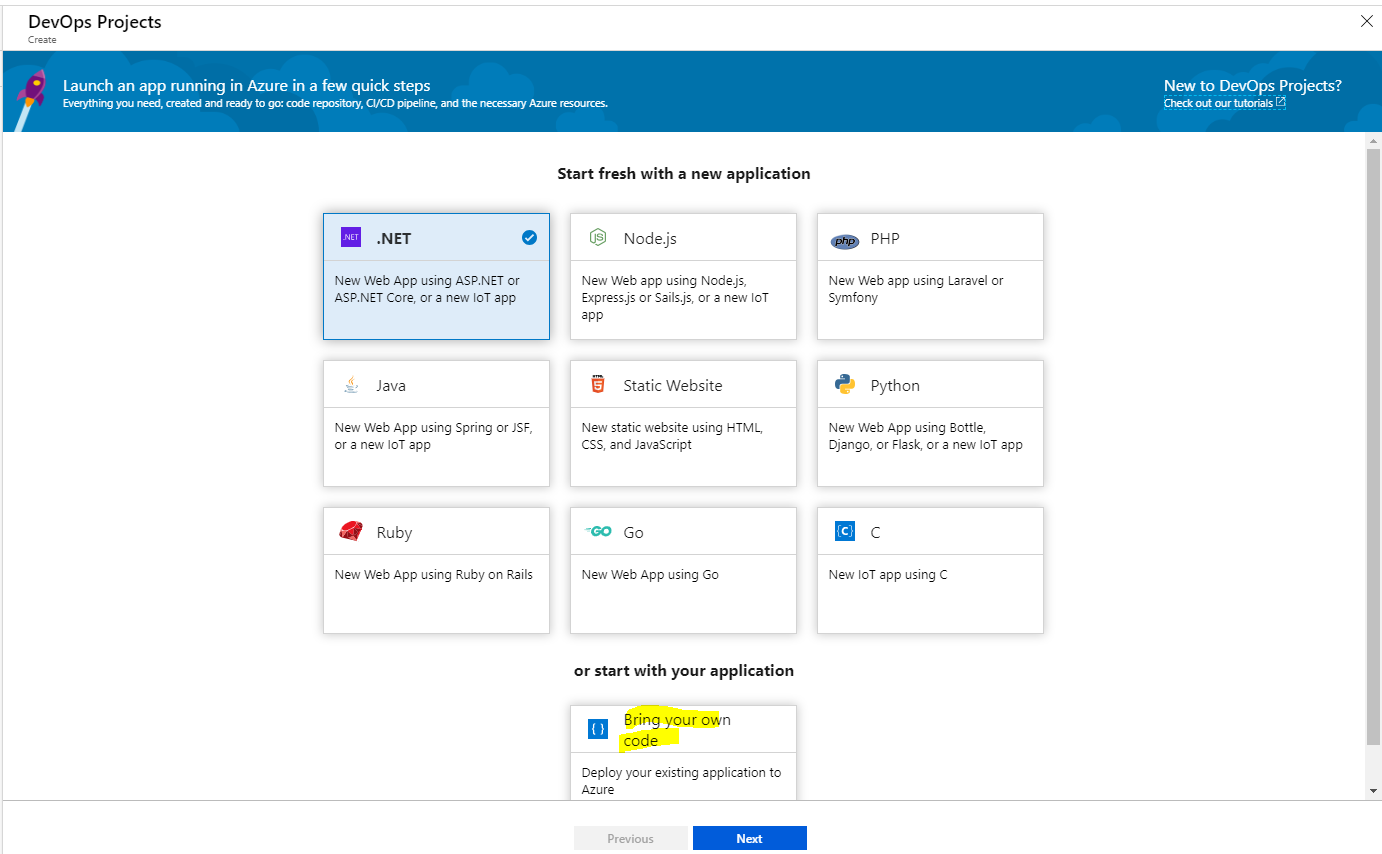
## Browse your website

Visit the URL to see your app running on Azure.

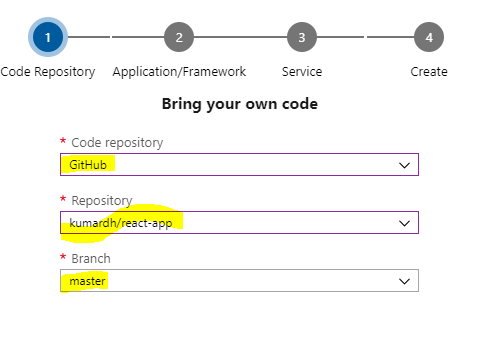


## Azure CI/CD

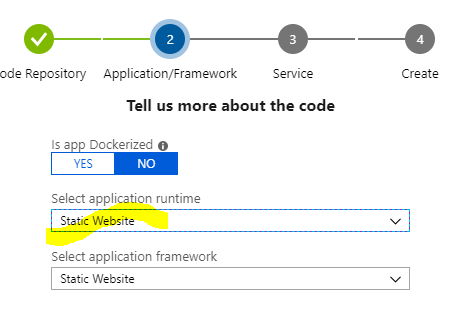
1. Create DevOps projectOpen your application – Bring your own Code.



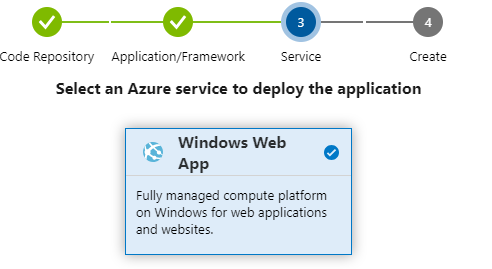
1. Set Code repository. I have use GitHub.



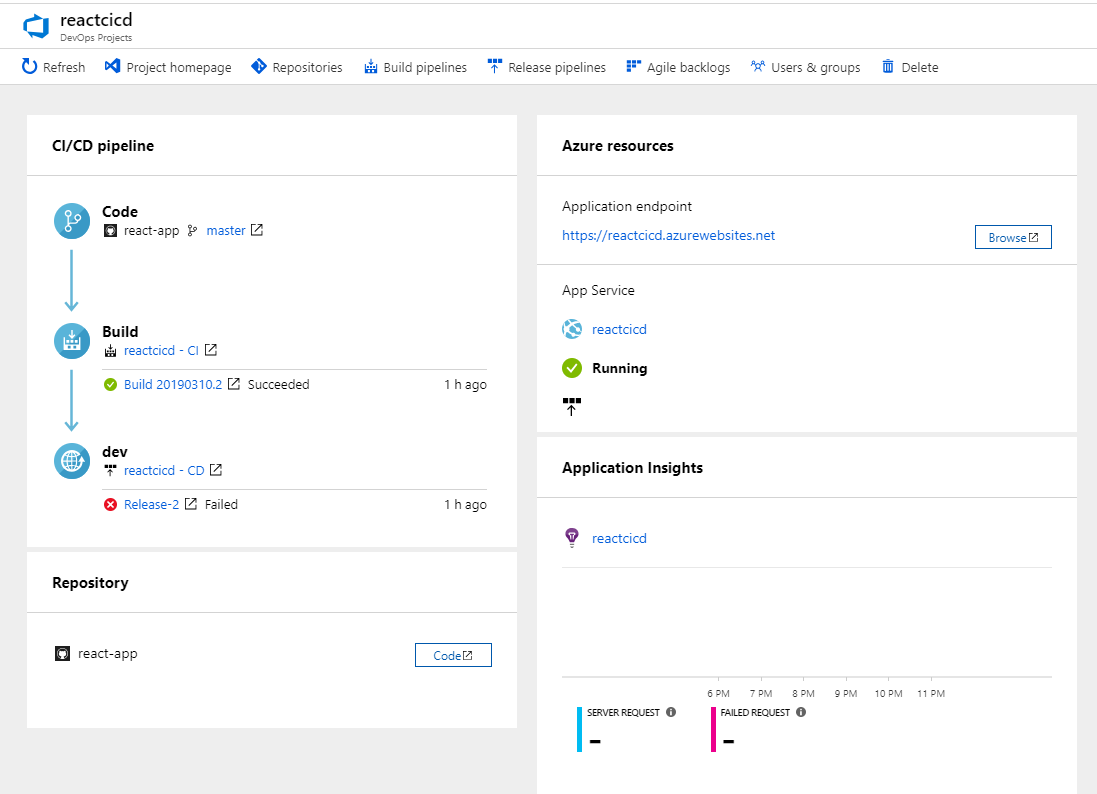
1. Select Static Application runtime.



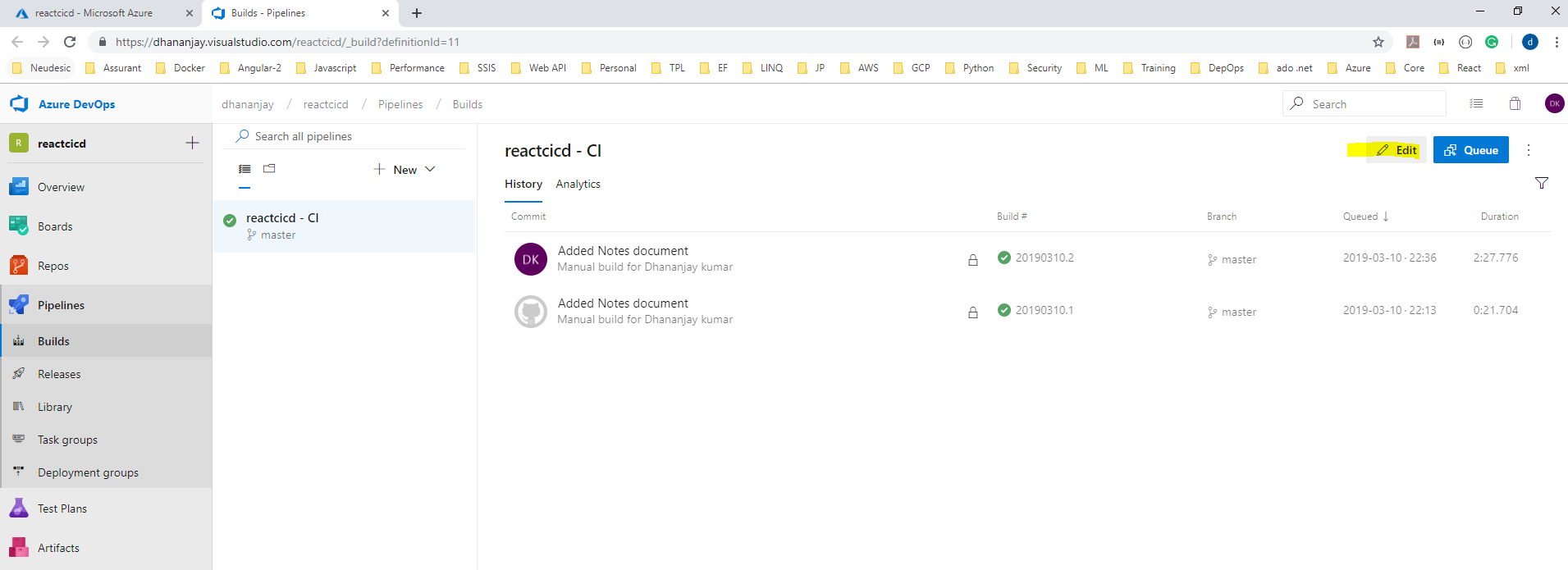
1. Select Windows Web App to deploy application.



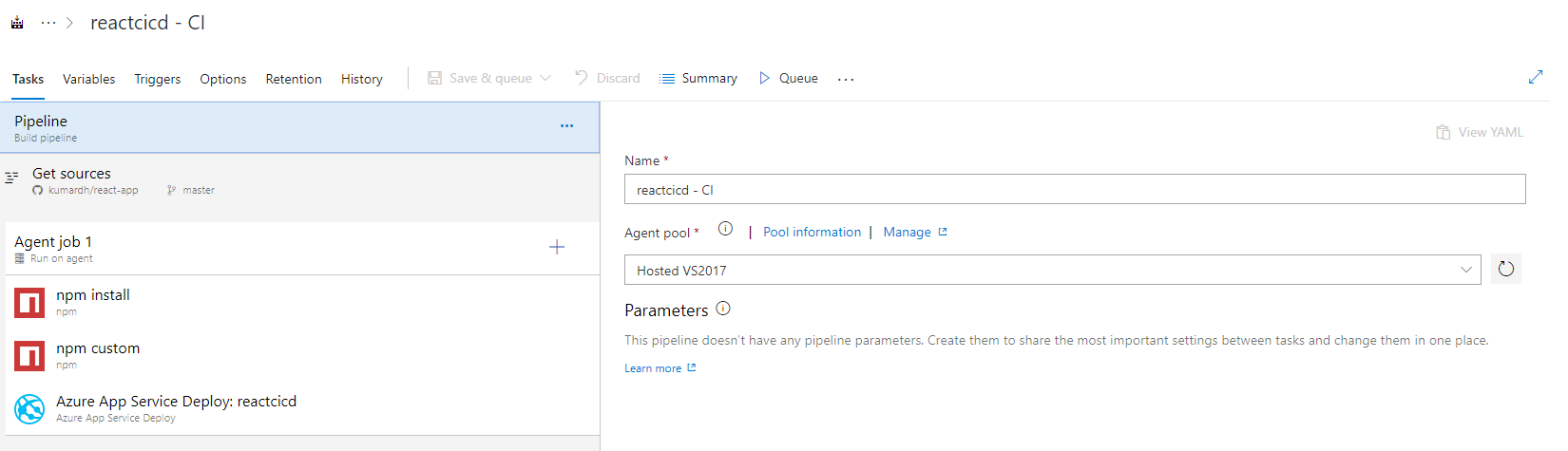
1. Click Next and create DevOps Project – In my case I have created as reactcicd.



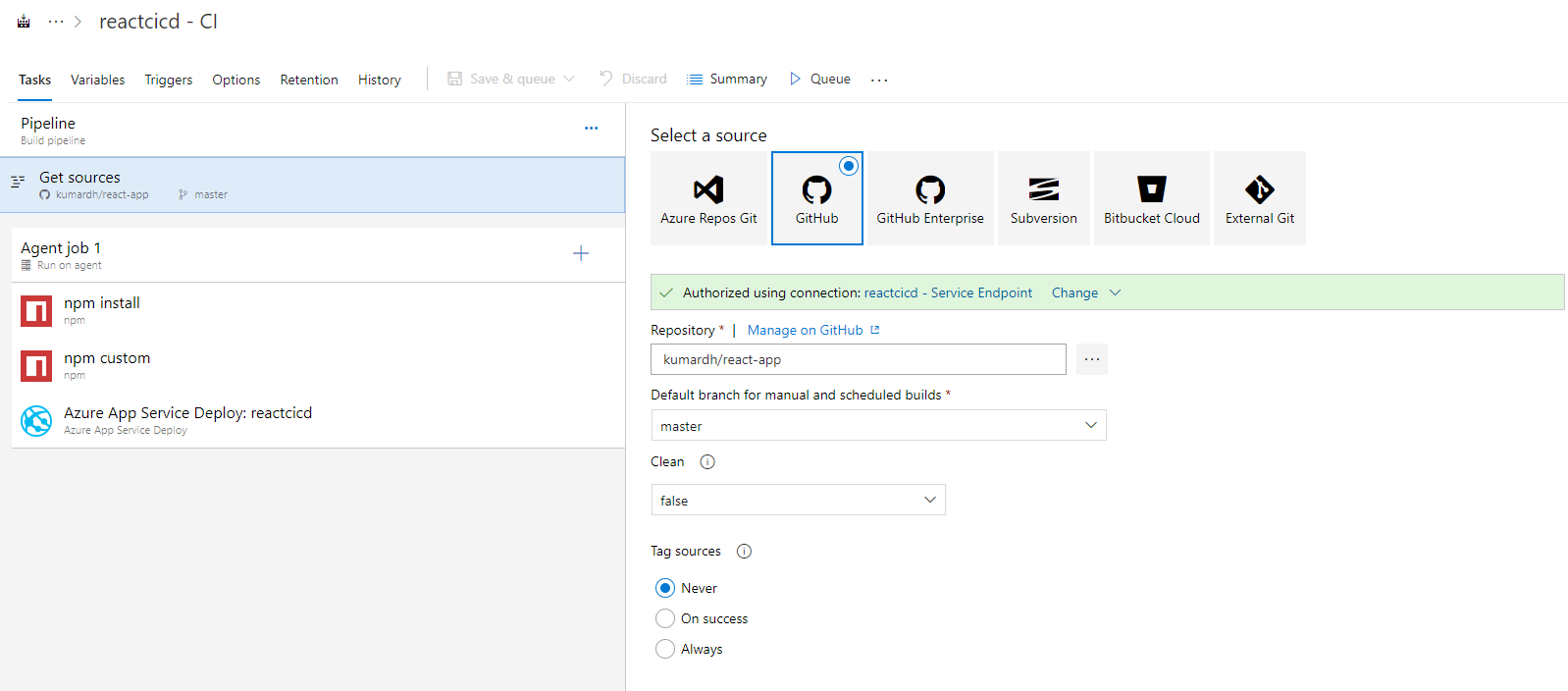
1. Open reactcicd – CI pipeline.



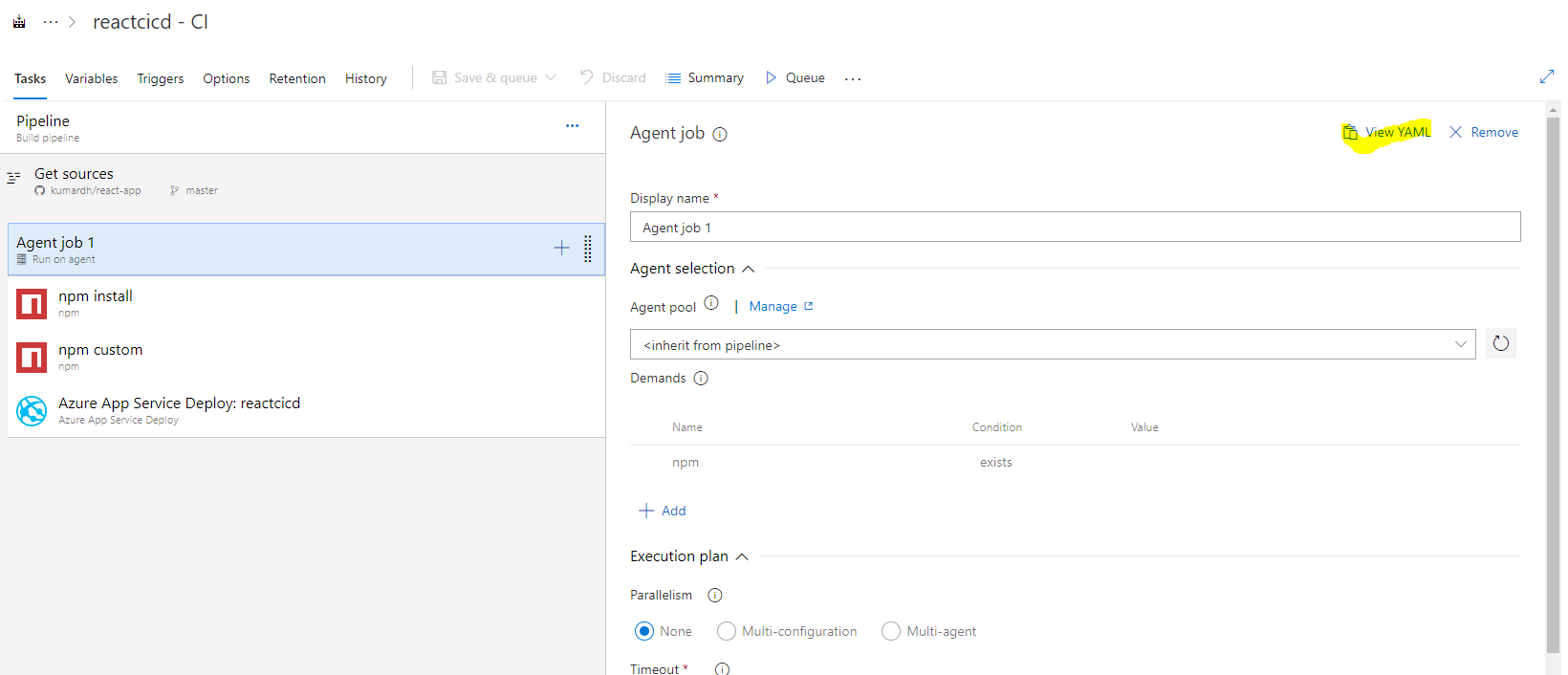
1. Click Edit.



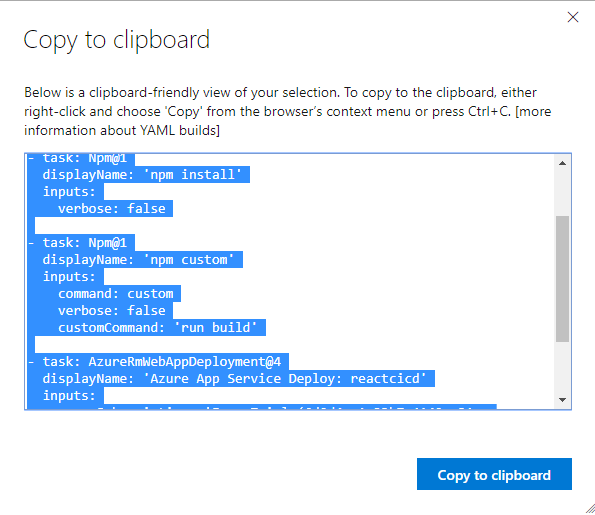
1. Below are the tasks created.



1. I have selected Hosted VS2017 Agent.



1. You can view YAML.



pool:

vmImage: Hosted VS2017

demands: npm

steps:

- task: Npm@1

displayName: 'npm install'

inputs:

verbose: false

- task: Npm@1

displayName: 'npm custom'

inputs:

command: custom

verbose: false

customCommand: 'run build'

- task: AzureRmWebAppDeployment@4

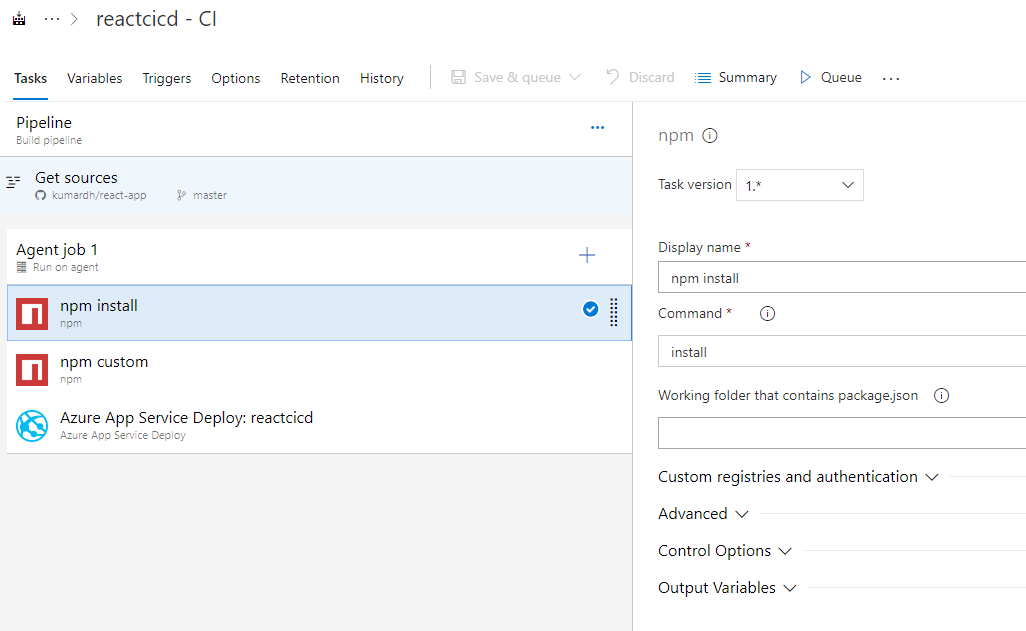
displayName: 'Azure App Service Deploy: reactcicd'

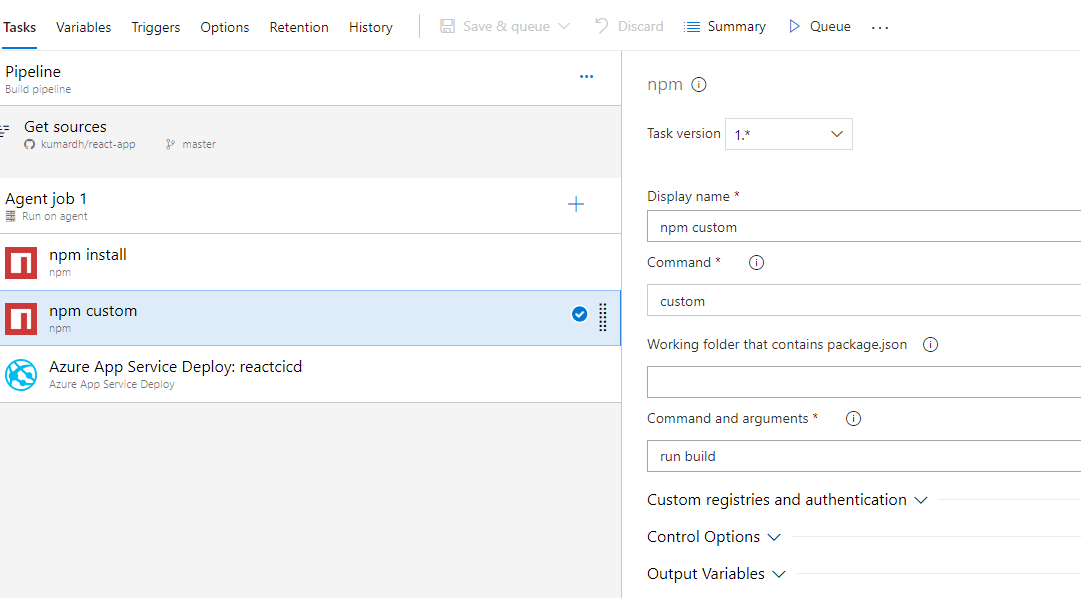
inputs:

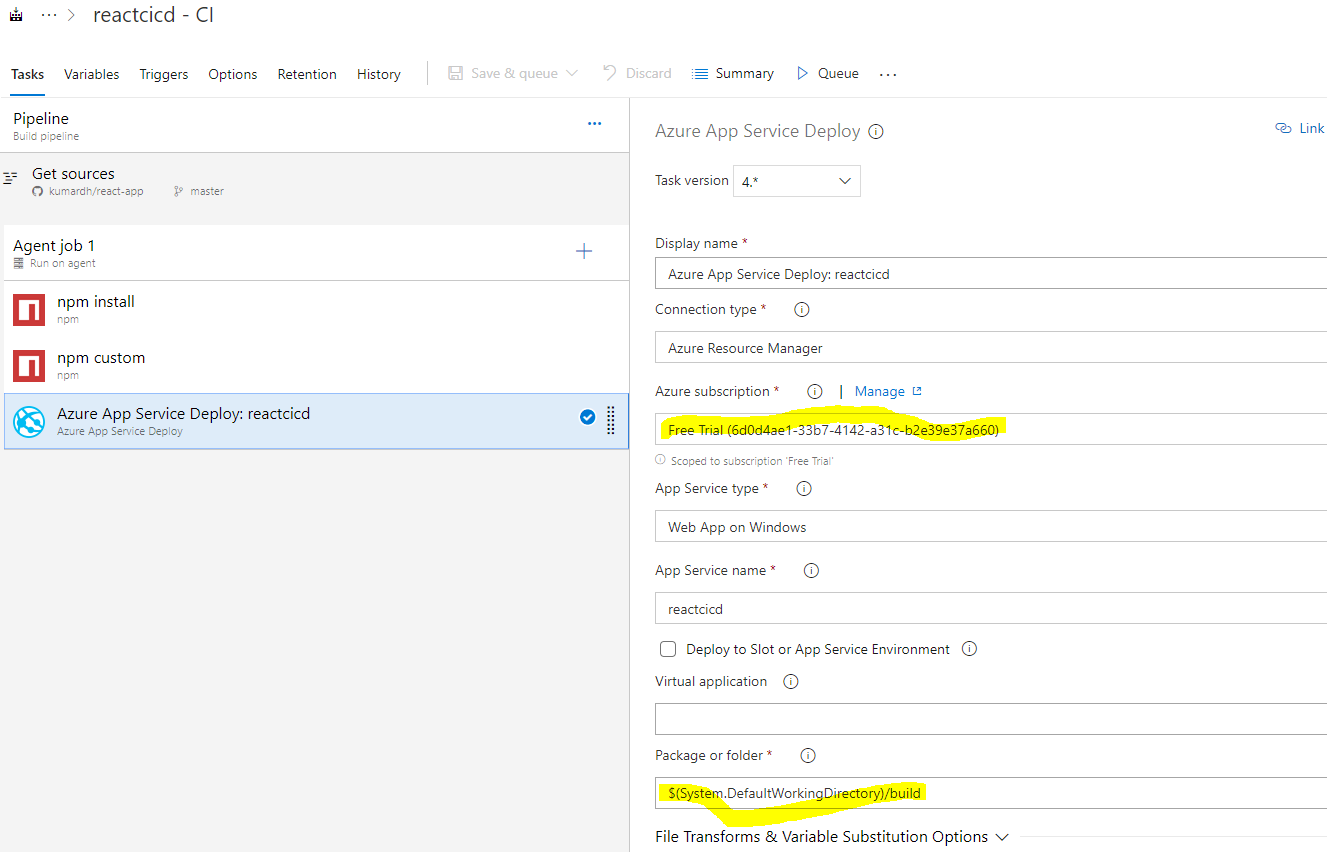
azureSubscription: 'Free Trial (6d0d4ae1-33b7-4142-a31c-b2e39e37a660)'

WebAppName: reactcicd

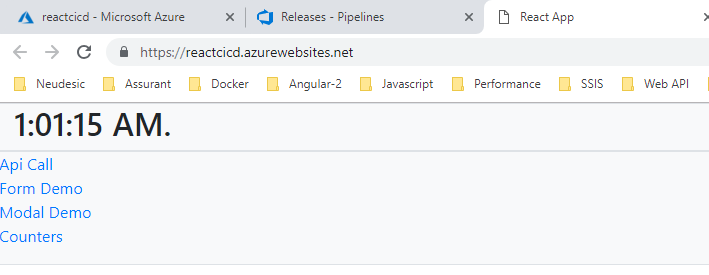
packageForLinux: '$(System.DefaultWorkingDirectory)/build'

1. Next Add below Taks to Agent job 1. 





1. Once done – Queue build and you will see the page



1. Now validate Continous Integration by changing code and checking in to Github.

