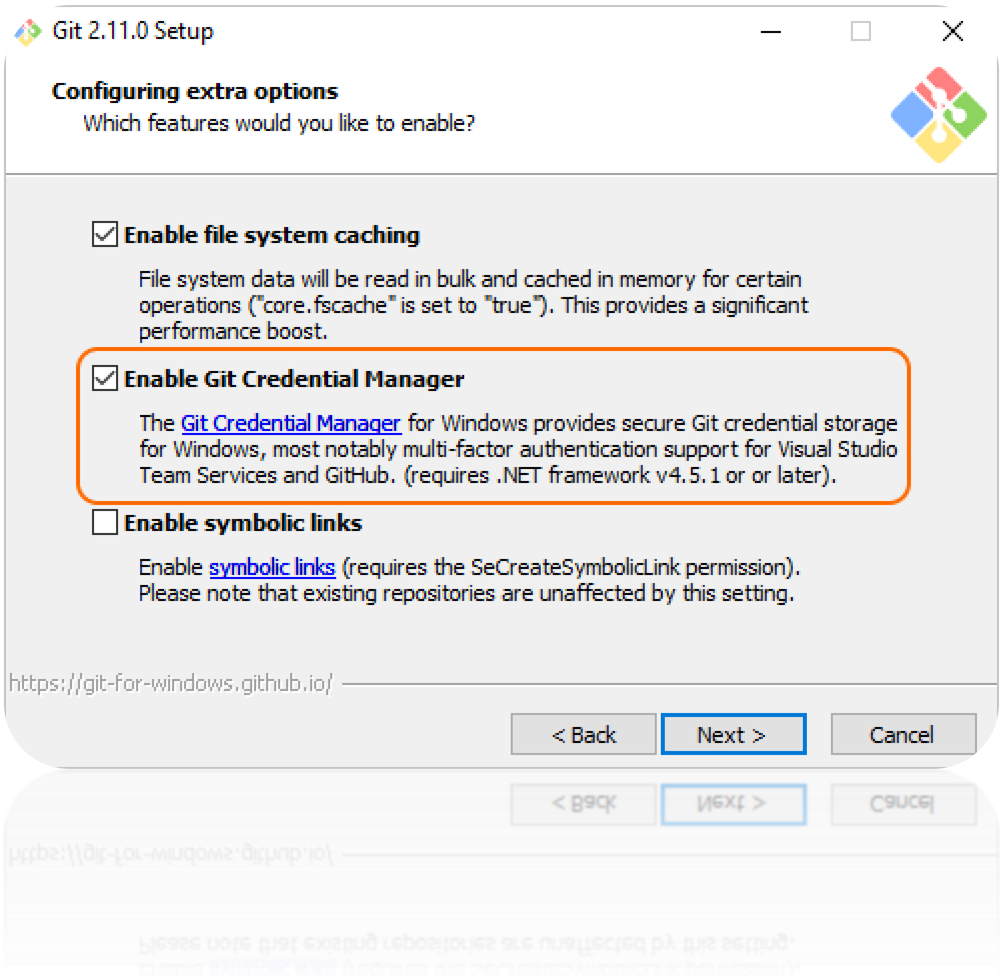
1.Pre-requisites

This lab assumes you have the Google Chrome browser installed and available for debugging. If you do not have Chrome installed, go to https://www.google.com/chrome/browser/

Download and run the latest Git for Windows installer, which includes the Git Credential Manager for Windows. Make sure to leave the Git Credential Manager installation option enabled when prompted.2



Configuring extra options
Which features would you like to enable?

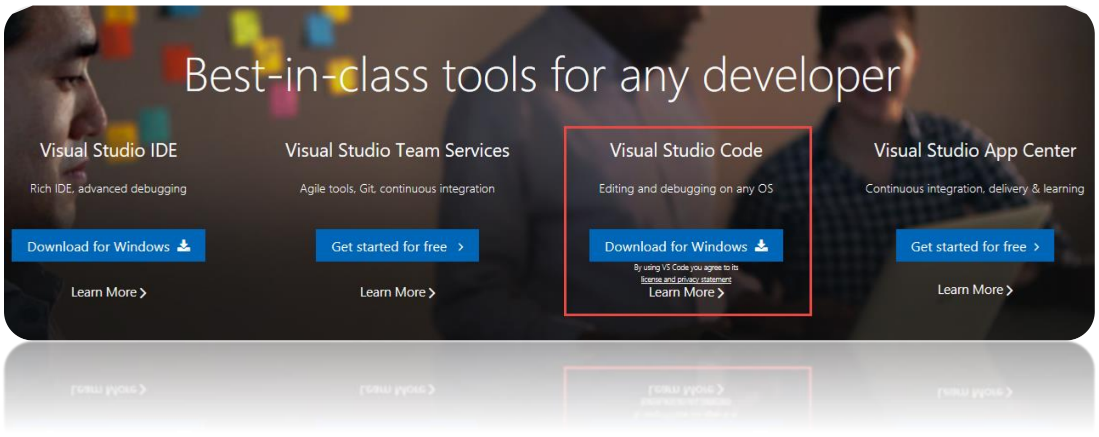
[✓] **Enable file system caching**
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

[✓] **Enable Git Credential Manager**
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

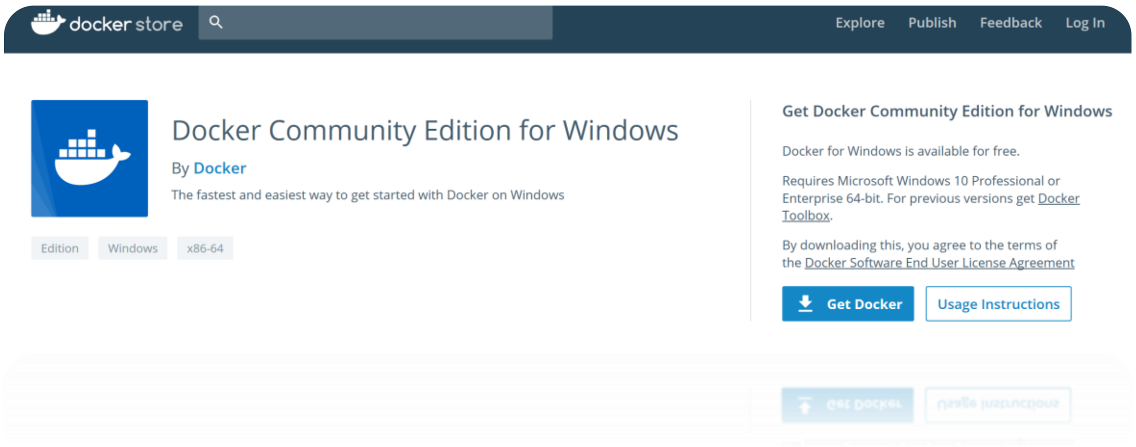
[ ] **Enable symbolic links**
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

Note: When you connect to a VSTS Git repository from your Git client for the first time, the credential manager prompts for your Microsoft Account or Azure Active Directory credentials. If your account has multi-factor authentication enabled, you are prompted to go through that experience as well.

Download Visual Studio Code from http://visualstudio.com


Best-in-class tools for any developer
Visual Studio IDE
Rich IDE, advanced debugging
Download for Windows
Learn More
Visual Studio Team Services
Agile tools, Git, continuous integration
Get started for free
Learn More
Visual Studio Code
Editing and debugging on any OS
Download for Windows
By using VS Code you agree to its license and privacy statement
Learn More
Visual Studio App Center
Continuous integration, delivery & learning
Get started for free
Learn More

Install Docker from https://docs.docker.com/install/



Docker Community Edition for Windows

By Docker

The fastest and easiest way to get started with Docker on Windows

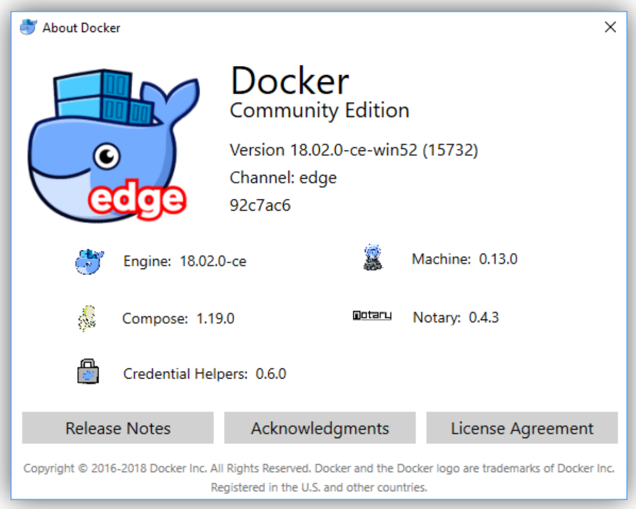
Get Docker Community Edition for Windows

Docker for Windows is available for free.

Requires Microsoft Windows 10 Professional or Enterprise 64-bit. For previous versions get Docker Toolbox.

By downloading this, you agree to the terms of the Docker Software End User License Agreement

Note: Make sure you install Docker "Edge" for windows, not the "Stable" release. This guide has been verified against the following Docker version:



Docker
Community Edition

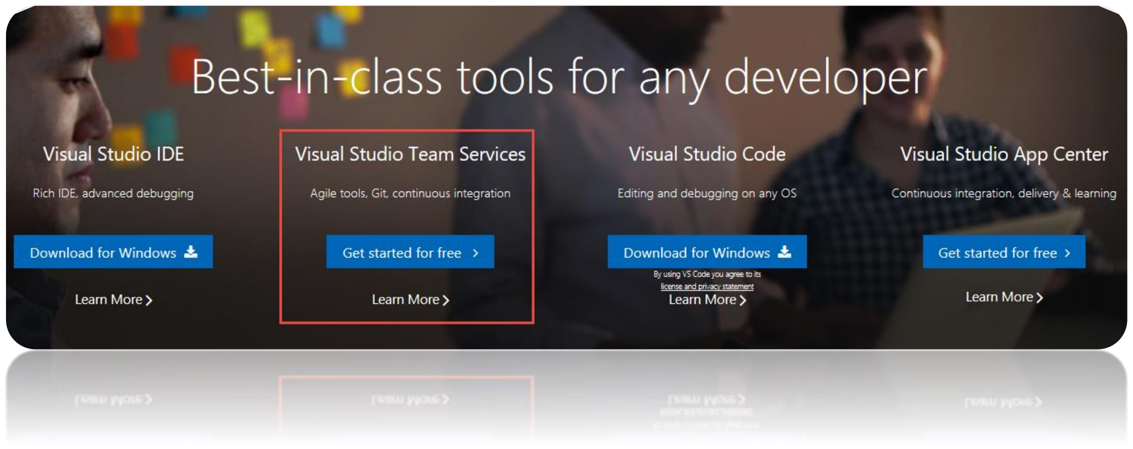
Version 18.02.0-ce-win52 (15732)
Channel: edge
92c7ac6

Engine: 18.02.0-ce
Machine: 0.13.0
Compose: 1.19.0
Notary: 0.4.3
Credential Helpers: 0.6.0

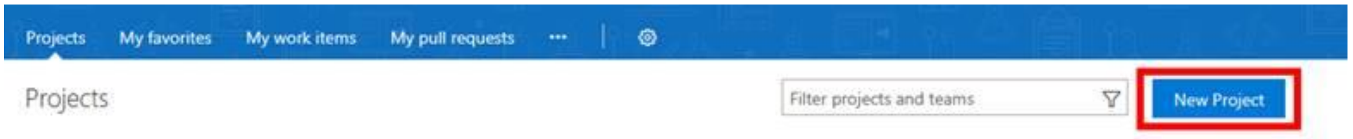
Copyright © 2016-2018 Docker Inc. All Rights Reserved. Docker and the Docker logo are trademarks of Docker Inc. Registered in the U.S. and other countries.

2.Create a Project in VSTS

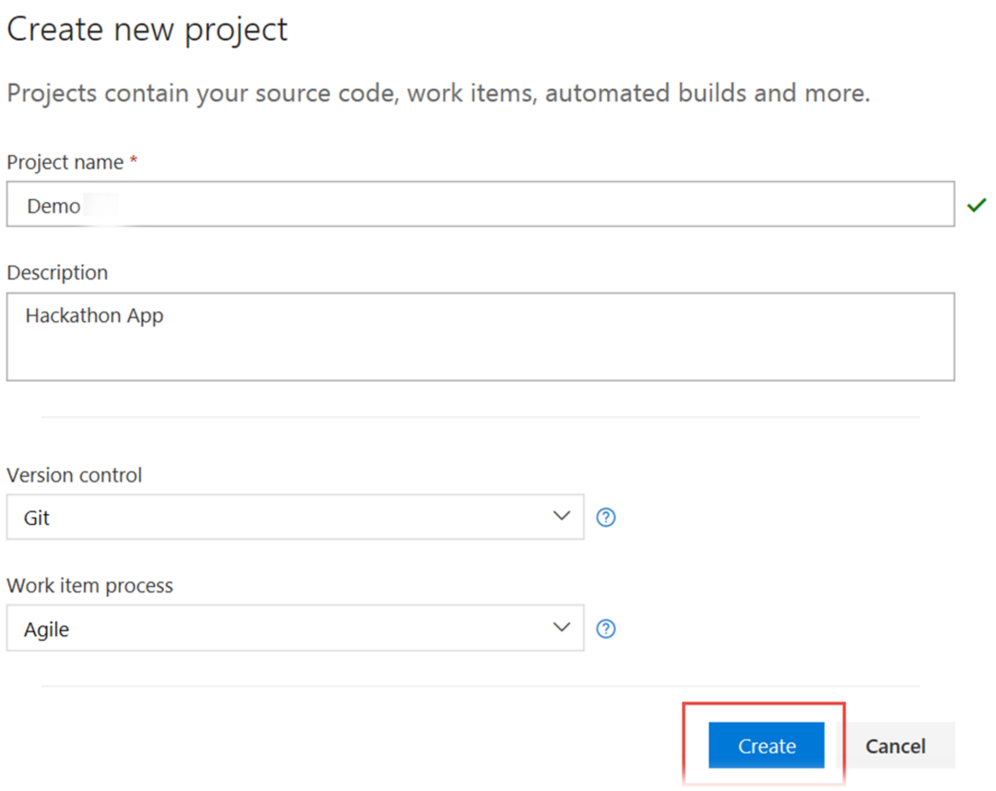
1. Create a new instance of Visual Studio Team Services by navigating to http://visualstudio.com


Best-in-class tools for any developer
Visual Studio IDE
Rich IDE, advanced debugging
Download for Windows
Learn More
Visual Studio Team Services
Agile tools, Git, continuous integration
Get started for free
Learn More
Visual Studio Code
Editing and debugging on any OS
Download for Windows
By using VS Code you agree to its license and privacy statement
Learn More
Visual Studio App Center
Continuous integration, delivery & learning
Get started for free
Learn More

2. Click on “**New Project**” in VSTS.



3. Enter Project Name, Description, Version control, and Work item process and click **Create**.


Create new project

Projects contain your source code, work items, automated builds and more.

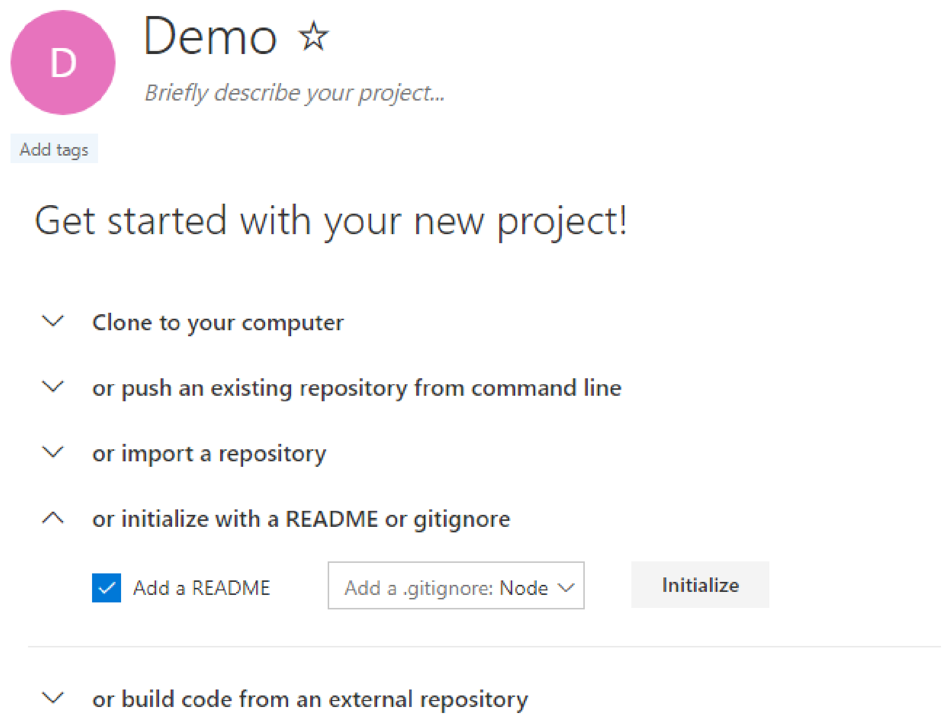
Project name *
Demo

Description
Hackathon App

Version control
Git

Work item process
Agile

4. Select “or initialize with a readme or gitignore”.
5. Add a .gitignore file by selecting “Node”,
6. Click Initialize.


Demo ☆

Briefly describe your project...

Add tags

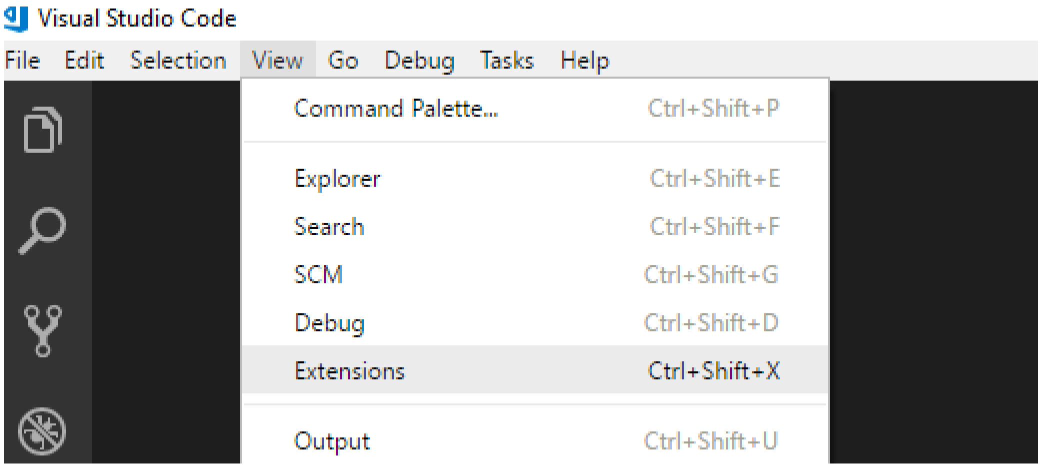
Get started with your new project!

- Clone to your computer
- or push an existing repository from command line
- or import a repository
- or initialize with a README or gitignore
  - ✓ Add a README
  - Add a .gitignore: Node
  - Initialize
- or build code from an external repository

Note: Readme file is used to give a brief introduction of the project and gitignore file is used to ignore tracking of files such as temp files and build results.

3. Open Visual Studio Code

1. Install Extensions by Selecting View→ Extensions and typing "javascript"



Recommended extensions to install:

Angular 5 and TypeScript/HTML VS Code Snippets

Angular 5 Snippets - TypeScript, Html, Angular Material, ngRx, RxJS & Flex Layout

ESLint

JavaScript (ES6) code snippets

npm IntelliSense

Debugger for Chrome

Visual Studio Team Services

Docker

Docker Explorer

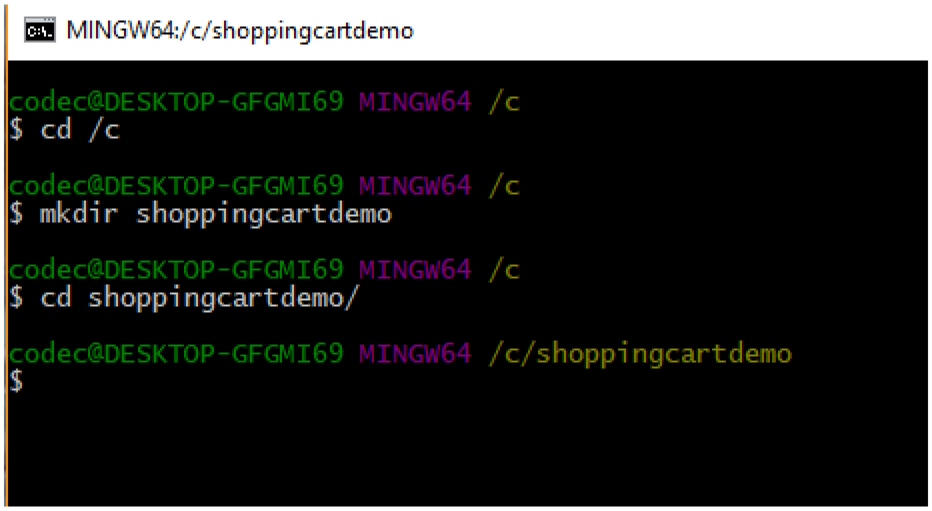
Nginx.Conf

Nginx.Conf Hint

Apache conf

Apache Conf Snippets

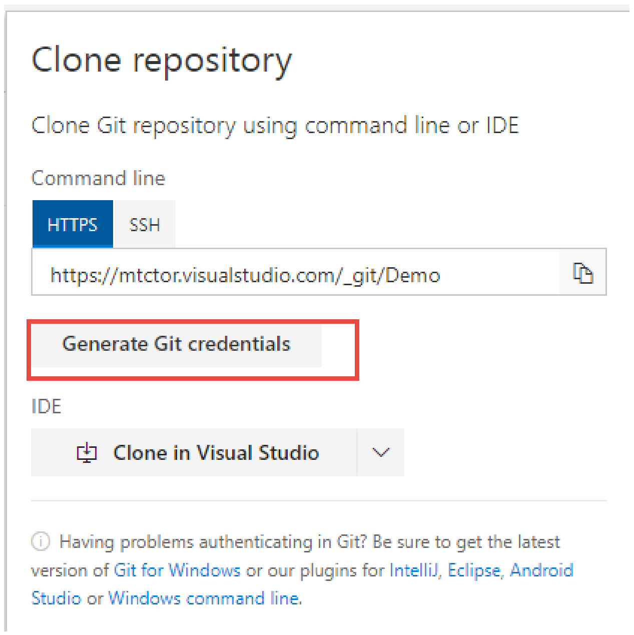
2. Launch Git Bash or use Windows Command line to execute the following commands to create our repository directory:



MINGW64:/c/shoppingcartdemo

```
codec@DESKTOP-GFGMI69 MINGW64 /c
$ cd /c

codec@DESKTOP-GFGMI69 MINGW64 /c
$ mkdir shoppingcartdemo

codec@DESKTOP-GFGMI69 MINGW64 /c
$ cd shoppingcartdemo/

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo
$
```

3. Open your VSTS project in your browser
4. Click on Clone in the upper right-hand corner
5. Generate Git Credentials:



Clone repository

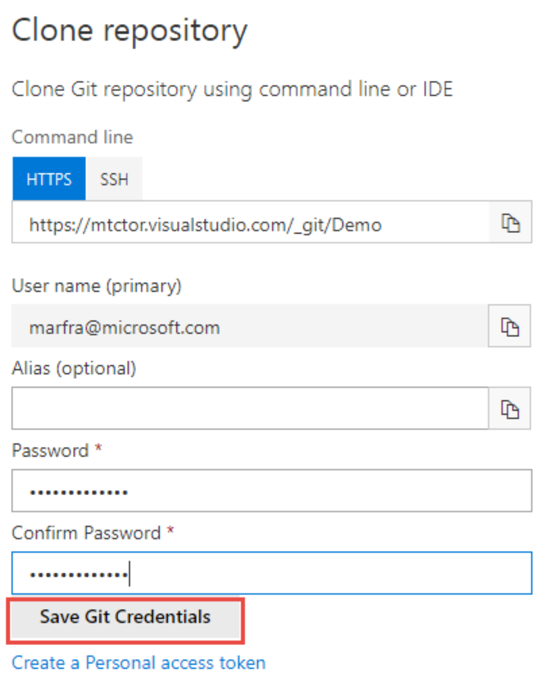
Clone Git repository using command line or IDE

Command line

https://mtctor.visualstudio.com/_git/Demo

Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

6. Then enter a new password and click Save Git Credentials:



Clone repository

Clone Git repository using command line or IDE

Command line

https://mtctor.visualstudio.com/_git/Demo

User name (primary)

marfra@microsoft.com

Alias (optional)

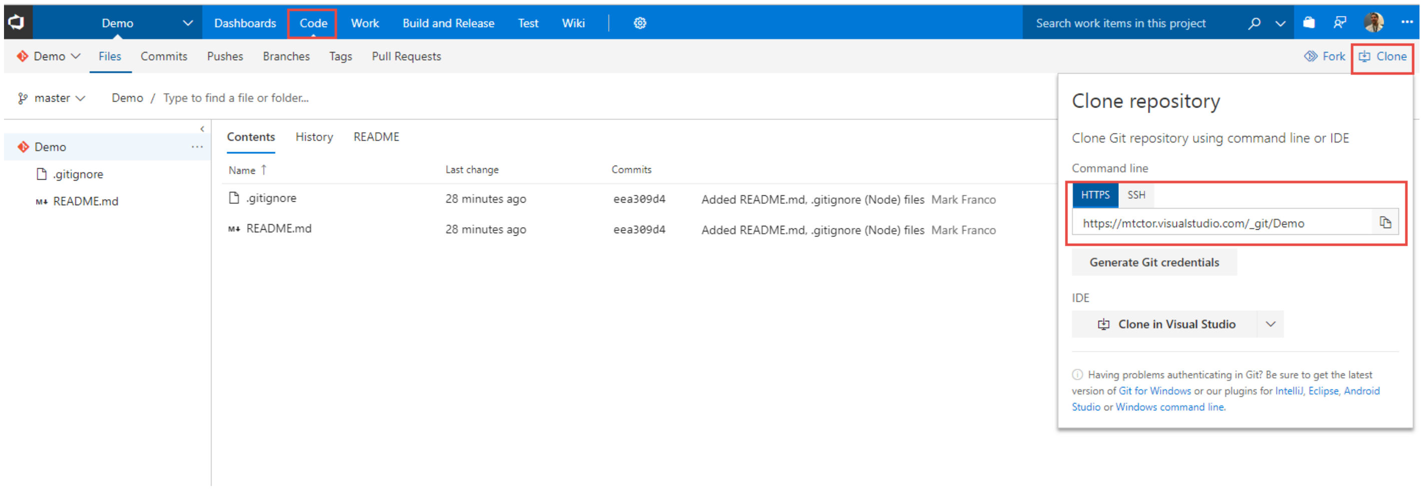
Password *

•••••••••••••

Confirm Password *

•••••••••••••

7. copy the git repository url as follows:



Name	Last change	Commits		
.gitignore	28 minutes ago	eea309d4	Added README.md, .gitignore (Node) files	Mark Franco
README.md	28 minutes ago	eea309d4	Added README.md, .gitignore (Node) files	Mark Franco

Clone repository

Clone Git repository using command line or IDE

Command line

HTTPS SSH

https://mtctor.visualstudio.com/_git/Demo

Generate Git credentials

IDE

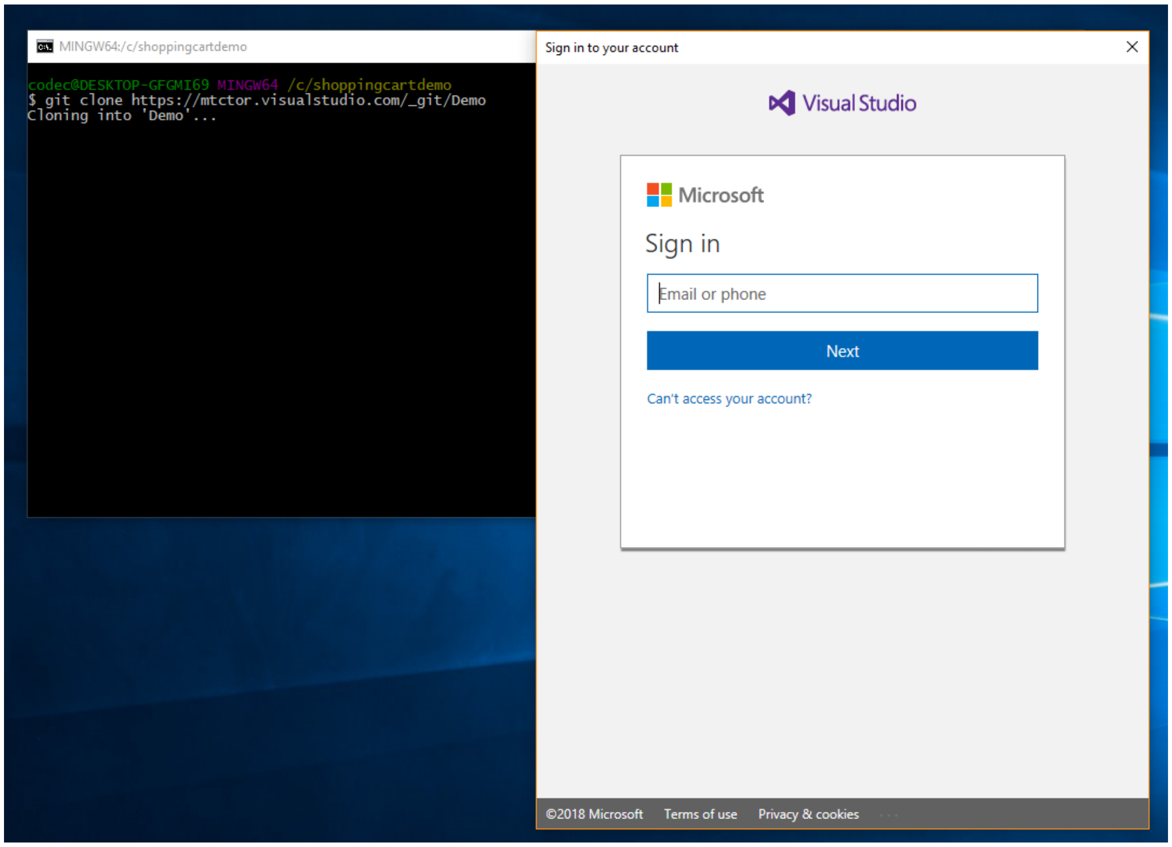
Clone in Visual Studio

Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

8. Clone the repository from the bash shell you opened earlier as follows:

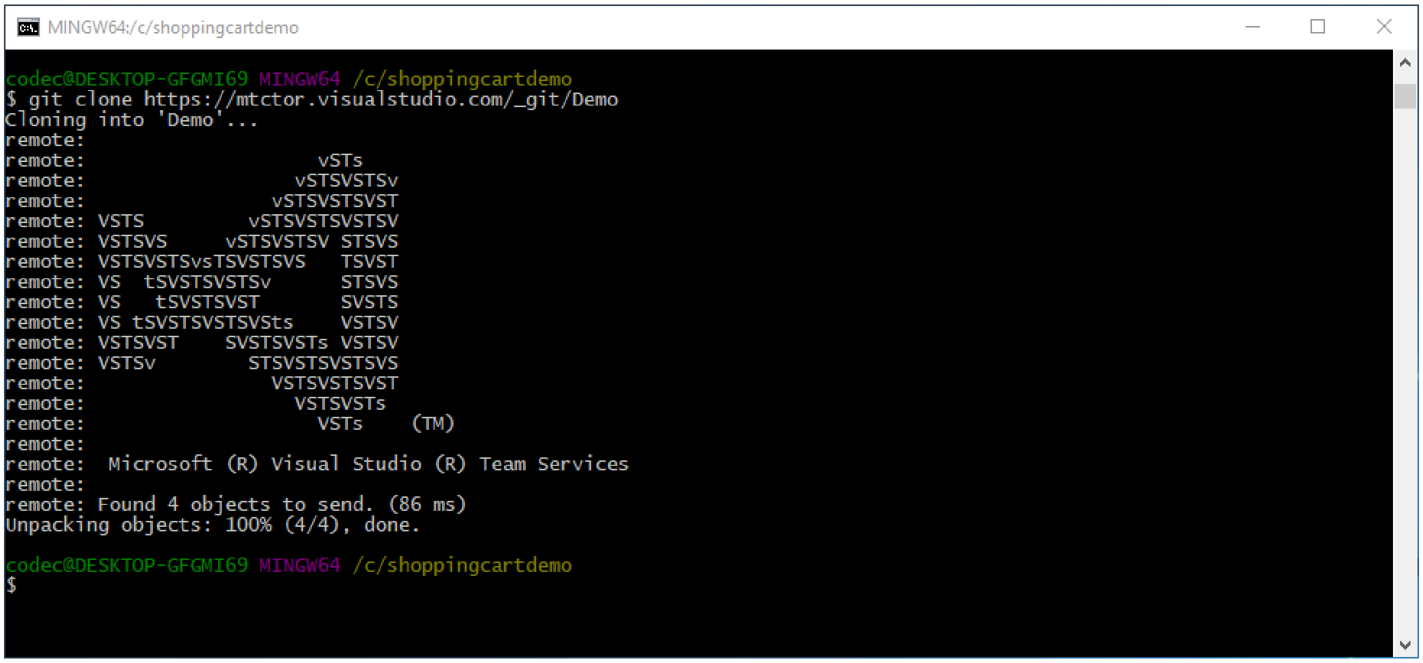
```
Git clone <git Repository you copied in previous step>
```

9. Enter your credentials you setup in previous steps



```
codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo
$ git clone https://mtctor.visualstudio.com/_git/Demo
Cloning into 'Demo'...
```

After successful login you should see:

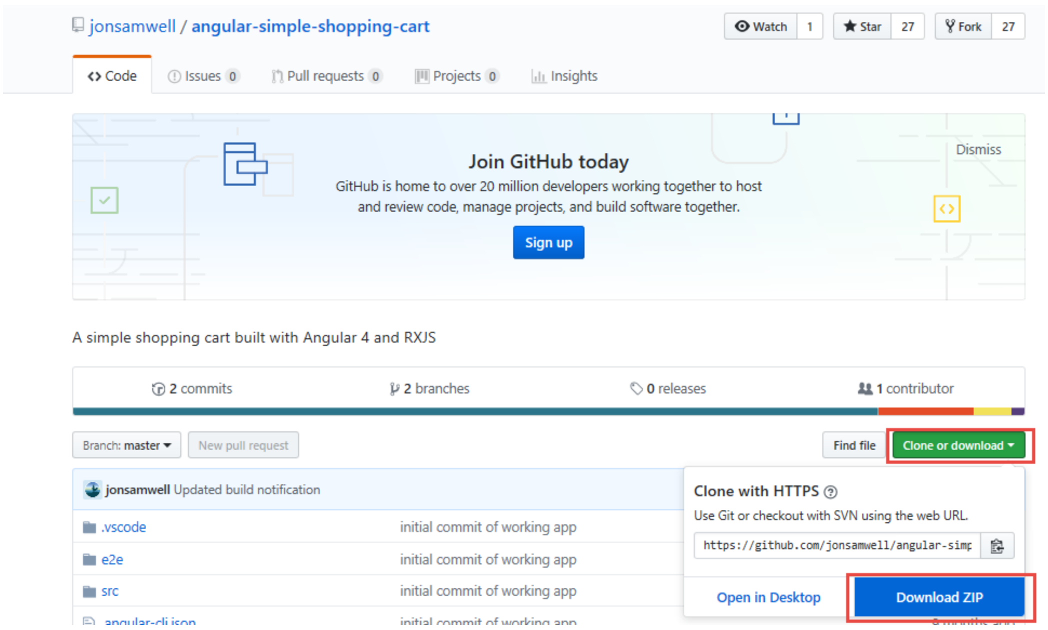


```
codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo
$ git clone https://mtctor.visualstudio.com/_git/Demo
Cloning into 'Demo'...
remote:
remote:                      vSTs
remote:                    vSTSVSTSv
remote:                  vSTSVSTSVST
remote: VSTS           vSTSVSTSVSTSV
remote: VSTSVS       vSTSVSTSV STSVS
remote: VSTSVSTSvsTSVSTSVS    TSVST
remote: VS   tSVSTSVSTSv      STSVS
remote: VS    tSVSTSVST       SVSTS
remote: VS tSVSTSVSTSVSts     VSTSV
remote: VSTSVST     SVSTSVSTs VSTSV
remote: VSTSv          STSVSTSVSTSVS
remote:                  VSTSVSTSVST
remote:                    VSTSVSTs
remote:                      VSTs    (TM)
remote:
remote:  Microsoft (R) Visual Studio (R) Team Services
remote:
remote: Found 4 objects to send. (86 ms)
Unpacking objects: 100% (4/4), done.

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo
$
```

4. Write code

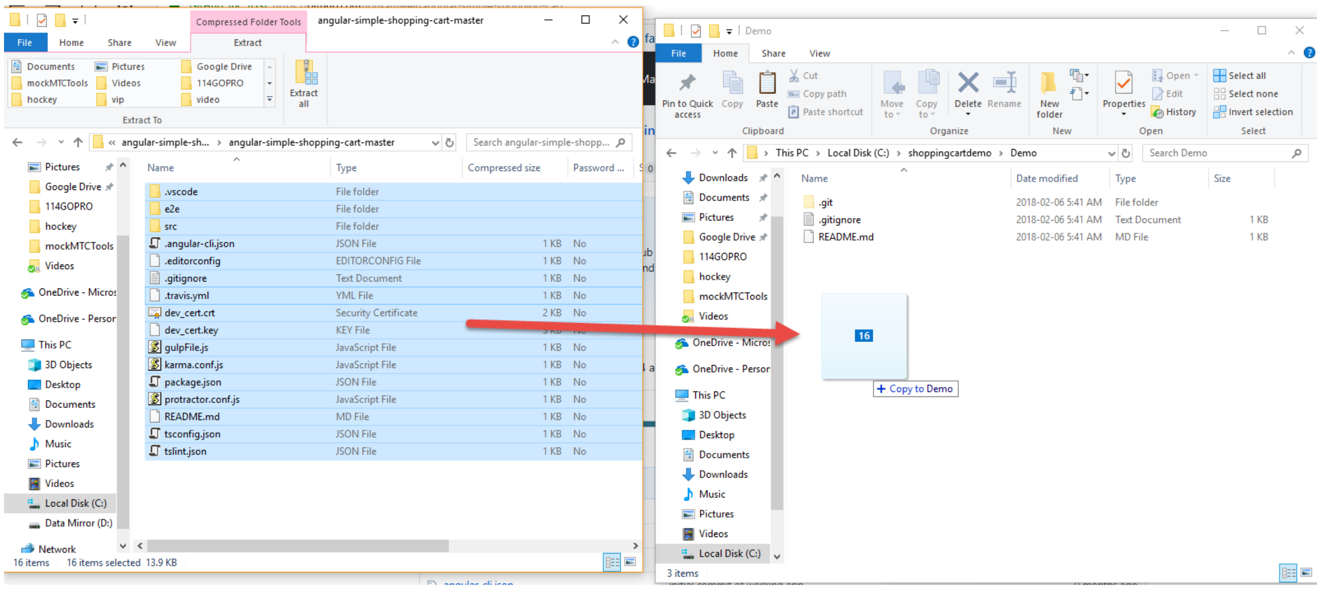
(not quite, we are just going to use an existing code base from GitHub and download the latest copy of the source to update our local repo).

1. Open the browser and navigate to https://github.com/jonsamwell/angular-simple-shopping-cart
2. Download code as follows:


jonsamwell / angular-simple-shopping-cart
Watch 1
Star 27
Fork 27
Code
Issues 0
Pull requests 0
Projects 0
Insights
Join GitHub today
GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.
Sign up
Dismiss
A simple shopping cart built with Angular 4 and RXJS
2 commits
2 branches
0 releases
1 contributor
Branch: master
New pull request
Find file
Clone or download
jonsamwell Updated build notification
.vscode initial commit of working app
e2e initial commit of working app
src initial commit of working app
angular-cli.json initial commit of working app
Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/jonsamwell/angular-simp
Open in Desktop
Download ZIP

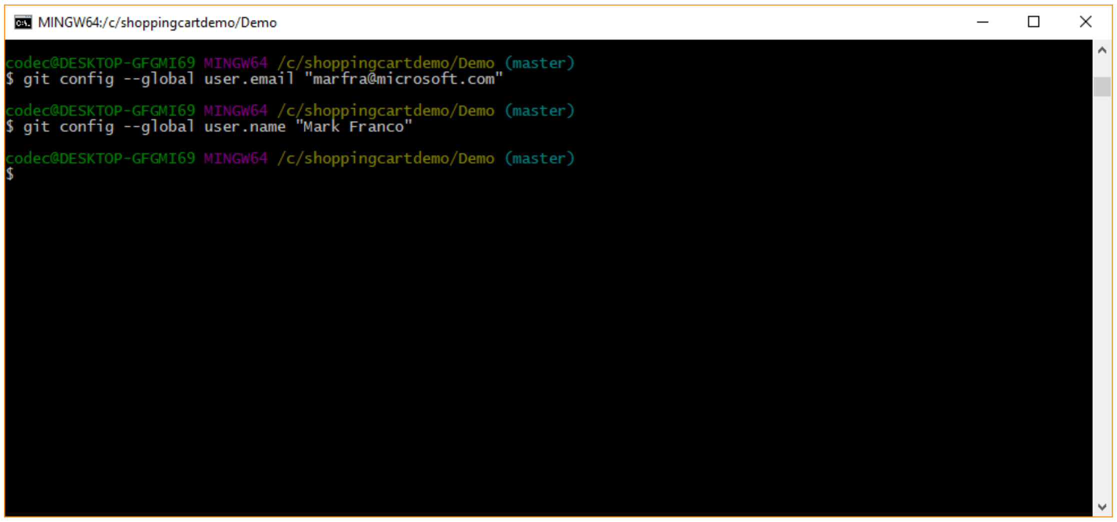
3. Extract the contents of the "angular-simple-shopping-cart-master" folder within the zip file to c:\shoppingcartdemo\demo

   Note: answer "replace" when duplicate files found.


Compressed Folder Tools
angular-simple-shopping-cart-master
Extract all
angular-simple-sh... > angular-simple-shopping-cart-master
Name
Type
Compressed size
Password ...
.vscode File folder
e2e File folder
src File folder
.angular-cli.json JSON File 1 KB No
.editorconfig EDITORCONFIG File 1 KB No
.gitignore Text Document 1 KB No
.travis.yml YML File 1 KB No
dev_cert.crt Security Certificate 2 KB No
dev_cert.key KEY File
gulpFile.js JavaScript File 1 KB No
karma.conf.js JavaScript File 1 KB No
package.json JSON File 1 KB No
protractor.conf.js JavaScript File 1 KB No
README.md MD File 1 KB No
tsconfig.json JSON File 1 KB No
tslint.json JSON File 1 KB No
16 items 16 items selected 13.9 KB
Demo
This PC > Local Disk (C:) > shoppingcartdemo > Demo
Name
Date modified
Type
Size
.git 2018-02-06 5:41 AM File folder
.gitignore 2018-02-06 5:41 AM Text Document 1 KB
README.md 2018-02-06 5:41 AM MD File 1 KB
16
+ Copy to Demo
3 items

4. Now we are going to add untracked files and commit our changes to our local repository, but before we can do that we have to tell Git who we are by issuing the two following commands:

```
git config --global user.email "you@outlook.com"
git config --global user.name "Your Name"
```

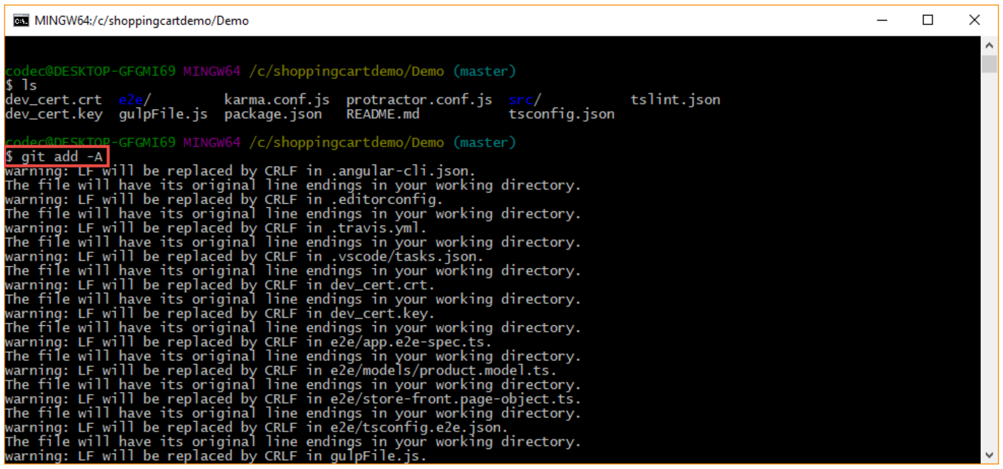


MINGW64:/c/shoppingcartdemo/Demo

```
codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ git config --global user.email "marfra@microsoft.com"

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ git config --global user.name "Mark Franco"

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$
```

5. Add untracked files as follows:

```
Cd \FirstApp
git add -A
```

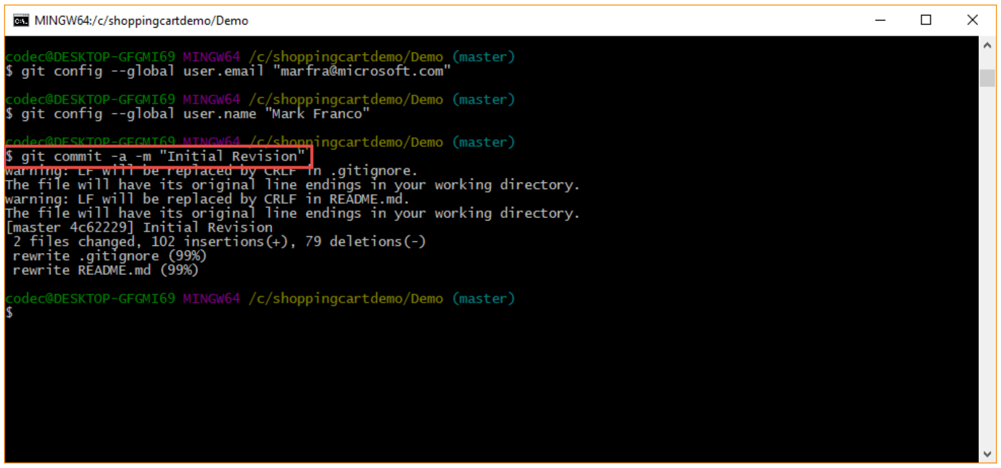


MINGW64:/c/shoppingcartdemo/Demo

```
codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ ls
dev_cert.crt  e2e/         karma.conf.js  protractor.conf.js  src/          tslint.json
dev_cert.key  gulpFile.js  package.json   README.md           tsconfig.json

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ git add -A
warning: LF will be replaced by CRLF in .angular-cli.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in .editorconfig.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in .travis.yml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in .vscode/tasks.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in dev_cert.crt.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in dev_cert.key.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in e2e/app.e2e-spec.ts.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in e2e/models/product.model.ts.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in e2e/store-front.page-object.ts.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in e2e/tsconfig.e2e.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in gulpFile.js.
```

6. Commit Changes:

```
git commit -a -m “Initial Revision”
```



MINGW64:/c/shoppingcartdemo/Demo

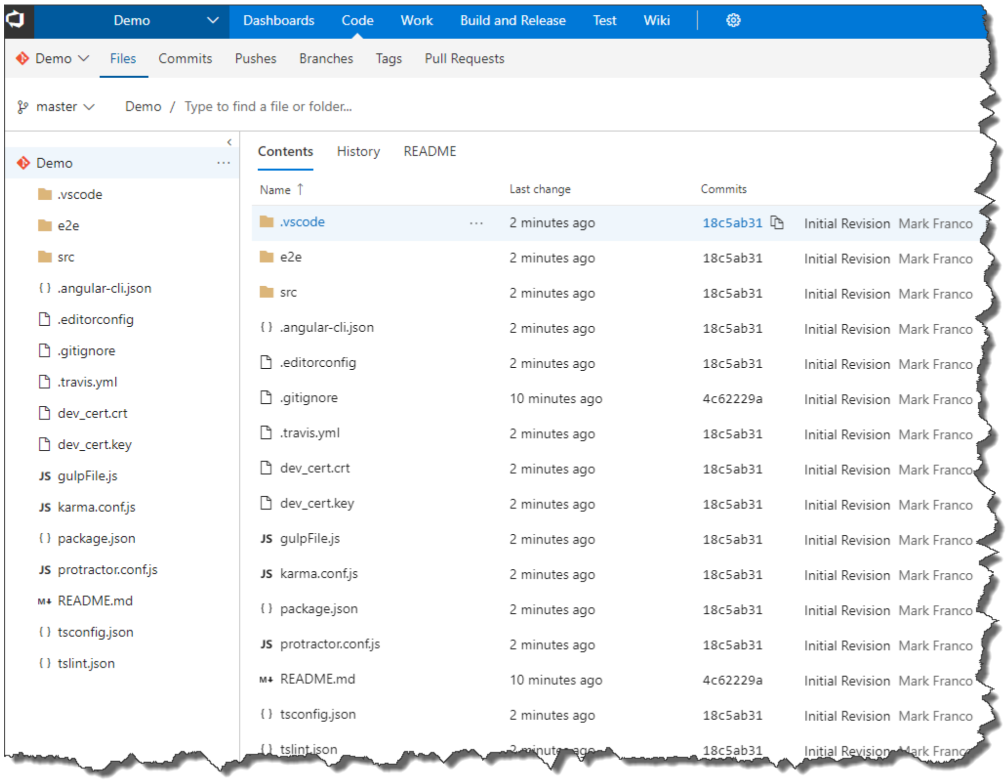
```
codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ git config --global user.email "marfra@microsoft.com"

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ git config --global user.name "Mark Franco"

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$ git commit -a -m "Initial Revision"
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
[master 4c62229] Initial Revision
 2 files changed, 102 insertions(+), 79 deletions(-)
 rewrite .gitignore (99%)
 rewrite README.md (99%)

codec@DESKTOP-GFGMI69 MINGW64 /c/shoppingcartdemo/Demo (master)
$
```

7. Push repository to VSTS into Master branch by executing the following command (no Screenshot):

```
Git push –repo <VSTS Git Repository url from previous steps>
```

i.e. git push –repo https://mtctor.visualstudio.com/_git/Demo

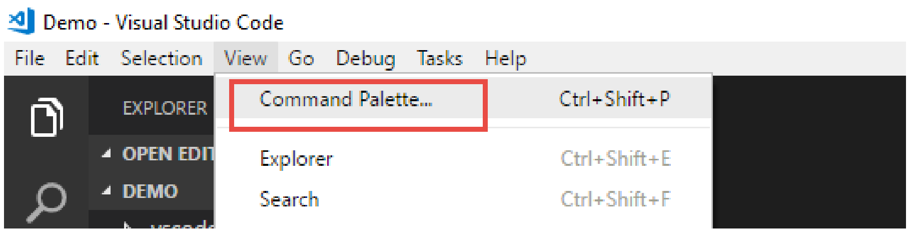
8. And Voila! You can now see your repository pushed up into VSTS:



Name	Last change	Commits		
.vscode	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
e2e	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
src	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
.angular-cli.json	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
.editorconfig	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
.gitignore	10 minutes ago	4c62229a	Initial Revision	Mark Franco
.travis.yml	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
dev_cert.crt	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
dev_cert.key	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
gulpFile.js	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
karma.conf.js	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
package.json	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
protractor.conf.js	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
README.md	10 minutes ago	4c62229a	Initial Revision	Mark Franco
tsconfig.json	2 minutes ago	18c5ab31	Initial Revision	Mark Franco
tslint.json	2 minutes ago	18c5ab31	Initial Revision	Mark Franco

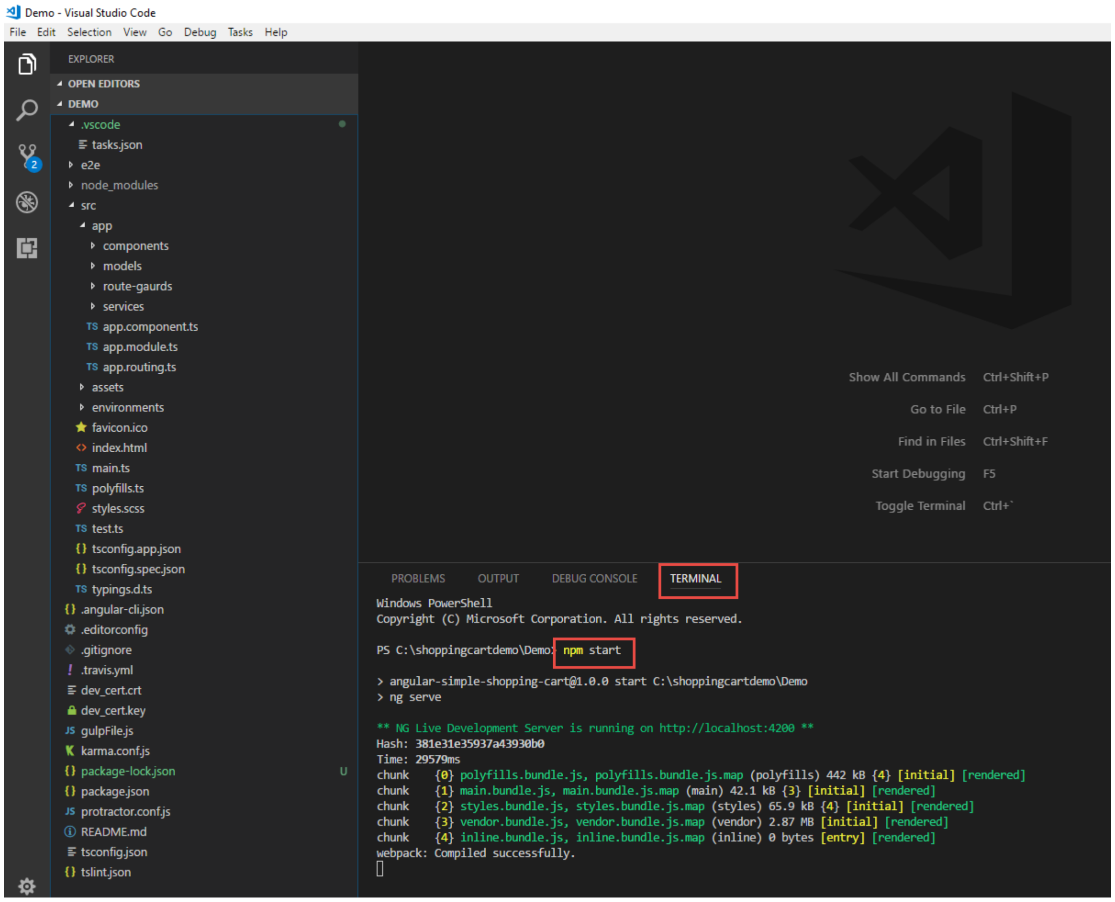
5. Launch VSCode and setup VSTS integration using the new experience

1. Open VSCode and Select File->Open folder: “C:\shoppingcartdemo\FirstApp”
2. Watch this step by step video on how to setup the new experience. Note: to open the Command Pallet as shown in the video, use the menu as follows:


Demo - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXPLORER
OPEN EDIT
DEMO
Command Palette... Ctrl+Shift+P
Explorer Ctrl+Shift+E
Search Ctrl+Shift+F

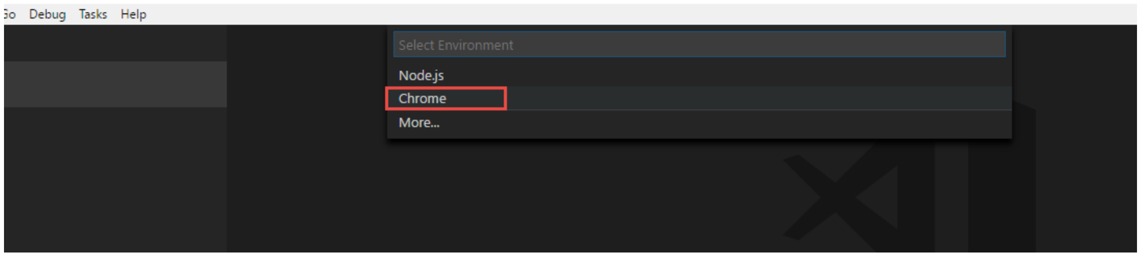
https://youtu.be/HnDNdm1WCIo?t=2m55s

3. Ensure all dependencies are current by running “npm install” in the VS Code terminal window
4. Run a local instance of the app to see how it runs by running “npm start” in the vscode terminal:

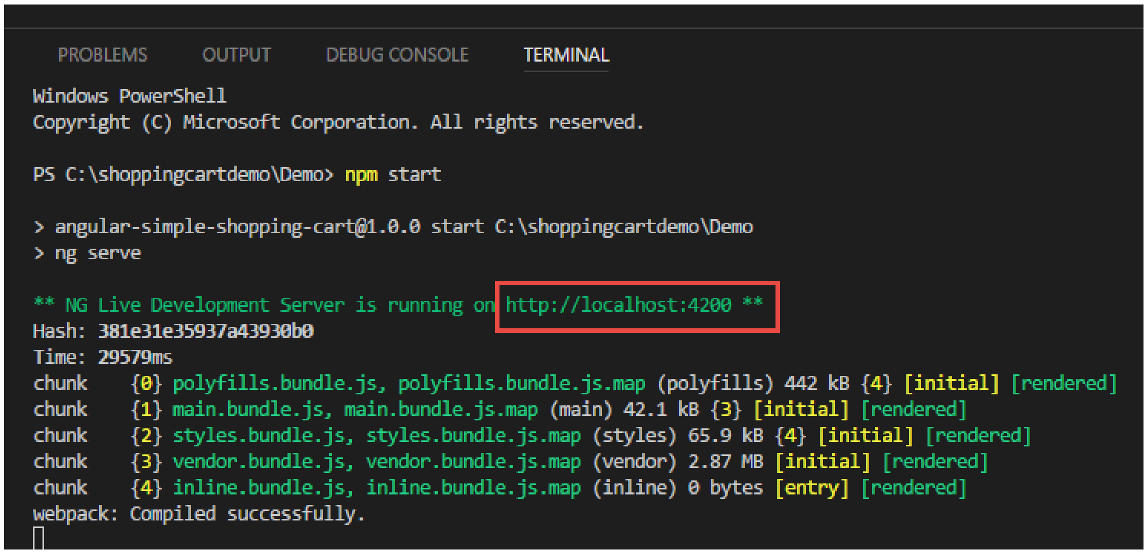


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\shoppingcartdemo\Demo> npm start

> angular-simple-shopping-cart@1.0.0 start C:\shoppingcartdemo\Demo
> ng serve

** NG Live Development Server is running on http://localhost:4200 **
Hash: 381e31e35937a43930b0
Time: 29579ms
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 442 kB {4} [initial] [rendered]
chunk    {1} main.bundle.js, main.bundle.js.map (main) 42.1 kB {3} [initial] [rendered]
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 65.9 kB {4} [initial] [rendered]
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.87 MB [initial] [rendered]
chunk    {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.
```

Your app is compiled and running under a node web server, but we need to add a launch file so we can launch a debugger window using Chrome. We do so by creating a new configuration file by selecting the "Debug→Add Configuration" menu item and selecting "Chrome" from the drop down.

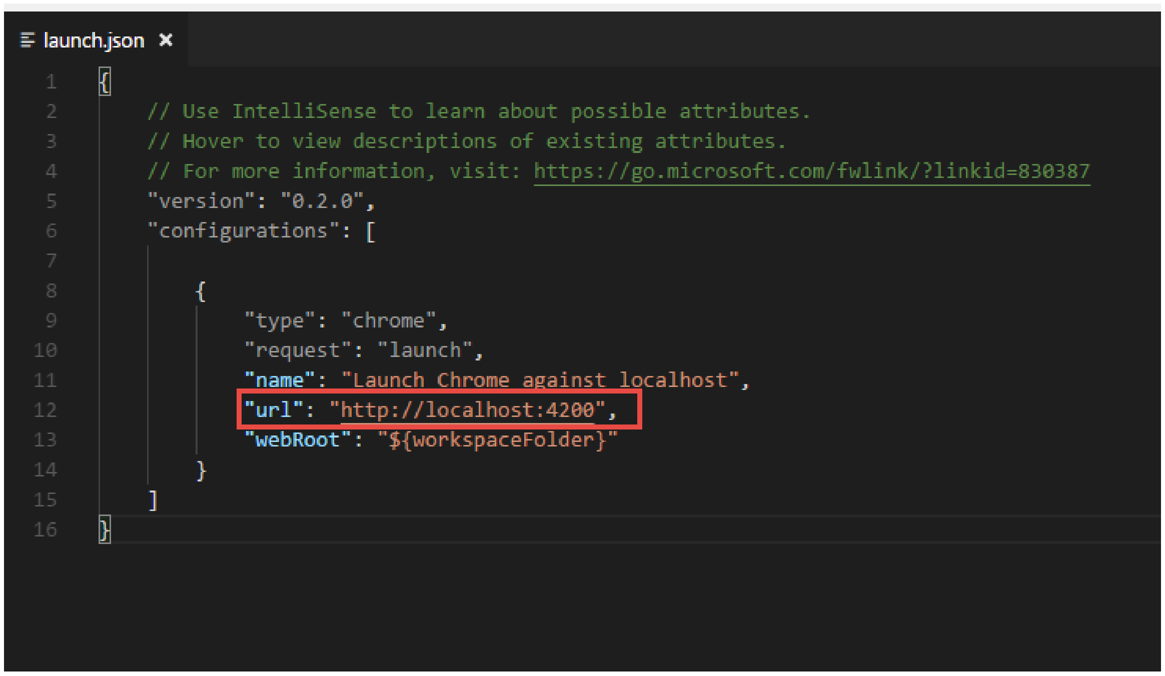


We need to ensure the new launch.json file is pointing to the correct url. Node will automatically assign a random port on your computer to host your angular application on and you can get this url from the previous step where you ran "NPM Start":



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\shoppingcartdemo\Demo> npm start

> angular-simple-shopping-cart@1.0.0 start C:\shoppingcartdemo\Demo
> ng serve

** NG Live Development Server is running on http://localhost:4200 **
Hash: 381e31e35937a43930b0
Time: 29579ms
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 442 kB {4} [initial] [rendered]
chunk    {1} main.bundle.js, main.bundle.js.map (main) 42.1 kB {3} [initial] [rendered]
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 65.9 kB {4} [initial] [rendered]
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.87 MB [initial] [rendered]
chunk    {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.
```

With this url , you are going to update the launch.json file and specifically update the "url" property of the Chrome configuration as such:



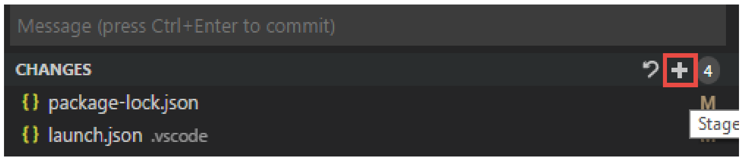
```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [

        {
            "type": "chrome",
            "request": "launch",
            "name": "Launch Chrome against localhost",
            "url": "http://localhost:4200",
            "webRoot": "${workspaceFolder}"
        }
    ]
}
```

Now click on Debug→Start debugging

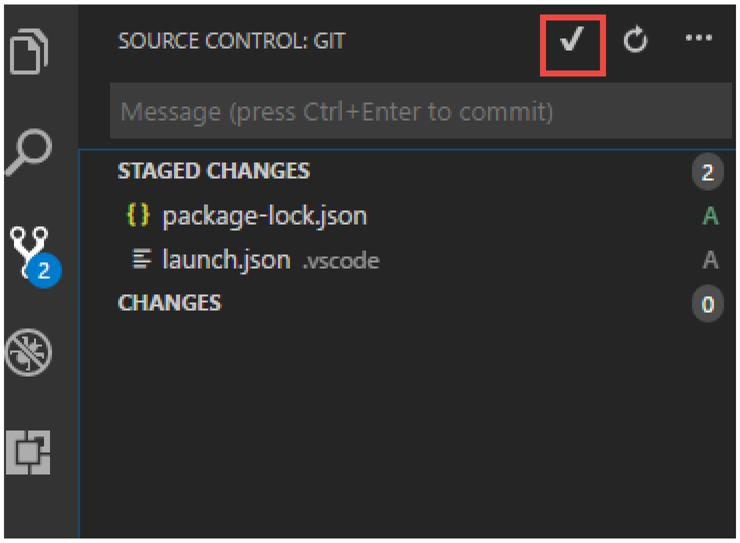
Final house keeping bits:

Check in your additional file “Launch.json” using the VSCODE IDE now:

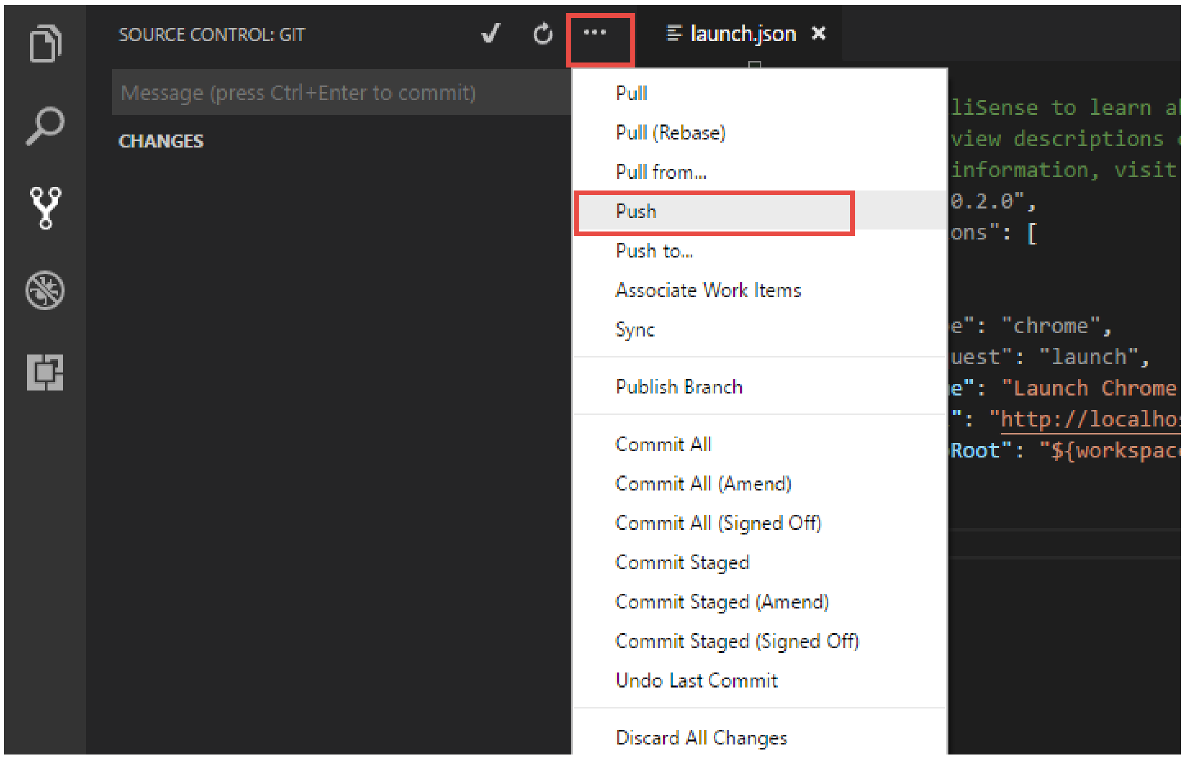
Add Files to local repository (Stage)



Commit Changes to local Repository



Push Changes from local repository to VSTS

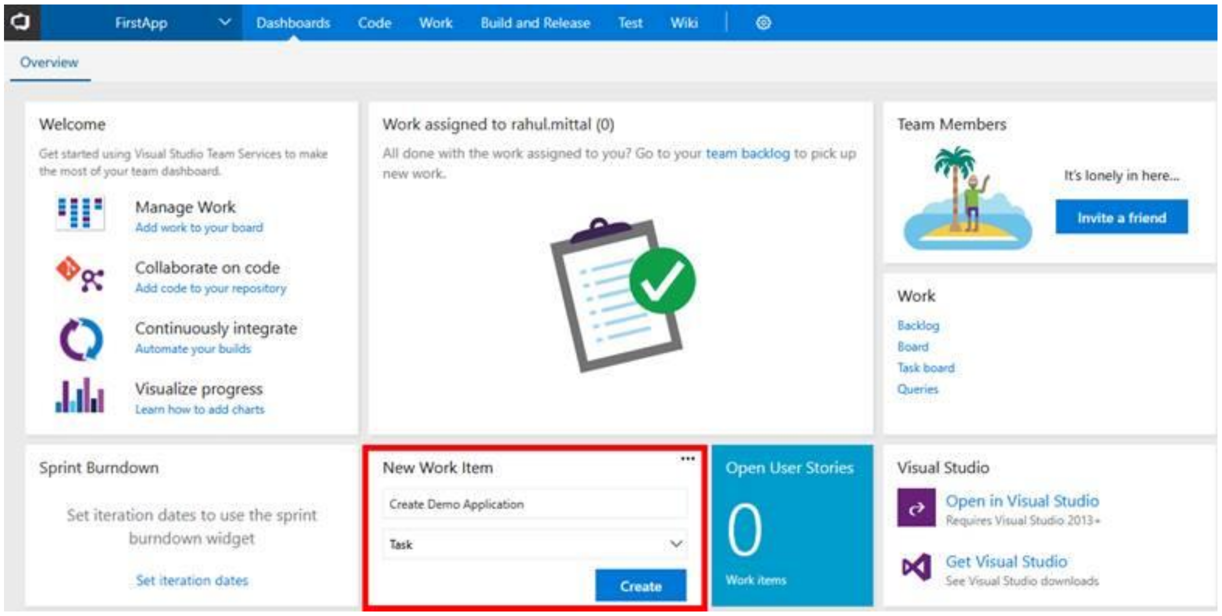


```
liSense to learn a
view descriptions
information, visit
0.2.0",
ons": [

e": "chrome",
uest": "launch",
e": "Launch Chrome
": "http://localho
Root": "${workspac
```

Development Complete…

6 Setting Up Work Item Check-in and Build Configuration

Go to VSTS dashboard and create a task. We will associate this task with check-in.


FirstApp | Dashboards | Code | Work | Build and Release | Test | Wiki

Overview

Welcome

Get started using Visual Studio Team Services to make the most of your team dashboard.

Manage Work — Add work to your board

Collaborate on code — Add code to your repository

Continuously integrate — Automate your builds

Visualize progress — Learn how to add charts

Work assigned to rahul.mittal (0)

All done with the work assigned to you? Go to your team backlog to pick up new work.

Team Members

It's lonely in here...

Invite a friend

Work: Backlog, Board, Task board, Queries

Sprint Burndown

Set iteration dates to use the sprint burndown widget

Set iteration dates

New Work Item

Create Demo Application

Task

Create

Open User Stories

0

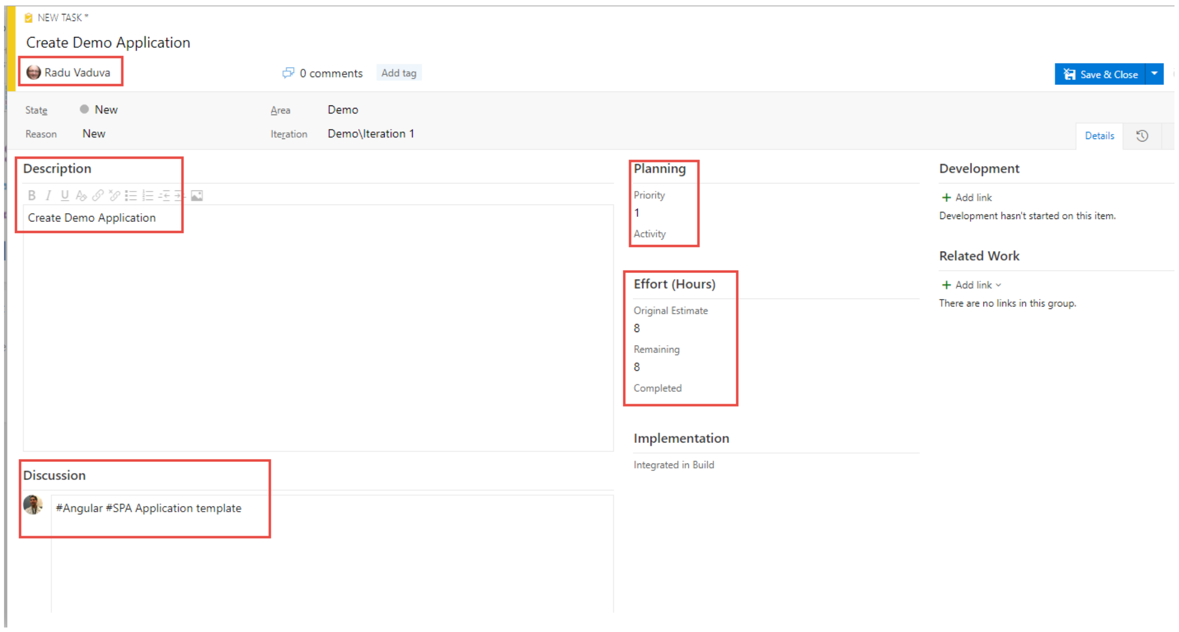
Work items

Visual Studio

Open in Visual Studio — Requires Visual Studio 2013+

Get Visual Studio — See Visual Studio downloads

Assign a task to a resource (Yourself in this case), enter description, set priority, and specify effort. Click Save and Close.


NEW TASK *

Create Demo Application

Radu Vaduva

0 comments | Add tag | Save & Close

State	New	Area	Demo
Reason	New	Iteration	Demo\Iteration 1

Description

Create Demo Application

Planning

Priority: 1

Activity:

Effort (Hours)

Original Estimate: 8

Remaining: 8

Completed:

Implementation

Integrated in Build

Development

Add link

Development hasn't started on this item.

Related Work

Add link

There are no links in this group.

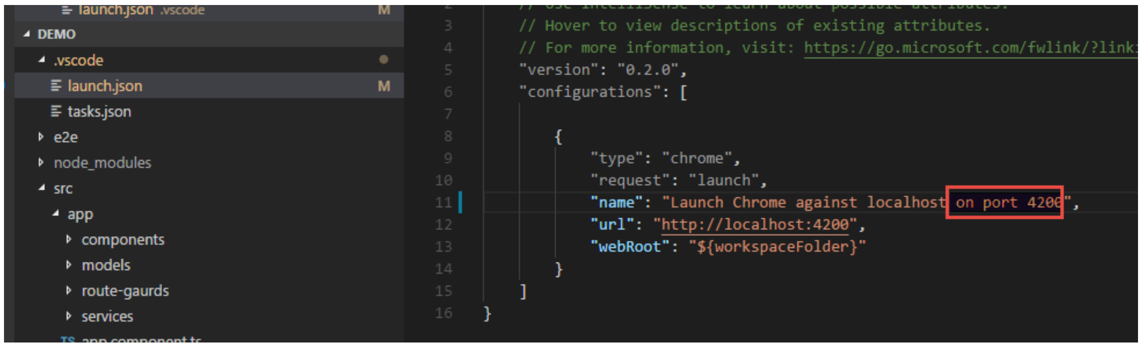
Discussion

#Angular #SPA Application template

When user saves, unique task number is assigned to each task.

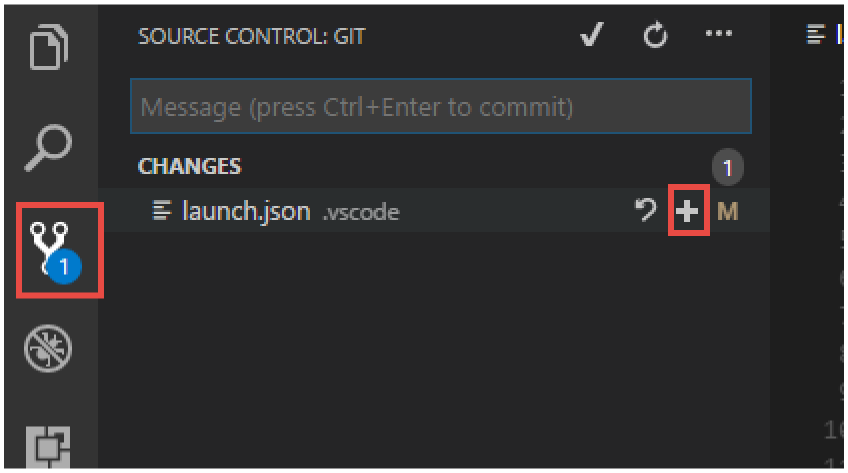
Go back to VSCODE, make changes to the launch.json file and associate the work item while committing the code.

Make the code change by appending "on port 4200" as shown below:

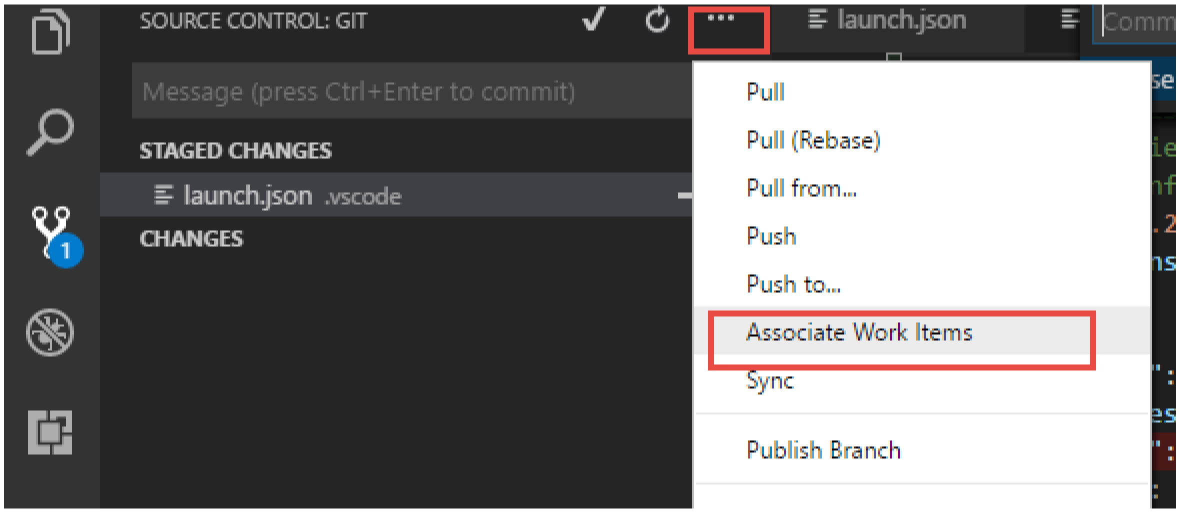


```
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?link
    "version": "0.2.0",
    "configurations": [

        {
            "type": "chrome",
            "request": "launch",
            "name": "Launch Chrome against localhost on port 4200",
            "url": "http://localhost:4200",
            "webRoot": "${workspaceFolder}"
        }
    ]
}
```

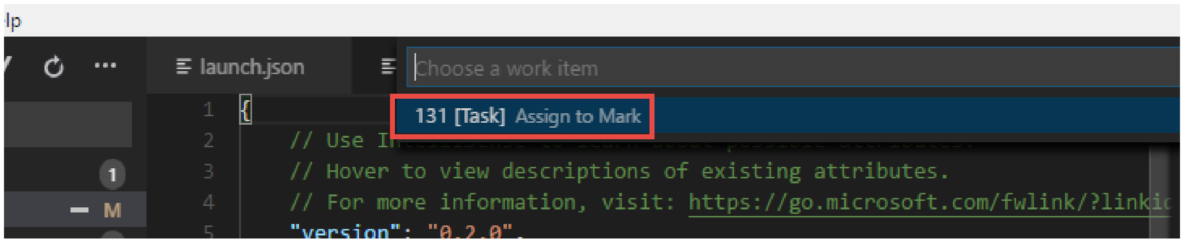
Add Change (Stage)



Commit Change by Associating work item:



Select Work Item task:



```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid
    "version": "0.2.0",
```

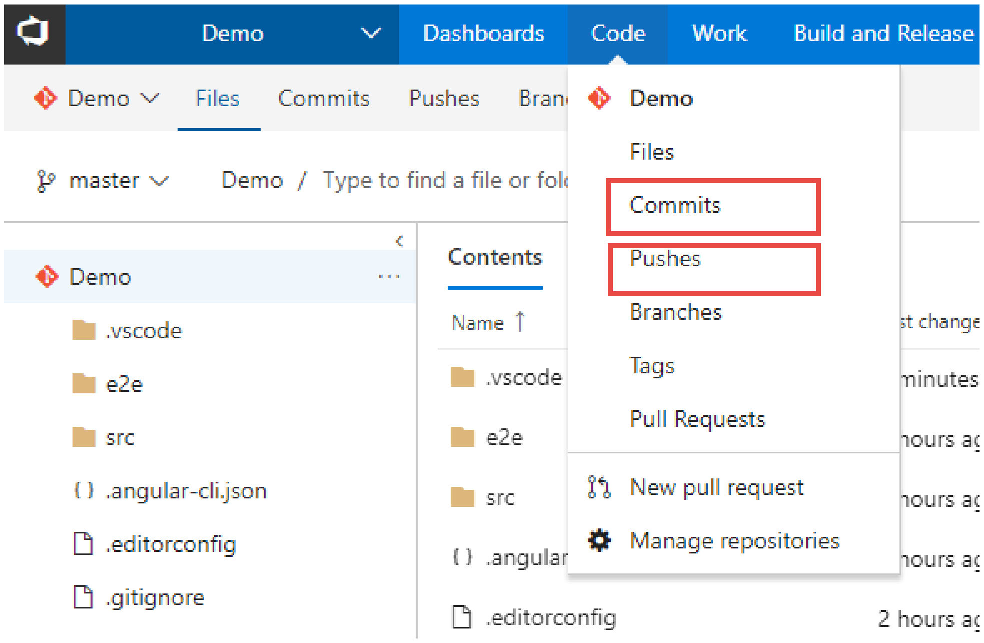
Commit Change with a message "Added port "

Push Change to VSTS.

Done.

When we again go to task board in VSTS, we can see development history associated with this item.

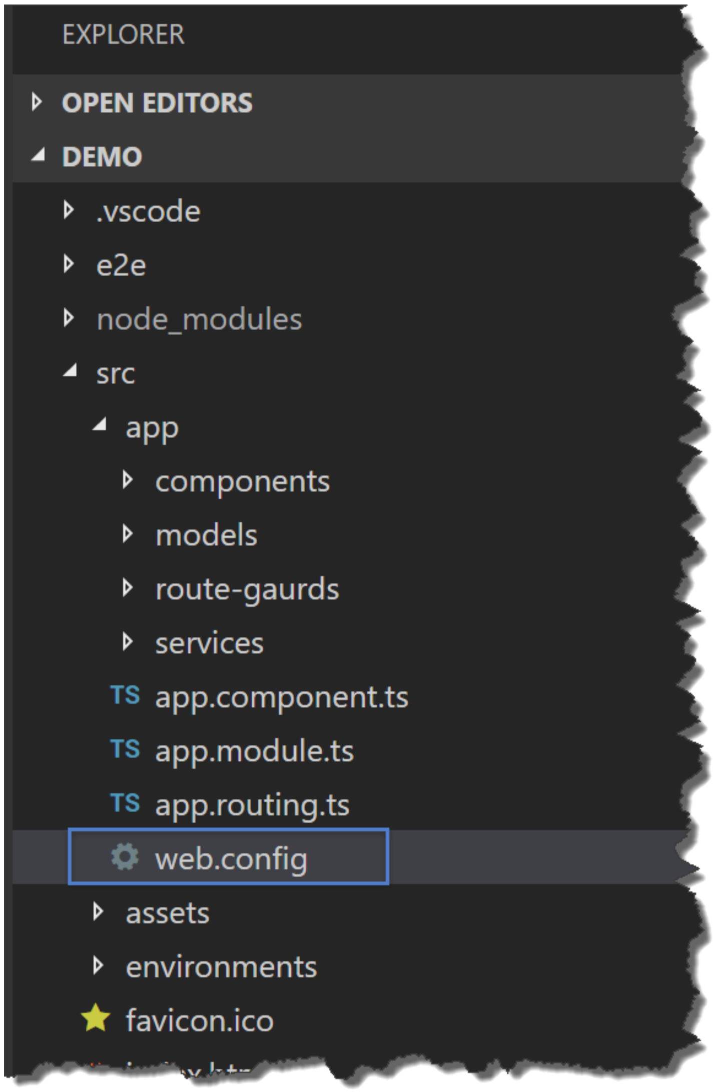
Check out here:



Deploy to Azure App Services

We will need to add a web.config file to instruct our underlying web server on Azure to rewrite all incoming request to serve our *index.html* file.

Create a new file named web.config in src\app\ by right-clicking on src\app folder and selecting 'New File'



Add the following contents to the web.config:

```
<configuration>

    <system.webServer>
        <staticContent>
            <mimeMap fileExtension=".json" mimeType="application/json" />
        </staticContent>

        <rewrite>
        <rules>
        <clear />

        <!-- ignore static files -->
        <rule name="AngularJS Conditions" stopProcessing="true">
        <match url="(assets/.*|.js|.css)" />
        <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
        <action type="None" />
        </rule>

        <!-- check if its root url and navigate to default page -->
        <rule name="Index Request" enabled="true" stopProcessing="true">
        <match url="^$" />
        <action type="Redirect" url="/home" logRewrittenUrl="true" />
        </rule>

        <!--remaining all other url's point to index.html file -->
        <rule name="AngularJS Wildcard" enabled="true">
        <match url="(.*)" />
        <conditions logicalGrouping="MatchAll" trackAllCaptures="false" />
        <action type="Rewrite" url="index.html" />
        </rule>

        </rules>
        </rewrite>
    </system.webServer>
</configuration>
```

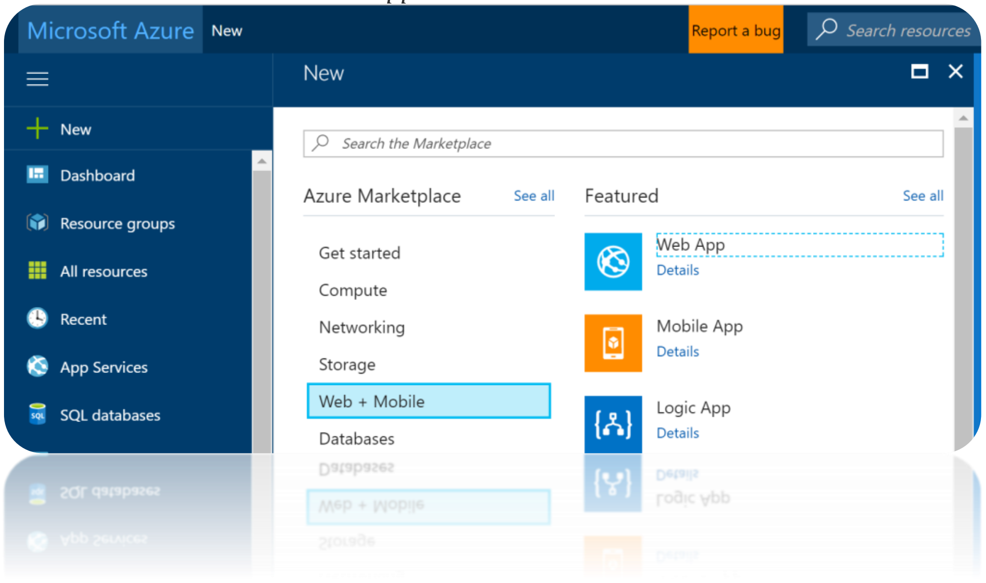
Modify the /gulpfile.js as follows to remove the code that modifies the index.html `<base href="/">` element. The developer added this code but it is no longer needed as you can leverage angular CLI to modify this directly. Also we have added a copy process to deploy the web.config to the distribution folder:

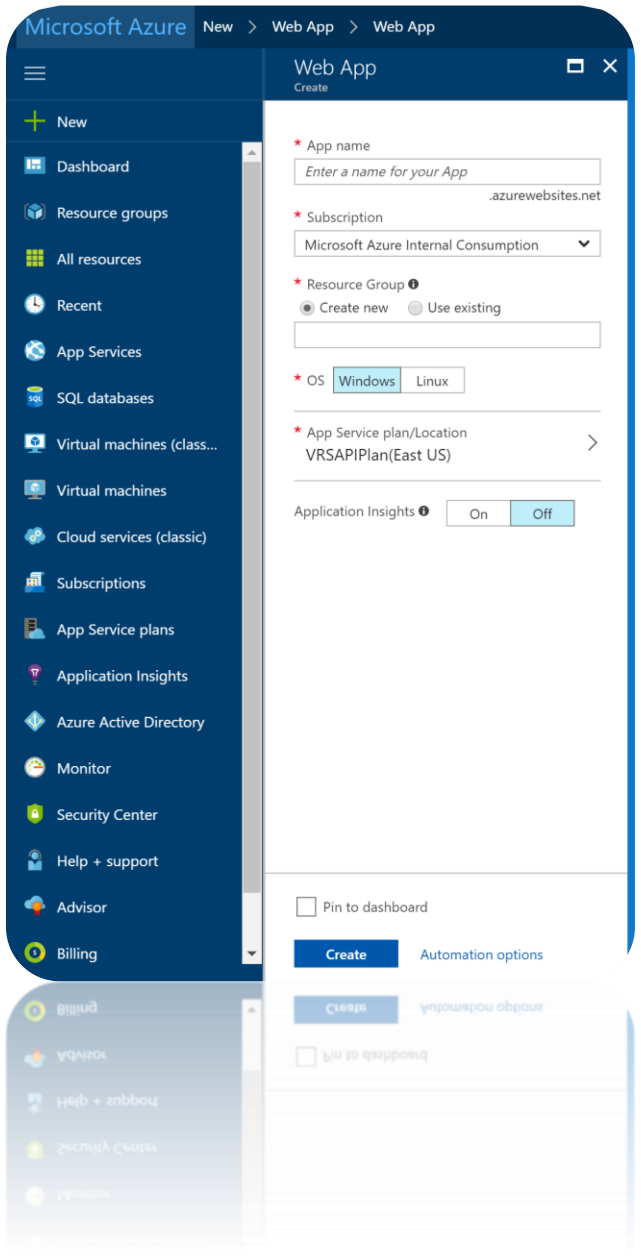
```
var gulp = require('gulp');

var replace = require('gulp-replace');
var htmlmin = require('gulp-htmlmin');

gulp.task('js:minify', function () {
  gulp.src(["./dist/main.*.js", "./dist/polyfills.*.js",
"./dist/inline.*.js"])
    .pipe(replace(/\/\*([\s\S]*?)\*\/[\s\S]?/g, ""))
    .pipe(gulp.dest("./dist"));
});

gulp.task('web:config', function () {
  gulp.src(["./src/app/web.config"])
    .pipe(gulp.dest("./dist"));
});


gulp.task("html:minify", function () {
  return gulp.src('dist/*.html')
    .pipe(htmlmin({ collapseWhitespace: true }))
    .pipe(gulp.dest('./dist'));
});

gulp.task("default", ["js:minify", "html:minify", "web:config"]);
```

Create the Azure App Service

The next step is to create an Azure Web App which will host our Angular application. You can sign up for a free or paid account and log in the Azure portal.

1. *New -> Web and Mobile -> Web App*





Web App
Create

* App name
Enter a name for your App
.azurewebsites.net

* Subscription
Microsoft Azure Internal Consumption

* Resource Group
[✓ Create new] Use existing

* OS
[✓ Windows] Linux

* App Service plan/Location
VRSAPIPlan(East US)

Application Insights
On [✓ Off]

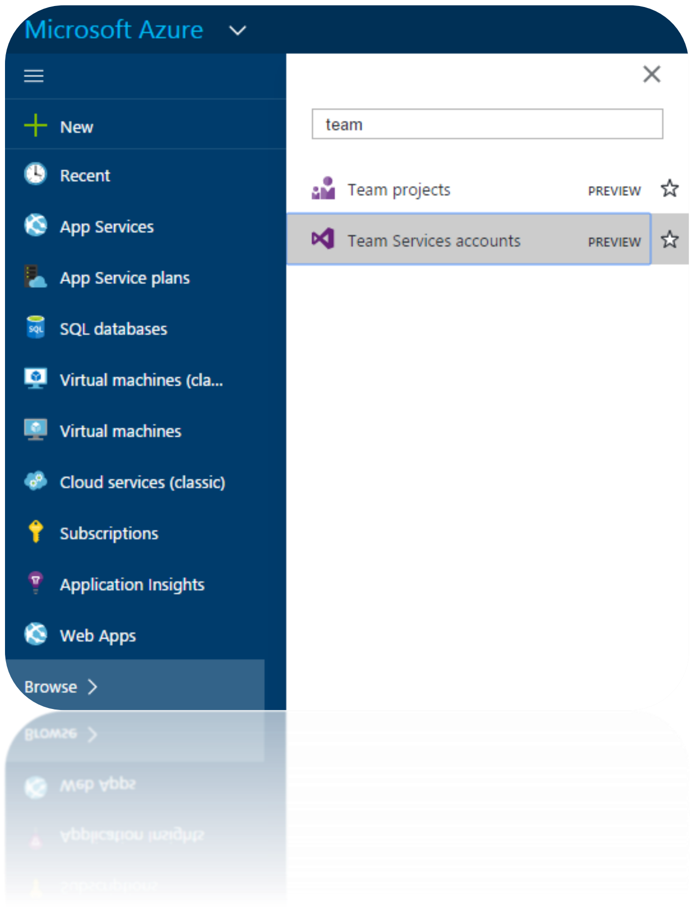
☐ Pin to dashboard

2. Fill out the required fields:
3. After pressing "Create" you should now have an Azure Web App created.

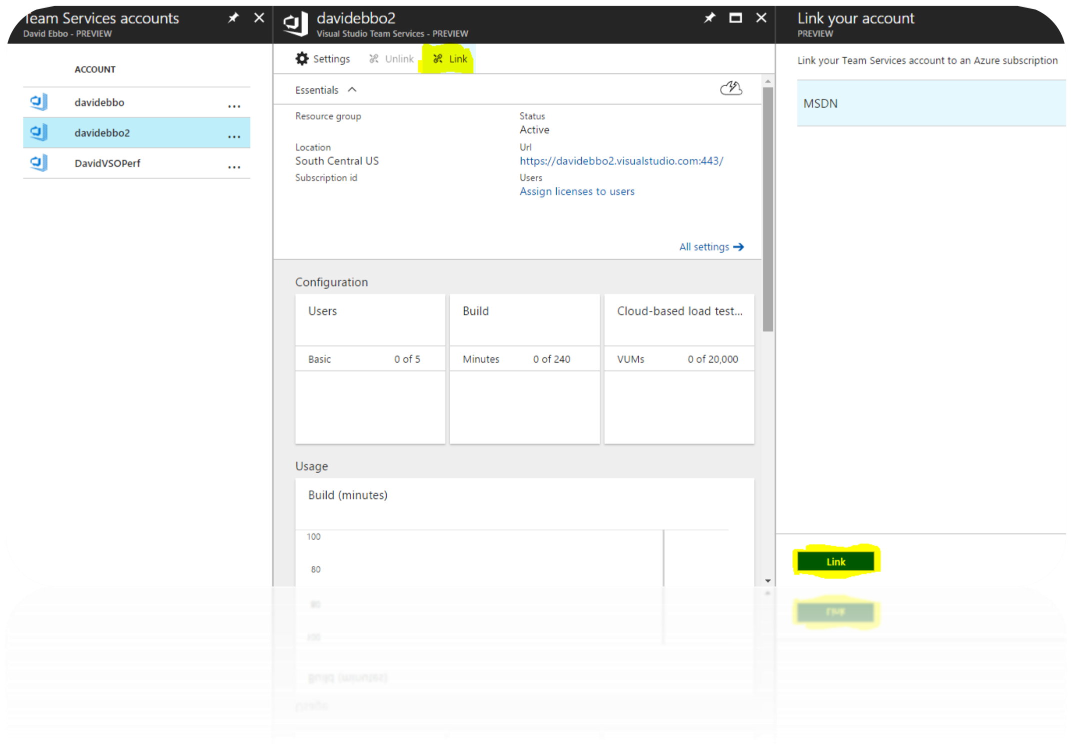
Linking your VSTS account to your Azure subscription

The last step is that you need to link your VSTS account to your Azure subscription (see also this post on this topic).

To do this, go to the Azure Portal, click More Services (image says 'Browse' but that was the old name) and search for 'Team':



Now select the relevant Team Services account, click Link button at the top, and then the Link button in the other blade:

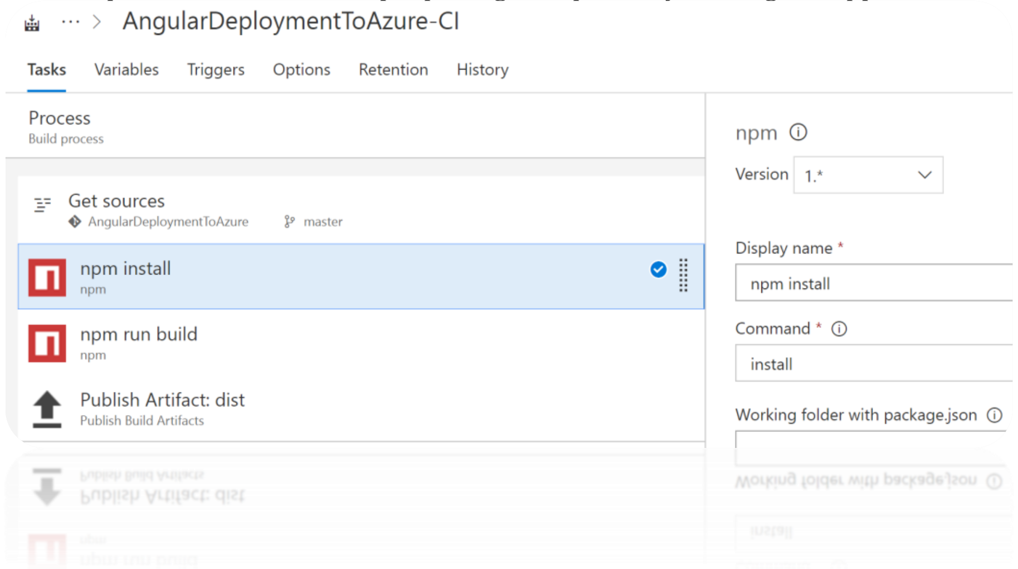


And you're done! You will now be able to set up continuous deploying to your git repos hosted in VSTS.

Setting Up CI Pipeline With VSTS

In the next steps we will set up our VSTS CI/CD pipeline to push the Angular application to the newly created Azure Web App. Start by creating a new build definition under VSTS:

1. *Build and Release -> Builds -> New*
2. Add an npm task to install the npm packages required by the Angular application



npm

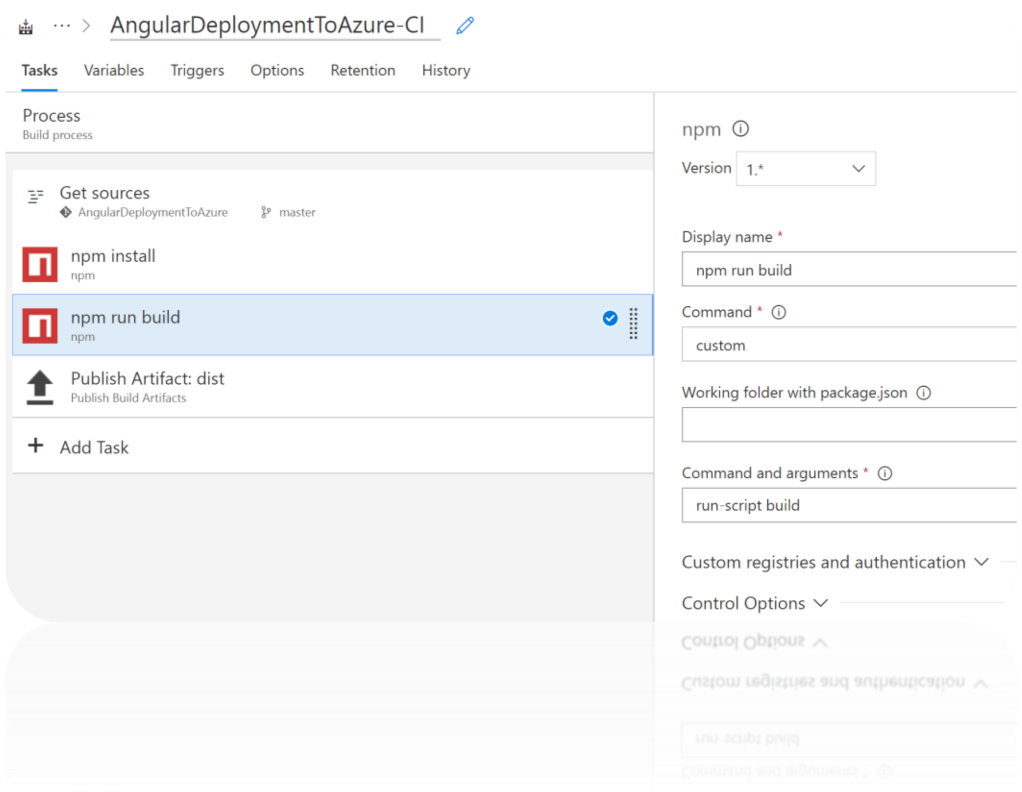
Version: 1.*

Display name *: npm install

Command *: install

Working folder with package.json:

3. Add another npm task to build the application and create the dist folder:



npm

Version: 1.*

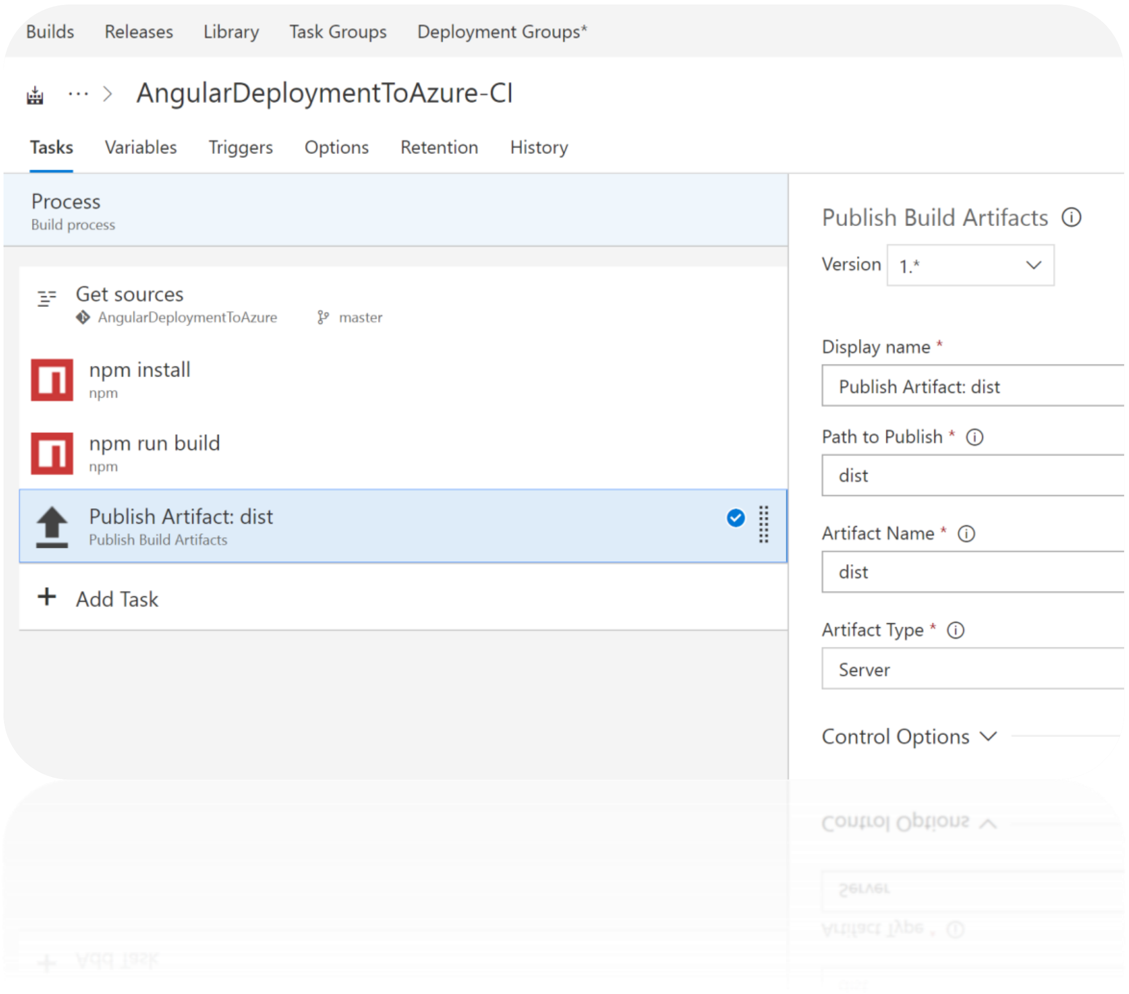
Display name *: npm run build

Command *: custom

Working folder with package.json:

Command and arguments *: run-script build

4. Add a publish artifact task that generates the dist artifact which will be provided later on as an input to our release definition:



Publish Build Artifacts

Version: 1.*

Display name *: Publish Artifact: dist

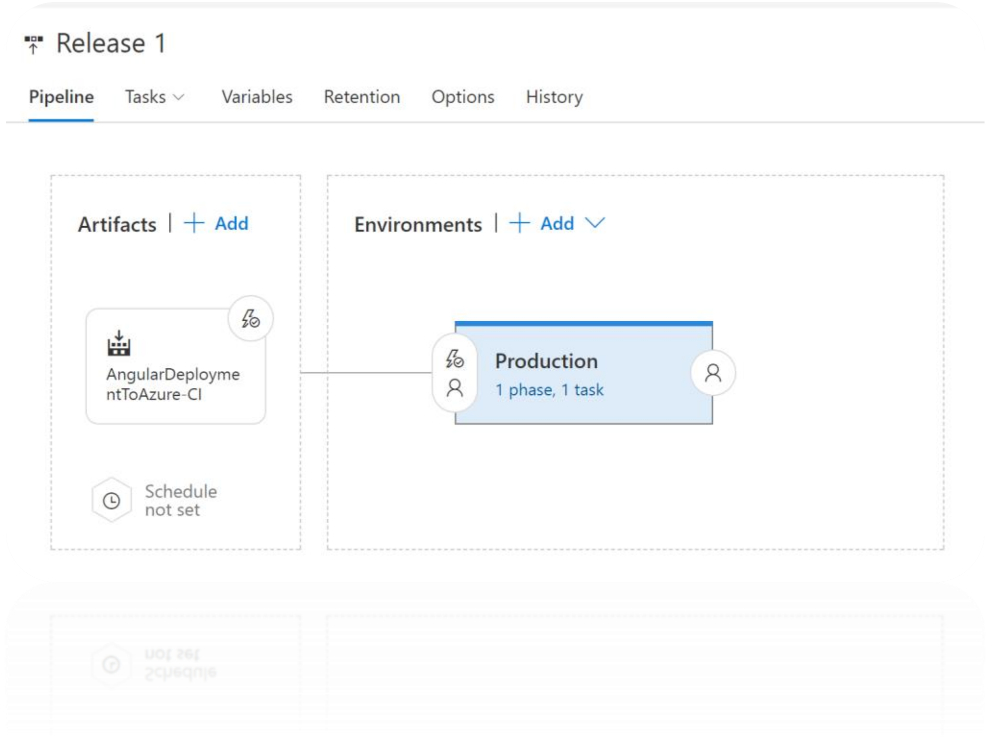
Path to Publish *: dist

Artifact Name *: dist

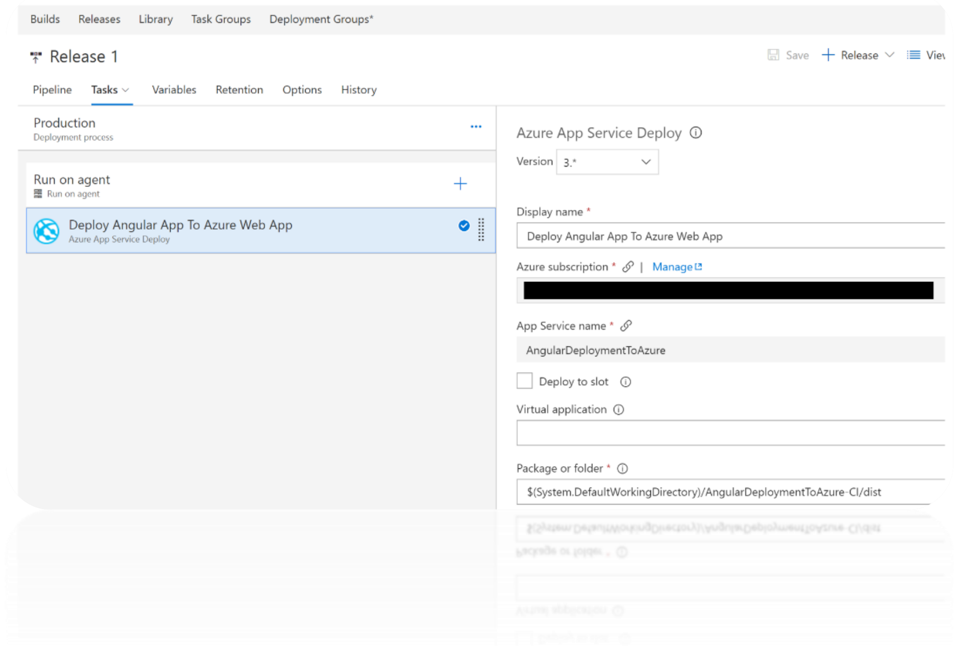
Artifact Type *: Server

Setting Up CD Pipeline With VSTS

The last step is to add a CD pipeline which will deploy the artifacts created by the build to the Azure Web App. In this demo I am keeping the release pipeline simple by deploying the artifacts directly to production. In a real life application you will probably create multiple environments before releasing to production (Development, QA, Staging, etc.):



The production environment includes a single task that deploys the Angular application to an Azure Web App:



Azure App Service Deploy

Version: 3.*

Display name *: Deploy Angular App To Azure Web App

Azure subscription * | Manage: [redacted]

App Service name *: AngularDeploymentToAzure

Deploy to slot: ☐

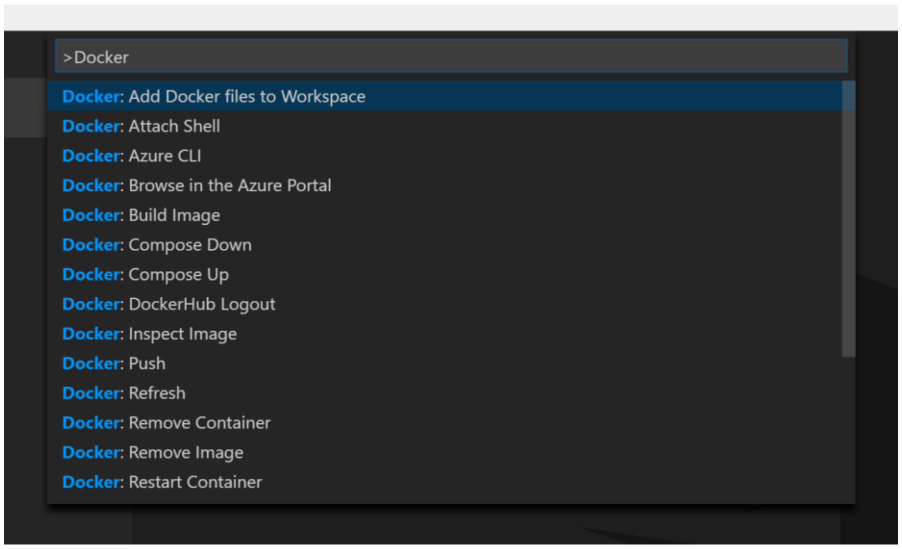
Virtual application:

Package or folder *: $(System.DefaultWorkingDirectory)/AngularDeploymentToAzure-CI/dist

That's it! You now have a fully functional CI/CD pipeline that will deploy your Angular application to an Azure Web App the next time you check in your code.

Bonus: Deploy your SPA to a Linux VM

1. Add Docker Files to workspace like so:



```
>Docker
Docker: Add Docker files to Workspace
Docker: Attach Shell
Docker: Azure CLI
Docker: Browse in the Azure Portal
Docker: Build Image
Docker: Compose Down
Docker: Compose Up
Docker: DockerHub Logout
Docker: Inspect Image
Docker: Push
Docker: Refresh
Docker: Remove Container
Docker: Remove Image
Docker: Restart Container
```

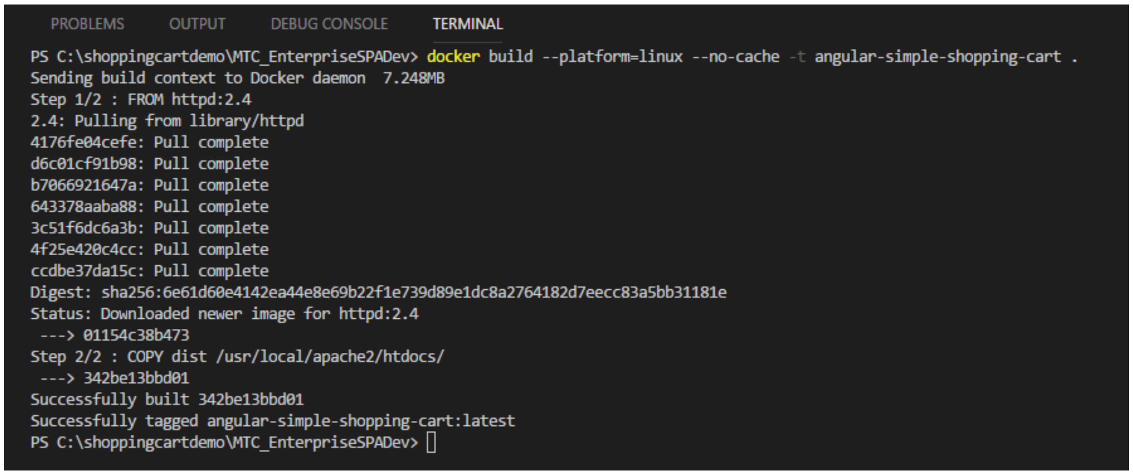
When Prompted Select: node.js and then set the Port to 4200. This is just to create the base implementation of the Docker image files, but we will replace the contents with our own commands for our SPA to work in a simple Apache Web Server image provided by the image library on the Docker public registry.

2. Once the DockerFile is added to your Angular application its time to add the necessary commands to assemble a docker image which will be used to create docker containers that will run on both the development machine as well as on the production server. We will assume that Apache 2.4 will be used as the web server.
3. Modify the contents of the DockerFile so that it builds an image based on the httpd:2.4 Docker Public image and copies the dist folder that is generated by the angular build process into the specified directory inside the image. Overwrite the DockerFile with the code below:

```
FROM httpd:2.4
COPY dist /usr/local/apache2/htdocs/
```

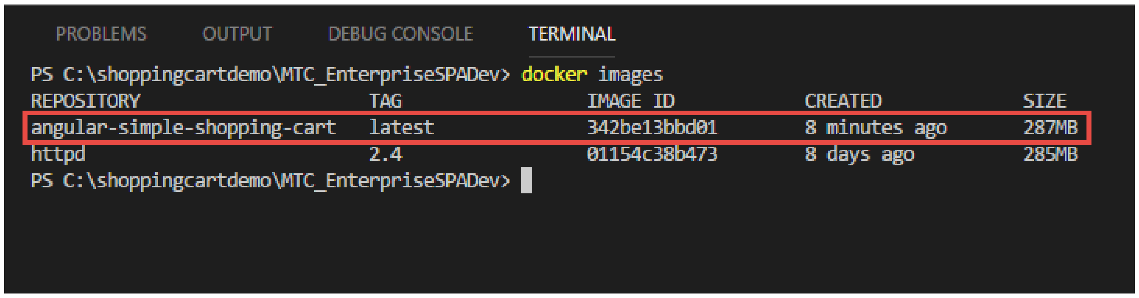
4. In the terminal Run: "npm run build"

5. In the terminal Run: “docker build --platform=linux --no-cache -t angular-simple-shopping-cart .”



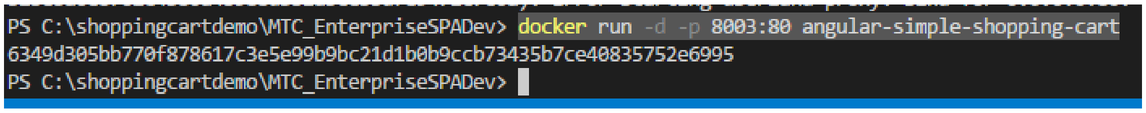
```
PS C:\shoppingcartdemo\MTC_EnterpriseSPADev> docker build --platform=linux --no-cache -t angular-simple-shopping-cart .
Sending build context to Docker daemon  7.248MB
Step 1/2 : FROM httpd:2.4
2.4: Pulling from library/httpd
4176fe04cefe: Pull complete
d6c01cf91b98: Pull complete
b7066921647a: Pull complete
643378aaba88: Pull complete
3c51f6dc6a3b: Pull complete
4f25e420c4cc: Pull complete
ccdbe37da15c: Pull complete
Digest: sha256:6e61d60e4142ea44e8e69b22f1e739d89e1dc8a2764182d7eecc83a5bb31181e
Status: Downloaded newer image for httpd:2.4
 ---> 01154c38b473
Step 2/2 : COPY dist /usr/local/apache2/htdocs/
 ---> 342be13bbd01
Successfully built 342be13bbd01
Successfully tagged angular-simple-shopping-cart:latest
PS C:\shoppingcartdemo\MTC_EnterpriseSPADev>
```

6. In the Terminal Run: “docker images”. You will see two images. The HTTPD which is your base image that was downloaded from the Docker registry and the angular-simple-shopping-cart image you just built.



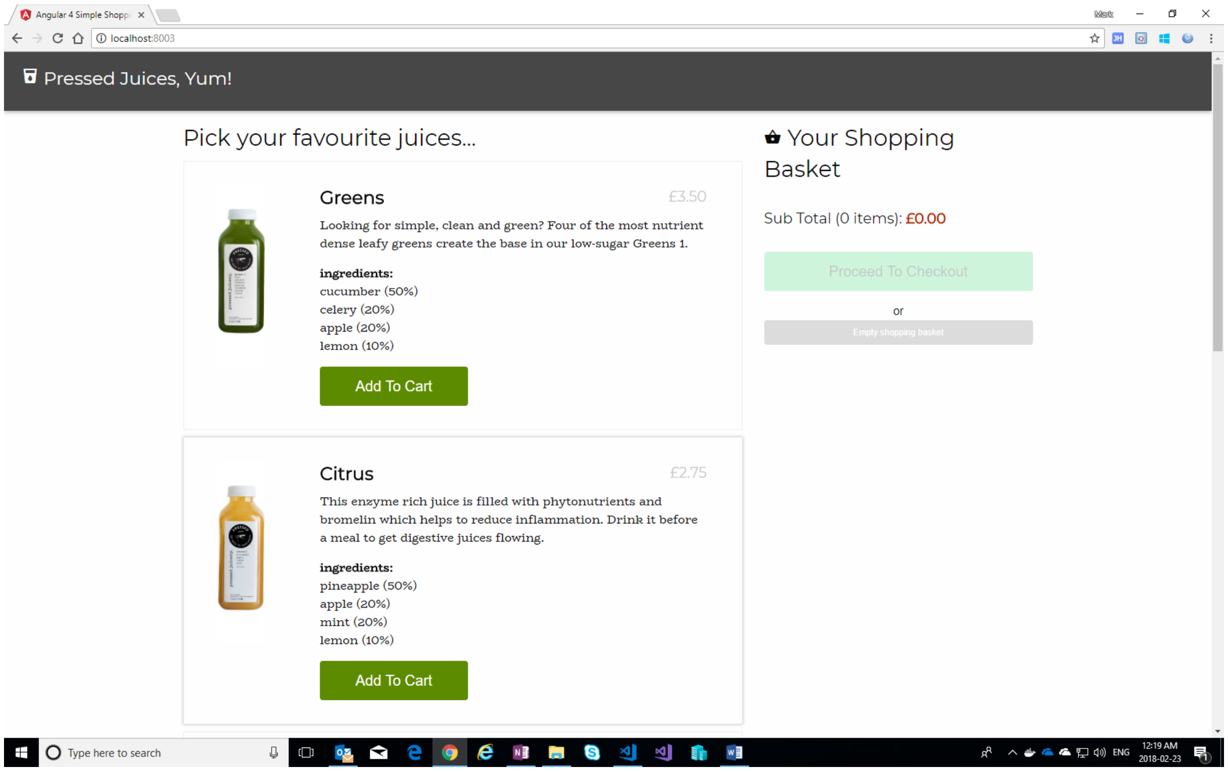
```
PS C:\shoppingcartdemo\MTC_EnterpriseSPADev> docker images
REPOSITORY                      TAG                 IMAGE ID            CREATED             SIZE
angular-simple-shopping-cart    latest              342be13bbd01        8 minutes ago       287MB
httpd                           2.4                 01154c38b473        8 days ago          285MB
PS C:\shoppingcartdemo\MTC_EnterpriseSPADev>
```

7. In the Terminal Run: “docker run -d -p 8003:80 angular-simple-shopping-cart”. Where 8003 is the port that will be used outside of the container and 80 is the port being exposed inside the container.



```
PS C:\shoppingcartdemo\MTC_EnterpriseSPADev> docker run -d -p 8003:80 angular-simple-shopping-cart
6349d305bb770f878617c3e5e99b9bc21d1b0b9ccb73435b7ce40835752e6995
PS C:\shoppingcartdemo\MTC_EnterpriseSPADev>
```

8. Open a web browser and navigate to: "localhost:8003". If successful, you should see our Angular SPA running locally as such:



Pick your favourite juices...

Greens £3.50

Looking for simple, clean and green? Four of the most nutrient dense leafy greens create the base in our low-sugar Greens 1.

ingredients:
cucumber (50%)
celery (20%)
apple (20%)
lemon (10%)

Add To Cart

Citrus £2.75

This enzyme rich juice is filled with phytonutrients and bromelin which helps to reduce inflammation. Drink it before a meal to get digestive juices flowing.

ingredients:
pineapple (50%)
apple (20%)
mint (20%)
lemon (10%)

Add To Cart

Your Shopping Basket

Sub Total (0 items): £0.00

Proceed To Checkout

or

Empty shopping basket

9. Next, we will deploy to azure App Services (Linux).

   https://blogs.msdn.microsoft.com/wael-kdouh/2018/01/02/deploying-your-dockerized-angular-application-to-azure-using-vsts/