PRACTICAL NO - 02

Develop Spring Boot API

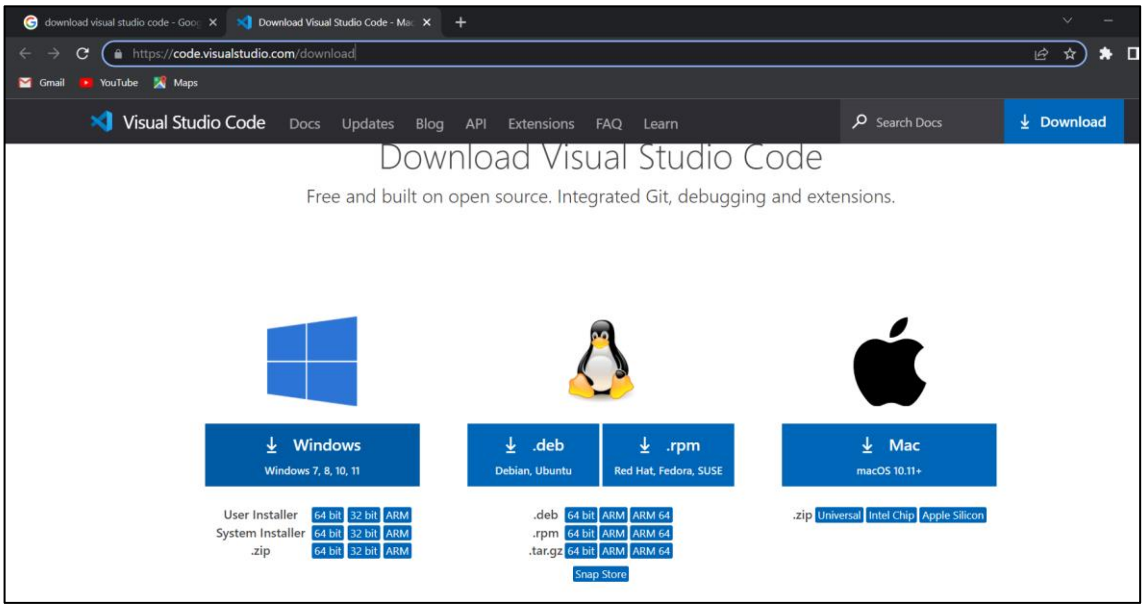
Setting up the Development Environment:

1. Install Visual Studio 2017 / Visual Studio 2019
2. Install Microsoft Azure Service Fabric SDK
3. Install Visual Studio Code
   a. Install Spring Boot Extensions Pack.
   b. Install Java Extensions Pack.
   c. Install Maven for Java.
4. Make sure that the Service Fabric Local cluster is in a running state.
5. Install Docker Desktop.
6. Access the Azure container registry.

Downloading and Installation:

1. Visual Studio Code

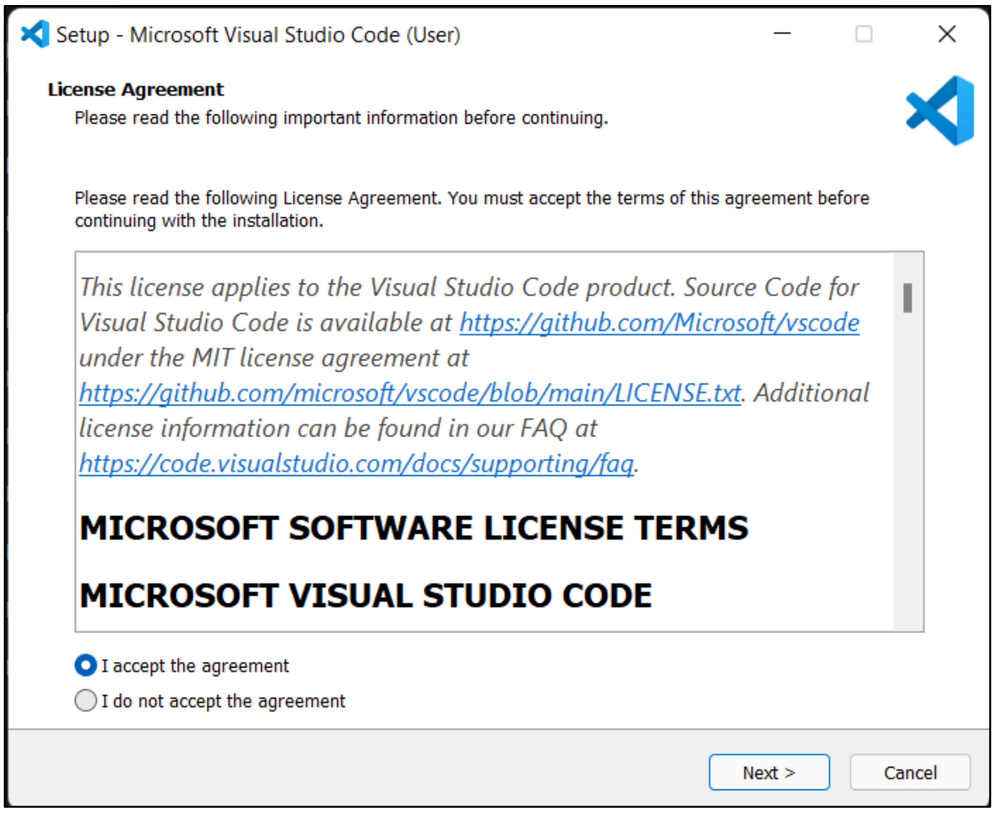
- Go to https://code.visualstudio.com/download and download a suitable installer for your system.



Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

- It will download *VSCodeUserSetup-x64-1.65.2* setup file. Run the setup file. The Microsoft Visual Studio Code Setup window will appear. Accept the agreement and click on Next.


Setup - Microsoft Visual Studio Code (User)

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

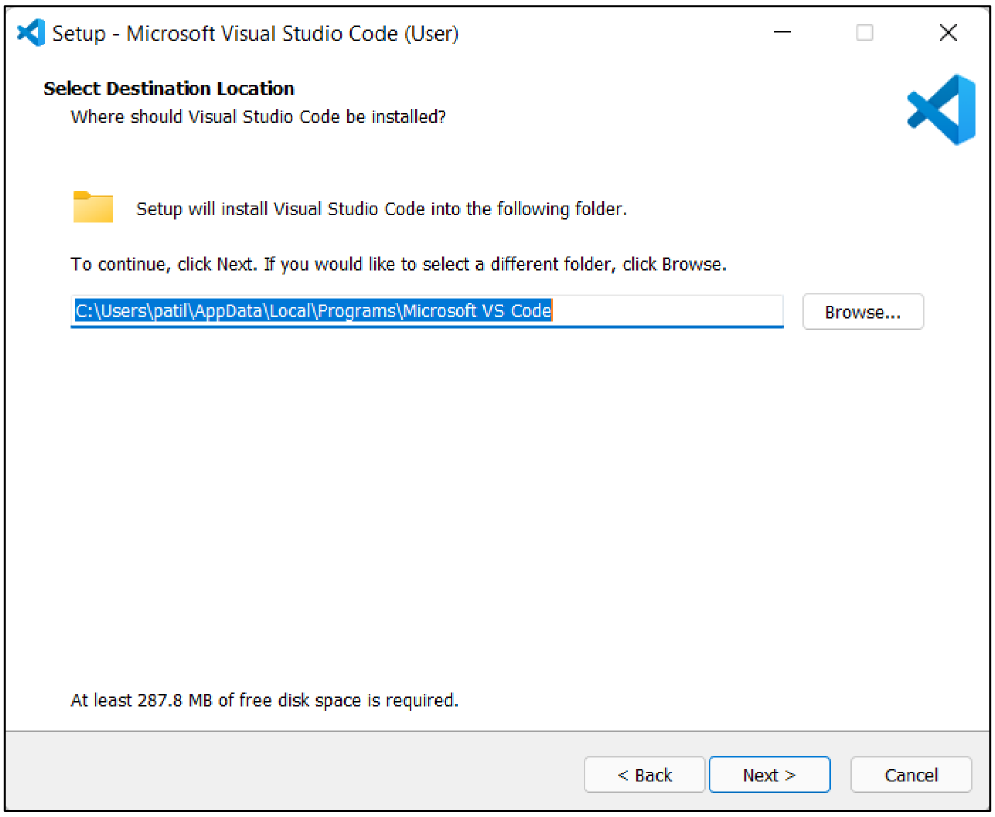
This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

[✓ I accept the agreement]
I do not accept the agreement

- Select the Destination path and click on Next


Setup - Microsoft Visual Studio Code (User)

Select Destination Location

Where should Visual Studio Code be installed?

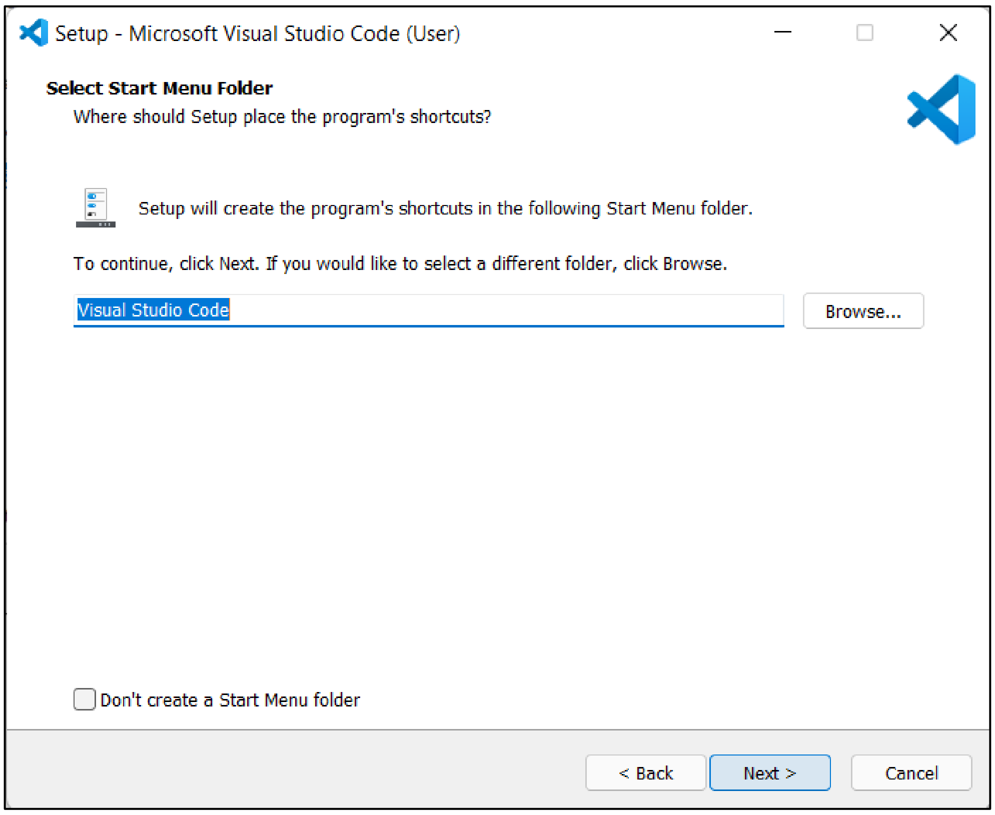
Setup will install Visual Studio Code into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\patil\AppData\Local\Programs\Microsoft VS Code

At least 287.8 MB of free disk space is required.

- Select Start Menu Folder and click on Next.



Setup - Microsoft Visual Studio Code (User)

Select Start Menu Folder

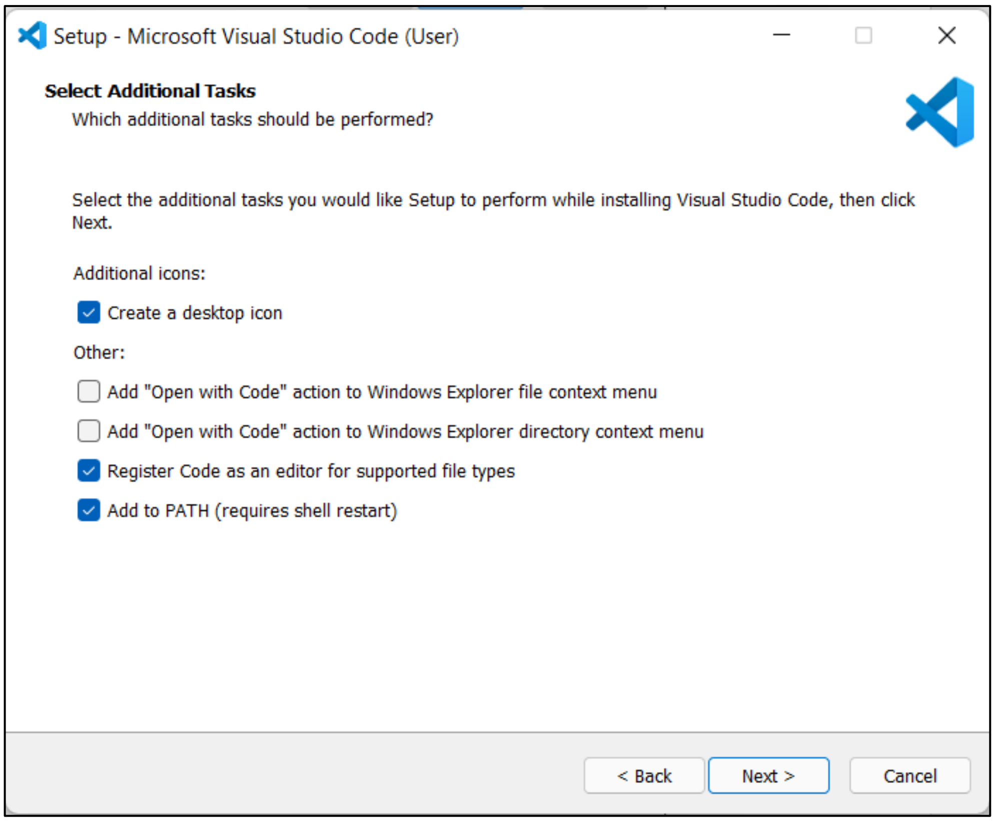
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Visual Studio Code

- Select Additional Tasks as shown below and click Next



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

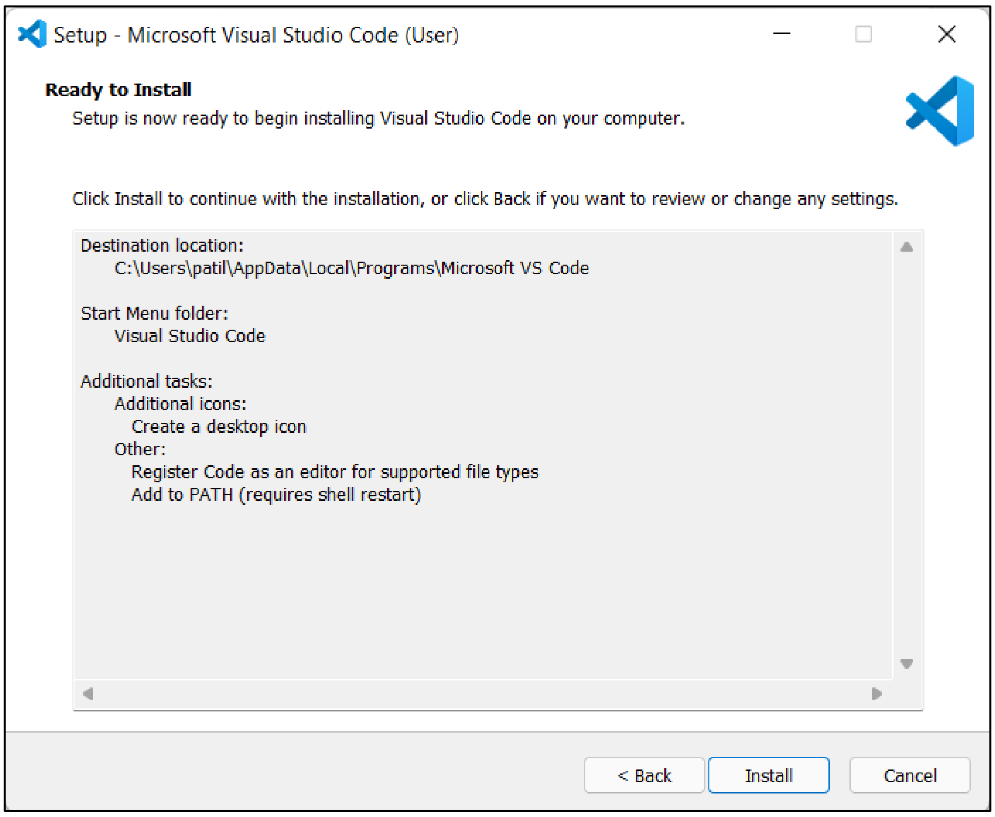
Additional icons:

- [✓] Create a desktop icon

Other:

- [ ] Add "Open with Code" action to Windows Explorer file context menu
- [ ] Add "Open with Code" action to Windows Explorer directory context menu
- [✓] Register Code as an editor for supported file types
- [✓] Add to PATH (requires shell restart)

- Now click on Install

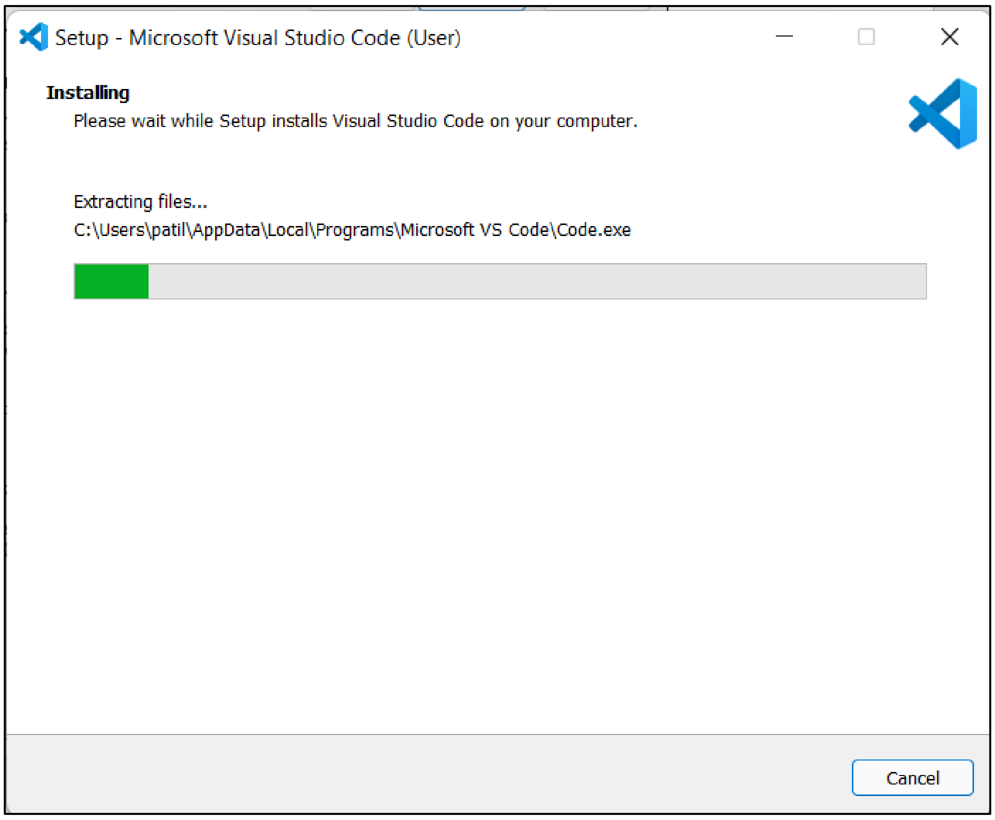


Ready to Install
Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Users\patil\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
    Visual Studio Code

Additional tasks:
    Additional icons:
      Create a desktop icon
    Other:
      Register Code as an editor for supported file types
      Add to PATH (requires shell restart)
```

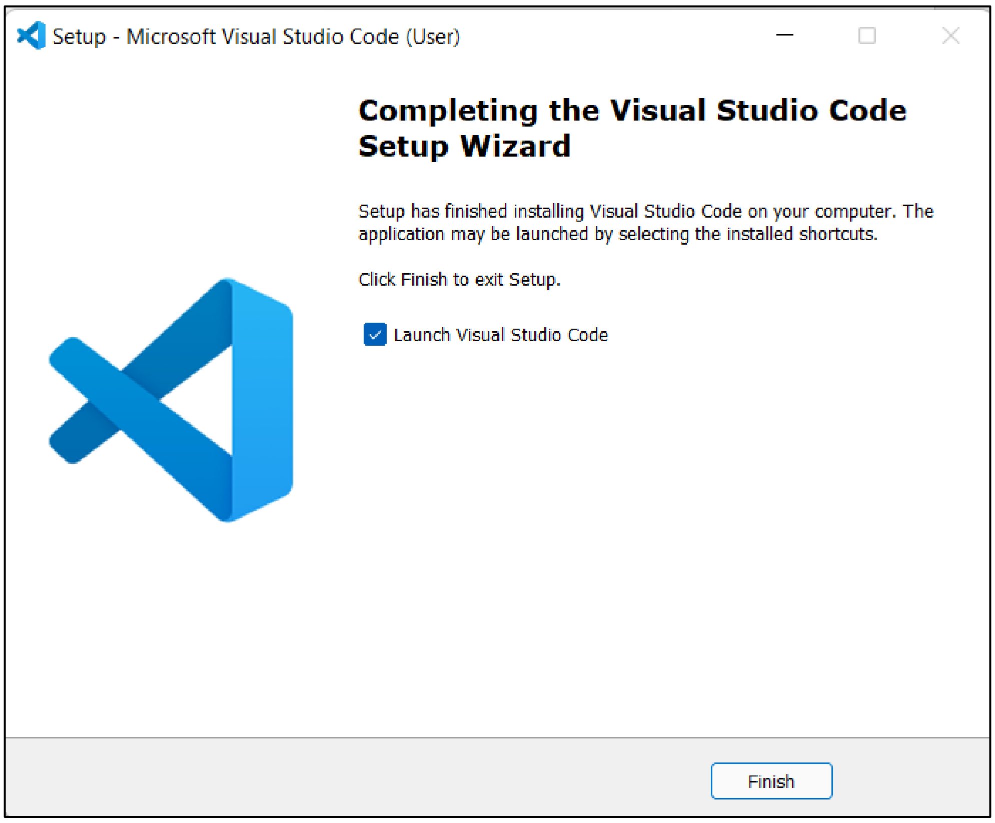
- It will start the installation process.



Installing
Please wait while Setup installs Visual Studio Code on your computer.

Extracting files...
C:\Users\patil\AppData\Local\Programs\Microsoft VS Code\Code.exe

- Finally Click Finish.



Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

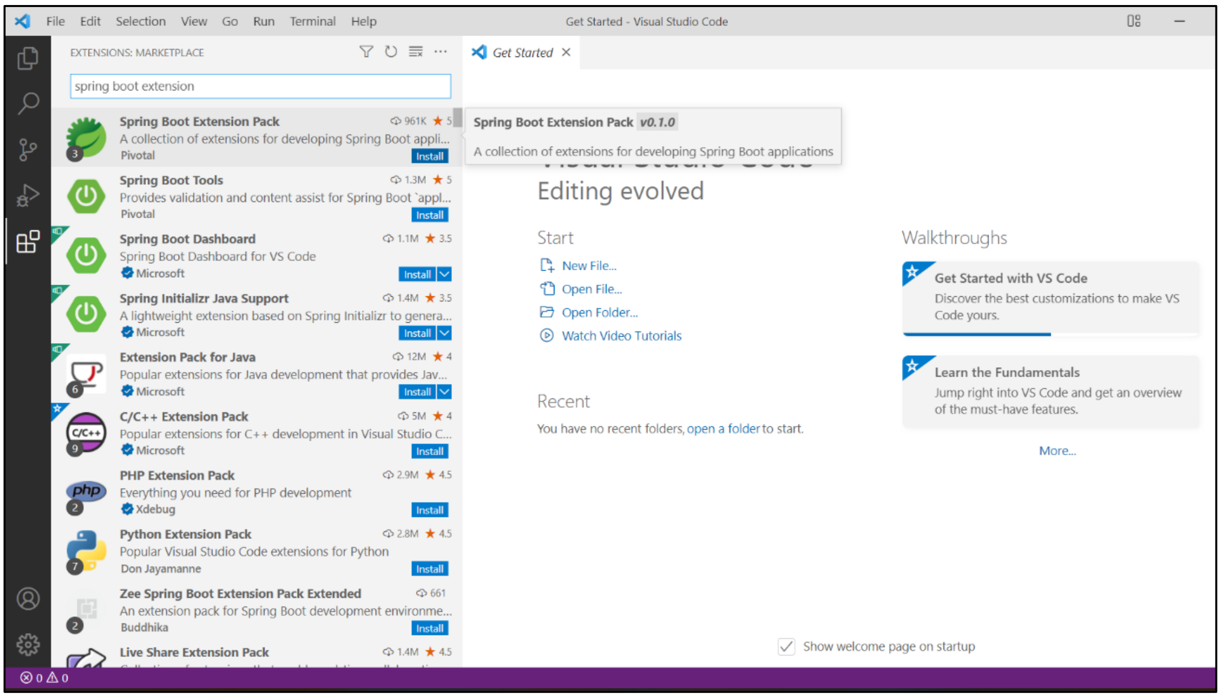
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Visual Studio Code

Finish

- Now to install extensions press ctrl + shift + X and search for Spring Boot Extension Pack and click Install.



File Edit Selection View Go Run Terminal Help — Get Started - Visual Studio Code

EXTENSIONS: MARKETPLACE

spring boot extension

Spring Boot Extension Pack — 961K ★ 5
A collection of extensions for developing Spring Boot appli...
Pivotal — Install

Spring Boot Tools — 1.3M ★ 5
Provides validation and content assist for Spring Boot `appl...
Pivotal — Install

Spring Boot Dashboard — 1.1M ★ 3.5
Spring Boot Dashboard for VS Code
Microsoft — Install

Spring Initializr Java Support — 1.4M ★ 3.5
A lightweight extension based on Spring Initializr to genera...
Microsoft — Install

Extension Pack for Java — 12M ★ 4
Popular extensions for Java development that provides Jav...
Microsoft — Install

C/C++ Extension Pack — 5M ★ 4
Popular extensions for C++ development in Visual Studio C...
Microsoft — Install

PHP Extension Pack — 2.9M ★ 4.5
Everything you need for PHP development
Xdebug — Install

Python Extension Pack — 2.8M ★ 4.5
Popular Visual Studio Code extensions for Python
Don Jayamanne — Install

Zee Spring Boot Extension Pack Extended — 661
An extension pack for Spring Boot development environme...
Buddhika — Install

Live Share Extension Pack — 1.4M ★ 4.5

Get Started

Spring Boot Extension Pack *v0.1.0*
A collection of extensions for developing Spring Boot applications

Editing evolved

Start
New File...
Open File...
Open Folder...
Watch Video Tutorials

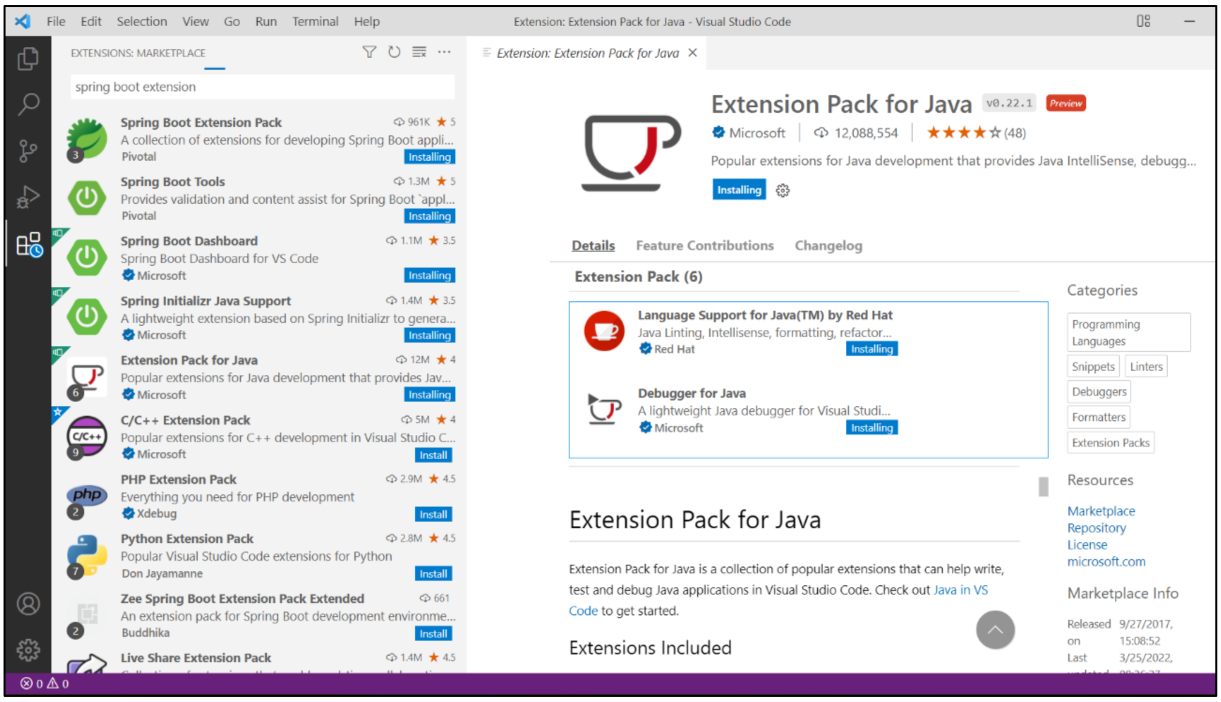
Recent
You have no recent folders, open a folder to start.

Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
More...

✓ Show welcome page on startup

⊗ 0 ⚠ 0

- Similarly install Extension Pack for Java



File Edit Selection View Go Run Terminal Help
Extension: Extension Pack for Java - Visual Studio Code
EXTENSIONS: MARKETPLACE
Extension: Extension Pack for Java

spring boot extension

Spring Boot Extension Pack 961K ★5
A collection of extensions for developing Spring Boot appli...
Pivotal Installing

Spring Boot Tools 1.3M ★5
Provides validation and content assist for Spring Boot `appl...
Pivotal Installing

Spring Boot Dashboard 1.1M ★3.5
Spring Boot Dashboard for VS Code
Microsoft Installing

Spring Initializr Java Support 1.4M ★3.5
A lightweight extension based on Spring Initializr to genera...
Microsoft Installing

Extension Pack for Java 12M ★4
Popular extensions for Java development that provides Jav...
Microsoft Installing

C/C++ Extension Pack 5M ★4
Popular extensions for C++ development in Visual Studio C...
Microsoft Install

PHP Extension Pack 2.9M ★4.5
Everything you need for PHP development
Xdebug Install

Python Extension Pack 2.8M ★4.5
Popular Visual Studio Code extensions for Python
Don Jayamanne Install

Zee Spring Boot Extension Pack Extended 661
An extension pack for Spring Boot development environme...
Buddhika Install

Live Share Extension Pack 1.4M ★4.5

Extension Pack for Java v0.22.1 Preview
Microsoft | 12,088,554 | ★★★★☆ (48)
Popular extensions for Java development that provides Java IntelliSense, debugg...
Installing

Details Feature Contributions Changelog

Extension Pack (6)

Language Support for Java(TM) by Red Hat
Java Linting, Intellisense, formatting, refactor...
Red Hat Installing

Debugger for Java
A lightweight Java debugger for Visual Studi...
Microsoft Installing

Categories: Programming Languages, Snippets, Linters, Debuggers, Formatters, Extension Packs

Resources: Marketplace, Repository, License, microsoft.com

Marketplace Info
Released on 9/27/2017, 15:08:52
Last 3/25/2022,

Extension Pack for Java

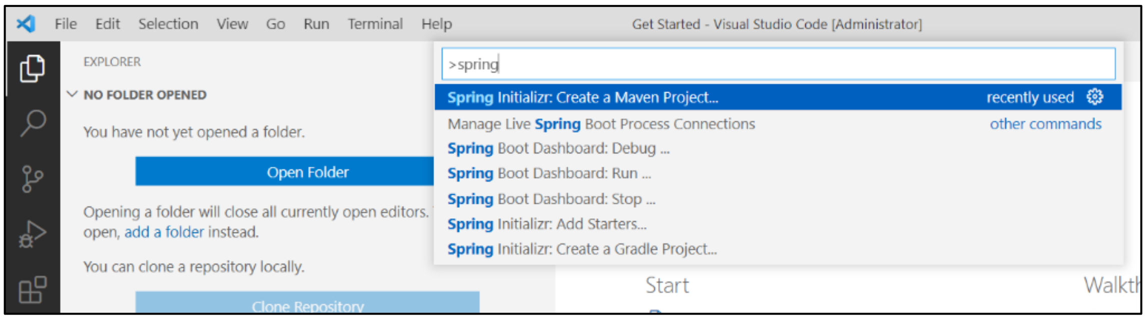
Extension Pack for Java is a collection of popular extensions that can help write, test and debug Java applications in Visual Studio Code. Check out Java in VS Code to get started.

Extensions Included

Develop a Spring Boot API

Now it’s time to get started on the application.

- Launch Visual Studio Code as an administrator.
- Press Ctrl+Shift+P to open the command palette.
- Enter spring in the command palette, and choose *Spring Initializer: Generate a maven*



File Edit Selection View Go Run Terminal Help
Get Started - Visual Studio Code [Administrator]
EXPLORER
NO FOLDER OPENED
Open Folder
Clone Repository
Start
Walkt

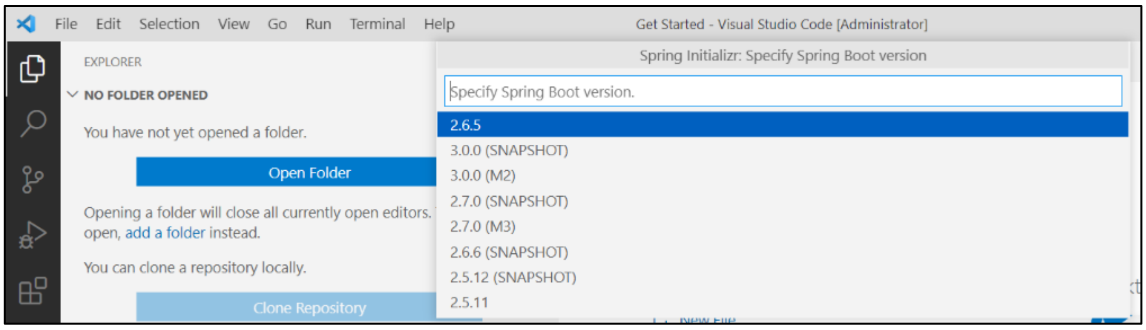
You have not yet opened a folder.

Opening a folder will close all currently open editors. open, add a folder instead.

You can clone a repository locally.

```
>spring
Spring Initializr: Create a Maven Project...          recently used
Manage Live Spring Boot Process Connections           other commands
Spring Boot Dashboard: Debug ...
Spring Boot Dashboard: Run ...
Spring Boot Dashboard: Stop ...
Spring Initializr: Add Starters...
Spring Initializr: Create a Gradle Project...
```

- Choose the latest Spring boot version.



File Edit Selection View Go Run Terminal Help
Get Started - Visual Studio Code [Administrator]
EXPLORER
NO FOLDER OPENED
Open Folder
Clone Repository

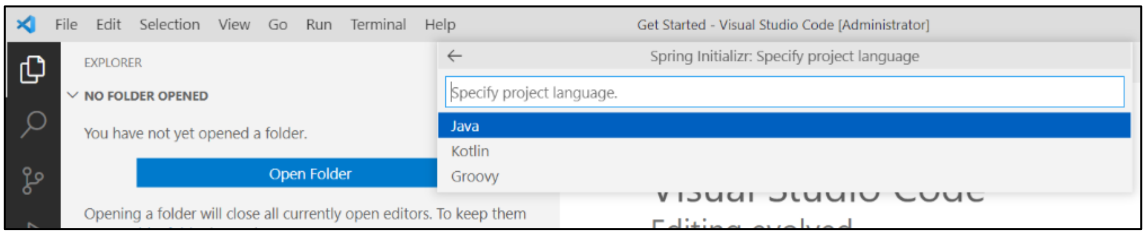
You have not yet opened a folder.

Opening a folder will close all currently open editors. open, add a folder instead.

You can clone a repository locally.

```
Spring Initializr: Specify Spring Boot version
Specify Spring Boot version.
2.6.5
3.0.0 (SNAPSHOT)
3.0.0 (M2)
2.7.0 (SNAPSHOT)
2.7.0 (M3)
2.6.6 (SNAPSHOT)
2.5.12 (SNAPSHOT)
2.5.11
```

- Choose *Java* for Specify Project Language.



You have not yet opened a folder.

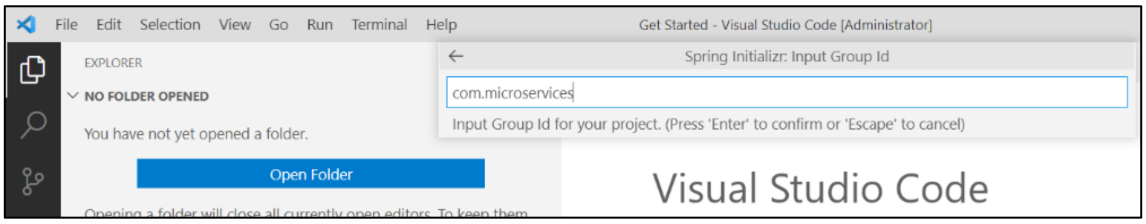
Opening a folder will close all currently open editors. To keep them

Spring Initializr: Specify project language

Specify project language.

Java
Kotlin
Groovy

- Enter *com.microservices* in the input group ID for your project



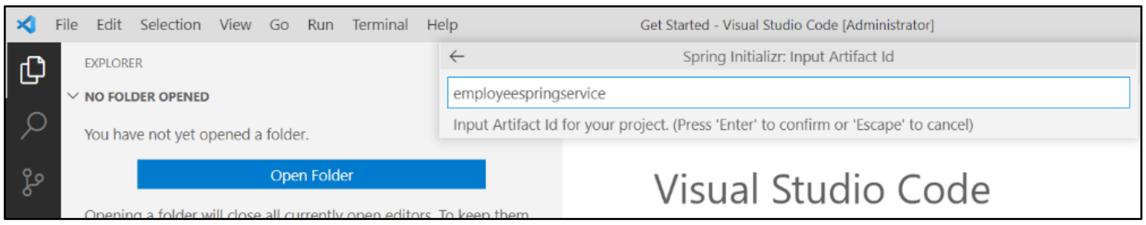
You have not yet opened a folder.

Spring Initializr: Input Group Id

com.microservices

Input Group Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

- Enter *employeespringservice* in the input artifact ID for your project.



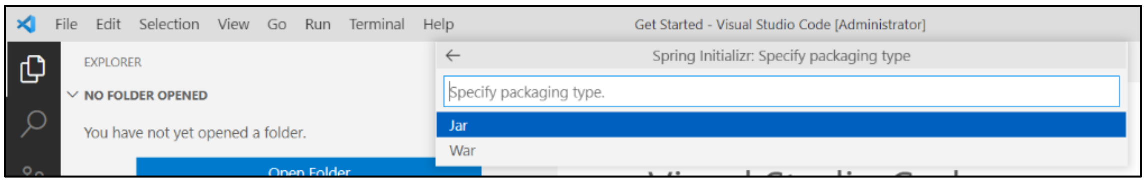
You have not yet opened a folder.

Spring Initializr: Input Artifact Id

employeespringservice

Input Artifact Id for your project. (Press 'Enter' to confirm or 'Escape' to cancel)

- Specify the packaging type as *jar*



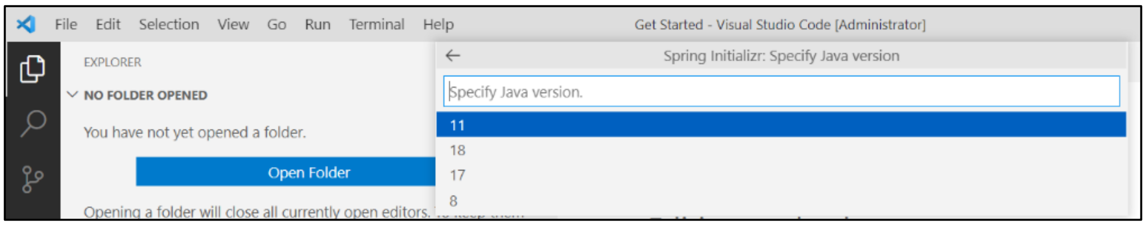
You have not yet opened a folder.

Spring Initializr: Specify packaging type

Specify packaging type.

Jar
War

- Specify Java version



You have not yet opened a folder.

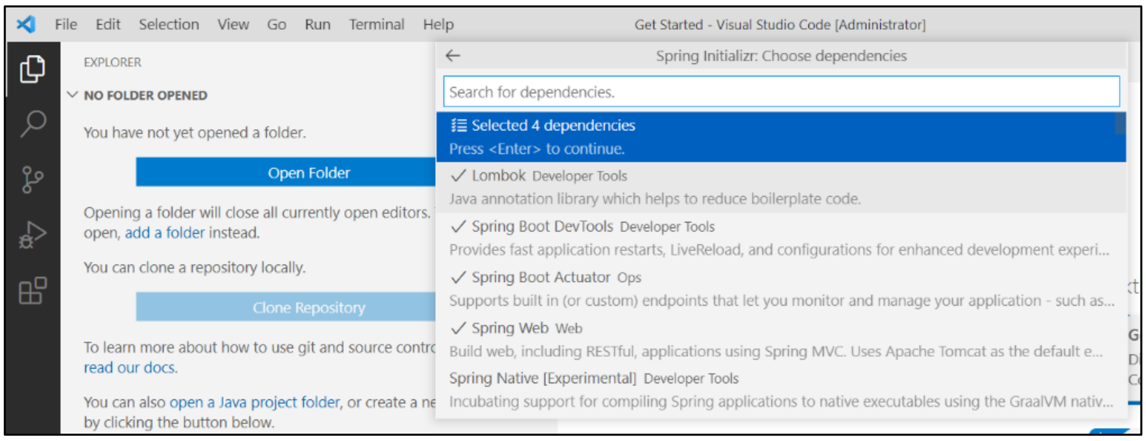
Opening a folder will close all currently open editors.

Spring Initializr: Specify Java version

Specify Java version.

11
18
17
8

- Choose the following dependencies.
    - a. DevTools
    - b. Lombok
    - c. Web
    - d. Actuator



Spring Initializr: Choose dependencies

Search for dependencies.

Selected 4 dependencies
Press <Enter> to continue.

✓ Lombok Developer Tools
Java annotation library which helps to reduce boilerplate code.

✓ Spring Boot DevTools Developer Tools
Provides fast application restarts, LiveReload, and configurations for enhanced development experi...

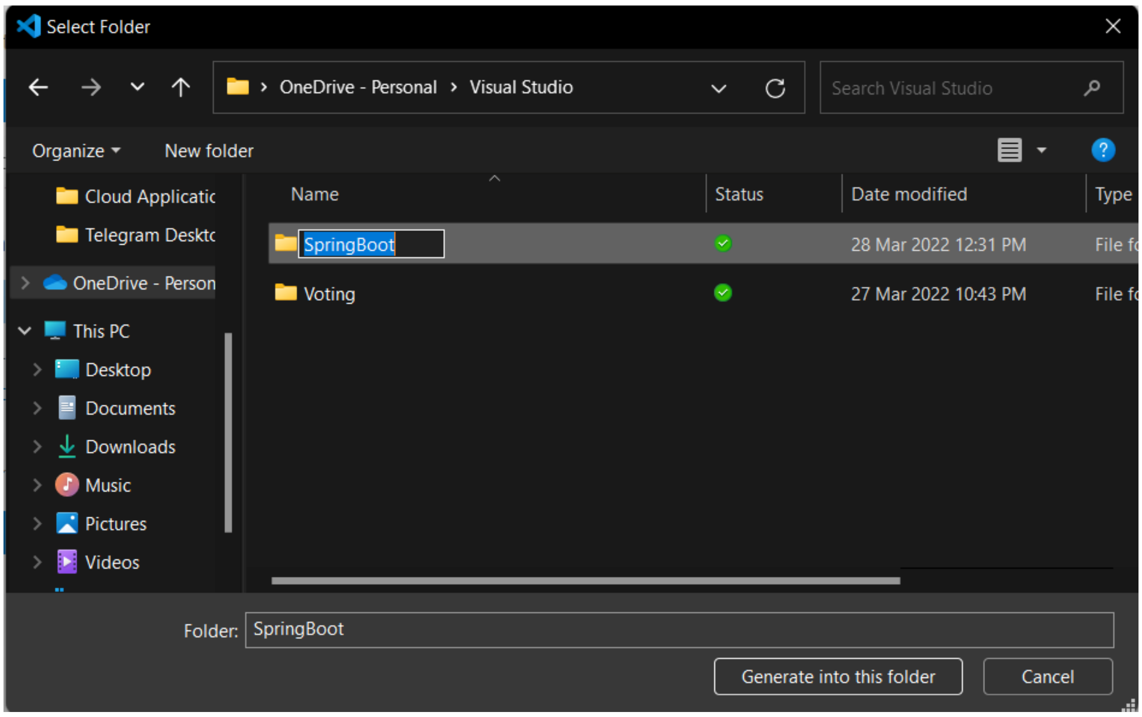
✓ Spring Boot Actuator Ops
Supports built in (or custom) endpoints that let you monitor and manage your application - such as...

✓ Spring Web Web
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default e...

Spring Native [Experimental] Developer Tools
Incubating support for compiling Spring applications to native executables using the GraalVM nativ...

File Edit Selection View Go Run Terminal Help
Get Started - Visual Studio Code [Administrator]
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source contr read our docs.
You can also open a Java project folder, or create a ne by clicking the button below.

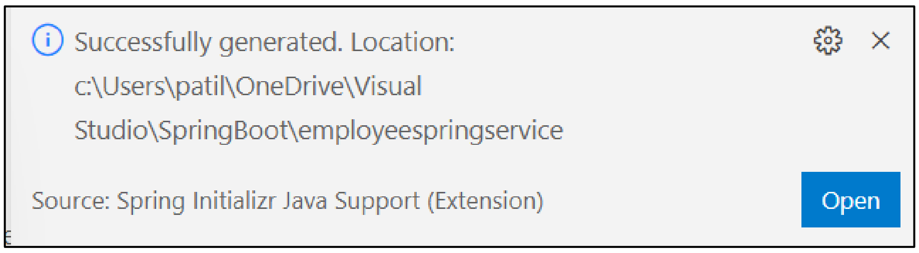
- Choose the path where you want to save the solution. And click on Generate into this folder.



Folder: SpringBoot

Name	Status	Date modified	Type
SpringBoot	✓	28 Mar 2022 12:31 PM	File fo
Voting	✓	27 Mar 2022 10:43 PM	File fo

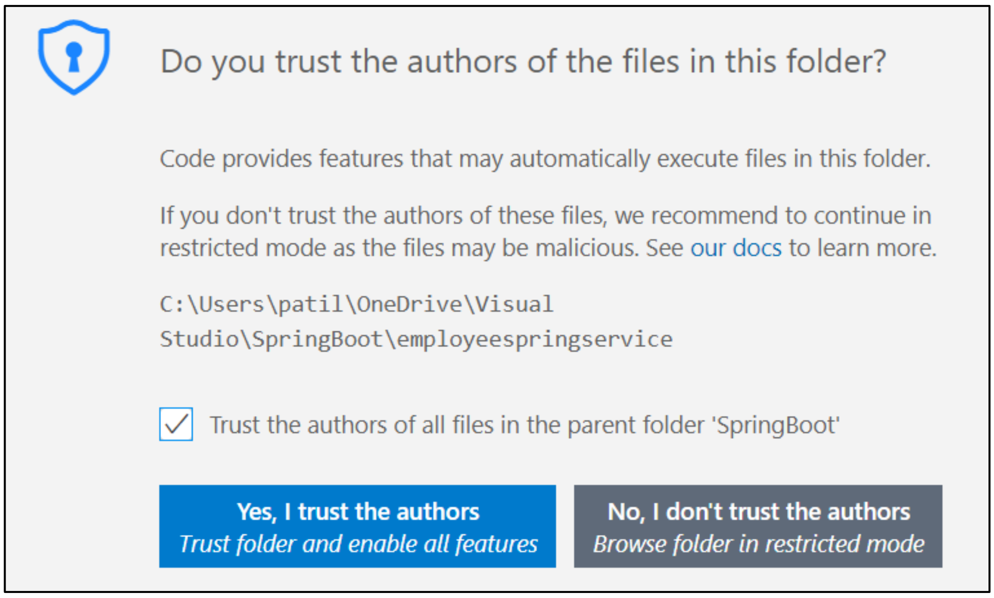
- You will get a successful generation message. Now click Open



Successfully generated. Location: c:\Users\patil\OneDrive\Visual Studio\SpringBoot\employeespringservice

Source: Spring Initializr Java Support (Extension)

- Select the checkbox Trust the authors and click on Yes as shown below.



Do you trust the authors of the files in this folder?

Code provides features that may automatically execute files in this folder.

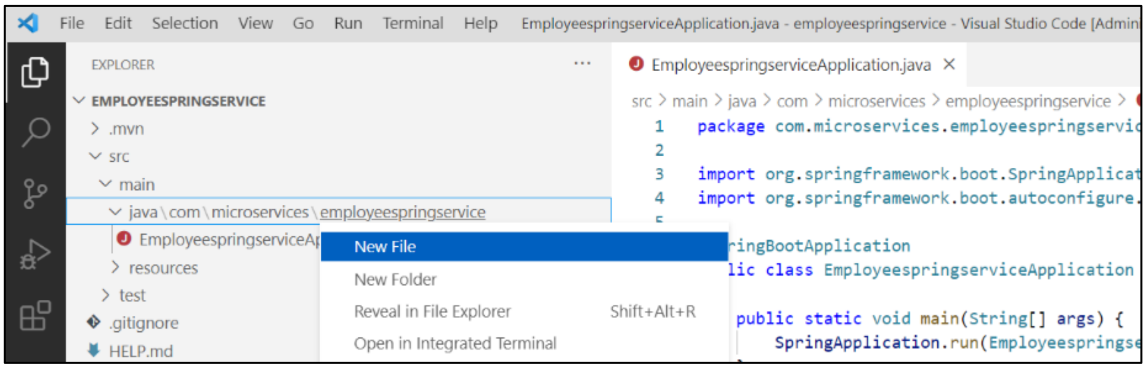
If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.

```
C:\Users\patil\OneDrive\Visual
Studio\SpringBoot\employeespringservice
```

✓ Trust the authors of all files in the parent folder 'SpringBoot'

Yes, I trust the authors
Trust folder and enable all features
No, I don't trust the authors
Browse folder in restricted mode

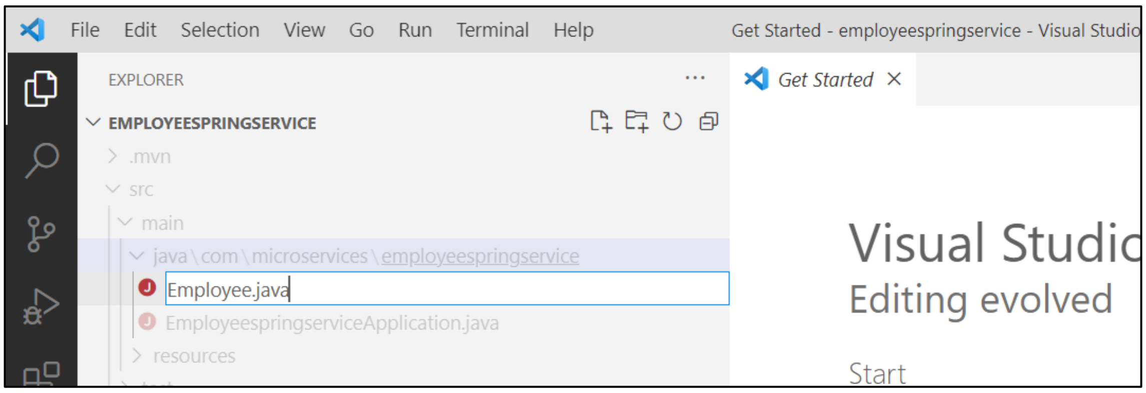
- Right-click the EMPLOYEESPRINGSERVICE folder under src ➤ main ➤ java ➤ com ➤ microservices, as shown below, and click Add File.



File
Edit
Selection
View
Go
Run
Terminal
Help
EmployeespringserviceApplication.java - employeespringservice - Visual Studio Code [Admin
EXPLORER
EMPLOYEESPRINGSERVICE
.mvn
src
main
java \ com \ microservices \ employeespringservice
EmployeespringserviceAp
resources
test
.gitignore
HELP.md
New File
New Folder
Reveal in File Explorer Shift+Alt+R
Open in Integrated Terminal
EmployeespringserviceApplication.java
src > main > java > com > microservices > employeespringservice >

```
package com.microservices.employeespringservic
import org.springframework.boot.SpringApplicat
import org.springframework.boot.autoconfigure.
ringBootApplication
lic class EmployeespringserviceApplication
public static void main(String[] args) {
    SpringApplication.run(Employeespringse
```

- Name the file *Employee.Java* and add the following code.


File Edit Selection View Go Run Terminal Help
Get Started - employeespringservice - Visual Studio
EXPLORER
Get Started
EMPLOYEESPRINGSERVICE
.mvn
src
main
java \ com \ microservices \ employeespringservice
Employee.java
EmployeespringserviceApplication.java
resources
Visual Studio
Editing evolved
Start

```
package com.microservices.employeespringservice;

import lombok.AllArgsConstructor;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.Setter;

/*** Employee ***/
@Getter
@Setter
@EqualsAndHashCode
@AllArgsConstructor
public class Employee {
    public String firstName;
    public String lastName;
    public String ipAddress;

    public Employee() {}

    public Employee(String firstName, String lastName, String ipAddress) {
        super();
        this.firstName = firstName;
        this.lastName = lastName;
        this.ipAddress = ipAddress;
    }
}
```

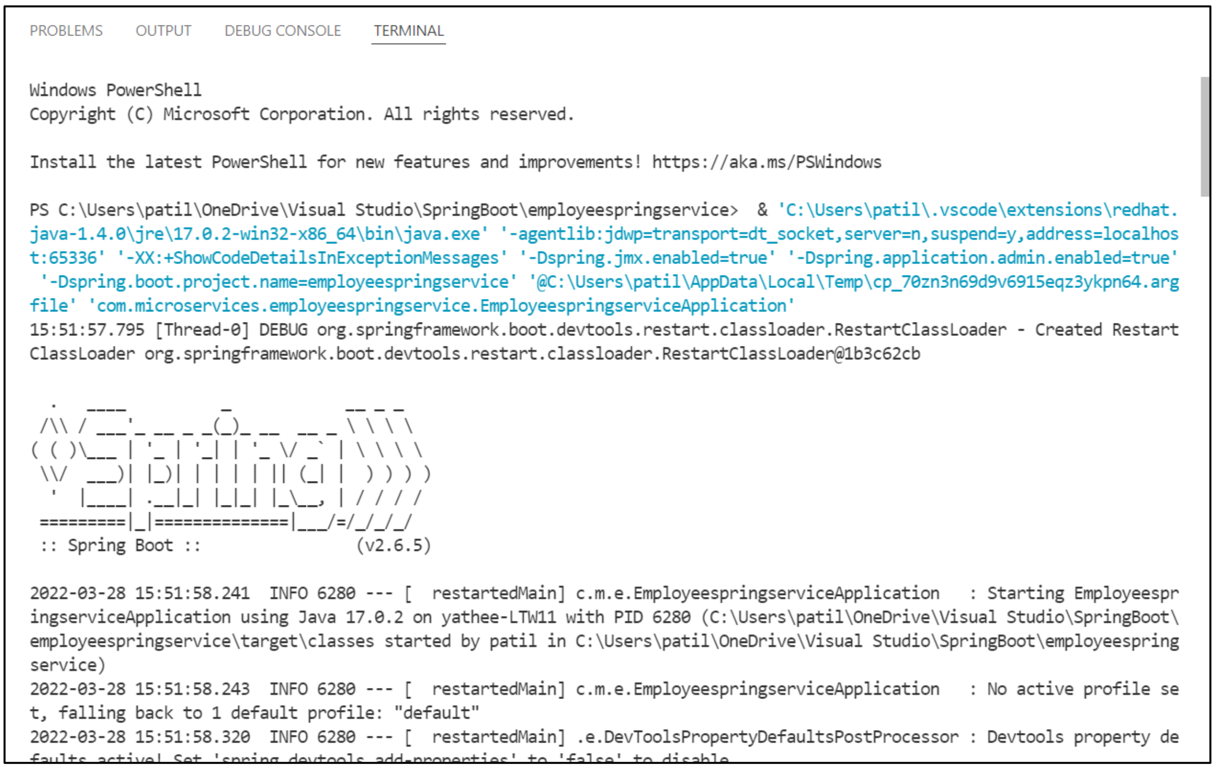
- Now let's create an employee service that returns an employee's information. Right-click the employeespringservice folder and add a file named *EmployeeService.java*. Add the following code to it.

```
package com.microservices.employeespringservice;

import java.net.InetAddress;
import java.net.UnknownHostException;
import org.springframework.stereotype.Service;

/** EmployeeService **/
@Service
public class EmployeeService {
    public Employee GetEmployee(String firstName, String  lastName){
        String ipAddress;
        try {
            ipAddress = InetAddress.getLocalHost().getHostAddress().toString();
        }
        catch (UnknownHostException e) {
            ipAddress = e.getMessage();
        }
        Employee employee = new Employee(firstName, lastName, ipAddress);
        return employee;
    }
}
```

- Now let's create an employee controller that invokes the employee service to return the details of an employee. Right-click the employeespringservice folder and add a file named *EmployeeController. java*. Add the following code to it.

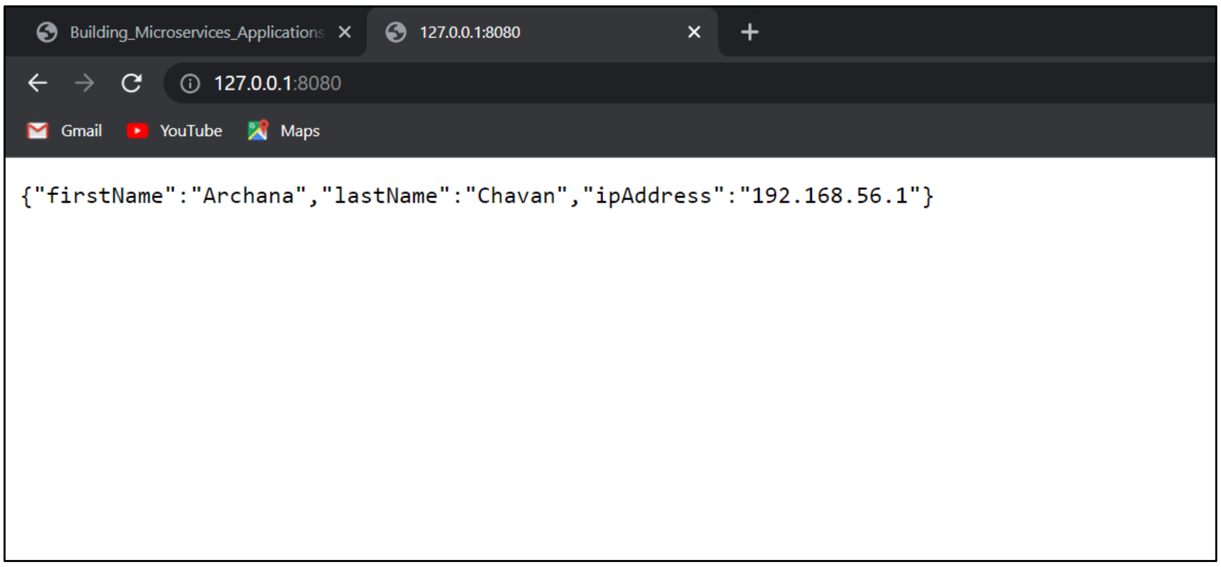
```
package com.microservices.employeespringservice;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ResponseBody;

/**  * EmployeeController  */
@Controller
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping("/")
    @ResponseBody
    public Employee getEmployee(){
        return employeeService.GetEmployee("Archana","Chavan");
    }
}
```

- Now you are ready for a simple REST-based service that returns employee information. Now press F5 to debug and run the application.



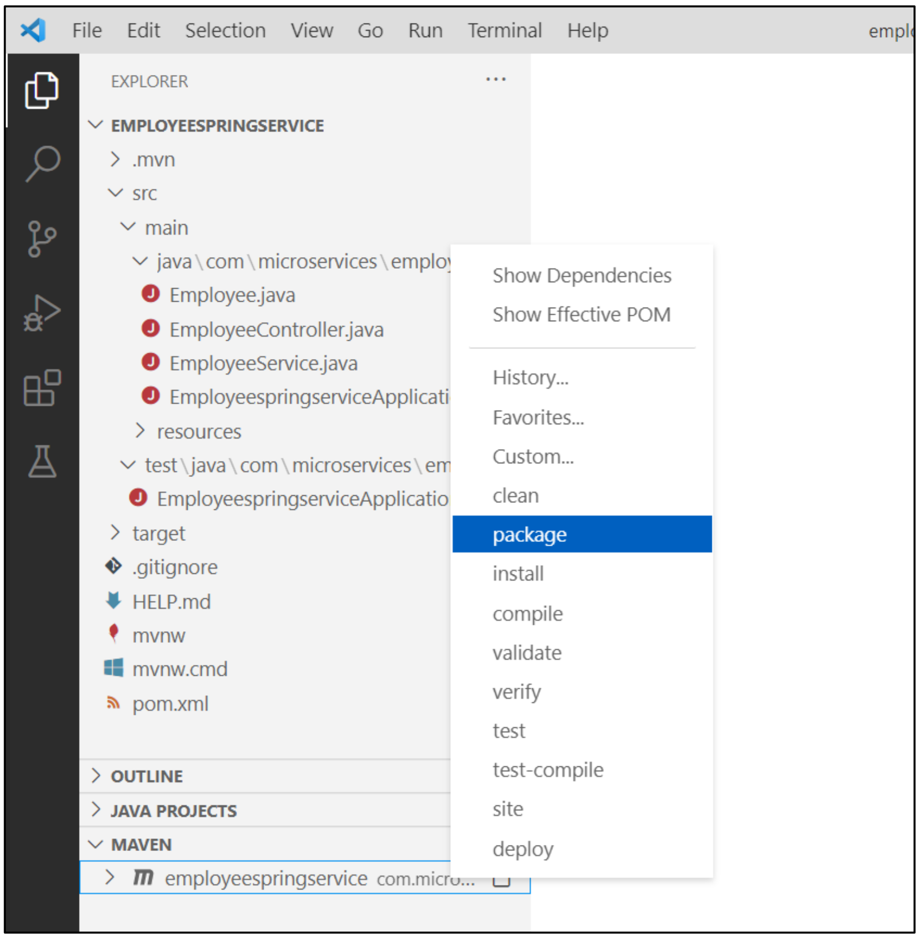
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\patil\OneDrive\Visual Studio\SpringBoot\employeespringservice>  & 'C:\Users\patil\.vscode\extensions\redhat.
java-1.4.0\jre\17.0.2-win32-x86_64\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhos
t:65336' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dspring.jmx.enabled=true' '-Dspring.application.admin.enabled=true'
 '-Dspring.boot.project.name=employeespringservice' '@C:\Users\patil\AppData\Local\Temp\cp_70zn3n69d9v6915eqz3ykpn64.arg
file' 'com.microservices.employeespringservice.EmployeespringserviceApplication'
15:51:57.795 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created Restart
ClassLoader org.springframework.boot.devtools.restart.classloader.RestartClassLoader@1b3c62cb

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.5)

2022-03-28 15:51:58.241  INFO 6280 --- [  restartedMain] c.m.e.EmployeespringserviceApplication   : Starting Employeespr
ingserviceApplication using Java 17.0.2 on yathee-LTW11 with PID 6280 (C:\Users\patil\OneDrive\Visual Studio\SpringBoot\
employeespringservice\target\classes started by patil in C:\Users\patil\OneDrive\Visual Studio\SpringBoot\employeespring
service)
2022-03-28 15:51:58.243  INFO 6280 --- [  restartedMain] c.m.e.EmployeespringserviceApplication   : No active profile se
t, falling back to 1 default profile: "default"
2022-03-28 15:51:58.320  INFO 6280 --- [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property de
faults active! Set 'spring.devtools.add-properties' to 'false' to disable
```

- Now go to url http://127.0.0.1:8080/ to see the output.

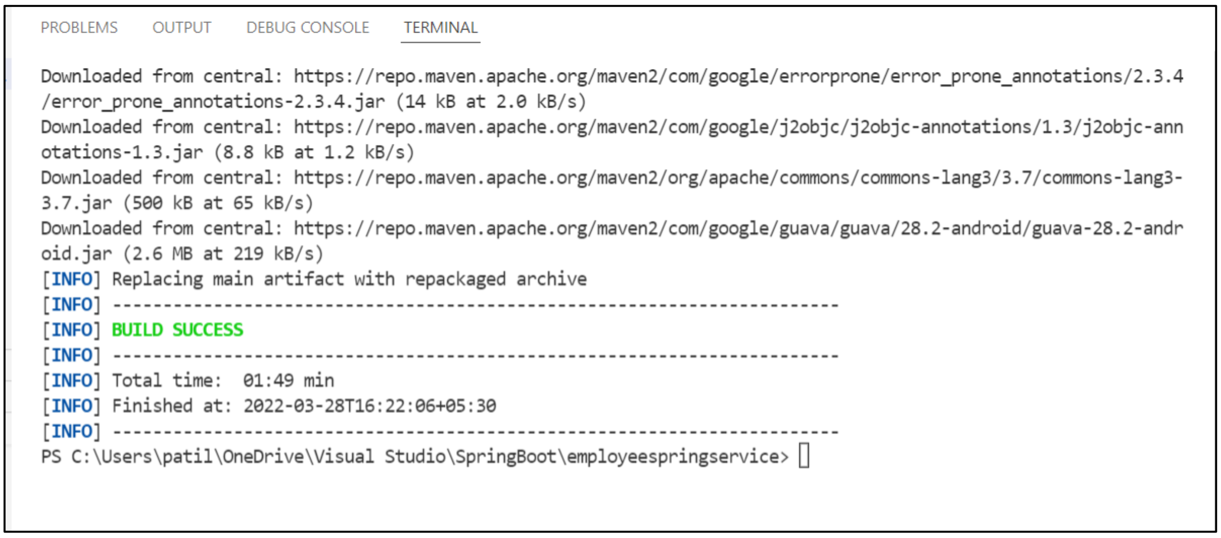


```
{"firstName":"Archana","lastName":"Chavan","ipAddress":"192.168.56.1"}
```

- Right-click *employeespringservice* under Maven Projects. Click package, as shown in Figure below. This generates the JAR file.



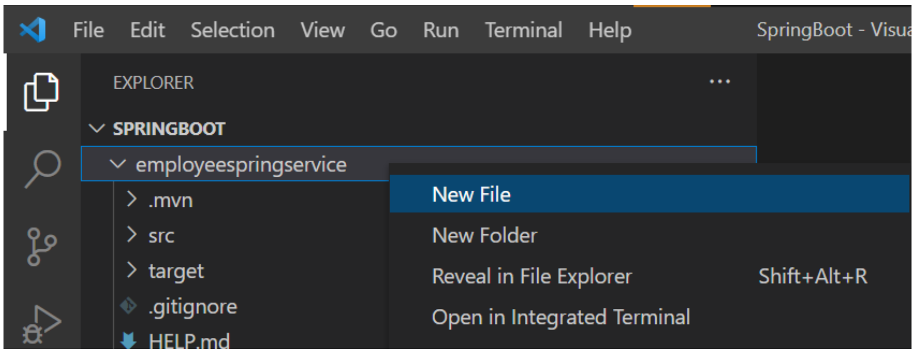
- It will start the build process of the jar file and you will get a BUILD SUCCESS message shown in fig below.



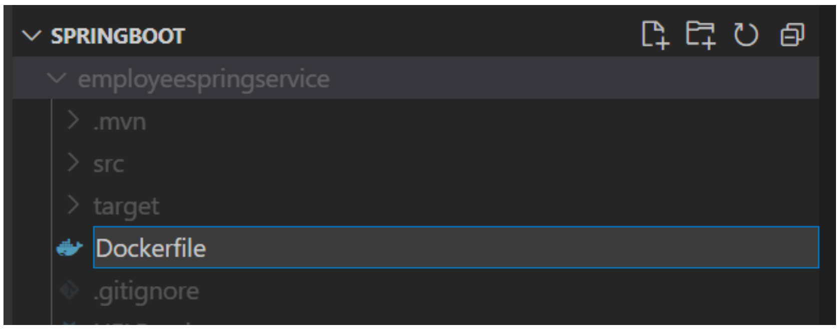
```
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/errorprone/error_prone_annotations/2.3.4
/error_prone_annotations-2.3.4.jar (14 kB at 2.0 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/j2objc/j2objc-annotations/1.3/j2objc-ann
otations-1.3.jar (8.8 kB at 1.2 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.7/commons-lang3-
3.7.jar (500 kB at 65 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/28.2-android/guava-28.2-andr
oid.jar (2.6 MB at 219 kB/s)
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:49 min
[INFO] Finished at: 2022-03-28T16:22:06+05:30
[INFO] ------------------------------------------------------------------------
PS C:\Users\patil\OneDrive\Visual Studio\SpringBoot\employeespringservice> 
```

Now you have a simple Spring Boot-based REST API, and you have generated a JAR file. We will now deploy it to Service Fabric as a guest executable. To deploy, we will use Visual Studio 2019.

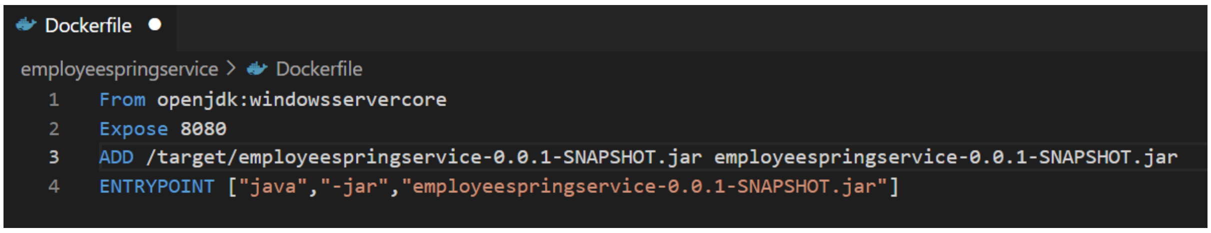
- *Create Dockerfile:*
    - employeespringservice > Right click >New File



    - Name it as *Dockerfile*



    - Add the following code to this file.



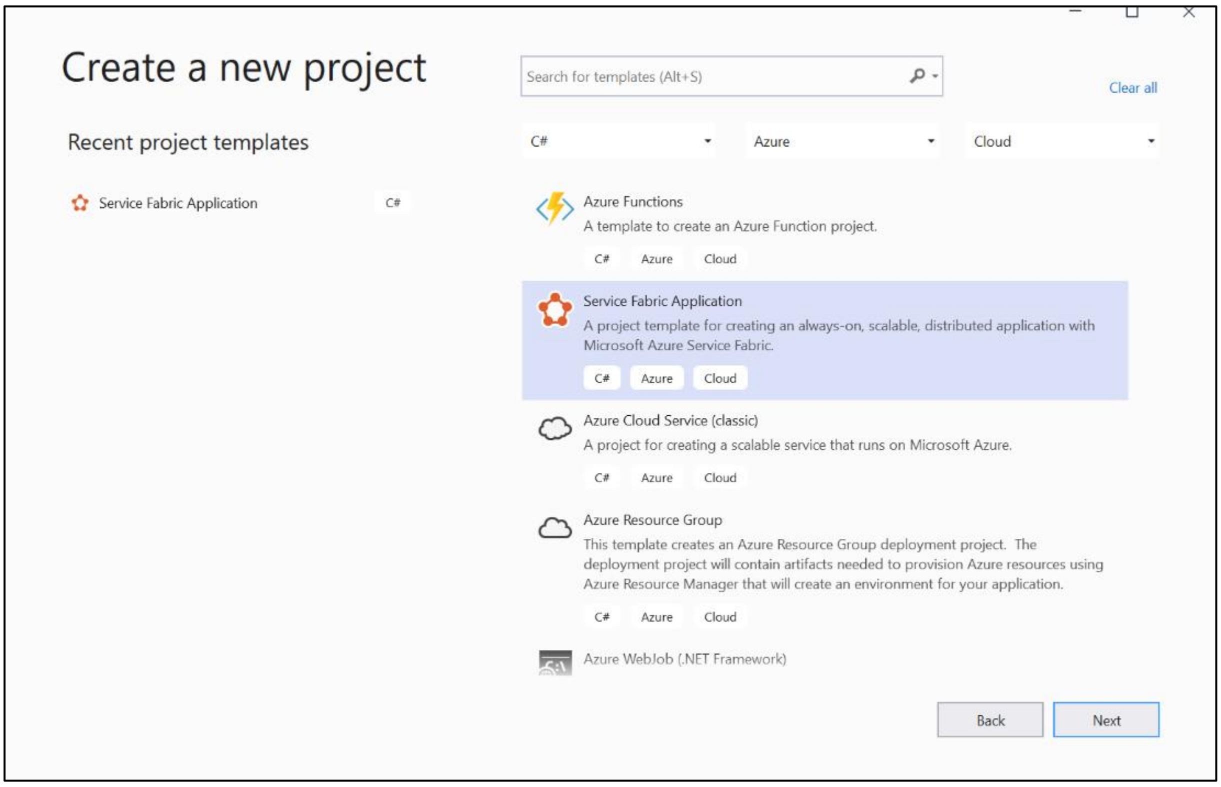
```
From openjdk:windowsservercore
Expose 8080
ADD /target/employeespringservice-0.0.1-SNAPSHOT.jar employeespringservice-0.0.1-SNAPSHOT.jar
ENTRYPOINT ["java","-jar","employeespringservice-0.0.1-SNAPSHOT.jar"]
```

Deploy a Spring Boot Service as a Guest Executable

After executing all the steps in the previous section, your development is complete. Please follow the steps in this section to deploy the developed Spring Boot application as a guest executable. This shows that it is possible to host a non-Microsoft stack application on a Service Fabric cluster by using a guest executable programming model. Service Fabric considers guest executables a stateless service.

- Launch Visual Studio 2019 as an administrator.

- Create a New Project
- Select C# ➤ Azure ➤ Cloud ➤ Service Fabric Application ➤ Next



Create a new project

Search for templates (Alt+S)

Clear all

Recent project templates

Service Fabric Application C#

C# | Azure | Cloud

Azure Functions
A template to create an Azure Function project.
C# Azure Cloud

Service Fabric Application
A project template for creating an always-on, scalable, distributed application with Microsoft Azure Service Fabric.
C# Azure Cloud

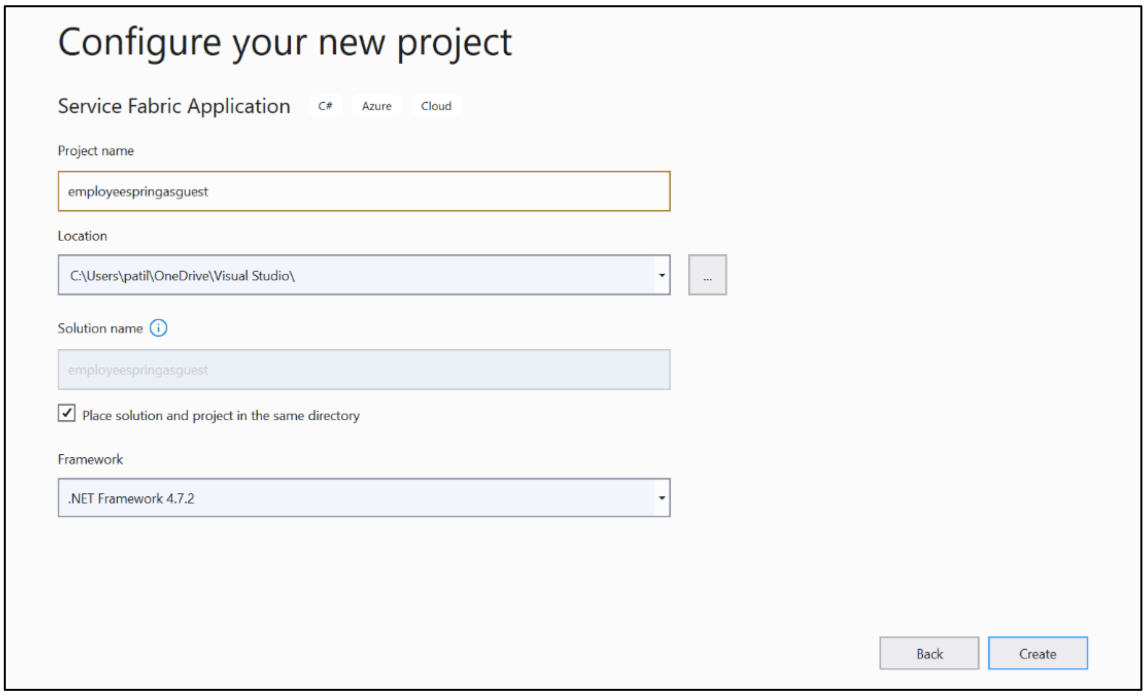
Azure Cloud Service (classic)
A project for creating a scalable service that runs on Microsoft Azure.
C# Azure Cloud

Azure Resource Group
This template creates an Azure Resource Group deployment project. The deployment project will contain artifacts needed to provision Azure resources using Azure Resource Manager that will create an environment for your application.
C# Azure Cloud

Azure WebJob (.NET Framework)

Back Next

- Name the application *employeespringasguest*, as shown in Figure ➤ Create



Configure your new project

Service Fabric Application C# Azure Cloud

Project name: employeespringasguest

Location: C:\Users\patil\OneDrive\Visual Studio\

Solution name: employeespringasguest

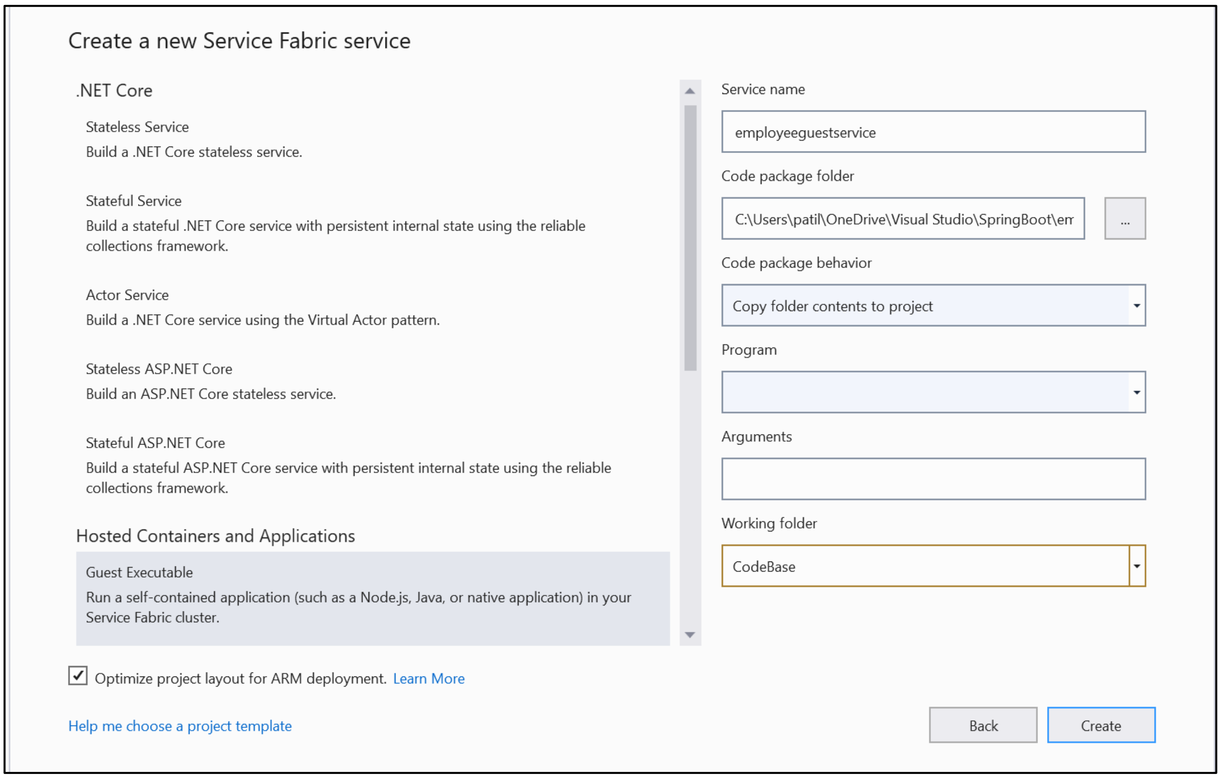
☑ Place solution and project in the same directory

Framework: .NET Framework 4.7.2

Back Create

- Select Guest Executable from Hosted Containers and

- In New Service Fabric Service, select the following (as shown in Figure below)
  - a. Service Name: *employeeguestservice*
  - b. Code Package Folder: Point to the target folder in which Visual Studio Code generated the JAR file for the Spring Boot service.
  - c. Code Package Behavior: Copy folder contents to a folder
  - d. Working Folder: CodeBase ➤ Next



Create a new Service Fabric service

.NET Core

Stateless Service
Build a .NET Core stateless service.

Stateful Service
Build a stateful .NET Core service with persistent internal state using the reliable collections framework.

Actor Service
Build a .NET Core service using the Virtual Actor pattern.

Stateless ASP.NET Core
Build an ASP.NET Core stateless service.

Stateful ASP.NET Core
Build a stateful ASP.NET Core service with persistent internal state using the reliable collections framework.

Hosted Containers and Applications

Guest Executable
Run a self-contained application (such as a Node.js, Java, or native application) in your Service Fabric cluster.

Service name: employeeguestservice

Code package folder: C:\Users\patil\OneDrive\Visual Studio\SpringBoot\em

Code package behavior: Copy folder contents to project

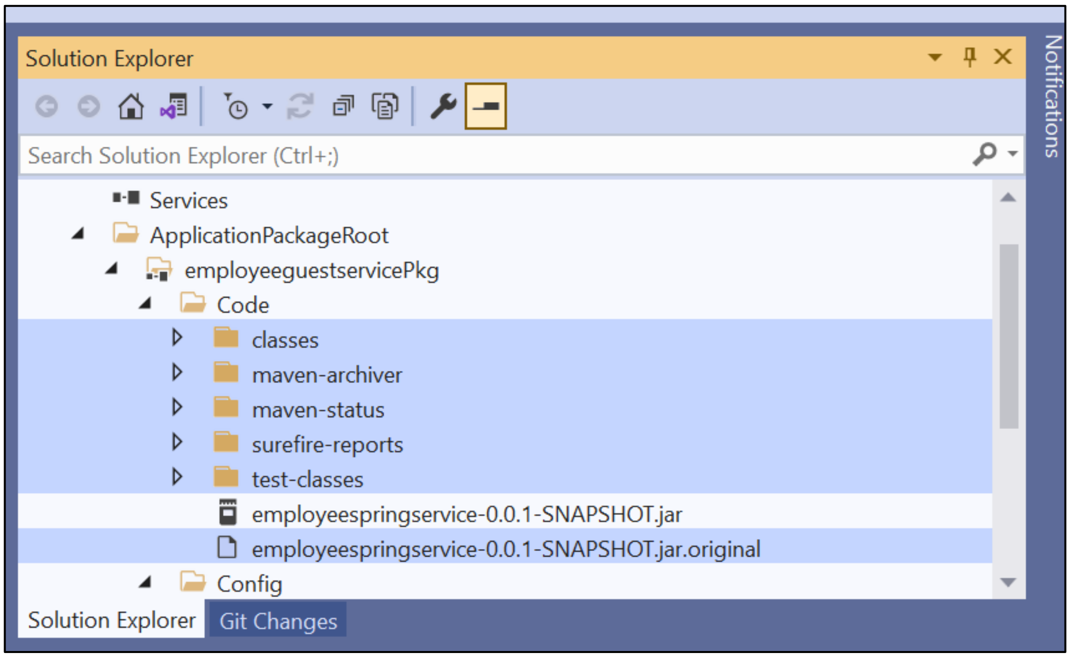
Program:

Arguments:

Working folder: CodeBase

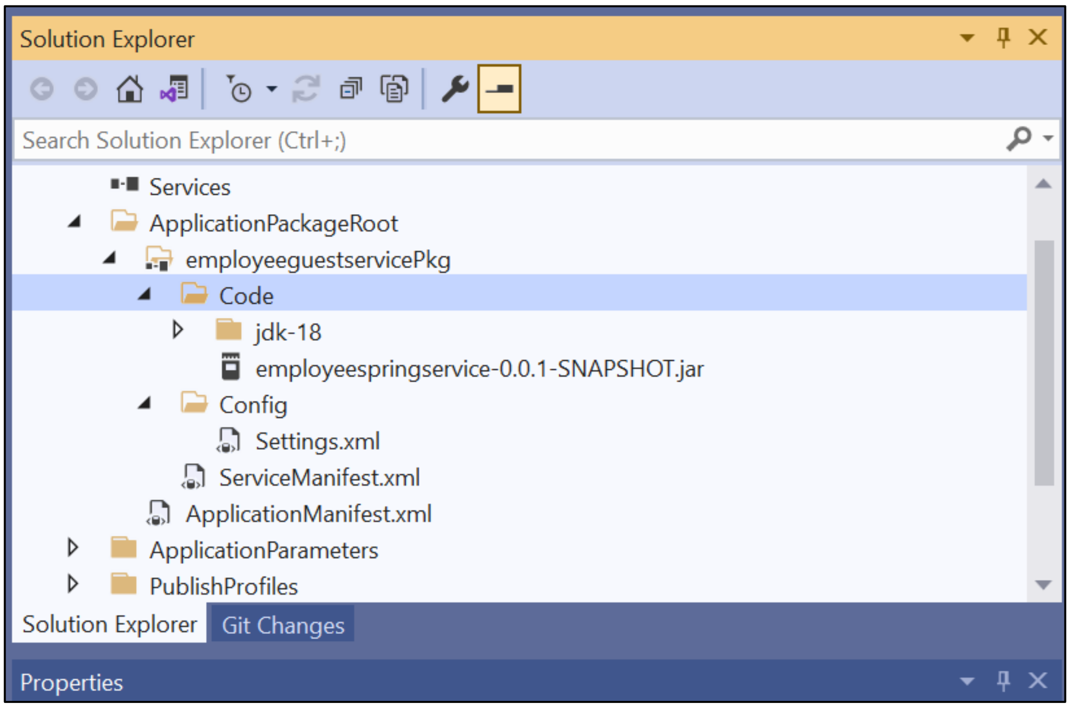
☑ Optimize project layout for ARM deployment. Learn More

- Delete the selected files shown in the Figure below from the Code folder.



Services
ApplicationPackageRoot
employeeguestservicePkg
Code
classes
maven-archiver
maven-status
surefire-reports
test-classes
employeespringservice-0.0.1-SNAPSHOT.jar
employeespringservice-0.0.1-SNAPSHOT.jar.original
Config

- We also need to upload the runtime to run the JAR. Generally, it resides in the JDK installation folder. Paste it in the Code folder, as shown in the Figure below.



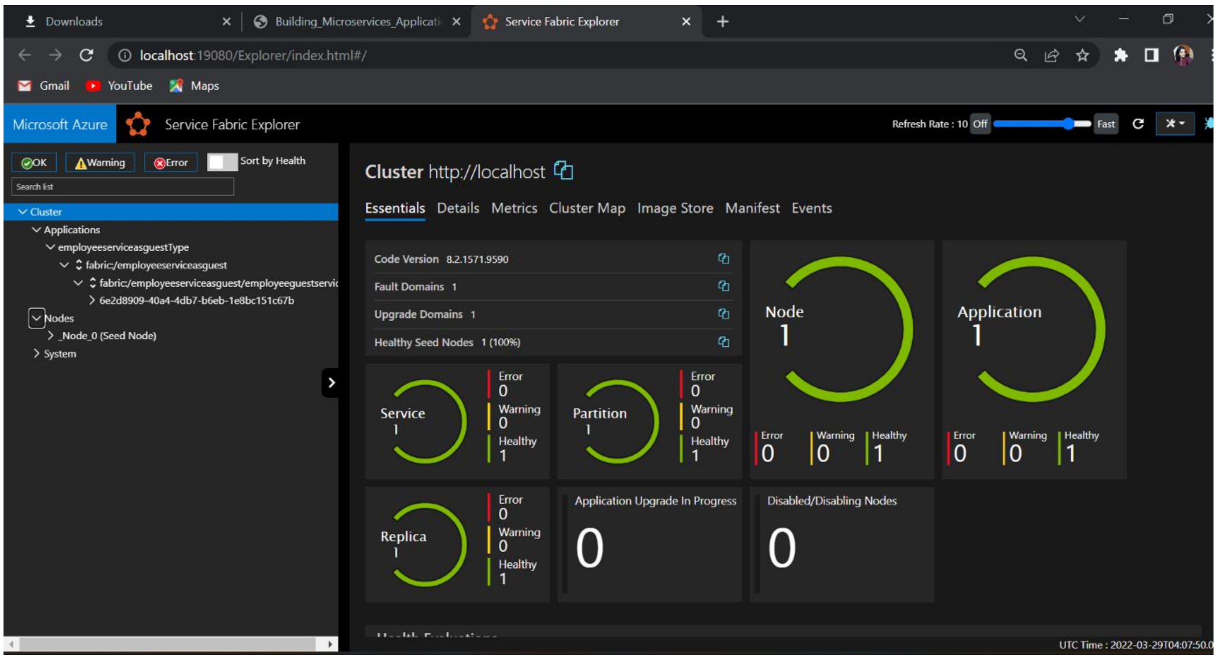
- Open ServiceManisfest.xml and set the following values.

```
<EntryPoint>
      <ExeHost>
        <Program>jdk-18\bin\java.exe</Program>
        <Arguments>-jar ..\..\employeespringservice-0.0.1-SNAPSHOT.jar</Arguments>
        <WorkingFolder>CodeBase</WorkingFolder>
        <!-- Uncomment to log console output (both stdout and stderr) to one of
the
             service's working directories. -->
        <!-- <ConsoleRedirection FileRetentionCount="5" FileMaxSizeInKb="2048"/> -
->
      </ExeHost>
    </EntryPoint>
  </CodePackage>
```

```
<Resources>
    <Endpoints>
      <!-- This endpoint is used by the communication listener to obtain the port
on which to
           listen. Please note that if your service is partitioned, this port is
shared with
```

```
          replicas of different partitions that are placed in your code. -->
      <Endpoint Name="employeeguestserviceTypeEndpoint" Protocol="http"
Port="8080" Type="Input"/>
    </Endpoints>
  </Resources>
```

- Make sure that the local Service Fabric cluster is up and running. Click F5. Browse the Service Fabric dashboard, as shown in the Figure below. The default URL is http://localhost:19080/Explorer/index.html. You see that your service is deployed.



localhost:19080/Explorer/index.html#/

Cluster http://localhost

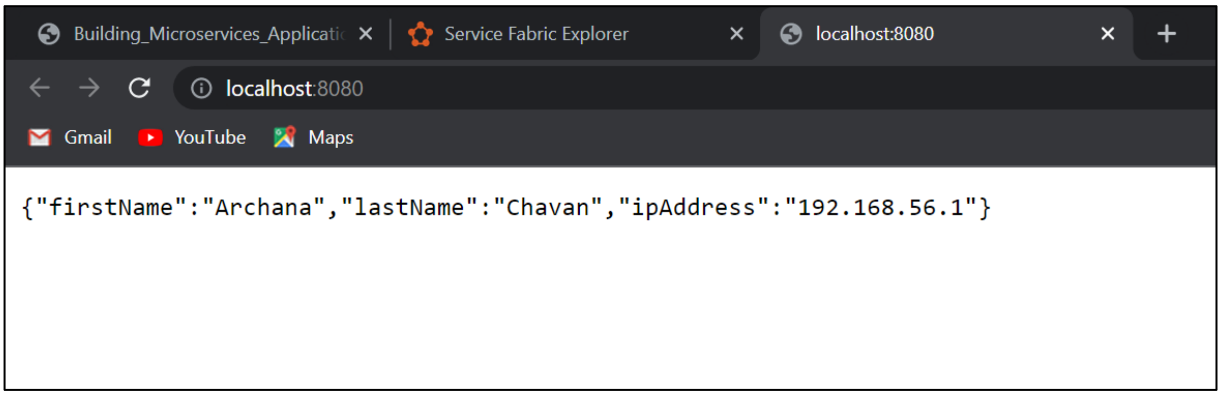
Cluster
- Applications
  - employeeserviceasguestType
    - fabric:/employeeserviceasguest
      - fabric:/employeeserviceasguest/employeeguestservic
        - 6e2d8909-40a4-4db7-b6eb-1e8bc151c67b
- Nodes
  - _Node_0 (Seed Node)
- System

Code Version 8.2.1571.9590
Fault Domains 1
Upgrade Domains 1
Healthy Seed Nodes 1 (100%)

Node 1 — Error 0, Warning 0, Healthy 1
Application 1 — Error 0, Warning 0, Healthy 1
Service 1 — Error 0, Warning 0, Healthy 1
Partition 1 — Error 0, Warning 0, Healthy 1
Replica 1 — Error 0, Warning 0, Healthy 1
Application Upgrade In Progress 0
Disabled/Disabling Nodes 0

UTC Time : 2022-03-29T04:07:50.0

- Browse http://localhost:8080 to access your service. In servicemanifest.xml, we specified the service port as 8080; you can browse the same on 8080, as shown in the Figure below.



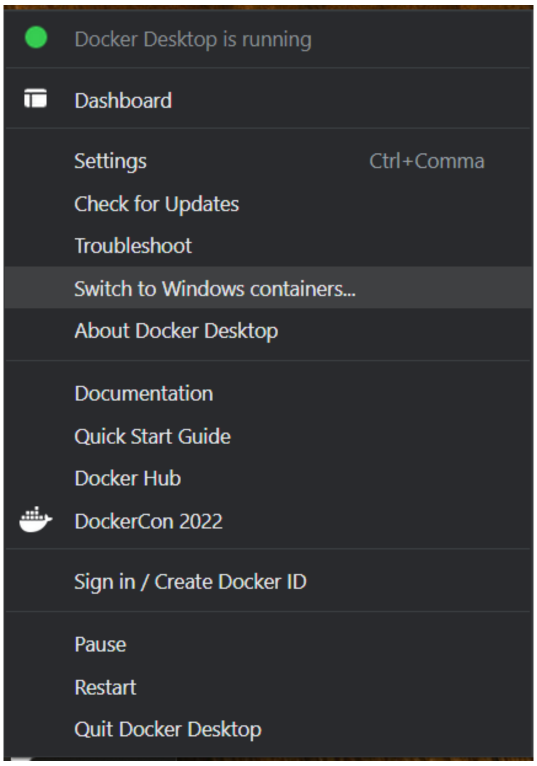
localhost:8080

```
{"firstName":"Archana","lastName":"Chavan","ipAddress":"192.168.56.1"}
```

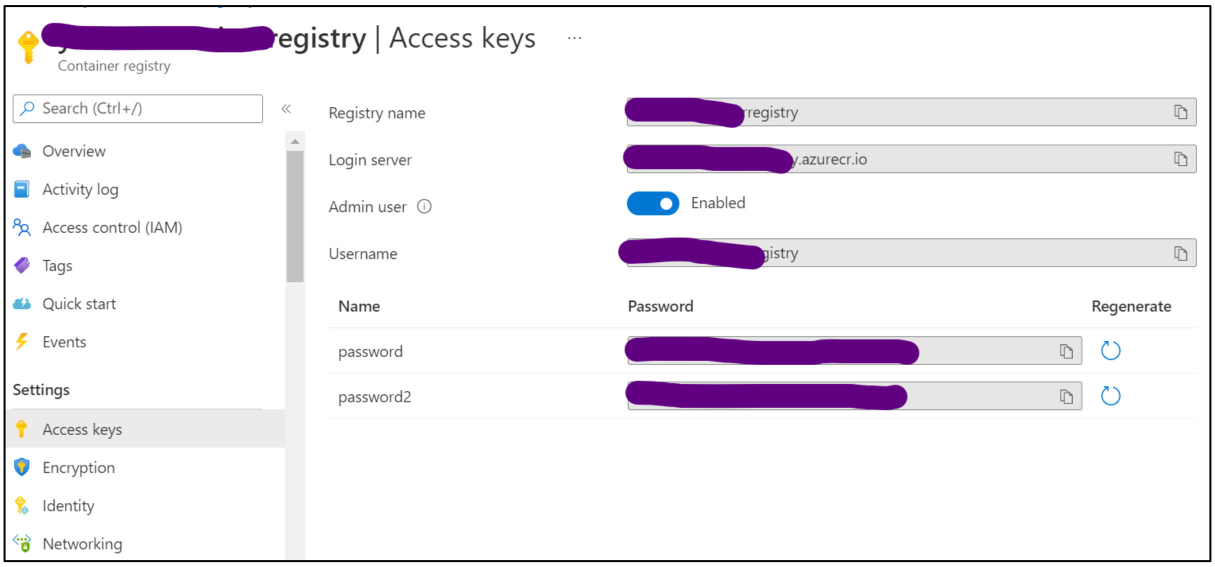
Deploy a Spring Boot Service as a Container

- So far, we have deployed the service as a guest executable in Service Fabric. Now we will follow the steps to deploy the Spring service as a container in Service Fabric. This explains that in addition to creating stateful and stateless services, Service Fabric also orchestrates containers like any other orchestrator, even if the application wasn't developed on a Microsoft stack.

- Open Visual Studio Code. Open the folder where *employeespringservice* exists. Open the *Dockerfile*.
- Make sure that the name of the JAR file is correct.
- Select Switch to Windows container... in Docker Desktop, as shown in Figure below



- Create the Azure Container Registry resource in the Azure portal. Enable the admin user, as shown in the Figure below



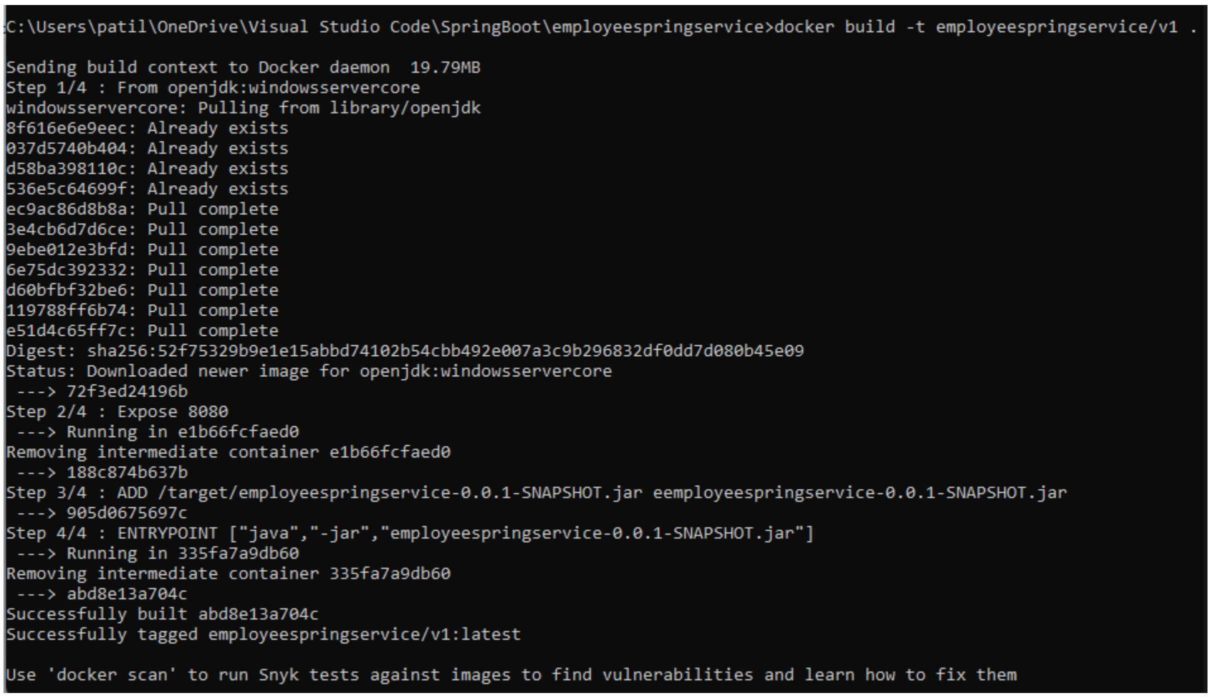
[redacted]registry | Access keys

Container registry

Registry name	[redacted]registry
Login server	[redacted].azurecr.io
Admin user	Enabled
Username	[redacted]gistry

Name	Password	Regenerate
password	[redacted]	
password2	[redacted]	

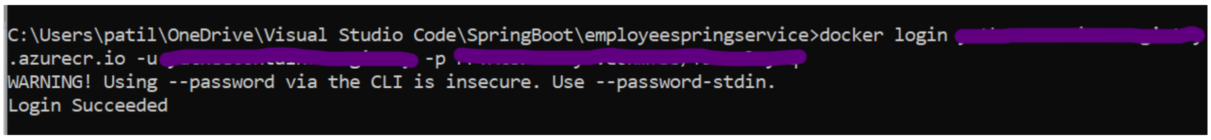
- Open the command prompt in Administrative Mode and browse to the directory where the Docker file exists.
- Fire the following command, including the period at the end. (This may take time because it downloads the Window Server core image from the Docker hub, as shown in Figure below)
  docker build -t employeespringservice/v1 .



```
C:\Users\patil\OneDrive\Visual Studio Code\SpringBoot\employeespringservice>docker build -t employeespringservice/v1 .

Sending build context to Docker daemon  19.79MB
Step 1/4 : From openjdk:windowsservercore
windowsservercore: Pulling from library/openjdk
8f616e6e9eec: Already exists
037d5740b404: Already exists
d58ba398110c: Already exists
536e5c64699f: Already exists
ec9ac86d8b8a: Pull complete
3e4cb6d7d6ce: Pull complete
9ebe012e3bfd: Pull complete
6e75dc392332: Pull complete
d60bfbf32be6: Pull complete
119788ff6b74: Pull complete
e51d4c65ff7c: Pull complete
Digest: sha256:52f75329b9e1e15abbd74102b54cbb492e007a3c9b296832df0dd7d080b45e09
Status: Downloaded newer image for openjdk:windowsservercore
 ---> 72f3ed24196b
Step 2/4 : Expose 8080
 ---> Running in e1b66fcfaed0
Removing intermediate container e1b66fcfaed0
 ---> 188c874b637b
Step 3/4 : ADD /target/employeespringservice-0.0.1-SNAPSHOT.jar eemployeespringservice-0.0.1-SNAPSHOT.jar
 ---> 905d0675697c
Step 4/4 : ENTRYPOINT ["java","-jar","employeespringservice-0.0.1-SNAPSHOT.jar"]
 ---> Running in 335fa7a9db60
Removing intermediate container 335fa7a9db60
 ---> abd8e13a704c
Successfully built abd8e13a704c
Successfully tagged employeespringservice/v1:latest

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

- Now the container image is available locally. You have to push the image to Azure Container Registry.
- Log in to Azure Container Registry using the admin username and password. Use the following command (also see Figure below).

docker login youracr.azurecr.io -u yourusername -p yourpassword

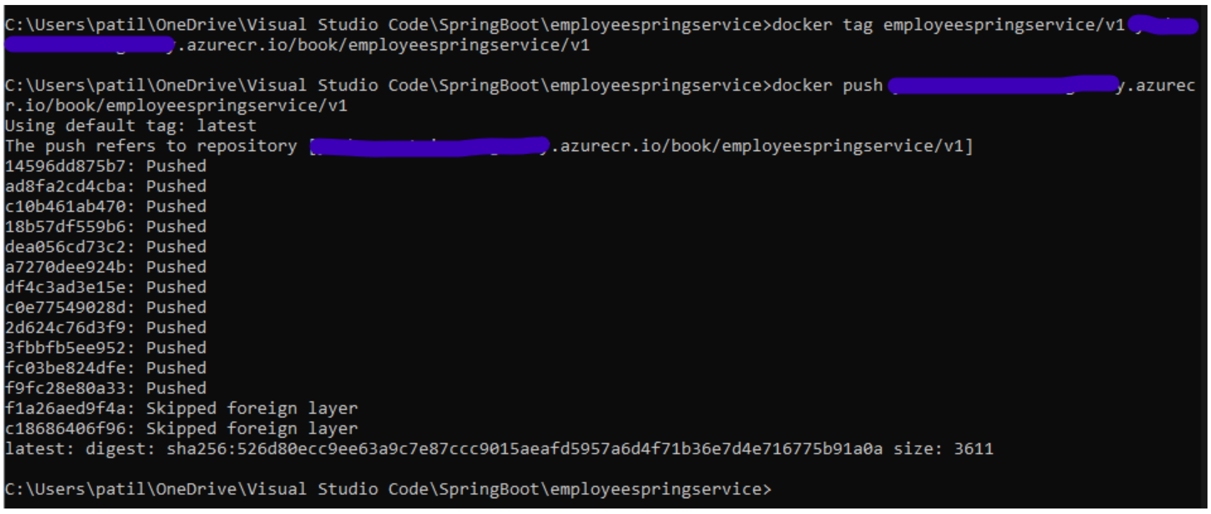


```
C:\Users\patil\OneDrive\Visual Studio Code\SpringBoot\employeespringservice>docker login [redacted]
.azurecr.io -u [redacted] -p [redacted]
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
```

Fire the following commands to upload the image to ACR (as shown in the Figure below).

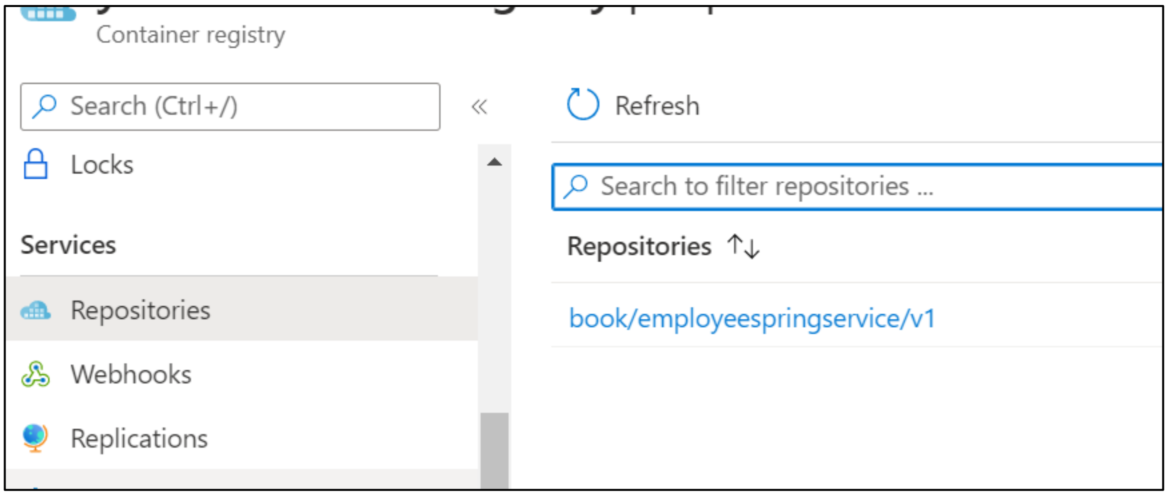
docker tag employeespringservice/v1 youracr.azurecr.io/book/employeespringservice/v1

docker push myservicefabric.azurecr.io/book/employeespringservice/v1



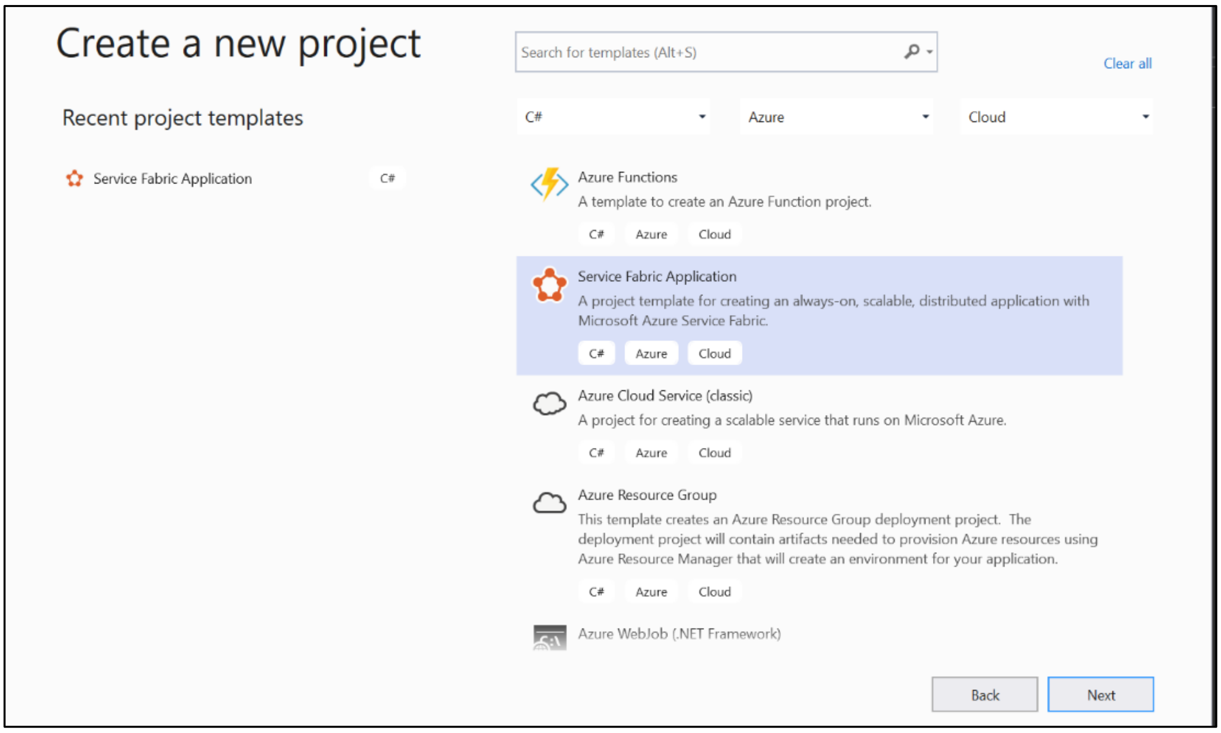
```
C:\Users\patil\OneDrive\Visual Studio Code\SpringBoot\employeespringservice>docker tag employeespringservice/v1 [redacted]
[redacted].azurecr.io/book/employeespringservice/v1

C:\Users\patil\OneDrive\Visual Studio Code\SpringBoot\employeespringservice>docker push [redacted].azurec
r.io/book/employeespringservice/v1
Using default tag: latest
The push refers to repository [[redacted].azurecr.io/book/employeespringservice/v1]
14596dd875b7: Pushed
ad8fa2cd4cba: Pushed
c10b461ab470: Pushed
18b57df559b6: Pushed
dea056cd73c2: Pushed
a7270dee924b: Pushed
df4c3ad3e15e: Pushed
c0e77549028d: Pushed
2d624c76d3f9: Pushed
3fbbfb5ee952: Pushed
fc03be824dfe: Pushed
f9fc28e80a33: Pushed
f1a26aed9f4a: Skipped foreign layer
c18686406f96: Skipped foreign layer
latest: digest: sha256:526d80ecc9ee63a9c7e87ccc9015aeafd5957a6d4f71b36e7d4e716775b91a0a size: 3611

C:\Users\patil\OneDrive\Visual Studio Code\SpringBoot\employeespringservice>
```

- Log in to the Azure portal and check if you can see your image in Repositories, as shown in the Figure below.



Since the container image is ready and uploaded in Azure Container Registry, let's create a Service Fabric project to deploy the container to the local Service Fabric cluster.

- Launch Visual Studio 2019 as an administrator.
- Create a New Project
- Select C# ➤ Azure ➤ Cloud ➤ Service Fabric Application ➤ Next


Create a new project

Search for templates (Alt+S)

Clear all

Recent project templates

Service Fabric Application C#

C# | Azure | Cloud

Azure Functions
A template to create an Azure Function project.
C# Azure Cloud

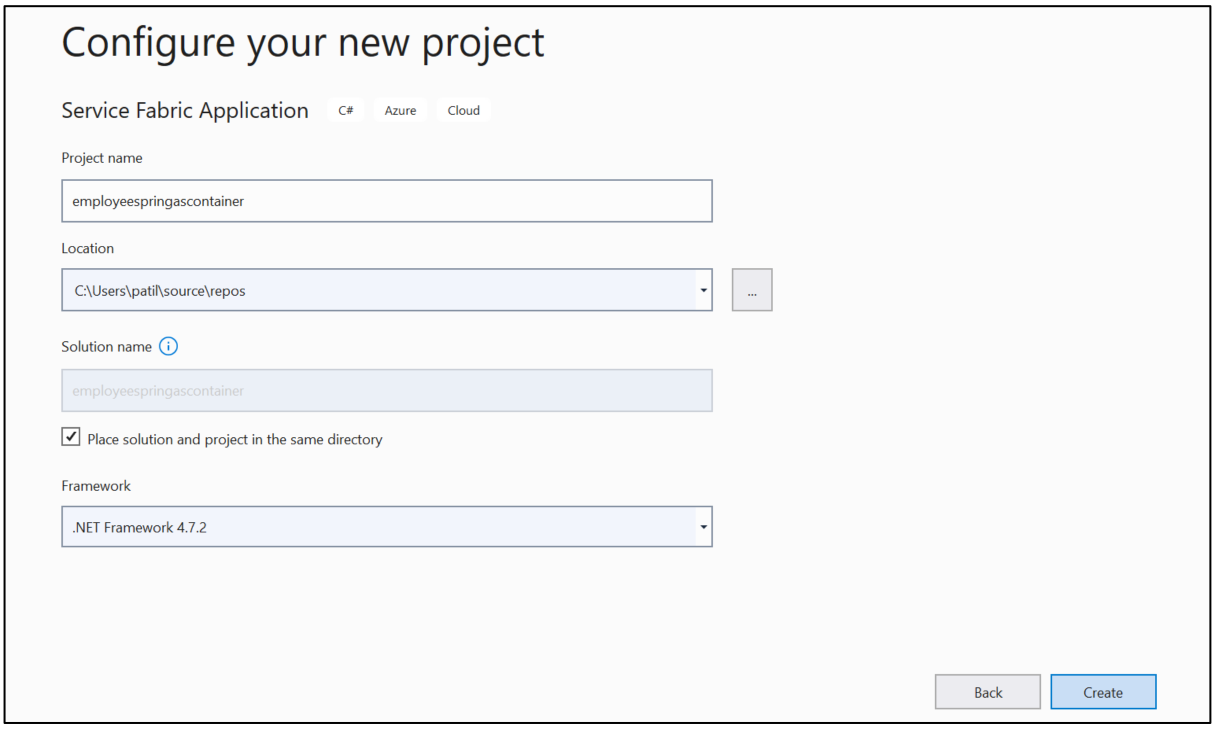
Service Fabric Application
A project template for creating an always-on, scalable, distributed application with Microsoft Azure Service Fabric.
C# Azure Cloud

Azure Cloud Service (classic)
A project for creating a scalable service that runs on Microsoft Azure.
C# Azure Cloud

Azure Resource Group
This template creates an Azure Resource Group deployment project. The deployment project will contain artifacts needed to provision Azure resources using Azure Resource Manager that will create an environment for your application.
C# Azure Cloud

Azure WebJob (.NET Framework)

- Name the application *employeespringascontainer*, as shown in Figure ➤ Create


Configure your new project

Service Fabric Application C# Azure Cloud

Project name: employeespringascontainer

Location: C:\Users\patil\source\repos

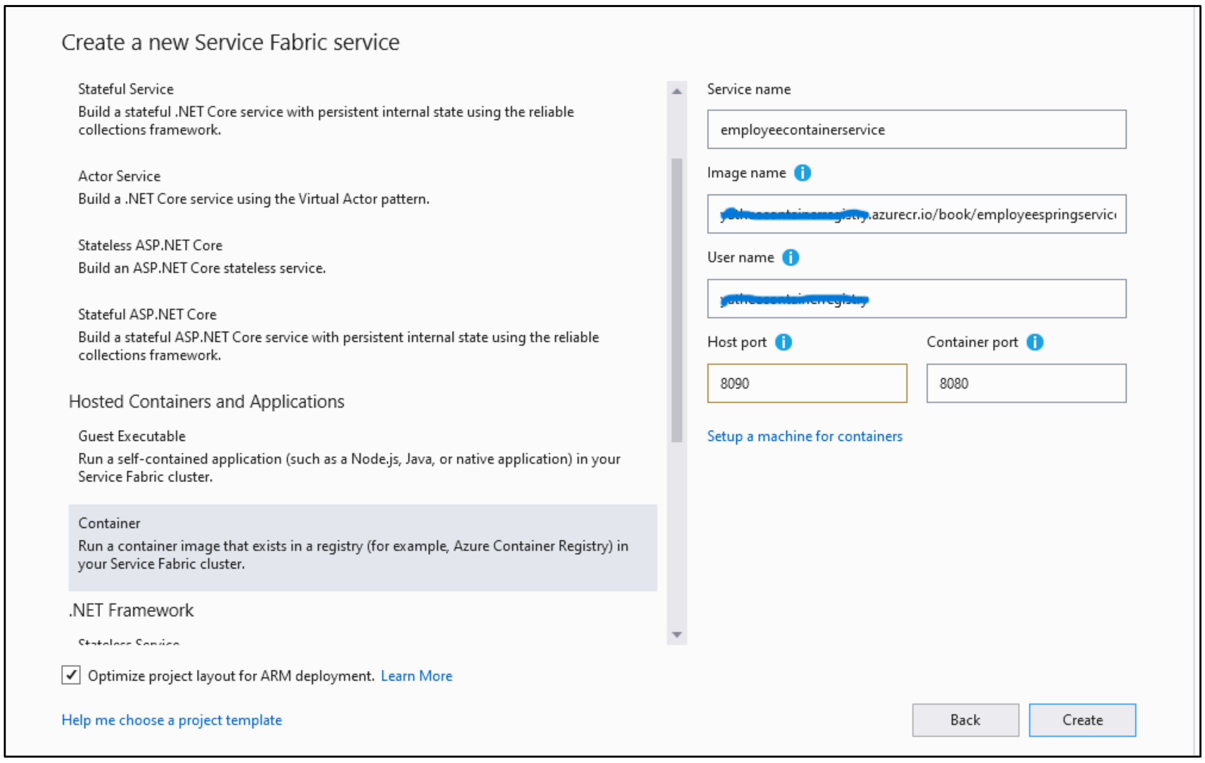
Solution name: employeespringascontainer

☑ Place solution and project in the same directory

Framework: .NET Framework 4.7.2

In New Service Fabric Service, select the following.

- Service Name: employeecontainerservice
- Image Name: youracr.azurecr.io/book/employeespringservice/v1
- User Name: Your username in the Azure Container Registry
- Host Port: 8090
- Container Port: 8080



Create a new Service Fabric service

Stateful Service
Build a stateful .NET Core service with persistent internal state using the reliable collections framework.

Actor Service
Build a .NET Core service using the Virtual Actor pattern.

Stateless ASP.NET Core
Build an ASP.NET Core stateless service.

Stateful ASP.NET Core
Build a stateful ASP.NET Core service with persistent internal state using the reliable collections framework.

Hosted Containers and Applications

Guest Executable
Run a self-contained application (such as a Node.js, Java, or native application) in your Service Fabric cluster.

Container
Run a container image that exists in a registry (for example, Azure Container Registry) in your Service Fabric cluster.

.NET Framework

Service name: employeecontainerservice

Image name: [redacted].azurecr.io/book/employeespringservic

User name: [redacted]

Host port: 8090

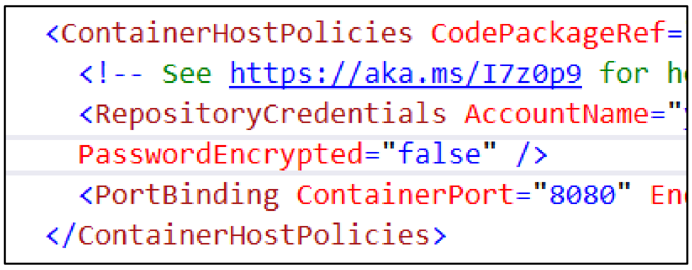
Container port: 8080

Setup a machine for containers

☑ Optimize project layout for ARM deployment. Learn More

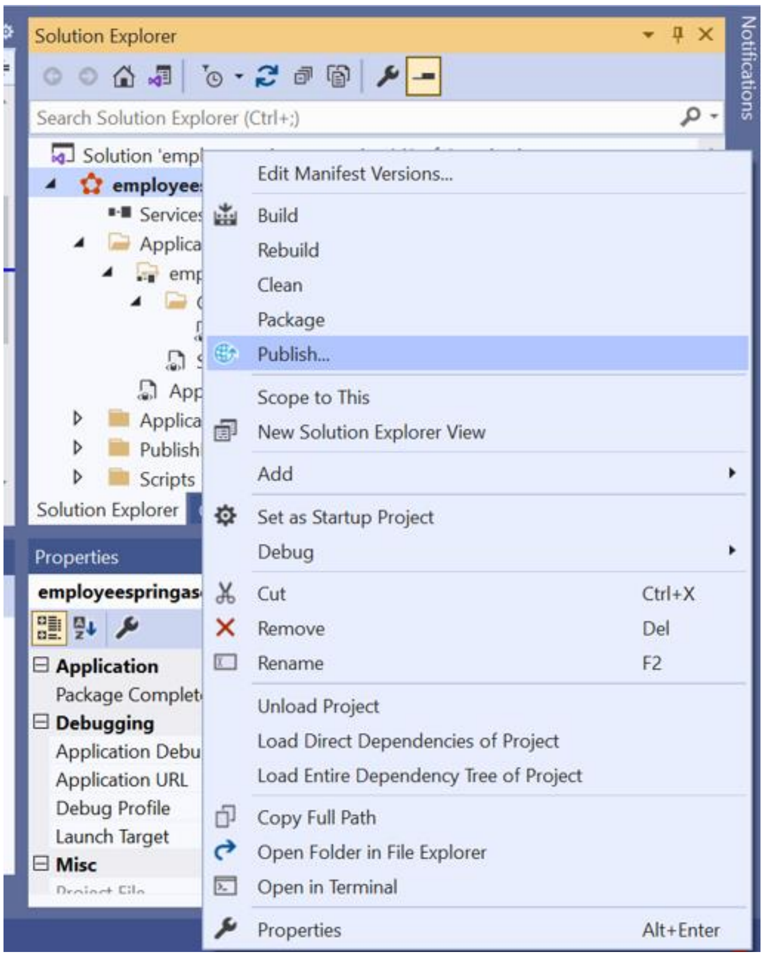
Help me choose a project template

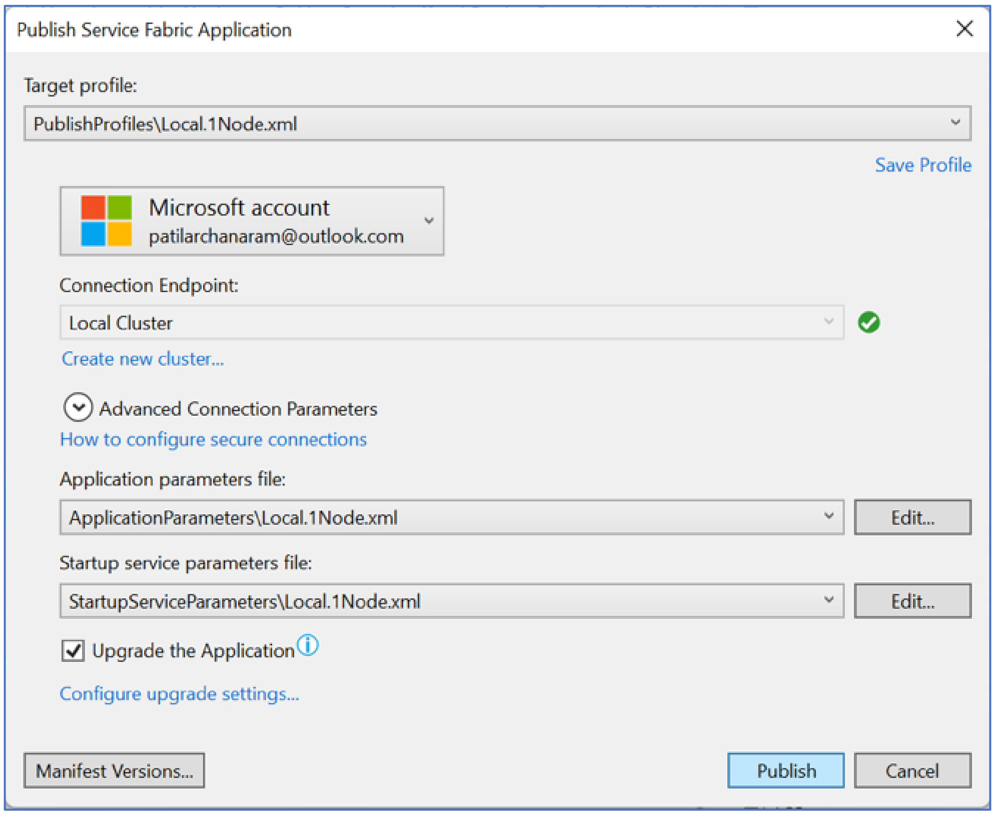
- Once the solution is created, open the ApplicationManifest.xml. Specify the right password for the admin user. (Since this is a sample, we kept the password unencrypted; for real-world applications, you have to encrypt the password.)



```
<ContainerHostPolicies CodePackageRef=
  <!-- See https://aka.ms/I7z0p9 for h
  <RepositoryCredentials AccountName="
  PasswordEncrypted="false" />
  <PortBinding ContainerPort="8080" En
</ContainerHostPolicies>
```

Now we are ready to build and deploy the container to the local Service Fabric cluster. Since we have given the user information to download the image from Azure Container Registry, Visual Studio downloads and deploys the container to the local Service Fabric cluster. Right-click the Service Fabric project and publish, as shown in the Figure below.




Publish Service Fabric Application

Target profile: PublishProfiles\Local.1Node.xml

Microsoft account
patilarchanaram@outlook.com

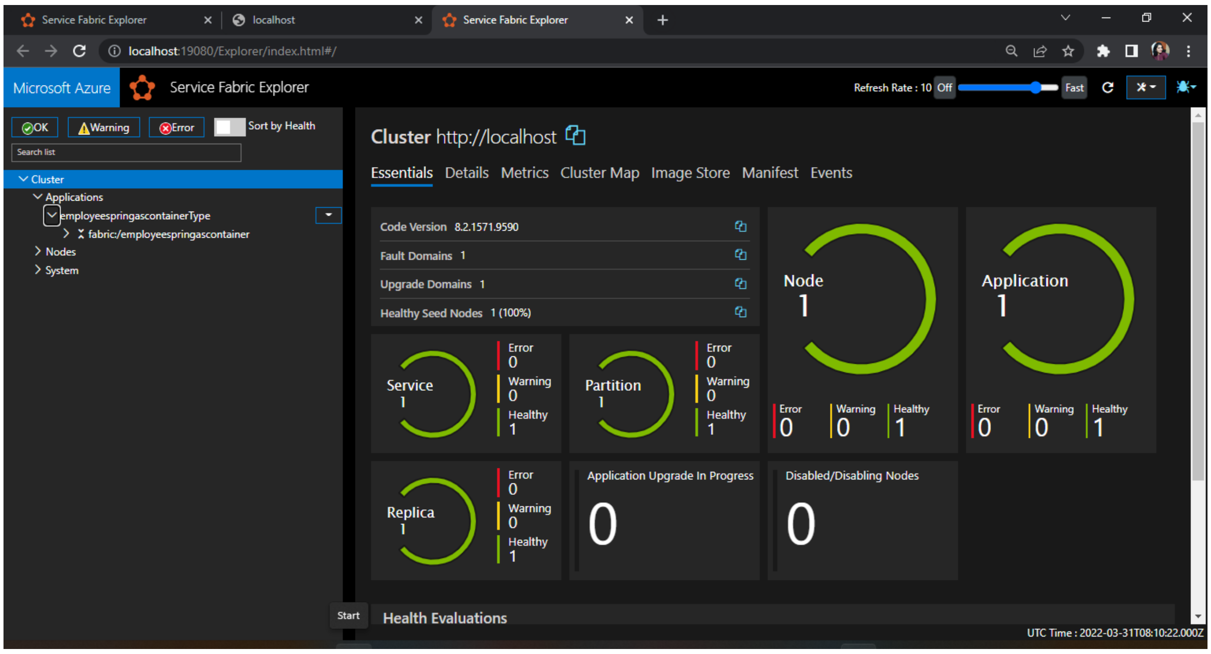
Connection Endpoint: Local Cluster

Application parameters file: ApplicationParameters\Local.1Node.xml

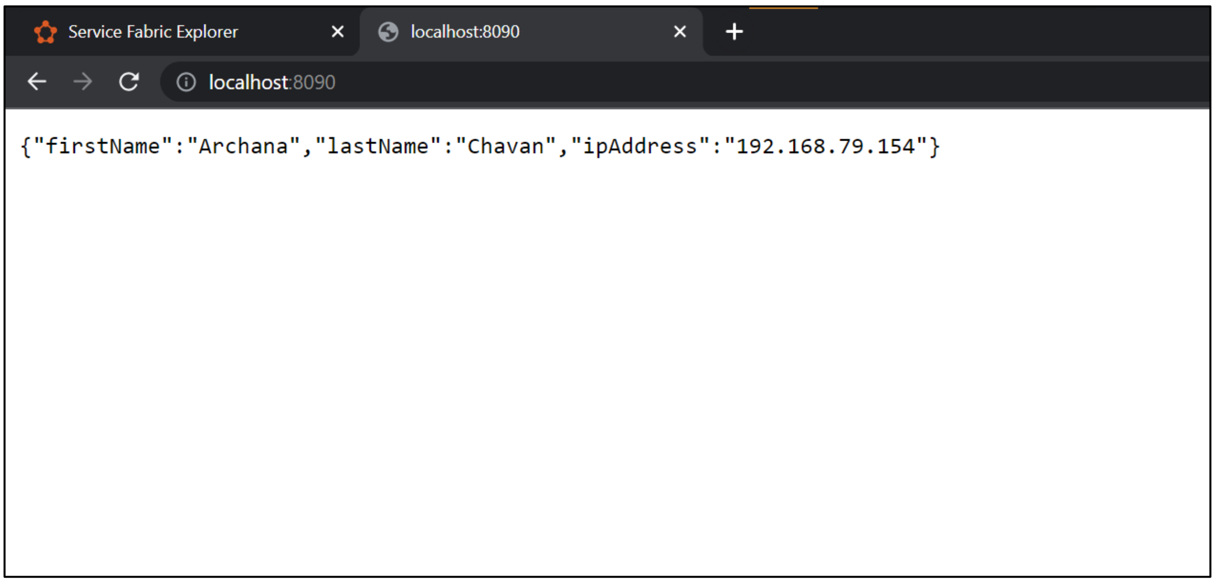
Startup service parameters file: StartupServiceParameters\Local.1Node.xml

[✓] Upgrade the Application
Save Profile
Create new cluster...
Advanced Connection Parameters
How to configure secure connections
Edit...
Edit...
Configure upgrade settings...
Manifest Versions...
Publish
Cancel

- Browse the Service Fabric dashboard. The default URL is *http://localhost:19080/Explorer/index.html*.
- Your service is deployed, as shown below



- Browse to http://localhost:8090/ to access your service. You get the response shown in the Figure below, which is served from the container run by Service Fabric.



```
{"firstName":"Archana","lastName":"Chavan","ipAddress":"192.168.79.154"}
```