

How to work with Azure Function

1. Install Azure CLI using Node NPM package manager

- Goto <https://nodejs.org> download and install latest version of nodejs
- Goto terminal and type: `npm install -g azure-cli`

```

Last login: Fri May 1 21:15:58 on tty000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Rajs-Air:~ rajbhatta$ npm install -g azure-cli
npm WARN deprecated azure-keyvault@3.0.5: This package has been deprecated. Please use @azure/keyvault-keys, @azure/keyvault-secrets or @azure/keyvault-certificates instead
npm WARN deprecated github@0.1.6: 'github' has been renamed to '@octokit/rest' (https://git.io/vNB11)
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated request@2.74.0: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated hawk@3.1.3: This module moved to @hapi/hawk. Please make sure to switch over as this distribution is no longer supported and may contain bugs and critical security issues.
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
npm WARN deprecated sntp@1.0.9: This module moved to @hapi/sntp. Please make sure to switch over as this distribution is no longer supported and may contain bugs and critical security issues.
npm WARN deprecated boom@2.10.1: This version has been deprecated in accordance with the hapi support policy (hapi.im/support). Please upgrade to the latest version

```

- Once installation is completed, use terminal and type `az`

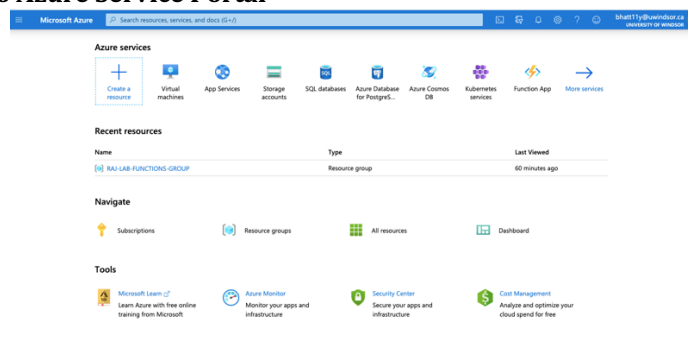
```

updated 1 package in 8.782s
[Rajs-Air:~ rajbhatta$ azure
info:
info:
info:
info:
info:
info:
info:
info: Microsoft Azure: Microsoft's Cloud Platform
info:
info: Tool version 0.10.20
help:
help: Display help for a given command
help: help [options] [command]
help:
help: Log in to an Azure subscription using Active Directory or a Microsoft a
ccount identity.
help: login [options]
help:
help: Log out from Azure subscription using Active Directory. Currently, the
user can log out only via Microsoft organizational account
help: logout [options] [username]
help:

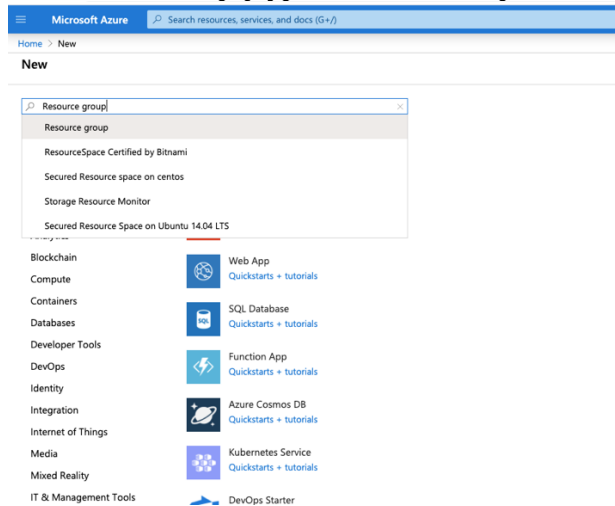
```

2. Work with Azure Service Portal Now (I'm using Azure Service provided for student)

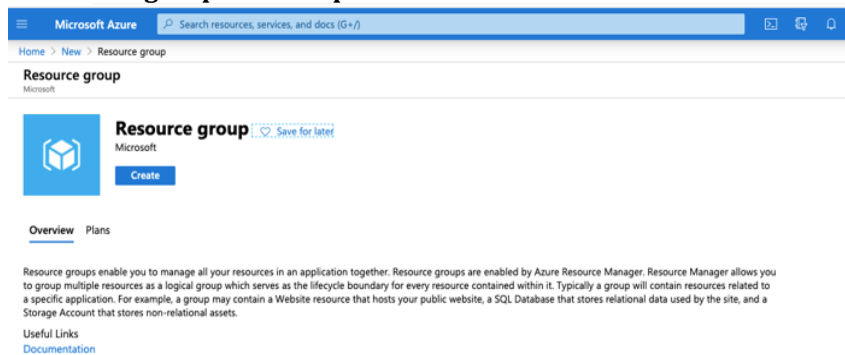
- **Goto Azure Service Portal**



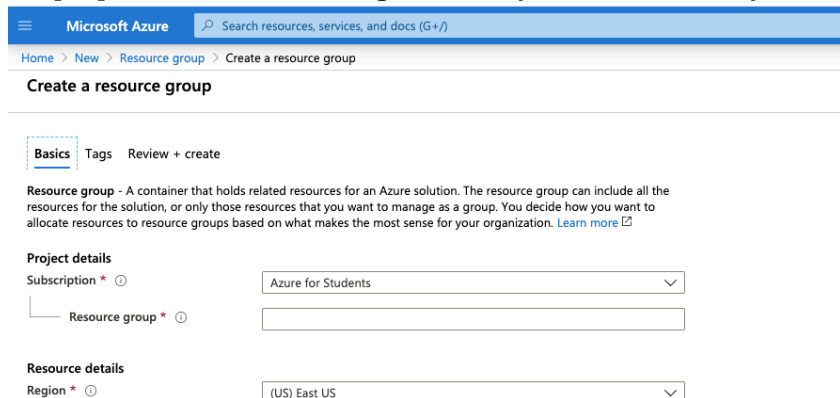
- **Create Resource Group (Type Resource Group on search bar)**



- **Select resource group from dropdown and click create button**

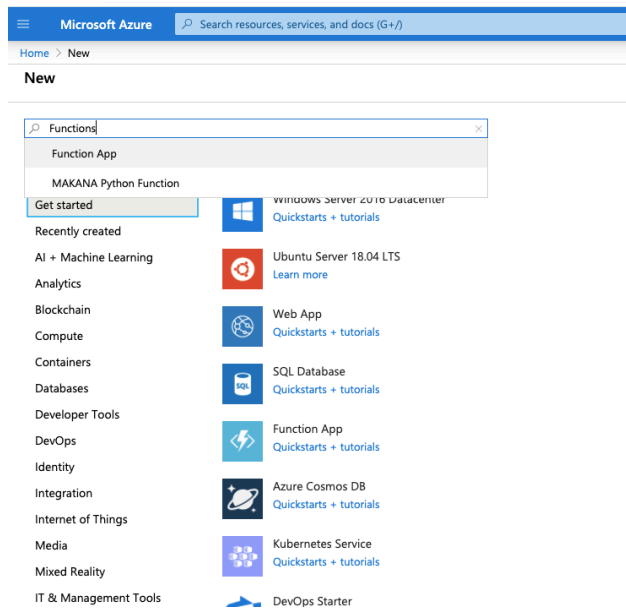


- **Provide proper name and select region where you want to create your resource group**

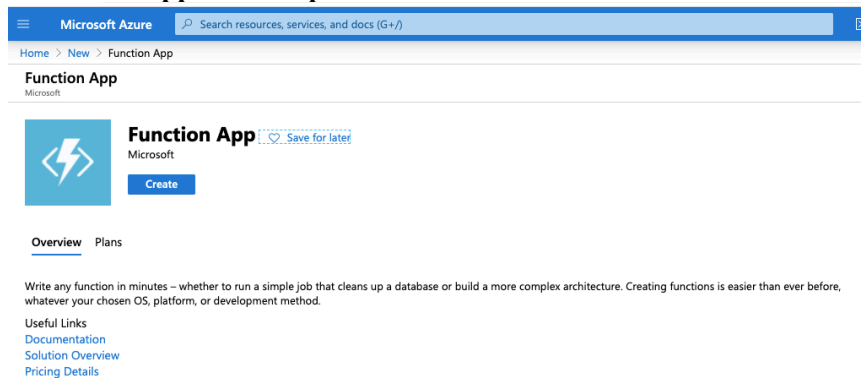


3. Create HttpTrigger Function

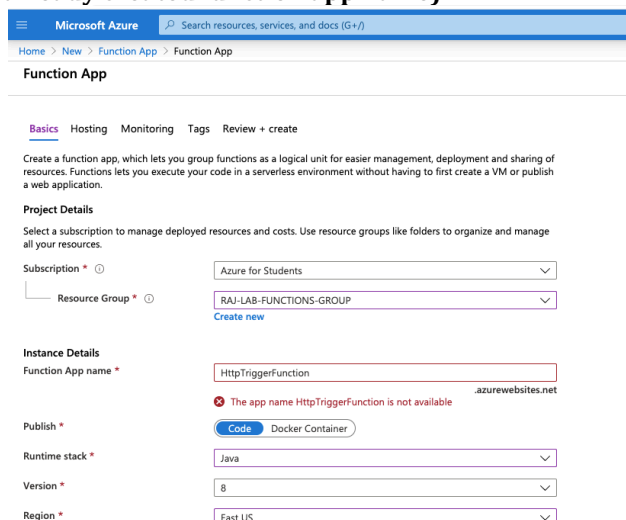
- **Goto Azure service portal home and search Function App**



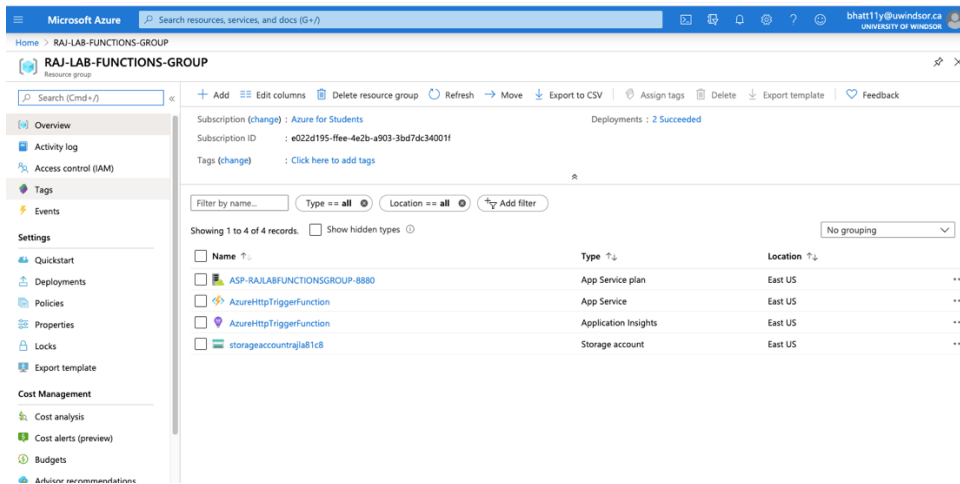
- **Select Function App from dropdown and click on Create**



- **Provide Proper Function App name along with proper language version (In my case I had already created function app name)**

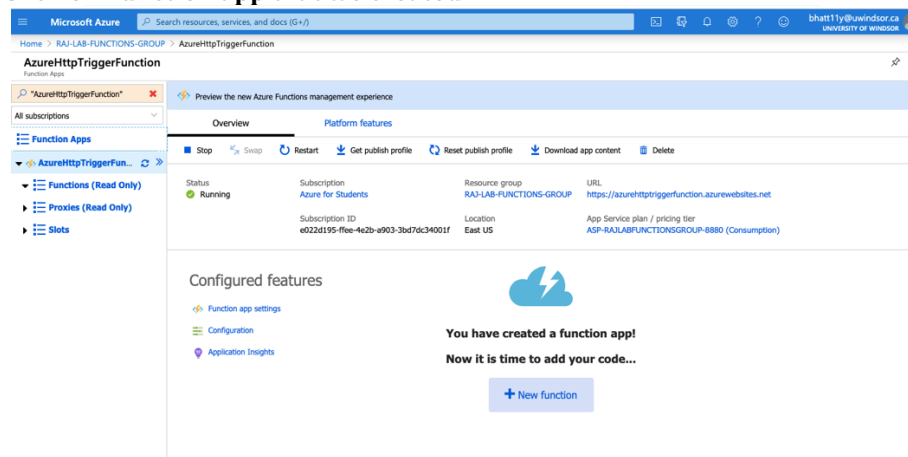


- **Now Goto Home -> Resource Group -> Select created resource group -> then you can see created Function App inside it as shown below:**

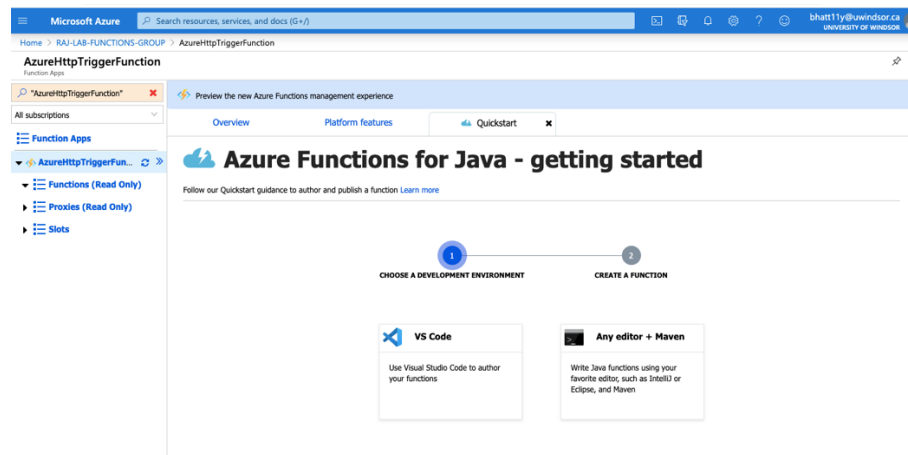


4. Create function inside Function App

- Click on Function app that we created



- Click on New Function



- **Select Any editor+Maven**

Microsoft Azure

Home > RAJ-LAB-FUNCTIONS-GROUP > AzureHttpTriggerFunction

AzureHttpTriggerFunction

Function Apps

"AzureHttpTriggerFunction"

All subscriptions

Function Apps

AzureHttpTriggerFun...

Functions (Read Only)

Proxies (Read Only)

Slots

Preview the new Azure Functions management experience

Overview Platform features Quickstart

Azure Functions for Java - getting started

Follow our Quickstart guidance to author and publish a function [Learn more](#)

1 CHOOSE A DEVELOPMENT ENVIRONMENT 2 CHOOSE A DEPLOYMENT METHOD 3 CREATE A FUNCTION

VS Code

Use Visual Studio Code to author your functions

Any editor + Maven

Write Java functions using your favorite editor, such as IntelliJ or Eclipse, and Maven

- **Direct publish using Maven**

Microsoft Azure

Home > RAJ-LAB-FUNCTIONS-GROUP > AzureHttpTriggerFunction

AzureHttpTriggerFunction

Function Apps

"AzureHttpTriggerFunction"

All subscriptions

Function Apps

AzureHttpTriggerFun...

Functions (Read Only)

Proxies (Read Only)

Slots

Preview the new Azure Functions management experience

Overview Platform features Quickstart

Azure Functions for Java - getting started

Follow our Quickstart guidance to author and publish a function [Learn more](#)

1 CHOOSE A DEVELOPMENT ENVIRONMENT 2 CHOOSE A DEPLOYMENT METHOD 3 CREATE A FUNCTION

Direct publish

Publish directly from Maven

Use Deployment Center

Configure your app so that you can check your code into source control and run it through a continuous deployment pipeline

- **Work with the provided command on your terminal.**

Microsoft Azure

Home > RAJ-LAB-FUNCTIONS-GROUP > AzureHttpTriggerFunction

AzureHttpTriggerFunction

Function Apps

"AzureHttpTriggerFunction"

All subscriptions

Function Apps

AzureHttpTriggerFun...

Functions (Read Only)

Proxies (Read Only)

Slots

Preview the new Azure Functions management experience

Overview Platform features Quickstart

Azure Functions for Java - getting started

Follow our Quickstart guidance to author and publish a function [Learn more](#)

1 CHOOSE A DEVELOPMENT ENVIRONMENT 2 CHOOSE A DEPLOYMENT METHOD 3 CREATE A FUNCTION

Install dependencies

Before you can get started, you should install the Java Developer Kit, version 8. Make sure that the JAVA_HOME environment variable gets set to the install location of the JDK. You will also need to install Apache Maven, version 3.0 or above.

You should also install Node 3.0 which includes npm. This is how you will obtain the Azure Functions Core Tools. If you prefer not to install Node, see the other installation options in our [Core Tools reference](#).

Run the following command to install the Core Tools package:

```
npm install -g azure-functions-core-tools
```

The Core Tools make use of .NET Core 2.1, so you should install that, too.

Lastly, install the Azure CLI 2.0. Once this is installed, make sure you are logged in by running the login command and following the onscreen instructions:

```
az login
```

Create an Azure Functions project

In the terminal window or from a command prompt, navigate to an empty folder for your project, and run the following command:

```
mvn archetype:generate -DarchetypeGroupId=com.microsoft.azure -DarchetypeArtifactId=azure-functions-archetype -DappName=az-urehttpTriggerFunction -DappRegion=(region) -DresourceGroup=(resourceGroup) -DgroupId=com.(functionAppName).group -DartifactId=(functionAppName)-functions -Dpackage=com.(functionAppName) -DinteractiveMode=false
```

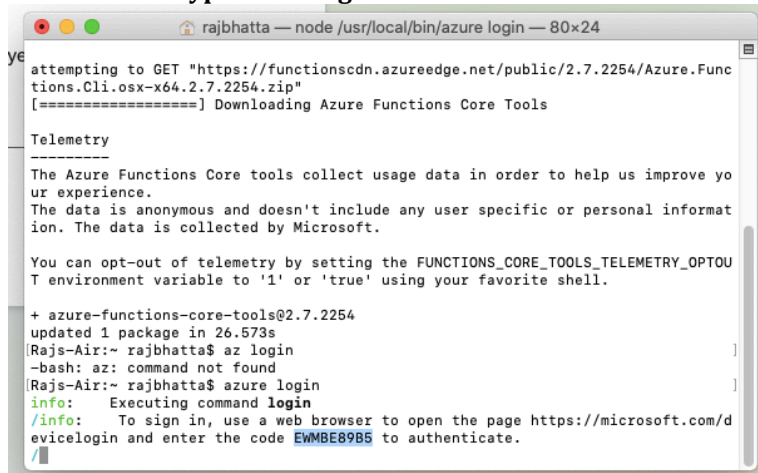
Create a function

Creating the project creates an HTTP function by default, so you don't have to do anything for this step right now. Later, if you want to add a new function, run the following command:

```
az functionapp create --name <functionapp-name> --resource-group <resource-group> --runtime java --functions
```

5. Working with Azure CLI for Azure verification

- Goto terminal and type azure login



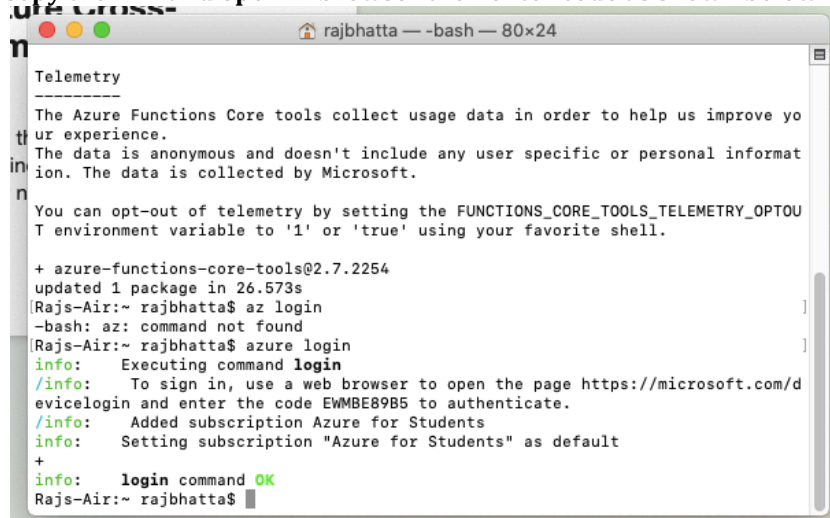
```
rajbhatta — node /usr/local/bin/azure login — 80x24
attempting to GET "https://functionscdn.azureedge.net/public/2.7.2254/Azure.Func
tions.Cli.osx-x64.2.7.2254.zip"
[=====] Downloading Azure Functions Core Tools

Telemetry
-----
The Azure Functions Core tools collect usage data in order to help us improve yo
ur experience.
The data is anonymous and doesn't include any user specific or personal informat
ion. The data is collected by Microsoft.

You can opt-out of telemetry by setting the FUNCTIONS_CORE_TOOLS_TELEMETRY_OPTOU
T environment variable to '1' or 'true' using your favorite shell.

+ azure-functions-core-tools@2.7.2254
updated 1 package in 26.573s
[Rajs-Air:~ rajbhatta$ az login
-bash: az: command not found
[Rajs-Air:~ rajbhatta$ azure login
info: Executing command login
/info: To sign in, use a web browser to open the page https://microsoft.com/d
evicelogin and enter the code EWMBE89B5 to authenticate.
```

- Copy the link and open in browser then enter code as shown below



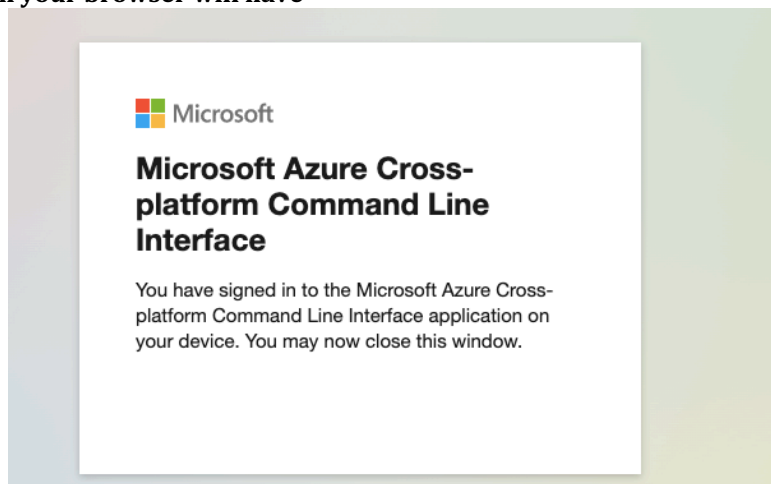
```
rajbhatta — -bash — 80x24

Telemetry
-----
The Azure Functions Core tools collect usage data in order to help us improve yo
ur experience.
The data is anonymous and doesn't include any user specific or personal informat
ion. The data is collected by Microsoft.

You can opt-out of telemetry by setting the FUNCTIONS_CORE_TOOLS_TELEMETRY_OPTOU
T environment variable to '1' or 'true' using your favorite shell.

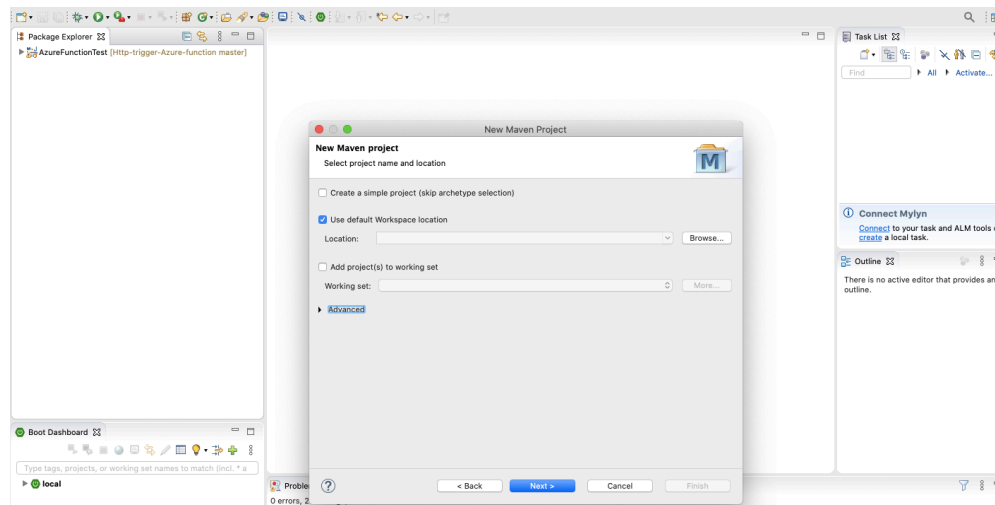
+ azure-functions-core-tools@2.7.2254
updated 1 package in 26.573s
[Rajs-Air:~ rajbhatta$ az login
-bash: az: command not found
[Rajs-Air:~ rajbhatta$ azure login
info: Executing command login
/info: To sign in, use a web browser to open the page https://microsoft.com/d
evicelogin and enter the code EWMBE89B5 to authenticate.
/info: Added subscription Azure for Students
info: Setting subscription "Azure for Students" as default
+
info: login command OK
Rajs-Air:~ rajbhatta$
```

- Then your browser will have

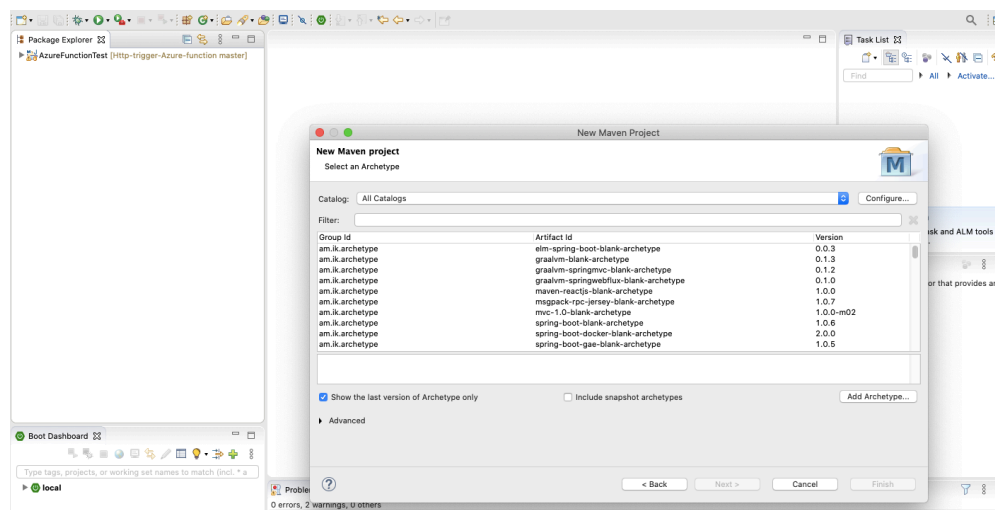


6. Create Azure Trigger Function for development using Eclipse and Maven

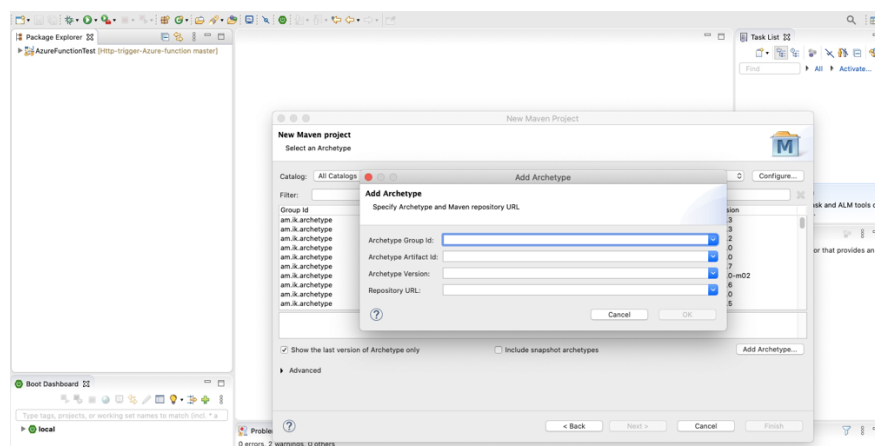
- Create new Maven project



- Click on next

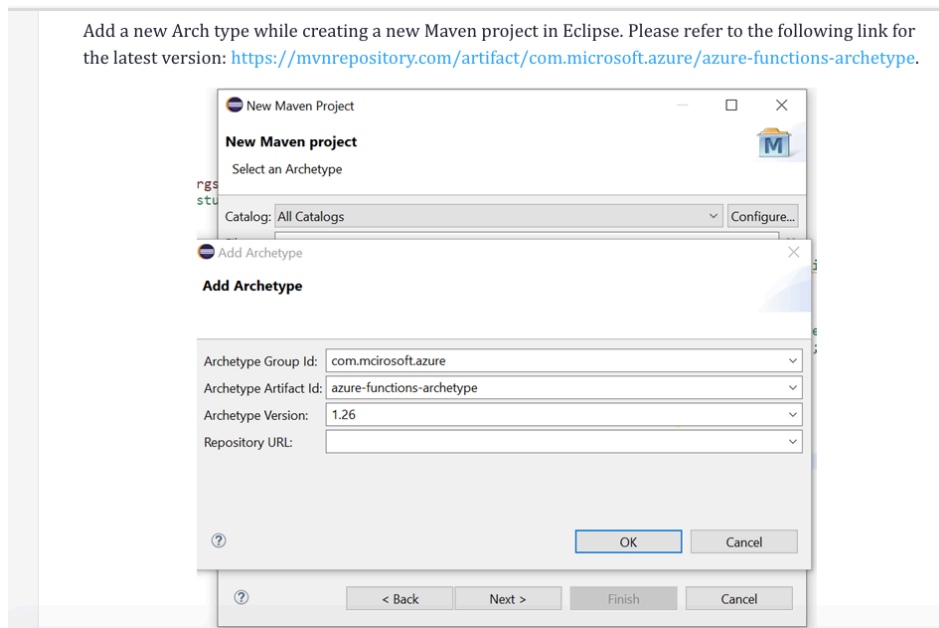


- Click on add Archetype

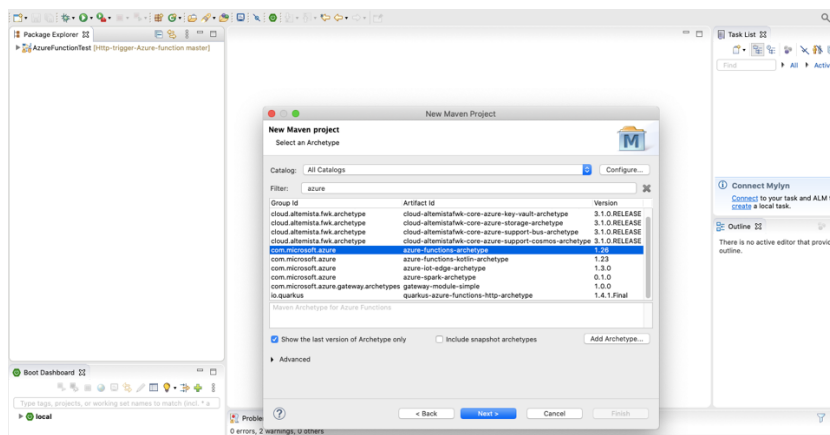


Note: Maven repo for latest Azure function archetype: <https://mvnrepository.com/artifact/com.microsoft.azure/azure-functions-archetype>

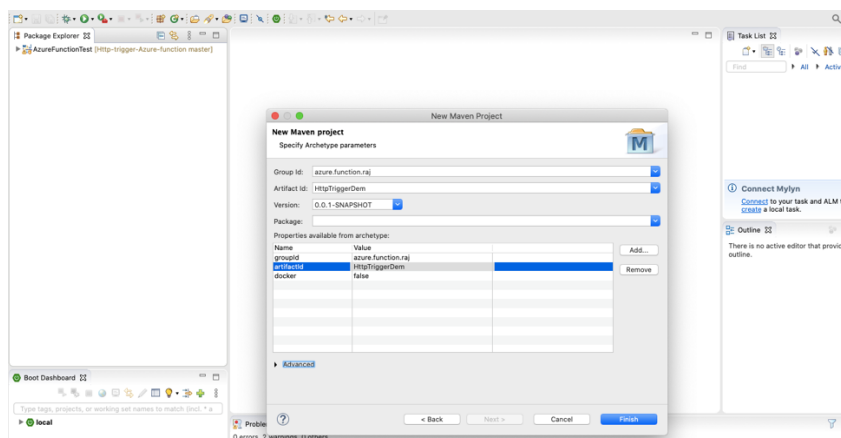
- Provide archetype as shown below:



- Search Azure function



- Fillup groupId and archetype



7. Update resource group, function app and region inside POM.XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.raj</groupId>
  <artifactId>AzureFunctionTest</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>Azure Java Functions</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <azure.functions.maven.plugin.version>1.4.1</azure.functions.maven.plugin.version>
    <azure.functions.java.library.version>1.3.1</azure.functions.java.library.version>
    <functionResourceGroup>RAJ-LAB-FUNCTIONS-GROUP</functionResourceGroup>
    <functionAppName>AzureHttpTriggerFunction</functionAppName>
    <functionAppRegion>westus</functionAppRegion>
    <stagingDirectory>${project.build.directory}/azure-functions/${functionAppName}</stagingDirectory>
  </properties>

  <repositories>
    <repository>
      <id>maven.snapshots</id>
      <name>Maven Central Snapshot Repository</name>
      <url>https://oss.sonatype.org/content/repositories/snapshots</url>
      <releases>
        <enabled>false</enabled>
      </releases>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
    </repository>
  </repositories>

  <pluginRepositories>
    <pluginRepository>
      <id>maven.snapshots</id>
      <name>Maven Central Snapshot Repository</name>
      <url>https://oss.sonatype.org/content/repositories/snapshots</url>
      <releases>
        <enabled>false</enabled>
      </releases>
      <snapshots>
        <enabled>true</enabled>
      </snapshots>
    </pluginRepository>
  </pluginRepositories>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.4.2</version>
      </dependency>
      <dependency>
        <groupId>org.mockito</groupId>
        <artifactId>mockito-core</artifactId>
        <version>2.23.4</version>
      </dependency>
      <dependency>
        <groupId>com.microsoft.azure.functions</groupId>
        <artifactId>azure-functions-java-library</artifactId>
        <version>${azure.functions.java.library.version}</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
```

```

<dependencies>
  <dependency>
    <groupId>com.microsoft.azure.functions</groupId>
    <artifactId>azure-functions-java-library</artifactId>
  </dependency>

  <!-- Test -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>com.microsoft.azure</groupId>
        <artifactId>azure-functions-maven-plugin</artifactId>
        <version>${azure.functions.maven.plugin.version}</version>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-resources-plugin</artifactId>
        <version>3.1.0</version>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-dependency-plugin</artifactId>
        <version>3.1.1</version>
      </plugin>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-clean-plugin</artifactId>
        <version>3.1.0</version>
      </plugin>
    </plugins>
  </pluginManagement>

  <plugins>
    <plugin>
      <groupId>com.microsoft.azure</groupId>
      <artifactId>azure-functions-maven-plugin</artifactId>

      <configuration>
        <resourceGroup>${functionResourceGroup}</resourceGroup>
        <appName>${functionAppName}</appName>
        <region>${functionAppRegion}</region>
        <appSettings>
          <!-- Run Azure Function from package file by default -->
          <property>
            <name>WEBSITE_RUN_FROM_PACKAGE</name>
            <value>1</value>
          </property>
          <property>
            <name>FUNCTIONS_EXTENSION_VERSION</name>
            <value>~2</value>
          </property>
        </appSettings>
      </configuration>

      <executions>
        <execution>
          <id>package-functions</id>
          <goals>

```

```

        <goal>package</goal>
      </goals>
    </execution>
  </executions>
</plugin>

```

```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-resources-plugin</artifactId>
  <executions>
    <execution>
      <id>copy-resources</id>
      <phase>package</phase>
      <goals>
        <goal>copy-resources</goal>
      </goals>
      <configuration>
        <overwrite>true</overwrite>
        <outputDirectory>${stagingDirectory}</outputDirectory>
        <resources>
          <resource>
            <directory>${project.basedir}</directory>
            <includes>
              <include>host.json</include>
              <include>local.settings.json</include>
            </includes>
          </resource>
        </resources>
      </configuration>
    </execution>
  </executions>
</plugin>

```

```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <executions>
    <execution>
      <id>copy-dependencies</id>
      <phase>prepare-package</phase>
      <goals>
        <goal>copy-dependencies</goal>
      </goals>
      <configuration>
        <outputDirectory>${stagingDirectory}/lib</outputDirectory>
        <overwriteReleases>>false</overwriteReleases>
        <overwriteSnapshots>>false</overwriteSnapshots>
        <overwriteIfNewer>true</overwriteIfNewer>
        <includeScope>runtime</includeScope>
        <excludeArtifactIds>azure-functions-java-library</excludeArtifactIds>
      </configuration>
    </execution>
  </executions>
</plugin>

```

<!-- Remove [obj](#) folder generated by .NET SDK in [maven](#) clean-->

```

<plugin>
  <artifactId>maven-clean-plugin</artifactId>
  <version>3.1.0</version>
  <configuration>
    <filesets>
      <fileset>
        <directory>obj</directory>
      </fileset>
    </filesets>
  </configuration>
</plugin>
</plugins>
</build>

```

```

</project>

```

8. Running Azure function locally

- Go inside project folder and type **mvn package**
- Then **mvn azure-functions:run** to run the Azure function

```

    resolutionException
[Rajs-Air:HttpTriggerDem rajbhata$ mvn azure-functions:run
[INFO] Scanning for projects...
[INFO] -----< azure.function.rajb:httptriggerdem >-----
[INFO] Building Azure Java Functions 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- azure-functions-maven-plugin:1.4.1:run (default-cli) @ HttpTriggerDem
[INFO] 
[INFO] Azure Function App's staging directory found at: /Users/rajbhata/Desktop/Azure-learning/Http-trigger-Azure-function/HttpTriggerDem/target/azure-function-s/AzureHttpTriggerFunction
[INFO] Azure Functions Core Tools found.
```

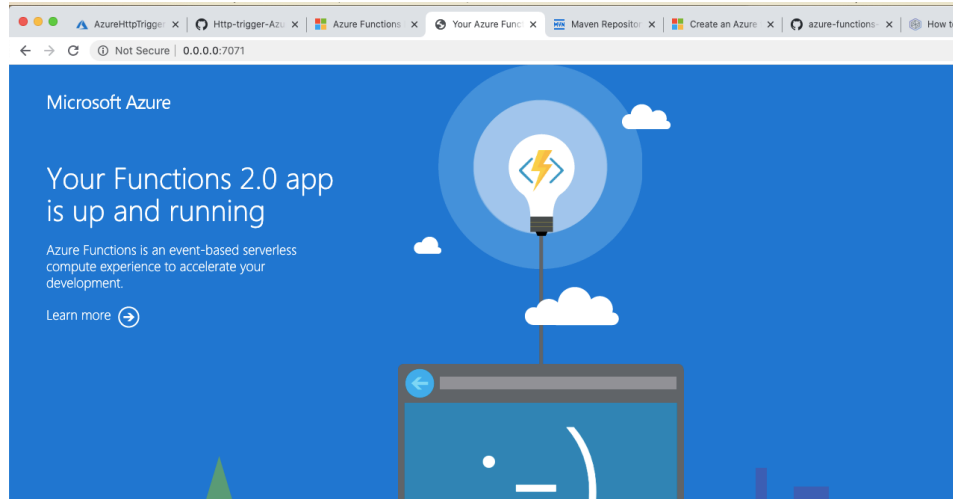
The terminal window shows the following output:

```

resolutionException
[Rajs-Air:HttpTriggerDem rajbhata$ mvn azure-functions:run
[INFO] Scanning for projects...
[INFO] -----< azure.function.rajb:httptriggerdem >-----
[INFO] Building Azure Java Functions 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- azure-functions-maven-plugin:1.4.1:run (default-cli) @ HttpTriggerDem
[INFO] 
[INFO] Azure Function App's staging directory found at: /Users/rajbhata/Desktop/Azure-learning/Http-trigger-Azure-function/HttpTriggerDem/target/azure-function-s/AzureHttpTriggerFunction
[INFO] Azure Functions Core Tools found.
```

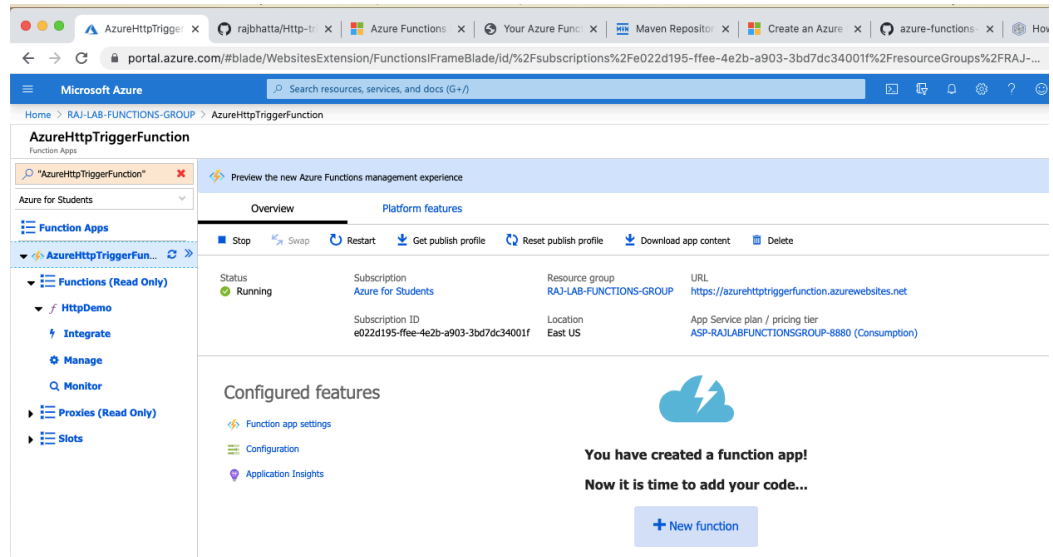
Below the terminal output, there is a decorative ASCII art logo. It features a central diamond shape composed of yellow percentage signs (%). This central shape is flanked by two vertical columns of blue at-signs (@), which are themselves flanked by more yellow percentage signs (%).

- **Goto web browser and type:** <http://0.0.0.0:7071/>



9. Deploying Azure Function

- Type Azure login to CLI and validate the login
- Then type: **mvn install package azure-functions:deploy**



Note: Http Demo is the function that we created along with above specified steps. Now, you can utilize this function to handle HttpGet and HttpPost

Note: Step to create Azure function in IntelliJ is provided here
<https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-maven-intellij>