**AWS Development Environment Setup**

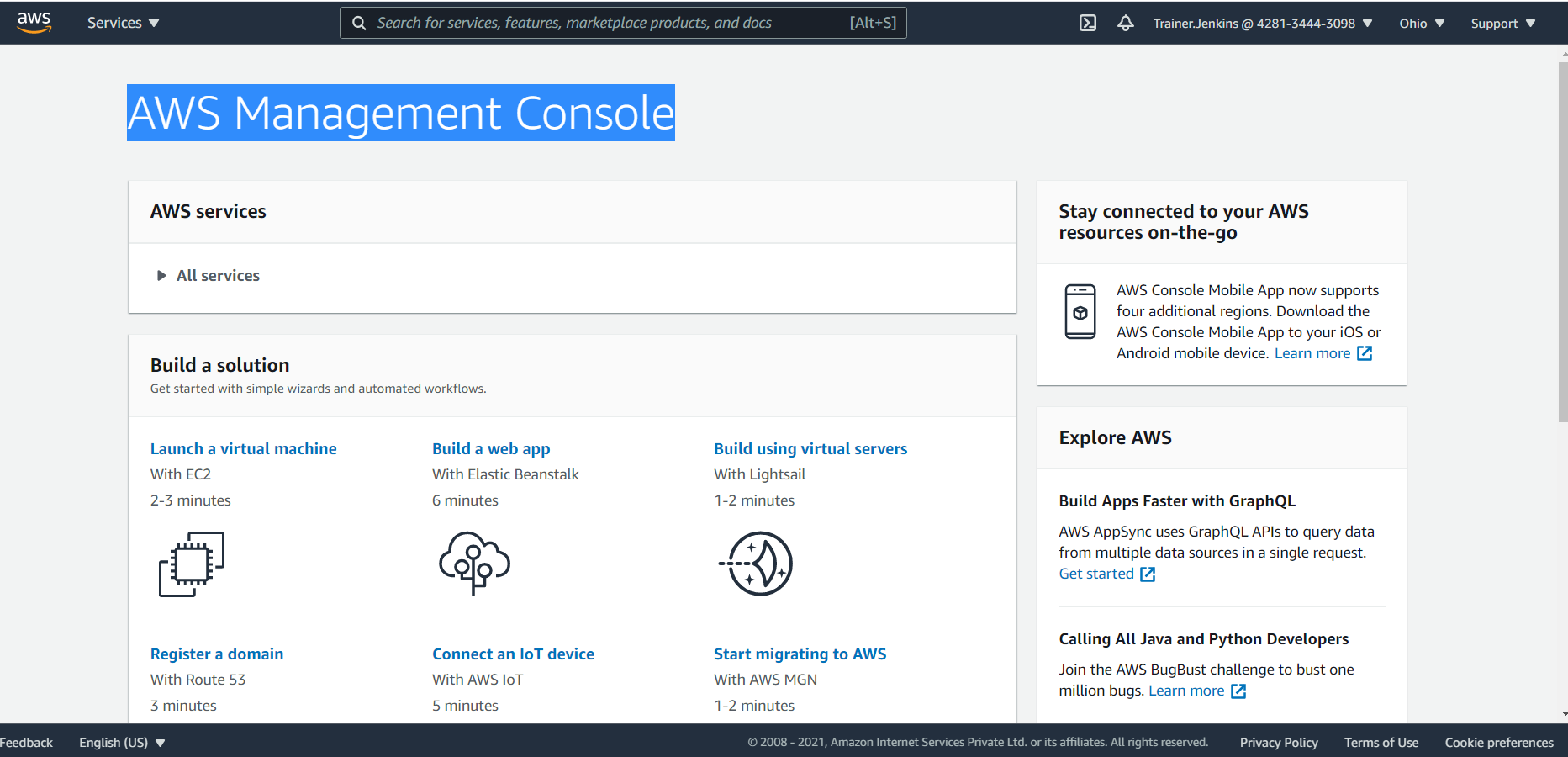
**Steps**

1. Create a Virtual Machine in the AWS Environment
2. Build SPFx Development environment setup
3. Build your first SharePoint client-side web part

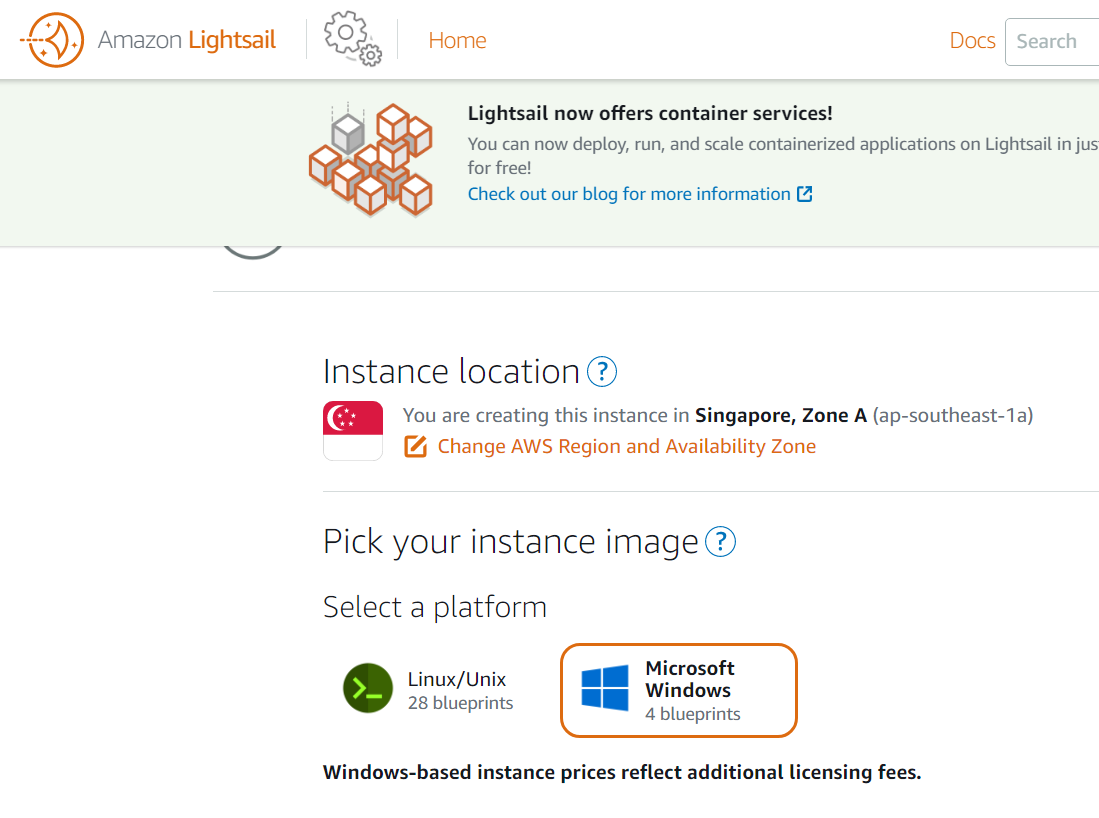
**Create a Virtual Machine in the AWS Environment**

1. Login AWS Console

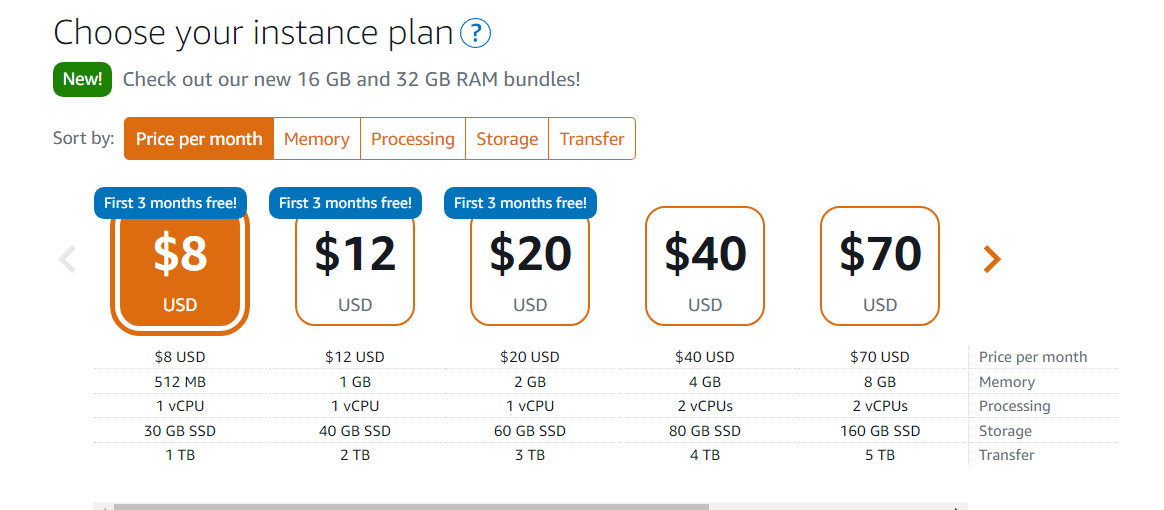
<https://428134443098.signin.aws.amazon.com/console>



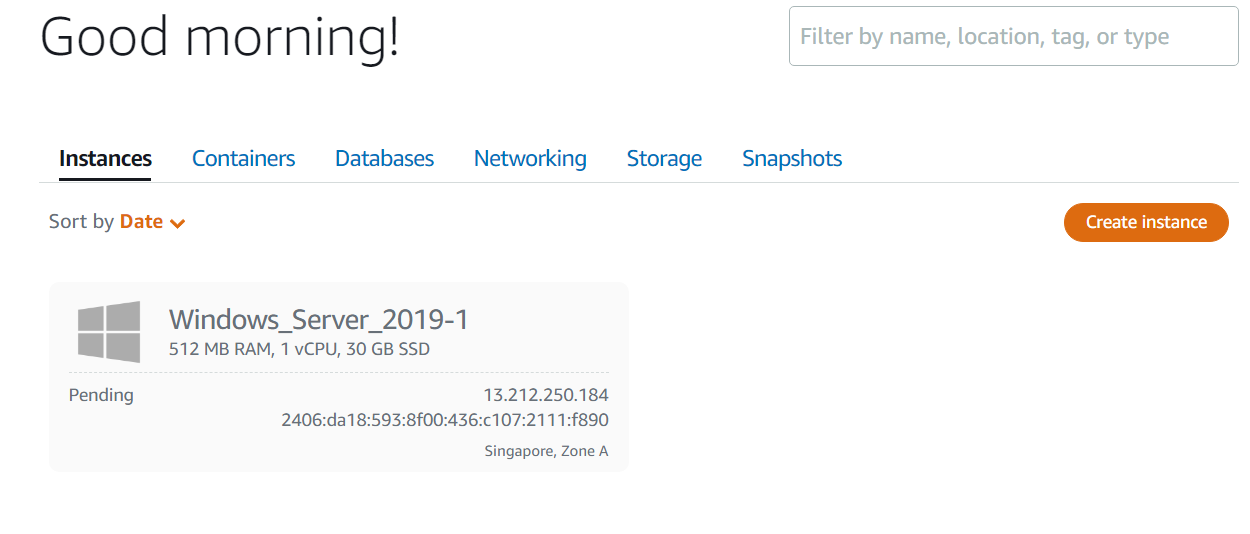
1. Click Build using virtual serves with LightSail
2. Select **Microsoft Windows**



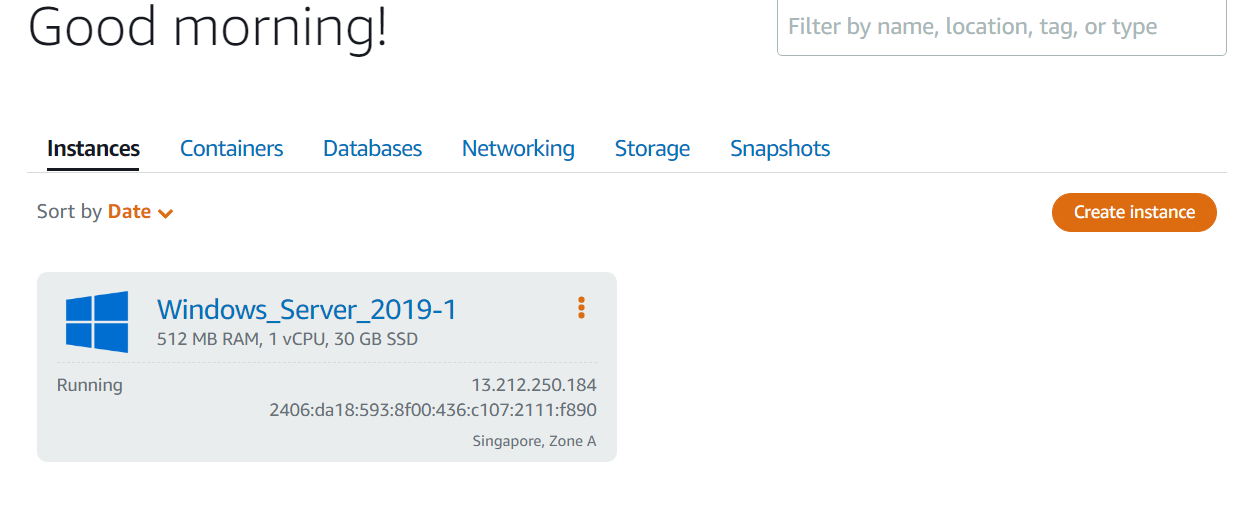
1. Select Price per month
2. Select $20 option



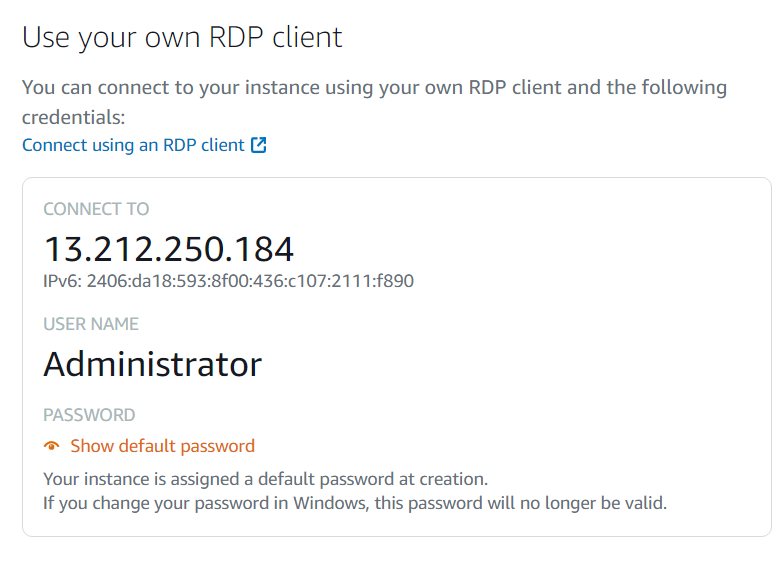
1. It is creating a VM



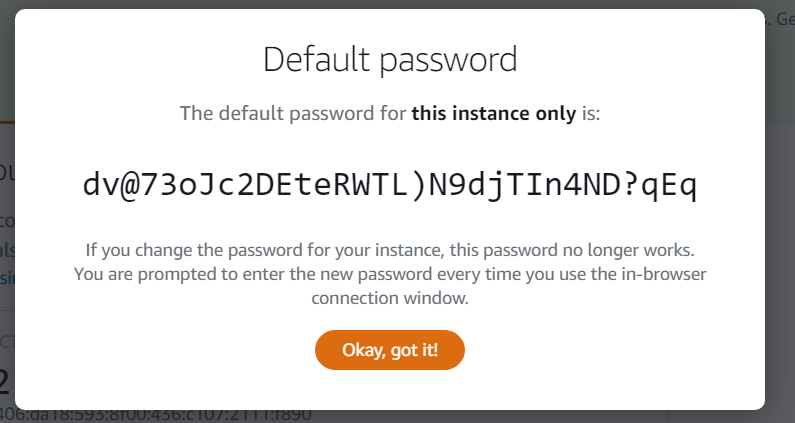
1. Running



1. Open it
2. Then Connect using an RDP client
3. In the Connect tab of the instance management page, choose Show default password.



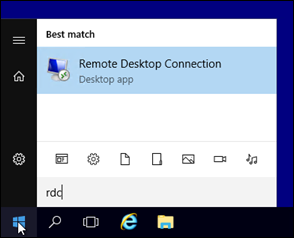
1. Highlight the default password that is displayed, and copy it by pressing **Ctrl+C or Cmd+C**. The password is now in your clipboard.



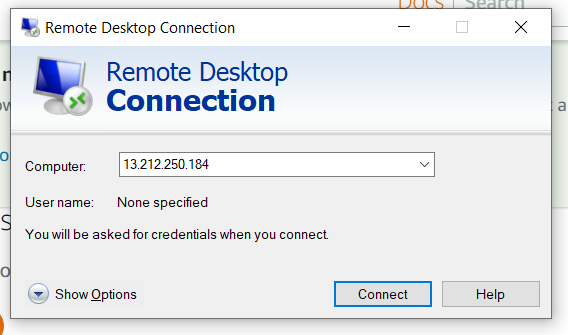
Configure RDC and connect to your Windows instance

Complete the following steps to configure RDC and connect to your Windows instance.

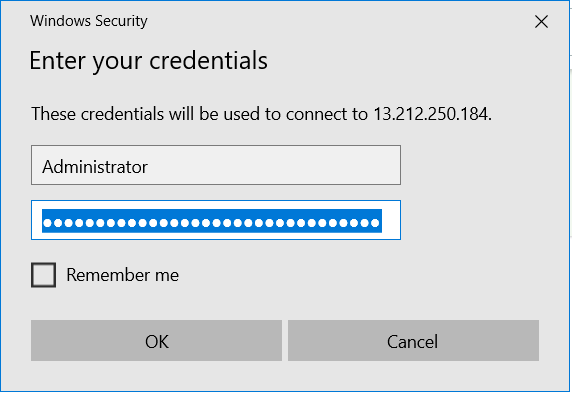
1. Open the Windows menu, and then search for Remote Desktop Connection or RDC.
2. Choose Remote Desktop Connection in the search results.



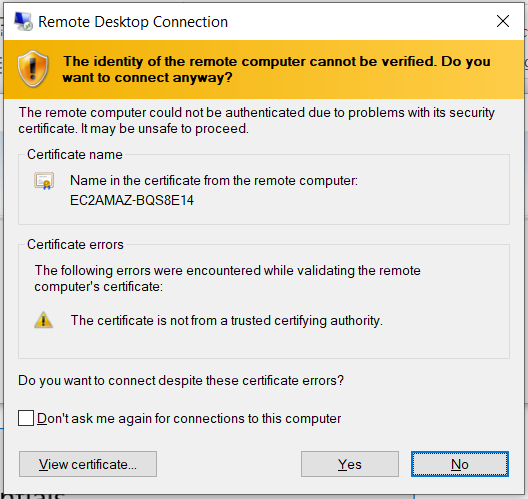
1. In the Computer text box, enter your Windows instance’s public IP address.



Click Connect and enter password

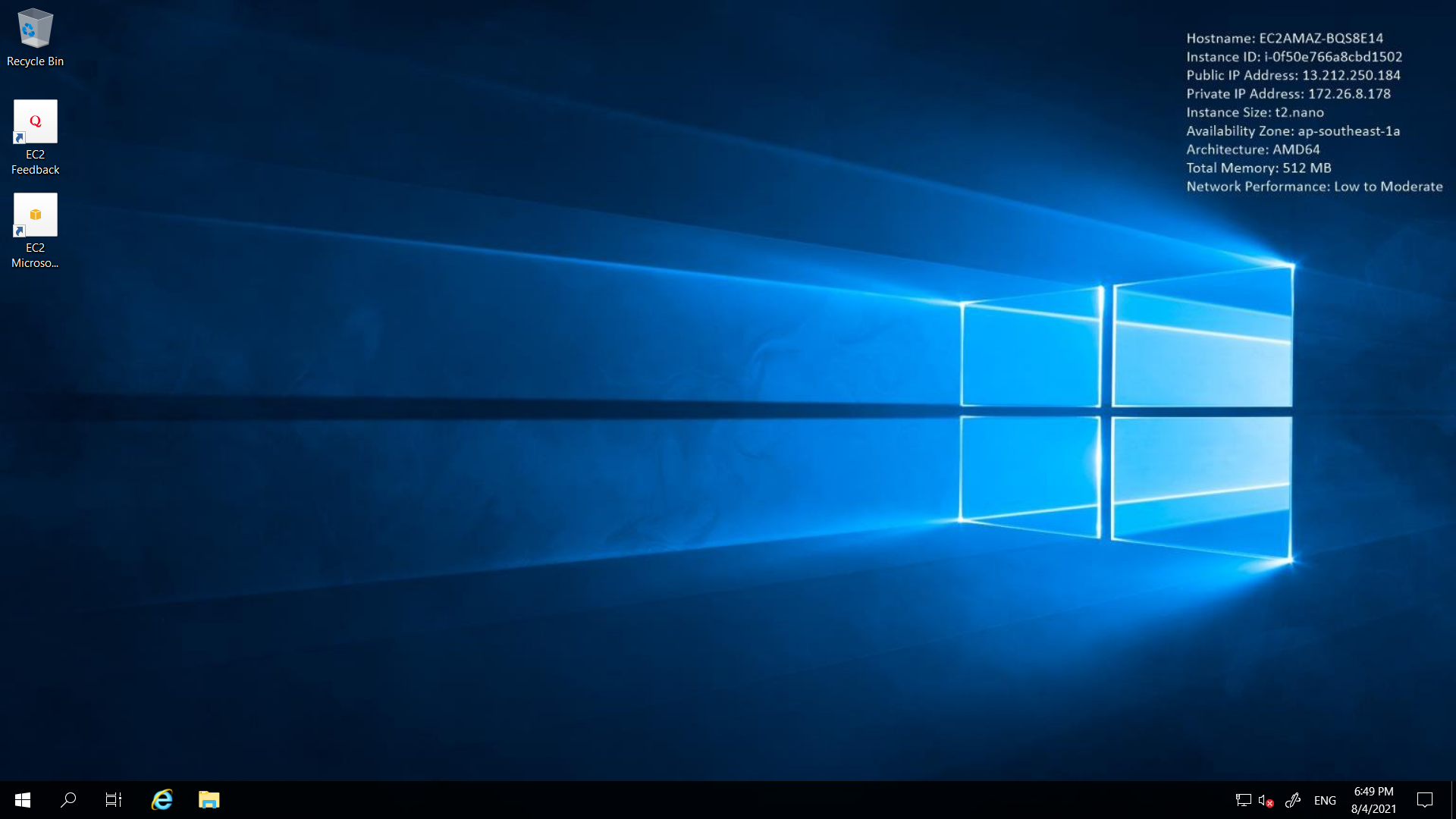


Click OK



Click Yes

It connect the Remote Desktop



**Build SPFx Development environment setup**

First Install **Brave** or **Edge Chromium** or **Google Chrome** browser

**Install developer tools**

**Install NodeJS**

Install NodeJS LTS version 14

https://nodejs.org/en/

* If you have NodeJS already installed, check that you have the correct version by using node -v.

**Install a code editor**

You can use any code editor or IDE that supports client-side development to build your web part, such as:

Visual Studio Code - <https://code.visualstudio.com/>

**Install Yeoman and gulp**

Yeoman helps you kick-start new projects and prescribes best practices and tools to help you stay productive. SharePoint client-side development tools include a Yeoman generator for creating new web parts. The generator provides common build tools, common boilerplate code, and a common playground website to host web parts for testing.

*Open Command Prompt 🡪 run as administrator*

npm install -g yo gulp

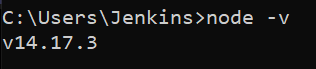
**Install Yeoman SharePoint generator**

The Yeoman SharePoint web part generator helps you quickly create a SharePoint client-side solution project with the right toolchain and project structure.

npm install -g @microsoft/generator-sharepoint

**Check version**

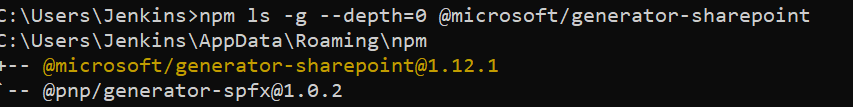
Node -v



Npm -v



npm ls -g --depth=0 @microsoft/generator-sharepoint



**Build your first SharePoint client-side web part**

Client-side web parts are client-side components that run inside the context of a SharePoint page. Client-side web parts can be deployed to SharePoint Online, and you can also use modern JavaScript tools and libraries to build them.

1. To create a new web part project

* Create a new project directory in your favourite location.
* Open command prompt
  + Run as administrator

*md Demo-webpart*

1. Go to the project directory

*cd Demo-webpart*

1. Create a new Demo web part by running the Yeoman SharePoint Generator.

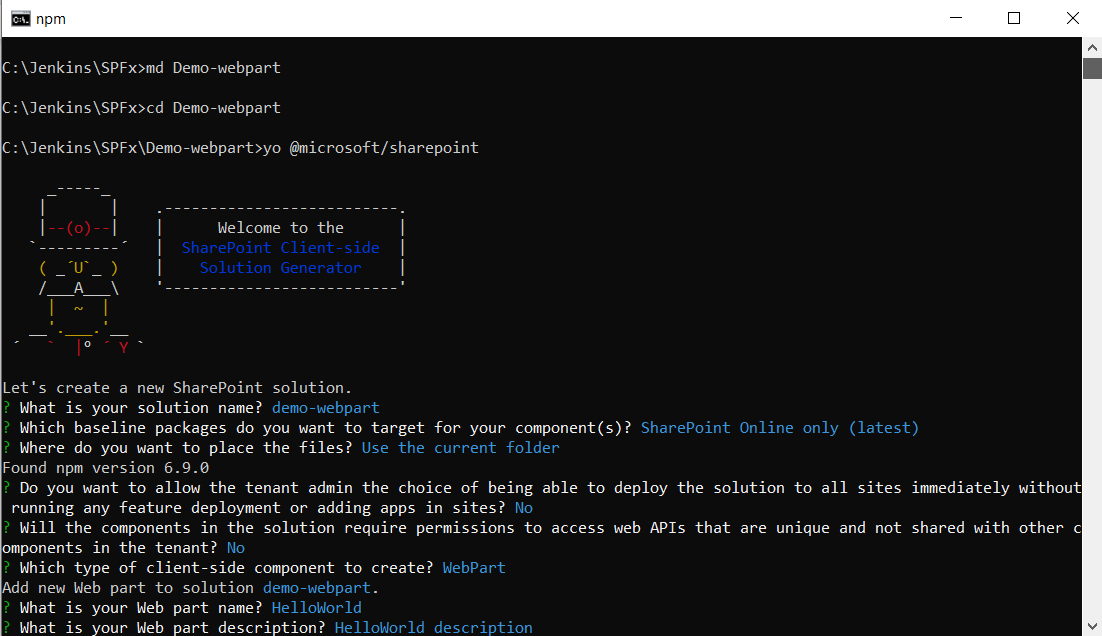
yo @microsoft/sharepoint

**When prompted**:

* Accept the default **demo-webpart** as your solution name, and then select Enter.
* Select SharePoint Online only (latest) and select Enter.
* Select Use the current folder for where to place the files.
* Select N to not allow the solution to be deployed to all sites immediately.
* Select N on the question if solution contains unique permissions.
* Select WebPart as the client-side component type to be created.

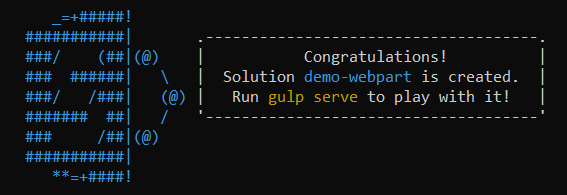
**The next set of prompts ask for specific information about your web part:**

* Accept the default HelloWorld as your web part name, and then select Enter.
* Accept the default HelloWorld description as your web part description, and then select Enter.
* Accept the default No JavaScript web framework as the framework you would like to use, and then select Enter



At this point, Yeoman installs the required dependencies and scaffolds the solution files along with the HelloWorld web part. This might take a few minutes.

When the scaffold is complete, you should see the following message indicating a successful scaffold.



**Preview the web part**

To preview your web part, build and run it on a local web server. The client-side toolchain uses HTTPS endpoint by default. This setting can be configured in the serve.json file located in the config folder, but we do recommend using the default values.

Switch to your console, ensure that you are still in the Demo-webpart directory, and then enter the following command:

gulp trust-dev-cert

**Note:**

*Developer certificate has to be installed ONLY once in your development environment, so you can skip this step, if you have already executed that in your environment.*

**gulp serve**

This command executes a series of gulp tasks to create a local, node-based HTTPS server on localhost:4321 and localhost:5432. The workbench is then launched in your default browser to preview web parts from your local dev environment.

For open visual studio code **code .** not working

got to System environment variable

Click

Environment variables

**Inside Environment variables->Path** put

C:\Users\{your\_username}\AppData\Local\Programs\Microsoft VS Code\bin.

Go to the project folder and open the cmd with it typing in the location bar and then type **code .**