

# Seminar 1.

## Launch a Linux Virtual Machine

with Amazon EC2

Amazon Elastic Compute Cloud (EC2) is the Amazon Web Service you use to create and run virtual machines in the cloud. AWS calls these virtual machines 'instances'. This step-by-step guide will help you successfully launch a Linux virtual machine on Amazon EC2 within our [AWS Free Tier](#).

---

### Step 1. Sign in/up for AWS

There are no additional charge for Amazon EC2 for this tutorial. The resources you create in this tutorial are Free Tier eligible. You have a choice either to use your classroom account or create a Free Tier account in Amazon. To create your Free Tier account follow this [link](#). During the tutorials we will be using Free Tier eligible resources and you should not be billed for your activities if you shut down your machines after each tutorial.

---

### Step 2. Launch an Amazon EC2 Instance

---

- a. Open the [AWS Management Console](#), so you can keep this step-by-step guide open. When the screen loads, enter your user name and password to get started. Then type *EC2* in the search bar and select Amazon EC2 to open the service console.

**EC2 Dashboard**

- Events
- Tags
- Reports
- Limits

**INSTANCES**

- Instances
- Spot Requests
- Reserved Instances

**IMAGES**

- AMIs
- Bundle Tasks

**ELASTIC BLOCK STORE**

- Volumes
- Snapshots

**NETWORK & SECURITY**

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

**LOAD BALANCING**

- Load Balancers

**AUTO SCALING**

- Launch Configurations
- Auto Scaling Groups

**Resources**

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

|                     |                   |
|---------------------|-------------------|
| 0 Running Instances | 0 Elastic IPs     |
| 0 Volumes           | 0 Snapshots       |
| 0 Key Pairs         | 0 Load Balancers  |
| 0 Placement Groups  | 1 Security Groups |

**Create Instance**

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

**Launch Instance**

Note: Your instances will launch in the US East (N. Virginia) region

**Service Health**

**Service Status:**

- US East (N. Virginia): This service is operating normally

**Availability Zone Status:**

- us-east-1a: Availability zone is operating normally
- us-east-1b: Availability zone is operating normally
- us-east-1c: Availability zone is operating normally
- us-east-1e: Availability zone is operating normally

[Service Health Dashboard](#)

**Scheduled Events**

**US East (N. Virginia):**  
No events

b. Select Launch Instance to create and configure your virtual machine.

**EC2 Dashboard**

- Events
- Tags
- Reports
- Limits

**INSTANCES**

- Instances
- Spot Requests
- Reserved Instances

**IMAGES**

- AMIs
- Bundle Tasks

**ELASTIC BLOCK STORE**

- Volumes
- Snapshots

**NETWORK & SECURITY**

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

**LOAD BALANCING**

- Load Balancers

**AUTO SCALING**

- Launch Configurations
- Auto Scaling Groups

**Resources**

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

|                     |                   |
|---------------------|-------------------|
| 0 Running Instances | 0 Elastic IPs     |
| 0 Volumes           | 0 Snapshots       |
| 0 Key Pairs         | 0 Load Balancers  |
| 0 Placement Groups  | 1 Security Groups |

**Create Instance**

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

**Launch Instance**

Note: Your instances will launch in the US East (N. Virginia) region

**Service Health**

**Service Status:**

- US East (N. Virginia): This service is operating normally

**Availability Zone Status:**

- us-east-1a: Availability zone is operating normally
- us-east-1b: Availability zone is operating normally
- us-east-1c: Availability zone is operating normally
- us-east-1e: Availability zone is operating normally

[Service Health Dashboard](#)

**Scheduled Events**

**US East (N. Virginia):**  
No events

## Step 3. Configure your Instance

You are now in the EC2 Launch Instance Wizard, which will help you configure and launch your instance.

- a. In this screen, you are shown options to choose an Amazon Machine Image (AMI). AMIs are preconfigured server templates you can use to launch an instance. Each AMI includes an operating system, and can also include applications and application servers.

For this tutorial, find *Amazon Linux AMI* and click *Select*.

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, and 7. Review. Below the tabs, a message states: 'An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.' A 'Quick Start' sidebar on the left lists 'My AMIs', 'AWS Marketplace', and 'Community AMIs'. The main area displays a list of AMIs:

- Amazon Linux AMI 2015.03.1 (HVM), SSD Volume Type - ami-0d4cf0b6**  
Free tier eligible  
Amazon Linux  
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.  
Root device type: ebs Virtualization type: hvm  
**Select** 64-bit
- Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-12663b7a**  
Free tier eligible  
Red Hat  
Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type  
Root device type: ebs Virtualization type: hvm  
**Select** 64-bit
- SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-aeb532c6**  
Free tier eligible  
SUSE Linux  
SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.  
Root device type: ebs Virtualization type: hvm  
**Select** 64-bit
- Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-d05e75b8**  
Free tier eligible  
Ubuntu  
Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
Root device type: ebs Virtualization type: hvm  
**Select** 64-bit
- Microsoft Windows Server 2012 R2 Base - ami-cd9339a6**  
Free tier eligible  
Windows  
Microsoft Windows Server 2012 R2 Standard edition with 64-bit architecture. [English]  
Root device type: ebs Virtualization type: hvm  
**Select** 64-bit
- Amazon RDS**  
Are you launching a database instance? Try Amazon RDS.  
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database of your choice (MySQL, PostgreSQL, Oracle, SQL Server) in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database management tasks, freeing you up to focus on your applications and business. [Learn more](#).  
**Launch a database using RDS**  
**Select** 64-bit
- Microsoft Windows Server 2012 R2 with SQL Server Express - ami-8359f1e8**  
Free tier eligible  
Windows  
Microsoft Windows Server 2012 R2 Standard edition, 64-bit architecture, Microsoft SQL Server 2014 Express edition, [English]  
Root device type: ebs Virtualization type: hvm  
**Select** 64-bit

- b. You will now choose an instance type. Instance types comprise of varying combinations of CPU, memory, storage, and networking capacity so you can choose the appropriate mix for your applications. For more information, see [Amazon EC2 Instance Types](#).

The default option of *t2.micro* should already be checked. This instance type is covered within the Free Tier and offers enough compute capacity to tackle simple workloads. Click *Review and Launch* at the bottom of the page.

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

| Filter by:  | All instance types                            | Current generation | Show/Hide Columns |
|---|---|--------------------|-------------------|
| Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only) |   |                    |                   |
| Family  | Type  | vCPUs              | Memory (GiB)      |
| General purpose   | t2.micro<br><small>Free tier eligible</small> | 1                  | 1                 |
| General purpose   | t2.small                                      | 1                  | 2                 |
| General purpose   | t2.medium                                     | 2                  | 4                 |
| General purpose   | t2.large                                      | 2                  | 8                 |
| General purpose   | m4.large                                      | 2                  | 8                 |
| General purpose   | m4.xlarge                                     | 4                  | 16                |
| General purpose   | m4.2xlarge                                    | 8                  | 32                |
| General purpose   | m4.4xlarge                                    | 16                 | 64                |
| General purpose   | m4.10xlarge                                   | 40                 | 160               |
| General purpose   | m3.medium                                     | 1                  | 3.75              |
| General purpose   | m3.large                                      | 2                  | 7.5               |
| General purpose   | m3.xlarge                                     | 4                  | 15                |
| General purpose   | m3.2xlarge                                    | 8                  | 30                |
| EBS-Optimized Available   |   |                    |                   |
| Network Performance   |   |                    |                   |
| Low to Moderate   |   |                    |                   |
| Moderate  |   |                    |                   |
| High  |   |                    |                   |
| 10 Gigabit  |   |                    |                   |
| Moderate  |   |                    |                   |
| Moderate  |   |                    |                   |
| High  |   |                    |                   |
| High  |   |                    |                   |
| High  |   |                    |                   |
| Cancel Previous Review and Launch Next: Configure Instance Details  |   |                    |                   |

( click to enlarge )

- c. You can review the configuration, storage, tagging, and security settings that have been selected for your instance. While you have the option to customize these settings, we recommend accepting the default values for this tutorial.

Click Launch at the bottom of the page.

## Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.

**⚠ Improve your instances' security. Your security group, launch-wizard-1, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**▼ AMI Details**

|   |  |          |
|---|--|----------|
| Free tier eligible  | Amazon Linux AMI 2015.03.1 (HVM), SSD Volume Type - ami-0d4cfdb6 | Edit AMI |
| The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages. |  |          |
| Root Device Type: ebs   | Virtualization type: hvm   |          |

**▼ Instance Type**

| Instance Type | ECUs     | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|----------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro      | Variable | 1     | 1            | EBS only              | -                       | Low to Moderate     |

**▼ Security Groups**

|                     |   |                      |           |
|---------------------|---|----------------------|-----------|
| Security group name | launch-wizard-1                                       | Edit security groups |           |
| Description         | launch-wizard-1 created 2015-09-11T13:35:57.266-07:00 |                      |           |
| Type                | Protocol  | Port Range           | Source    |
| SSH                 | TCP   | 22                   | 0.0.0.0/0 |

**► Instance Details**

- Storage Edit storage
- Tags Edit tags

Cancel Previous Launch

- d. On the next screen you will be asked to choose an existing key pair or create a new key pair. A key pair is used to securely access your Linux instance using SSH. AWS stores the public part of the

key pair which is just like a house lock. You download and use the private part of the key pair which is just like a house key.

Select Create a new key pair and give it the name MyKeyPair. Next click the Download Key Pair button.

After you download the MyKeyPair key, you will want to store your key in a secure location. If you lose your key, you won't be able to access your instance. If someone else gets access to your key, they will be able to access your instance.

Windows users: We recommend saving your key pair in your user directory in a sub-directory called .ssh (ex. C:\user\{yourusername}\.ssh\MyKeyPair.pem).

Tip: You can't use Windows Explorer to create a folder with a name that begins with a period unless you also end the folder name with a period. After you enter the name (.ssh.), the final period is removed automatically.

Mac/Linux users: We recommend saving your key pair in the .ssh sub-directory from your home directory (ex. ~/.ssh/MyKeyPair.pem).

Tip: On MacOS, the key pair is downloaded to your Downloads directory by default. To move the key pair into the .ssh sub-directory, enter the following command in a terminal window: mv ~/Downloads/MyKeyPair.pem ~/.ssh/MyKeyPair.pem

After you have stored your key pair, click Launch Instance to start your Linux instance.

## Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

MyKeyPair

Download Key Pair



You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel

Launch Instances

e. Click View Instances on the next screen to view your instances and see the status of the instance you have just started.

Launch Status

✓ Your instances are now launching

The following instance launches have been initiated: I- [REDACTED] [View launch log](#)

ⓘ Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

▼ Here are some helpful resources to get you started

- How to connect to your Linux instance
- Amazon EC2 User Guide
- Learn about AWS Free Usage Tier
- Amazon EC2: Discussion Forum

While your instances are launching you can also

Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)

Create and attach additional EBS volumes (Additional charges may apply)

Manage security groups

[View Instances](#)

f. In a few minutes, the *Instance State* column on your instance will change to "running" and a Public IP address will be shown. You can refresh these Instance State columns by pressing the refresh button on the right just above the table. Copy the Public IP address of your AWS instance, so you can use it when we connect to the instance using SSH in Step 4.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Commands, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, and Placement Groups. The main area shows a table of instances. One row is selected, showing details for an instance with the following information:

| Instance ID | Name       | Instance Type | Availability Zone | Instance State | Status Checks  | Alarm Status | Public DNS |
|-------------|------------|---------------|-------------------|----------------|----------------|--------------|------------|
| [REDACTED]  | [REDACTED] | t2.micro      | us-east-1a        | running        | 2/2 checks ... | None         | [REDACTED] |

Below the table, the instance details are expanded:

| Instance ID | Public DNS                  |
|-------------|-----------------------------|
| [REDACTED]  | Public IP: 52. [REDACTED] 5 |

Other visible details include:

- Description:** Instance ID: [REDACTED], Instance state: running, Instance type: t2.micro, Private DNS: ip-[REDACTED].ec2.internal, Private IPs: [REDACTED], Secondary private IPs: [REDACTED], VPC ID: vpc-434f9a27.
- Public DNS:** Public IP: 52. [REDACTED] 5, Elastic IP: [REDACTED].
- Availability zone:** us-east-1a.
- Security groups:** launch-wizard-4, view rules.
- Scheduled events:** No scheduled events.
- AMI ID:** amzn-ami-hvm-2015.09.1.x86\_64-gp2.

## Step 4. Connect to your Instance

After launching your instance, it's time to connect to it using SSH.

Windows users: Select Windows below to see instructions for installing Git Bash which includes SSH.

Mac/Linux user: Select Mac / Linux below to see instructions for opening a terminal window.

- a. Download Git for Windows [here](#). Run the downloaded installer accepting the default settings (this will install Git Bash as part of Git).



git --distributed-is-the-new-centralized



Search entire site...

About

Documentation

Blog

Downloads

GUI Clients

Logos

Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloading Git



Your download is starting...

You are downloading the latest (2.6.4) 64-bit version of **Git for Windows**. This is the most recent [maintained build](#). It was released **2 days ago**, on 2015-12-14.

If your download hasn't started, [click here to download manually](#).

### Other Git for Windows downloads

[Git for Windows Setup](#)

[32-bit Git for Windows Setup](#).

[64-bit Git for Windows Setup](#).

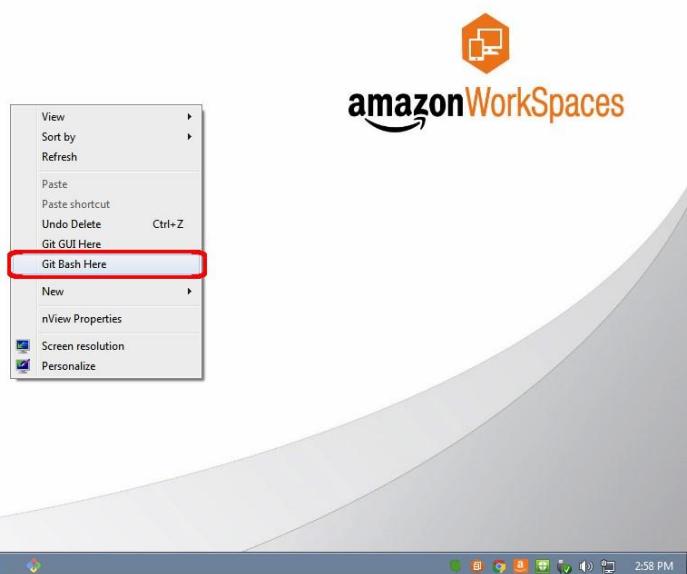
[Git for Windows Portable \("thumbdrive edition"\)](#)

[32-bit Git for Windows Portable](#).

[64-bit Git for Windows Portable](#).

The current source code release is version 2.6.4. If you want the newer version, you can build it from [the source code](#).

- b. Right click on your desktop (not on an icon or file) and select Git Bash Here to open a Git Bash command prompt.



c. Use SSH to connect to your instance. In this case the user name is ec2-user, the SSH key is stored in the directory we saved it to in step 3 part d, and the IP address is from step 3 part f. The format is:

```
ssh -i {full path of your .pem file} ec2-user@{instance IP address}
```

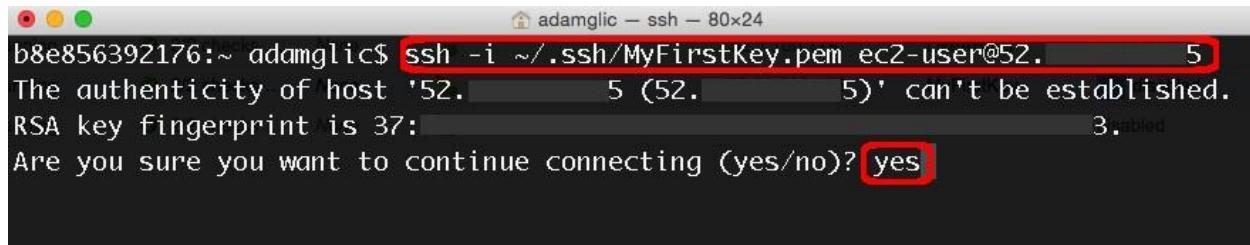
Enter the following:

```
ssh -i 'c:\Users\yourusername\.ssh\MyKeyPair.pem' ec2-user@{IP_Address}
```

Example: ssh -i 'c:\Users\adamglic\.ssh\MyKeyPair.pem' ec2-user@52.27.212.125 You'll see a response similar to the following:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established. RSA key fingerprint is 1f:51:ae:28:df:63:e9:d8:cf:38:5d:87:2d:7b:b8:ca:9f:f5:b1:6f. Are you sure you want to continue connecting (yes/no)?
```

Type yes and press enter.



```
adamglic - ssh - 80x24
b8e856392176:~ adamglic$ ssh -i ~/.ssh/MyFirstKey.pem ec2-user@52.27.212.125
The authenticity of host '52.27.212.125' can't be established.
RSA key fingerprint is 37:3e:2b:4f:5c:6d:7e:8f:90:ab:c2:de:ef:12:34:56.
Are you sure you want to continue connecting (yes/no)? yes
```

---

d. You'll

see a response similar to the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA) to the list of known hosts.
```

You should then see the welcome screen for your instance and you are now connected to your AWS Linux virtual machine in the cloud.

```

adamglic ~ ec2-user@ip-172-31-52-176:~ ssh -i ~/.ssh/MyFirstKey.pem ec2-user@52.172.31.5
The authenticity of host '52.172.31.5 (52.172.31.5)' can't be established.
RSA key fingerprint is 37:0d:93:1f:2e:4b:3c:1d:5b:4a:1c:3a:2b:13:ab:ed.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.172.31.5' (RSA) to the list of known hosts.

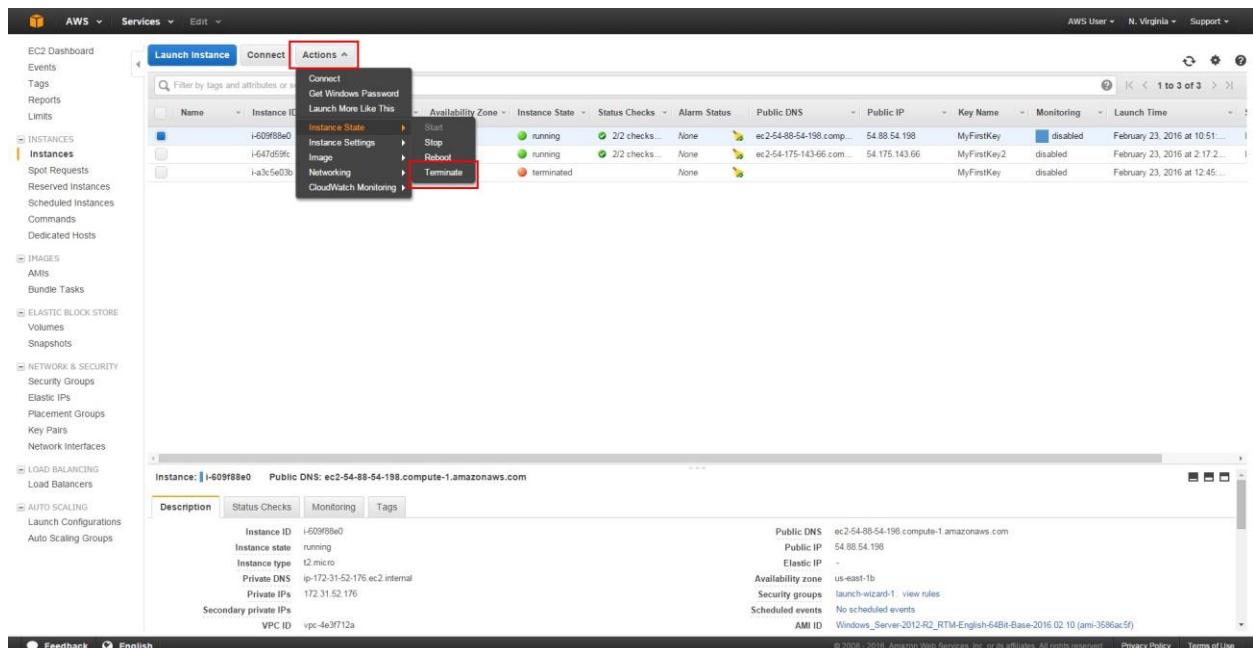
              _|_ _|_
              | | /   Amazon Linux AMI
              ___\_\_|_|_|
https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
11 package(s) needed for security, out of 27 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-52-176 ~]$ 

```

## Step 5. Terminate your Instance

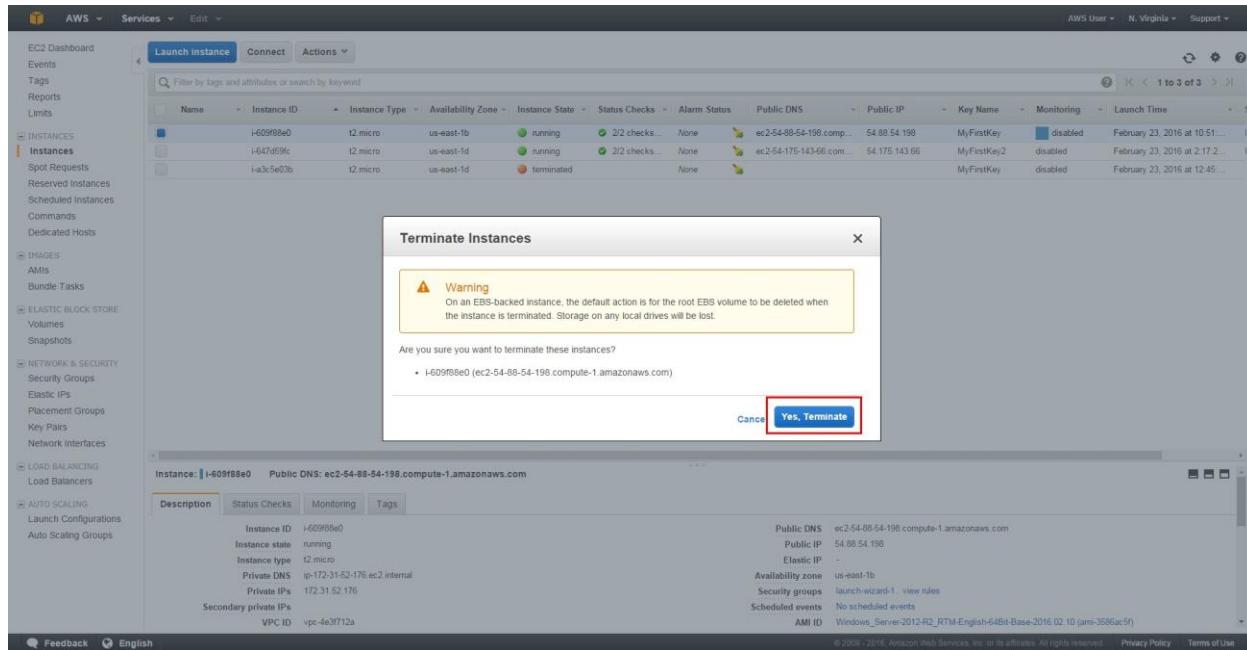
You can easily terminate the instance from the EC2 console. In fact, it is a best practice to terminate instances you are no longer using so you don't keep getting charged for them.

- Back on the EC2 Console, select the box next to the instance you created. Then click the Actions button, navigate to *Instance State*, and click Terminate.



b. You will be asked to confirm your termination - select Yes, Terminate.

Note: This process can take several seconds to complete. Once your instance has been terminated, the Instance State will change to *terminated* on your EC2 Console.



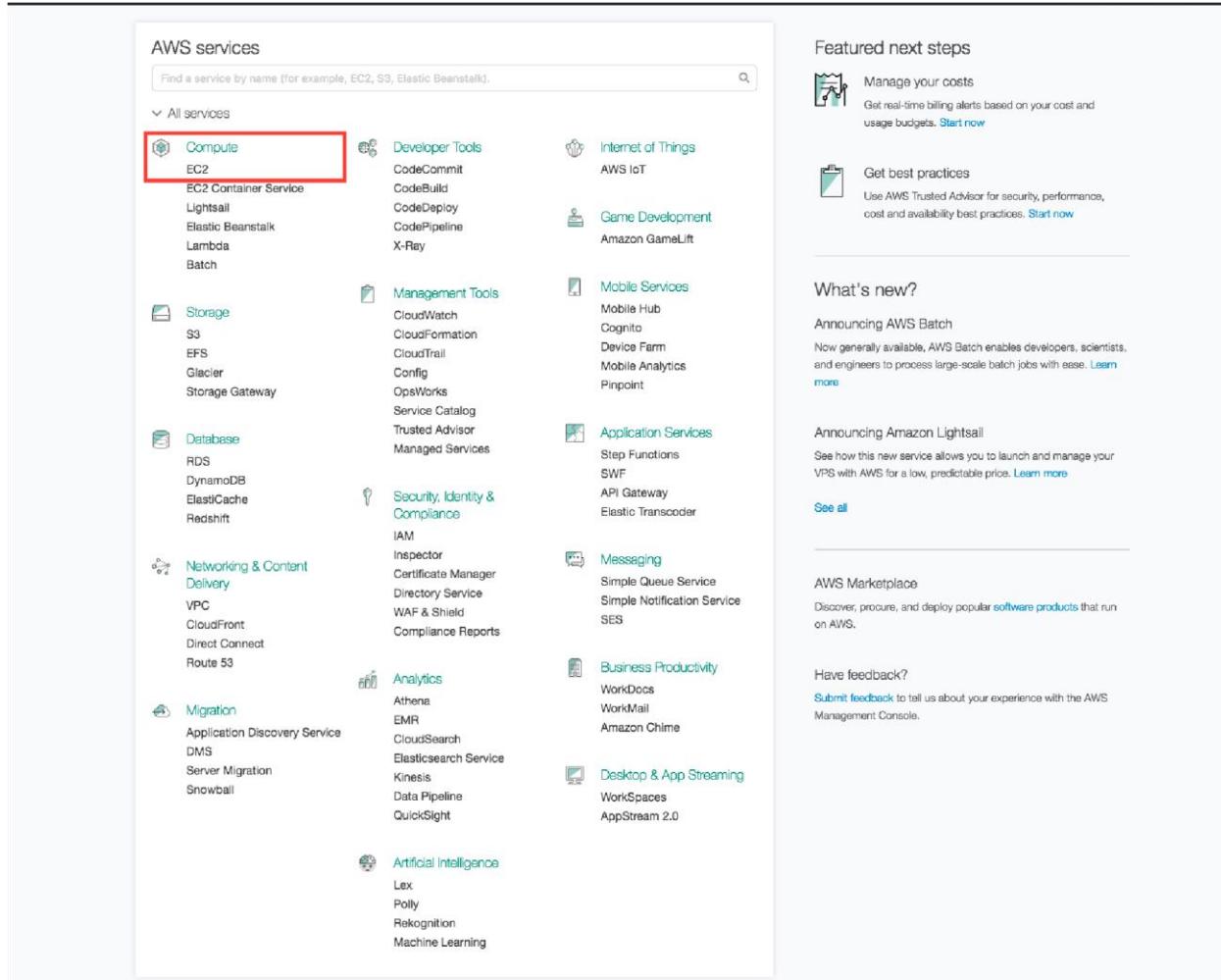
You have learned how to use Amazon EC2 to launch, configure, connect, and terminate an instance in the cloud.

## Launch a Windows Virtual Machine with Amazon EC2

In this task, you will learn how to launch, configure, and connect to a Windows Virtual Machine using Amazon Elastic Compute Cloud (EC2).

# Step 1: Enter the EC2 Dashboard

When you [click here](#), the AWS management console will open in a new browser tab, so you can keep this tutorial open. Find EC2 under *Compute* and click to open the Amazon EC2 Console.



The screenshot shows the AWS Management Console homepage. On the left, there's a sidebar titled "AWS services" with a search bar at the top. Below the search bar is a "All services" dropdown. Under "Compute", the "EC2" service is highlighted with a red box. Other services listed under Compute include EC2 Container Service, Lightsail, Elastic Beanstalk, Lambda, and Batch. To the right of the sidebar, there are several sections: "Developer Tools" (CodeCommit, CodeBuild, CodeDeploy, CodePipeline, X-Ray), "Internet of Things" (AWS IoT), "Game Development" (Amazon GameLift), "Management Tools" (CloudWatch, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog), "Mobile Services" (Mobile Hub, Cognito, Device Farm, Mobile Analytics, Pinpoint), "Database" (RDS, DynamoDB, ElastiCache, Redshift), "Application Services" (Step Functions, SWF, API Gateway, Elastic Transcoder), "Networking & Content Delivery" (VPC, CloudFront, Direct Connect, Route 53), "Messaging" (Simple Queue Service, Simple Notification Service, SES), "Analytics" (Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, Data Pipeline, QuickSight), "Business Productivity" (WorkDocs, WorkMail, Amazon Chime), "Migration" (Application Discovery Service, DMS, Server Migration, Snowball), and "Artificial Intelligence" (Lex, Polly, Rekognition, Machine Learning). On the right side of the page, there's a "Featured next steps" section with links to "Manage your costs", "Get best practices", and "What's new?". The "What's new?" section includes announcements for AWS Batch and Amazon Lightsail, along with a link to "See all". There's also a "Have feedback?" section with a link to "Submit feedback".

# Step 2: Create and Configure Your Virtual Machine

a. You are now in the Amazon EC2 console. Click Launch Instance

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area has three main sections: 'Resources' (listing 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 0 Key Pairs, 0 Placement Groups, 0 Elastic IPs, 0 Snapshots, 0 Load Balancers, and 3 Security Groups), 'Account Attributes' (listing Supported Platforms [VPC], Default VPC [vpc-4e3f712a], and Resource ID length management), and 'Additional Information' (links to Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us). Below these is the 'AWS Marketplace' section. The central part of the dashboard features 'Create Instance' instructions and a prominent 'Launch Instance' button, which is highlighted with a red box. There are also sections for 'Service Health' (showing Service Status for US East (N. Virginia) and Availability Zone Status for us-east-1b, us-east-1c, us-east-1d, and us-east-1e, all marked as operating normally) and 'Scheduled Events' (showing 'No events'). At the bottom, there are links for Feedback, English, and a footer with copyright information and links to Privacy Policy and Terms of Use.

- b. With Amazon EC2, you can specify the software and specifications of the instance you want to use. In this screen, you are shown options to choose an Amazon Machine Image (AMI), which is a template that contains the software configuration required to launch your instance.

For this tutorial, find *Microsoft Windows Server 2012 R2 Base* and click Select.

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Tag Instance   6. Configure Security Group   7. Review   [Cancel and Exit](#)

**Step 1: Choose an Amazon Machine Image (AMI)**  
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

| Quick Start   | AMIs   | Cancel and Exit            |
|---|--|----------------------------|
| <input type="checkbox"/> My AMIs  | <a href="#">Amazon Linux AMI 2015.09.2 (HVM), SSD Volume Type - ami-8fceef5</a> <a href="#">Select</a><br>Amazon Linux The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.<br>Root device type: ebs Virtualization type: hvm | < < 1 to 22 of 22 AMIs > > |
| <input type="checkbox"/> AWS Marketplace  | <a href="#">Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-2051294</a> <a href="#">Select</a><br>Red Hat Red Hat Enterprise Linux version 7.2 (HVM). EBS General Purpose (SSD) Volume Type<br>Root device type: ebs Virtualization type: hvm  | 64-bit                     |
| <input type="checkbox"/> Community AMIs   | <a href="#">SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type - ami-b7b4fedd</a> <a href="#">Select</a><br>SUSE Linux SUSE Linux Enterprise Server 12 Service Pack 1 (HVM). EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.<br>Root device type: ebs Virtualization type: hvm                            | 64-bit                     |
| <input type="checkbox"/> Free tier only ⓘ   | <a href="#">Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-fce3c696</a> <a href="#">Select</a><br>Ubuntu Ubuntu Server 14.04 LTS (HVM). EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).<br>Root device type: ebs Virtualization type: hvm                               | 64-bit                     |
|   | <a href="#">Microsoft Windows Server 2012 R2 Base - ami-3588ac5f</a> <a href="#">Select</a><br>Windows Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]<br>Root device type: ebs Virtualization type: hvm  | 64-bit                     |
| <a href="#">Are you launching a database instance? Try Amazon RDS.</a> <a href="#">Hide</a> |  |                            |

c. You will now choose an instance type. Instance types comprise of varying combinations of CPU, memory, storage, and networking capacity so you can choose the appropriate mix for your applications. For more information, see [Amazon EC2 Instance Types](#).

For this tutorial, select the default option of *t2.micro* - this instance type is covered within the free tier. Then click Review and Launch at the bottom of the page.

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Tag Instance   6. Configure Security Group   7. Review

**Step 2: Choose an Instance Type**

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2 micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

|                                     | Family          | Type                           | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|-------------------------------------|-----------------|--------------------------------|-------|--------------|-----------------------|-------------------------|---------------------|
| <input type="checkbox"/>            | General purpose | t2.nano                        | 1     | 0.5          | EBS only              | -                       | Low to Moderate     |
| <input checked="" type="checkbox"/> | General purpose | t2.micro<br>Free tier eligible | 1     | 1            | EBS only              | -                       | Low to Moderate     |
| <input type="checkbox"/>            | General purpose | t2.small                       | 1     | 2            | EBS only              | -                       | Low to Moderate     |
| <input type="checkbox"/>            | General purpose | t2.medium                      | 2     | 4            | EBS only              | -                       | Low to Moderate     |
| <input type="checkbox"/>            | General purpose | t2.large                       | 2     | 8            | EBS only              | -                       | Low to Moderate     |
| <input type="checkbox"/>            | General purpose | m4.large                       | 2     | 8            | EBS only              | Yes                     | Moderate            |
| <input type="checkbox"/>            | General purpose | m4.xlarge                      | 4     | 16           | EBS only              | Yes                     | High                |
| <input type="checkbox"/>            | General purpose | m4.2xlarge                     | 8     | 32           | EBS only              | Yes                     | High                |
| <input type="checkbox"/>            | General purpose | m4.4xlarge                     | 16    | 64           | EBS only              | Yes                     | High                |
| <input type="checkbox"/>            | General purpose | m4.10xlarge                    | 40    | 160          | EBS only              | Yes                     | 10 Gigabit          |
| <input type="checkbox"/>            | General purpose | m3.medium                      | 1     | 3.75         | 1 x 4 (SSD)           | -                       | Moderate            |
| <input type="checkbox"/>            | General purpose | m3.large                       | 2     | 7.5          | 1 x 32 (SSD)          | -                       | Moderate            |

Cancel Previous **Review and Launch** Next: Configure Instance Details

Feedback English | © 2008–2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. | [Privacy Policy](#) | [Terms of Use](#)

- d. You can review the options that are selected for your instance which include AMI Details, Instance Type, Security Groups, Instance Details, Storage, and Tags. You can leave these at the defaults and click Launch from the bottom of the page.

Note: For detailed information on your options, see [Launching an Instance](#)

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Tag Instance   6. Configure Security Group   7. Review

**Step 7: Review Instance Launch**

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**AMI Details**

**Microsoft Windows Server 2012 R2 Base - ami-3586ac5f**

**Free tier eligible** Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]

**Instance Type**

| Instance Type | ECUs     | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|----------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro      | Variable | 1     | 1            | EBS only              | -                       | Low to Moderate     |

**Security Groups**

**Security group name** launch-wizard-1  
**Description** launch-wizard-1 created 2016-02-23T10:48:41.982-08:00

| Type | Protocol | Port Range | Source    |
|------|----------|------------|-----------|
| RDP  | TCP      | 3389       | 0.0.0.0/0 |

**Instance Details**

**Storage**

**Cancel** **Previous** **Launch**

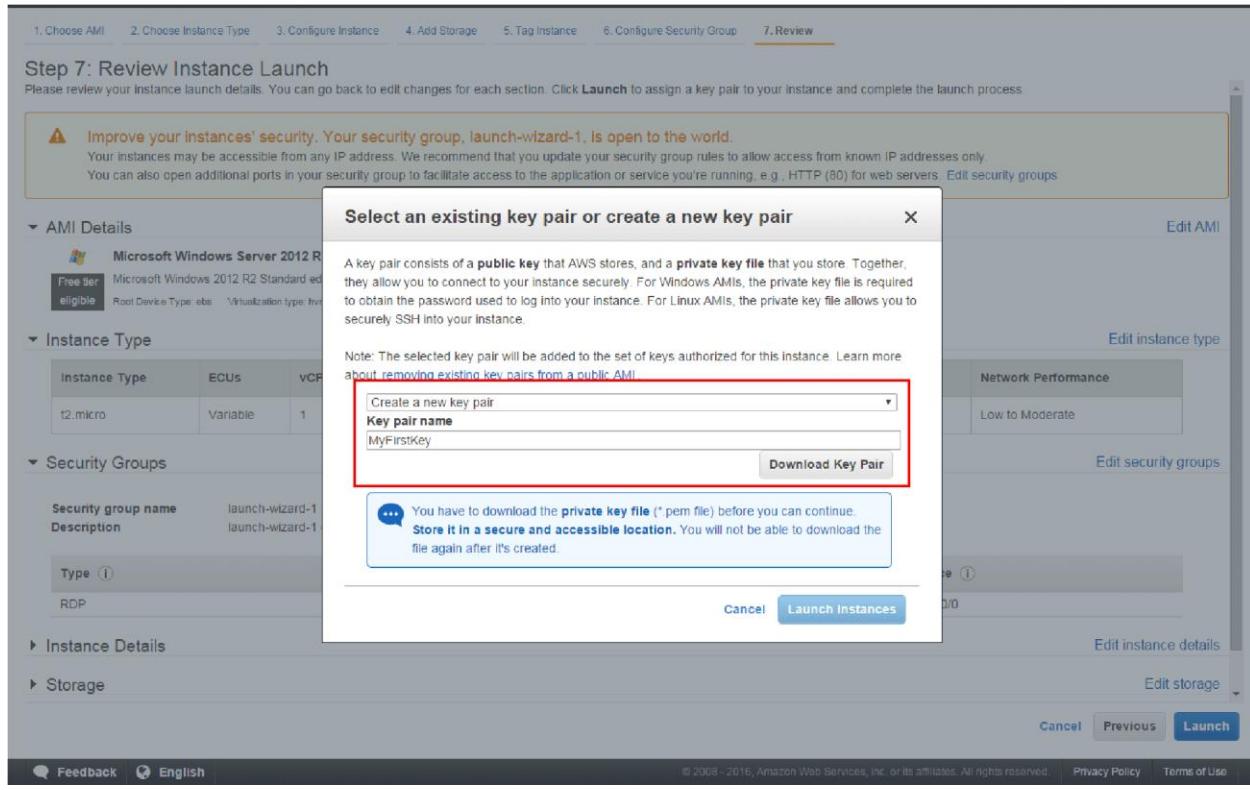
## Step 3: Create a Key Pair and Launch Your Instance

To connect to your virtual machine, you need a key pair. A key pair is used to log into your instance (just like your house key is used to enter your home).

a. In the popover, select *Create a new key pair* and name it MyFirstKey. Then click Download Key Pair. MyFirstKey.pem will be downloaded to your computer -- make sure to save this key pair in a safe location on your computer.

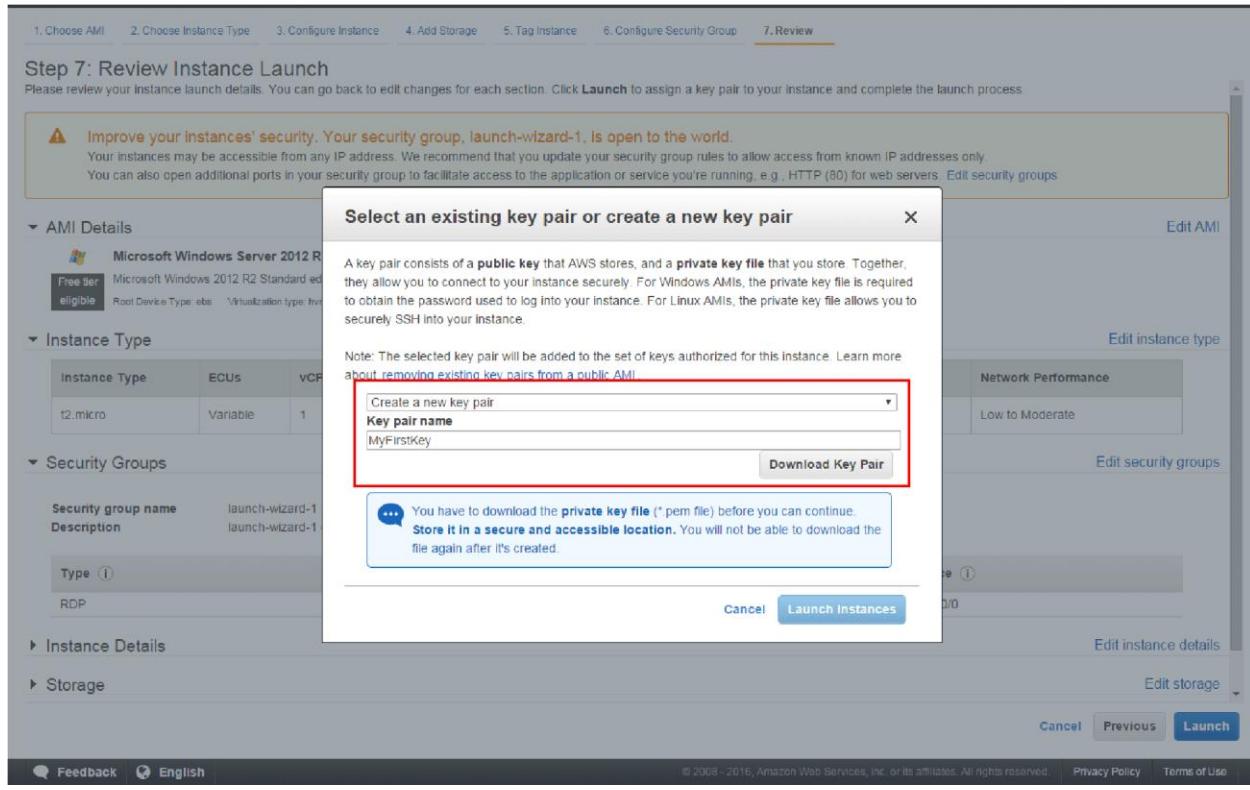
- Windows users: We recommend saving your key pair in your user directory in a sub-directory called .ssh (ex.C:\user\{yourusername}\.ssh\MyFirstKey.pem).
- Mac/Linux users: We recommend saving your key pair in the .ssh subdirectory from your home directory (ex.~/.ssh/MyFirstKey.pem).

Note: If you don't remember where you store your SSH private key (the file you are downloading), you won't be able to connect to your virtual machine.



- b. After you have downloaded and saved your key pair, click Launch Instance to start your Windows Server instance.

Note: It can take a few minutes to launch your instance.



c. On the next screen, click View Instances to view the instance you have just created and see its status.

---

## Launch Status

>Your instances are now launching

The following instance launches have been initiated: i-609f88e0 [View launch log](#)

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

### How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. Find out how to connect to your instances.

▼ Here are some helpful resources to get you started

- [Amazon EC2: User Guide](#)
- [How to connect to your Windows instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Microsoft Windows Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

[Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)

[Create and attach additional EBS volumes](#) (Additional charges may apply)

[Manage security groups](#)

[View instances](#)

[Feedback](#) [English](#)

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

## Step 4: Connect to Your Instance

After launching your instance, it's time to retrieve the administrator password and connect to it using a Remote Desktop Protocol (RDP) client.

---

a. Select the Windows Server instance you just created and click Connect

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Feedback. The main area has tabs for Launch Instance, Connect, and Actions. Below that is a search bar and a table with columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and Public IP. One row is selected, showing details for an instance named i-609f88e0, which is a t2.micro type in us-east-1b, currently running, with a status of Initializing, no alarm, and a public DNS of ec2-54-88-54-198.compute-1.amazonaws.com and a public IP of 54.88.54.198.

b. In order to connect to your Windows virtual machine instance, you will need a user name and password:

- The User name defaults to *Administrator*
- To receive your password, click Get Password

## Connect To Your Instance

X

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download Remote Desktop File](#)

When prompted, connect to your instance using the following details:

**Public DNS** ec2-54-88-54-198.compute-1.amazonaws.com

**User name** Administrator

**Password** [Get Password](#)

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

[Close](#)

- c. In order to retrieve the password, you will need to locate the Key Pair you created in Step 3. Click Choose File and browse to the directory you stored *MyFirstKey.pem*. Your Key Pair will surface in the text box.

Click Decrypt Password.

## Connect To Your Instance > Get Password

X

The following Key Pair was associated with this instance when it was created.

**Key Name** MyFirstKey2.pem

In order to retrieve your password you will need to specify the path of this Key Pair on your local machine:

**Key Pair Path**  No file chosen

Or you can copy and paste the contents of the Key Pair below:

-----BEGIN RSA PRIVATE KEY-----

- d. You now have a decrypted password for your Windows Server instance. Make sure to save this information in a secure location. It is your Windows Server admin login credentials.

## Connect To Your Instance

X

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download Remote Desktop File](#)

When prompted, connect to your instance using the following details:

**Public DNS** ec2-54-88-54-198.compute-1.amazonaws.com

**User name** Administrator

**Password** JC!gkswh)%

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

[Close](#)

- e. Click Download Remote Desktop File and open the file.

## Connect To Your Instance

X

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download Remote Desktop File](#)

When prompted, connect to your instance using the following details:

**Public DNS** ec2-54-88-54-198.compute-1.amazonaws.com

**User name** Administrator

**Password** JC!gkswh)%

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

[Close](#)

- f. When prompted log in to the instance, use the User Name and Password you generated in to connect to your virtual machine.

Note: When you complete this step, you might get a warning that the security certificate could not be authenticated. Simply choose yes and proceed to complete the connection to your Windows Server instance



## Step 5: Terminate Your Windows VM

You can easily terminate the Windows Server VM from the Amazon EC2 console. In fact, it is a best practice to terminate instances you are no longer using so you don't keep getting charged for them.

- 
- a. Back on the EC2 Console, select the box next to the instance you created. Then click the Actions button, navigate to *Instance State*, and click Terminate.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, Auto Scaling, and Load Balancing. The main area lists three instances:

| Name | Instance ID | Instance State | Availability Zone | Status Checks | Alarm Status | Public DNS                               | Public IP     | Key Name    | Monitoring | Launch Time                   |
|------|-------------|----------------|-------------------|---------------|--------------|--|---------------|-------------|------------|-------------------------------|
|      | i-609f88e0  | running        | us-east-1b        | 2/2 checks... | None         | ec2-54-88-54-198.compute-1.amazonaws.com | 54.88.54.198  | MyFirstKey  | disabled   | February 23, 2016 at 10:51... |
|      | i-647659c   | running        | us-east-1d        | 2/2 checks... | None         | ec2-54-175-143-66.com                    | 54.175.143.66 | MyFirstKey2 | disabled   | February 23, 2016 at 2:17...  |
|      | i-a3c5e03b  | terminated     | us-east-1d        | None          | None         | None                                     | None          | MyFirstKey  | disabled   | February 23, 2016 at 12:45... |

A context menu is open over the first instance (i-609f88e0). The 'Actions' dropdown has several options: Connect, Get Windows Password, Launch More Like This, Start, Stop, Reboot, and Terminate. The 'Terminate' option is highlighted with a red box.

The detailed view for instance i-609f88e0 shows its configuration:

| Description           | Value                         |
|-----------------------|-------------------------------|
| Instance ID           | i-609f88e0                    |
| Instance state        | running                       |
| Instance type         | t2.micro                      |
| Private DNS           | ip-172-31-52-176.ec2.internal |
| Private IPs           | 172.31.52.176                 |
| Secondary private IPs |                               |
| VPC ID                | vpc-4e3f712a                  |

At the bottom, there are tabs for Status Checks, Monitoring, and Tags. The Monitoring tab is selected. The status bar at the bottom right indicates: © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

b. You will be asked to confirm your termination - select Yes, Terminate.

**Note:** This process can take several seconds to complete. Once your instances has been terminated, the Instance State will change to **terminated** on your EC2 Console.

The screenshot shows the AWS EC2 Instances page with the same three instances listed. The first instance (i-609f88e0) is selected. A 'Terminate Instances' dialog box is open over it, containing the following text:

**Warning**  
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

- i-609f88e0 (ec2-54-88-54-198.compute-1.amazonaws.com)

At the bottom right of the dialog box, the 'Yes, Terminate' button is highlighted with a red box.

The detailed view for instance i-609f88e0 is shown again, with the Monitoring tab selected. The status bar at the bottom right indicates: © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

You have created, configured, and connected to your first Windows Virtual Machine in the cloud with Amazon EC2.

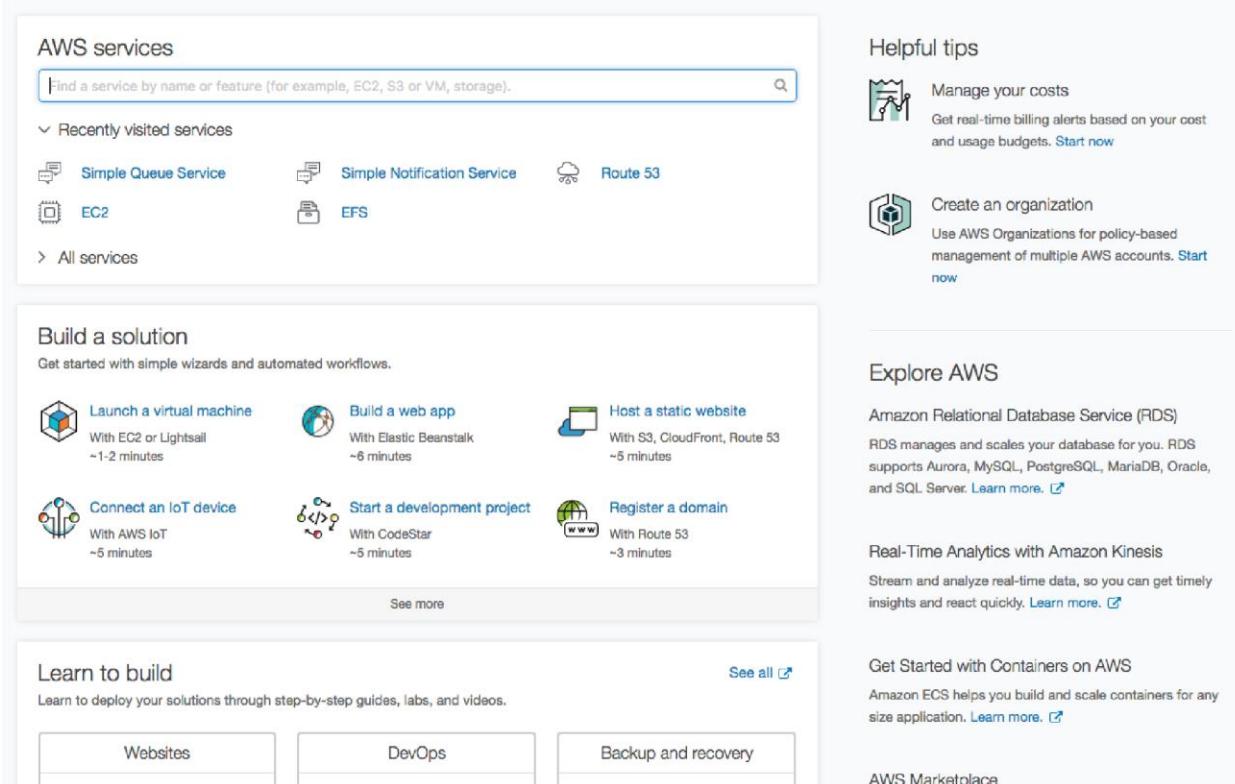
# BONUS

## Store and Retrieve a File

These step-by-step instructions will help you store your files in the cloud using Amazon Simple Storage Solution (S3). Amazon S3 is a service that enables you to store your data (referred to as *objects*) in at massive scale. In this task, you will create an Amazon S3 bucket, upload a file, retrieve the file and delete the file.

### Step 1. Enter the Amazon S3 Console

When you [click here](#), the AWS Management Console will open in a new browser window, so you can keep this step-by-step guide open. When the screen loads, enter your user name and password to get started. Then type S3 in the search bar and select S3 to open the console.



The screenshot shows the AWS Management Console homepage. At the top, there's a search bar with placeholder text "Find a service by name or feature (for example, EC2, S3 or VM, storage)." Below the search bar, under "Recently visited services," are links for Simple Queue Service, EC2, Simple Notification Service, EFS, and Route 53. There's also a link to "All services." To the right, under "Helpful tips," are two sections: "Manage your costs" (with a graph icon) and "Create an organization" (with a hexagon icon). Below these are "Build a solution" and "Learn to build" sections, each with several options like "Launch a virtual machine" (with EC2 or Lightsail, ~1-2 minutes), "Connect an IoT device" (With AWS IoT, ~5 minutes), "Build a web app" (With Elastic Beanstalk, ~8 minutes), "Start a development project" (With CodeStar, ~5 minutes), "Host a static website" (With S3, CloudFront, Route 53, ~5 minutes), "Register a domain" (With Route 53, ~3 minutes), and "Websites," "DevOps," and "Backup and recovery." At the bottom, there are "Explore AWS" sections for "Amazon Relational Database Service (RDS)" (described as managing and scaling database for you, supporting Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server), "Real-Time Analytics with Amazon Kinesis" (described as stream and analyze real-time data), "Get Started with Containers on AWS" (described as building and scaling containers for any size application), and "AWS Marketplace."

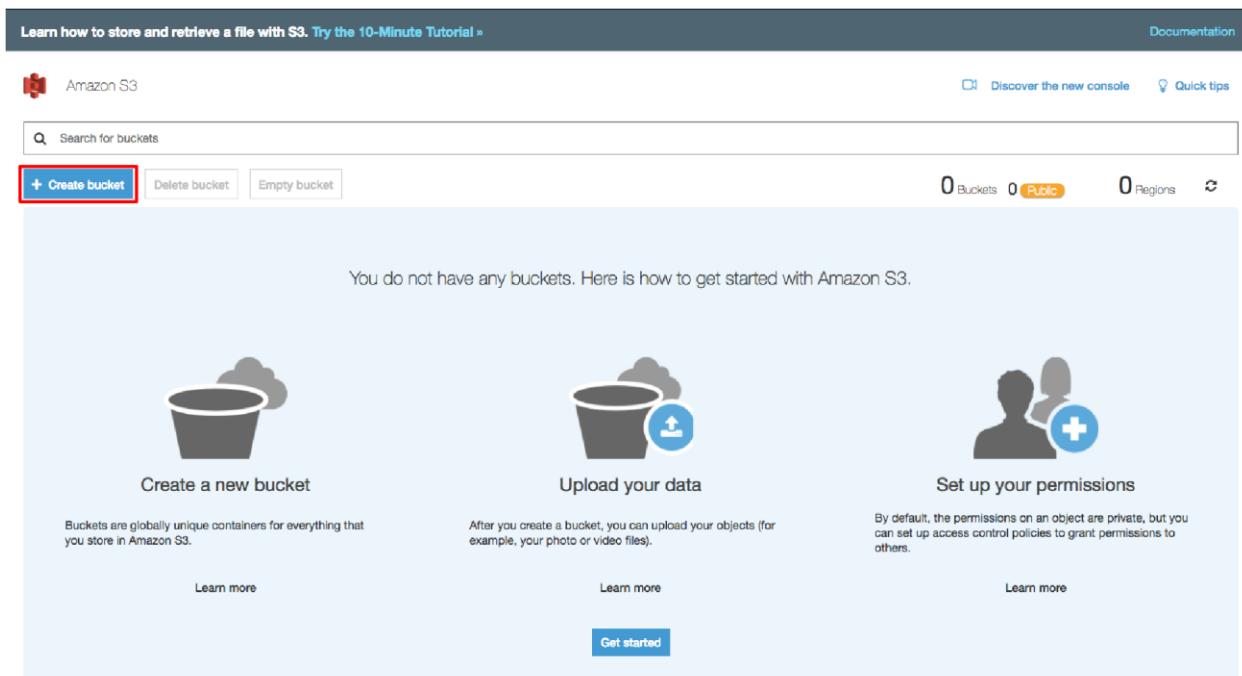
## Step 2. Create an S3 Bucket

In this step, you will create an Amazon S3 *bucket*. A bucket is the container you store your files in.

- 
- a. In the S3 dashboard, click Create Bucket.

If this is the first time you have created a bucket, you will see a screen that looks like the image pictured here.

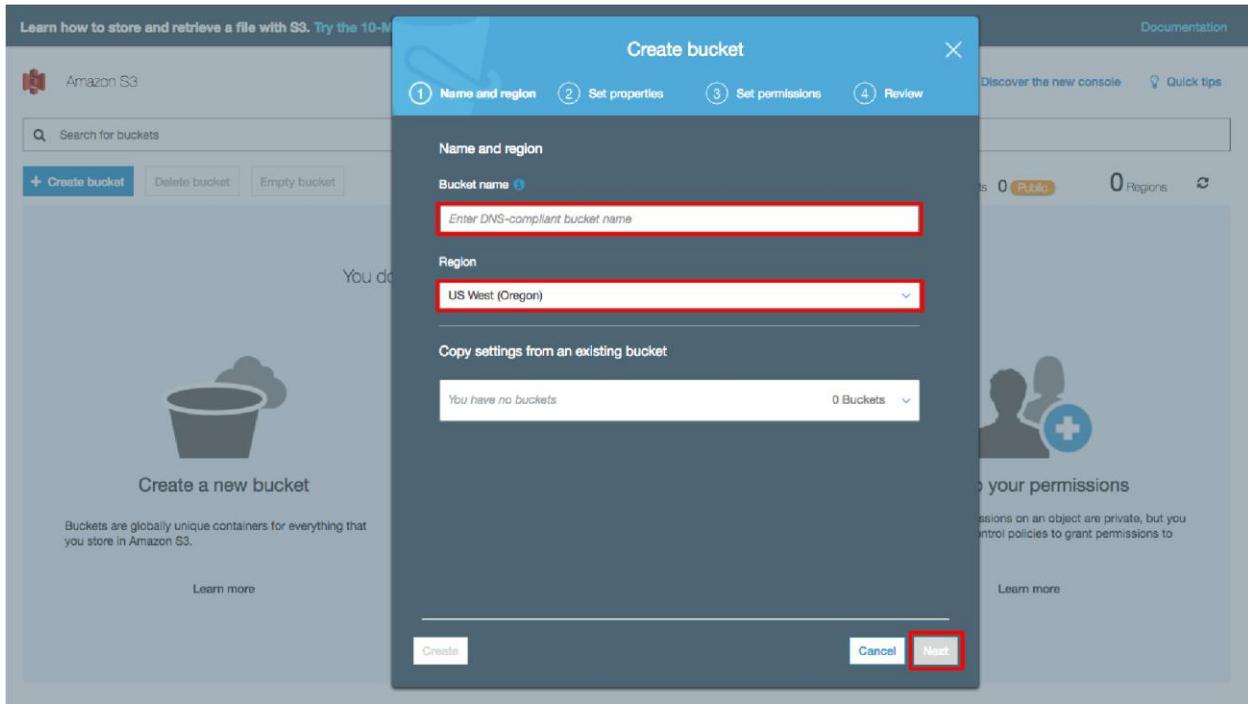
If you have already created S3 buckets, your S3 dashboard will list all the buckets you have created.



The screenshot shows the Amazon S3 dashboard. At the top, there is a banner with a link to a tutorial and a 'Documentation' button. Below the banner, the Amazon S3 logo is on the left, and navigation links for 'Discover the new console' and 'Quick tips' are on the right. A search bar labeled 'Search for buckets' is in the center. Below the search bar, there are three buttons: '+ Create bucket' (which is highlighted with a red box), 'Delete bucket', and 'Empty bucket'. To the right of these buttons, it says '0 Buckets' and '0 Public'. Further right are '0 Regions' and a refresh icon. The main content area has a light blue background. It displays a message: 'You do not have any buckets. Here is how to get started with Amazon S3.' Below this message are three sections: 'Create a new bucket' (with an icon of a bucket and cloud), 'Upload your data' (with an icon of a bucket and a plus sign), and 'Set up your permissions' (with an icon of two people and a plus sign). Each section has a brief description and a 'Learn more' link. At the bottom of the main content area is a blue 'Get started' button.

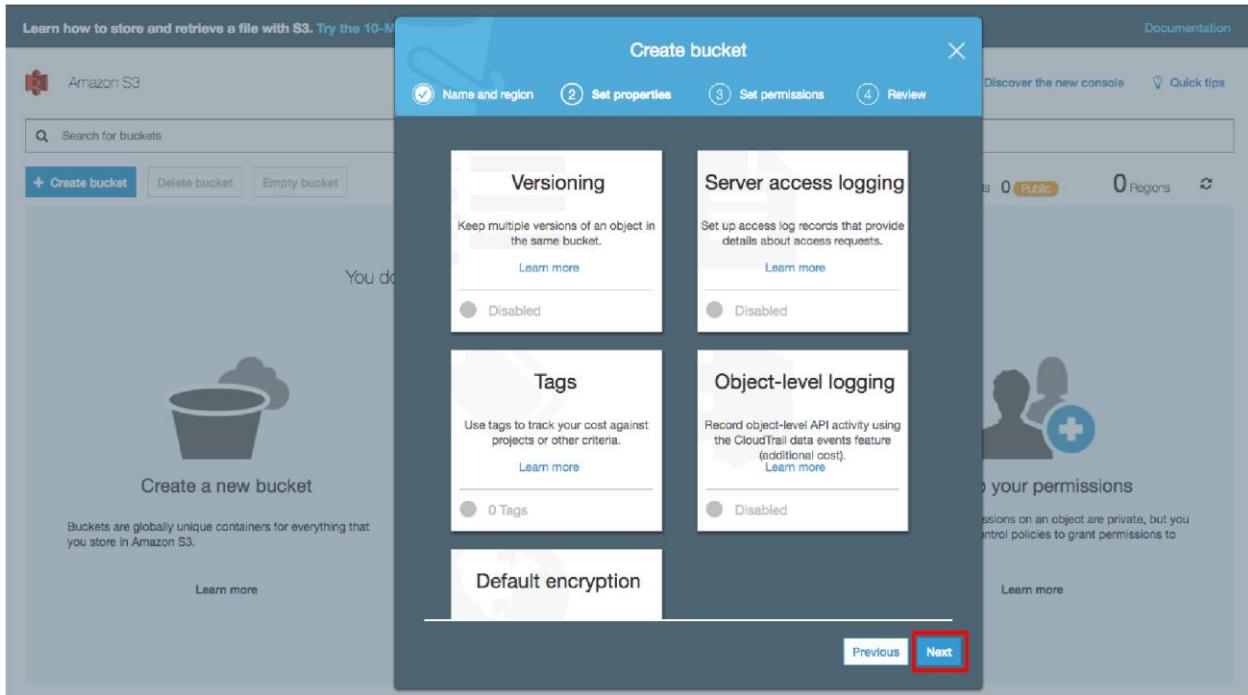
- b. Enter a bucket name. Bucket names must be unique across all existing bucket names in Amazon S3. There are a number of other [restrictions on S3 bucket names](#) as well. Then select a region to create your bucket in.

Select Next.

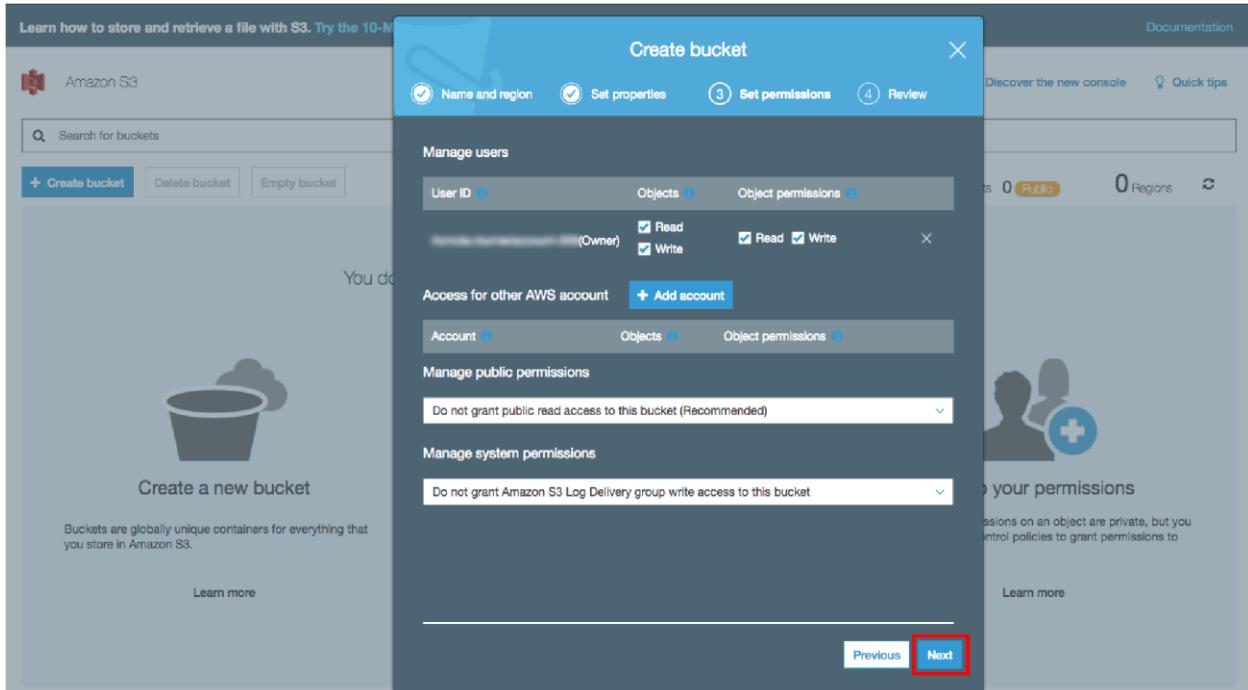


- c. You have many useful options for your S3 bucket including [Versioning](#), [Server Access Logging](#), [Tags](#), [Object-level Logging](#) and [Default Encryption](#). We won't enable them for this tutorial.

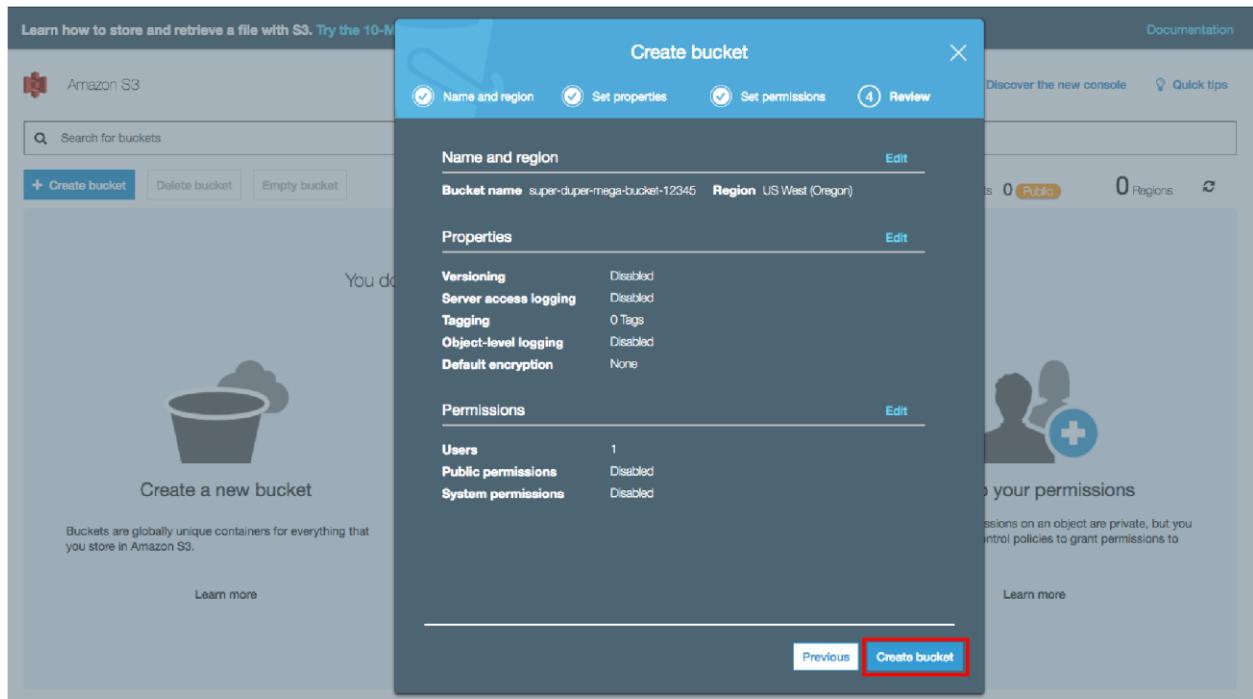
Select Next.



- d. You have the ability to set permission settings for your S3 bucket. Leave the default values and select Next.



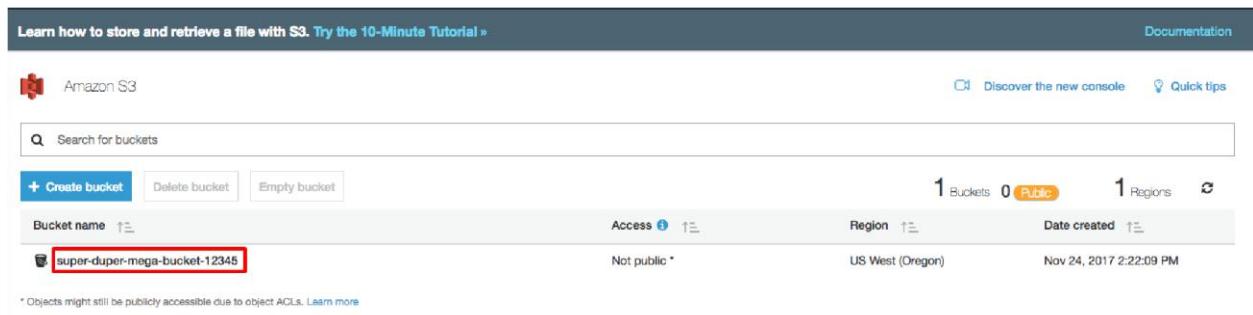
- e. Review your configuration settings and select Create bucket.



## Step 3. Upload a File

In this step, you will upload a file to your new Amazon S3 bucket.

- a. You will see your new bucket in the S3 console. Click on your bucket's name to navigate to the bucket.

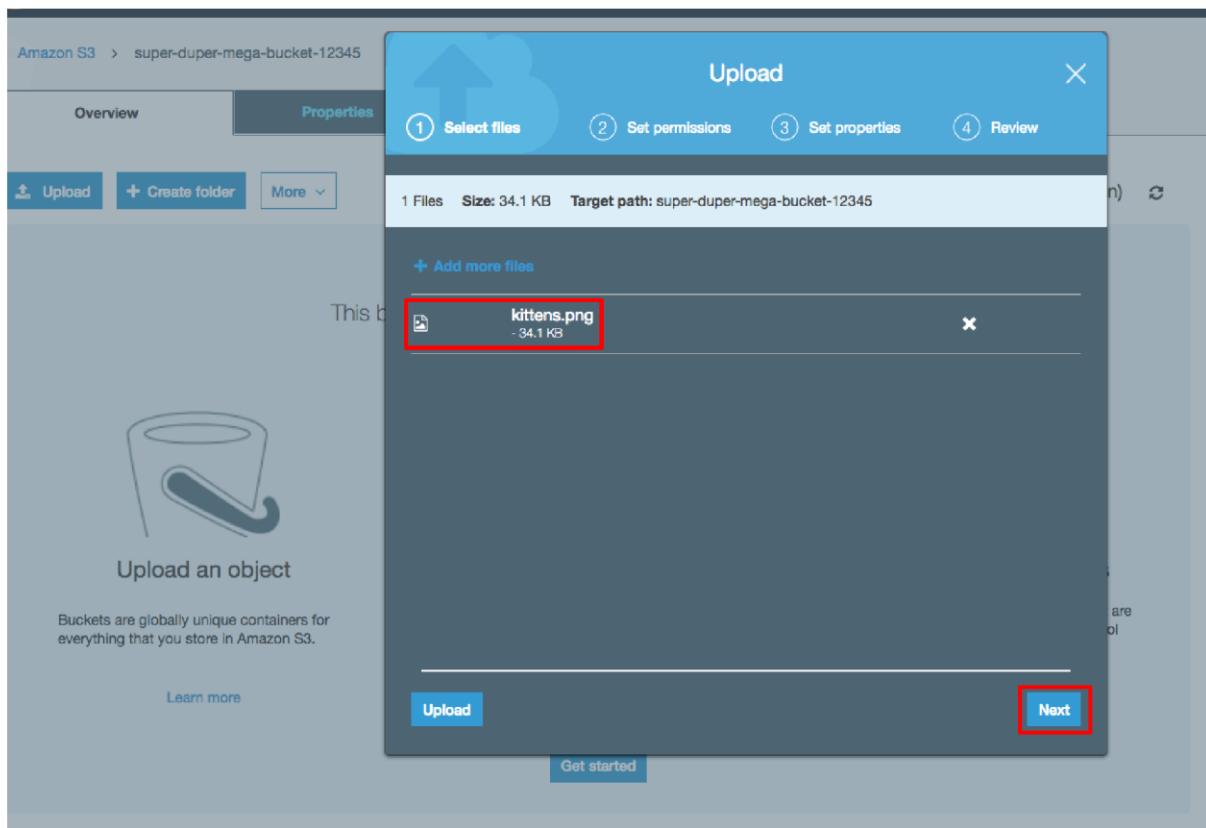


(click to zoom)

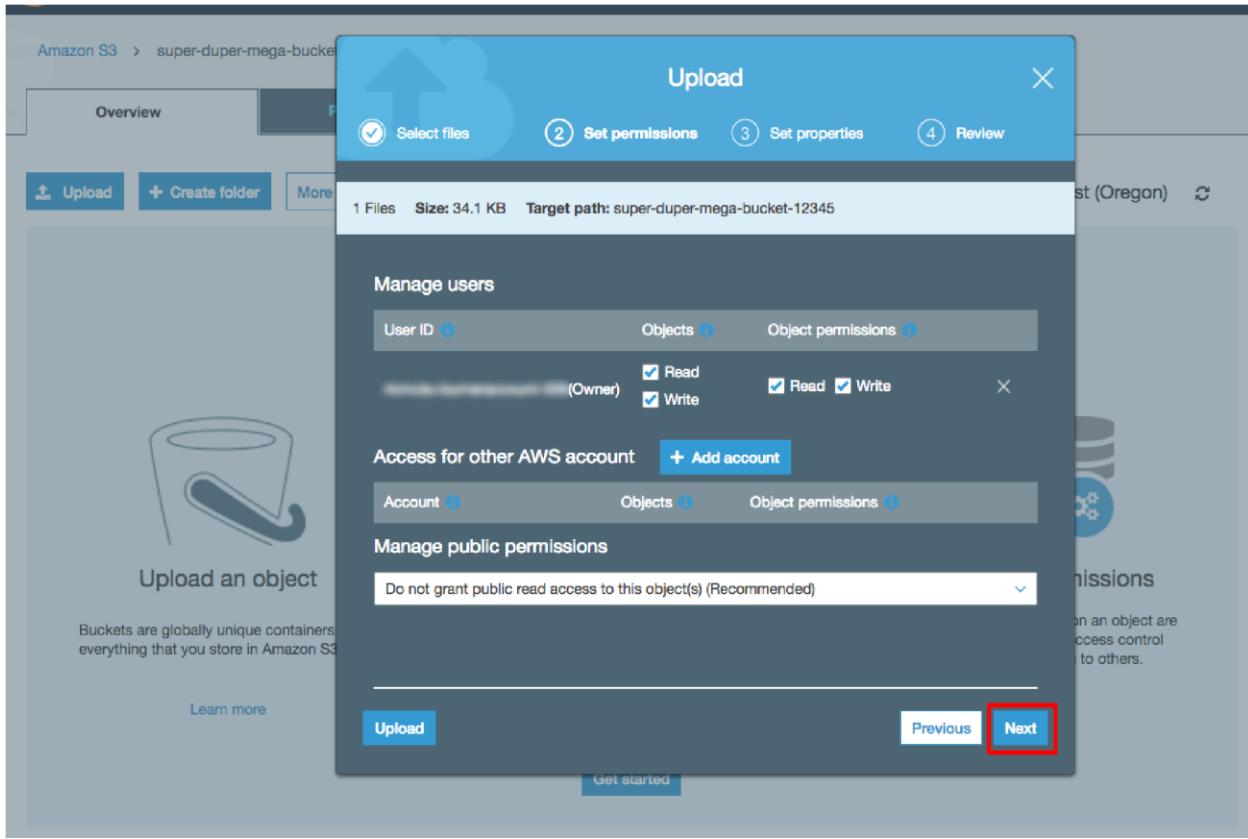
b. You are in your bucket's home page. Select Upload.

The screenshot shows the Amazon S3 console interface for a bucket named "super-duper-mega-bucket-12345". At the top, there are tabs for Overview, Properties, Permissions, and Management. Below these, a secondary navigation bar includes an "Upload" button, which is highlighted with a red box, and other options like "+ Create folder" and "More". On the right, it shows the location "US West (Oregon)" and a refresh icon. The main content area displays a message: "This bucket is empty. Upload new objects to get started." It features three large icons: "Upload an object" (a bucket with a file), "Set object properties" (two user silhouettes with a plus sign), and "Set object permissions" (two cylinders with a gear). Each section has a "Learn more" link and a "Get started" button at the bottom. The "Upload" button is the primary focus of the instruction.

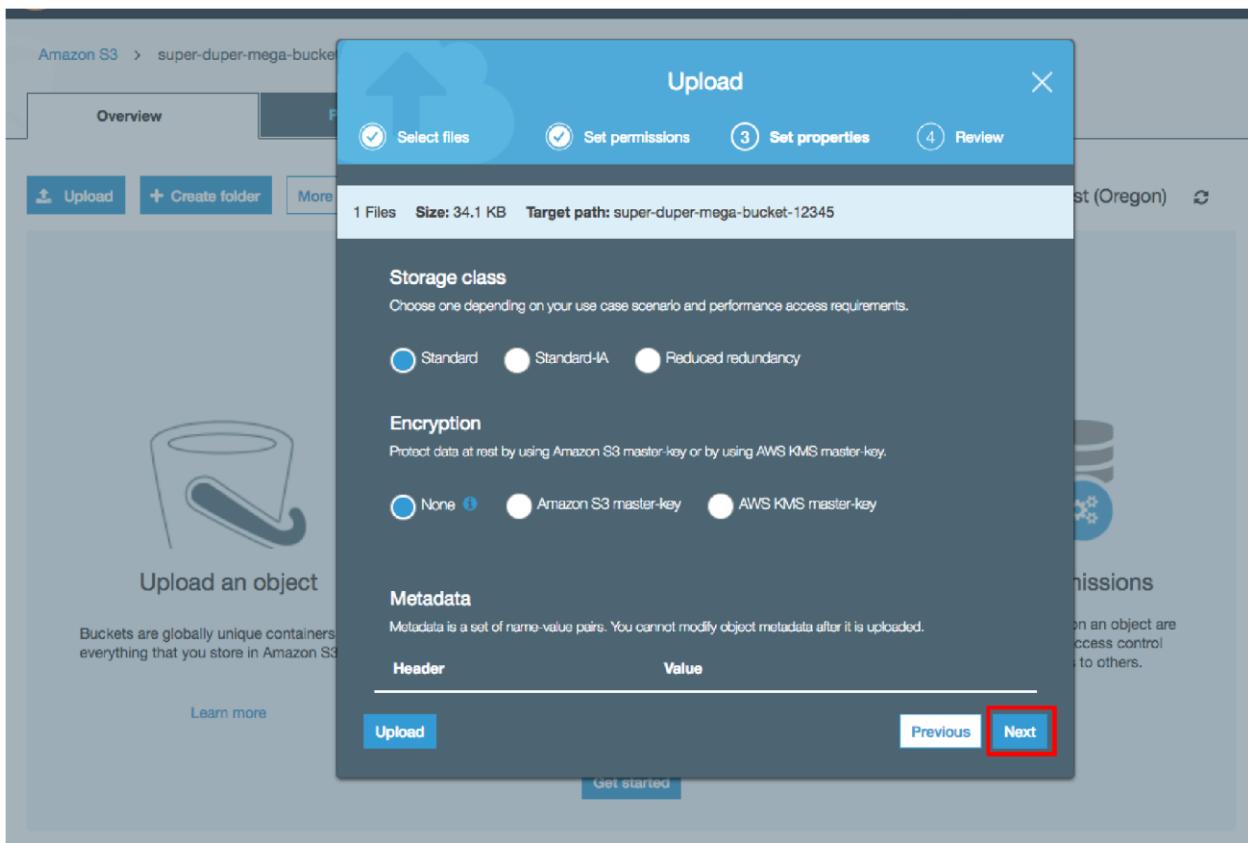
c. To select a file to upload, either click Add files and select a sample file that you would like to store OR Drag and Drop a file on the upload box.  
Select Next after you have selected a file to upload.



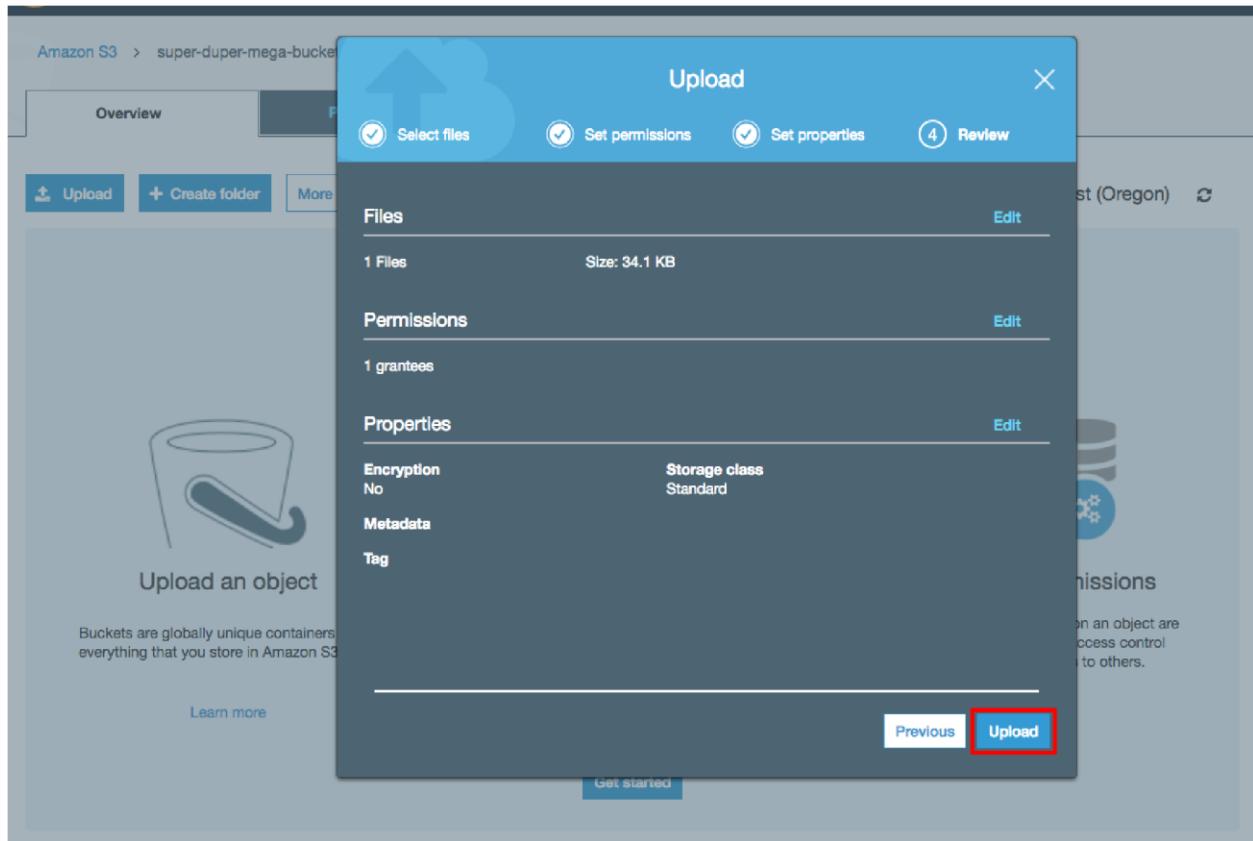
- d. You have the ability to set permission settings on your object. For this tutorial, leave the default values and select Next.



- e. You have the ability to set property settings like storage class, encryption, and metadata with your object. Leave the default values and select Next.



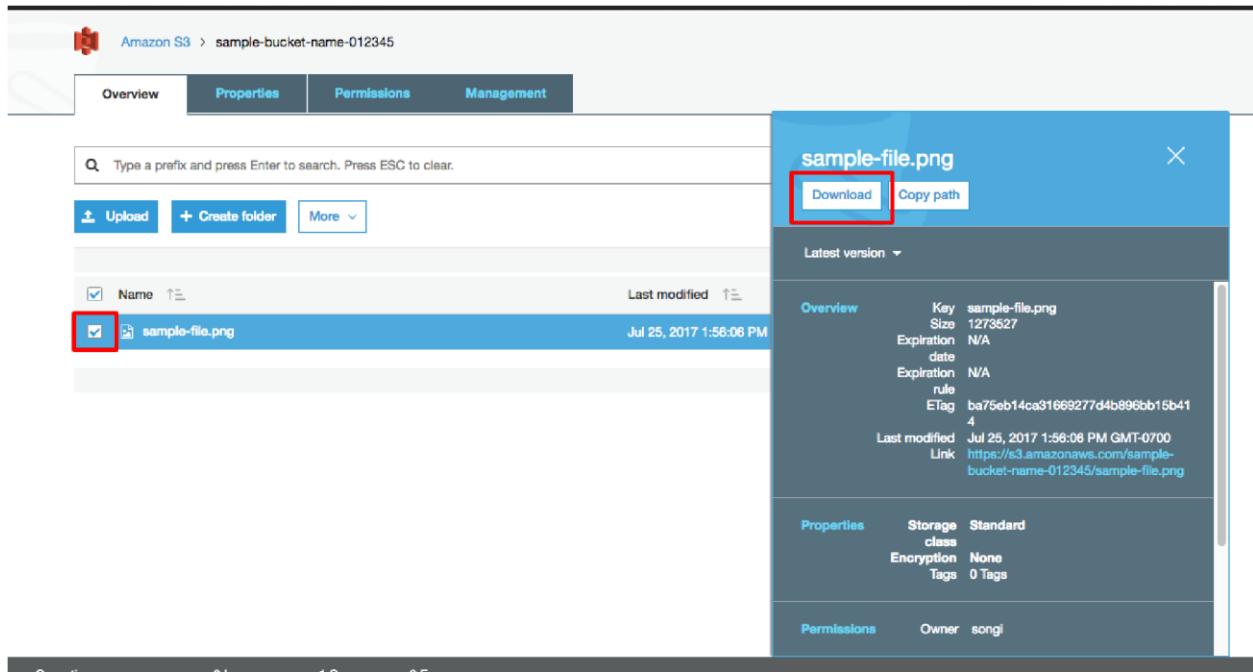
- f. Review your configurations and select Upload. You will see your object in your bucket's home screen.



## Step 4: Retrieve the Object

In this step, you will download the file from your Amazon S3 bucket.

- 
- Select the checkbox next to the file you would like to download, then select Download.



## Step 5: Delete the Object and Bucket

You can easily delete your object and bucket from the Amazon S3 console. In fact, it is a best practice to delete resources you are no longer using so you don't keep getting charged for them.

- 
- a. You will first delete your object. Select the checkbox next to the file you want to delete and select More > Delete.

Amazon S3 > super-duper-mega-bucket-12345

Overview Properties Permissions Management

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder More ▾

US West (Oregon) Viewing 1 to 1

|   | Last modified                    | Size    | Storage class |
|---|----------------------------------|---------|---------------|
| <input checked="" type="checkbox"/>  Kittens.png | Nov 24, 2017 4:23:11 PM GMT-0800 | 34.1 KB | Standard      |

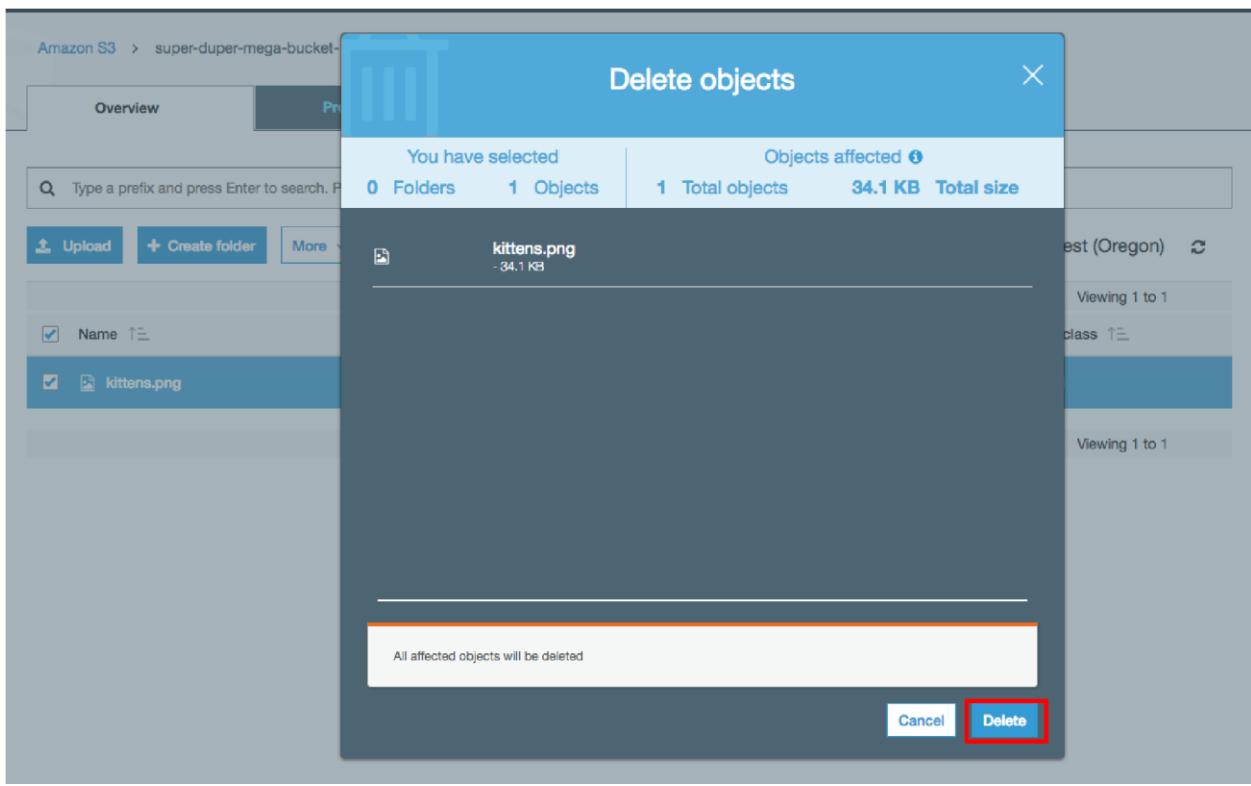
Viewing 1 to 1

Kittens.png

More ▾

- Open
- Get size
- Download as
- Rename
- Delete**
- Undo delete
- Cut
- Copy
- Paste
- Change storage class
- Initiate restore
- Change encryption
- Change metadata
- Make public
- Add tags

- b. Review and confirm the object you want to delete. Select Delete.



- c. Click on Amazon S3 to view all your buckets in the region.

Amazon S3 > super-duper-mega-bucket-12345

Overview Properties Permissions Management

Upload Create folder More

US West (Oregon)

This bucket is empty. Upload new objects to get started.

 Upload an object

Buckets are globally unique containers for everything that you store in Amazon S3.

[Learn more](#)

 Set object properties

After you create a bucket, you can upload your objects (for example, your photo or video files).

[Learn more](#)

 Set object permissions

By default, the permissions on an object are private, but you can set up access control policies to grant permissions to others.

[Learn more](#)

**Get started**

- d. Click to the right of the bucket name of the bucket you created to selected it, then click Delete. Type in the name of your bucket and click Confirm.

Learn how to store and retrieve a file with S3. Try the 10-Minute Tutorial »

Documentation



Amazon S3

Discover the new console

Quick tips

Search for buckets

+ Create bucket

Delete bucket

Empty bucket

1 Buckets 0 Public

1 Regions

Bucket name ↑↓

Access ↑↓

Region ↑↓

Date created ↑↓

super-duper-mega-bucket-12345

Not public \*

US West (Oregon)

Nov 24, 2017 2:22:09 PM

\* Objects might still be publicly accessible due to object ACLs. Learn more

You have backed up your first file to the cloud by creating an Amazon S3 bucket and uploading your file as an S3 object. Amazon S3 is designed for 99.99999999% durability to help ensure that your data is always available when you want it. You've also learned how to retrieve your backed up file and how to delete the file and bucket.

## Seminar 2

# Creating and Connect to MS SQL in Amazon RDS

In this seminar, you will learn how to create different database Instances, connect to these databases, and delete the DB instances. You will do this using Amazon Relational Database Service (Amazon RDS). Everything done in this seminar is free tier eligible as long as you delete the instance that you have created after you have used it during the class.

Step 1:

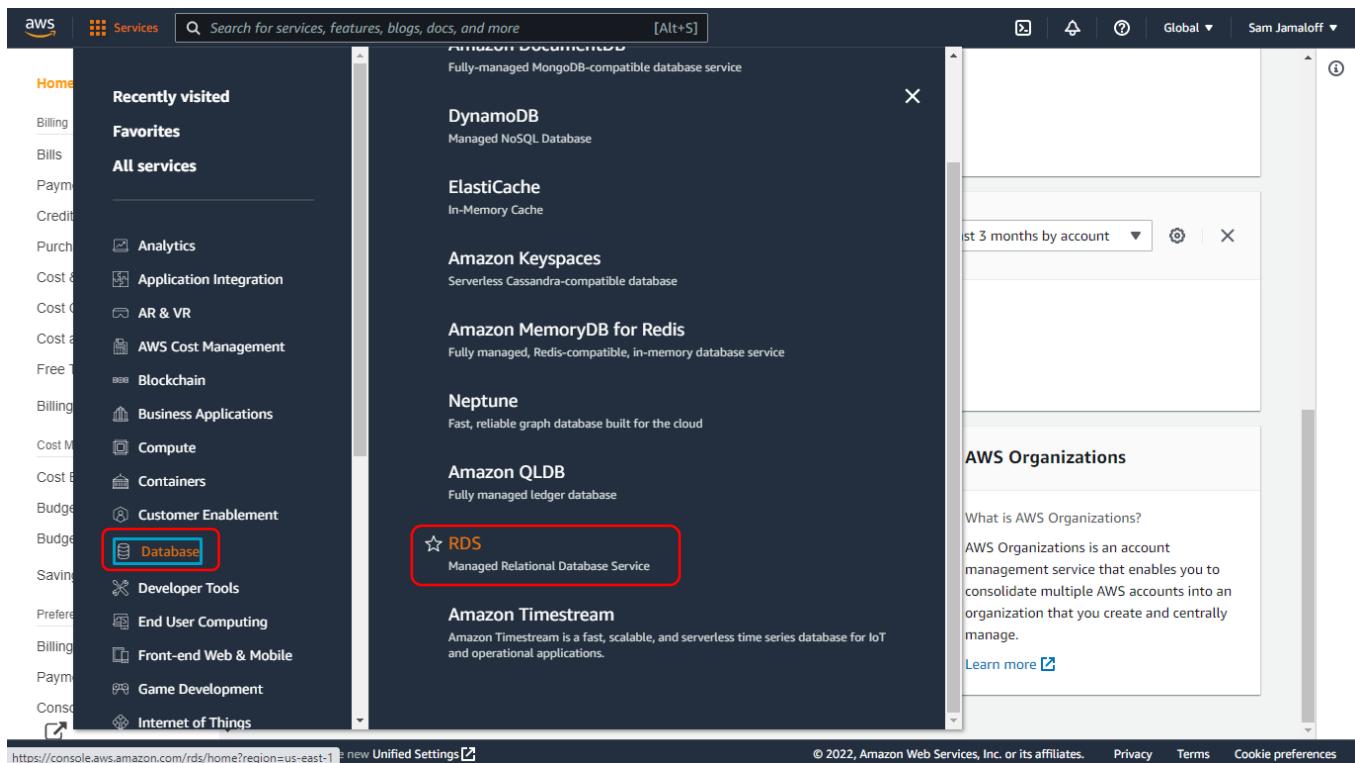
Enter the RDS Console When you [click here](#), the AWS management console will open in a new browser window, so you can keep this step-by-step guide open. When this screen loads, find RDS under Database and click to open the Amazon RDS Console.

Step 2:

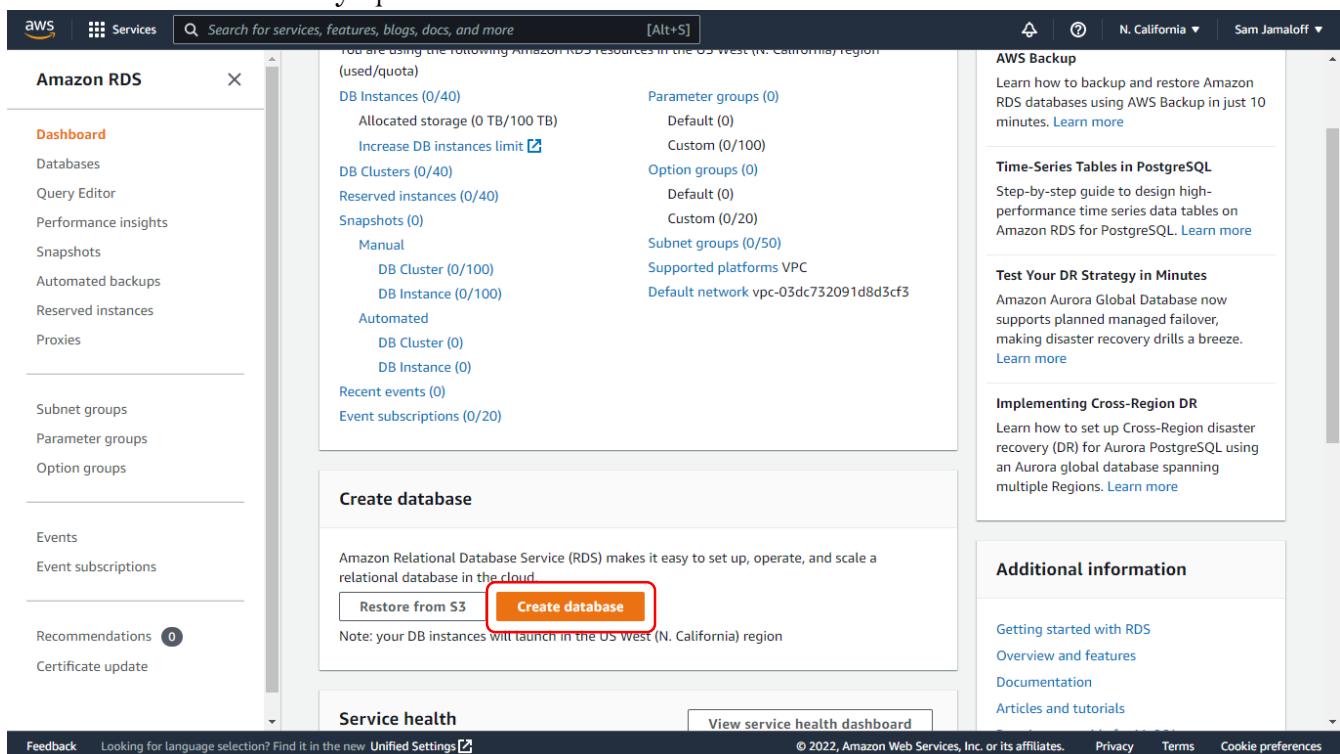
Create a Microsoft SQL Server DB Instance In this step, we will use Amazon RDS to create a Microsoft SQL Server DB Instance with db.t3.micro DB instance class, 20 GB of storage, and automated backups enabled with a retention period of one day. As a reminder, all of this is free tier eligible for 7 days and as long as you delete the instance after you have used it.

- a. In the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance.

Note: AWS Cloud resources are housed in highly available data center facilities in different areas of the world. Each Region contains multiple distinct locations called Availability Zones. You have the ability to choose which Region to host your Amazon RDS activity in.



b. And in the newly opened window Click “*Create Database*” button.



c. You now have an option to choose your engine. For seminar purposes click the “*Microsoft SQL Server*” icon. Select *SQL Server Express* edition. For this seminar with Microsoft SQL Server you can easily set it up via choosing easy set up option.

S | Services | Search for services, features, blogs, docs, and more | [Alt+S] | N. California | Sam Jamaloff

## Create database

**Choose a database creation method** [Info](#)

Standard create  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Configuration**

Engine type [Info](#)

Amazon Aurora 

MySQL 

MariaDB 

PostgreSQL 

Oracle 

Microsoft SQL Server 

DB instance size

Feedback Looking for language selection? Find it in the new Unified Settings [?](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- d. Choose Free-tier option for this seminar.
- e. You can change:
  - DB instance identifier into anything (this will be shown in the list of your databases)
  - Master username, if not changed it is admin by default
  - Master password (You have to set new password, if not created AWS will not create for you and cannot be move to the next step)
- f. From there click “Create Database” button

S | Services | Search for services, features, blogs, docs, and more [Alt+S] | N. California | Sam Jamaloff

DB instance size

Production  
db.r5.xlarge  
4 vCPUs  
32 GB RAM  
500 GB  
3.369 USD/hour

Dev/Test  
db.m5.large  
2 vCPUs  
8 GB RAM  
100 GB  
1.026 USD/hour

Free tier  
db.t2.micro  
1 vCPUs  
1 GB RAM  
20 GB  
0.031 USD/hour

DB instance identifier  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.  
database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username [Info](#)  
Type a login ID for the master user of your DB instance.  
admin

1 to 16 alphanumeric characters. First character must be a letter.

Auto generate a password  
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)  
.....  
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm password [Info](#)  
.....

Feedback

Looking for language selection? Find it in the new [Unified Settings](#).

© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Database creation might take some time from 3-15 minutes at max.

So, while waiting you can move to the next step and install our SQL client to connect to your newly created database.

Go to [this website](#), and download latest SQL Server management studio.

## Download SSMS

To download SSMS 19 Preview 3, visit [Download SSMS 19](#).

[↓ Free Download for SQL Server Management Studio \(SSMS\) 18.12.1 ↗](#)

SSMS 18.12.1 is the latest general availability (GA) version. If you have a previous GA version of SSMS 18 installed, installing SSMS 18.12.1 upgrades it to 18.12.1.

- Release number: 18.12.1
- Build number: 15.0.18424.0
- Release date: June 21, 2022

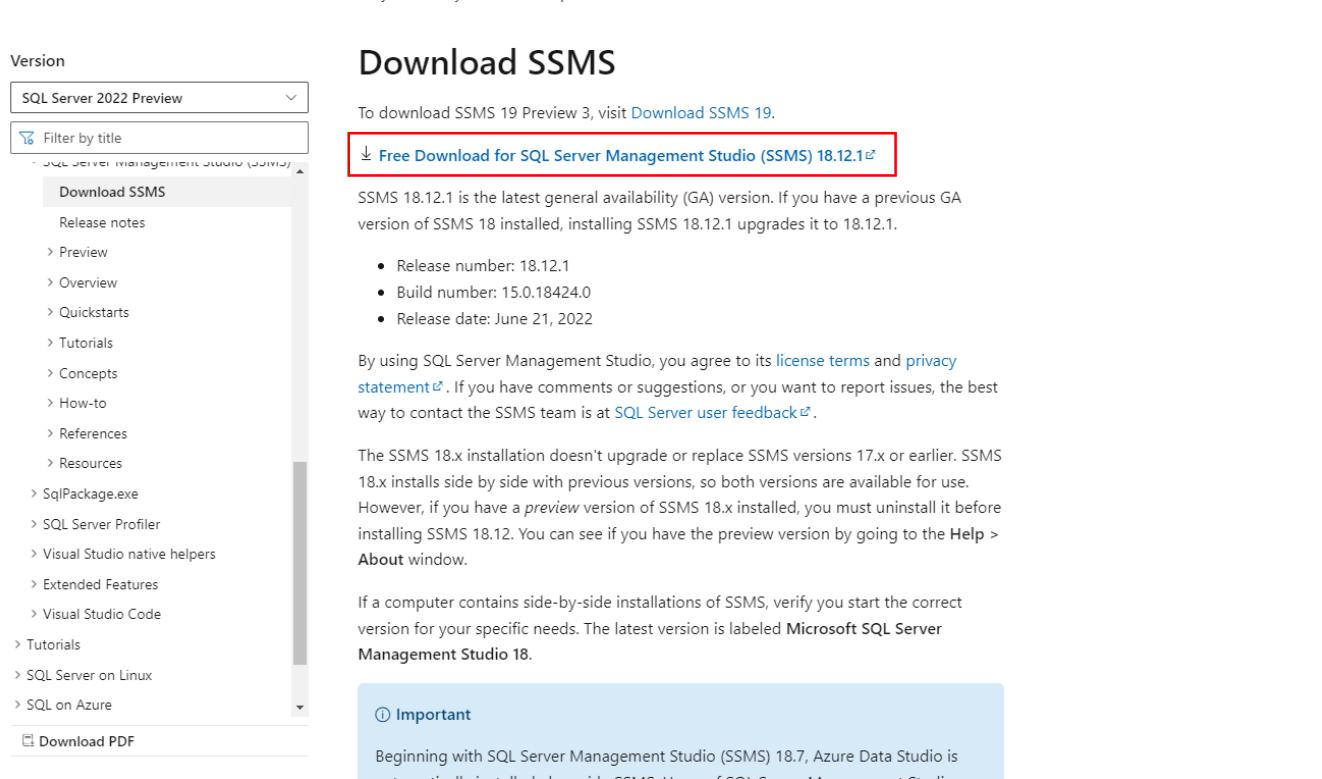
By using SQL Server Management Studio, you agree to its [license terms](#) and [privacy statement](#). If you have comments or suggestions, or you want to report issues, the best way to contact the SSMS team is at [SQL Server user feedback](#).

The SSMS 18.x installation doesn't upgrade or replace SSMS versions 17.x or earlier. SSMS 18.x installs side by side with previous versions, so both versions are available for use. However, if you have a *preview* version of SSMS 18.x installed, you must uninstall it before installing SSMS 18.12. You can see if you have the preview version by going to the **Help > About** window.

If a computer contains side-by-side installations of SSMS, verify you start the correct version for your specific needs. The latest version is labeled **Microsoft SQL Server Management Studio 18**.

**Important**

Beginning with SQL Server Management Studio (SSMS) 18.7, Azure Data Studio is



Install it. After installation click windows button and search for “*Microsoft SQL Server Management Studio*”, once you found open it.

## Download SSMS

Version

All Apps Documents Web More

Best match

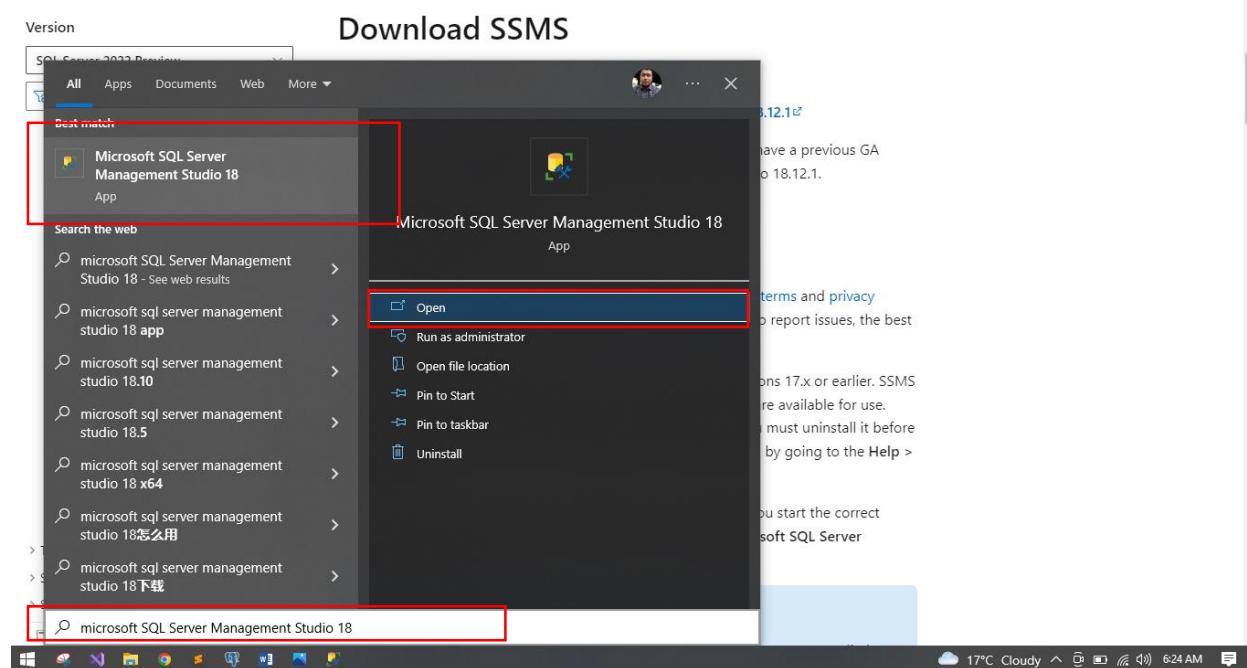
Microsoft SQL Server Management Studio 18 App

Search the web

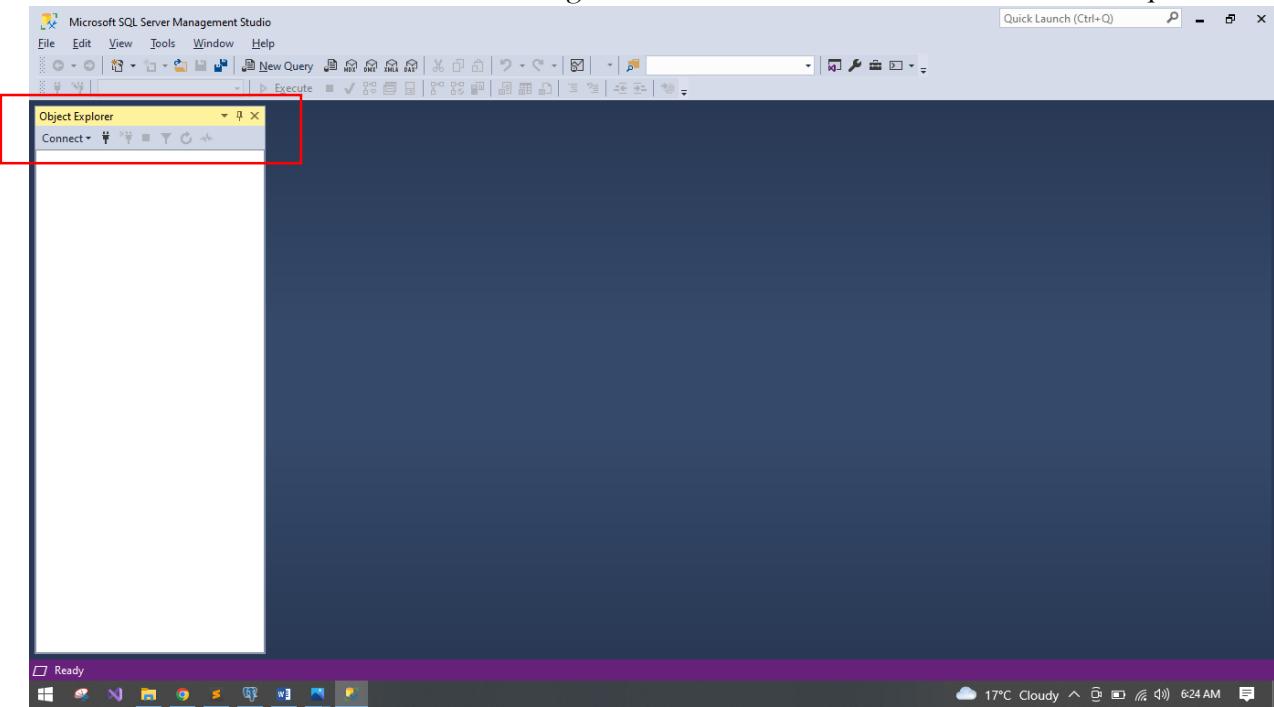
- microsoft SQL Server Management Studio 18 - See web results
- microsoft sql server management studio 18 app
- microsoft sql server management studio 18.10
- microsoft sql server management studio 18.5
- microsoft sql server management studio 18 x64
- microsoft sql server management studio 18.5.2
- microsoft sql server management studio 18.5.2.1
- microsoft SQL Server Management Studio 18

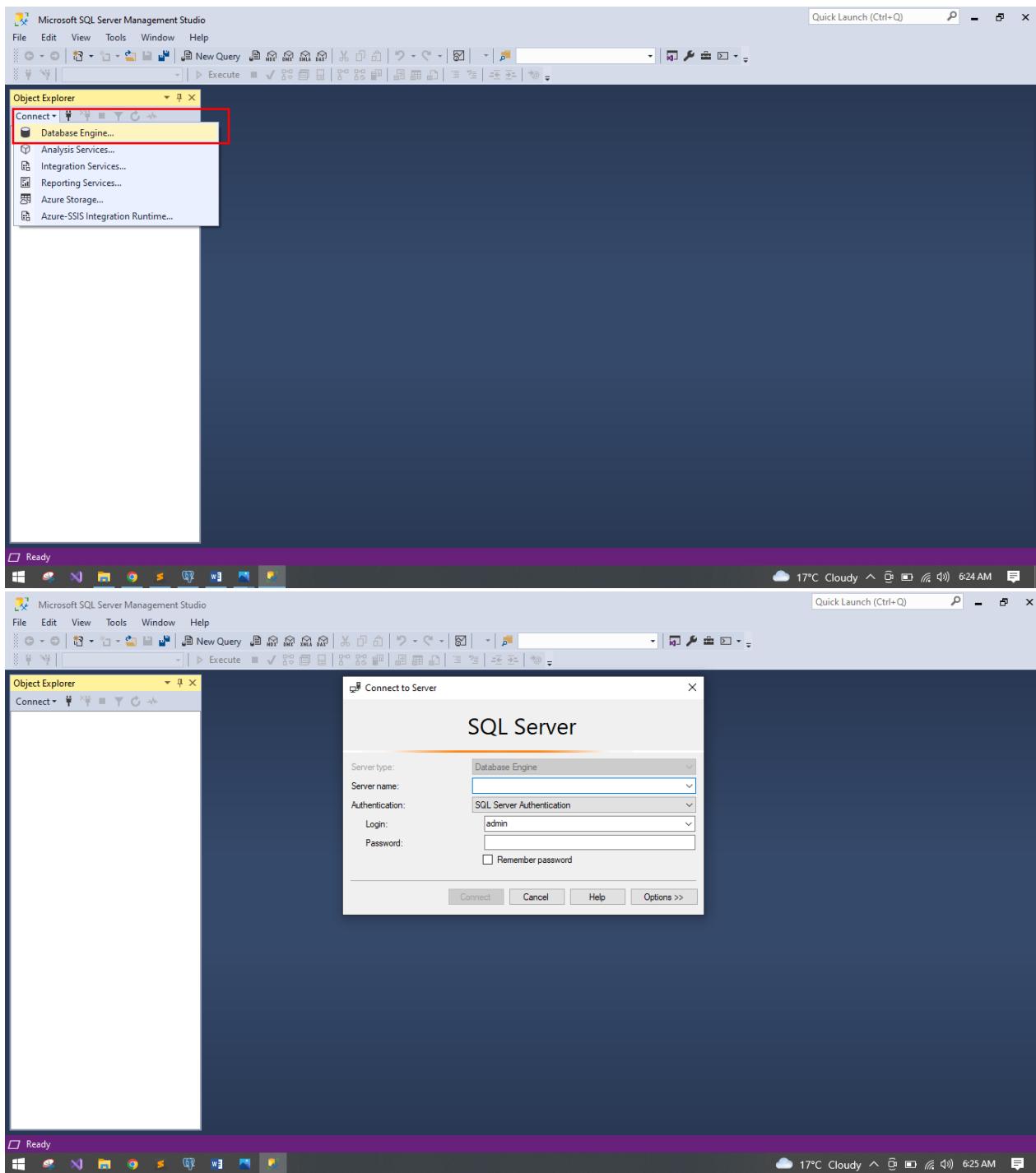
Microsoft SQL Server Management Studio 18 App

Open Run as administrator Open file location Pin to Start Pin to taskbar Uninstall



On the left most corner of the newly opened window of “Object Explorer” tab there is “Connect” button, and from there click “Database Engine”, click on that, and the window will be open.





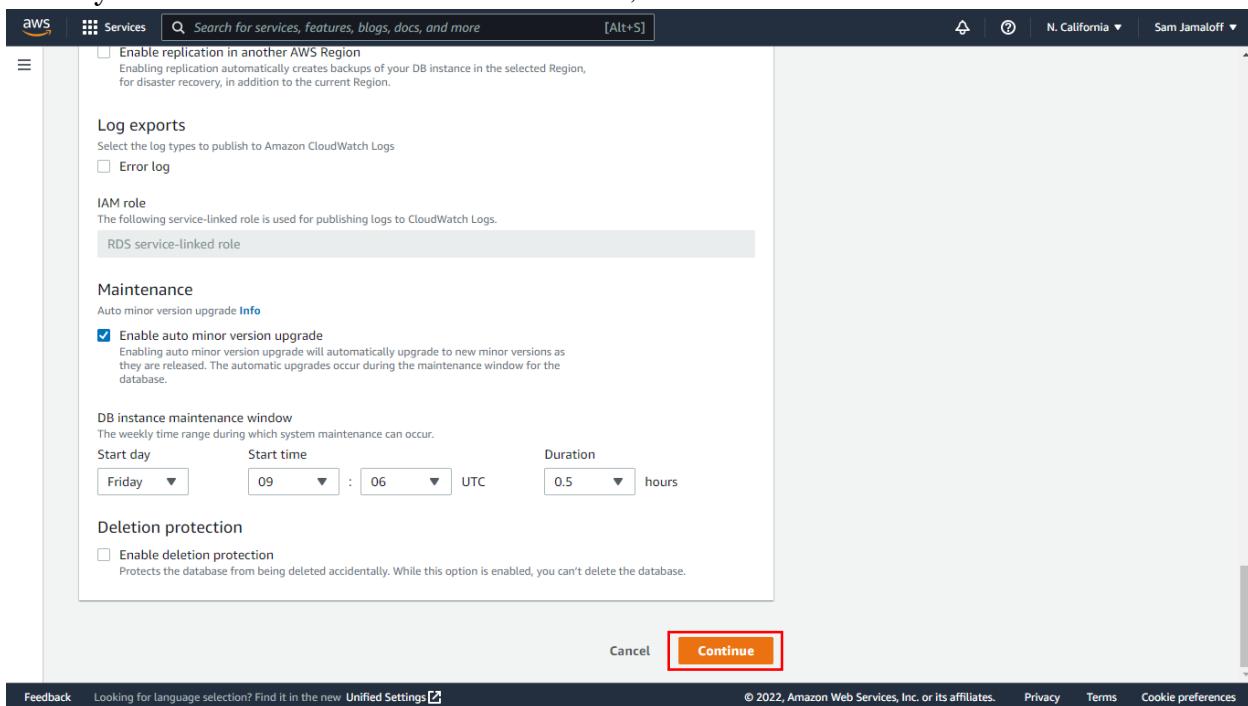
After your database is set up, go to your database instance and do the following. Mission out on doing this steps will lead to **server time-out** or **server not found** issues. Find Modify button on the right most corner, and click on that.

The screenshot shows the AWS RDS console with the database-1 instance selected. The 'Modify' button in the top right corner is highlighted with a red box. The 'Summary' section displays various instance details such as DB identifier, CPU usage, status, engine, and region. Below the summary, there are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The 'Connectivity & security' tab is active.

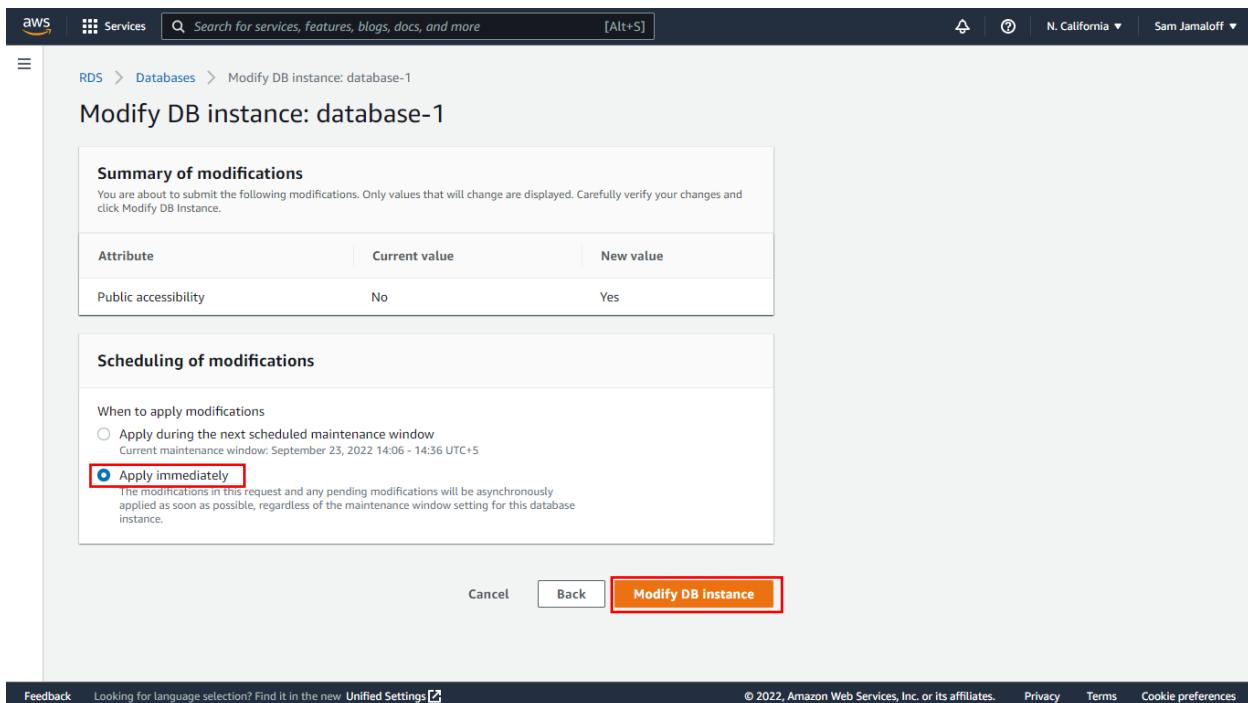
Once you are inside modification menu, scroll down until you find connectivity menu from the list. Click on additional configuration, and change the selected option to “*Public access*”. Leave everything as it is. Don’t change nothing.

The screenshot shows the 'Modify DB Instance' wizard on the 'Connectivity' step. The 'Publicly accessible' radio button under 'Additional configuration' is highlighted with a red box. Other configuration options shown include DB Subnet group (default-vpc), Security group (Choose security groups), Certificate authority (rds-ca-2019), and Database port (1433). The 'Feedback' bar at the bottom indicates looking for language selection in the new Unified Settings.

When you scroll down there is Continue button, click on that.



On the new window choose “*Apply immediately*” option, and Modify DB instance.



And now go back to your DB instance. Because we created our database instance using the best preloaded practices by choosing easy set-up default at the beginning, default “*default security*

*group*" option is used. In order to access to it we need to modify the VPC of our instance. On the "Connectivity and security" click on "*default {name\_of\_sg}*".

The screenshot shows the AWS RDS console for a database named 'database-1'. The 'Summary' section displays basic information like DB identifier, CPU usage, status, and engine. Below it, the 'Connectivity & security' tab is selected, showing details about the endpoint, networking, and security groups. A red box highlights the 'VPC security groups' field, which contains 'default (sg-03dc)'.

From "inbound rules" rules select and click "Edit inbound rules"

The screenshot shows the AWS EC2 Security Groups console. It lists a single security group named 'sg-03dc' with the description 'default VPC security group'. Below the list, under the heading 'sg-03dc - default', the 'Inbound rules' tab is selected. A red box highlights the 'Edit inbound rules' button.

Once you are in edit menu of inbound rules click *add rule*, from the drop down menu of type section find "MSSQL" the port range will be automatically set to 1433 (which is default port

number of MSSQL, for source choose My IP, your browser will automatically identify your IP and will set it for you. Click save rules.

The screenshot shows the AWS EC2 console with the path: EC2 > Security Groups > 11 - default > Edit inbound rules. The title is "Edit inbound rules". A note says "Inbound rules control the incoming traffic that's allowed to reach the instance." Below is a table for "Inbound rules". A rule named "sgr-1" is listed with "All traffic" as the type, "All" for both protocol and port range, and "Custom" for source, which is set to "sg-". An "Add rule" button is highlighted with a red box. At the bottom are "Cancel", "Preview changes", and "Save rules" buttons.

The screenshot shows the AWS EC2 console with the path: EC2 > Security Groups > sg- > default > Edit inbound rules. The title is "Edit inbound rules". A note says "Inbound rules control the incoming traffic that's allowed to reach the instance." Below is a table for "Inbound rules". A dropdown menu for "Type" is open, showing options like SMTPS, IMAPS, POP3S, and MSSQL. The "MSSQL" option is highlighted with a red box. Other options like NFS, MySQL/Aurora, RDP, Redshift, PostgreSQL, Oracle-RDS, WinRM-HTTP, and Custom TCP are also listed. An "Add rule" button is at the bottom left. At the bottom are "Cancel", "Preview changes", and "Save rules" buttons.

Inbound rules

| Security group rule ID | Type        | Protocol | Port range | Source | Description - optional |
|------------------------|-------------|----------|------------|--------|------------------------|
| sgr-0                  | All traffic | All      | All        | Custom |                        |
| -                      | MSSQL       | TCP      | 1433       | Custom |                        |

Add rule

Save rules

Go to your RDS menu where all databases are listed.

Click on your instance and find its “*Endpoint*” and “*Port*” for connecting using Microsoft SQL Server Management Studio(MSSMS).

Amazon RDS

Connectivity & security

| Endpoint & port        | Networking                                  | Security   |
|------------------------|---|--|
| Endpoint<br>[REDACTED] | Availability Zone<br>us-west-1b             | VPC security groups<br>default (sg-00000000)                     |
| Port<br>1433           | VPC<br>vpc-00000000                         | Active   |
|                        | Subnet group<br>default-vpc-                | Publicly accessible<br>Yes                                       |
|                        | Subnets<br>subnet-00000000, subnet-00000001 | Certificate authority<br>rds-ca-2019                             |
|                        | Network type<br>IPv4                        | Certificate authority date<br>August 22, 2024, 22:08 (UTC+05:00) |

Security group rules (3)

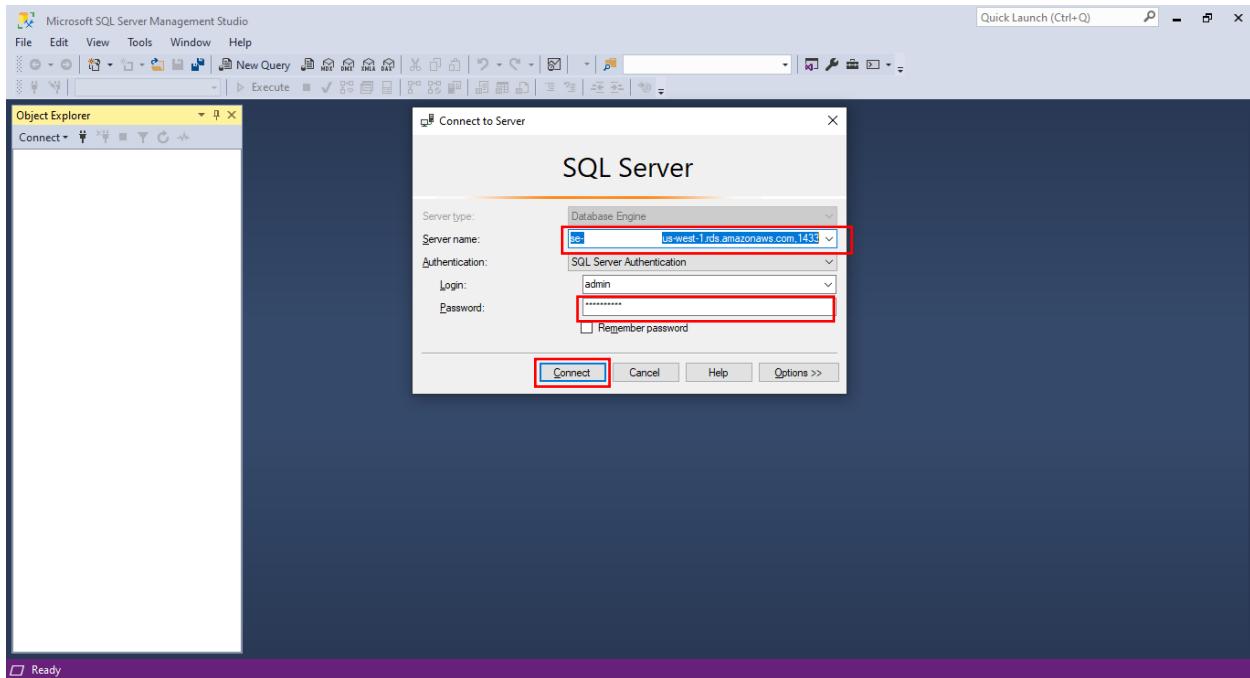
Replication (1)

Using concatenate them using comma, and write it to Server name in MSSMS like below.

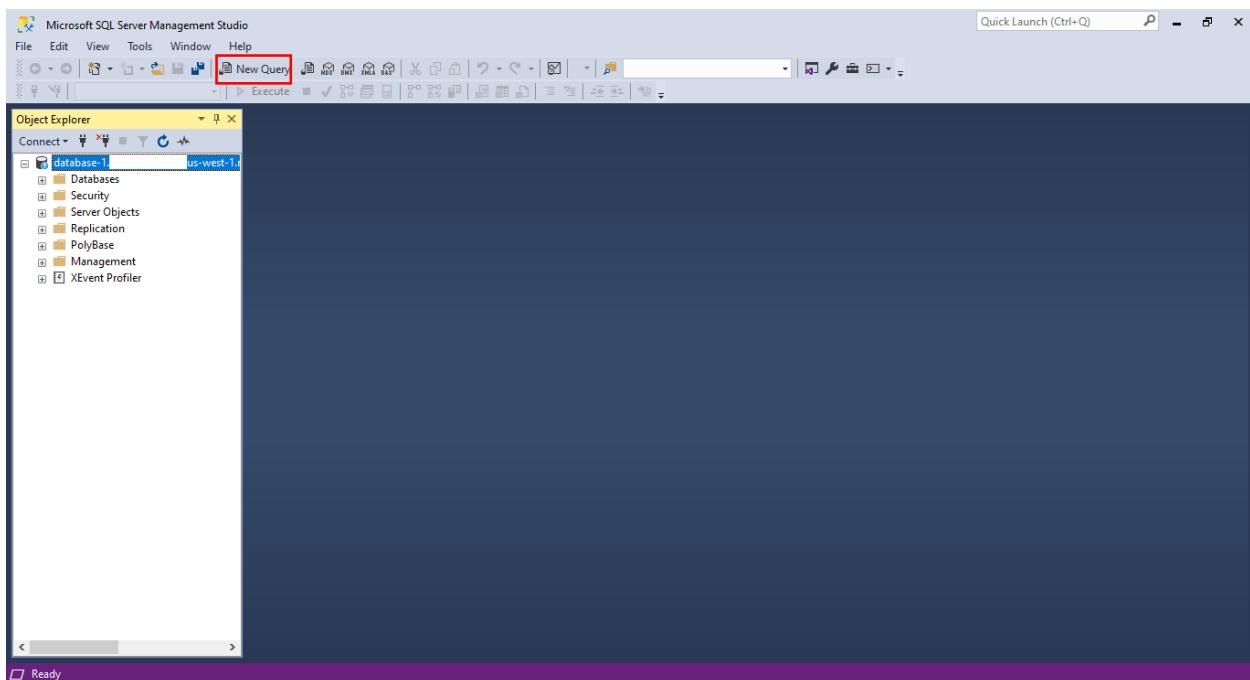
database-1.thisisfakeurl.us-west-1.rds.amazonaws.com,1433.

If you did change your login info whilst creating your instance everything you should give that credentials MSSMS on connection prompt

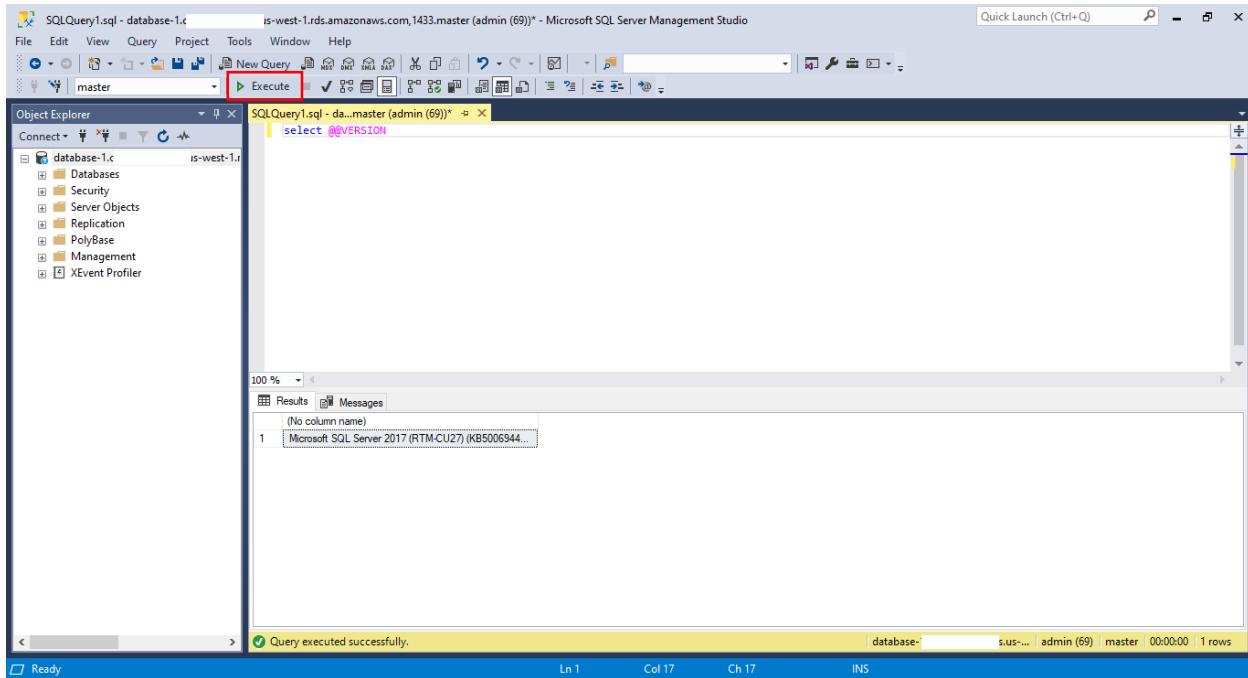
To the password section give your own password and click connect.



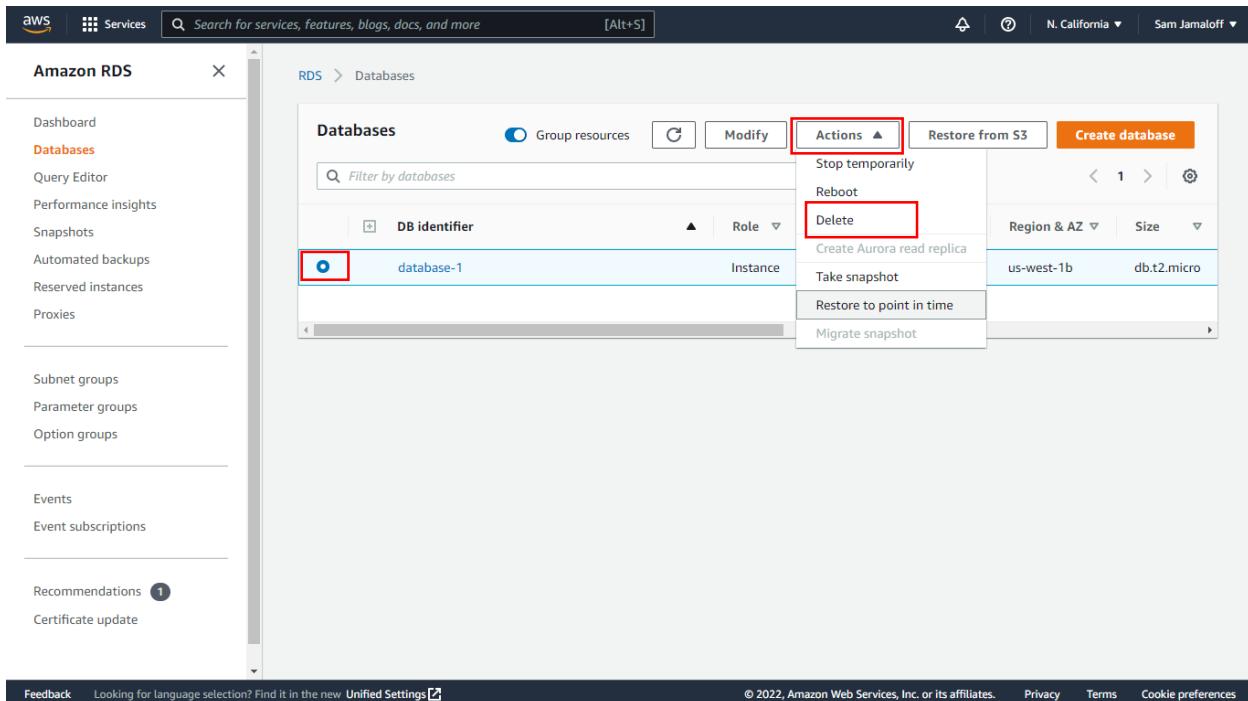
Connected. Now you can run any query by clicking New Query button'



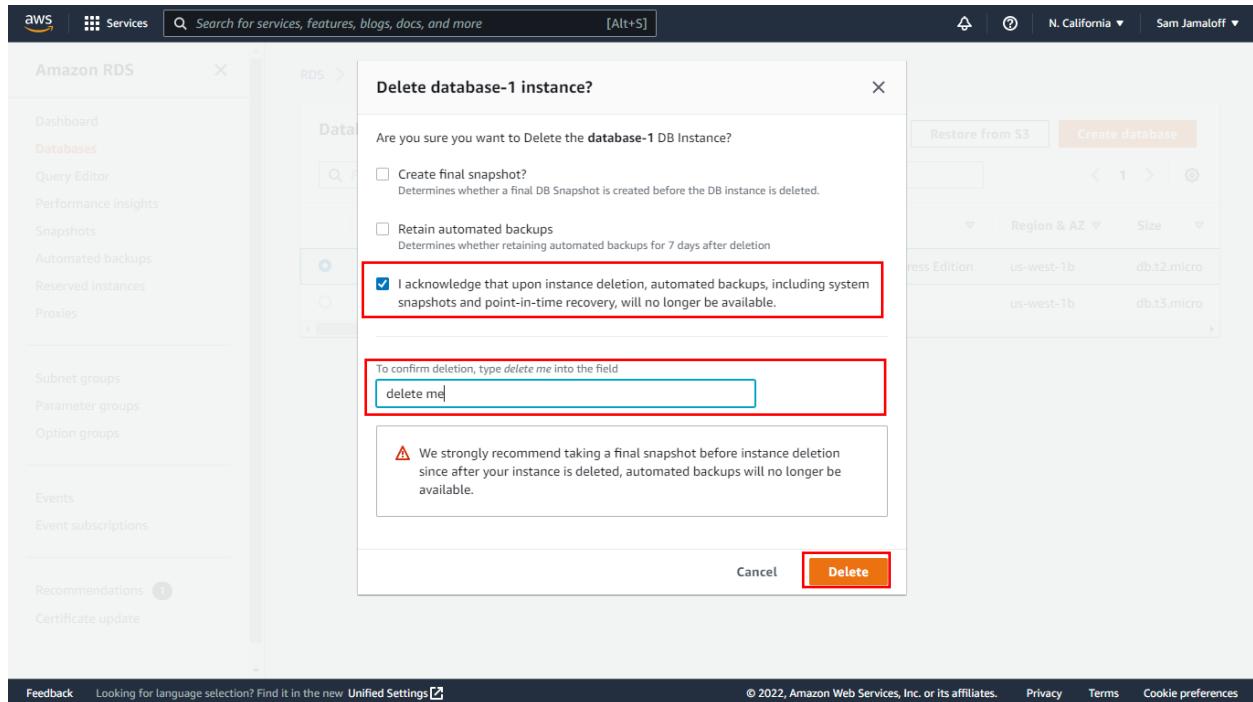
Write your query and execute it by clicking into execute button.



You can play with it as much as you would like and once you are done, go to your RDS list select the instance that you would like to terminate, and from action drop-down menu choose delete and click on that.



Now unselect default options and select “*I acknowledge ...*” and to confirm type *delete me*. After clicking delete button the instance will be deleted.



Feedback

Looking for language selection? Find it in the new [Unified Settings](#).

© 2022, Amazon Web Services, Inc. or its affiliates.

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

# Creating and connecting to PostgreSQL on amazon RDS. (using pgAdmin)

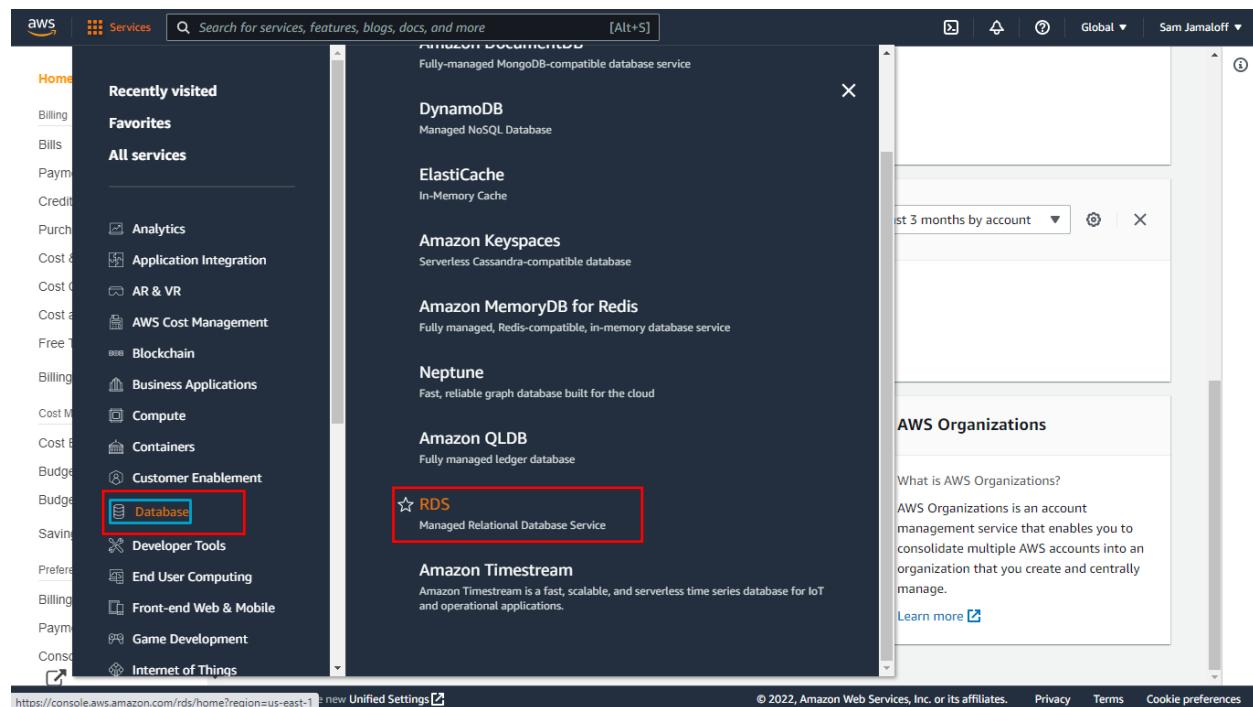
In this task, you will learn how to create an environment to run your PostgreSQL database. Several steps are very similar to those you did for MSSQL database.

**Step 1:** Enter the RDS Console When you [click here](#), the AWS management console will open in a new browser window, so you can keep this step-by-step guide open. When this screen loads, find RDS under Database and click to open the Amazon RDS Console.

## Step 2: Create a PostgreSQL DB Instance

In this step, we will use Amazon RDS to create a PostgreSQL DB Instance with db.t3.micro DB instance class, 20 GB of storage, and automated backups enabled with a retention period of one day. As a reminder, all of this is free tier eligible.

- a. In the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance.
  - Note: AWS Cloud resources are housed in highly available data center facilities in different areas of the world. You have the ability to choose which Region to host your Amazon RDS activity in.



- b. In the Create database section, choose Create database.

Screenshot of the AWS Amazon RDS console showing the main dashboard and a "Create database" modal.

The left sidebar shows navigation links for Databases, Query Editor, Performance insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, Recommendations (0), and Certificate update.

The main content area displays resource counts: DB Instances (0/40), DB Clusters (0/40), Reserved instances (0/40), Snapshots (0), Recent events (0), and Event subscriptions (0/20). It also lists Parameter groups (0), Option groups (0), and Subnet groups (0/50).

A "Create database" modal is open, containing the text: "Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud." It features two buttons: "Restore from S3" and a prominent orange "Create database" button, which is highlighted with a red box.

The "Service health" section indicates "View service health dashboard".

The right sidebar contains promotional cards for AWS Backup, Time-Series Tables in PostgreSQL, Test Your DR Strategy in Minutes, Implementing Cross-Region DR, and Additional information (Getting started with RDS, Overview and features, Documentation, Articles and tutorials).

- c. You now have options to select your engine. For this tutorial, choose the PostgreSQL icon

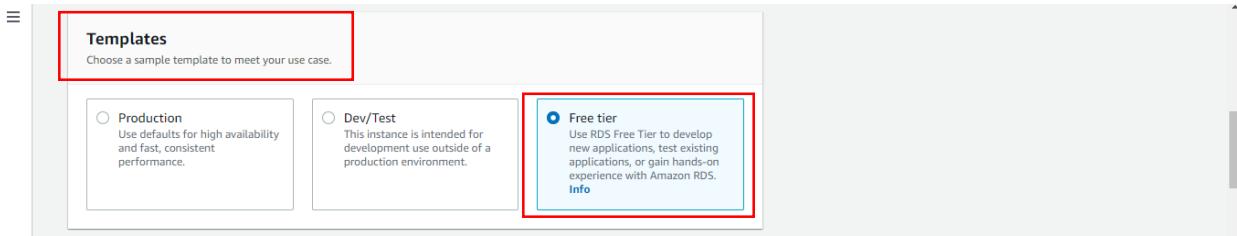
Screenshot of the "Choose a database creation method" step in the RDS creation wizard.

The "Standard create" option is selected and highlighted with a red box. The "Easy create" option is also shown.

The "Engine options" section allows selecting the engine type. "PostgreSQL" is selected and highlighted with a red box. Other options include Amazon Aurora, MySQL, MariaDB, Oracle, and Microsoft SQL Server.

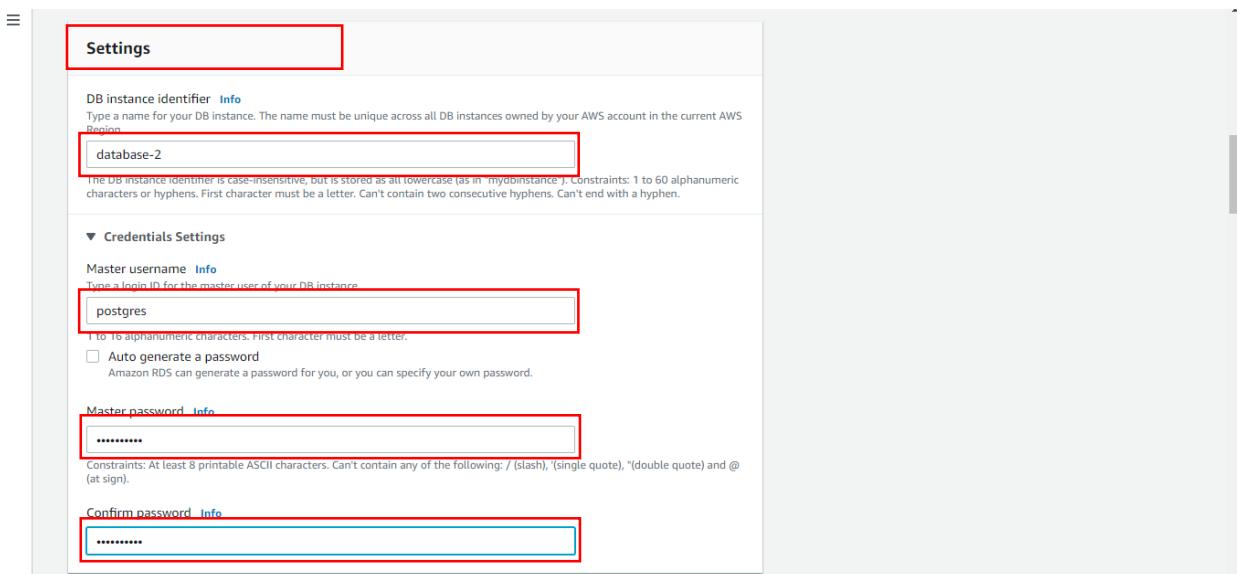
The "Version" dropdown is set to "PostgreSQL 13.7-R1".

- d. Be sure to choose free tier option from the list.

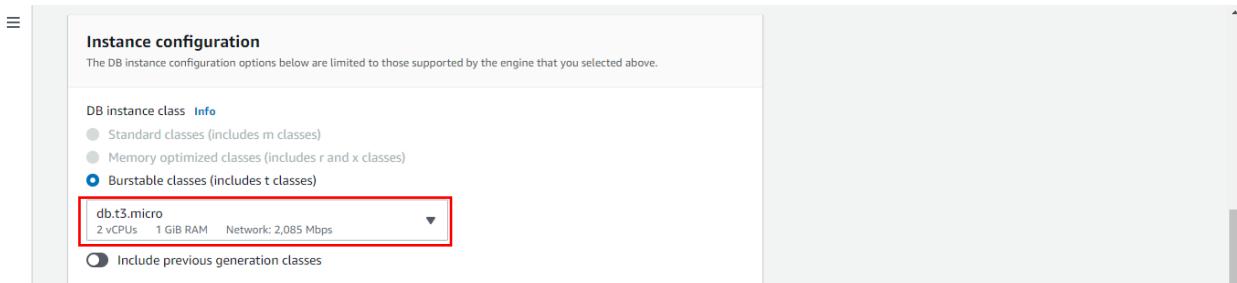


- e. You will now configure your DB instance. The list below shows the example settings that you can use for this tutorial:

- Settings:
  - i. DB instance identifier: You can rename it or leave it as it is.
  - ii. Master username: You can rename it or leave it as it.
  - iii. Master password: Should be provided to access from outside network or any SQL clients that your local machines has



- Instance configuration:
  - i. Leave everything as it is, the other options are not suitable with free tier.
  - ii. To see a list of supported instance classes, see [Amazon RDS Product Details](#).



- Storage
  - i. Storage type: General Purpose SSD (gp2)
  - ii. Allocated storage: (20 GiB)
  - iii. Storage autoscaling: Enabled

**Storage**

Storage type [Info](#)  
General Purpose SSD (gp2)  
Baseline performance determined by volume size

Allocated storage  
20 GiB  
(Minimum: 20 GiB. Maximum: 6,144 GiB) Higher allocated storage can improve IOPS performance.

Storage autoscaling [Info](#)  
Provides dynamic scaling support for your database's storage based on your application's needs.  
 Enable storage autoscaling  
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold [Info](#)  
Charges will apply when your database autoscales to the specified threshold  
1000 GiB  
Minimum: 22 GiB. Maximum: 6,144 GiB

- Connectivity: (Must be DONE)
  - i. Public access: Choose yes, in order to be able to access to our instance from outside world.
  - ii. VPC security group: Create new VPC and give it a name

**Connectivity** [Info](#)

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) [Info](#)  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-03dc732091d8d3cf3)  
Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB Subnet group [Info](#)  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-03dc732091d8d3cf3

**Public access** [Info](#)

Yes  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

New VPC security group name

**Availability Zone** [Info](#)

**▼ Additional configuration**

**Database port** [Info](#)

TCP/IP port that the database will use for application connections.

- Database authentication: Leave everything as it is.

**Database authentication**

**Database authentication options** [Info](#)

Password authentication  
Authenticates using database passwords.

Password and IAM database authentication  
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication  
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

- Monitoring:
  - i. Additional configuration: Change initial database name.

**Monitoring**

**Performance Insights** [Info](#)

Turn on Performance Insights [Info](#)

Retention period [Info](#)

AWS KMS key [Info](#)

Account  
414634816993

KMS key ID  
alias/aws/rds

**⚠ You can't change the KMS key after enabling Performance Insights.**

**► Additional configuration**

Enhanced Monitoring

**Additional configuration**  
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#)  
myDb  
If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)  
default.postgres13

Option group [Info](#)  
default:postgres-13

Backup

Enable automated backups  
Creates a point-in-time snapshot of your database

- Create Database button: Click it

Maintenance window [Info](#)  
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.  
 Choose a window  
 No preference

Deletion protection  
 Enable deletion protection  
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

**Estimated monthly costs**

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application usage exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

**Important** You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

[Cancel](#) **Create Database**

f. Your DB Instance is now being created. Click View Your DB Instances.

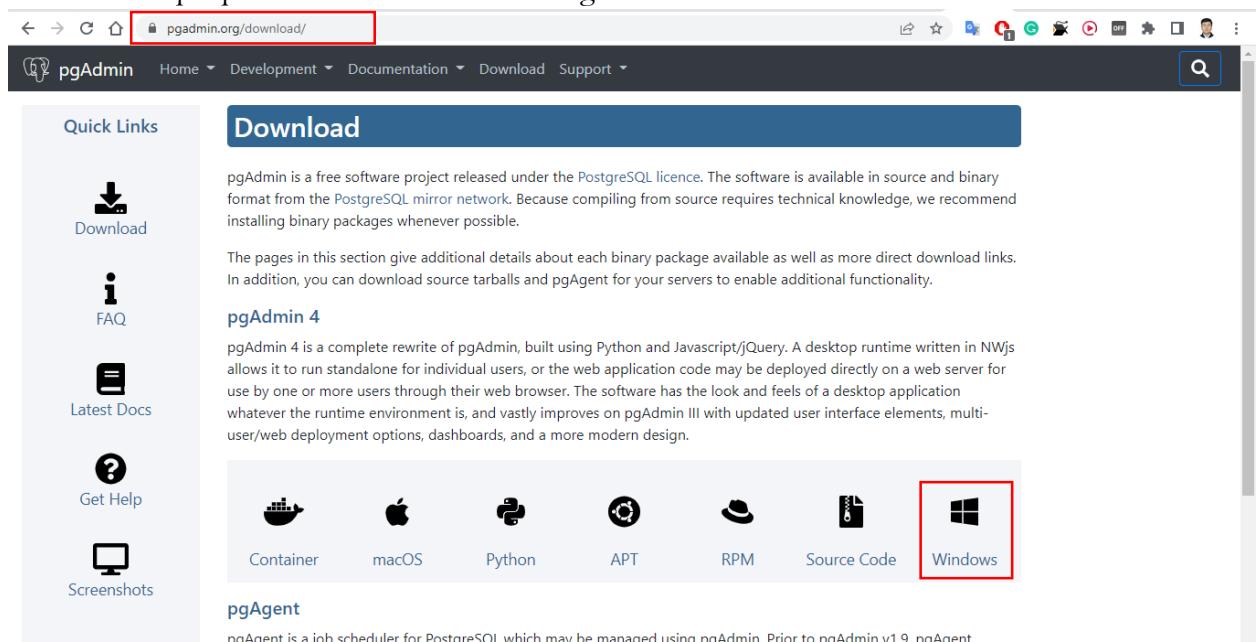
- Note: Depending on the DB instance class and storage allocated, it could take several minutes for the new DB instance to become available. The new DB instance appears in the list of DB instances on the RDS console. The DB instance will have a status of creating until the DB instance is created and ready for use. When the state changes to available, you can connect to a database on the DB instance.
- Feel free to move on to the next step as you wait for the DB instance to become available.

## Step 3: Download a SQL Client

After the database instance creation is complete and the status changes to available, you can connect to a database on the DB instance using any standard SQL client. In this step, we will download pgAdmin a popular PostgreSQL client.

Note: Remember to run PostgreSQL client from the same device on the same network from which you created the DB Instance. The security group your database is placed in is configured to allow connection only from the device from which you created the DB instance. If you try to connect from a different network or device, your IP address would have changed. Your database can be configured to be accessed from any IP address, but for this tutorial we will keep it simple.

- a. Go to [pgAdmin website](https://pgadmin.org/download/), and download a software that is compatible with your OS. For this seminar purpose we will be downloading for windows.



The screenshot shows a web browser displaying the pgAdmin download page at [pgadmin.org/download/](https://pgadmin.org/download/). The page has a dark header with the pgAdmin logo and navigation links for Home, Development, Documentation, Download, and Support. Below the header is a sidebar titled "Quick Links" with icons for Download, FAQ, Latest Docs, Get Help, and Screenshots. The main content area has a blue header "Download". It contains text about pgAdmin being a free software project released under the PostgreSQL licence, and a note that binary packages are recommended over source compilation. It also mentions pgAdmin 4 as a rewrite built with Python and JavaScript/JQuery, running in NW.js. Below this is a row of download links for Container, macOS, Python, APT, RPM, Source Code, and Windows. The "Windows" link is highlighted with a red box. At the bottom, there is a section for pgAgent, which is described as a job scheduler for PostgreSQL managed via pgAdmin.

- b. Download and install latest version.

**pgAdmin 4 (Windows)**

**Download**

**Maintainer:** pgAdmin Development Team

pgAdmin is available for 64 bit Windows™ 7 SP1 (desktop) or 2008R2 (server) and above, up to v4.30.

v5.0 and later are supported on Windows 8 (desktop) or 2012 (server) and above.

32 bit Windows support is available for versions up to v4.29.

The packages below include both the Desktop Runtime and Web Application:

- pgAdmin 4 v6.14 (released Sept. 22, 2022)**
- [pgAdmin 4 v6.13 \(released Aug. 25, 2022\)](#)
- [pgAdmin 4 v6.12 \(released July 28, 2022\)](#)
- [pgAdmin 4 v6.11 \(released June 30, 2022\)](#)
- [pgAdmin 4 v6.10 \(released June 2, 2022\)](#)
- [pgAdmin 4 v6.9 \(released May 12, 2022\)](#)
- [pgAdmin 4 v6.8 \(released April 7, 2022\)](#)
- [pgAdmin 4 v6.7 \(released March 14, 2022\)](#)

## Step 4: Connecting:

Once you have installed search for pgAdmin from your local machine and run (open) it.

All Apps Documents Web More

Best match

**pgAdmin 4 v6** App

Search the web

- pgAdmin 4 v6 - See web results
- pgadmin 4 v6 download
- pgadmin 4 v6.0
- pgadmin 4 v6.2

pgAdmin 4 v6 App

- Open**
- Run as administrator
- Open file location
- Pin to Start
- Pin to taskbar
- Uninstall

On the dashboard you will see **Add New Server**, click on that.

- **Register – Server** pop-up window will appear
- In the **General** section, there is a **Name**, you can give any name of your preference.
- In the connection tab you need to give your instance information.
  - i. **Hostname:** is the same as the **Endpoint** of our instance.

- ii. **Port:** 5432
- iii. **Maintenance database:** postgres
- iv. **Username:** your instance {master username} that can be found from configuration tab of instance, if not set it is “postgres” by default
- v. **Password:** {your instance password}

- Click Save button to establish connection.

The screenshot shows the AWS VPC Endpoint configuration page under the 'Connectivity & security' tab. It displays the following details:

| Endpoint & port      | Networking  | Security   |
|----------------------|---|--|
| Endpoint: [REDACTED] | Availability Zone: us-west-1b<br>VPC: vpc-<br>Subnet group: default-vpc-03dc732091d8d3cf3<br>Subnets: subnet-<br>Network type: IPv4 | VPC security groups: mySecurityGroup (sg-)<br>Status: Active<br>Publicly accessible: Yes<br>Certificate authority: rds-ca-2019<br>Certificate authority date: August 22, 2024, 22:08 (UTC+05:00) |

**Security group rules (2)**

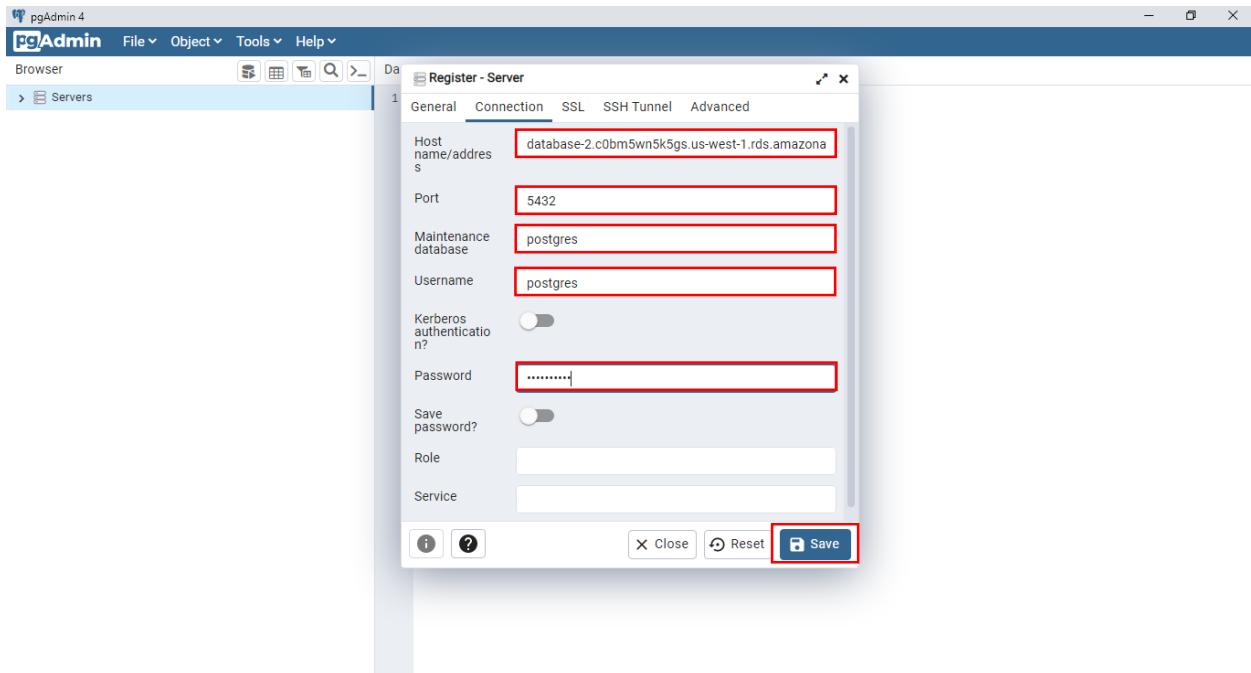
**Replication (1)**

The screenshot shows the pgAdmin 4 interface with the 'Register - Server' dialog box open. The dialog box has the following fields:

- Name: Any Name (highlighted with a red box)
- Server group: Servers
- Background: (button)
- Foreground: (button)
- Connect now? (switch)
- Comments: (text area)

A validation error message is displayed at the bottom: "Either Host name, Address or Service must be specified." Below the dialog are buttons for Close, Reset, and Save.

The pgAdmin interface includes a sidebar with 'Servers (1)', a toolbar with various icons, and a footer with links to Planet PostgreSQL and Community Support.



Congratulations. You have successfully connected to your PostgreSQL Database instance that is in AWS.

You can play with it as much you would like and after you have done, **DO NOT FORGET** to delete your instance.

## Step 5: Deleting your instance

Now unselect default options and select “I acknowledge …” and to confirm type *delete me*. After clicking delete button the instance will be deleted.

The screenshot shows the AWS RDS Databases page. A database named "database-2" is selected (indicated by a red box). In the Actions menu (also highlighted with a red box), the "Delete" option is selected. A confirmation dialog box titled "Delete database-2 instance?" is displayed. Inside the dialog, there are three checkboxes: "Create final snapshot?", "Retain automated backups?", and "I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available." The third checkbox is checked and highlighted with a red box. Below the checkboxes, a text input field contains the text "delete me". At the bottom of the dialog, there is a warning message: "⚠ We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available." The "Delete" button at the bottom right of the dialog is also highlighted with a red box.

## Seminar 3

# Microservices using ASP.NET Core

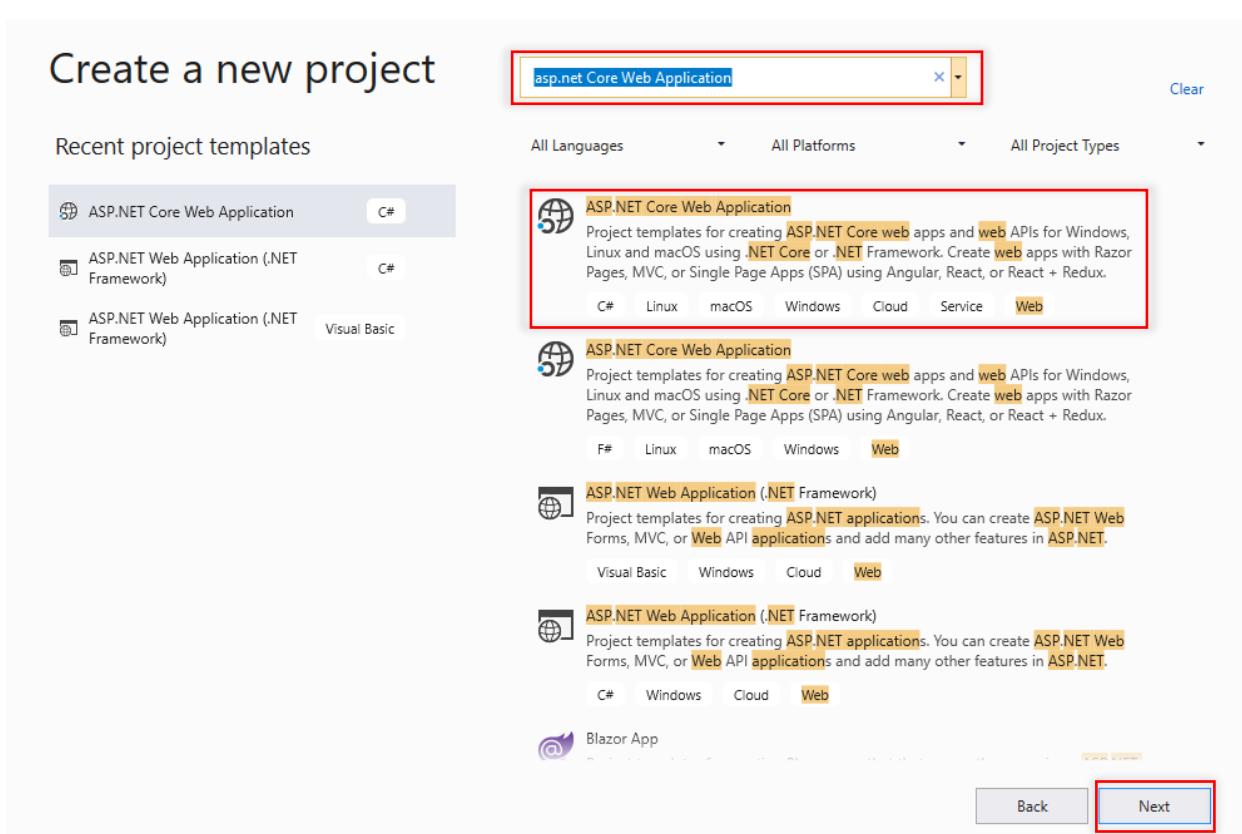
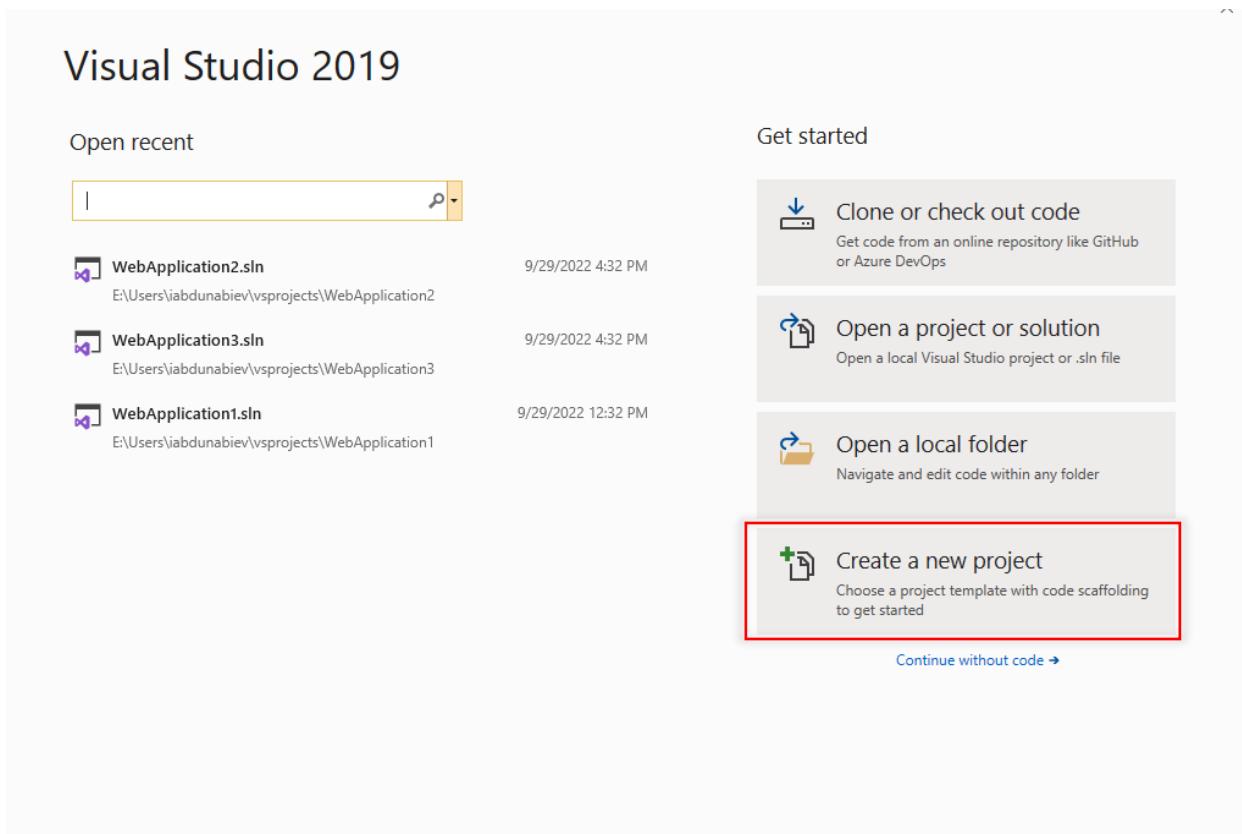
The term Micro-services portrays a software development. Micro-services are more about applying certain number of principles and architectural patterns to software development. Each micro-service lives independently, but on the other hand, also all rely on each other. All micro-services in a project get deployed in production at their own pace, on-premise or in the cloud, independently, living side by side. In this seminar, you will learn how to create a simple CRUD micro-service-based API using ASP.NET Core.

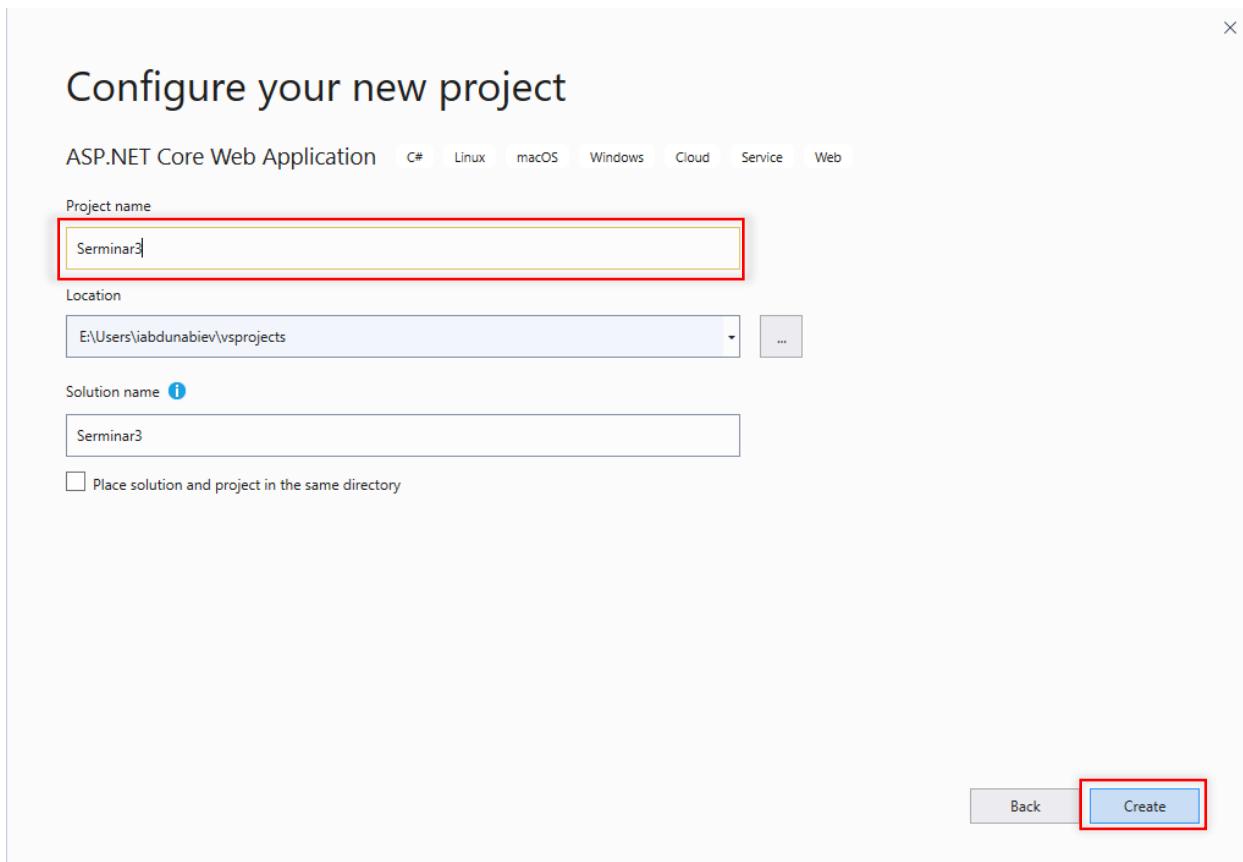
## **Step 1: Create a Microsoft SQL Server DB Instance.**

In this step, we will use Amazon RDS to create a Microsoft SQL Server DB Instance with db.t2.microDB instance class, 20 GB of storage. This step is similar to the task in the previous seminar, thus, please follow the necessary instructions from that seminar. (For this you can refer to Seminar 2, creating and connecting to MSSQL in Amazon RDS part)

## **Step 2: CRUD Micro-service using ASP.NET Core**

This part will demonstrate how to create a Product micro-service using ASP.NET Core step by step. Creating an Asp.NET Core Application Solution 1. Open the Visual Studio and add a new ASP.NET Core Web Application and give it a meaningful name.





## Create a new ASP.NET Core web application

The screenshot shows the 'Create a new ASP.NET Core web application' dialog. At the top, there are dropdown menus for '.NET Core' (set to 'ASP.NET Core 3.0') and 'Authentication' (set to 'No Authentication').  
**Project Templates:**

- Empty**: An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
- API**: A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers. This option is highlighted with a red box.
- Web Application**: A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.
- Web Application (Model-View-Controller)**: A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
- Angular**: A project template for creating an ASP.NET Core application with Angular.
- React.js**: A project template for creating an ASP.NET Core application with React.js.

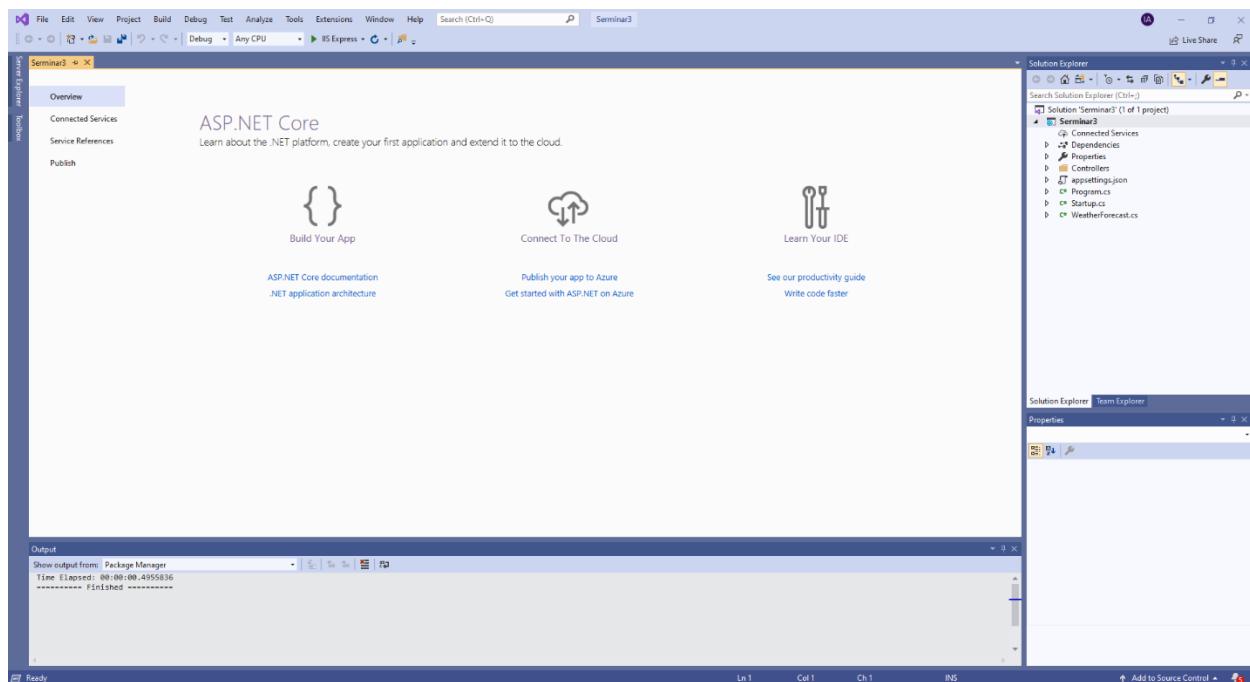
**Advanced Options:**

- Configure for HTTPS
- Enable Docker Support  
(Requires Docker Desktop)

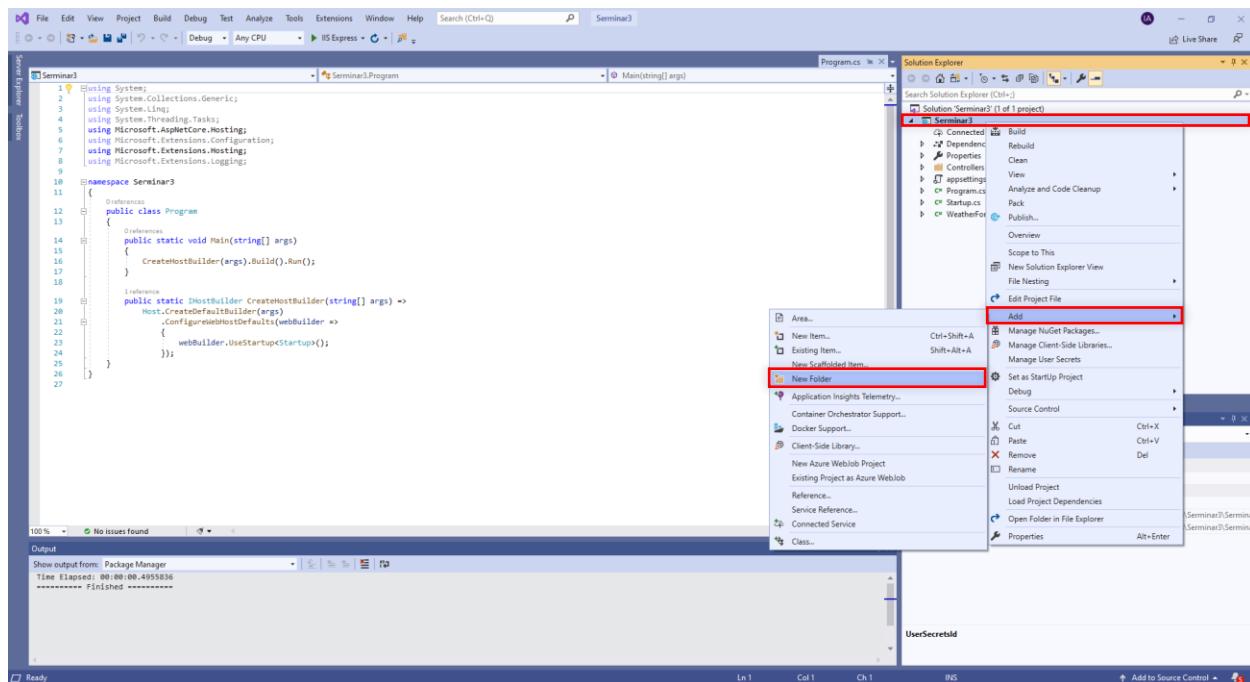
Platform: Linux

Author: Microsoft  
Source: .NET Core 3.0.0

At the bottom, there are 'Back' and 'Create' buttons, with 'Create' being highlighted by a red box.



Now create a folder called Model inside `{your_project_name}`, like it was shown in picture below.



Screenshot of Visual Studio showing the Solution Explorer with a red box around the 'Model' folder. The 'Model' folder contains 'appsettings.json', 'Program.cs', 'Startup.cs', and 'WeatherForecast.cs'. The code editor shows Program.cs with the following code:

```

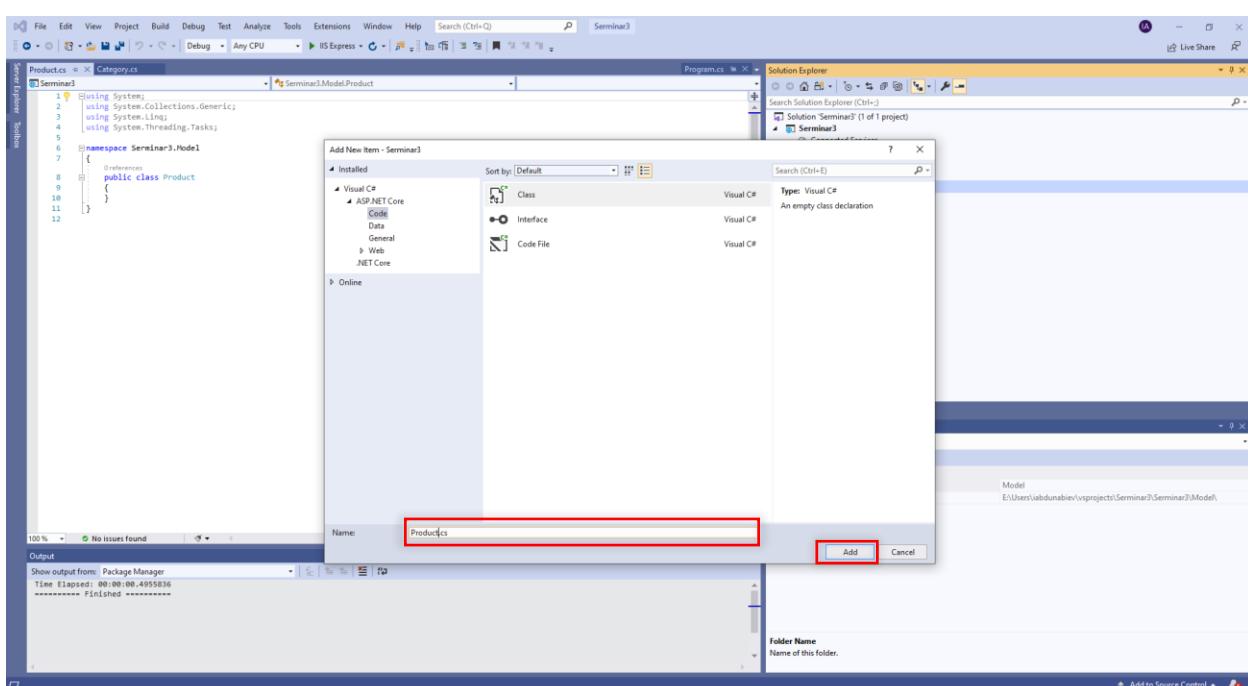
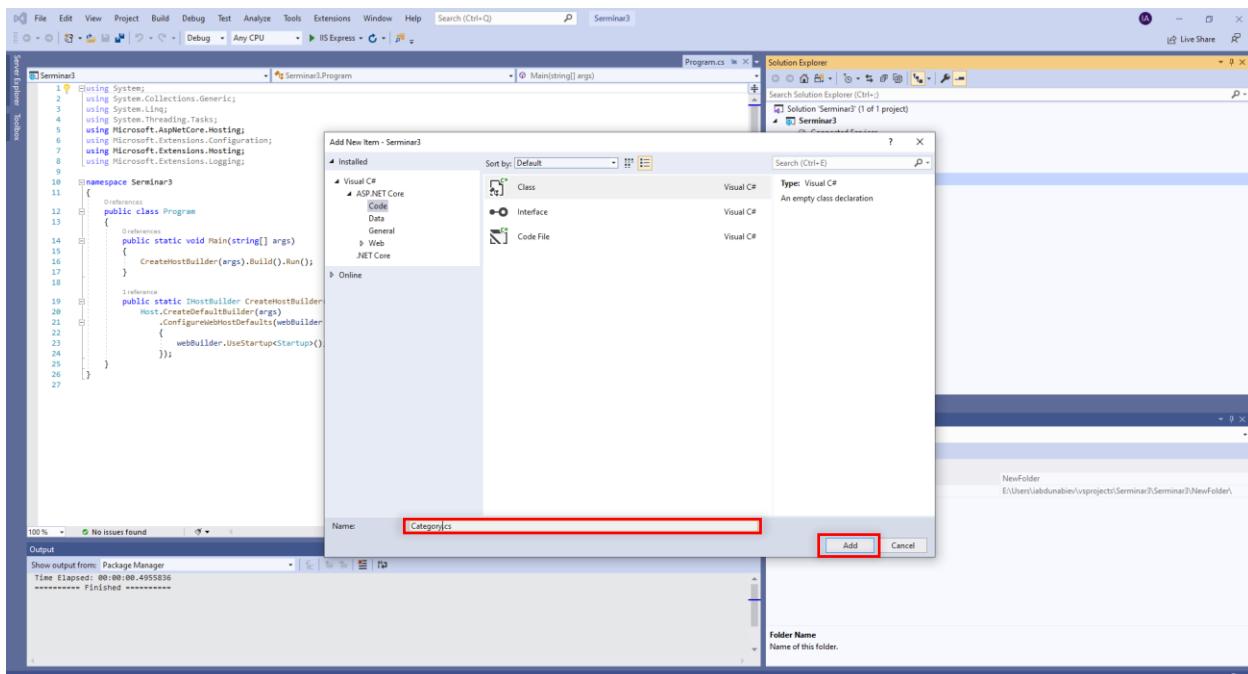
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using Microsoft.AspNetCore.Hosting;
6  using Microsoft.Extensions.Configuration;
7  using Microsoft.Extensions.Hosting;
8  using Microsoft.Extensions.Logging;
9
10 namespace Seminar3
11 {
12     public class Program
13     {
14         public static void Main(string[] args)
15         {
16             CreateHostBuilder(args).Build().Run();
17         }
18
19         public static IHostBuilder CreateHostBuilder(string[] args) =>
20             Host.CreateDefaultBuilder(args)
21                 .ConfigureWebHostDefaults(webBuilder =>
22                     {
23                         webBuilder.UseStartup<Startup>();
24                     });
25     }
26 }

```

The Output window shows "No issues found".

Inside model folder create classes, *Category* and *Product*.

Screenshot of Visual Studio showing a context menu open over the 'Model' folder in the Solution Explorer. A red box highlights the 'Add' option in the context menu. The context menu includes options like 'Controller...', 'New Item...', 'Existing Item...', 'New Scaffolded Item...', 'New Folder', 'Application Insights Telemetry...', 'Container Orchestrator Support...', 'Docker Support...', 'Client-Side Library...', 'New Azure WebApp Project', 'Existing Project as Azure WebApp', and 'Class...'. The code editor shows Program.cs with the same code as before.



Add new properties: ID, Name, Description, to the Category class.

```

133 % No Issues found
Output
Show output from: Package Manager
Time Elapsed: 00:00:00.4955836
***** Finished *****

```

Add new properties: ID, Name, Description, Price and ProductCategory to the Product class.  
Do not forget to save changes that you have done to the files.

```

133 % No Issues found
Output
Show output from: Package Manager
Time Elapsed: 00:00:00.4955836
***** Finished *****

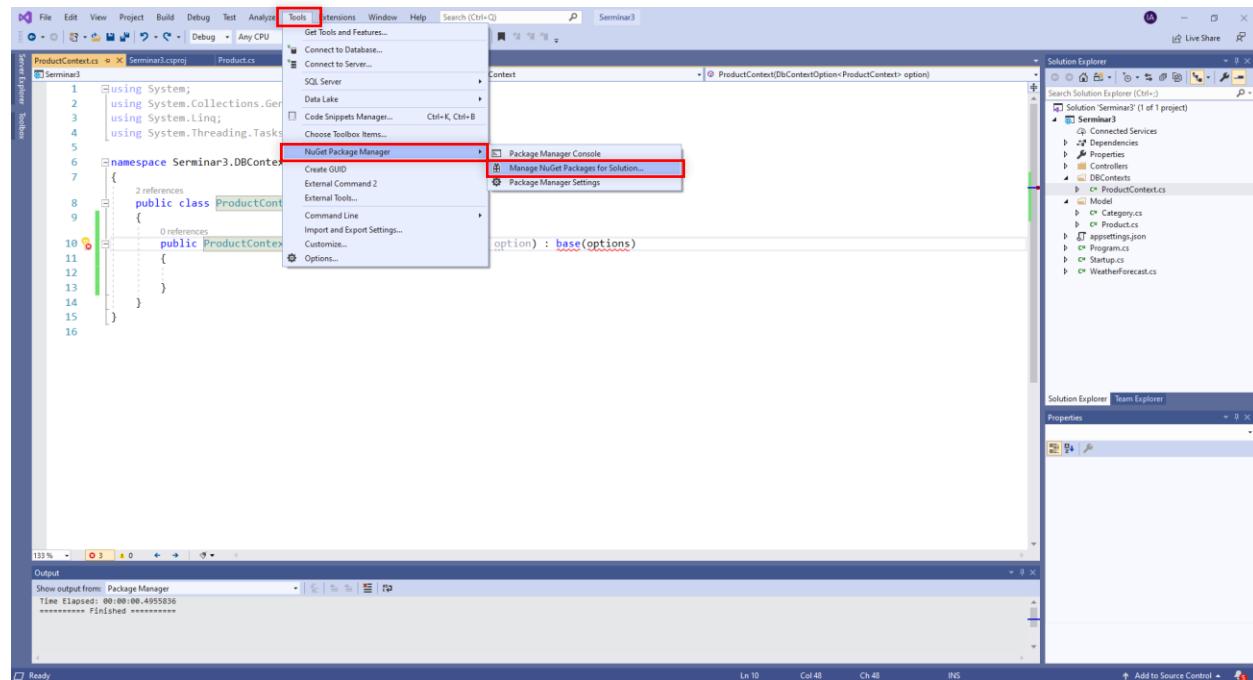
```

## Enabling Entity Framework (EF) Core

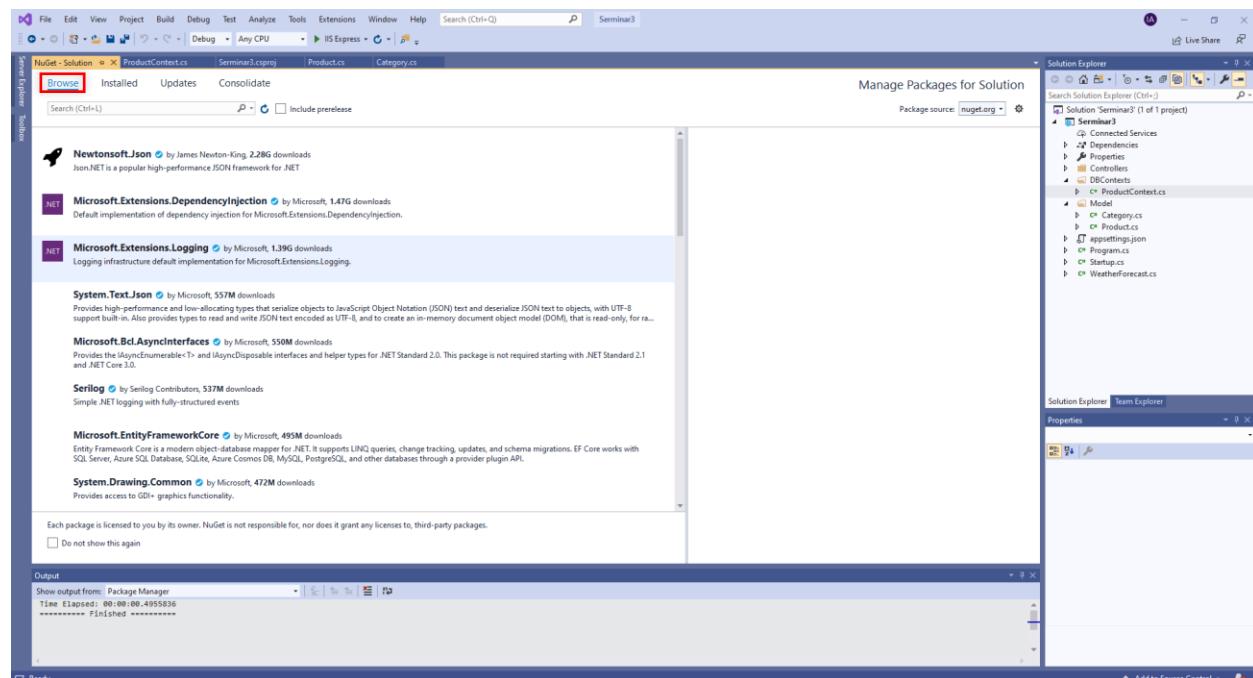
.NET Core API Project has built support for EF Core and all the related dependencies should be downloaded at the time of project creation.

If it is not present it could be explicitly added to the project via NuGet Packages.

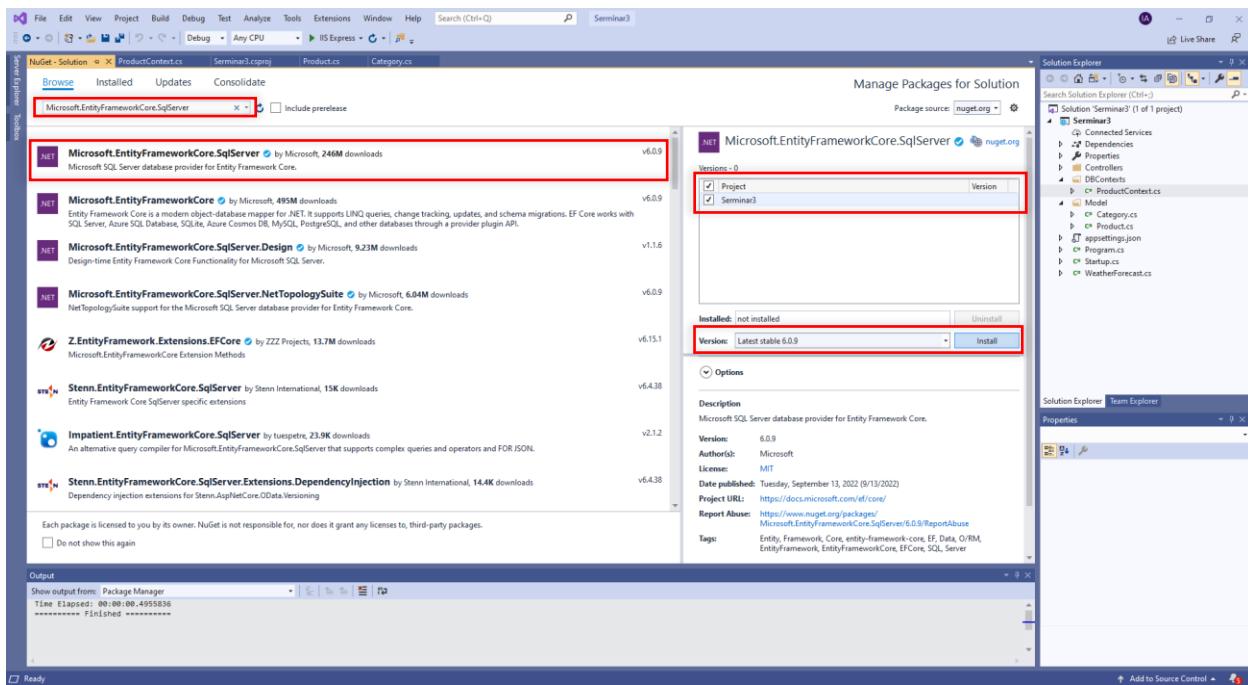
In your Visual Studio: Tools > NuGet Package Manager > Manage NuGet Package for Solution click on that



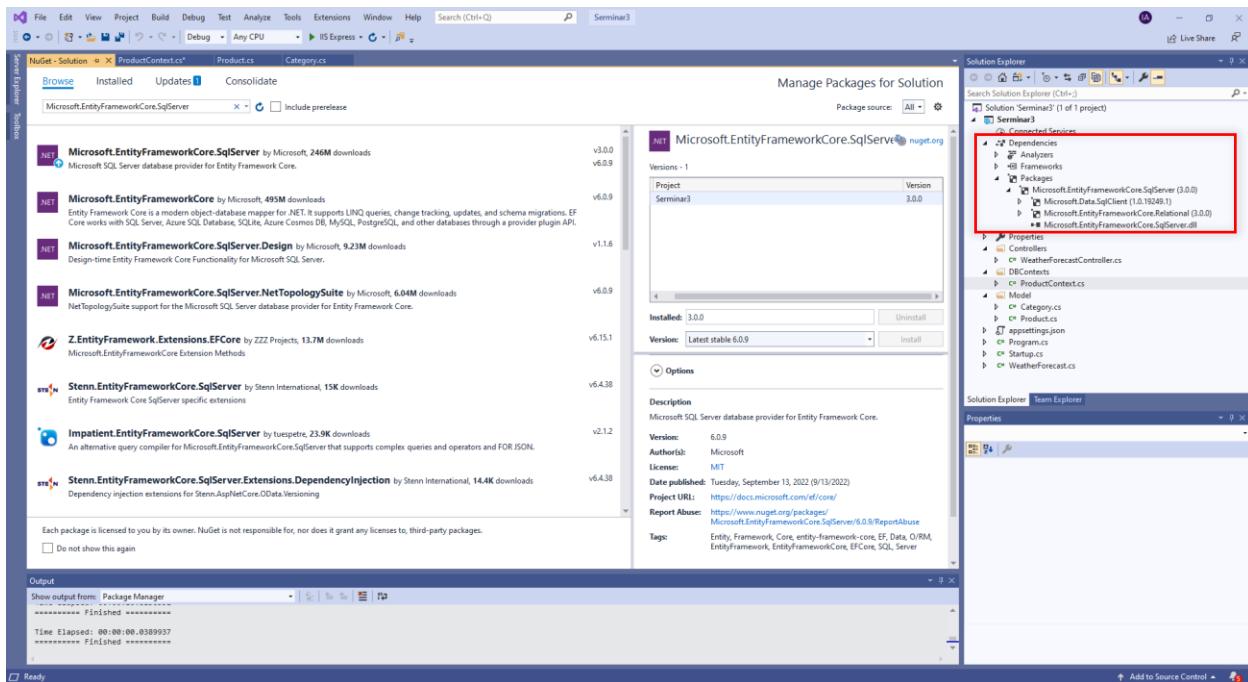
New window will show up and you need to click on **Browse**



Into the Search type Entity search for **Microsoft.EntityFrameworkCore.SqlServer**. Choose which project, choose the version that is compatible, and click install.



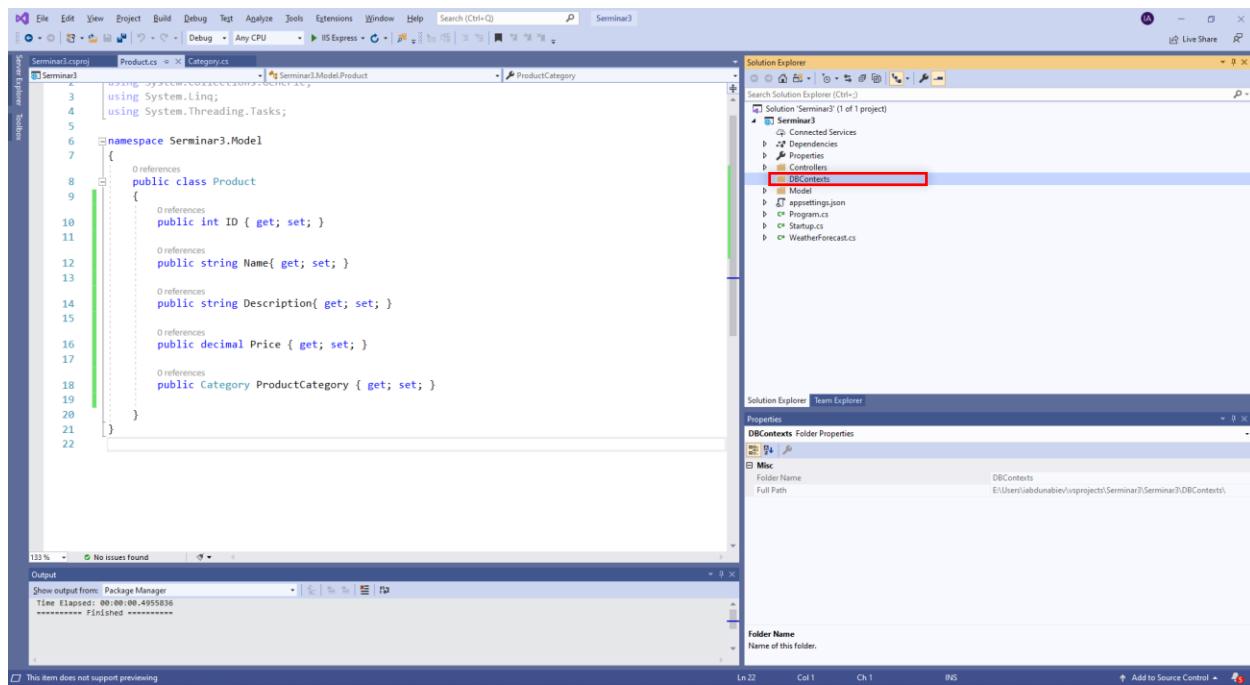
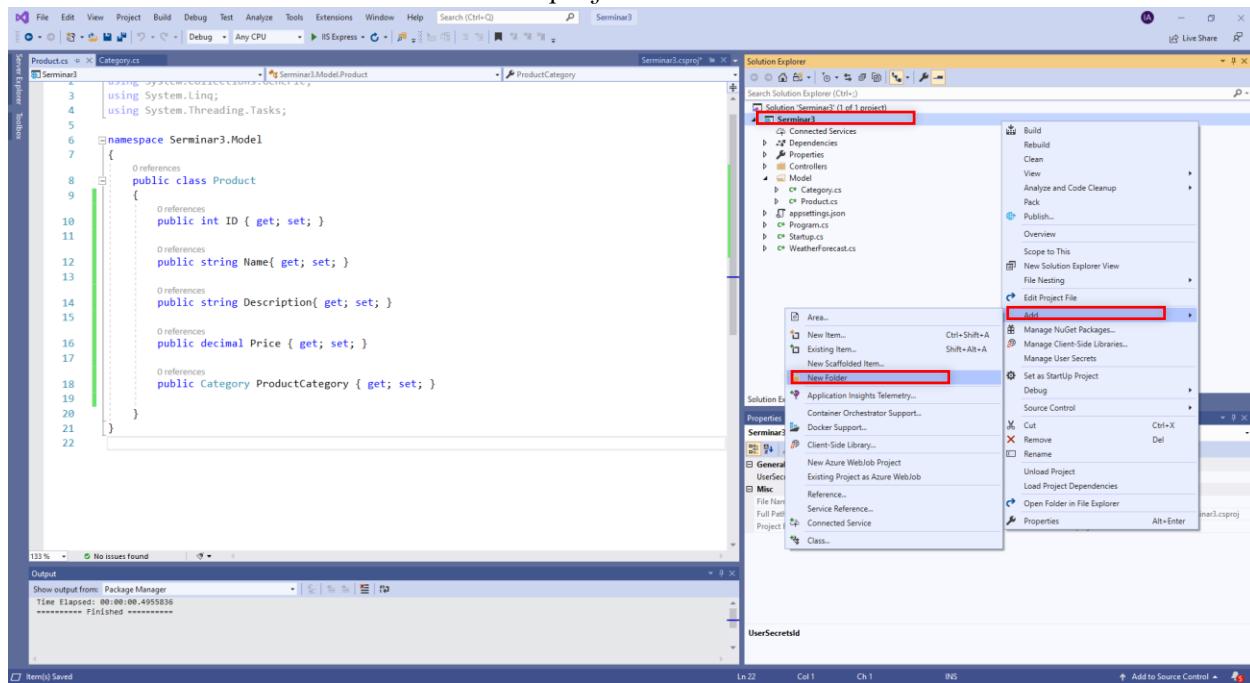
After installation you, you can check installed NuGet Entity Framework like shown in picture below.



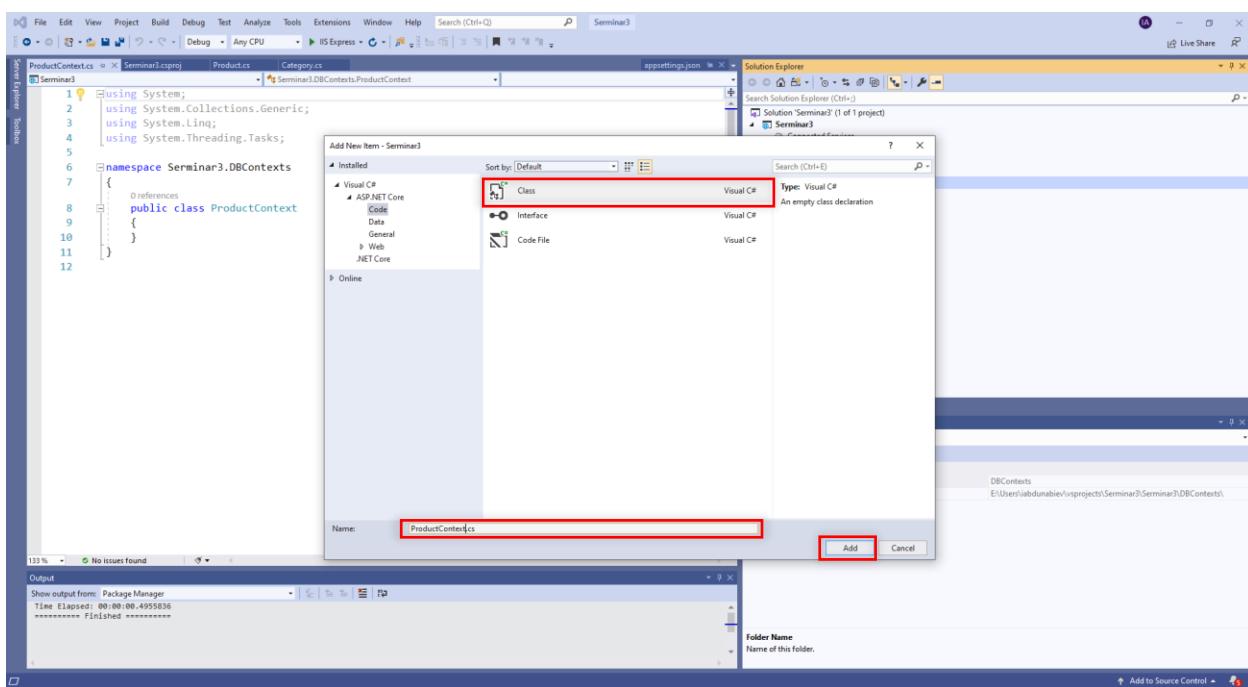
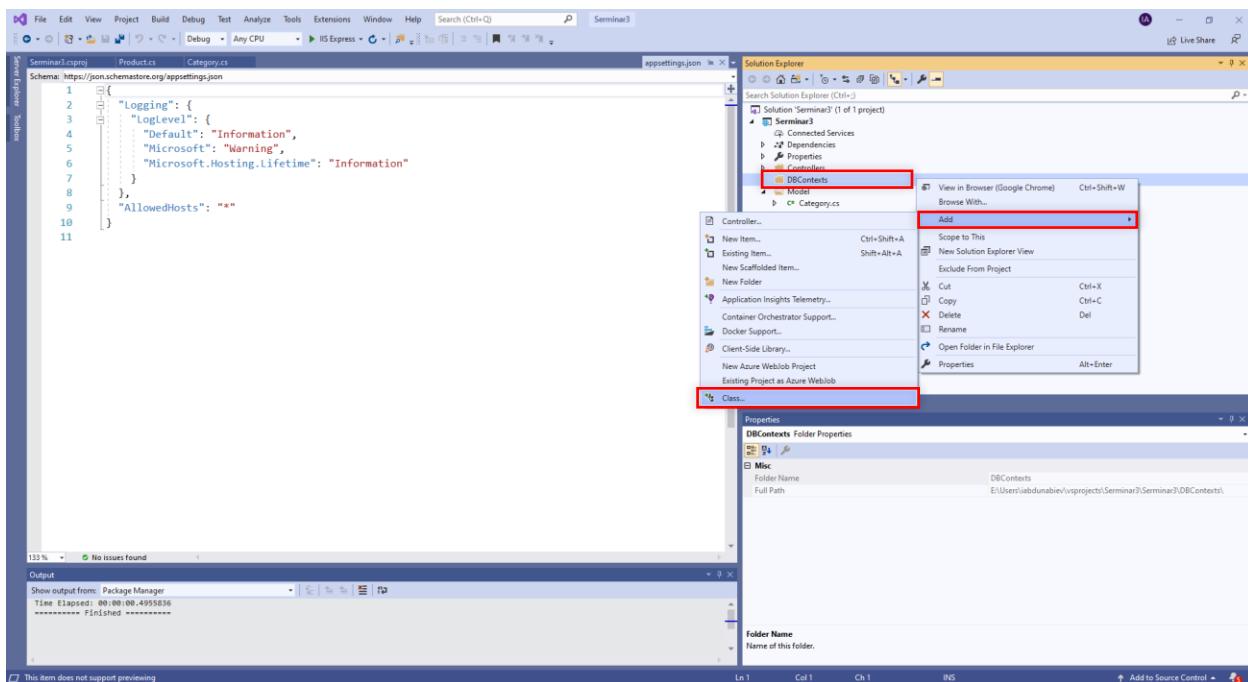
## Adding EF Core DbContext

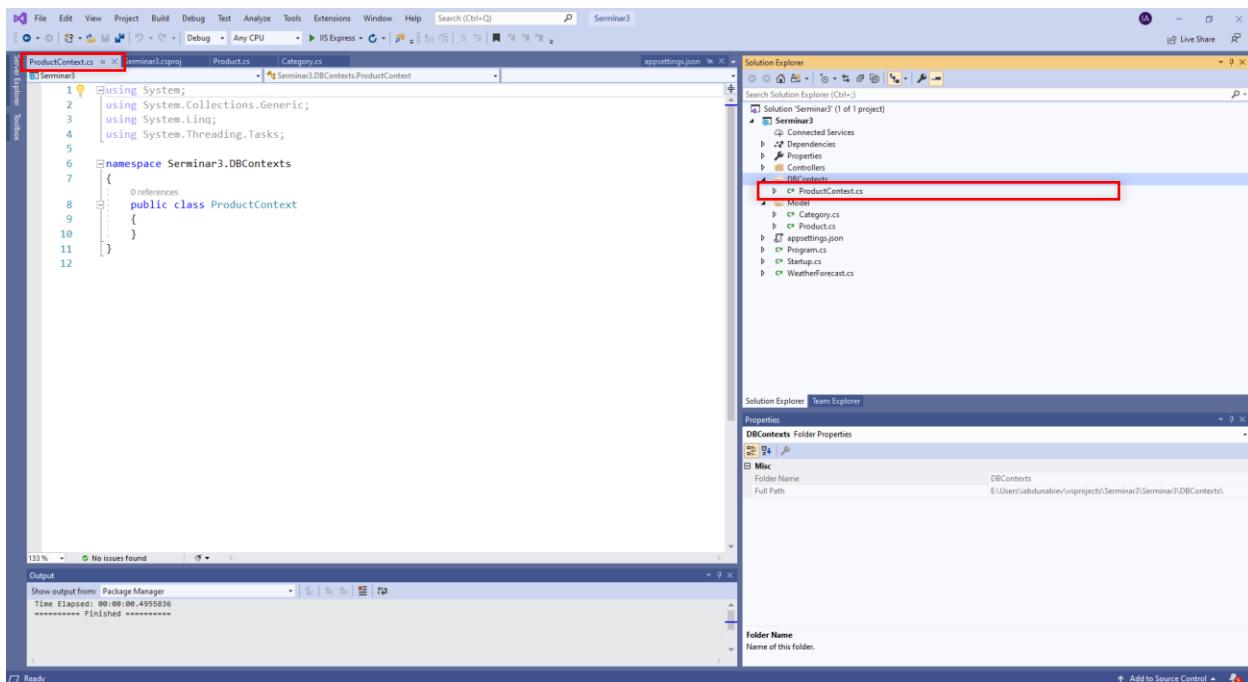
A database context is needed so that models could interact with the database.

1. Add new folder name DBContexts to the project.



2. Add a new class named ProductContext which includes the DbSet properties, for Product and Categories,



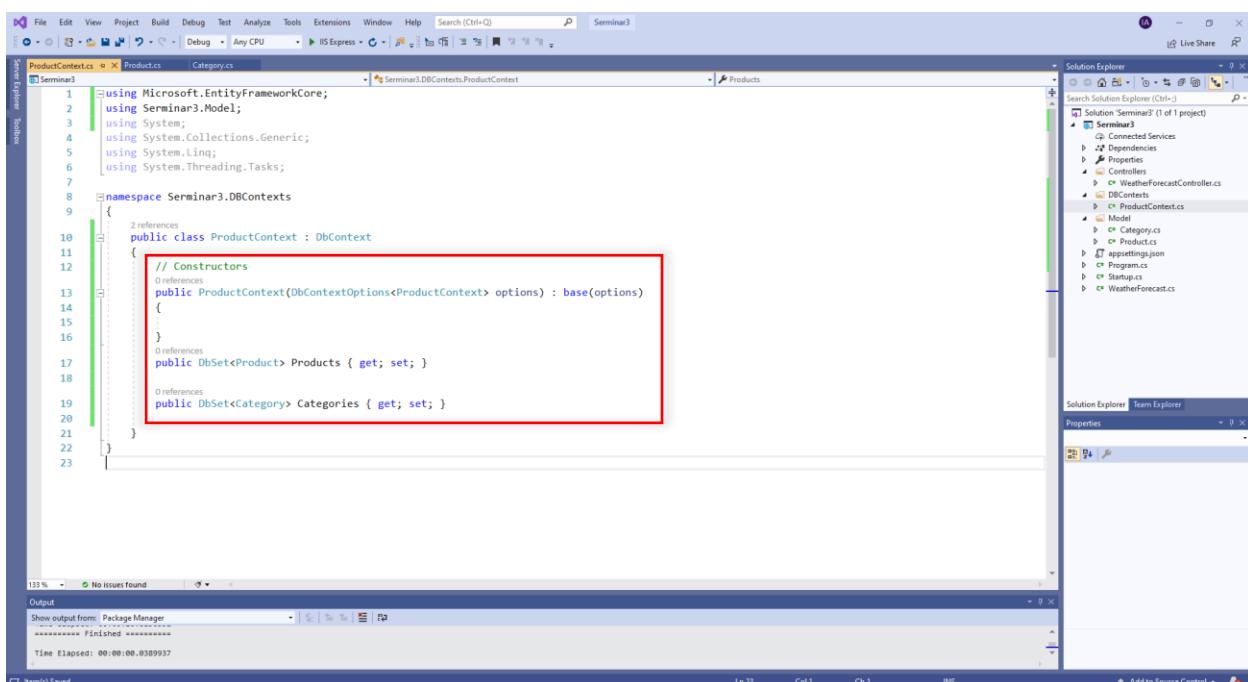


Create a Constructor and other following code into ProductContext file as shown in picture below.

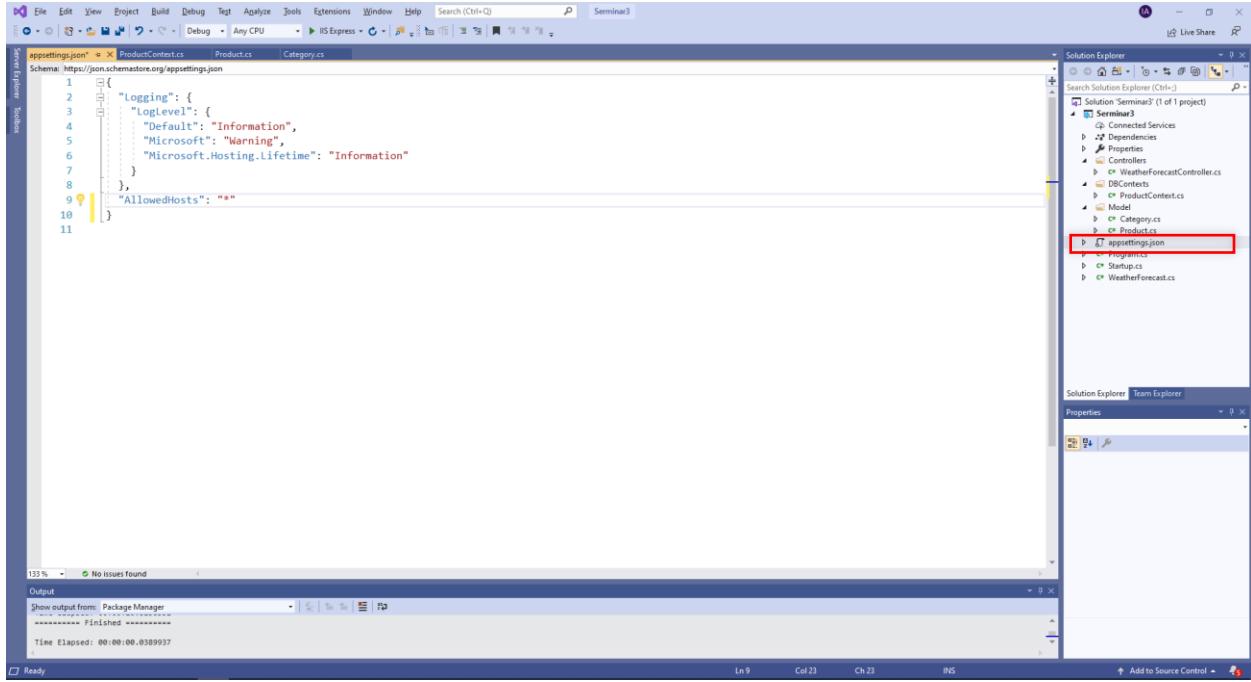
```
// Constructors DO NOT FORGET That Product Context class extends DB Context
public ProductContext(DbContextOptions<ProductContext> options) : base(options){}

public DbSet<Product> Products { get; set; }

public DbSet<Category> Categories { get; set; }
```



Now go to appsettings.json file and we will be editing connection string to our database that is in Amazon RDS. (Up till this window hopefully your database is already created, because we will be creating a connection to it)



To edit put comma after the last entry that you have, in my case that is "AllowedHosts": "\*". After comma copy and paste the connection string and

```
"ConnectionStrings": {
  "ProductDB": "Server=AWS-SERVER-ADDRESS;Database=Catalog;UserId=admin;password=1234567890"
}
```

You need to edit this to the entries of which is specific to your database instance in your Amazon RDS.

**Server** > name that can be found in endpoint of instance

**Database** > leave it as Catalog, we will be creating a database in our own machine with this name.

**User Id** > admin, or the name that you gave while creating

**Password** > password that was used while creating instance.

The screenshot shows the Visual Studio IDE interface. The main window displays the `appsettings.json` file, which contains configuration settings for a .NET application. A specific section of the JSON code is highlighted with a red box:

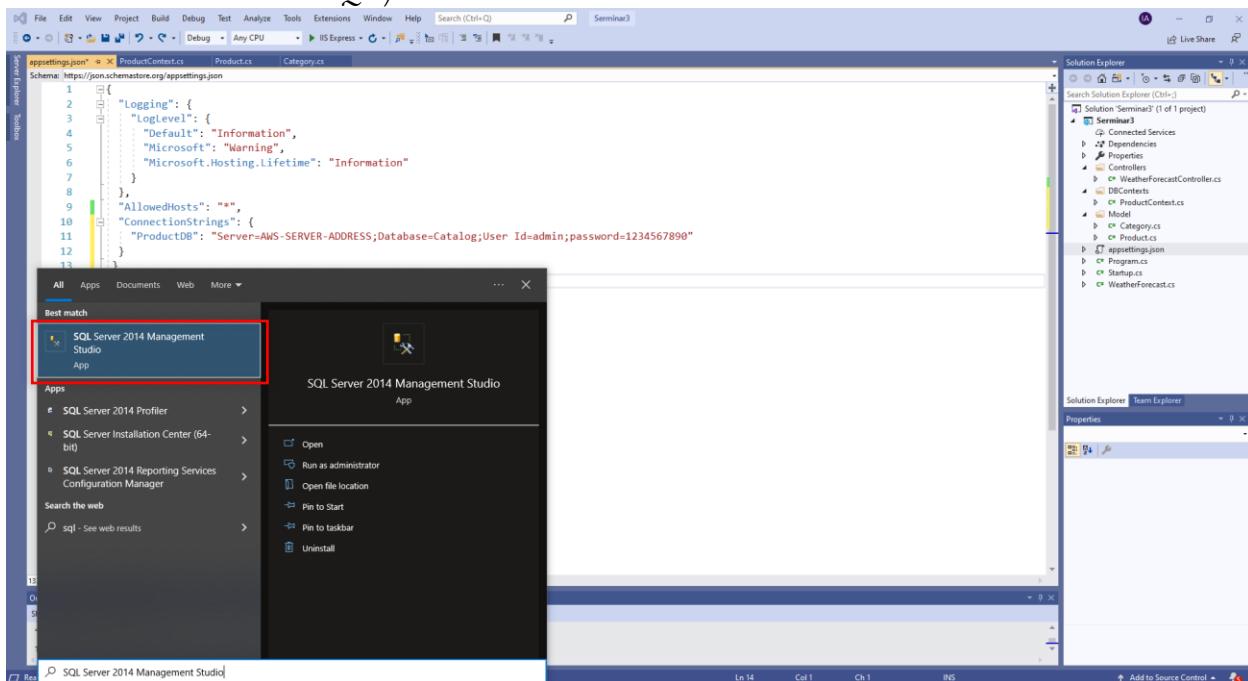
```

    "ConnectionStrings": {
        "ProductDB": "Server=AWS-SERVER-ADDRESS;Database=Catalog;User Id=admin;password=1234567890"
    }
  
```

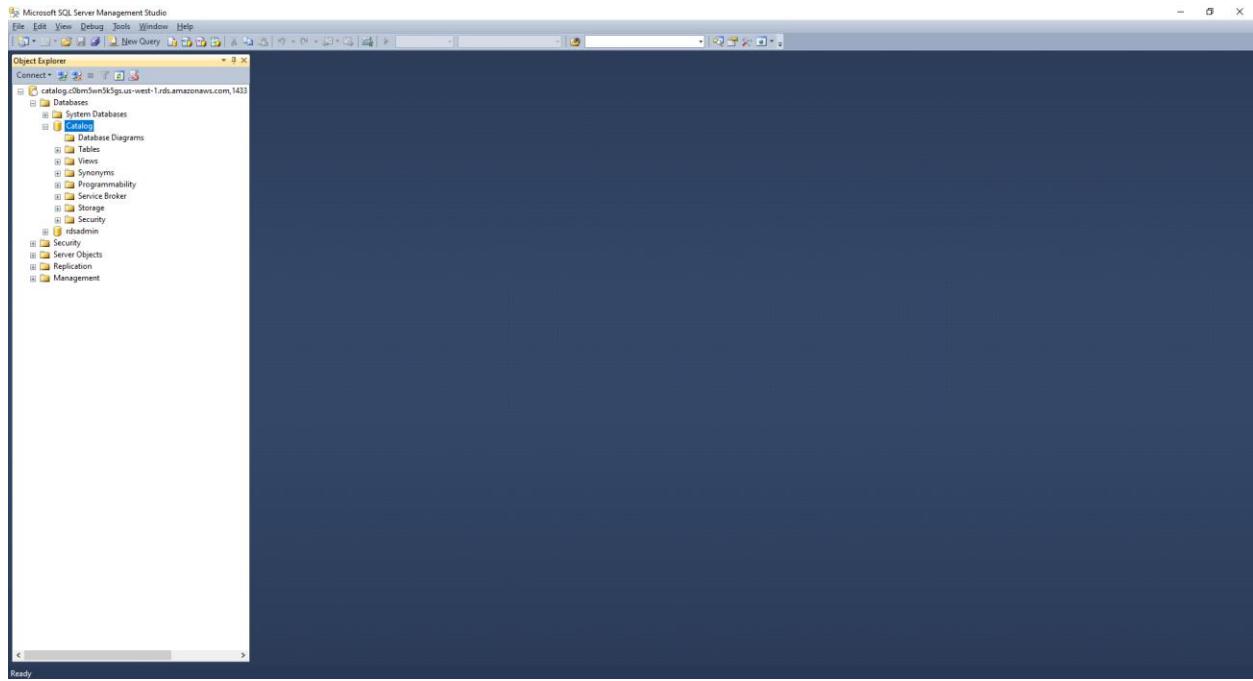
The Solution Explorer on the right shows the project structure for `Seminar3`, including files like `WeatherForecastController.cs`, `ProductContent.cs`, and `appsettings.json`. The Properties and Team Explorer tabs are also visible at the bottom.

Edit and save the file.

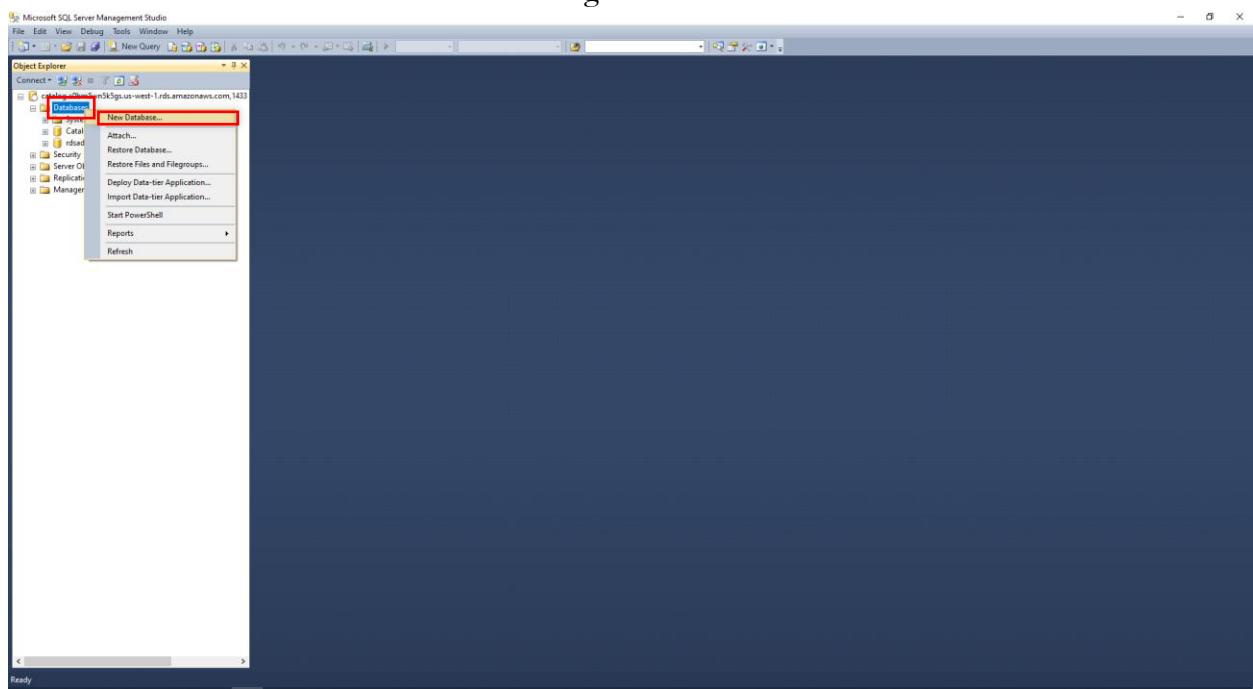
Check for the connection using MSSQL software. (Check Seminar 2 for how to use and connect to AWS RDS MicroSoftSQL)

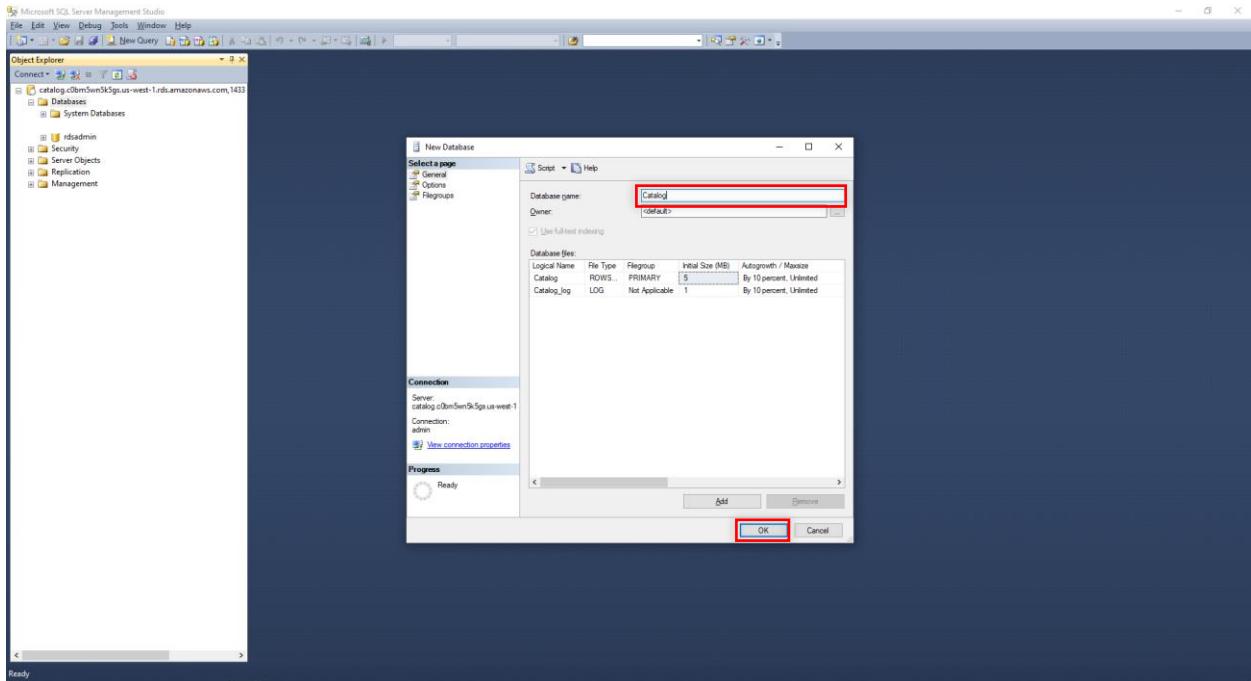


Window after connecting to the database.



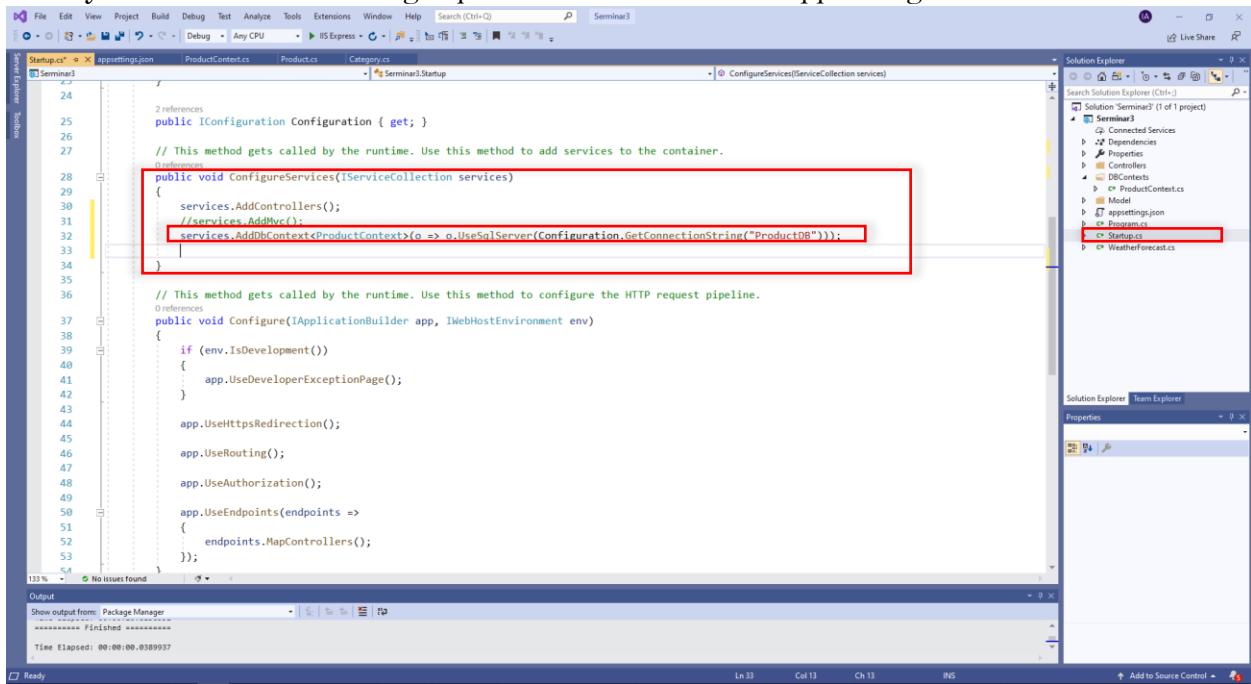
Create a new database with the name Catalog.





This will create a database in AWS

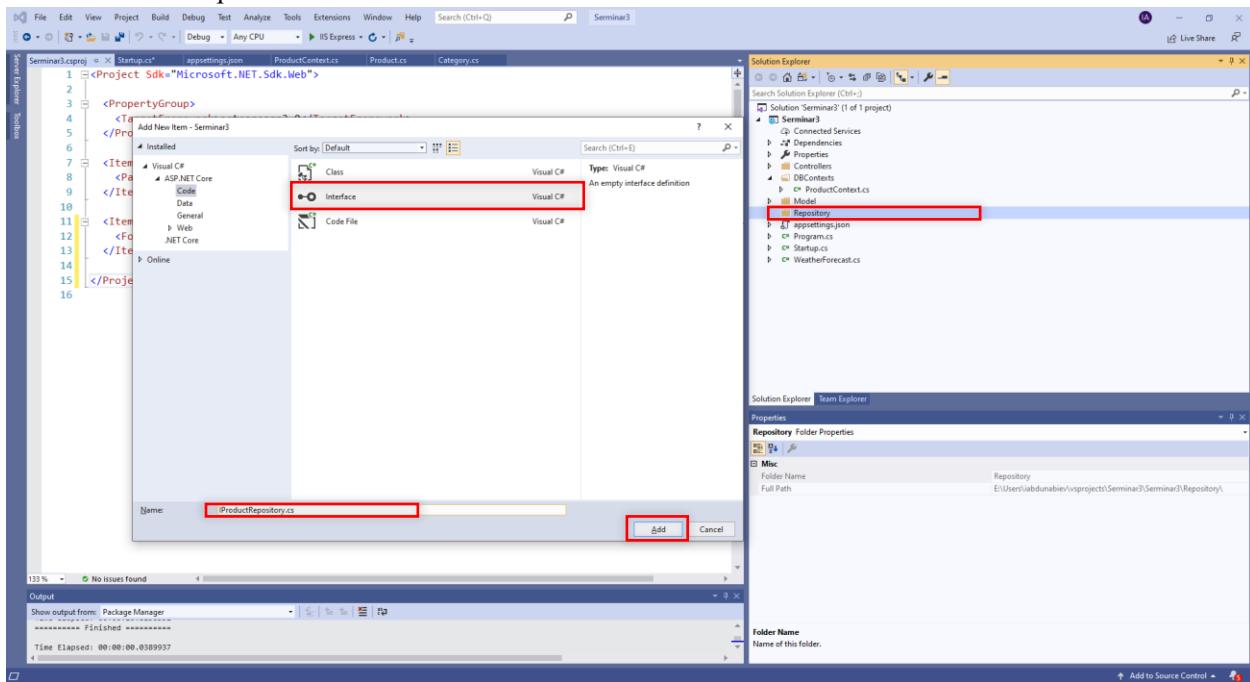
Open the Startup.cs file to add the SQL server db provider for EF Core. Add the code  
`services.AddDbContext<ProductContext>(o =>`  
`o.UseSqlServer(Configuration.GetConnectionString("ProductDB")));`  
 under ConfigureServices method. Note that in the GetConnectionString method the name of the key of the connection string is passed that was added in appsettings file.



## Adding Repository.

Repository works as a micro component of micro-service that encapsulates the data access layer and helps in data persistence and testability as well.

1. Add a new folder named Repository in the project and add an Interface name IProductRepository in that folder. Add the methods in the interface that performs CRUD operations for Product micro-service



Add following code to IProductRepository.cs file do not forget to make interface public (it is not public in the screen shot but if you don't do that it will not work

```
void InsertProduct(Product product);

void UpdateProduct(Product product);

void DeleteProduct(int productId);

Product GetProductById(int Id);

IEnumerable<Product> GetProducts();
```

```

using System.Threading.Tasks;
namespace Seminar3.Repository
{
    public interface IProductRepository
    {
        void InsertProduct(Product product);
        void UpdateProduct(Product product);
        void DeleteProduct(int productId);
        Product GetProductById(int Id);
        IEnumerable<Product> GetProducts();
    }
}

```

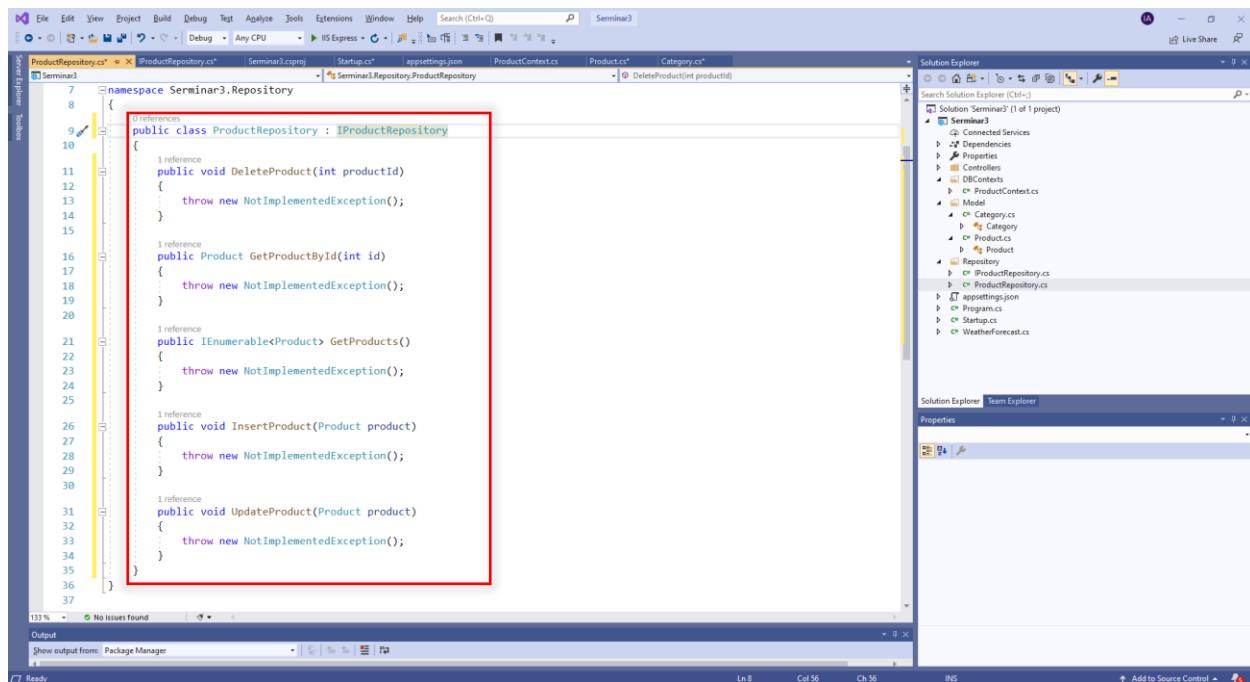
2. Add a new concrete class named ProductRepository in the same Repository folder that implements IProductRepository. All these methods need implementation
3. Add the implementation for the methods via accessing context methods

The screenshot shows the Visual Studio interface with the Solution Explorer pane open. The 'Repository' folder under the 'Seminar3' project is selected. A context menu is open over this folder, with the 'Add' option highlighted. Under the 'Add' option, the 'Class...' item is also highlighted with a red box.

```

using System.Threading.Tasks;
namespace Seminar3.Repository
{
    public interface IProductRepository
    {
        void InsertProduct(Product product);
        void UpdateProduct(Product product);
        void DeleteProduct(int productId);
        Product GetProductById(int Id);
        IEnumerable<Product> GetProducts();
    }
}

```



Replace the generated code with the following.

```

private readonly ProductContext _dbContext;
public ProductRepository(ProductContext dbContext)
{
    _dbContext = dbContext;
}
public void DeleteProduct(int productId)
{
    var product = _dbContext.Products.Find(productId);
    _dbContext.Products.Remove(product);
    Save();
}
public Product GetProductById(int productId)
{
    var prod = _dbContext.Products.Find(productId);
    _dbContext.Entry(prod).Reference(s => s.Category).Load();
    return prod;
}
public IEnumerable<Product> GetProducts()
{
    return _dbContext.Products.Include(s => s.Category).ToList();
}
public void InsertProduct(Product product)
{
    _dbContext.Add(product);
    Save();
}

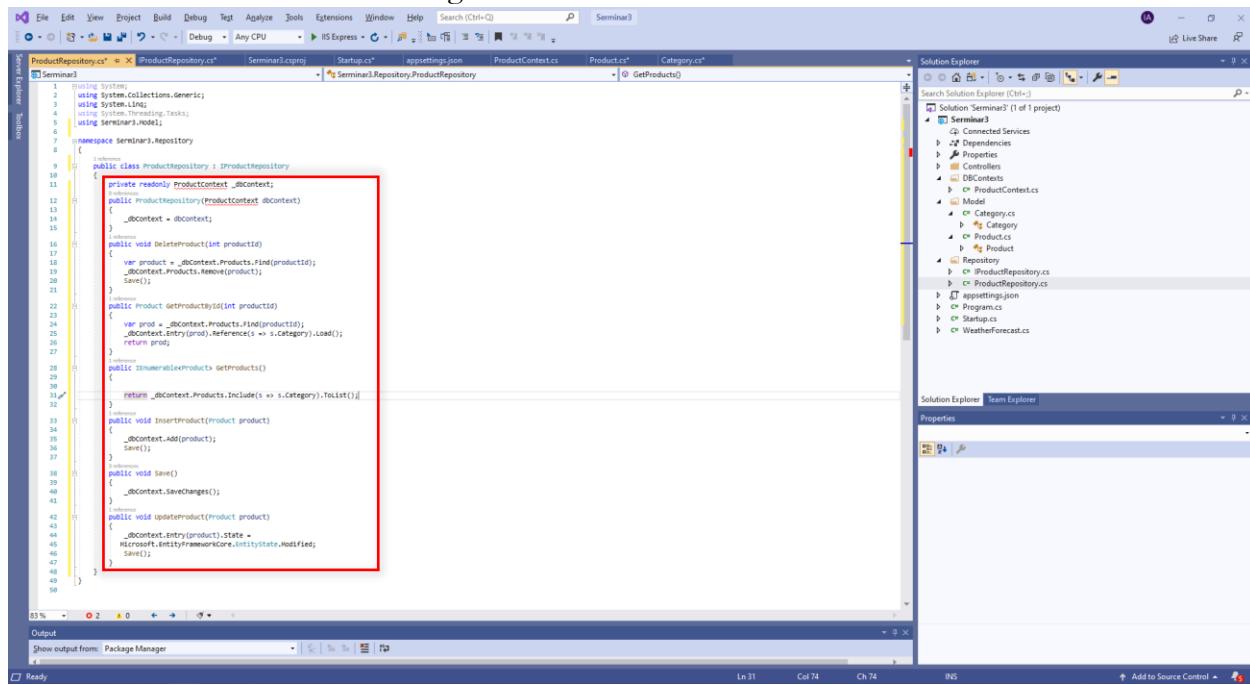
```

```

    }
    public void Save()
    {
        _dbContext.SaveChanges();
    }
    public void UpdateProduct(Product product)
    {
        _dbContext.Entry(product).State =
            Microsoft.EntityFrameworkCore.EntityState.Modified;
        Save();
    }
}

```

And it should be something like this.



You need to assure to fix the red underlined mistakes and by “[using Seminar3.DBContexts;](#)”

After that you need to fix s.Category in [public Product GetProductById\(int productId\)](#), by changing it to s.ProductCategory.

The same goes for s.Category in [public IEnumerable<Product> GetProducts\(\)](#) function. After adding [using Microsoft.EntityFrameworkCore](#) you need to change s.Category to ProductCategory.

4. Open the Startup class in the project and add the code as services.AddTransient(); inside ConfigureServices method so that the repository's dependency is resolved at a run time when needed.

The screenshot shows the Visual Studio IDE with the following details:

- File Explorer**: Shows the project structure for "Seminar3".
- Solution Explorer**: Shows the project structure for "Seminar3" (1 of 1 project). It includes files like `Startup.cs`, `appsettings.json`, `Category.cs`, `Product.cs`, `ProductRepository.cs`, and `WeatherForecast.cs`.
- Code Editor**: The `Startup.cs` file is open. A red box highlights the line of code where the transient service is registered:

```
services.AddTransient<IPrductRepository, ProductRepository>();
```
- Status Bar**: Shows "121 %", "No Issues found", and "Output" tab.
- Output Window**: Shows "Show output from: Package Manager".

## Adding a Controller

The micro-service should have an endpoint for which a controller is needed which exposes the HTTP methods to the client as endpoints of the service methods.

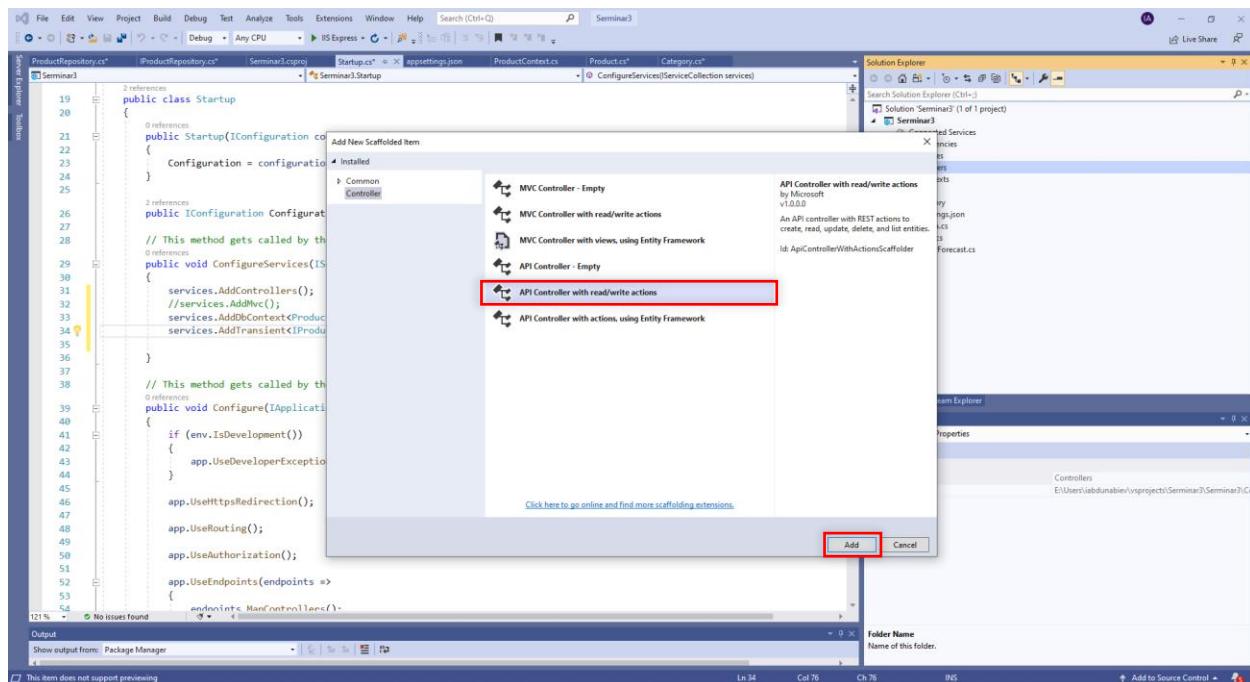
1. Right click on the Controllers folder and add a new Controller

The screenshot shows the Visual Studio IDE interface with the following details:

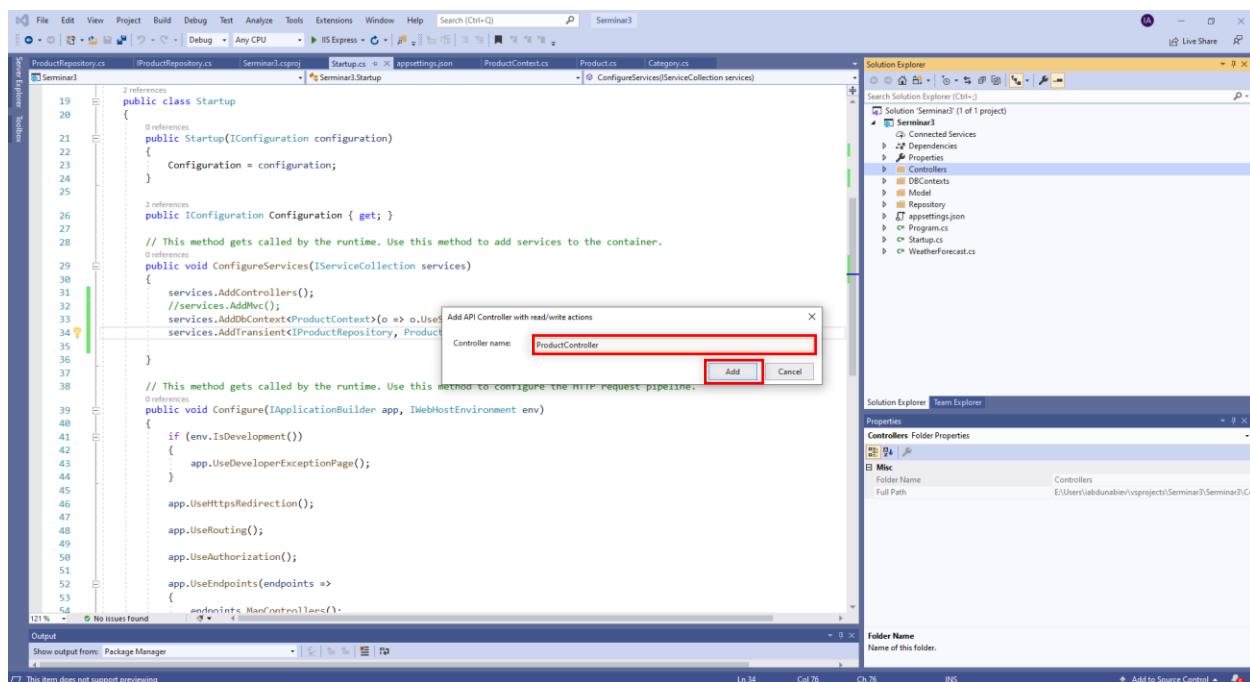
- Solution Explorer:** Shows the project structure for "Seminar3". A context menu is open over the "Controllers" folder, with the "Add" option highlighted.
- Code Editor:** The file "Startup.cs" is open, showing the configuration code for the application.
- Status Bar:** Shows the current file (Startup.cs), line number (Ln 34), column number (Col 76), character position (Ch 76), and the word count (INS).

The "Add" option in the context menu is highlighted with a red box, indicating the next step in the process of creating a new controller.

2. Select the option “API Controller with read/write actions” to add the controller



### 3. Give the name of the controller as ProductController



4. Add implementation to the methods by calling the repository methods as shown below. The basic implementation is shown here for the sake of understanding the concept. The methods could be attribute routed and could be decorated with more annotations as per need.

```

private readonly IProductRepository _productRepository;
public ProductController(IProductRepository productRepository)
{
    _productRepository = productRepository;
}

// GET: api/Product
[HttpGet]
public IActionResult Get()
{
    var products = _productRepository.GetProducts();
    //return new string[] { "value1", "value2" };
}

// GET: api/Product/5
[HttpGet("{id}", Name = "Get")]
public IActionResult Get(int id)
{
    var product = _productRepository.GetProductById(id);
    //return new string[] { "value1", "value2" };
}

// POST: api/Product
[HttpPost]
public IActionResult Post([FromBody]Product product)
{
    using (var scope = new TransactionScope())
    {
        _productRepository.InsertProduct(product);
        scope.Complete();
        return CreatedAtAction(nameof(Get), new { id = product.Id }, product);
    }
}

// PUT: api/Product/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody]Product product)
{
    if (product == null)
    {
        using (var scope = new TransactionScope())
        {
            _productRepository.UpdateProduct(product);
            scope.Complete();
            return new OkObjectResult();
        }
    }
    return new NoContentResult();
}

// DELETE: api/Product/5
[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    _productRepository.DeleteProduct(id);
}

```

`private readonly IProductRepository _productRepository;`  
`public ProductController(IProductRepository productRepository)`

```
{
    _productRepository = productRepository;
}
```

`// GET: api/Product`  
`[HttpGet]`  
`public IActionResult Get()`

```
{
    var products = _productRepository.GetProducts();
    return new OkObjectResult(products);
    //return new string[] { "value1", "value2" };
}
```

`// GET: api/Product/5`  
`[HttpGet("{id}", Name = "Get")]`  
`public IActionResult Get(int id)`

```
{
    var product = _productRepository.GetProductById(id);
    return new OkObjectResult(product);
    //return "value";
}
```

`// POST: api/Product`

```

[HttpPost]
public IActionResult Post([FromBody]Product product)
{
    using (var scope = new TransactionScope())
    {
        _productRepository.InsertProduct(product);
        scope.Complete();
        return CreatedAtAction(nameof(Get), new { id = product.Id }, product);
    }
}

// PUT: api/Product/5
[HttpPut("{id}")]
public IActionResult Put(int id, [FromBody]Product product)
{
    if (product != null)
    {
        using (var scope = new TransactionScope())
        {
            _productRepository.UpdateProduct(product);
            scope.Complete();
            return new OkResult();
        }
    }
    return new NoContentResult();
}

// DELETE: api/ApiWithActions/5
[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    _productRepository.DeleteProduct(id);
    return new OkResult();
}

```

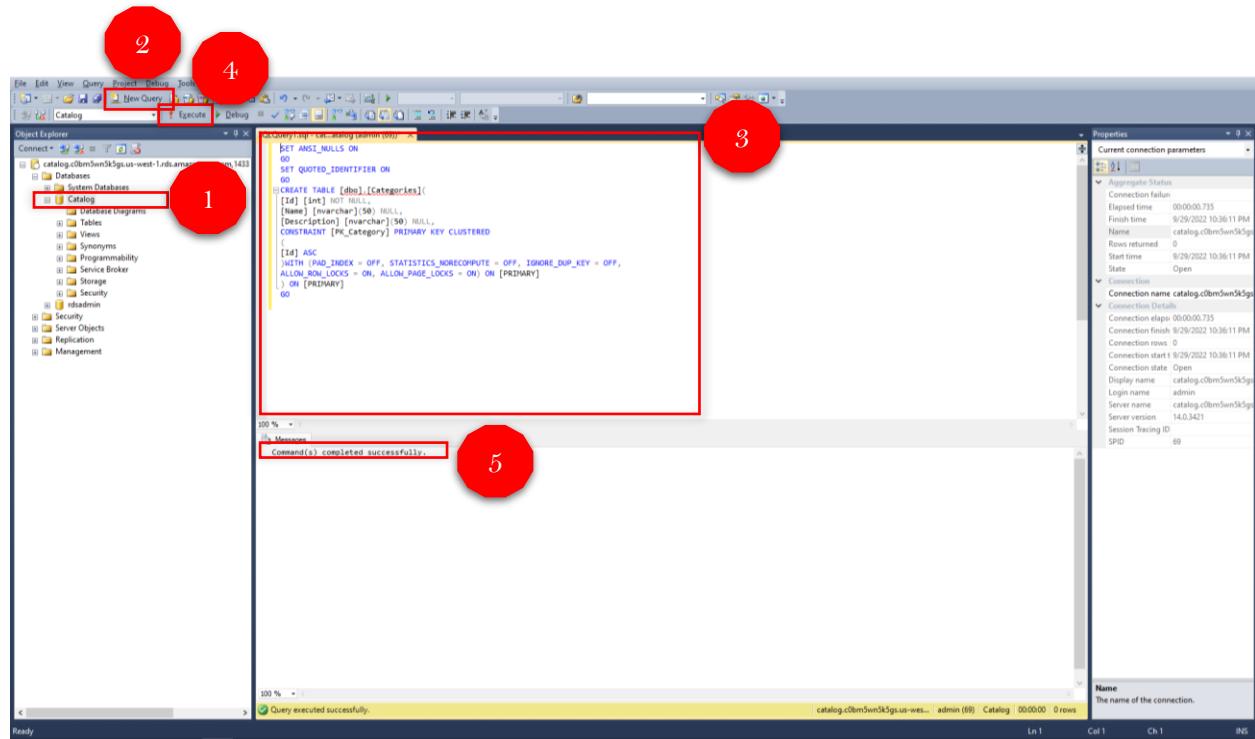
## Step 3: Create a new database to the Microsoft SQL Server Instance

In this step, you will create a database to store the tables related to the Products and Categories using SQL Server Management Studio.

Execute the following scripts in SQL Server Management Studio to create necessary tables in the specially created for this seminar database:

## 1. For Category objects:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](
[Id] [int] NOT NULL,
[Name] [nvarchar](50) NULL,
[Description] [nvarchar](50) NULL,
CONSTRAINT [PK_Category] PRIMARY KEY CLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

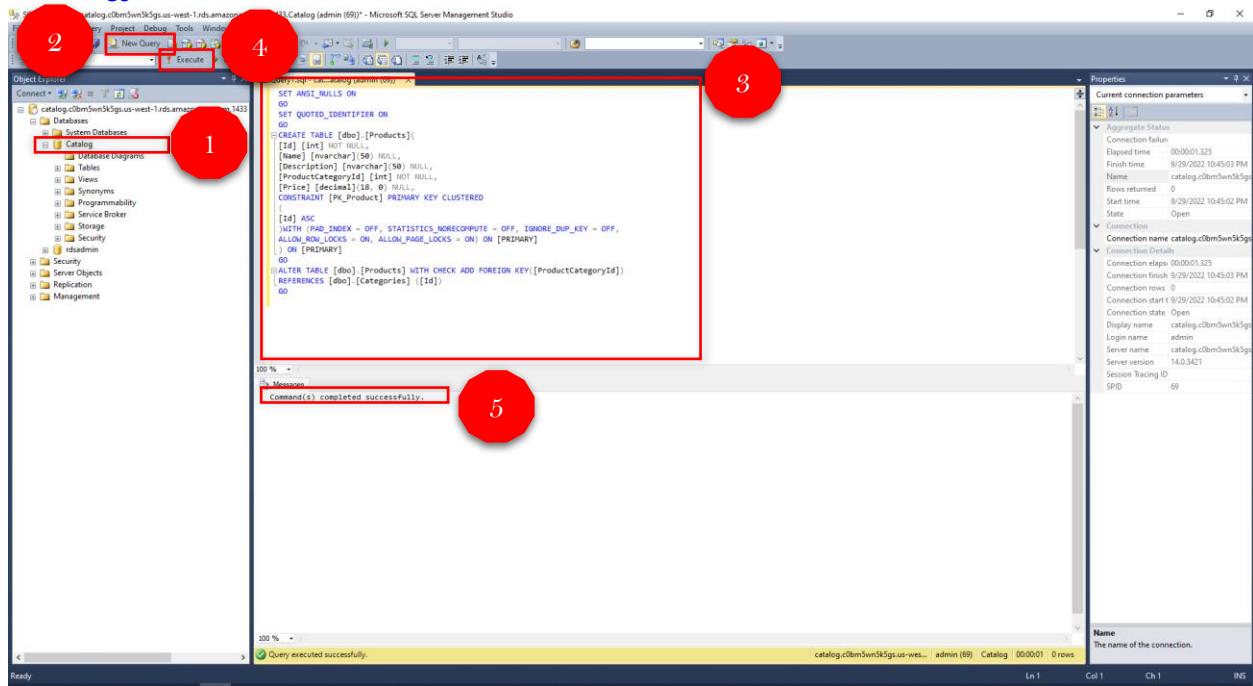


2. For Products objects:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Products](
[Id] [int] NOT NULL,
[Name] [nvarchar](50) NULL,
[Description] [nvarchar](50) NULL,
[ProductCategoryId] [int] NOT NULL,
[Price] [decimal](18, 0) NULL,
CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Products] WITH CHECK ADD FOREIGN KEY([ProductCategoryId])
REFERENCES [dbo].[Categories] ([Id])
GO

```



Add several categories (like Electronics, Grocery, etc.) into Categories table.

SQLQuery3.sql - catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (69)] - Microsoft SQL Server Management Studio

```
USE [catalog]
GO
INSERT INTO [dbo].[Categories]
([Id]
,[Name]
,[Description])
VALUES
(1
,'Electronics'
,'None')
GO
```

Properties

Current connection parameters

- Aggregate Status
- Connection failures
- Elapsed time
- Finish time
- Name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (69)]
- Rows returned: 0
- Start time
- Status: Open

Connection

- Connection name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (69)]
- Connection Details
- Connection elapsed time
- Connection finish time
- Connection rows returned: 0
- Connection start time
- Connection state: Open
- Display name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (69)]
- Login name: admin
- Server name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com
- Server version: 14.0.3421
- Session Tracing ID: 69
- SPID: 69

Name: The name of the connection.

catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (69)] Catalog 00:00:00 0 rows

Ln 10 Col 26 Ch 26 INS

Ready

SQLQuery4.sql - catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (70)] - Microsoft SQL Server Management Studio

```
USE [catalog]
GO
INSERT INTO [dbo].[Products]
([Id]
,[Name]
,[Description]
,[ProductCategoryId]
,[Price])
VALUES
(1
,'PC'
,'Comps'
,1
,24.95)
GO
```

Properties

Current connection parameters

- Aggregate Status
- Connection failures
- Elapsed time: 00:00:00.516
- Finish time: 9/29/2022 11:02:13 PM
- Name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (70)]
- Rows returned: 0
- Start time: 9/29/2022 11:02:12 PM
- Status: Open

Connection

- Connection name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (70)]
- Connection Details
- Connection elapsed time: 00:00:00.516
- Connection finish time: 9/29/2022 11:02:13 PM
- Connection rows returned: 0
- Connection start time: 9/29/2022 11:02:12 PM
- Connection state: Open
- Display name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (70)]
- Login name: admin
- Server name: catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com
- Server version: 14.0.3421
- Session Tracing ID: 70
- SPID: 70

Name: The name of the connection.

(1 row(s) affected)

catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (70)] Catalog 00:00:00 0 rows

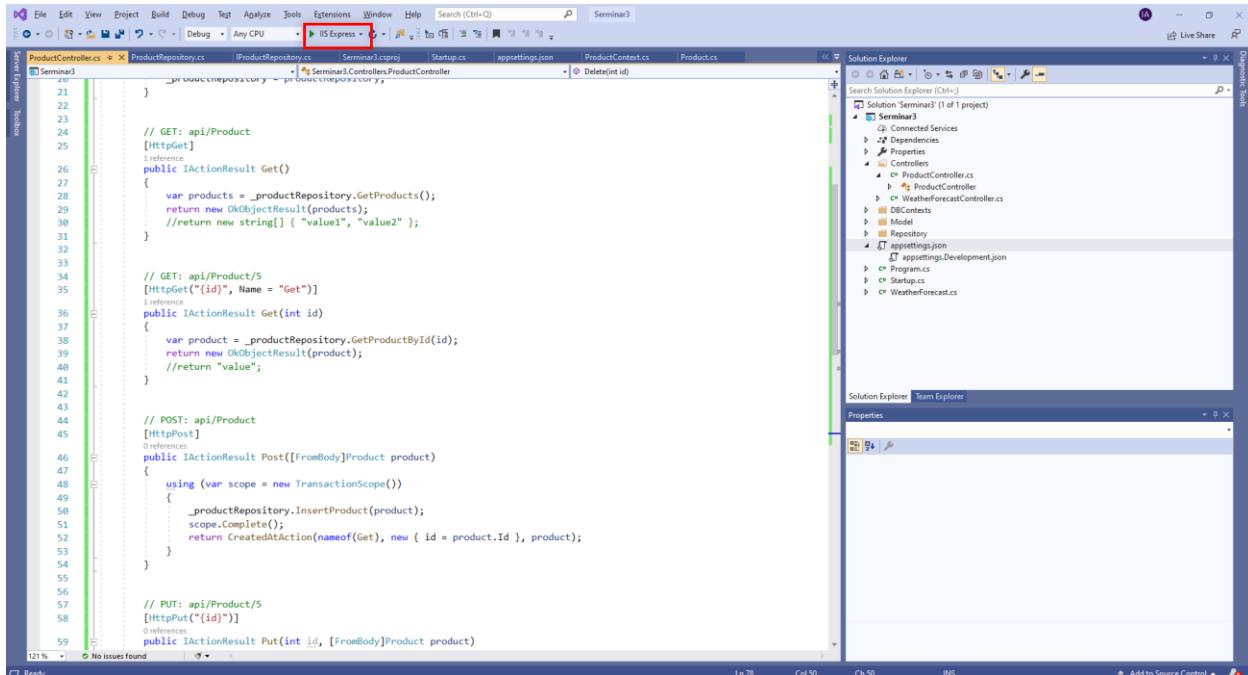
catalog.c0bm5vn5k5gs.us-west-1.rds.amazonaws.com,1433.Catalog [admin (70)] Catalog 00:00:00 0 rows

Ln 15 Col 19 Ch 19 INS

Matches: (

## Step 4: Run the Product Micro-service

The service could be run via IIS Express i.e. Visual Studio default. Choose IIS Express in the Visual Studio as shown below and press F5 or click that IIS Express button itself.



The application will be up once the browser page is launched. Since it has nothing to show, it will be blank, but the service could be tested via any API testing client.



Postman is suggested for testing the service endpoint. Please use Postman for testing Get/Put/Post methods

## Step 5: Delete the DB Instance

You can easily delete the Microsoft SQL Server DB Instance from the Amazon RDS console. It is a best practice to delete instances that you are no longer using so that you don't keep getting charged for them. Deleting an instance was covered in the previous seminar.

You have created your first Micro-service application using .NET Core. This application can be run on Linux machines. Please try to create a new instance and run the newly created app in your Linux machine.

## Seminar 4

# Consuming REST Service in ASP.NET

The term microservices portrays a software development. Microservices is more about applying a certain number of principles and architectural patterns to software development. Each microservice lives independently, but on the other hand, also all rely on each other. All microservices in a project get deployed in production at their own pace, on-premises or in the cloud, independently, living side by side.

In this seminar, you will learn how to consume REST Service in ASP.NET MVC application with the help of HttpClient. As a REST service we will use the REST service that we developed in the previous seminar using ASP.NET Core.

## Step 0.

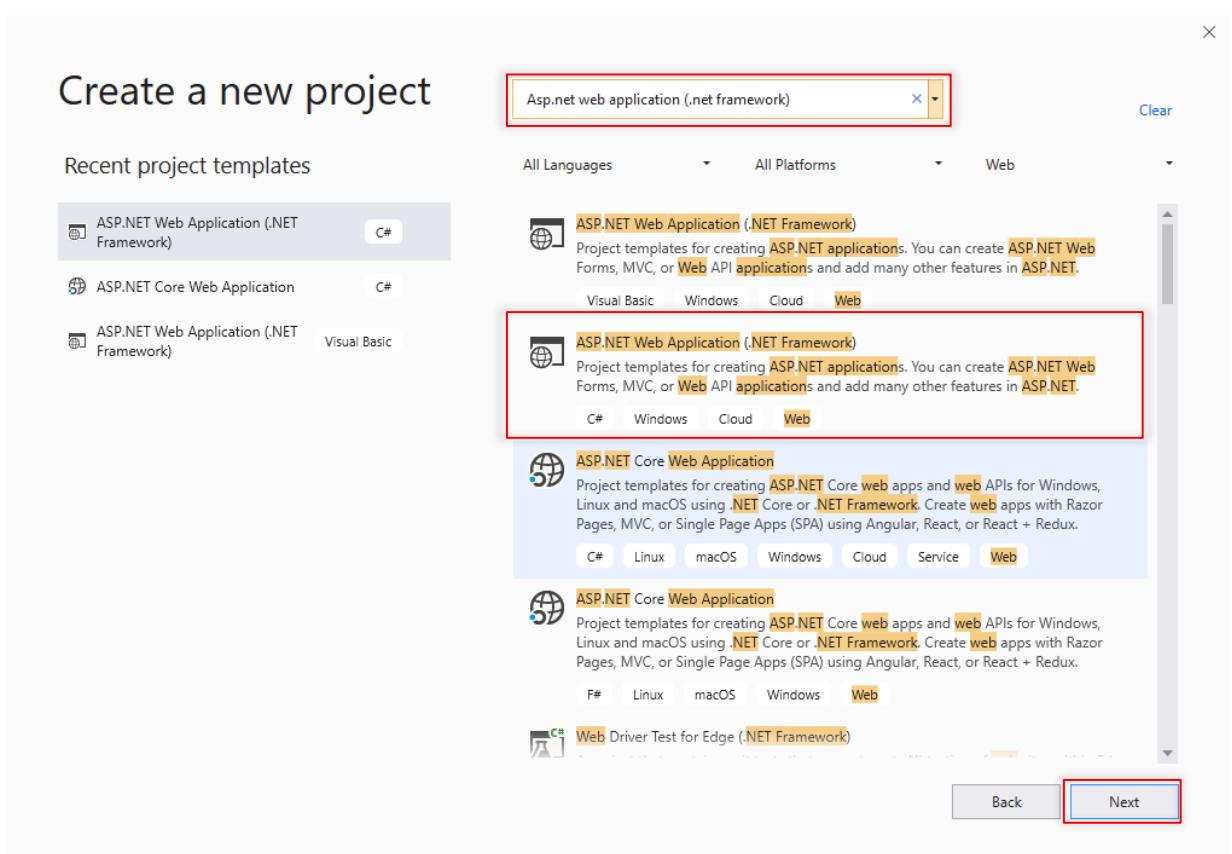
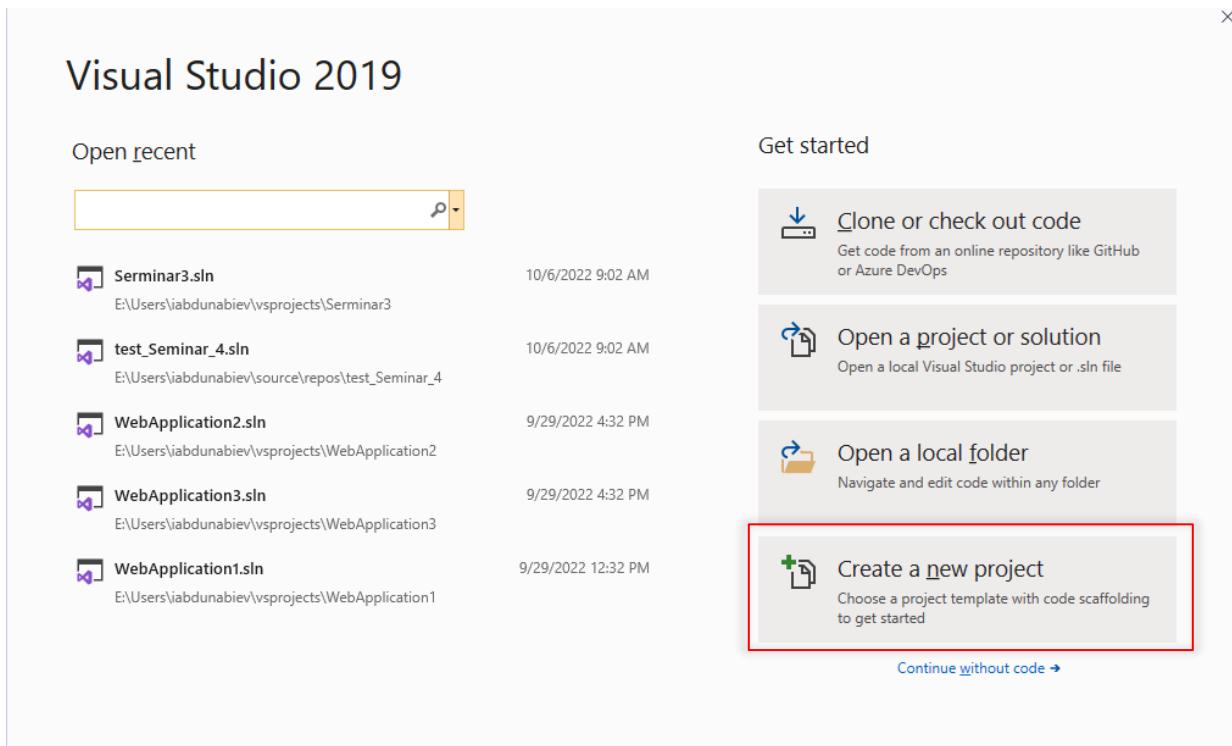
Student supposed to have an application of previous seminar that is running with that is connected to AWS Microsoft SQL Server with modifications previously done. We are going to be need to localhost URL and use it later when we are going to connect to it.

## Step 1. Create MVC Application.

This part will demonstrate how to create Product microservice using ASP.MVC Core step by step.

### Creating an ASP.NET Core Application

1. Open Visual Studio and add new “Project”. Select “ASP.NET Web Application Template”, provide the Project with a meaningful name and click OK. The following Window will appear. Choose empty project template and check on MVC option.



X

## Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name

Seminar4

Location

E:\Users\iabdunabiev\source\repos



Solution name i

Seminar4

Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Back

Create

2. The preceding step creates the simple empty ASP.NET MVC application without Model, View, and Controller. The Solution Explorer of created web application will look like the following.

## Create a new ASP.NET Web Application

**Empty**  
An empty project template for creating ASP.NET applications. This template does not have any content in it.

**Web Forms**  
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

**MVC**  
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

**Web API**  
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

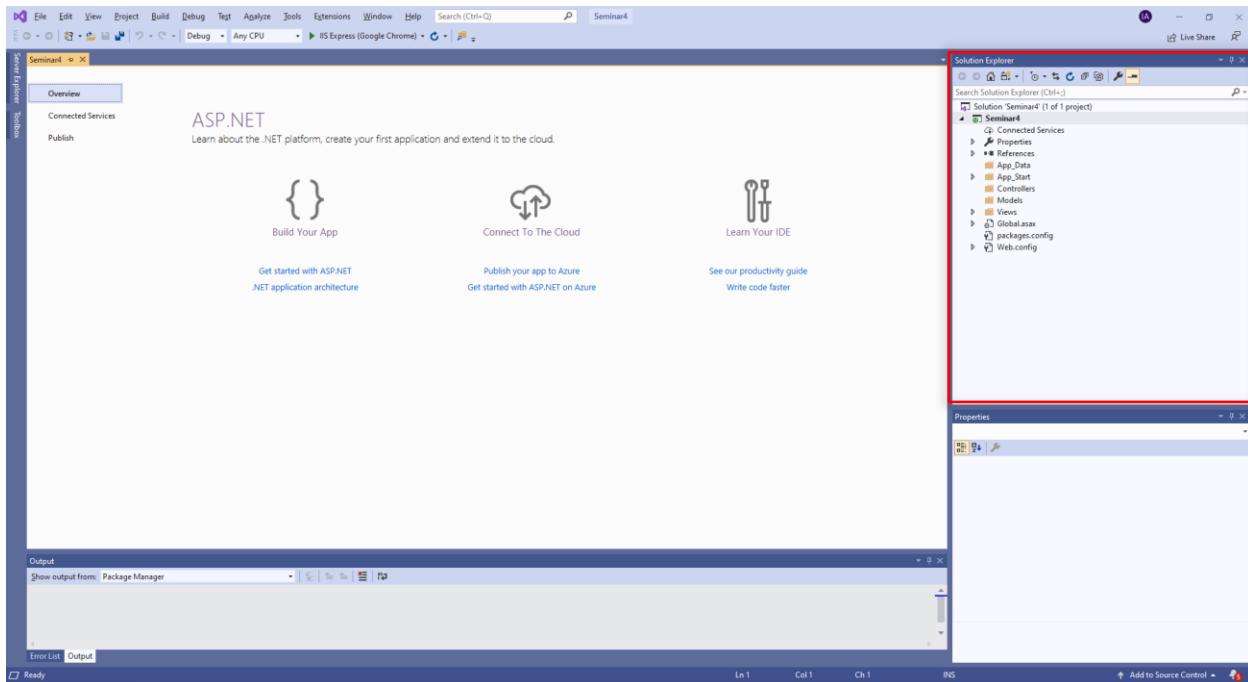
**Single Page Application**  
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

**Authentication**  
No Authentication  
[Change](#)

**Add folders & core references**  
 Web Forms  
 MVC  
 Web API

**Advanced**  
 Configure for HTTPS  
 Docker support  
(Requires [Docker Desktop](#))  
 Also create a project for unit tests  
[Seminar4.Tests](#)

[Back](#) [Create](#)

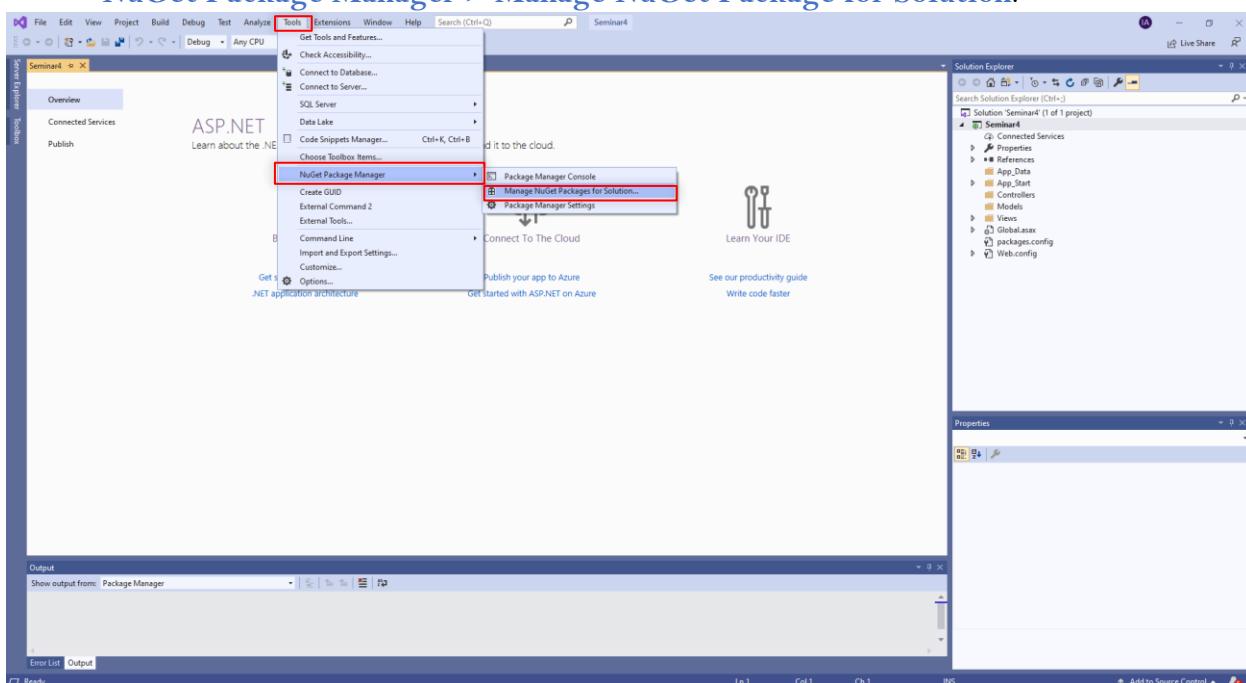


## Step 2. Install HttpClient and WebAPI.Client libraries from NuGet

1. We are going to use HttpClient to consume the web API REST Service, so we need to install this library from NuGet Package Manager.

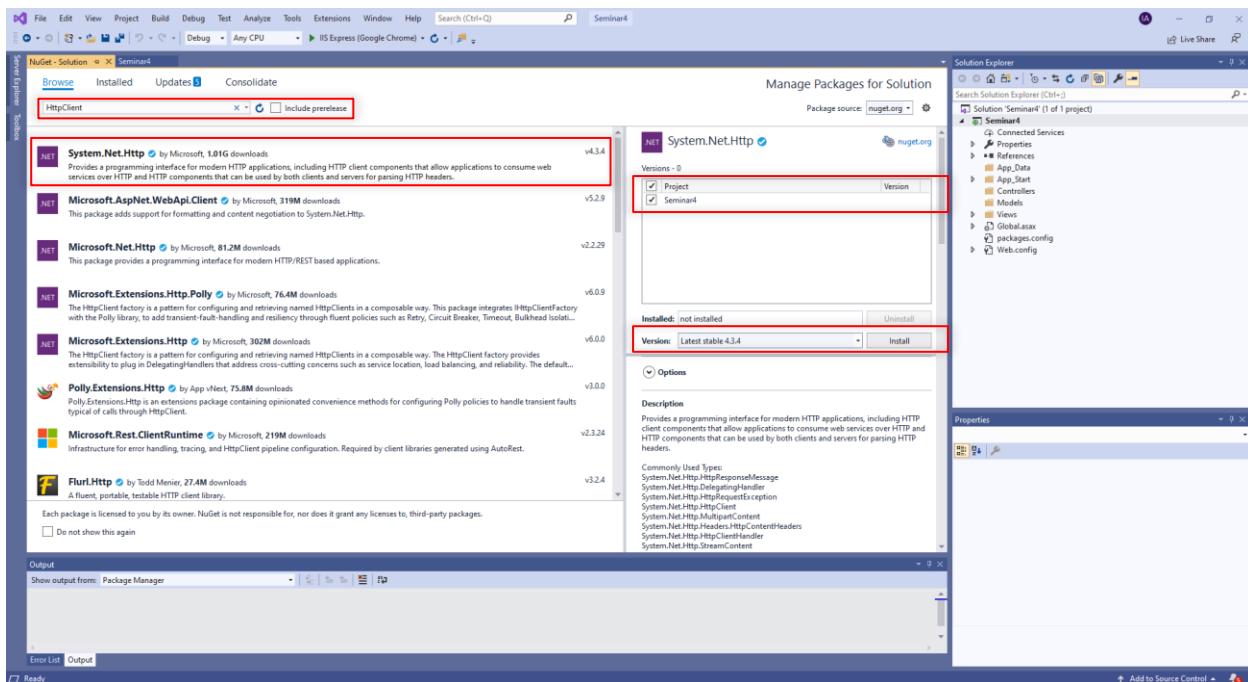
### What is HttpClient?

HttpClient is base class which is responsible to send HTTP request and receive HTTP response resource i.e., from REST service. To install HttpClient, click on **Tools > NuGet Package Manager > Manage NuGet Package for Solution**.

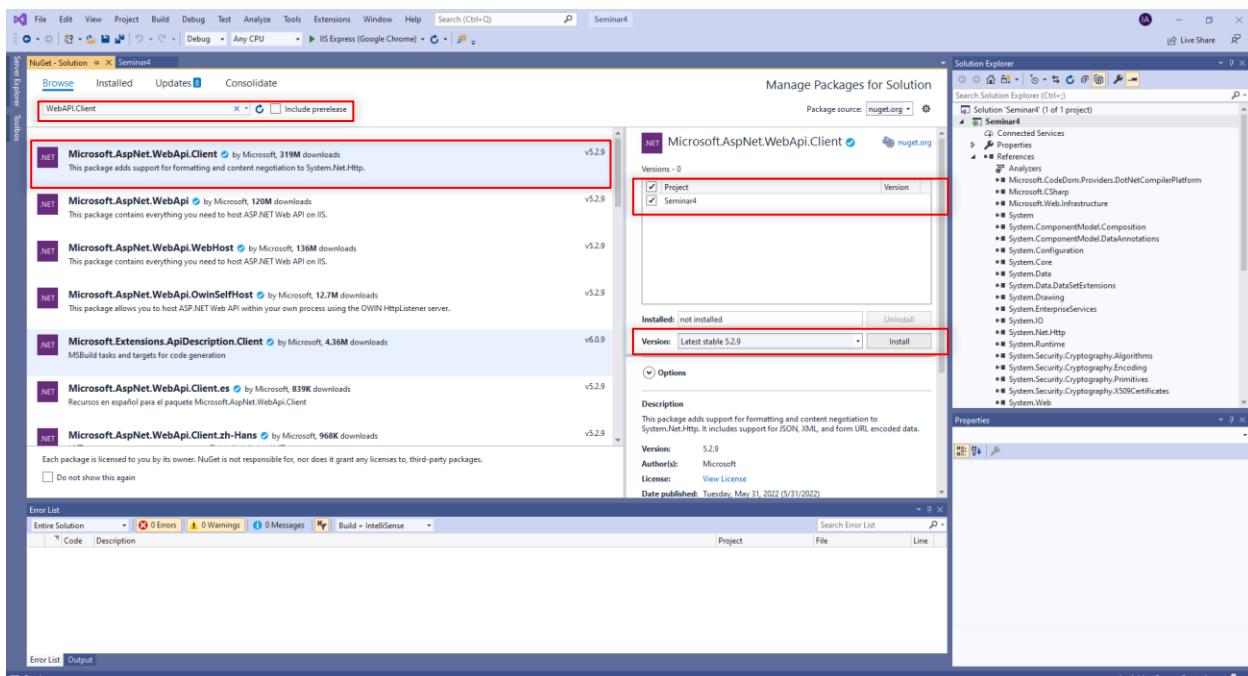


From there Search for HttpClient as shown in the following Figure.

Now, click on “Install” button after choosing the appropriate version.



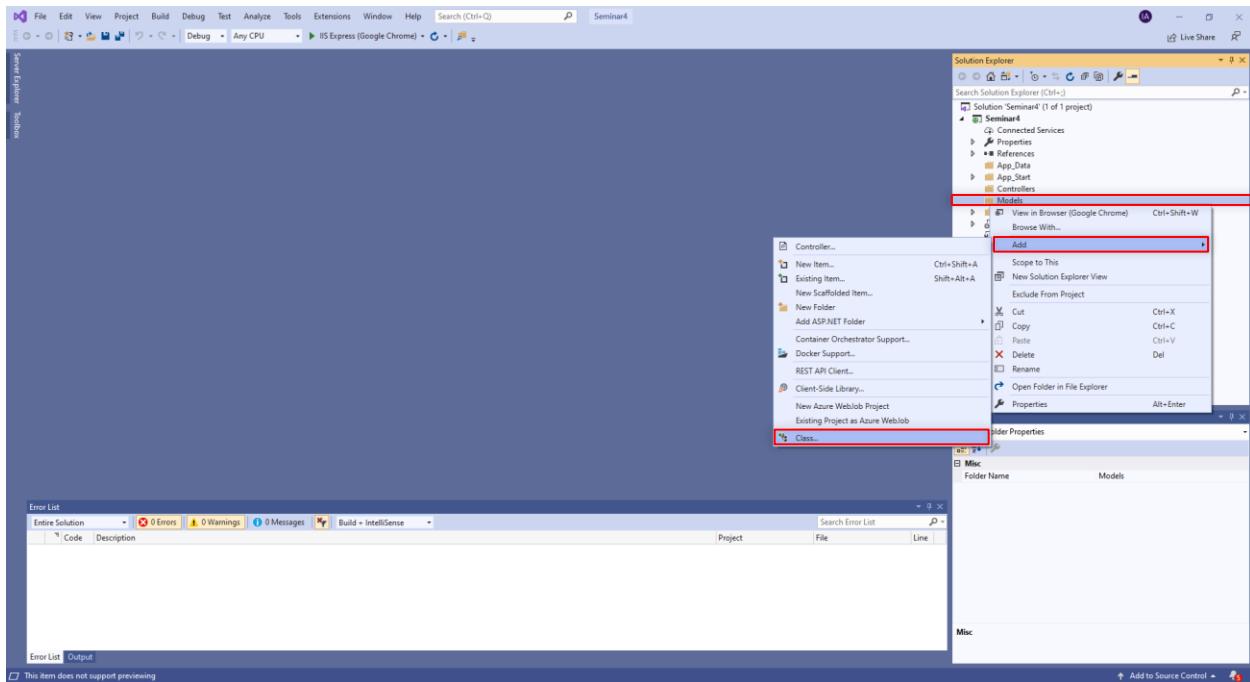
- Now install WebAPI.Client package. This package is used for formation and content negotiation which provides support for System.Net.Http. To install, click on **Tools > NuGet Package Manager > NuGet Package for Solution** and search for WebAPI.Client, as shown in following.



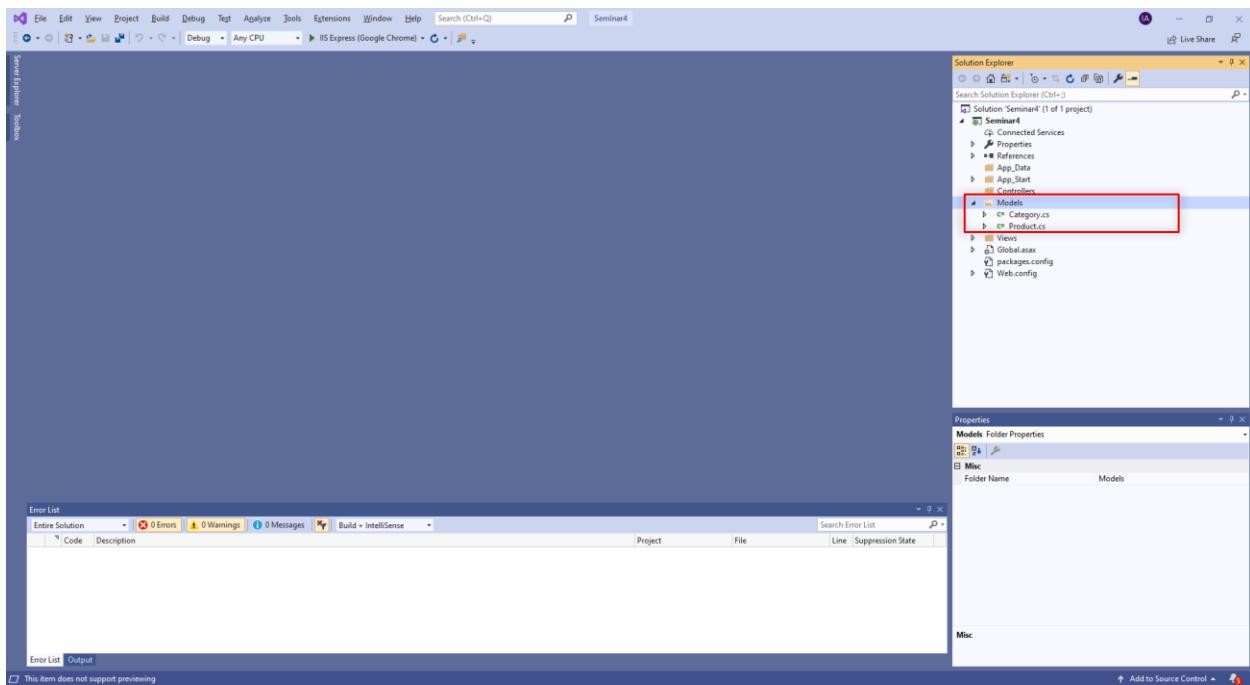
Now we have installed necessary NuGet Package to consume Web API Rest service in web application.

## Step 3: Create a Model Classes.

Now, let's us create the Model classes names [Product](#) and [Category](#), by right clicking on Models folder with same number of entities which are exposing by our hosted Web API REST service to exchange the data. The code snippet of created classes will look like this.



The Product and Category classes would have the properties that are following in the pictures below.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 
6 namespace Seminar4.Models
7 {
8     public class Product
9     {
10         public int Id { get; set; }
11         public string Name { get; set; }
12         public string Description { get; set; }
13         public decimal Price { get; set; }
14         public Category ProductCategory { get; set; }
15     }
16 }
```

Solution Explorer shows the project structure:

- Solution Seminar4 (1 of 1 project)
  - Connected Services
  - Properties
  - App\_Browsers
  - App\_Data
  - App\_Start
  - Controllers
  - Models
    - Category.cs
    - Product.cs
  - Views
  - Global.asax
  - packages.config
  - Web.config

Properties window for Product.cs:

|          |  |
|----------|--|
| Advanced | Build Action: Compile<br>Copy to Output Directory: Do not copy   |
| Misc     | File Name: Product.cs<br>Full Path: E:\Users\abdunnie\source\repos\Seminar4\Seminar4\Models\Product.cs |

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 
6 namespace Seminar4.Models
7 {
8     public class Category
9     {
10         public int Id { get; set; }
11         public string Name { get; set; }
12         public string Description { get; set; }
13     }
14 }
```

Solution Explorer shows the project structure:

- Solution Seminar4 (1 of 1 project)
  - Connected Services
  - Properties
  - App\_Browsers
  - App\_Data
  - App\_Start
  - Controllers
  - Models
    - Category.cs
    - Product.cs
  - Views
  - Global.asax
  - packages.config
  - Web.config

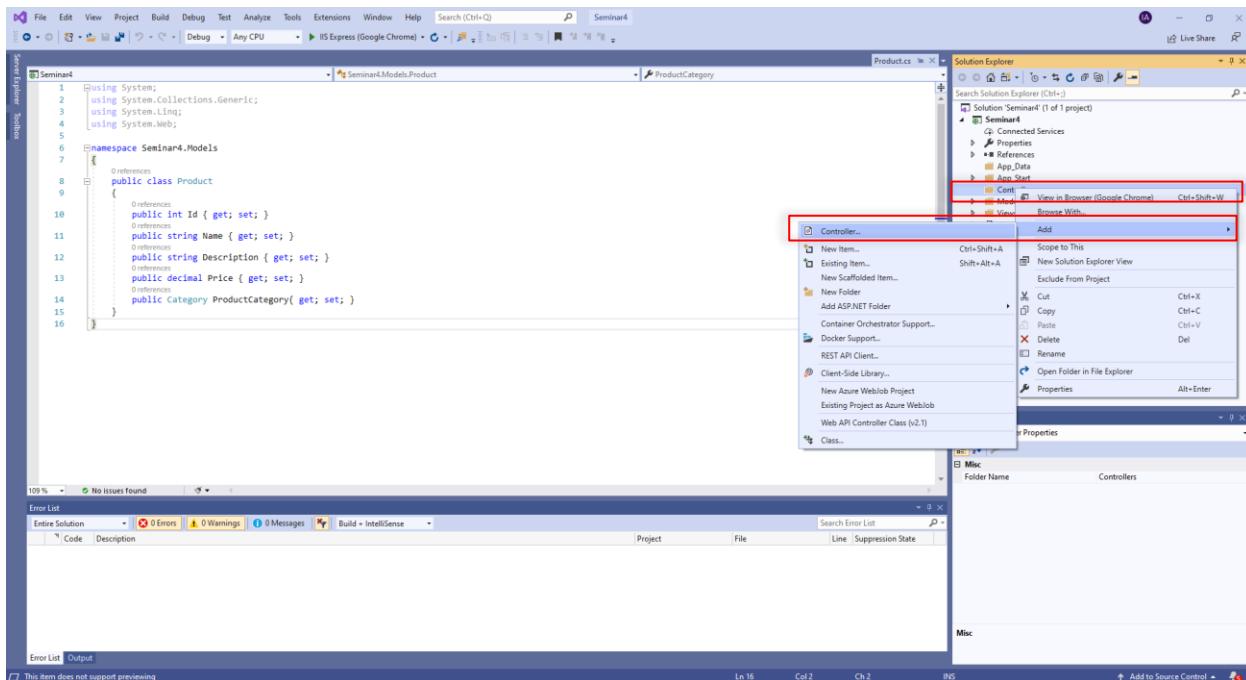
Properties window for Category.cs:

|          |  |
|----------|--|
| Advanced | Build Action: Compile<br>Copy to Output Directory: Do not copy   |
| Misc     | File Name: Category.cs<br>Full Path: E:\Users\abdunnie\source\repos\Seminar4\Seminar4\Models\Category.cs |

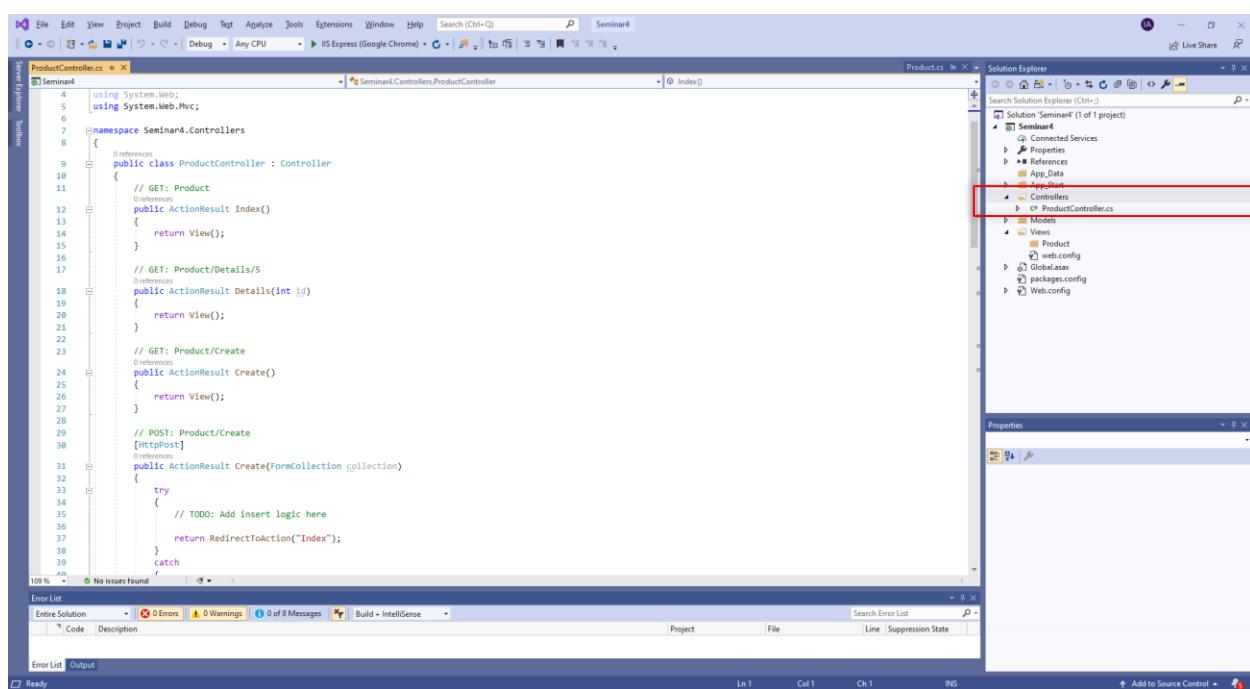
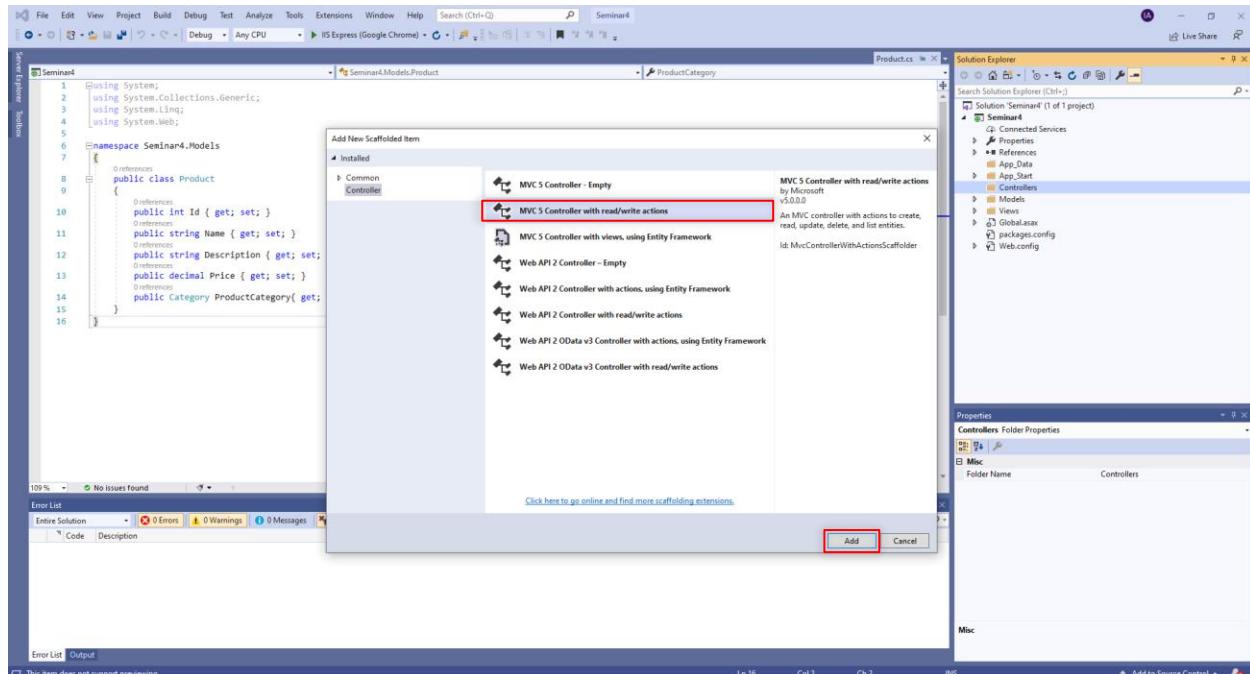
|  |  |
|--|--|
| <pre> public class Category {     public int Id { get; set; }     public string Name{ get; set; }     public string Description { get; set; } } </pre> | <pre> public class Product {     public int Id { get; set; }     public string Name { get; set; }     public string Description { get; set; }     public decimal Price { get; set; }     public Category ProductCategory{ get; set; } } </pre> |
|--|--|

## Step 4. Create Controllers

Now, let us add ASP.NET MVC Controller, as shown in the screenshot given below (right click on the Controllers folder in the Solution and select **Add > Controller**.



Choose *MVC {version number} Controller* with read/write actions and name the controller as **ProductController**, because we will be working with Product only for this class.



Our hosted Web API REST Service includes these two methods, as given below.

### Product (GET)

Product/ID (POST) which takes id as input parameter

We are going to call Product method which returns all products. The hosted web API REST service base URL is <http://localhost:57719> and to call Product from hosted web API REST

service, The URL should be Base url+api+apicontroller name +web api method name as following,

In the preceding URL `http://localhost:57719` is the base address of web API service, it can be different as per your server. “API” is used to differentiate between Web API controller and MVC controller requests. Product is the Web API controller name. After modifying the code of `ProductController` class, the code will look like the following:

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `ProductController.cs` file under the `Seminar4` namespace. The code implements a `ProductController` class with methods for listing products and retrieving details. A red box highlights the logic for sending a GET request to a local API endpoint and deserializing the response into a list of `Product` objects. The `Solution Explorer` pane on the right shows the project structure for `Seminar4`, including files like `Product.cs`, `ProductController.cs`, and `Web.config`. The `Error List` and `Output` panes at the bottom indicate no issues found.

```
namespace Seminar4.Controllers
{
    [Route("api/[controller]")]
    public class ProductController : Controller
    {
        // async Because we are connecting to API
        // Task<ActionResult>
        // GET: Product
        public async Task<ActionResult> Index()
        {
            //Hosted web API REST Service base url
            string Baseurl = "https://localhost:44358/";
            List<Product> ProdInfo = new List<Product>();
            using (var client = new HttpClient())
            {
                //Passing service base url
                client.BaseAddress = new Uri(Baseurl);
                client.DefaultRequestHeaders.Clear();

                //Define request data format
                client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

                //Sending request to find web api REST service resource using HttpClient
                HttpResponseMessage Res = await client.GetAsync("api/Product");

                //Checking the response is successful or not which is sent HttpClient
                if (Res.IsSuccessStatusCode)
                {
                    //Storing the response details received from web api
                    var PrResponse = Res.Content.ReadAsStringAsync().Result;

                    //Deserializing the response received from web api and storing the Product list
                    ProdInfo = JsonConvert.DeserializeObject<List<Product>>(PrResponse);
                }
                //returning the Product list to view
                return View(ProdInfo);
            }
        }

        // GET: Product/Details/S
        public ActionResult Details(int id)
        {
        }
    }
}
```

## Step 5: Create Strongly Typed View

A screenshot of the Visual Studio IDE interface. The main window shows a C# code editor with the file `ProductController.cs` open. A context menu is displayed over a selected portion of the code, specifically the line `return View();`. The menu items visible include:

- Add View...
- Run Selected Code Locally
- Go To View Ctrl+M, Ctrl+G
- Quick Actions and Refactorings... Ctrl+.
- Rename... Ctrl+R, Ctrl+B
- Remove and Sort Usings Ctrl+R, Ctrl+G
- View Code F7
- Peek Definition Alt+F12
- Go To Definition F12
- Go To Implementation Ctrl+F12
- Find All References Shift+F12
- View Call Hierarchy Ctrl+K, Ctrl+T
- Create Unit Tests
- Breakpoint F11
- Run To Cursor Ctrl+F10
- Execute in Interactive Ctrl+E, Ctrl+E
- Snippet
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Annotation
- Outlining

The Solution Explorer, Properties, and Task List panes are also visible on the right side of the interface.

Now, click on "Add" button. It will create View named index after modifying the default code. The code snippet of the Index View looks like the following:

The screenshot shows the Visual Studio IDE with the ProductController.cs file open in the code editor. A modal dialog titled 'Add View' is displayed, prompting for view details. The 'View name' field is set to 'Index', the 'Template' is 'List', and the 'Model class' is 'Product (Seminar4.Models)'. The 'Add' button in the dialog is highlighted with a red box. The Solution Explorer on the right shows the project structure, including the ProductController.cs file.

```

27     //Passing service base url
28     client.BaseAddress = new Uri(Baseurl);
29     client.DefaultRequestHeaders.Clear();
30
31     //Define request data format
32     client.DefaultRequestHeaders.Accept.Add(new
33         MediaTypeWithQualityHeaderValue("application/json"));
34
35     //Sending request to find web api REST service resource using HttpClient
36     HttpResponseMessage Res = await client.GetAsync("api/Product");
37
38     //Checking the response is successful or not which is sent by HttpClient
39     if (Res.IsSuccessStatusCode)
40     {
41         //Storing the response details received from web api
42         var Preresponse = Res.Content.ReadAsStringAsync().Result;
43
44         //Deserializing the response received from web api and storing into productInfo object
45         ProdInfo = JsonConvert.DeserializeObject<List<Product>>(Preresponse);
46     }
47
48     //returning the Product list to view
49     return View(ProdInfo);
50 }
51
52 // GET: Product/Details/
53 public ActionResult Details(int id)
54 {
55     return View();
56 }
57
58 // GET: Product/Create
59 [HttpGet]
60 public ActionResult Create()
61 {
62     return View();
63 }
64
65 // POST: Product/Create
66 [HttpPost]
67 public ActionResult Create(FormCollection collection)
68 {
69 }

```

## Step 6 Run the application

When you run the application, the following screen will appear.

The screenshot shows a browser window displaying the 'Index' view of the application. The page title is 'Index - My ASP.NET Application'. The content area shows a table with a header 'Create New' and columns 'Name', 'Description', and 'Price'. Below the table is a copyright notice: '© 2022 - My ASP.NET Application'.

| Create New |                    |         |                         |
|------------|--------------------|---------|-------------------------|
| Name       | Description        | Price   |                         |
| Computer   | Personal Computers | 981.00  | Edit   Details   Delete |
| Smartphone | iPhone             | 1981.00 | Edit   Details   Delete |
| Headphones | Sony WH1000XM3     | 384.00  | Edit   Details   Delete |
| Milk       | Drinkable Milk     | 1.00    | Edit   Details   Delete |
| Cookies    | Roman Cookies      | 1.00    | Edit   Details   Delete |
| Cereal     | Atkins Cereals     | 4.00    | Edit   Details   Delete |
| Bed        | Beds for Bedroom   | 225.00  | Edit   Details   Delete |
| Chair      | Chair to seat      | 45.00   | Edit   Details   Delete |
| Table      | Classroom tables   | 145.00  | Edit   Details   Delete |

## Step 7: Homework and Delete AWS Instance.

**Homework:** Implement [Edit](#), [Details](#), [Delete](#) functions. You should have a function for [Adding](#) new Product and [Category](#).

Stop your applications that are running, [ASP.NET](#) and [.NET Core](#) based applications.

[Delete](#) your AWS database instance after you have done working.

## Seminar 5.

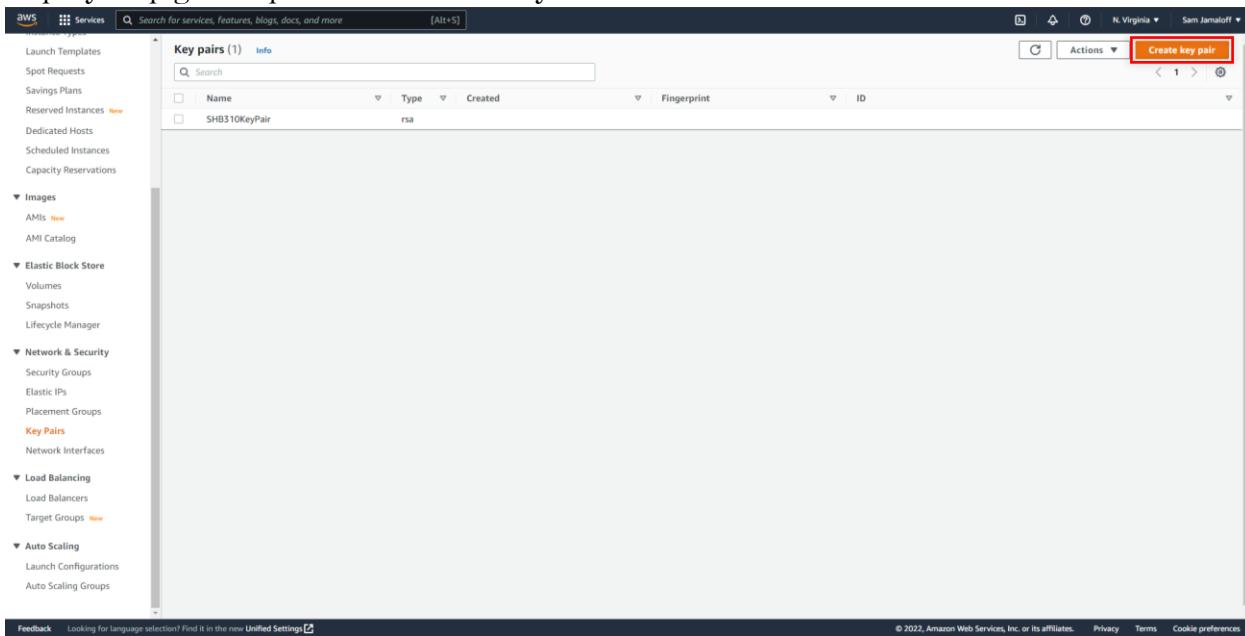
# Deploy Code from GitHub to a Virtual Machine

In this seminar, you will learn how to deploy application code to a virtual machine on AWS. You will use AWS CodeDeploy, a service that automates code deployments to AWS or on-premises servers, to deploy code to virtual machines that you create and manage with Amazon EC2. Everything done in this seminar is free tier eligible.

## Step 1: Create a Key Pair

You will need to create a key pair to access your virtual machine with Amazon EC2. If you already have a key pair, skip ahead to Step 2.

- a. When you [click here](#), (or go for EC2 services and when you scroll down there is a key pairs section in Network and Security section, there is Key Pairs. Go ahead and click on that) The AWS Management Console will open in a new browser window, so you can keep this step-by-step guide open. Click Create Key Pair.



- b. Give a proper naming and for this Seminar 5 Purposes we are going to name it as *MyKeyPairSeminar5* and choose *.pem* option because we are going to be connecting to our instance using **ssh**. Right after you click Create Key Pair key will be downloaded to your local machine. Find its location and remember it, since we will be using that exact location later, for referring while connecting to our instance in AWS.

**Note:** Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To learn more about key pairs, see [Amazon EC2 Key Pairs](#).

**Create key pair**

**Key pair**  
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name  The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)

- RSA
- ED25519

Private key file format

- .pem For use with OpenSSH
- .ppk For use with PuTTY

Tags - optional  
No tags associated with the resource.

Add new tag You can add up to 50 more tags.

Cancel **Create key pair**

## Step 2: Set Up a GitHub Account

In the next 2 steps, we will create a repository for our code online and upload the code for our website to be stored in the repository. Then we will deploy the code from the repository to our virtual machine.

---

[Join](#) to GitHub with your accounts, if you have one, if you don't have any you can create one [here](#)

Sign in to GitHub

Username or email address

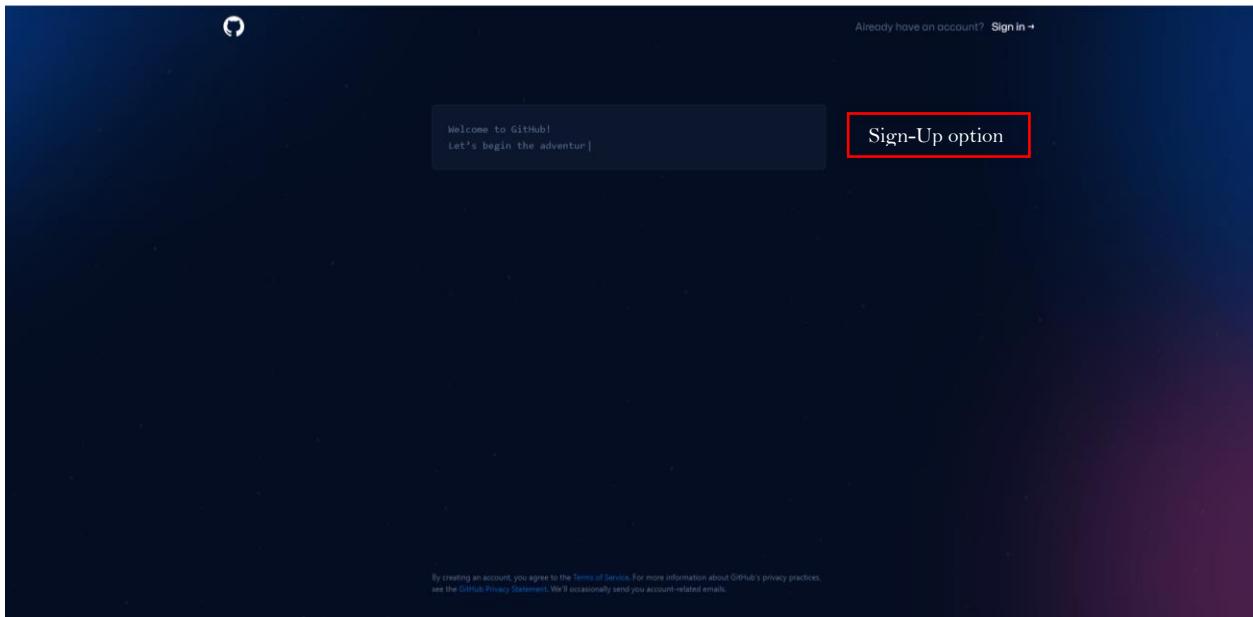
Password [Forgot password?](#)

**Sign in**

New to GitHub? [Create an account.](#)

Sign-In Option

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)



## Step 3: Create a GitHub Repository

In this step, we will use git to upload the code for our website to be stored in GitHub.

---

1. On the GitHub home page, do one of the following:
  - a. In Your repositories, choose “New”
2. In the Create a new repository page, do the following:
  - a. In the Repository name box, enter *CodeDeployGitHubDemo*.
  - b. Select Public.  
Note: Selecting the default “*Public*” option means that anyone can see this repository. You can select the Private option to limit who can see and commit to the repository.
  - c. Clear the Initialize this repository with a README check box. You will create a README.md file manually in the next step instead.
  - d. Choose Create repository
3. Follow the instructions for your local machine type to use the command line to create the repository

Download the temporary website from intranet and remember where you have downloaded it.

Create a new folder with the name *CodeDeployGitHubDemo*, in a location which is easy to remember, and unzip recently downloaded website there.

Open GitBash on that folder by right clicking into free space of that window, and run the following commands, one at a time:

```
git config --global user.name "your_user_name"  
git config --global user.email "your_user_email"
```

Above commands are for configuring GitHub account for locally.

```
git init  
git add .  
git commit -m "init: First Commit"  
git branch -M main
```

now create a ssh key from your local machine into GitHub for smooth communication and being able to push your code easier

```
ssh-keygen -t ed25519 -C "your_user_email_for_github"  
eval `ssh-agent -s`  
ssh-add ~/.ssh/id_ed25519  
clip < ~/.ssh/id_ed25519.pub (this code will copy public-key of your ssh key into clipboard so that you can use it later when you need you paste it into GitHub )  
Go to Settings of your GitHub Accounts' settings, from there you will find "Access".  
From access select SSH and GPG keys. Click "New SSH Key" button.  
Account > Settings > Access(SSH and GPG Keys) > New SSH Key.  
Give a title and paste clipped public-key into key text-area and Click add SSH key button.
```

And now go back to GitBash window. Type the following commands

```
git remote add origin {git@github.com:[user name]/CodeDeployGitHubDemo.git}  
this command is for adding remote origin and connecting into it using ssh.
```

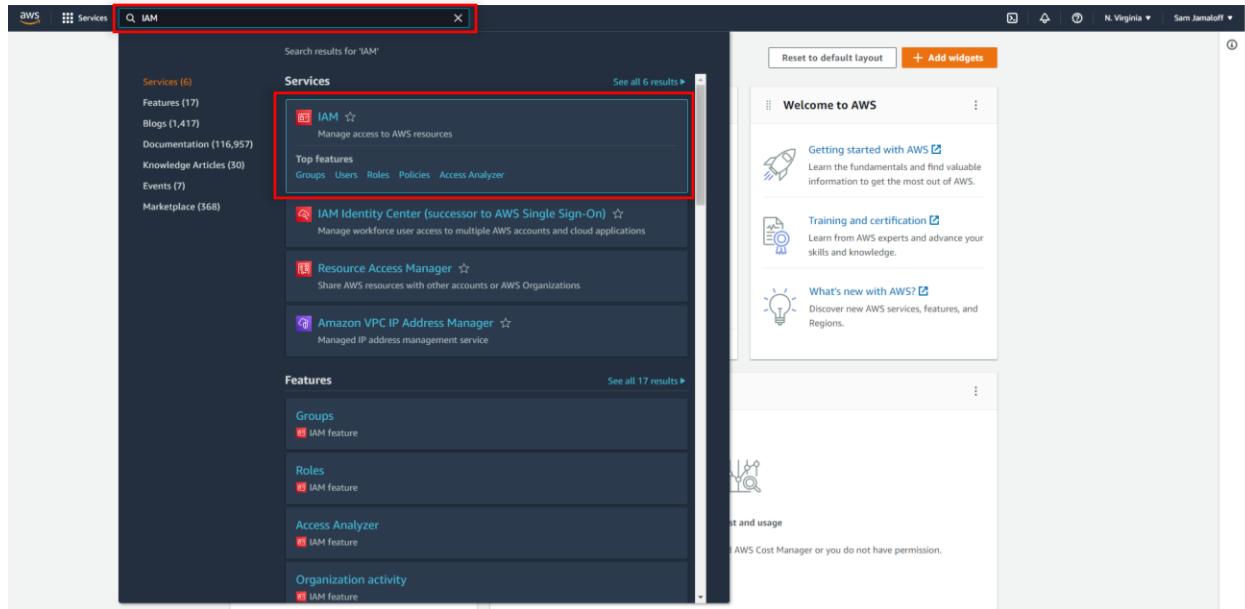
```
git push -u origin main
```

Now check your GitHub repository for code.

## Step 4: Create an IAM Instance Profile for Your Amazon EC2 Instances

Your Amazon EC2 instances need permission to access the GitHub repositories where the applications are stored. To launch Amazon EC2 instances that are compatible with CodeDeploy, you must create an additional IAM role, an instance profile. This step shows you how to create an IAM instance profile to attach to your Amazon EC2 instances. This role gives CodeDeploy permission to access the Amazon S3 buckets or GitHub repositories where your application is stored.

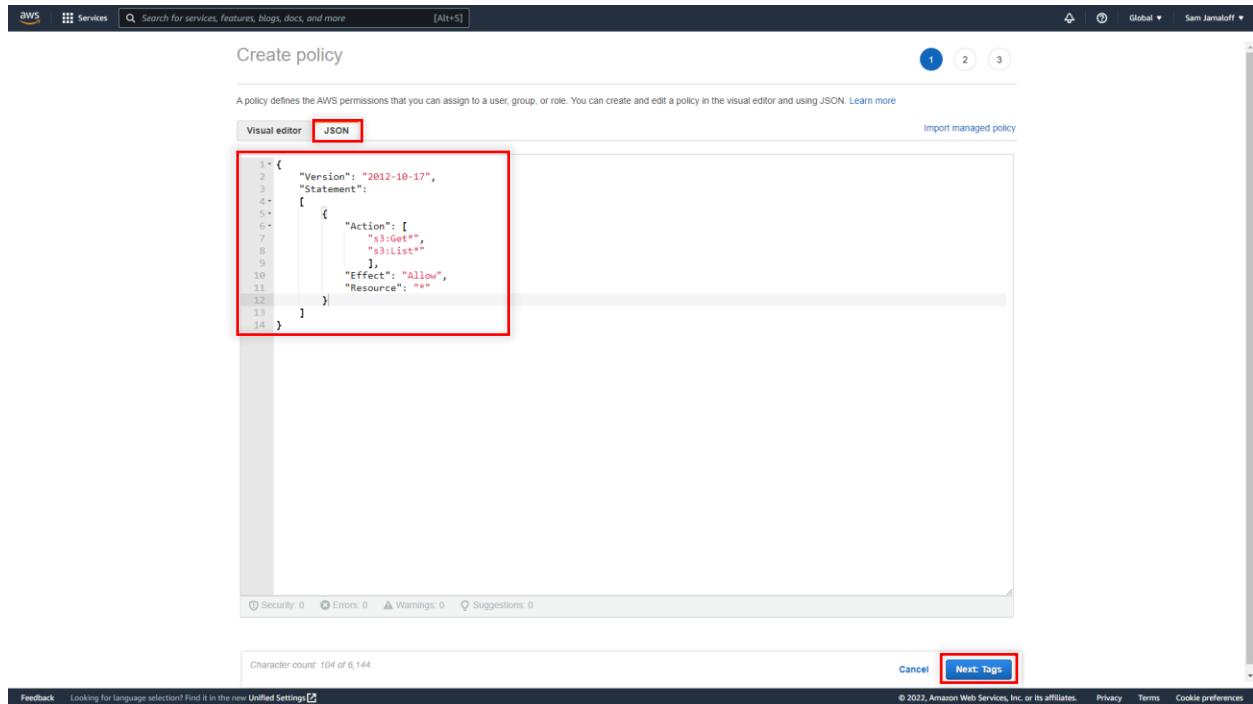
1. Sign-in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the IAM console, in the navigation pane, choose Policies, and then choose Create policy. (If a Get Started button appears, choose it, and then choose Create Policy.)



The screenshot shows the IAM Policies page with 975 policies listed. The 'Policies' link in the navigation bar is highlighted with a red box. The top right corner has a 'Create policy' button, which is also highlighted with a red box.

| Policy name                            | Type                       | Used as | Description   |
|--|----------------------------|---------|---|
| AWSDirectConnectReadOnlyAccess         | AWS managed                | None    | Provides read only access to AWS Direct Connect via the AWS Management Console.                                   |
| AmazonGlacierReadOnlyAccess            | AWS managed                | None    | Provides read only access to Amazon Glacier via the AWS Management Console.                                       |
| AWSMarketplaceFullAccess               | AWS managed                | None    | Provides the ability to subscribe and unsubscribe to AWS Marketplace software, allows users to manage Market...   |
| ClientVPNServiceRolePolicy             | AWS managed                | None    | Policy to enable AWS Client VPN to manage your Client VPN endpoints.  |
| AWSSSOIdentityAdministrator            | AWS managed                | None    | Administrator access for SSO Directory  |
| AWSIoT1ClickReadOnlyAccess             | AWS managed                | None    | Provides read only access to AWS IoT 1-Click.   |
| AutoScalingConsoleReadOnlyAccess       | AWS managed                | None    | Provides read-only access to Auto Scaling via the AWS Management Console.   |
| AmazonDMSRedshiftS3Role                | AWS managed                | None    | Provides access to manage S3 settings for Redshift endpoints for DMS.   |
| AWSQuickSightListIAM                   | AWS managed                | None    | Allow QuickSight to list IAM entities   |
| AWSHealthFullAccess                    | AWS managed                | None    | Allows full access to the AWS Health APIs and Notifications and the Personal Health Dashboard                     |
| AlexaForBusinessGatewayExecution       | AWS managed                | None    | Provide gateway execution access to AlexaForBusiness services   |
| AmazonElasticTranscoder_ReadOnlyAcc... | AWS managed                | None    | Grants users read-only access to Elastic Transcoder and list access to related services.                          |
| AmazonRDSFullAccess                    | AWS managed                | None    | Provides full access to Amazon RDS via the AWS Management Console.  |
| SupportUser                            | AWS managed - job function | None    | This policy grants permissions to troubleshoot and resolve issues in an AWS account. This policy also enables ... |
| AmazonEC2FullAccess                    | AWS managed                | None    | Provides full access to Amazon EC2 via the AWS Management Console.  |
| SecretsManagerReadWrite                | AWS managed                | None    | Provides read/write access to AWS Secrets Manager via the AWS Management Console. Note: this excludes IA...       |
| AWSToITThingsRegistration              | AWS managed                | None    | This policy allows users to register things at bulk using AWS IoT StartThingRegistrationTask API                  |
| AmazonDocDBReadOnlyAccess              | AWS managed                | None    | Provides read-only access to Amazon DocumentDB with MongoDB compatibility. Note that this policy also grant...    |
| AmazonMQApiFullAccess                  | AWS managed                | None    | Provides full access to AmazonMQ via our API/SDK  |
| AWSLambdaVirtuallyAllCloudWatchLogs    | AWS managed                | None    | Provides full access to CloudWatch Logs via the Lambda API  |

3. On the Create policy page, paste the following in the JSON tab:
4. { "Action": [ "s3:Get\*", "s3>List\*" ], "Effect": "Allow", "Resource": "\*" } into square brackets like it was shown in the picture

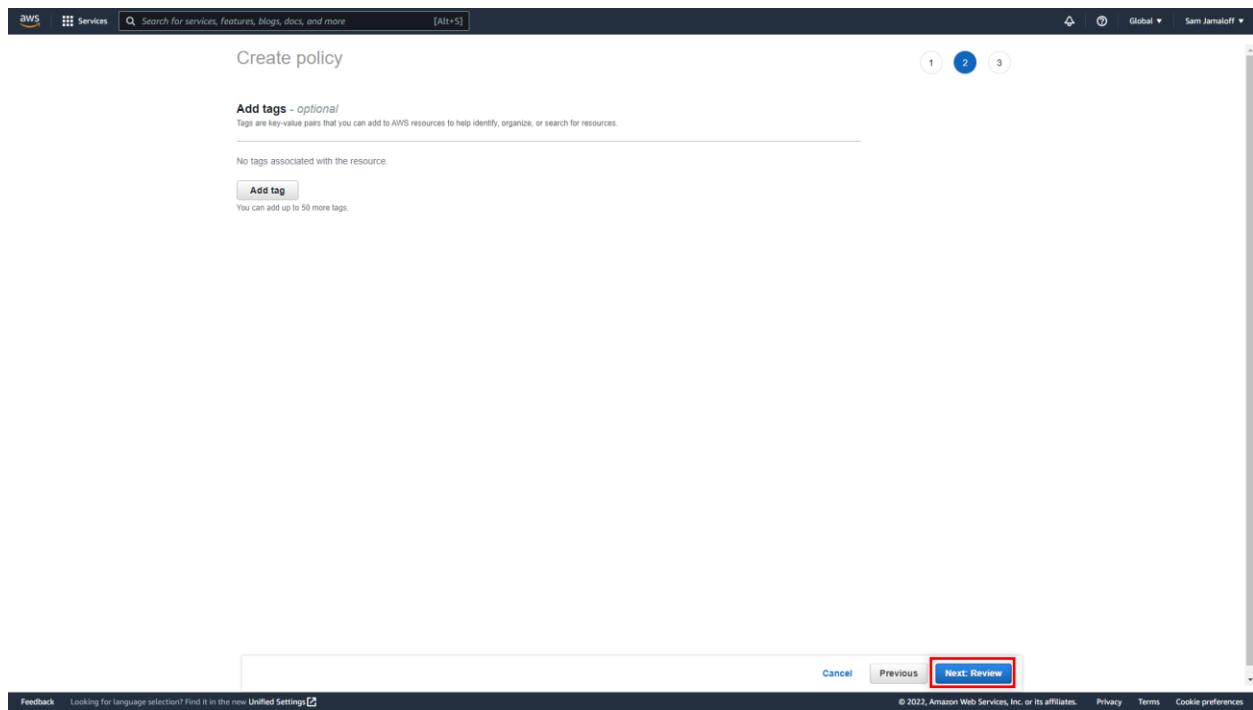


A screenshot of the AWS Create policy interface. The top navigation bar shows 'Services' and a search bar. Below it, a title 'Create policy' is followed by a progress bar with steps 1, 2, and 3. A note says 'A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more'. There are two tabs: 'Visual editor' and 'JSON', with 'JSON' selected and highlighted with a red box. The JSON code is displayed in a code editor:

```
1 "Version": "2012-10-17",
2 "Statement":
3 [
4 {
5     "Action": [
6         "s3:Get*",
7         "s3:List*"
8     ],
9     "Effect": "Allow",
10    "Resource": "*"
11 }
12 ]
13 }
```

The JSON code is highlighted with a red box. Below the code editor, there are status indicators: Security: 0, Errors: 0, Warnings: 0, and Suggestions: 0. At the bottom right, there are buttons for 'Cancel', 'Next: Tags' (highlighted with a red box), and 'Previous'.

5. Click tags, Do Not change anything in tags, and click “Next Review”



6. Give a name as **EC2CodeDeployPolicy** in the Policy Name Box. Then click Create Policy button.

Create policy

Review policy

Name\* EC2CodeDeployPolicy

Description

Summary

| Service                                      | Access level           | Resource      | Request condition |
|--|------------------------|---------------|-------------------|
| Allow (1 of 334 services) Show remaining 334 | Full List Limited Read | All resources | None              |

Tags

| Key                                   | Value |
|---------------------------------------|-------|
| No tags associated with the resource. |       |

\* Required

Cancel Previous Create policy

7. In the navigation pane, choose Roles, and then choose Create role

Identity and Access Management (IAM)

IAM > Roles

Roles (4) Info

An IAM role is an identity that you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

| Role name                       | Trusted entities                                  | Last activity  |
|---------------------------------|---|----------------|
| AWSServiceRoleForRDS            | AWS Service: rds (Service-Linked Role)            | 28 minutes ago |
| AWSServiceRoleForSupport        | AWS Service: support (Service-Linked Role)        | -              |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service-Linked Role) | -              |
| rds-monitoring-role             | AWS Service: monitoring rds                       | 2 days ago     |

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

|  |   |  |
|--|---|--|
|  |   |  |
| Access AWS from your non AWS workloads   | X.509 Standard  | Temporary credentials  |
| Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS. | Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities. | Use temporary credentials with ease and benefit from the enhanced security they provide. |

Feedback Looking for language selection? Find it in the new Unified Settings.

8. On the Create role page, choose AWS service, and from the Choose the service that will use this role list, choose EC2, and click next

Screenshot of the AWS IAM 'Create role' wizard Step 1: Select trusted entity.

The 'Trusted entity type' section shows three options:

- AWS service** (selected): Allows AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**: Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

The 'Use case' section shows two common use cases:

- EC2** (selected): Allows EC2 instances to call AWS services on your behalf.
- Lambda**: Allows Lambda functions to call AWS services on your behalf.

The 'Next Step' button is highlighted with a red box.

Screenshot of the AWS IAM 'Create role' wizard Step 2: Add permissions.

The 'Permissions policies' section shows a list of available policies:

| Policy name           | Type      | Description   |
|-----------------------|-----------|---|
| EC2CodeDeployPolicy   | Custom... | Provides read only access to AWS Direct Connect via the AWS Management Console.   |
| AWSDirectConnect...   | AWS m...  | Provides read only access to Amazon Direct Connect via the AWS Management Console.  |
| AmazonGlacierRea...   | AWS m...  | Provides read only access to Amazon Glacier via the AWS Management Console.   |
| AWSMarketplaceFu...   | AWS m...  | Provides the ability to subscribe and unsubscribe to AWS Marketplace software, allows users to manage Marketplace software instances from the Marketplace You...    |
| AWSSSODirectory...    | AWS m...  | Administrator access for SSO Directory.   |
| AWSIoT1ClickRead...   | AWS m...  | Provides read only access to AWS IoT 1-Click.   |
| AutoScalingConsol...  | AWS m...  | Provides read-only access to Auto Scaling via the AWS Management Console.   |
| AmazonDMSRedsh...     | AWS m...  | Provides access to manage S3 settings for Redshift endpoints for DMS.   |
| AWSQuickSightList...  | AWS m...  | Allow QuickSight to list IAM entities.  |
| AWSHealthFullAcces... | AWS m...  | Allows full access to the AWS Health APIs and Notifications and the Personal Health Dashboard.  |
| AlexaForBusinessG...  | AWS m...  | Provide gateway execution access to AlexaForBusiness services.  |
| AmazonElasticTran...  | AWS m...  | Grants users read-only access to Elastic Transcoder and list access to related services.  |
| AmazonRDSFullAc...    | AWS m...  | Provides full access to Amazon RDS via the AWS Management Console.  |
| SupportUser           | AWS m...  | This policy grants permissions to troubleshoot and resolve issues in an AWS account. This policy also enables the user to contact AWS support to create and mana... |
| AmazonEC2FullAcc...   | AWS m...  | Provides full access to Amazon EC2 via the AWS Management Console.  |
| SecretsManagerRe...   | AWS m...  | Provides read/write access to AWS Secrets Manager via the AWS Management Console. Note: this excludes IAM actions, so combine with IAMFullAccess if rotation ...    |
| AWSIoTThingsRegi...   | AWS m...  | This policy allows users to register things at bulk using AWS IoT StartThingRegistrationTask API.   |
| AmazonDynamoDBrea...  | AWS m...  | Provides read-only access to Amazon DynamoDB with ManagedDB compatibility. Note that this policy does not provide access to Amazon RDS and Amazon Neptune.          |

The 'Create policy' button is located at the top right of the policy list.

9. Choose the Permission that you recently created with the name **EC2CodeDeployPolicy** and click Next.
10. Give a name to your role as **EC2DeployRole** and click on Create role.

Screenshot of the AWS IAM 'Add permissions' step. A red box highlights the 'EC2CoreDeployPolicy' policy in the list.

**Add permissions**

**Permissions policies (Selected: 1/659) Info**

Choose one or more policies to attach to your new role.

**Filter policies by property or policy name and press enter:**

| Policy name (?)   | Type     | Description  |
|---|----------|--|
| <input checked="" type="checkbox"/> EC2CoreDeployPolicy | Custom   | Provides read only access to AWS Direct Connect via the AWS Management Console.  |
| <input type="checkbox"/> AWSDirectConnect...            | AWS m... | Provides read only access to AWS Direct Connect via the AWS Management Console.  |
| <input type="checkbox"/> AmazonGlacierRea...            | AWS m... | Provides read only access to Amazon Glacier via the AWS Management Console.  |
| <input type="checkbox"/> AWSMarketplaceFu...            | AWS m... | Provides the ability to subscribe and unsubscribe to AWS Marketplace software, allows users to manage Marketplace software instances from the Marketplace Your Software page, and provides administrative access to EC2. |
| <input type="checkbox"/> AWSSSODirectory...             | AWS m... | Administrator access for SSO Directory   |
| <input type="checkbox"/> AWSIoTClickRead...             | AWS m... | Provides read only access to AWS IoT Click   |
| <input type="checkbox"/> AutoScalingConse...            | AWS m... | Provides read only access to Auto Scaling via the AWS Management Console   |
| <input type="checkbox"/> AmazonMSKMobile...             | AWS m... | Provides access to manage 53 settings for Redshift endpoints for CMS.  |
| <input type="checkbox"/> AWSQuickSightList...           | AWS m... | Allow QuickSight to list IAM entities  |
| <input type="checkbox"/> AWSHealthHealthc...            | AWS m... | Allows full access to the AWS Health APIs and Notifications and the Personal Health Dashboard  |
| <input type="checkbox"/> AlexaForBusinessG...           | AWS m... | Provide gateway execution access to AlexaForBusiness services  |
| <input type="checkbox"/> AmazonElasticTran...           | AWS m... | Grants users read-only access to Elastic Transcoder and list access to related services  |
| <input type="checkbox"/> AmazonRDSFullAcc...            | AWS m... | Provides full access to Amazon RDS via the AWS Management Console  |
| <input type="checkbox"/> Supporter                      | AWS m... | This policy grants permissions to troubleshoot and resolve issues in an AWS account. This policy also enables the user to contact AWS support to create and manage cases   |
| <input type="checkbox"/> AmazonC2FullAcc...             | AWS m... | Provides full access to Amazon C2 via the AWS Management Console   |
| <input type="checkbox"/> SecretsManagerGet...           | AWS m... | Provides read/write access to AWS Secrets Manager via the AWS Management Console. Note: this excludes IAM actions, so combine with IAMFullAccess if rotation configuration is required.                                  |
| <input type="checkbox"/> AWSIoTThingRegis...            | AWS m... | This policy allows users to register things at bulk using AWS IoT ThingRegistrationTask API  |
| <input type="checkbox"/> AmazonDocDBRea...              | AWS m... | Provides read-only access to Amazon DocumentDB with MongoDB compatibility. Note that this policy also grants access to Amazon RDS and Amazon Neptune resources.  |
| <input type="checkbox"/> AmazonSQSFull...               | AWS m... | Provides full access to Amazon SQS via our API/SDK   |
| <input type="checkbox"/> AWSMediaStoreMedi...           | AWS m... | Provides read-only permissions for MediaStore APIs   |

**Set permissions boundary - optional** Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

**Next Step**

Screenshot of the AWS IAM 'Name, review, and create' step. A red box highlights the 'Role name' input field containing 'EC2DeployRole'.

**Name, review, and create**

**Role details**

**Role name**

Enter a meaningful name to identify this role.

**EC2DeployRole**

Maximum 64 characters. Use alphanumeric and '-' '\_' characters

**Description**

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

**Step 1: Select trusted entities**

```

1+ []
  "Version": "2012-10-17",
2+   "Statement": [
3+     {
4+       "Effect": "Allow",
5+       "Action": [
6+         "sts:AssumeRole"
7+       ],
8+       "Principal": [
9+         {
10+           "Service": [
11+             "ec2.amazonaws.com"
12+           ]
13+         }
14+       ]
15+     }
16+   ]

```

**Step 2: Add permissions**

**Permissions policy summary**

**Cancel** **Previous** **Next Step**

The screenshot shows the AWS IAM Role Creation Wizard. On the left, a navigation pane lists 'Identity and Access Management (IAM)' and various management options like User groups, Users, Roles, Policies, and Access reports. The main area is titled 'Step 1: Select trusted entities' and displays a JSON policy document:

```

1+ [{
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": [
10      "service",
11      "ec2.amazonaws.com"
12    ]
13  }
14]
15]
16

```

Below this, 'Step 2: Add permissions' is shown with a table for 'Permissions policy summary'. It lists a single policy named 'EC2CodeDeployPolicy' which is 'Customer managed' and attached as a 'Permissions policy'. A 'Tags' section follows, with an 'Add tag' button and a note about adding up to 50 more tags. At the bottom right, there are 'Cancel', 'Previous', and 'Create role' buttons, with 'Create role' being highlighted.

## Step 5 Provision an Instance

In this step, you will create or configure the instance that you will deploy the sample application. You can deploy to an amazon EC2 instance or an on-premises instance that is running one of the operating systems, supported by CodeDeploy. We will use Windows server 2016 for this seminar.

In one of the previous seminars, we have already considered creating an instance. To quickly create a new instance for this seminar, use the Amazon EC2 console. Choose Windows Server 2016 Base

1. Sign-in to your AWS console and open the Amazon EC2 console.
2. In the navigation pane choose Instances, and then choose Launch Instance
3. Locate the operating system and Version you want to use and then choose Select. (Make sure it is Free-Tier Eligible) In our case it is Windows Server 2016 Base.

**Name and tags**

Name: **CodeDeployVM**

**Application and OS Images (Amazon Machine Image)**

An AMI is a template that contains the software configuration (operating system, application server, and application) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

**AMI from catalog** Quick Start

Amazon Machine Image (AMI)

Windows\_Server-2016-English-Full-BASE-2022.09.14.ami-0b3e423be2b1ed909

| Catalog         | Published               | Architecture | Virtualization | Root device type | ENI Enabled |
|-----------------|-------------------------|--------------|----------------|------------------|-------------|
| Quickstart AMIs | 2022-09-14T18:38:16.00Z | x86_64       | hvm            | ebs              | Yes         |

**Instance type**

**Summary**

Number of instances: **1**

Software Image (AMI): Microsoft Windows Server 2016 ...

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel      Launch instance

- Select the key pair that you currently have access to. The one that was created at the beginning of our seminar.

**Instance type**

**t2.micro** Free tier eligible

Family: t2 1 vCPU 1 GiB Memory  
On-Demand Linux pricing: 0.0116 USD per Hour  
On-Demand Windows pricing: 0.0162 USD per Hour

**Key pair (login)**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name: **MyKeyPairSeminar5**

Create new key pair

**Network settings**

Network: vpc-0e93ca42c0d7194f  
Subnet: No preference (Default subnet in any availability zone)  
Auto-assign public IP: Info  
Enable  
Firewall (security groups): Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group     Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

Allow RDP traffic from Anywhere  
Helps you connect to your instance  
0.0.0.0/0

Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

**Summary**

Number of instances: **1**

Software Image (AMI): Microsoft Windows Server 2016 ...

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel      Launch instance

5. In network settings pane choose an option for creating new security group for your instance. Select “Allow RDP traffic from”, “Allow HTTPs” and “Allow HTTP”. Make sure that is available from anywhere.

**Network settings**

**Create security group**

**Allow RDP traffic from Anywhere**

**Allow HTTPs traffic from the internet**

**Allow HTTP traffic from the internet**

**Summary**

Number of instances: 1

Software Image (AMI): Microsoft Windows Server 2016 ... [read more](#)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GB

**Free tier:** in your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage, free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Launch Instance**

6. Leave the storage configuration as it is.

**Configure storage**

**Add new volume**

**Advanced details**

**Purchasing option**

**Domain join directory**

**IAM instance profile**

**Hostname type**

**DNS Hostname**

**Instance auto-recovery**

**Launch Instance**

7. In advanced Details there is IAM instance profile make sure to choose the one that was created previously.

The screenshot shows the 'Advanced details' section of the AWS EC2 instance configuration. A red box highlights the 'IAM instance profile' dropdown, which contains the value 'EC2DeployRole arn:aws:iam::414654816993:instance-profile/EC2DeployRole'. This step ensures the correct IAM role is assigned to the instance.

8. Click launch instance.

The screenshot shows the 'Instances (1) info' section of the AWS EC2 dashboard. A red box highlights the 'Launch instances' button, which is used to start the newly created EC2 instance.

9. Now connect to our newly created instance using RDP connection like we did in the seminar 1.

10. On your remote server install we will be installing CodeDeploy manually.

- Open PowerShell and type the following.

- b. wget -Uri {your\_area\_based\_local\_installation\_link} -OutFile "C:\Users\Administrator\Downloads\codedeploy-agent.msi"  
 your area based installation link can be found [here](#).

**Example:** wget -Uri https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/codedeploy-agent.msi -OutFile "C:\Users\Administrator\Downloads\ codedeploy-agent.msi"

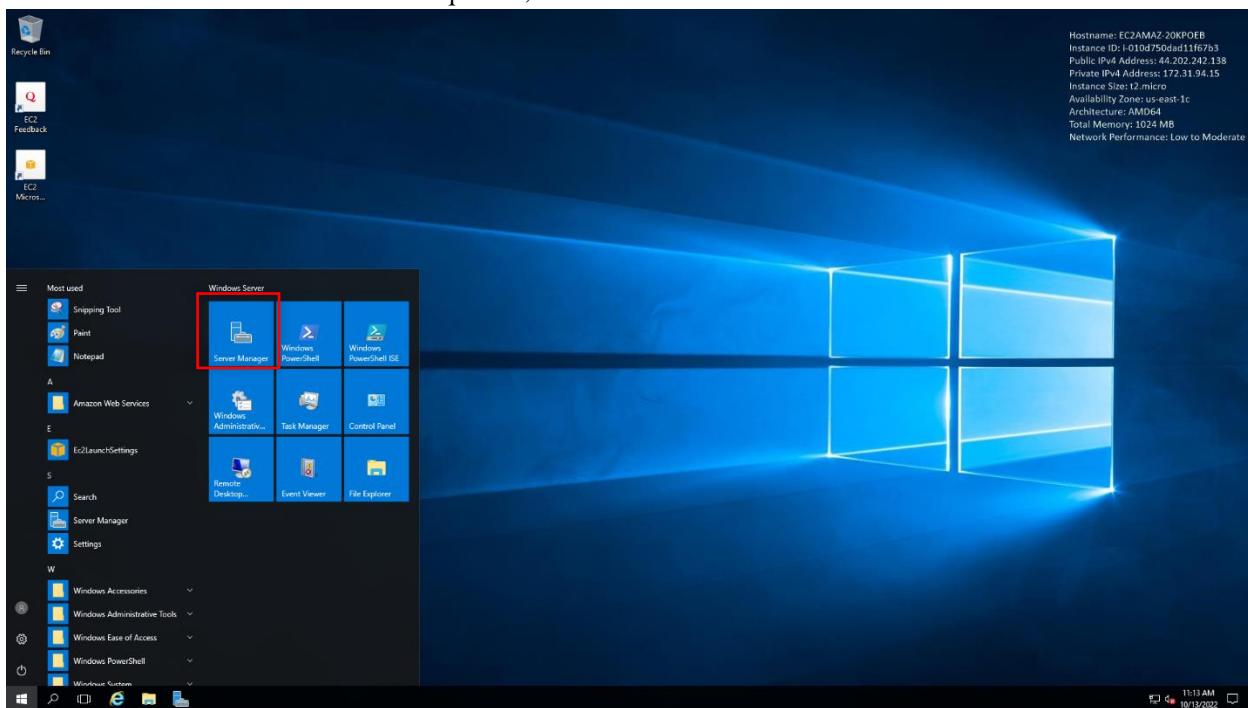
- c. Go to the location that file was downloaded and double click into file to install.  
 11. To check CodeDeploy agent working run following command in PowerShell.

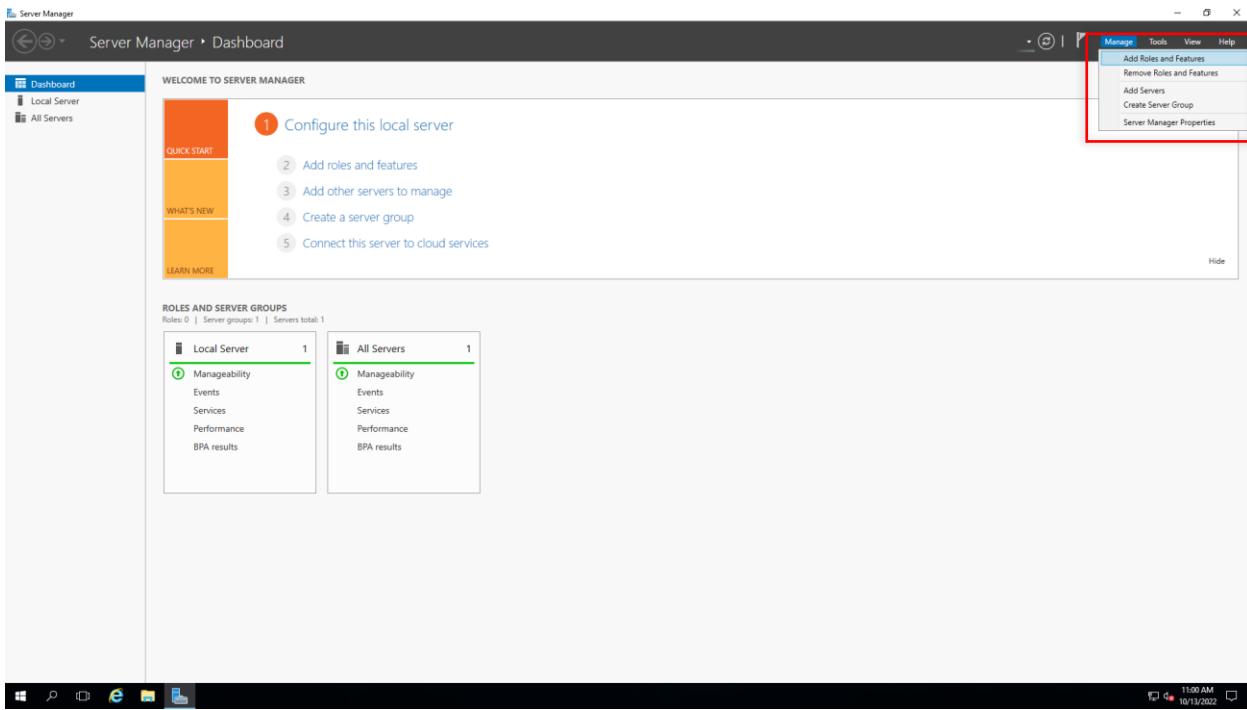
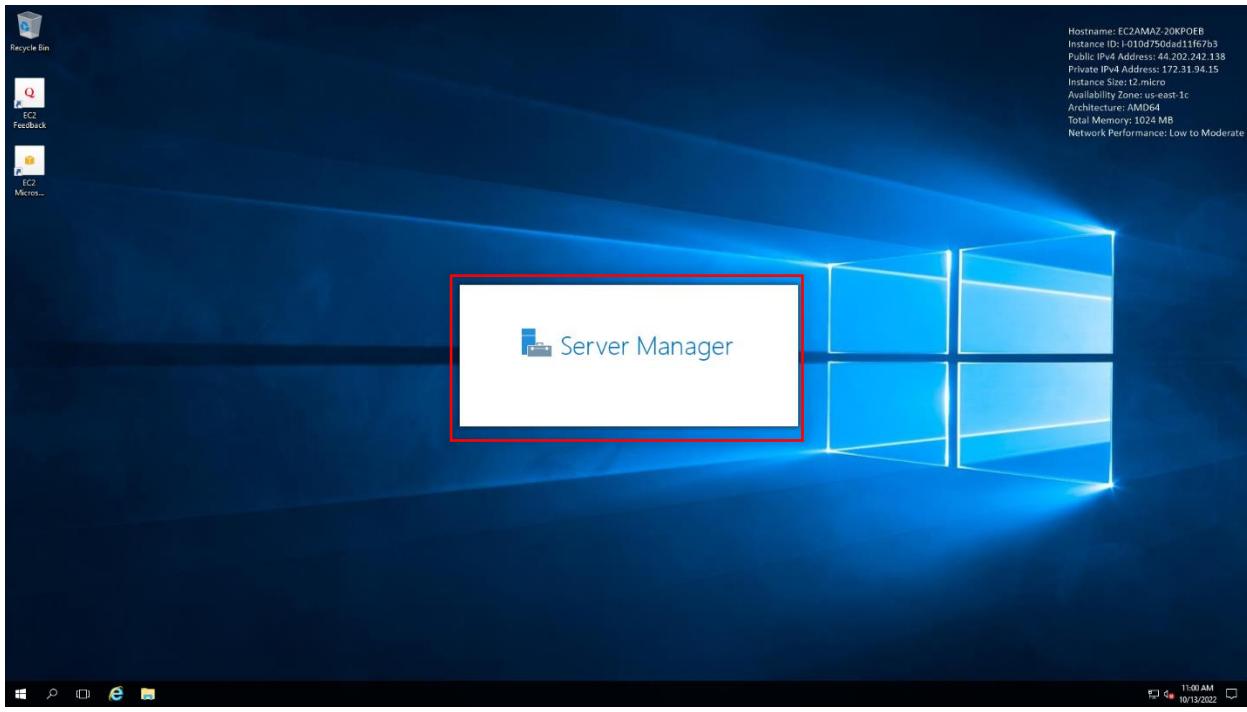
**powershell.exe -Command Get-Service -Name codedeployagent**

## Step 6: Enabling IIS and required IIS components on Windows Server 2016

For our lab we need that IIS be enable and specific IIS components be enabled on Windows Server 2016. Follow the steps that are given below in order to run and configure the IIS components in Windows 2016 Base.

1. Open Server Manager and click Manage > Add Roles and Features. Click Next.
2. Select Role-based or feature-based installation and click Next.
3. Select the appropriate server. The local server is selected by default. Click Next.
4. Enable Web Server (IIS) and click Next.
5. No additional features are necessary to install the Web Adaptor, so click Next.
6. On the Web Server Role (IIS) dialog box, click Next.
7. On the Select role services dialog box, verify that the web server components listed below are enabled. Click Next.
8. Verify that your settings are correct and click Install.
9. When the installation completes, click Close to exit the wizard.





## Web Server

### Common HTTP Features

#### Default Document

#### Static Content

### Security

#### Request Filtering

#### Basic Authentication

#### Windows Authentication

### Application Development

#### .NET Extensibility 4.6 .NET

#### Extensibility ASP.NET 4.6 ASP.NET

#### ISAPI Extensions

#### ISAPI Filters

## Management Tools

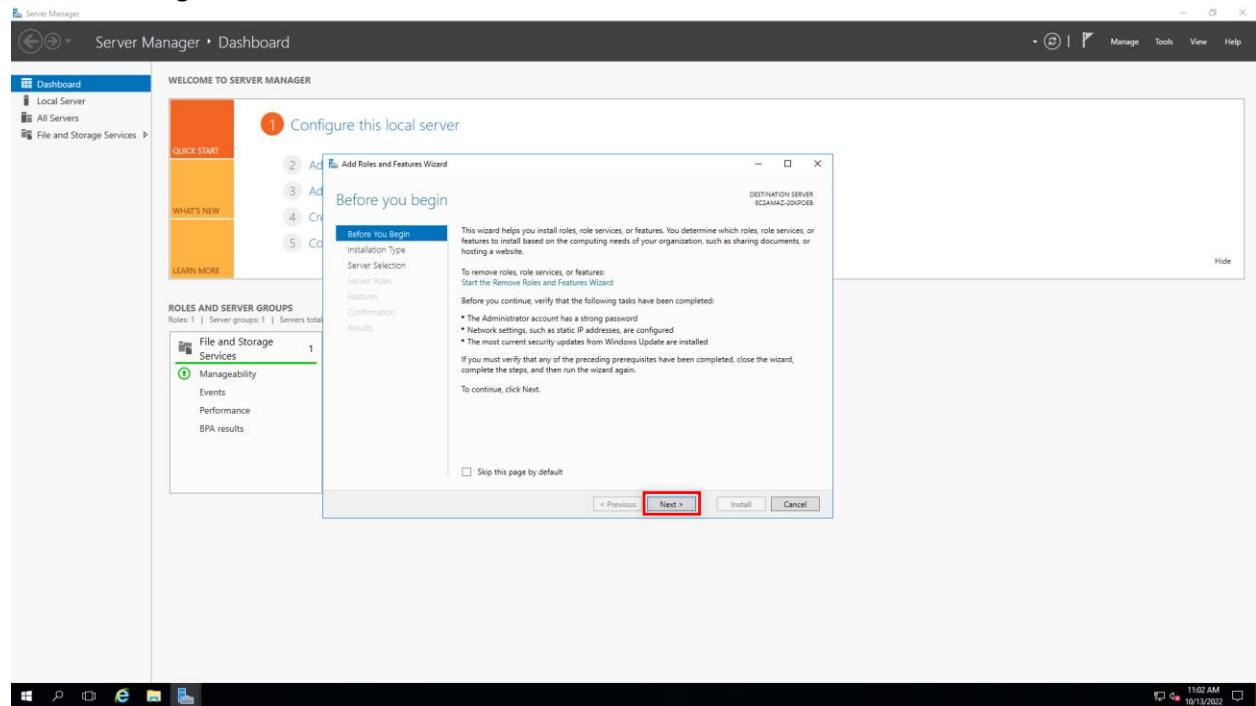
### IIS Management Console

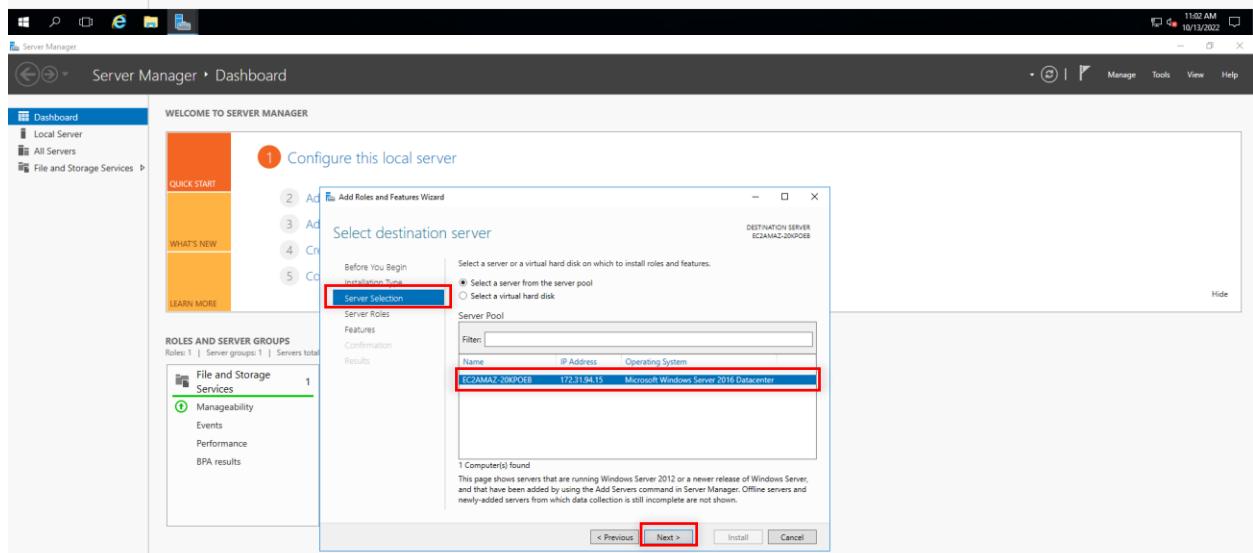
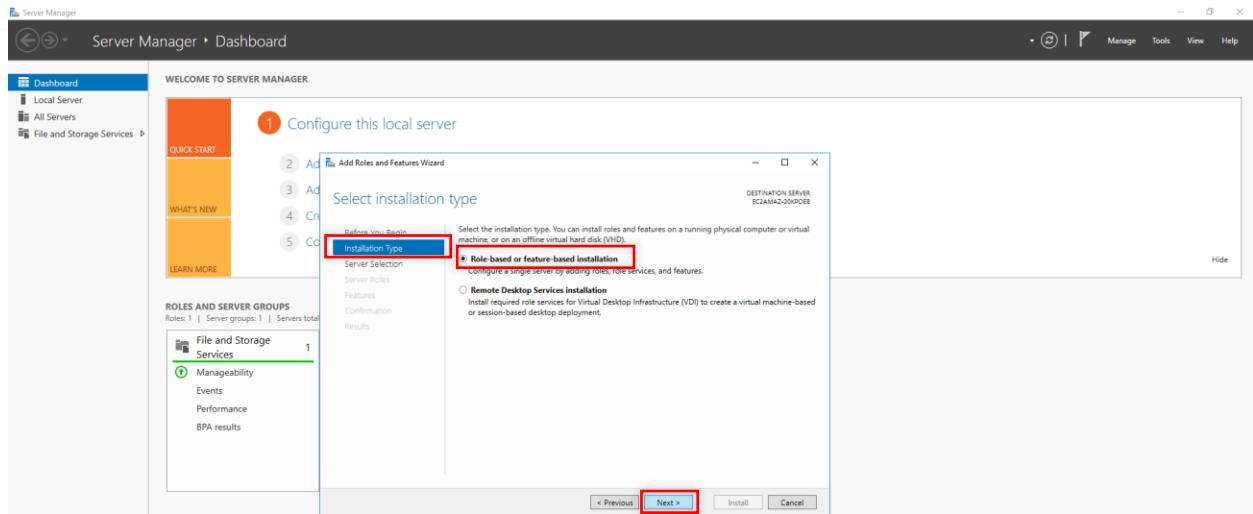
### IIS 6 Management Compatibility

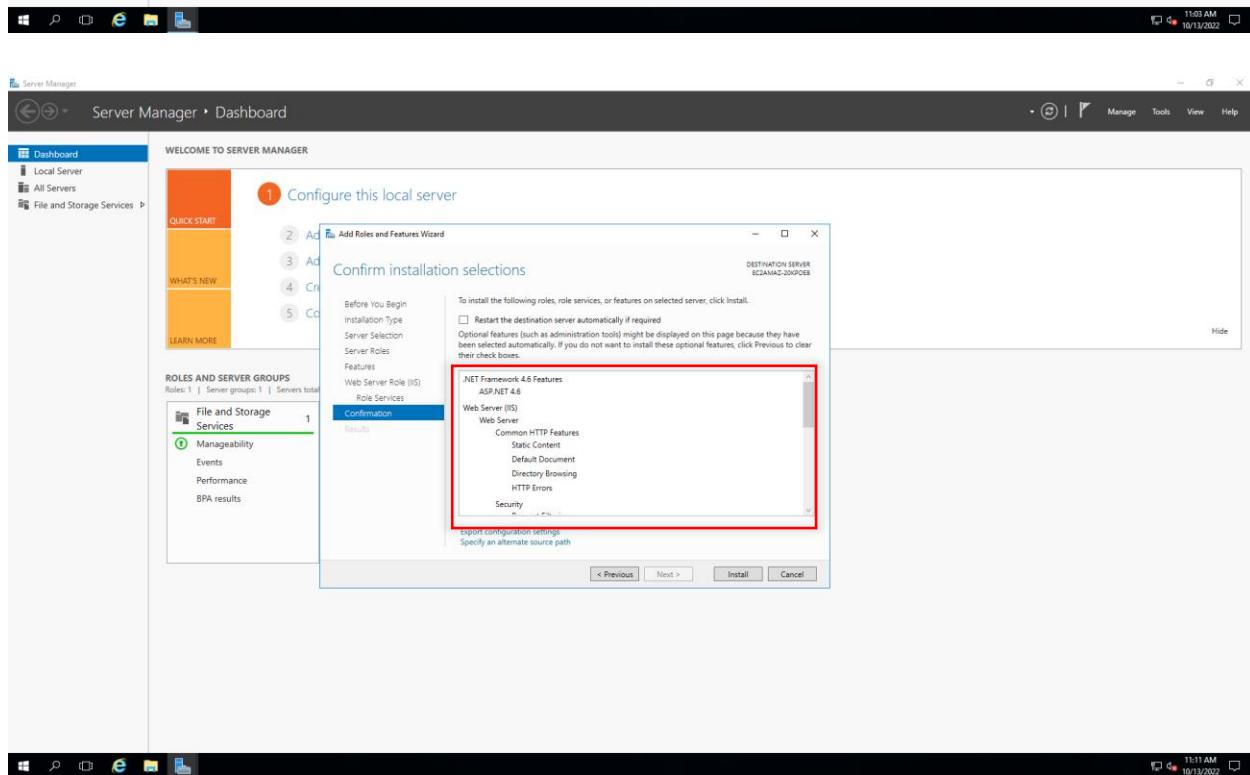
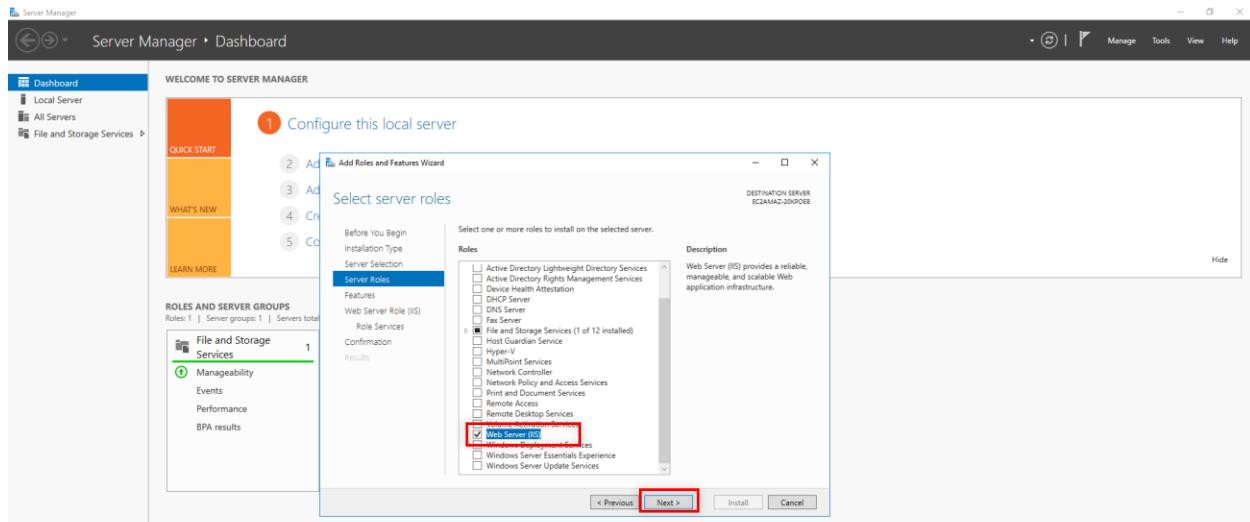
### IIS 6 Metabase Compatibility

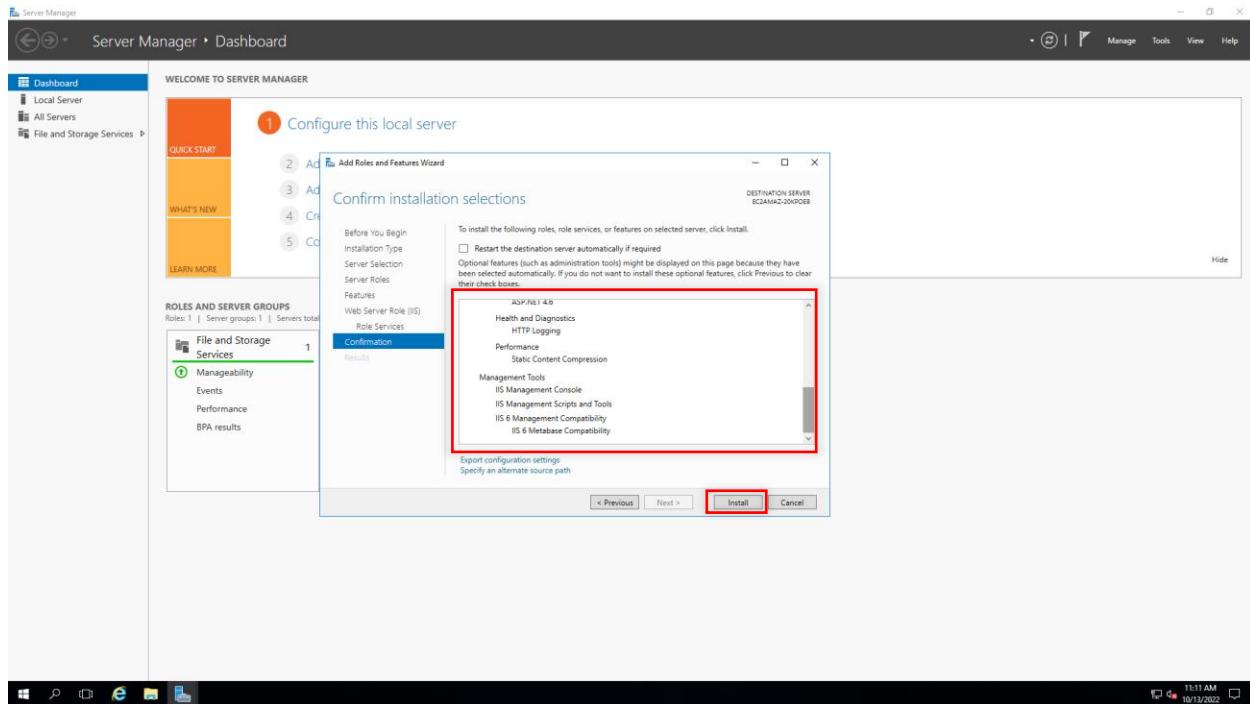
### IIS Management Scripts and Tools

### Management Service



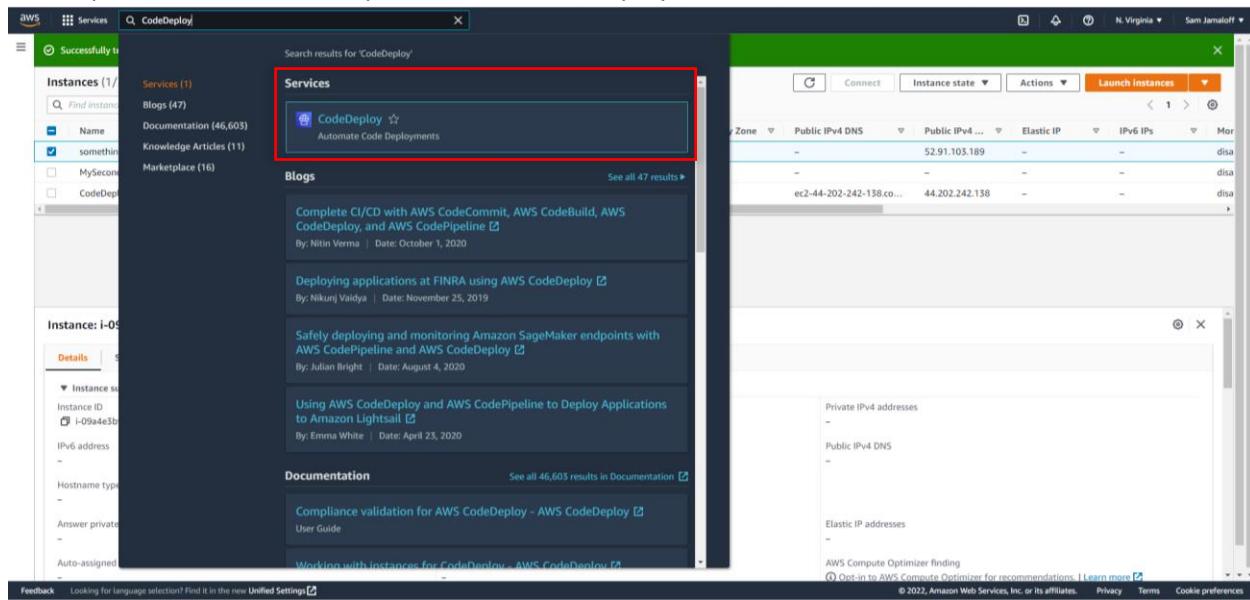






## Step 7: Enter the CodeDeploy Console

Click the home icon on the upper left corner of the AWS Management Console. Find CodeDeploy under Developer Tools and click to open the AWS CodeDeploy Console.



## Step 8. Create a Service Role for CodeDeploy

In AWS, service roles are used to grant permissions to an AWS service so it can access AWS resources. The policies that you attach to the service role determine which AWS resources the service can access and what it can do with those resources.

The service role you create for CodeDeploy must be granted the permissions required for your computer platform. If you deploy to more than one compute platform, create one service role for each. To add permissions, attach one or more of the following AWS-supplied policies: For EC2/On-Premises deployments, attach the AWSCodeDeployRole policy. It provides the permissions for your service role to:

- a. Read the tags on your instances or identify your Amazon EC2 instances by Amazon EC2 Auto Scaling group names.
- b. Read, create, update, and delete Amazon EC2 Auto Scaling groups, lifecycle hooks, and scaling policies.
- c. Publish information to Amazon SNS topics.
- d. Retrieve information about CloudWatch alarms.
- e. Read and update Elastic Load Balancing.

1. Go to the aws` IAM section
2. In the navigation pane chose roles and then chose Create Role
3. On the Create Role Page, choose AWS service, and from the Choose the service that will use the role list choose Code Deploy and click Next.
  - a. From Select your use case, choose your use case:
    - i. For EC2/On-Premises deployments, choose CodeDeploy.
    - ii. For Amazon ECS deployments, choose CodeDeploy - ECS.
    - iii. For AWS Lambda deployments, choose CodeDeploy for Lambda.

Screenshot of the AWS IAM 'Create role' wizard Step 1: Select trusted entity. The 'AWS service' option is selected and highlighted with a red box. The 'Use cases' dropdown is also highlighted with a red box, showing 'CodeDeploy' selected. The 'Next Step' button is at the bottom right.

4. Leave everything as it is and click next.

The screenshot shows the 'Add permissions' step of the IAM role creation wizard. A red box highlights the 'Permissions policies (1)' section, which contains a single policy named 'AWSCodeDeployRole'. Below this, there's an optional 'Set permissions boundary' section. At the bottom right, the 'Next Step' button is highlighted with a red box.

5. Give a Role Name as CodeDeployServiceRole and scroll down to finish creation.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. A red box highlights the 'Role name' field, which contains 'CodeDeployServiceRole'. Below it is a 'Description' field with placeholder text. The 'Step 1: Select trusted entities' section at the bottom contains a large block of JSON policy code. The 'Edit' button is visible to the right of the policy editor.

```

1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Sid": "",
5         "Effect": "Allow",
6         "Principal": "*",
7         "Action": [
8             "sts:AssumeRole"
9         ],
10        "Service": [
11            "codedeploy.amazonaws.com"
12        ]
13    }
14 ]
15 ]
16 ]
17 []

```

Step 2: Add permissions

Permissions policy summary

| Policy name       | Type        | Attached as        |
|-------------------|-------------|--------------------|
| AWSCodeDeployRole | AWS managed | Permissions policy |

Tags

Add tags - Optional Info

No tags associated with the resource.

Add tag You can add up to 50 more tags.

Cancel Previous Create role

## Step 9: Create an Application and Deployment Group

In this step, we will use the CodeDeploy console to create an application and deployment group to use to deploy the sample revision from your GitHub repository.

- From Step 7: Go to Developer Tools > CodeDeploy > Applications.  
Click Create application Button.

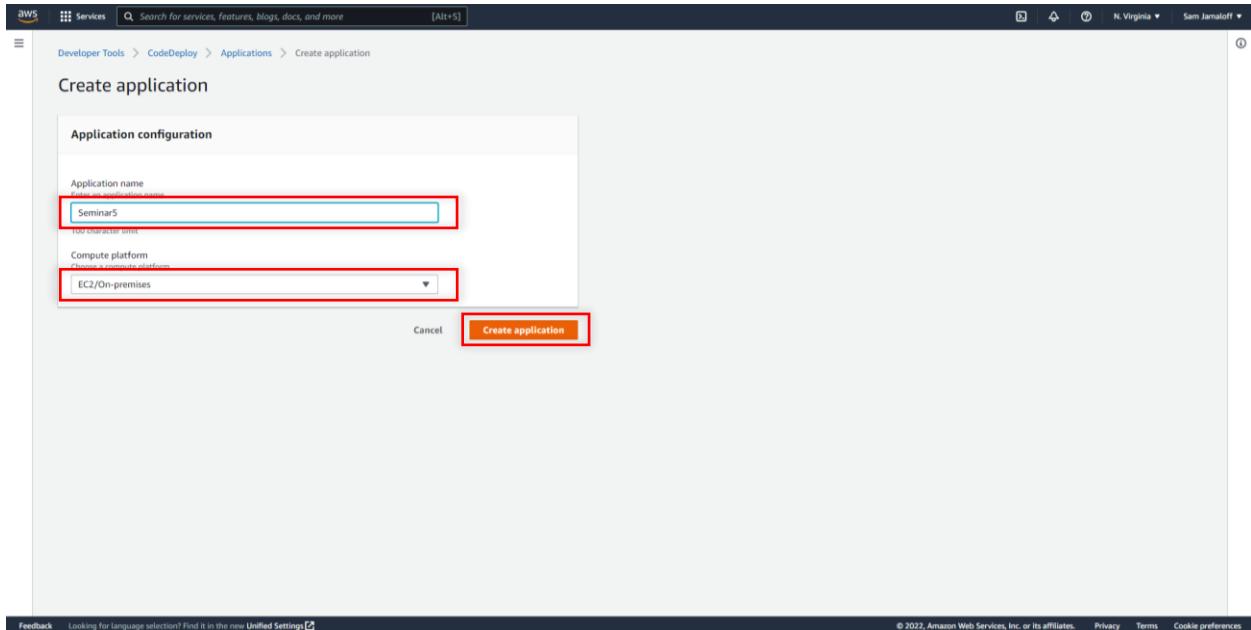
Developer Tools > CodeDeploy > Applications

Create application

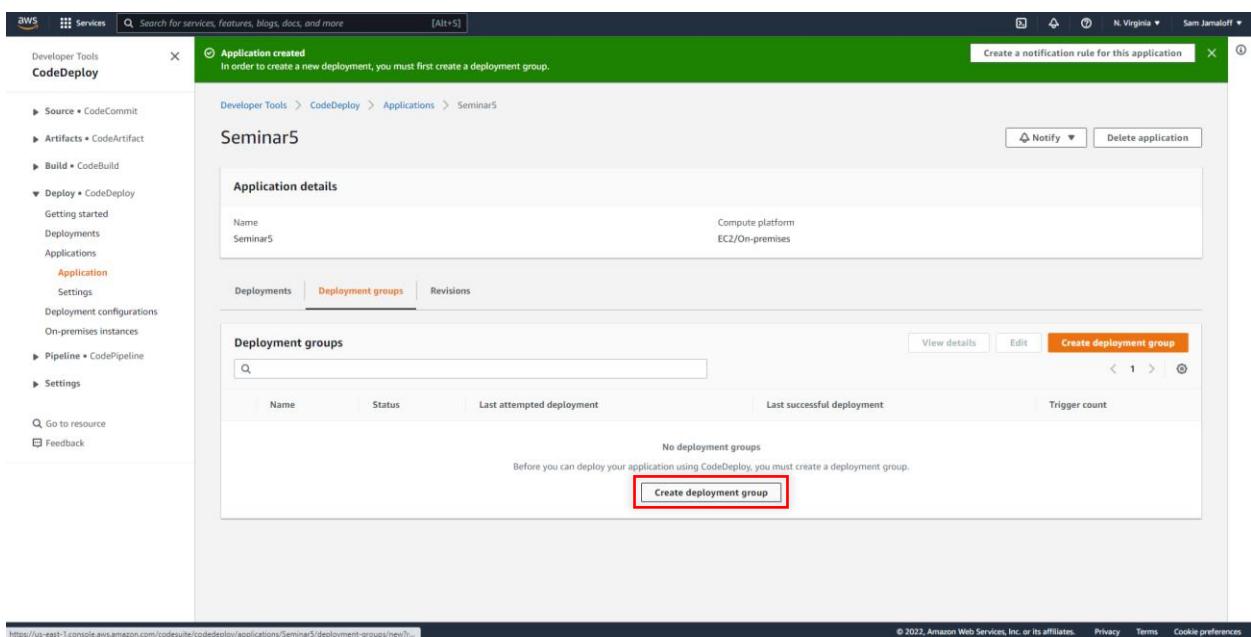
| Application name                               | Compute platform | Created |
|--|------------------|---------|
| No results<br>There are no results to display. |                  |         |

Feedback Looking for language selection? Find it in the new Unified Settings ?

- In the Application name enter Seminar5.  
For Compute Platform choose EC2/On-Premises  
Click Create application.



### 3. On Deployment Groups click Create Deployment Group



4. In Deployment Group creation window give a name as CodeDeployTestGrp  
In Service Role choose CodeDeploy  
In Deployment Type, choose in-Place.

Create deployment group

**Application**

Application  
Seminar  
Compute type  
EC2/On-premises

**Deployment group name**

Enter a deployment group name  
CodeDeployTestGrp

**Service role**

Enter a service role  
arn:aws:iam:414634816993:role/CodeDeployServiceRole

**Deployment type**

Choose how to deploy your application

In-place  
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update.

Blue/green  
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, traffic from the original environment is redirected to the new instances.

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- In Environment configuration, depending on the type of instance you are using, choose Amazon EC2 instances or On-premises instances. For Key and Value, enter the instance tag key and value that was applied to your instance as part of Provision an Instance step.

**Environment configuration**

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

Amazon EC2 Auto Scaling groups

Amazon EC2 instances  
1 unique matched instance. Click here for details

You can add up to three groups of tags for EC2 instances to this deployment group.  
One tag group: Any instance identified by the tag group will be deployed to.  
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

**Tag group 1**

|        |                  |
|--------|------------------|
| Key    | Value - optional |
| Q Name | Q CodeDeployVM   |

Add tag  
+ Add tag group

On-premises instances

You can add up to three groups of tags for EC2 instances to this deployment group.  
One tag group: Any instance identified by the tag group will be deployed to.  
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

**Tag group 1**

|     |                  |
|-----|------------------|
| Key | Value - optional |
|     |                  |

Add tag  
+ Add tag group

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- In Deployment configuration, choose CodeDeployDefault.AllAtOnce.
- In Load Balancer, clear Enable load balancing.
- Expand Advanced.
- In Alarms, select Ignore alarm configuration.
- Choose Create deployment group and continue to the next step.

The screenshot shows the 'Deployment settings' page in the AWS CodeDeploy console. At the top, there's a search bar and navigation links. The main area is titled 'Deployment settings' and contains several sections:

- Deployment configuration:** A dropdown menu showing 'CodeDeployDefault.AllAtOnce' with a red box around it, followed by a button 'Create deployment configuration'.
- Load balancer:** A section with a checkbox 'Enable load balancing' which is unchecked.
- Advanced - optional:** A section with a red box around it, containing 'Triggers' and 'Alarms' sub-sections. Both are currently empty.
- Feedback:** A note at the bottom left: 'Looking for language selection? Find it in the new [Unified Settings](#)'.
- Footer:** Standard AWS footer with links to 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the 'Create deployment group' page in the AWS CodeDeploy console. It has several sections:

- Triggers:** A section stating 'No triggers have been created for this deployment group.'
- Alarms:** A section with a red box around it, containing a 'Name' field and a note: 'No Amazon CloudWatch alarms have been created for this deployment group'. It also includes checkboxes for 'Ignore alarm configuration' and 'Continue deployment even if alarm status is unavailable'.
- Rollbacks:** A section with checkboxes for 'Enable deployment rollbacks for this deployment group', 'Roll back when a deployment fails', 'Roll back when alarm thresholds are met', and 'Disable rollbacks' (which is checked).
- Deployment group tags:** A section with a 'Add tag' button.
- Buttons:** 'Cancel' and a large red 'Create deployment group' button.
- Feedback:** A note at the bottom left: 'Looking for language selection? Find it in the new [Unified Settings](#)'.
- Footer:** Standard AWS footer with links to 'Privacy', 'Terms', and 'Cookie preferences'.

## Step 10: Monitor and Verify the Deployment

In this step, you will use the CodeDeploy console to verify the success of the deployment. You will use your web browser to view the web page that was deployed to the instance you created or configured.

1. On the Deployment group details page, choose Create deployment.

The screenshot shows the AWS CodeDeploy console with the path: Developer Tools > CodeDeploy > Applications > Seminar5 > CodeDeployTestGrp. The 'Deployment group details' section is visible, showing the deployment group name 'CodeDeployTestGrp', application name 'Seminar5', service role ARN 'arn:aws:iam::414634816993:role/CodeDeployServiceRole', and deployment configuration 'CodeDeployDefault.AllAtOnce'. The 'Create deployment' button is highlighted with a red box.

2. In Deployment group, choose CodeDeployTestGrp.

In Revision type, choose GitHub.

In Connect to GitHub, do one of the following:

- a. To create a connection for CodeDeploy applications to a GitHub account, sign out of GitHub in a separate web browser tab. In GitHub account, enter a name to identify this connection, and then choose Connect to GitHub. The webpage prompts you to authorize CodeDeploy to interact with GitHub for the application named CodeDeployTestApp.
- b. To use a connection you have already created, in GitHub account, select its name, and then choose Connect to GitHub.
- c. To create a connection to a different GitHub account, sign out of GitHub in a separate web browser tab. Choose Connect to a different GitHub account, and then choose Connect to GitHub.

Follow the instructions on the Sign in page to sign in with your GitHub account.

On the Authorize application page, choose Authorize application

The screenshot shows two windows side-by-side. On the left is the 'Create deployment' wizard in the AWS Management Console. It has several sections: 'Deployment settings' (Application: Seminar5, Deployment group: Q\_CodeDeployTestGrp, Compute platform: EC2/On-premises, Deployment type: In-place, Revision type: My application is stored in Amazon S3), 'GitHub token name' (Q\_Seminar5), and 'Deployment description' (Deployment description - optional). A red box highlights the 'My application is stored in GitHub' section and the 'Q\_Seminar5' input field. On the right is a 'Authorize application - Google Chrome' window from GitHub. It displays 'Authorize AWS CodeDeploy (N. Virginia)' and a message from 'AWS CodeDeploy (N. Virginia) by aws-codedeploy' asking for permission to access the user's account. It includes 'Repositories' (Public and Private), 'Organization access' (hy-dcc), and a 'Request' button. A red box highlights the 'Authorize aws-codedeploy' button.

For the next step you will need last commit ID of your repository.  
Go to your repository and find short commit ID as shown in the picture. Click on that.

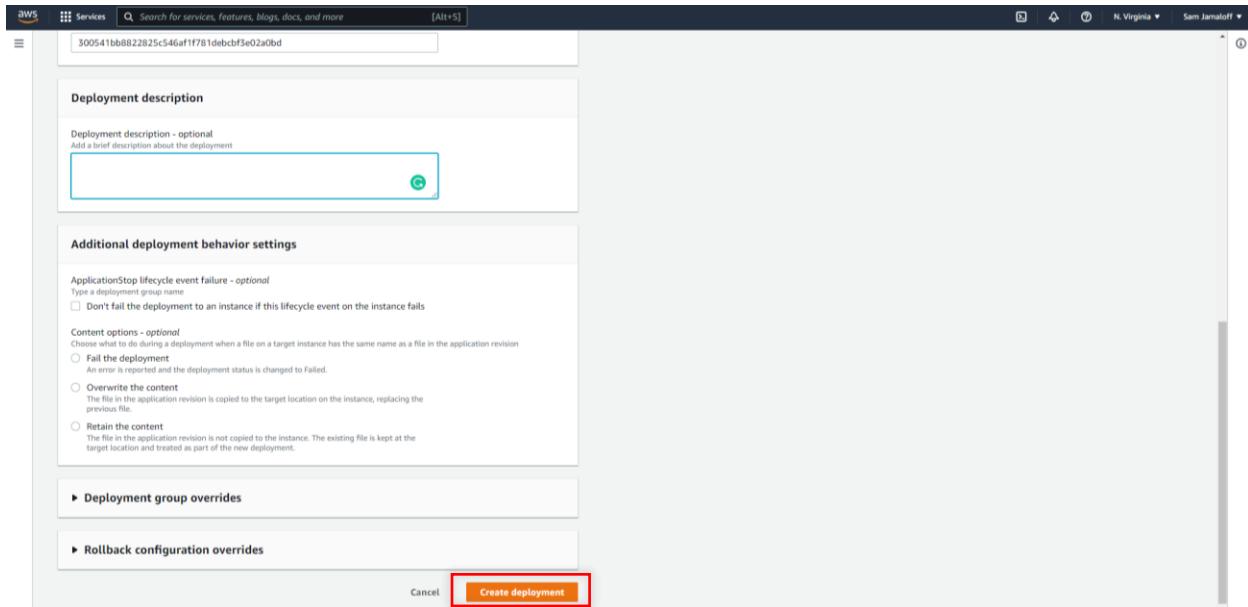
The screenshot shows a GitHub repository page for 'IlkhomjonUgli / CodeDeployGitHubDemo'. The 'Code' tab is selected, showing the 'main' branch with 1 commit. The commit list includes: 'IlkhomjonUgli init: MyFirst Commit' (308541b, 9 hours ago). A red box highlights the commit hash '308541b'. Other files listed include css, fonts, img, LICENSE, appspec.yml, deletewebsite.bat, deletewebsite.ps1, index.html, installwebsite.bat, and installwebsite.ps1, all with the same timestamp. The page also shows sections for 'About' (No description, website, or topics provided, MIT license, 0 stars, 1 watching, 0 forks), 'Releases' (No releases published, Create a new release), 'Packages' (No packages published, Publish your first package), and 'Languages' (CSS 36.1%, HTML 34.0%, PowerShell 18.4%, Batchfile 11.5%). At the bottom, there are links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

A new screen will open and from find the long version of git Commit and Copy it to the Clipboard.

The screenshot shows a GitHub repository page for 'IlkhomjonUgli / CodeDeployGitHubDemo'. The commit history is displayed, with the first commit being 'init: MyFirst Commit' by 'IlkhomjonUgli' 9 hours ago. The commit message and its long SHA-1 hash ('300b41bb822b25c546a1f781debcf3e02adbd') are highlighted with a red box.

3. On the CodeDeploy Create deployment page, in Repository name, enter the GitHub username you used to sign in, followed by a forward slash (/), followed by the name of the repository where you pushed your application revision (for example, my-github-username/CodeDeployGitHubDemo)

The screenshot shows the 'Create deployment' wizard in the AWS Management Console. The 'Deployment group' is set to 'CodeDeployTestGrp'. Under 'Revision type', the 'My application is stored in GitHub' option is selected. In the 'GitHub token name' section, a token named 'Seminar5' is connected. The 'Repository name' field contains 'IlkhomjonUgli/CodeDeployGitHubDemo' and the 'Commit ID' field contains '300b41bb822b25c546a1f781debcf3e02adbd', both of which are highlighted with red boxes.



## Step 11: Monitor and Verify the Deployment

In this step, you will use the CodeDeploy console to verify the success of the deployment. You will use your web browser to view the web page that was deployed to the instance you created or configured.

Copy Public IPV4 DNS and paste it in the new tab of any browser.

| Instance summary for i-010d750dad11f67b3 (CodeDeployVM)  |   |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
|--|---|---------------|--|-------------------------|---|------------|----------|------------------|---------|----------|--|------------------------|----------|-----------------|----------|-------------|--|--------------|---|------------------------|---------|-----------|--------|-------------------------|----------|------------------|---|---------------|-------------------|-------------------------|---|
| Instance ID  | i-010d750dad11f67b3 (CodeDeployVM)  |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| IPv6 address   | -   |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| Hostname type  | Private IP only   |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| IP name  | ip-172-31-94-15.ec2.internal  |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| Answer private resource DNS name   | IPv4 (A)  |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| Auto-assigned IP address   | 44.202.242.138 [Public IP]  |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| IAM Role   | ECDDeployRole   |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| <b>Details</b>   | <b>Security</b>   <b>Networking</b>   <b>Storage</b>   <b>Status checks</b>   <b>Monitoring</b>   <b>Tags</b> |               |  |                         |   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| <b>Instance details</b> <table border="1"> <tr> <td>Platform</td> <td>Windows</td> <td>AMI ID</td> <td>ami-0b3e423be2b1ed909</td> <td>Monitoring</td> <td>disabled</td> </tr> <tr> <td>Platform details</td> <td>Windows</td> <td>AMI name</td> <td>Windows_Server-2016-English-Full-Base-2022.09.14</td> <td>Termination protection</td> <td>Disabled</td> </tr> <tr> <td>Stop protection</td> <td>Disabled</td> <td>Launch time</td> <td>Thu Oct 13 2022 14:06:29 GMT+0500 (Uzbekistan Standard Time) (about 5 hours)</td> <td>AMI location</td> <td>amazon/Windows_Server-2016-English-Full-Base-2022.09.14</td> </tr> <tr> <td>Instance auto-recovery</td> <td>Default</td> <td>Lifecycle</td> <td>normal</td> <td>Stop-hibernate behavior</td> <td>disabled</td> </tr> <tr> <td>AMI Launch index</td> <td>0</td> <td>Key pair name</td> <td>MyKeyPairSeminar5</td> <td>State transition reason</td> <td>-</td> </tr> </table> |   | Platform      | Windows  | AMI ID                  | ami-0b3e423be2b1ed909                                   | Monitoring | disabled | Platform details | Windows | AMI name | Windows_Server-2016-English-Full-Base-2022.09.14 | Termination protection | Disabled | Stop protection | Disabled | Launch time | Thu Oct 13 2022 14:06:29 GMT+0500 (Uzbekistan Standard Time) (about 5 hours) | AMI location | amazon/Windows_Server-2016-English-Full-Base-2022.09.14 | Instance auto-recovery | Default | Lifecycle | normal | Stop-hibernate behavior | disabled | AMI Launch index | 0 | Key pair name | MyKeyPairSeminar5 | State transition reason | - |
| Platform   | Windows   | AMI ID        | ami-0b3e423be2b1ed909  | Monitoring              | disabled  |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| Platform details   | Windows   | AMI name      | Windows_Server-2016-English-Full-Base-2022.09.14                             | Termination protection  | Disabled  |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| Stop protection  | Disabled  | Launch time   | Thu Oct 13 2022 14:06:29 GMT+0500 (Uzbekistan Standard Time) (about 5 hours) | AMI location            | amazon/Windows_Server-2016-English-Full-Base-2022.09.14 |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| Instance auto-recovery   | Default   | Lifecycle     | normal   | Stop-hibernate behavior | disabled  |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |
| AMI Launch index   | 0   | Key pair name | MyKeyPairSeminar5  | State transition reason | -   |            |          |                  |         |          |  |                        |          |                 |          |             |  |              |   |                        |         |           |        |                         |          |                  |   |               |                   |                         |   |

If you can see the web page as the following, then Congratulations. You have successfully been able to CodeDeploy your Static web Application.



## Success!



A sample website on AWS has been successfully setup  
Your HTML app is up and running on AWS

### Next up

- Host a static website
- Learn more about Code Deploy
- Go to AWS Portal
- Learn more about all you can do with AWS by visiting the documentation

# Seminar 6

## Running Docker on AWS

In this seminar, you will learn about Docker deployment using Amazon Web Services (AWS) for hosting and how to package your code so that it runs the same way in development and in production. Everything done in this seminar is free tier eligible.

The brief outline of the seminar is provided below:

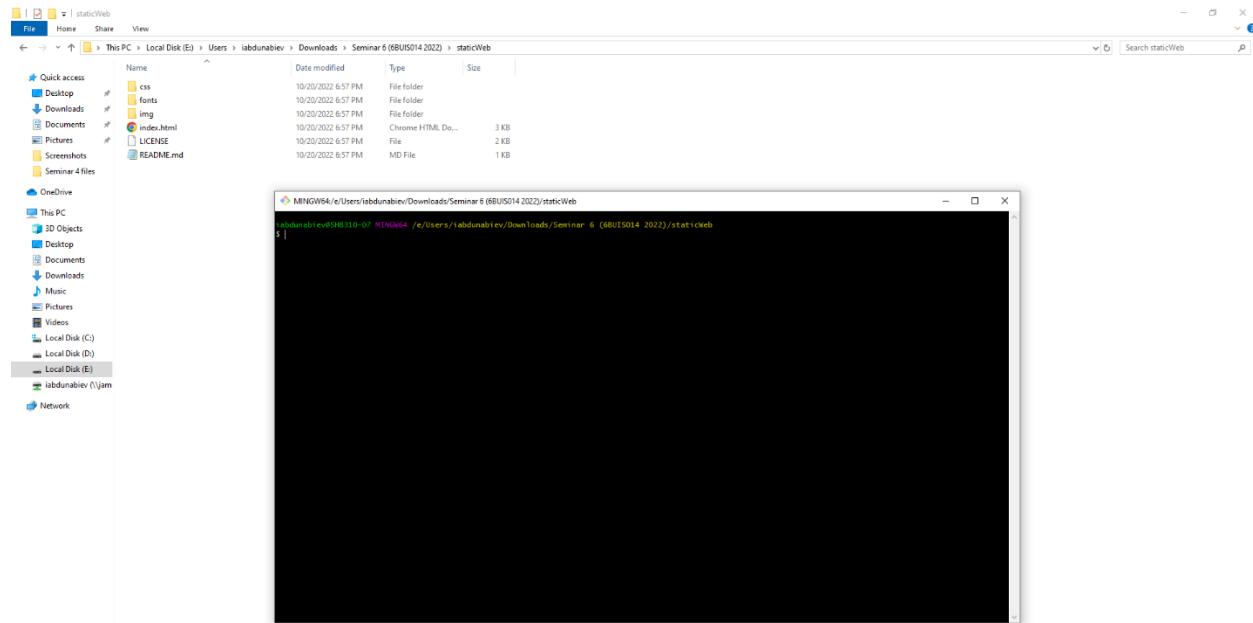
Deploying Docker containers manually

1. Writing DockerFile (Dockerfile)
2. Git Command to push to a repository
3. Creating User Data commands in order to run everything at once
4. Launching an EC2 Instance

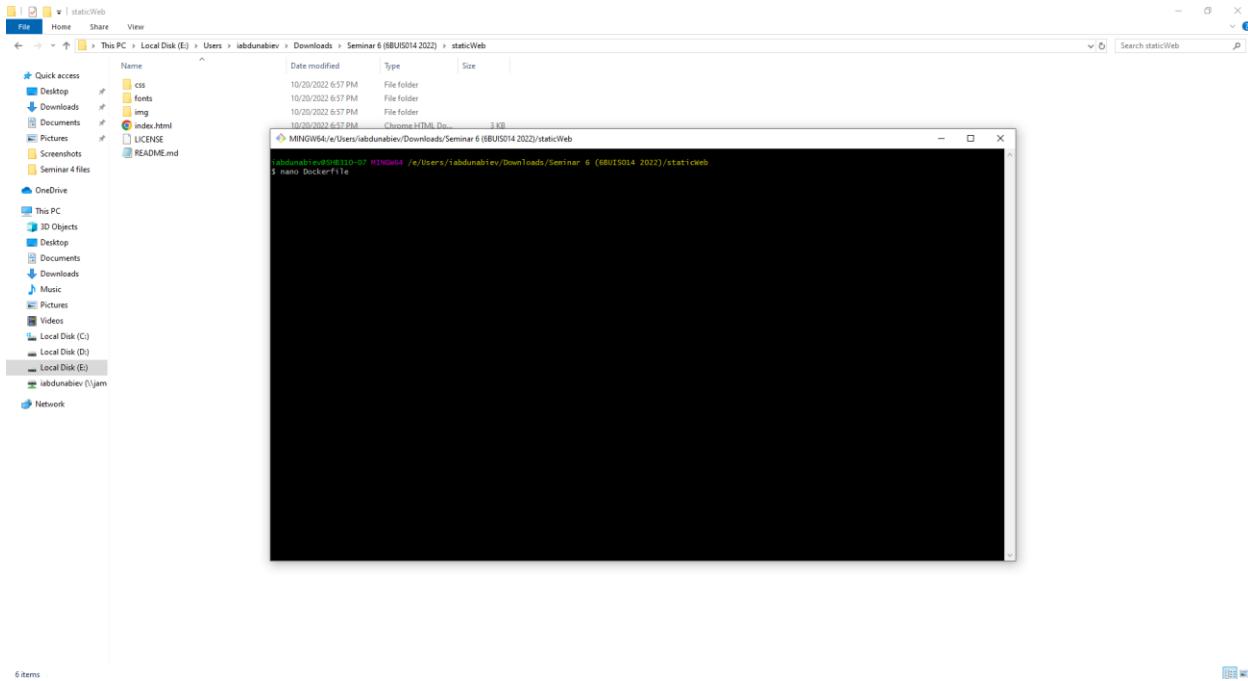
First before writing DockerFile, we need to download the static web-site that is in intranet.

Unzip file to a location that is easy to refer to later.

Open GitBash on that folder and type the Following



nano Dockerfile

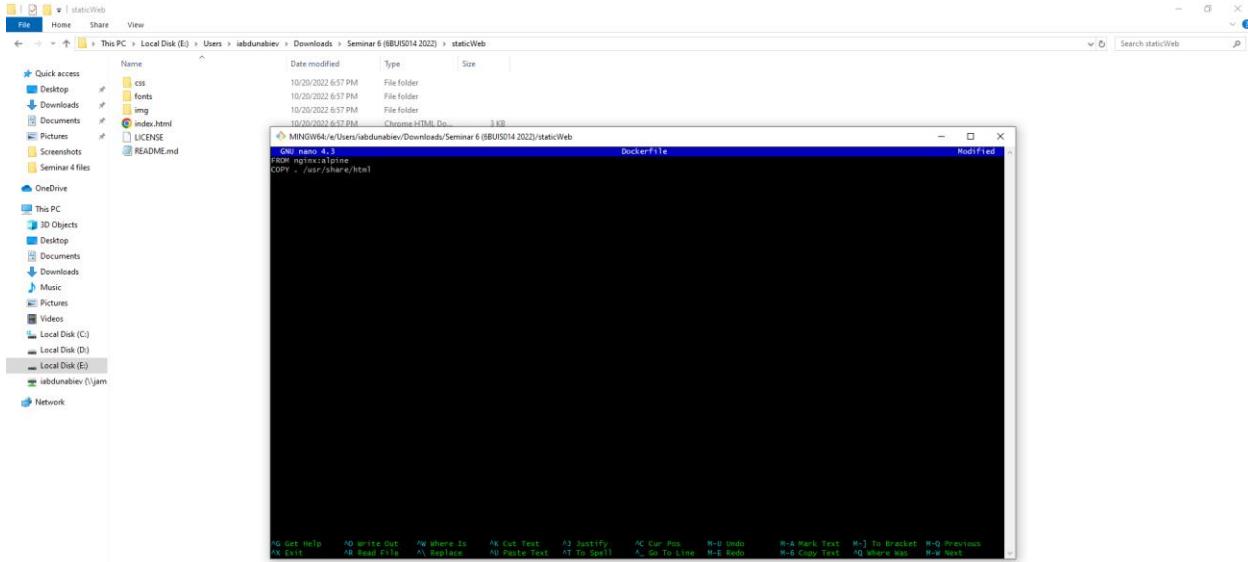


6 items

and into that newly created file type the following.

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

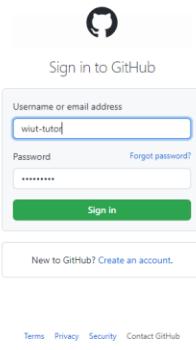
In here when this Dockerfile is executed it will install an nginx into it



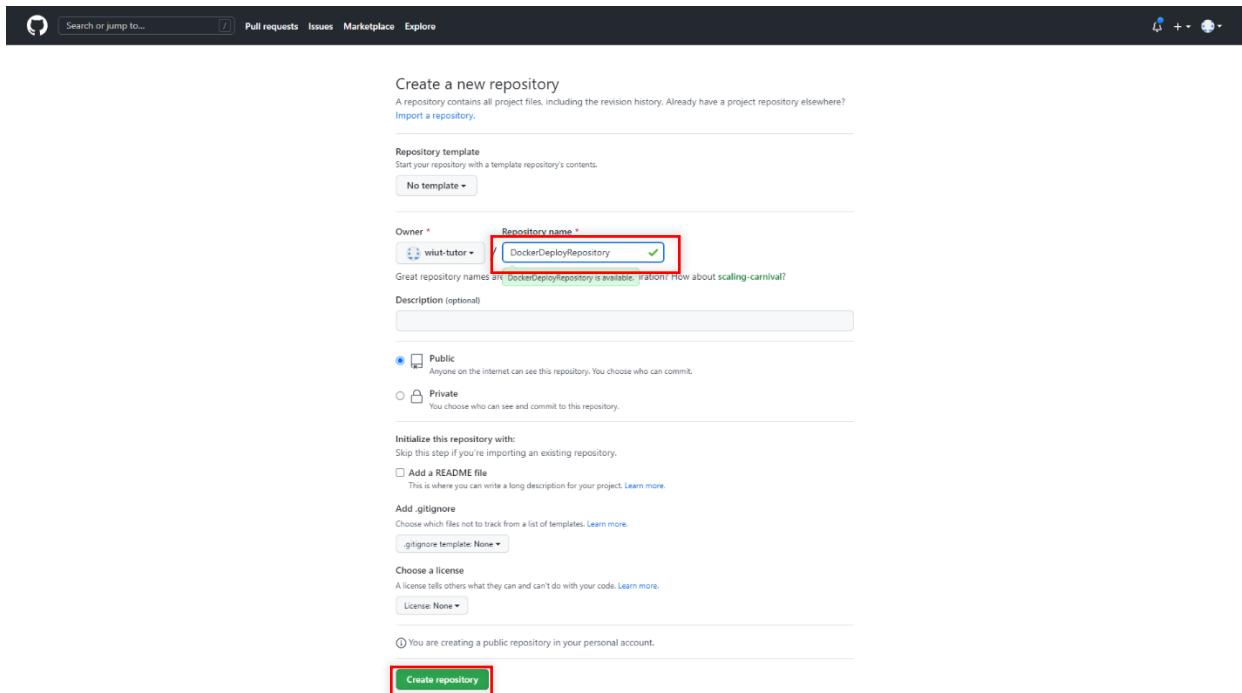
6 items

And quit with saving the file.

Now, go to your github and create a new repository, with any name. For example: DockerDeployRepository.



The screenshot shows the GitHub sign-in page. A green success message at the top right says "Signed in successfully". Below it, the GitHub logo is displayed. The sign-in form has "wiut-tutor" in the "Username or email address" field and a password entered in the "Password" field. A "Sign in" button is visible. At the bottom, there's a link to "Create an account".

The screenshot shows the "Create a new repository" page on GitHub. The "Repository name" field is filled with "DockerDeployRepository", which is highlighted with a red box. The "Owner" dropdown is set to "wiut-tutor". The "Description (optional)" field is empty. Under "Initialize this repository with:", the "Public" radio button is selected. The "Add a README file" checkbox is unchecked. The "Add .gitignore" section shows "gitignore template: None". The "Choose a license" section shows "License: None". A note at the bottom left says "You are creating a public repository in your personal account." A large green "Create repository" button is at the bottom.

Now from gitbash

Push your code to the newly created repository. (You can refer how to push your code to GitHub to Seminar 5 file)

The screenshot shows a GitHub repository page for 'will-tutor / DockerDeployRepository'. The repository is public and has one branch ('main') and no tags. The commit history shows a single commit from 'will-tutor' titled 'int First Commit' made 6 minutes ago. The repository has 0 stars, 1 watch, and 0 forks. It includes files like 'css', 'fonts', 'img', 'Dockerfile', 'LICENSE', 'README.md', and 'index.html'. The 'README.md' file contains a table with language statistics: CSS 51.2%, HTML 48.1%, and Dockerfile 0%. There are sections for 'Sample HTML website', 'License', and 'Contributing'.

What is User data in AWS.

That is the commands list that needs to be executed right after the instance had been launched in AWS. This used data section can be found from the advanced section of launch wizard. At the bottom.

Update commands

For the most cases when you connect to your instance for the first time using ssh the ssh prompt will ask user (you) to update the instance for better performance.

That is done by typing the following

```
sudo yum update -y
```

Installing docker into running instance

What is docker

Docker is an open-source platform that enables developers to build, deploy, run, update and manage containers-standardized, executable components that combine application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. ([source](#))

To install docker in our instance we are going to be need the following command.

```
sudo yum install -y docker
```

and to enable docker as a service we are going to need type the following command

```
sudo service docker start
```

Now our instance is running, we are going to be downloading out static file from github.

In order to do that first we need to have github installed. To install github we need to type following command

```
sudo yum install -y git
```

We will be creating a folder to work with files that are downloaded from GitHub

```
mkdir website {or any name instead of website}
```

```
cd website
```

Pulling everything (our static Web site from GitHub)

```
git init
```

```
git remote add origin {"your_repository_using_https" }.git
```

in my case that is

```
git remote add origin https://github.com/wiut-tutor/DockerDeployRepository.git
```

```
git pull origin main
```

Now we need to change the user mode (permissions) of our docker command in our terminal for better execution of our docker instance without sudo.

```
Sudo usermod -a -G docker ec2-user
```

Building docker

```
docker build -t webserver-image:v1 .
```

Running docker

```
docker run -d -p 80:80 webserver-image:v1
```

All the commands are run one-by-one together during the instance initiation.

The only thing we need to do is to put these commands combined into user date while initiating the instance.

```

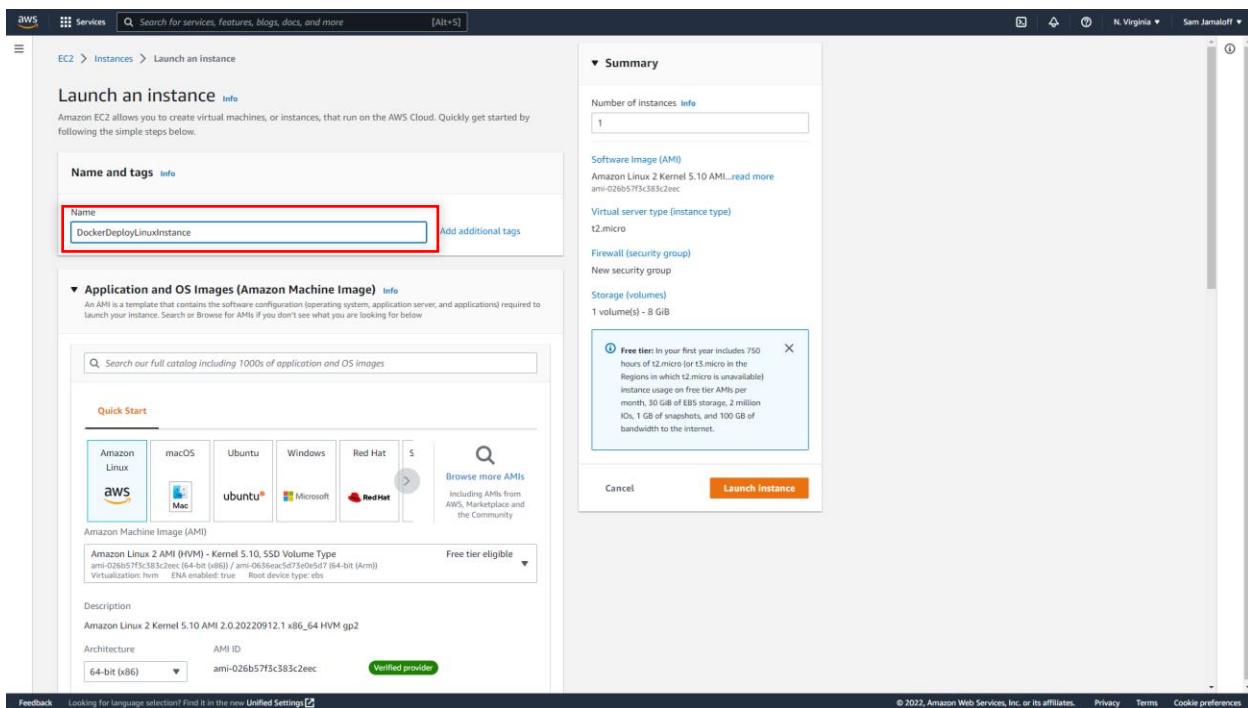
#!/bin/bash
sudo yum update -y
sudo yum install -y docker
sudo service docker start
sudo yum install -y git
mkdir website
cd website
git init
git remote add origin https://github.com/wiut-tutor/DockerDeployRepository.git
git pull origin main

sudo usermod -a -G docker ec2-user

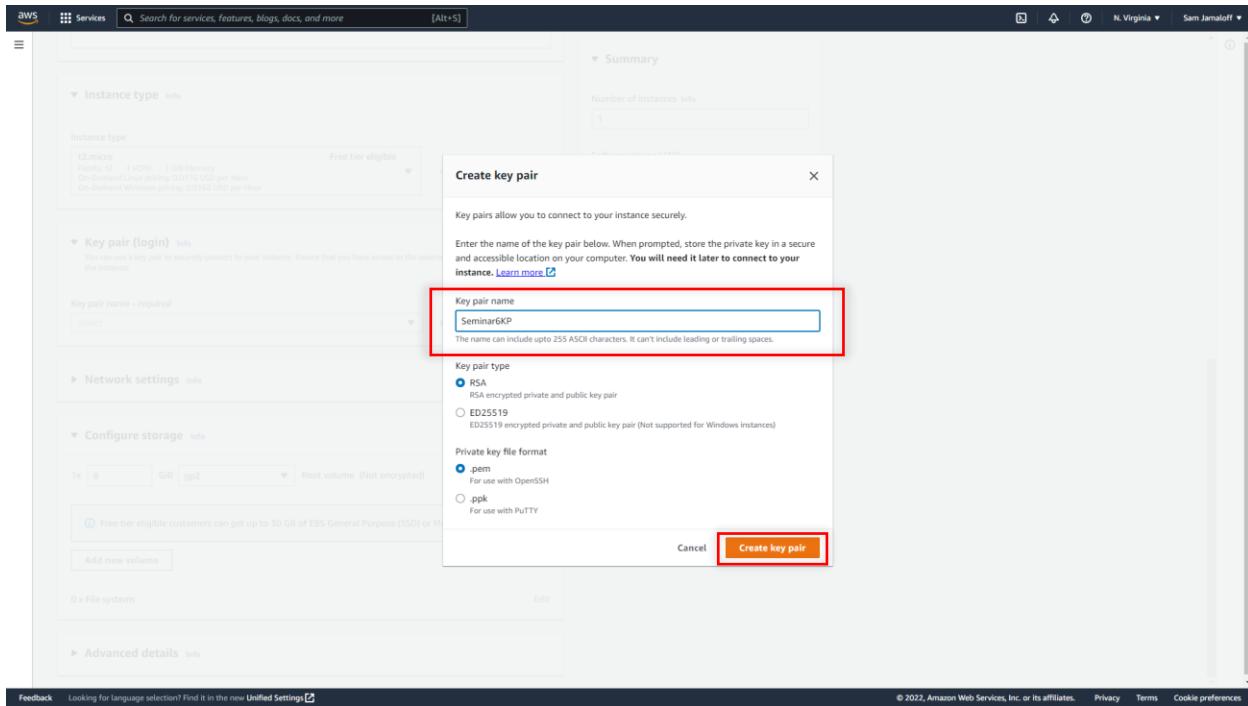
docker build -t webserver-image:v1 .
docker run -d -p 80:80 webserver-image:v1

```

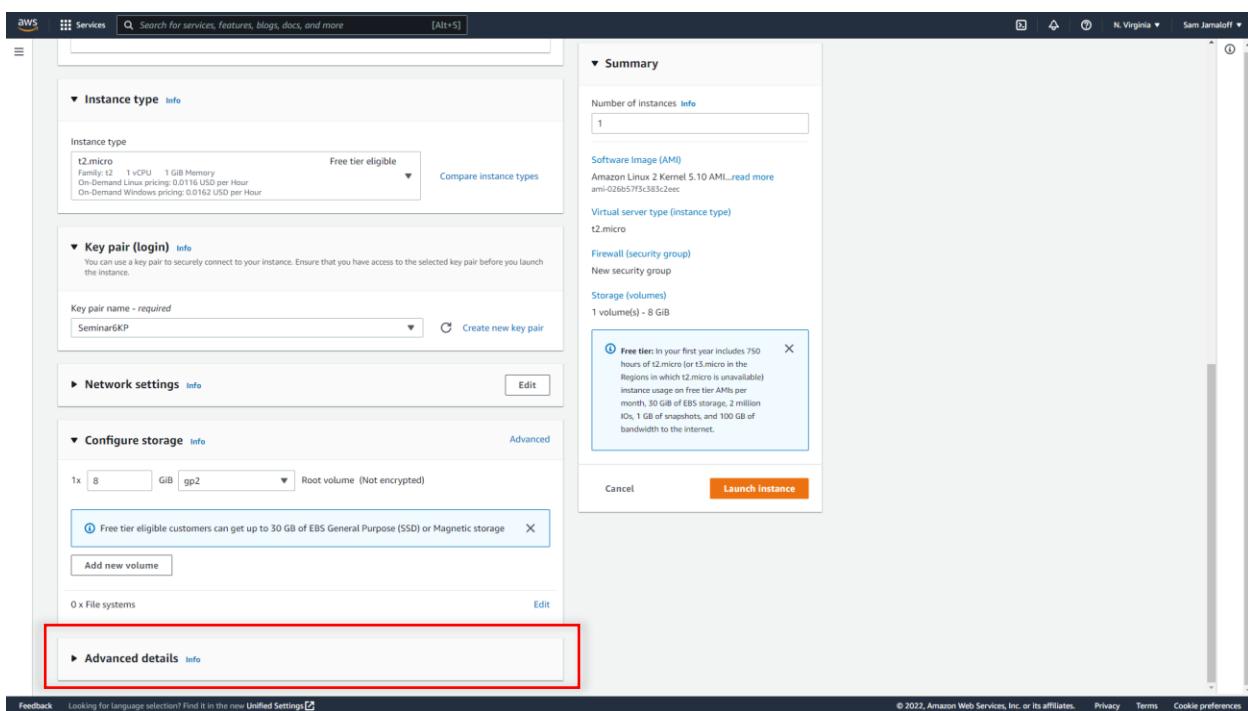
Now your user data is ready go to EC2 launch wizard and launch your instance.

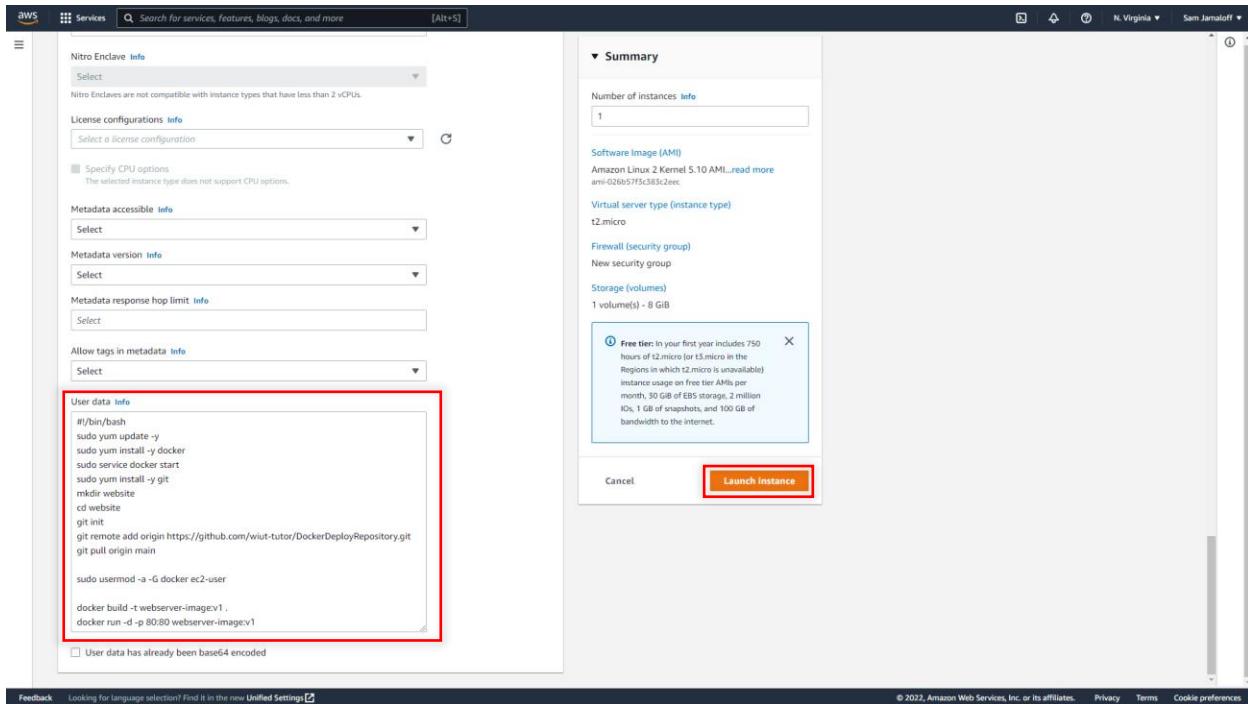


Screenshot of the AWS CloudFormation console showing the creation of a new stack. A modal window titled "Create key pair" is open, prompting for a key pair name. The input field contains "Seminar6KP". The "Create key pair" button is highlighted with a red box.

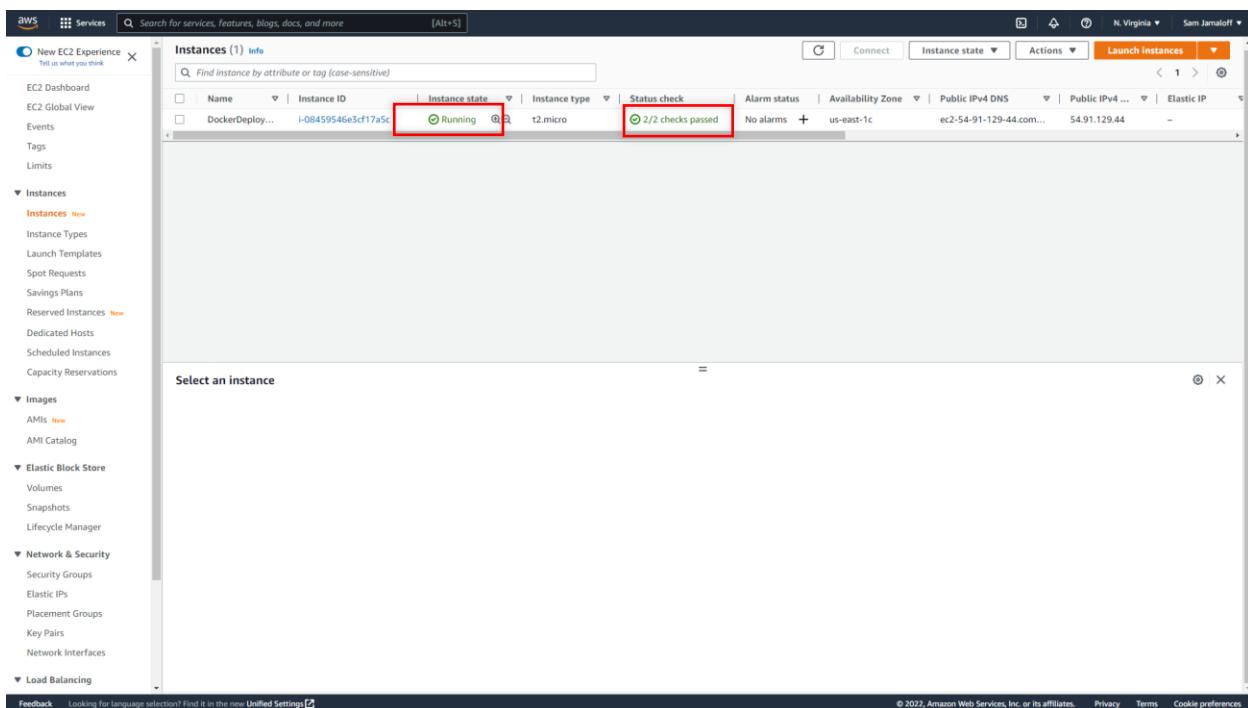


Screenshot of the AWS CloudFormation console showing the creation of a new stack. The "Advanced details" section is highlighted with a red box. A tooltip provides information about free tier usage for t2.micro instances.





We need to wait about 4-5 minutes then we will check for your instance. If Instance is running and status check is okay or 2/2 checked than we can access to it, and you can see you website running.





## Success!



A sample website on AWS has been successfully setup  
Your HTML app is up and running on AWS

### Next up

- Host a static website
- Learn more about Code Deploy
- Go to AWS Portal
- Learn more about all you can do with AWS by visiting the documentation

# Seminar 7

## Working with Cloudera

### Step 1. Installing Cloudera into VirtualBox

1. In your machines there should be VirtualBox installed, if not there is a [link for you to download](#) and install it.
2. Download Cloudera VM image from the link that was provided in WIUT intranet, or from [here](#).

The screenshot shows the WIUT Learning Board interface. On the left, there's a sidebar with various module links like Module Description, Lessons, Rating, etc. The main area displays a list of files under 'Lesson 7 HDFS Commands in VirtualBox'. A red box highlights the 'Cloudera files' section, which contains two files: 'Lecture 7 (6BUIS014C 2022).ppsx' and 'Lecture 7 (6BUIS014C 2022).pdf'.

| File   | Description              |
|--|--------------------------|
| Lecture 7 (6BUIS014C 2022).ppsx  | PowerPoint presentation  |
| Lecture 7 (6BUIS014C 2022).pdf   | PDF document             |
| % Cloudera files   | Cloudera files           |
| A link for Telegram channel to join for preparation for AWS Cloud Practitioner Certification | Link to Telegram channel |

Google Drive

Search in Drive

Shared with me > Cloudera Files

Name ↑

Owner Last modified File size

cloudera-quickstart-vm-5.12.0-0-virtualbox.zip Tutor Sam Oct 21, 2020 5.49 GB

cloudera-quickstart-vm-5.12.0-0-vmware.zip Tutor Sam Oct 21, 2020 5.41 GB

New My Drive Computers Shared with me Recent Stared Trash Storage (91% full) 13.67 GB of 15 GB used Buy storage

cloudera-quickstart-vm-5.12.0-0-virtualbox.zip

Open with ▾

Shared with me > Cloudera Files

Name ↑

Owner Last modified File size

cloudera-quickstart-vm-5.12.0-0-virtualbox.zip Tutor Sam Oct 21, 2020 5.49 GB

cloudera-quickstart-vm-5.12.0-0-vmware.zip Tutor Sam Oct 21, 2020 5.41 GB

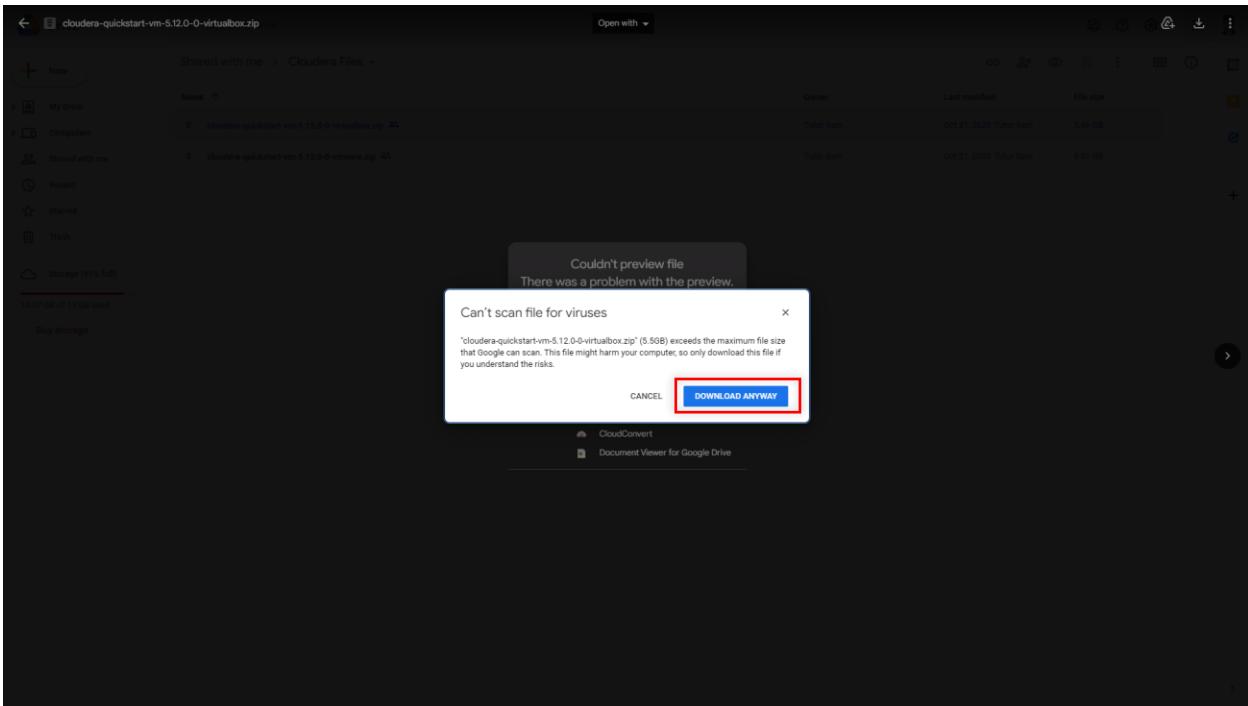
Couldn't preview file  
There was a problem with the preview.

Download Connect more apps...

Try one of the apps below to open or edit this item

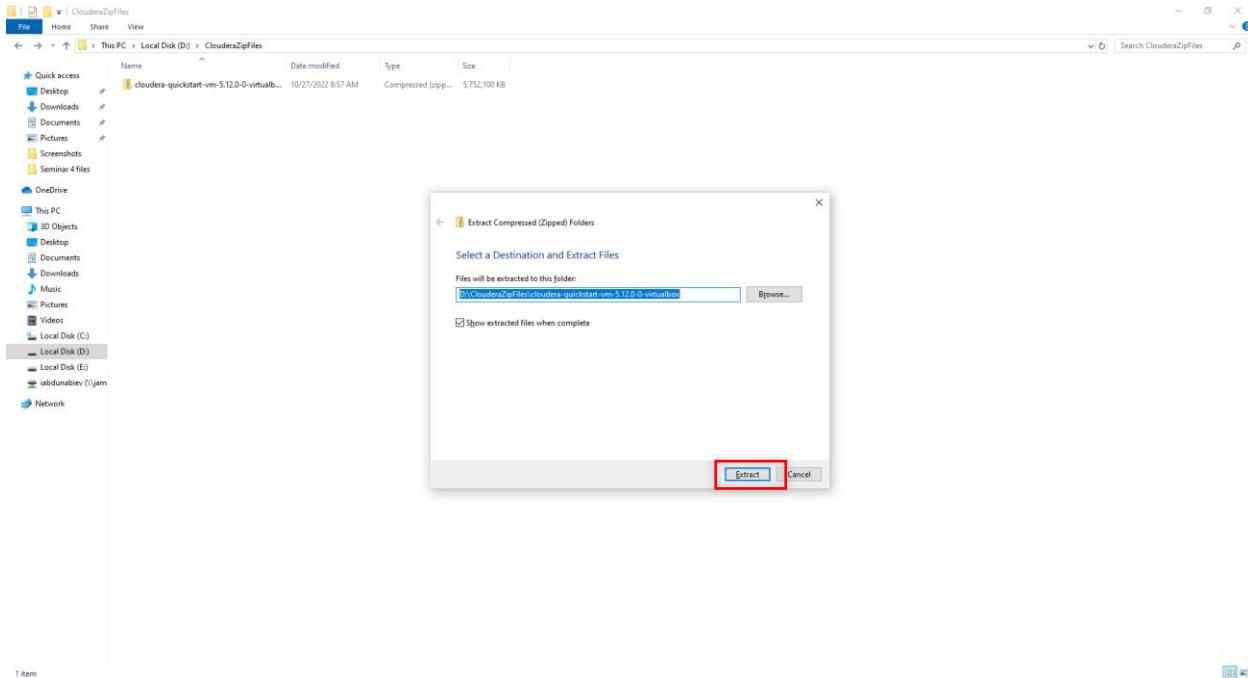
Suggested third-party apps

- ZIP Extractor
- CloudConvert
- Document Viewer for Google Drive

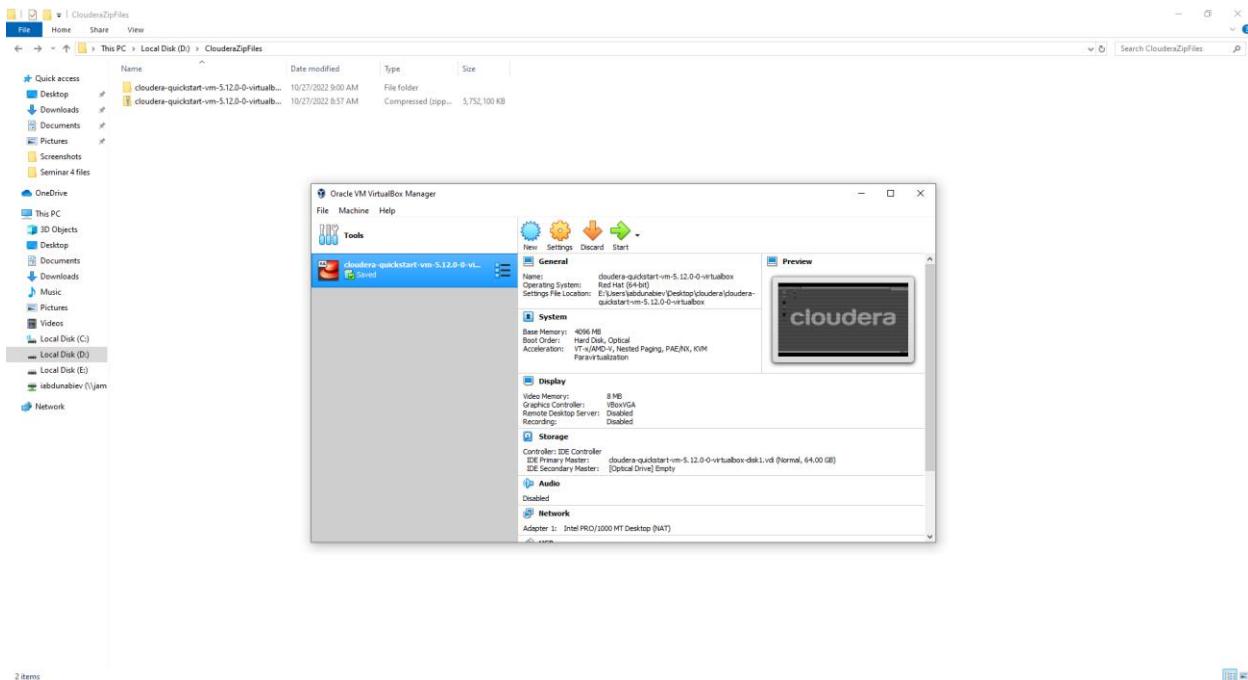


3. Unzip the Cloudera VM image to the folder that is going to be easy for you refer to later.

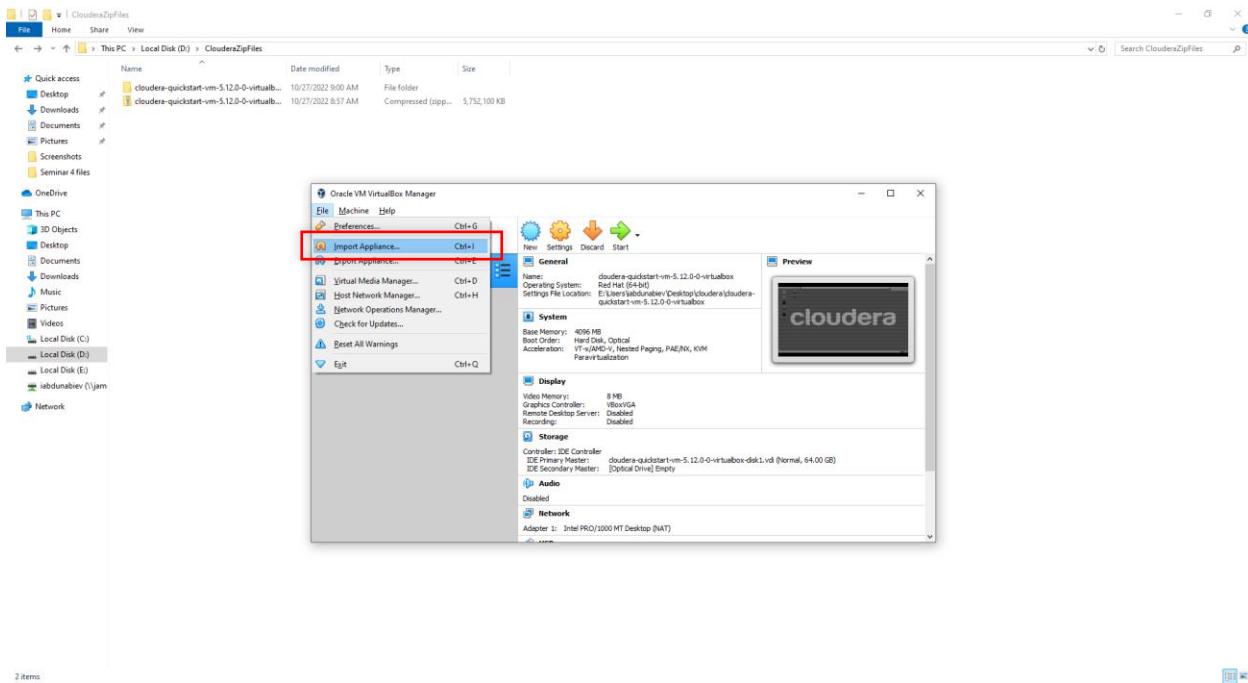




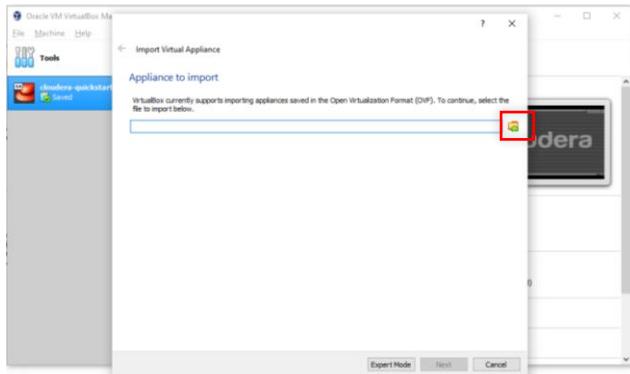
#### 4. Start Virtual box



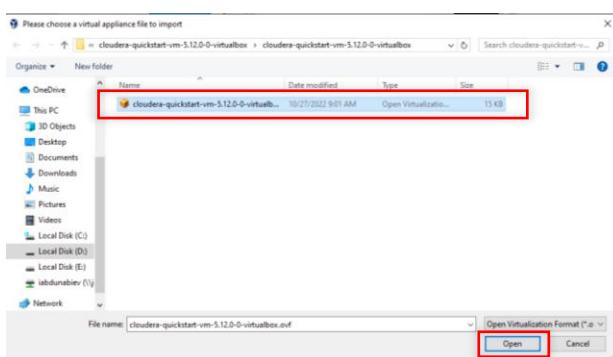
5. Begin the importing the VM by going to file -> Import Appliance



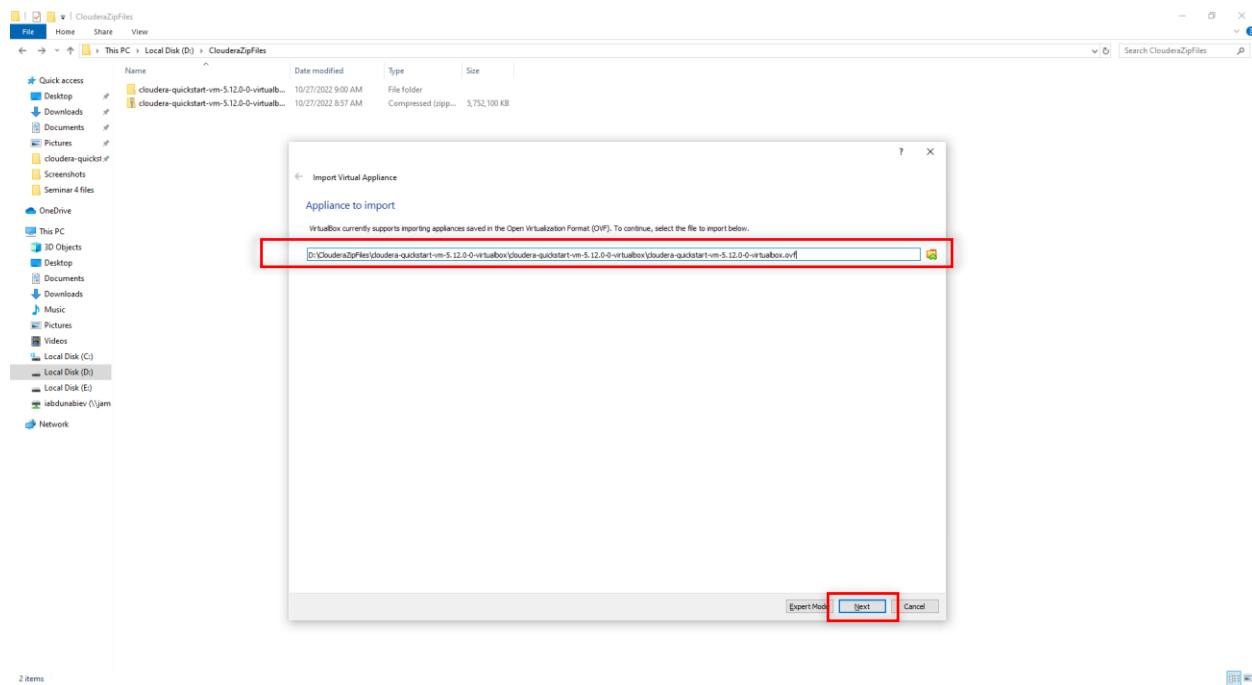
6. Click the folder icon



7. Refer to the folder that you have recently extracted all the file and select the Cloudera file with .ovf extension. In your example that is "cloudera-quickstart-vm-5.12.0-0-virtualbox.zip" and click open to proceed.

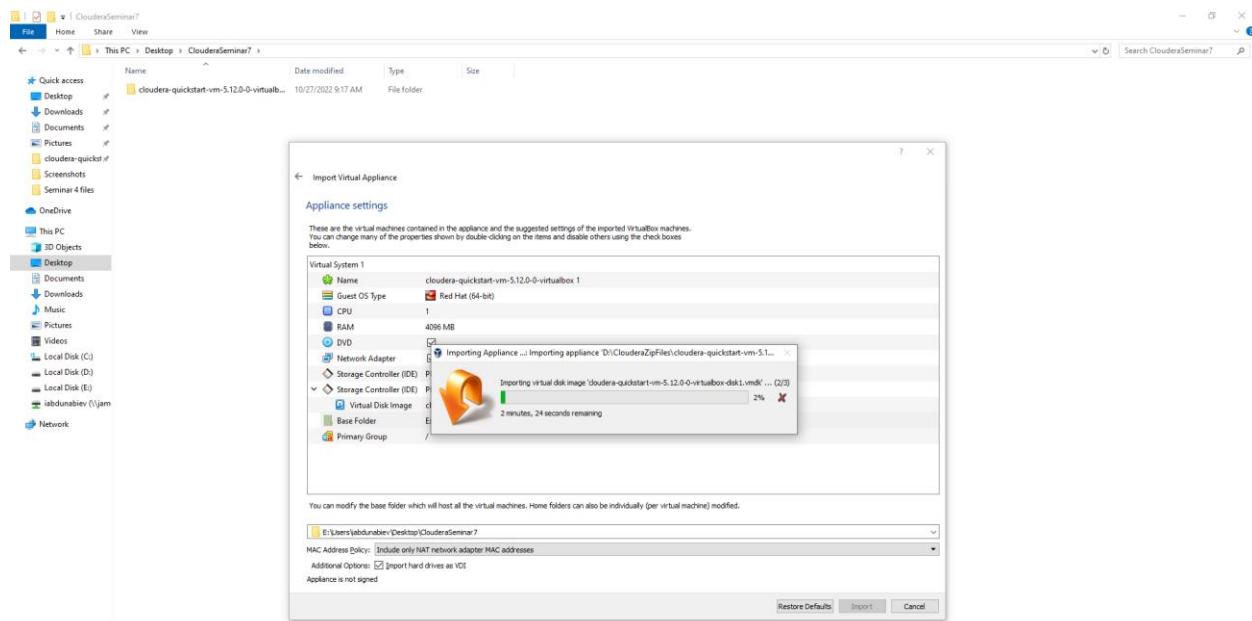


## 8. Click Next to Proceed



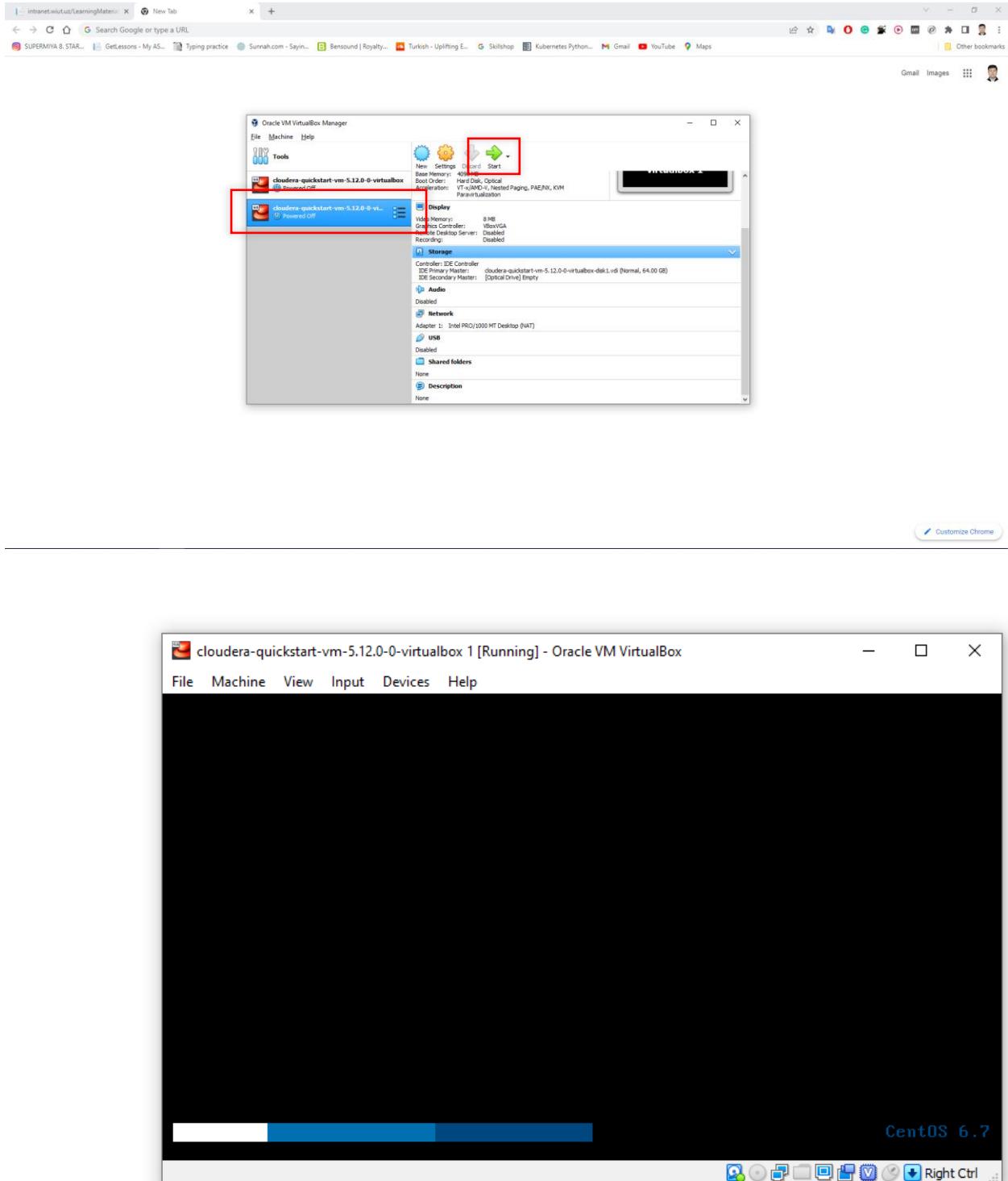
## 9. Click import.

If you are doing this seminar on premises of University's computer than there is going to be an error because you are trying to write to the folder of which you don't have access to. To fix that, create a new folder in a desktop and change the Base folder section of newly opened window prior to the import and change it to the new created folder on your desktop.

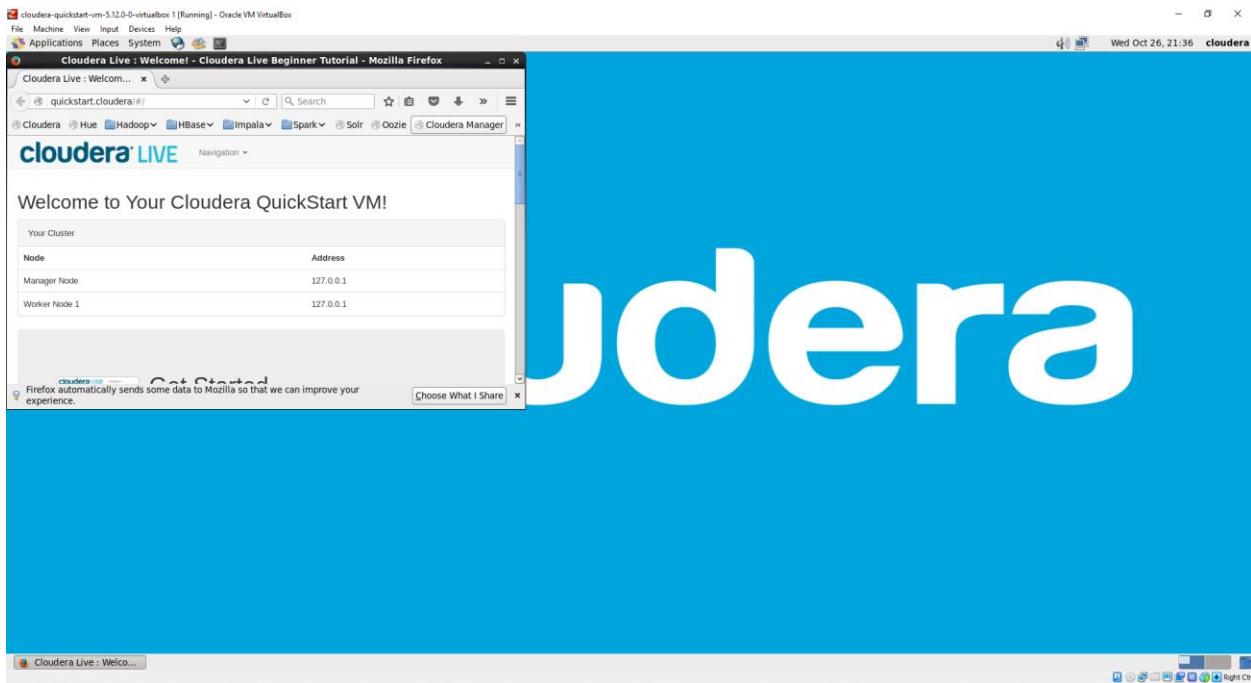


10. Virtual machine image will be imported in 4-5 minutes.

11. Launch Cloudera VM, by choosing the newly installed VM image and Start



It is going to take some time to run.



This is the final window of Cloudera installation.

## Step 2. List of Hadoop Commands

HDFS is a major component of the Hadoop ecosystem which is responsible for storing large data sets of structured or unstructured data across various nodes. The first step towards learning Big Data tools is executing HDFS commands and exploring how HDFS works.

Apache Hadoop has come up with a simple and yet basic Command Line interface, a simple interface to access the underlying distributed file system. In this seminar you will learn the basic HDFS file system commands which are like UNIX file system commands

1. version.

Command Usage: `version`

Example:

```
[cloudera@quickstart ~]$ hadoop version
```

Description: Shows the version of Hadoop installed.

2. `mkdir`

Command Usage: `mkdir <path>`

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /user/cloudera/dir1
```

Description: This command takes the as an argument and creates the directory

3. ls

Command Usage: ls <path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera
```

Description: This command displays the contents of the directory specified by . It shows the name, permissions, owner, size and modification date of each entry.

Second Example:

```
[cloudera@quickstart ~]$ hdfs dfs -ls -R /user
```

Description: This command behaves like ls but displays entries in all the subdirectories recursively

4. put

First, let's download one text file into our system. Execute the following command:

```
[cloudera@quickstart ~]$ wget
```

```
https://filesamples.com/samples/document/txt/sample3.txt
```

Now we have sample3.txt file inside our /home/cloudera directory

Command Usage: put <localsrc><dest>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/sample3.txt
```

```
/user/cloudera/dir1
```

Description: This command copies the file in the local filesystem to the file in DFS.

5. copyFromLocal

Command Usage: copyFromLocal <localsrc> <dest>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -copyFromLocal
```

```
/home/cloudera/sample3.txt /user/cloudera/dir1
```

Description: This command is like put command. But the source should refer to local file.

6. get

Command Usage: get <src> <localhost>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -get /user/cloudera/dir1 /home/cloudera
```

Description: This Hadoop shell command copies the file in HDFS identified by <src> to file in local file system identified by <localhost>

Second Example: First, create a folder named dir2 in the hdfs system, then download file <https://filesamples.com/samples/document/txt/sample2.txt> and put it in folder dir2

```
[cloudera@quickstart ~]$ wget  
https://filesamples.com/samples/document/txt/sample2.txt
```

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir dir2  
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/sample2.txt  
/user/cloudera/dir2
```

```
[cloudera@quickstart ~]$ hdfs dfs -getmerge /user/cloudera/dir1/sample3.txt  
/user/cloudera/dir2/sample2.txt /home/cloudera/sample.txt
```

Description: This HDFS command retrieves all files in the source path entered by the user in HDFS. And merges them into one single file created in the local file system identified by local destination.

Third Example:

```
[cloudera@quickstart ~]$ hdfs dfs -getfattr -d /user/cloudera/dir1
```

Description: This HDFS command displays if there is any extended attribute names and values for the specified file or directory.

Options:

- R: It lists the attributes for all files and directories recursively.
- n name: It shows the named extended attribute value.
- d: It shows all the extended attribute values associated with the pathname.
- e encoding: Encodes values after extracting them. The valid coded forms that are “text”, “hex”, and “base64”. The values which are encoded as text strings gets enclosed with double quotes (“ ”). It uses prefix 0x for hexadecimal conversion. And 0s for all the values which gets coded as base64.

## 7. copyToLocal

Command Usage: copyToLocal <src> <localdest>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -copyToLocal /user/cloudera/dir1 /home
```

Description: It is like get command. Only the difference is that in this the destination of copied file should refer to a local file.

## 8. cat

Command Usage: cat <file-name>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/dir2/sample2.txt
```

Description: This Hadoop shell command displays the contents of file on console or stdout.

## 9. mv

Command usage: mv <src> <dest>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -mv /user/cloudera/dir1/sample3.txt  
/user/cloudera/dir2
```

Description: This Hadoop shell command moves the file from the specified source to destination within HDFS.

10. cp

Command Usage: cp <src> <dest>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -cp /user/cloudera/dir2/sample2.txt  
/user/cloudera/dir1
```

Description: This Hadoop shell command copies the file or directory from given source to destination within HDFS.

## Congratulations!

You have successfully completed your very first introduction to Big Data seminar.

Below you may find a cheat sheet with most used HDFS commands.

## Hadoop HDFS Command Cheatsheet

| List Files                |  |
|---------------------------|--|
| hdfs dfs -ls /            | List all the files/directories for the given hdfs destination path.  |
| hdfs dfs -ls -d /hadoop   | Directories are listed as plain files. In this case, this command will list the details of hadoop folder.                          |
| hdfs dfs -ls -h /data     | Format file sizes in a human-readable fashion (eg 64.0m instead of 67108864).  |
| hdfs dfs -ls -R /hadoop   | Recursively list all files in hadoop directory and all subdirectories in hadoop directory.   |
| hdfs dfs -ls /hadoop/dat* | List all the files matching the pattern. In this case, it will list all the files inside hadoop directory which starts with 'dat'. |

| Read/Write Files  |   |
|---|---|
| hdfs dfs -text /hadoop/derby.log                        | HDFS Command that takes a source file and outputs the file in text format on the terminal. The allowed formats are zip and TextRecordInputStream. |
| hdfs dfs -cat /hadoop/test                              | This command will display the content of the HDFS file test on your stdout .  |
| hdfs dfs -appendToFile /home/ubuntu/test1 /hadoop/text2 | Appends the content of a local file test1 to a hdfs file test2.   |

| Upload/Download Files                               |   |
|---|---|
| hdfs dfs -put /home/ubuntu/sample /hadoop           | Copies the file from local file system to HDFS.   |
| hdfs dfs -put -f /home/ubuntu/sample /hadoop        | Copies the file from local file system to HDFS, and in case the local already exists in the given destination path, using -f option with put command will overwrite it. |
| hdfs dfs -put -l /home/ubuntu/sample /hadoop        | Copies the file from local file system to HDFS. Allow DataNode to lazily persist the file to disk. Forces replication factor of 1.                                      |
| hdfs dfs -put -p /home/ubuntu/sample /hadoop        | Copies the file from local file system to HDFS. Passing -p preserves access and modification times, ownership and the mode.   |
| hdfs dfs -get /newfile /home/ubuntu/                | Copies the file from HDFS to local file system.   |
| hdfs dfs -get -p /newfile /home/ubuntu/             | Copies the file from HDFS to local file system. Passing -p preserves access and modification times, ownership and the mode.   |
| hdfs dfs -get /hadoop/*.txt /home/ubuntu/           | Copies all the files matching the pattern from local file system to HDFS.   |
| hdfs dfs -copyFromLocal /home/ubuntu/sample /hadoop | Works similarly to the put command, except that the source is restricted to a local file reference.   |
| hdfs dfs -copyToLocal /newfile /home/ubuntu/        | Works similarly to the put command, except that the destination is restricted to a local file reference.  |
| hdfs dfs -moveFromLocal /home/ubuntu/sample /hadoop | Works similarly to the put command, except that the source is deleted after it's copied.  |

| File Management                        |   |
|--|---|
| hdfs dfs -cp /hadoop/file1 /hadoop1    | Copies file from source to destination on HDFS. In this case, copying file1 from hadoop directory to hadoop1 directory.                         |
| hdfs dfs -cp -p /hadoop/file1 /hadoop1 | Copies file from source to destination on HDFS. Passing -p preserves access and modification times, ownership and the mode.                     |
| hdfs dfs -cp -f /hadoop/file1 /hadoop1 | Copies file from source to destination on HDFS. Passing -f overwrites the destination if it already exists.                                     |
| hdfs dfs -mv /hadoop/file1 /hadoop1    | Move files that match the specified file pattern <src> to a destination <dst>. When moving multiple files, the destination must be a directory. |
| hdfs dfs -rm /hadoop/file1             | Deletes the file (sends it to the trash).   |

|  |   |
|--|---|
| <code>hdfs dfs -rm -r /hadoop</code>         | Deletes the directory and any content under it recursively.   |
| <code>hdfs dfs -rm -skipTrash /hadoop</code> | The <code>-skipTrash</code> option will bypass trash, if enabled, and delete the specified file(s) immediately.                   |
| <code>hdfs dfs -rm -f /hadoop</code>         | If the file does not exist, do not display a diagnostic message or modify the exit status to reflect an error.                    |
| <code>hdfs dfs -rmdir /hadoop1</code>        | Delete a directory.   |
| <code>hdfs dfs -mkdir /hadoop2</code>        | Create a directory in specified HDFS location.  |
| <code>hdfs dfs -mkdir -f /hadoop2</code>     | Create a directory in specified HDFS location. This command does not fail even if the directory already exists.                   |
| <code>hdfs dfs -touchz /hadoop3</code>       | Creates a file of zero length at <code>&lt;path&gt;</code> with current time as the timestamp of that <code>&lt;path&gt;</code> . |

| Ownership and Validation                              |   |
|---|---|
| <code>hdfs dfs -checksum /hadoop/file1</code>         | Dump checksum information for files that match the file pattern <code>&lt;src&gt;</code> to stdout.   |
| <code>hdfs dfs -chmod 755 /hadoop/file1</code>        | Changes permissions of the file.  |
| <code>hdfs dfs -chmod -R 755 /hadoop</code>           | Changes permissions of the files recursively.   |
| <code>hdfs dfs -chown ubuntu:ubuntu /hadoop</code>    | Changes owner of the file. 1st ubuntu in the command is owner and 2nd one is group.   |
| <code>hdfs dfs -chown -R ubuntu:ubuntu /hadoop</code> | Changes owner of the files recursively.   |
| <code>hdfs dfs -chgrp ubuntu /hadoop</code>           | Changes group association of the file.  |
| <code>hdfs dfs -chgrp -R ubuntu /hadoop</code>        | Changes group association of the files recursively.   |
| Filesystem  |   |
| <code>hdfs dfs -df /hadoop</code>                     | Shows the capacity, free and used space of the filesystem.  |
| <code>hdfs dfs -df -h /hadoop</code>                  | Shows the capacity, free and used space of the filesystem. <code>-h</code> parameter Formats the sizes of files in a human-readable fashion.                              |
| <code>hdfs dfs -du /hadoop/file</code>                | Show the amount of space, in bytes, used by the files that match the specified file pattern.  |
| <code>hdfs dfs -du -s /hadoop/file</code>             | Rather than showing the size of each individual file that matches the pattern, shows the total (summary) size.  |
| <code>hdfs dfs -du -h /hadoop/file</code>             | Show the amount of space, in bytes, used by the files that match the specified file pattern. Formats the sizes of files in a human-readable fashion.                      |
| Administration  |   |
| <code>hdfs balancer -threshold 30</code>              | Runs a cluster balancing utility. Percentage of disk capacity. This overwrites the default threshold.   |
| <code>hadoop version</code>                           | To check the version of Hadoop.   |
| <code>hdfs fsck /</code>                              | It checks the health of the Hadoop file system.   |
| <code>hdfs dfsadmin -safemode leave</code>            | The command to turn off the safemode of NameNode.   |
| <code>hdfs dfsadmin -refreshNodes</code>              | Re-read the hosts and exclude files to update the set of Datanodes that are allowed to connect to the Namenode and those that should be decommissioned or recommissioned. |
| <code>hdfs namenode -format</code>                    | Formats the NameNode.   |

# Seminar 8.

## HBase Shell Commands

HBase is a data model similar to Google's Big Table designed to provide quick random access to huge amounts of structured data. This seminar provides an introduction to HBase and commands to interact with HBase shell.

### 1. Starting shell.

HBase contains an interactive shell using which you can communicate with HBase. Open a terminal

At the prompt –type in:

```
[cloudera@quickstart ~]$ hbase shell
```

Other shell commands:

- **status** – provides the status of HBase, for example, the number of servers.
- **version** – provides the version of HBase being used.
- **table\_help** – provides help for table-reference commands.
- **whoami** – provides information about the user.

### 2. Creating a table using HBase shell

You can create a table using the create command, here you must specify the table name and the Column Family name. The syntax to create a table in HBase shell is shown below.

```
create '<table name>','<column family>'
```

Example:

```
hbase(main):002:0> create 'emp', 'personal data', 'professional data'
```

Verification:

```
hbase(main):002:0> list
TABLE
emp
2 row(s) in 0.0340 seconds
```

### 3. Inserting data using HBase shell

To create data in an HBase table, **put** command is used

```
put '<table name>','row1','<colfamily:colname>','<value>'
```

Example:

```
hbase(main):005:0> put 'emp','1','personal data:name','raju'
0 row(s) in 0.6600 seconds
hbase(main):006:0> put 'emp','1','personal data:city','hyderabad'
0 row(s) in 0.0410 seconds
hbase(main):007:0> put 'emp','1','professional
data:designation','manager'
0 row(s) in 0.0240 seconds
hbase(main):007:0> put 'emp','1','professional data:salary','50000'
0 row(s) in 0.0240 seconds
```

### 4. Updating data using HBase shell

You can update an existing cell value using the **put** command. To do so, just follow the same syntax and mention your new value as shown below.

```
put 'table name','row ','Column family:column name','new value'
```

The newly given value replaces the existing value, updating the row.

Example:

Suppose there is a table in HBase called emp with the following data.

```
hbase(main):003:0> scan 'emp'
ROW COLUMN + CELL
row1 column = personal:name, timestamp = 1418051555, value = raju
row1 column = personal:city, timestamp = 1418275907, value = Hyderabad
row1 column = professional:designation, timestamp = 14180555,value = manager
row1 column = professional:salary, timestamp = 1418035791555,value = 50000
1 row(s) in 0.0100 seconds
```

The following command will update the city value of the employee named 'Raju' to Delhi..

```
hbase(main):002:0> put 'emp','row1','personal:city','Delhi'  
0 row(s) in 0.0400 seconds
```

## 5. Reading data using HBase shell

The get command is used to read data from a table in HBase. Using get command, you can get a single row of data at a time.

```
get '<table name>','row1'
```

Example:

```
hbase(main):012:0> get 'emp', '1'  
  
COLUMN          CELL  
  
personal : city timestamp = 1417521848375, value = hyderabad  
personal : name timestamp = 1417521785385, value = ramu  
professional: designation timestamp = 1417521885277, value = manager  
professional: salary timestamp = 1417521903862, value = 50000  
  
4 row(s) in 0.0270 seconds
```

## 6. Reading a Specific Column

Given below is the syntax to read a specific column using the **get** method.

```
hbase> get 'table name', 'rowid', {COLUMN => 'column family:column name'}
```

Example: Given below is the example to read a specific column in HBase table

```
hbase(main):015:0> get 'emp', 'row1', {COLUMN => 'personal:name'}  
COLUMN          CELL  
personal:name timestamp = 1418035791555, value = raju  
1 row(s) in 0.0080 seconds
```

## 7. Deleting a specific cell in a table

Using the delete command, you can delete a specific cell in a table. The syntax of delete command is as follows:

```
delete '<table name>', '<row>', '<column name >', '<time stamp>'
```

Example:

```
hbase(main):006:0> delete 'emp', '1', 'personal data:city',
1417521848375
0 row(s) in 0.0060 seconds
```

## 8. Deleting all cells in a row

Using the “deleteall” command, you can delete all the cells in a row. Given below is the syntax of deleteall command

```
deleteall '<table name>', '<row>'
```

Example:

```
hbase(main):007:0> deleteall 'emp','1'
0 row(s) in 0.0240 seconds
```

## 9. Scanning using HBase shell

The scan command is used to view the data in HTable. Using the scan command, you can get the table data. Its syntax is as follows:

```
scan '<table name>'
```

Example: The following example shows how to read data from a table using the scan command. Here we are reading the emp table.

```
hbase(main):010:0> scan 'emp'

ROW                               COLUMN + CELL

1 column = personal data:city, timestamp = 1417521848375, value = hyderabad
1 column = personal data:name, timestamp = 1417521785385, value = ramu
```

```
1 column = professional data:designation, timestamp = 1417585277, value =
manager

1 column = professional data:salary, timestamp = 1417521903862, value = 50000

1 row(s) in 0.0370 seconds
```

## 10. Dropping a table using HBase shell

Using the drop command, you can delete a table. Before dropping a table, you have to disable it.

```
hbase(main):018:0> disable 'emp'
0 row(s) in 1.4580 seconds

hbase(main):019:0> drop 'emp'
0 row(s) in 0.3060 seconds
/usr/cloudera/dir1
```

## Congratulations!

You have successfully completed your seminar on HBase shell commands.

# Seminar 9.

# Introduction to MapReduce

MapReduce is a programming paradigm that runs in the background of Hadoop to provide scalability and easy data-processing solutions. This seminar is an introduction to Map/Reduce that implements Wordcount with streaming using Python programming language.

## 1. Review the mapper and reducer code.

Review the following python code. Read the comments to understand it.

## File 1: wordcount\_mapper.py

```

keys = line.split() # split line at blanks (by default),
# and return a list of keys
for key in keys:    # a for loop through the list of keys
    value = 1
    print('{0}\t{1}'.format(key, value) )
    # the {} is replaced by 0th,1st items in format list
    # also, note that 'tab' is the Hadoop's default character
    # that separates the key from the value

```

## File 2: wordcount\_reducer.py

```

#!/usr/bin/env python

# -----
#This reducer code will input a line of text and
#      output <word, total-count>
# -----
import sys

last_key = None # initialize these variables
running_total = 0

# -----
# Loop through file
# -----
for input_line in sys.stdin:
    input_line = input_line.strip()
    # -----
    # Get Next Word
    # -----
    this_key, value = input_line.split("\t", 1)
    # the Hadoop default is tab separates key value
    # the split command returns a list of strings,
    # in this case the line is split into 2 variables
    value = int(value) # int() will convert a string to integer
    # (this program does no error checking)

    # -----
    # Key Check part
    #      if this current key is same
    #          as the last one Consolidate
    #      otherwise  Emit

```

```

# -----
if last_key == this_key:      #check if key has changed ('==' is
                            #      logical equality check
    running_total += value   # add value to running total
else:
    if last_key:            # if this key that was just read in
                            # is different, and the previous
                            # (i.e. last) key is not empty,
                            # then output
                            # the previous <key running-count>
        print( "{0}\t{1}".format(last_key, running_total) )
                            # hadoop expects tab (i.e. '\t')
                            # separation
    running_total = value   # reset values
    last_key = this_key

# write down the last line at the end of the file
if last_key == this_key:
    print( "{0}\t{1}".format(last_key, running_total))

```

Other shell commands:

## 2. Create and prepare Python files

Open gedit text editor or type in the following command to open a text editor,

```

> gedit wordcount_mapper.py
> gedit wordcount_reducer.py

```

and then cut and paste the above lines for wordcount\_mapper.py into the text editor, save, and exit. Repeat for wordcount\_reducer.py.

Execute the following to make it executable:

```

> chmod +x wordcount_mapper.py
> chmod +x wordcount_reducer.py

```

Verification. To see that the indentations line up as above execute:

```

> more wordcount_mapper.py
> more wordcount_reducer.py

```

### 3. Create some text data

During the last seminar we put 2 files in `/user/cloudera/dir2` directory in HDFS. You can work with these files or you can download 2 different text files from the Internet. For example, you can download “Peter Pan” by J. M. Barrie (<https://www.gutenberg.org/files/16/16-0.txt>) and “Adventures of Huckleberry Finn” by Mark Twain (<https://www.gutenberg.org/files/76/76-0.txt>). The commands to download files were given in the previous seminar.

Then create a directory on the HDFS file system (if already exists that's OK):

```
hdfs dfs -mkdir /user/cloudera/input
```

and copy the files from local filesystem to the HDFS filesystem

```
hdfs dfs -put /home/cloudera/file1.txt /user/cloudera/input  
hdfs dfs -put /home/cloudera/file2.txt /user/cloudera/input
```

Make sure your files are there:

```
hdfs dfs -ls /user/cloudera/input
```

### 4. Execute MapReduce command

Run the Hadoop WordCount example with the input and output specified.

Note that your file paths may differ. The ‘\’ just means the command continues on next line.

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \  
-input /user/cloudera/input \  
-output /user/cloudera/output_new \  
-mapper /home/cloudera/wordcount_mapper.py \  
-reducer /home/cloudera/wordcount_reducer.py
```

Hadoop prints out a whole lot of logging or error information.

### 5. Inspect the results

View the output directory. How many files are there?

```
hdfs dfs -ls /user/cloudera/output_new
```

Look at the files there and check out the contents to see the result, e.g.:

```
hdfs dfs -cat /user/cloudera/output_new/part-00000
```

## 6. Inspect the reducer

Let's change the number of reduce tasks to see its effects. Setting it to 0 will execute no reducer and only produce the map output. (Note the output directory is changed in the snippet below because Hadoop doesn't like to overwrite output).

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \
    -input /user/cloudera/input \
    -output /user/cloudera/output_new_0 \
    -mapper /home/cloudera/wordcount_mapper.py \
    -reducer /home/cloudera/wordcount_reducer.py \
    -numReduceTasks 0
```

How many files were written to the directory? Get the output file from this run, and then upload it

```
> hdfs dfs -getmerge /user/cloudera/output_new_0/* wrdcnt_0_output.txt
```

Try to notice the differences between the output when the reducers are run in Step 4, versus the output when there are no reducers and only the mapper is run in this step. The point of the task is to be aware of what the intermediate results look like. A successful submission will have words and counts that are not accumulated (which the reducer performs).

# Congratulations!

You have successfully completed your seminar on MapReduce.

# Seminar 10.

## Hive and HQL

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop and makes querying and analyzing easy. In this seminar, you will process data by creating Hive tables, populating them with data, and querying them.

### 1. Preparing for Data Processing.

Before you can use Hive to analyze data, you must upload the source data to HDFS. We will analyze data about domestic flights in the US. Files airline\_delay\_causes.csv, airports.csv, and carriers.csv are available on Intranet. Download these files to your virtual machine and then upload the files to HDFS

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /user/cloudera/data  
[cloudera@quickstart ~]$ hdfs dfs -put airline_delay_causes.csv  
/user/cloudera/data
```

We can use these data to answer, for example, the questions like:

- a. How many flights were there from JFK to LAX?
- b. What was the average arrival delay for all delayed flights?
- c. What was the average departure delay for each state?

etc. During the query steps for this seminar, please write HQL queries to answer these questions.

### 2. Creating Hive Tables

Now that you have uploaded the source data, you can create Hive tables. Verify that the files have been uploaded to the shared cluster storage. In the

console, enter the following command to view the contents of the /user/cloudera/data folder in the HDFS file system.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/data
```

We will use the Hive command line interface to work with Hive. Enter the following command to start the Hive command line interface.

```
[cloudera@quickstart ~]$ hive
```

In the Hive command line interface, enter the following HiveQL statement to create a table named **flights**

```
CREATE TABLE flights
(year INT,
month INT,
carrier STRING,
carrier_name STRING,
airport STRING,
airport_name STRING,
arr_flights DOUBLE,
arr_del15 DOUBLE,
carrier_ct DOUBLE,
weather_ct DOUBLE,
nas_ct DOUBLE,
security_ct DOUBLE,
late_aircraft_ct DOUBLE,
rr_cancelled DOUBLE,
arr_diverted DOUBLE,
arr_delay DOUBLE,
carrier_delay DOUBLE,
weather_delay DOUBLE,
nas_delay DOUBLE,
security_delay DOUBLE,
late_aircraft_delay DOUBLE)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
TBLPROPERTIES ('skip.header.line.count'='1');
```

Note that the names of the columns were taken from the file. Also note that TBLPROPERTIES is used to skip the first row when loading data from files.

Verify that the table has been created:

```
> SHOW TABLES;
```

Enter the following command to query the table, and verify that no rows are returned:

```
> SELECT * FROM flights;
```

Using the same principle, create tables for airports and carriers.

## 3. Load the Source Data into the Table

In the Hive command line interface, enter the following HiveQL statement to move the flight file you previously uploaded into the folder for the **flights** table.

```
> LOAD DATA INPATH '/user/cloudera/data/ airline_delay_causes.csv' INTO TABLE flights;
```

Enter the following command to return the first 20 rows in the flights table:

```
SELECT * FROM flights LIMIT 20;
```

Examine the output. Check if you have NULL values inside.

Using the same principle, load data into airports and carriers table.

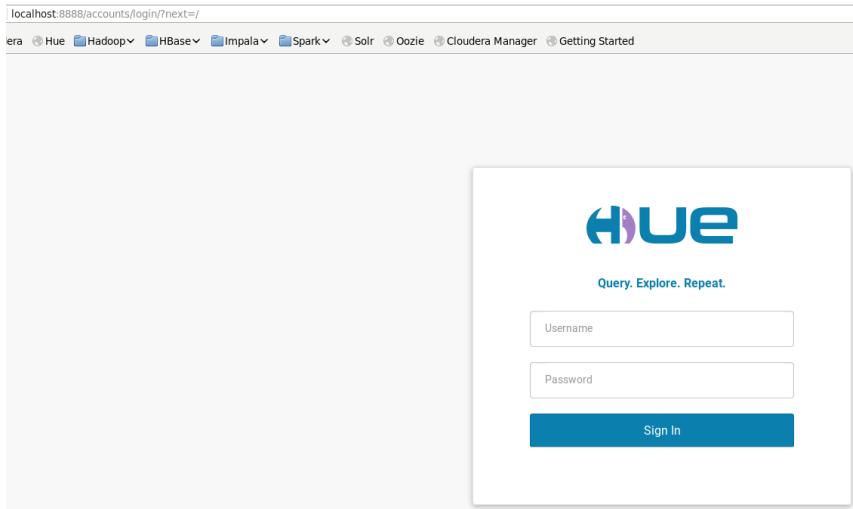
Then execute an aggregate query. For example, count the number of all rows in the **flights** table.

```
SELECT COUNT(*) FROM flights;
```

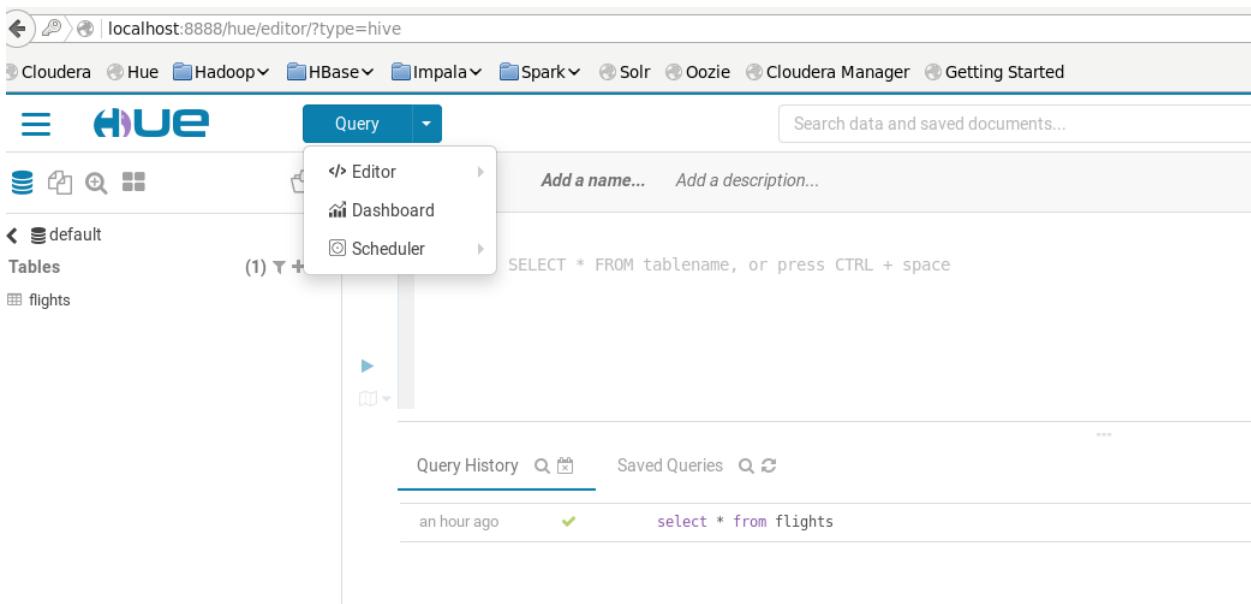
Examine the output. Based on the output, what can you say about the underlying technique used to execute this query?

## 4. Query data using GUI

We're going to use Hue's Hive query editor app to query our tables. Hue provides a web-based interface for many of the tools in CDH and can be found on port 8888 of your Master Node (<http://localhost:8888>). In the QuickStart VM, the administrator username for Hue is 'cloudera' and the password is 'cloudera'.



Once you are inside of Hue, click on Query Editors, and open the Hive Query Editor.



Execute several queries to select rows based on some specific criteria. Also write queries to answer the questions we set earlier in Step 1.

## Congratulations!

You have successfully completed your seminar on Hive. Now you know how to create a table, load data from file and query the data in CLI or GUI. As a continuation to learning Hive, try to do the additional seminar and learn how to load data to Hive from a relational database.

# Seminar 11.

## Exploring SparkSQL and Spark DataFrames

Apache Spark is an analytics engine for large-scale data processing. Spark SQL is a Spark module for structured data processing. Unlike the basic Spark RDD API, the interfaces provided by Spark SQL provide Spark with more information about the structure of both the data and the computation being performed. Internally, Spark SQL uses this extra information to perform extra optimizations. There are several ways to interact with Spark SQL including SQL and the Dataset API. In this seminar we will explore Spark SQL. By the end of the seminar you will know how to load csv file into SparkSQL, how to filter rows and columns of a Spark DataFrame and how to group and perform aggregate functions on columns in a Spark DataFrame. Before running this seminar please follow the instructions to install Anaconda platform on Cloudera VM.

### 1. Preparing for Data Processing.

Before you can use Spark, you must upload the source data to HDFS. We will analyze Australian weather data. File weatherAUS.zip available on Intranet and contains a csv file. Download this file to your virtual machine, extract the content, and then upload the csv file to HDFS

```
[cloudera@quickstart ~]$ unzip -o weatherAUS.zip
```

```
[cloudera@quickstart ~]$ hdfs dfs -put weatherAUS.csv  
/user/cloudera/data
```

We can use this data to answer, for example, the questions like:

- a. What are the min, max temperatures per day in each location?
- b. What was the average temperatures at 3 pm in each location?

c. What was the total rainfall in each location?

etc. During the query steps for this seminar, please write queries to answer these questions.

## 2. Starting Pyspark

Now that you have uploaded the source data, we will use IPython Notebooks to work with Spark. Verify that the files have been uploaded to the shared cluster storage. In the console, enter the following command to view the contents of the /user/cloudera/data folder in the HDFS file system.

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/data
```

The IPython Notebook is now known as the Jupyter Notebook. It is an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media. We will use it to run python commands. Enter the following command to start the Notebook.

```
[cloudera@quickstart ~]$ pyspark -- packages com.databricks:spark-csv_2.10:1.5.0
```

Note that we specify a package to load when the Pyspark is starting. This package will help us to load csv files as DataFrames

Pyspark will start and a new webpage will open. When the web browser shows the default file tree create a new “Python 3” notebook:

The screenshot shows the Cloudera Manager interface with the HDFS file tree. The left sidebar lists various services: Cloudera, Hue, Hadoop, HBase, Impala, Spark, Solr, Oozie, Cloudera Manager, and Getting Started. The main area displays the HDFS file tree under the 'Files' tab. A context menu is open over a folder named 'big-data-3', showing options: Upload, New (with sub-options Text File, Folder, Terminal), and Notebooks (with sub-options Python 3).

Type in the following commands:

```
In [1]: from pyspark.sql import SQLContext  
import pandas  
  
In [2]: sqlsc = SQLContext(sc)  
  
In [3]: df = sqlsc.read.format('csv').options(header='true').load('/user/cloudera/data/weatherAUS.csv')
```

```
from pyspark.sql import SQLContext  
import pandas  
  
sqlsc = SQLContext(sc)  
  
df =  
sqlsc.read.format('csv').options(header='true').load('/user/cloudera/data/weatherAUS.csv')
```

The first line imports the SQLContext module, which is needed access SQL databases in Spark. It also loads pandas library for future use. The second line creates a new SQLContext from the SparkContext sc. The third line creates a new Spark DataFrame in the variable df for the csv file.

Now we can execute various commands against the dataset.

### 3. Executing commands

**View Spark DataFrame schema and count rows.** We can call the printSchema() method to view the schema of the DataFrame:

```
In [4]: df.printSchema()  
#df.select('Year').show(5)  
  
root  
|-- Date: string (nullable = true)  
|-- Location: string (nullable = true)  
|-- MinTemp: string (nullable = true)  
|-- MaxTemp: string (nullable = true)  
|-- Rainfall: string (nullable = true)  
|-- Evaporation: string (nullable = true)  
|-- Sunshine: string (nullable = true)  
|-- WindGustDir: string (nullable = true)  
|-- WindGustSpeed: string (nullable = true)  
|-- WindDir9am: string (nullable = true)  
|-- WindDir3pm: string (nullable = true)  
|-- WindSpeed9am: string (nullable = true)  
|-- WindSpeed3pm: string (nullable = true)  
|-- Humidity9am: string (nullable = true)  
|-- Humidity3pm: string (nullable = true)  
|-- Pressure9am: string (nullable = true)  
|-- Pressure3pm: string (nullable = true)  
|-- Cloud9am: string (nullable = true)
```

The description lists the name and data type of each column.

We can also call the count() method to count the number of rows in the DataFrame:

```
In [6]: df.count()
```

```
Out[6]: 142193
```

**View contents of DataFrame.** We can call the show() method to view the contents of the DataFrame. The argument specifies how many rows to display:

```
In [5]: df.show(5)
```

|    | Date       | Location | MinTemp | MaxTemp | Rainfall | Evaporation | Sunshine | WindGustDir | WindGustSpeed | WindDir9am | WindDir3pm | WindSpeed9am | WindSpeed3pm | Humidity9am | Humidity3pm | Pressure9am | Pressure3pm | Cloud9am | Cloud3pm | Temp9am | Temp3pm | RainToday | RISK_MM | RainTomorrow |
|----|------------|----------|---------|---------|----------|-------------|----------|-------------|---------------|------------|------------|--------------|--------------|-------------|-------------|-------------|-------------|----------|----------|---------|---------|-----------|---------|--------------|
| 20 | 2008-12-01 | Albury   | 13.4    | 22.9    | 0.6      | NA          | NA       | W           | 44            | W          | WNW        | No           | 0            | 24          | 71          | 22          | 1007.7      | 1007.1   | 8        | 16.9    | 21.8    | No        | 0       |              |
| 4  | 2008-12-02 | Albury   | 7.4     | 25.1    | 0        | NA          | NA       | WNW         | 44            | NNW        | WSW        | No           | 0            | 22          | 44          | 25          | 1010.6      | 1007.8   | NA       | 17.2    | 24.3    | No        | 0       |              |
| 19 | 2008-12-03 | Albury   | 12.9    | 25.7    | 0        | NA          | NA       | WSW         | 46            | W          | WSW        | No           | 0            | 26          | 38          | 30          | 1007.6      | 1008.7   | 2        | 21      | 23.2    | No        | 0       |              |
| 11 | 2008-12-04 | Albury   | 9.2     | 28      | 0        | NA          | NA       | NE          | 24            | SE         | E          | No           | 1            | 9           | 45          | 16          | 1017.6      | 1012.8   | NA       | 18.1    | 26.5    | No        | 1       |              |
| 7  | 2008-12-05 | Albury   | 17.5    | 32.3    | 1        | NA          | NA       | W           | 41            | ENE        | NW         | No           | 0.2          | 20          | 82          | 33          | 1010.8      | 1006     | 7        | 17.8    | 29.7    | No        | 0.2     |              |

only showing top 5 rows

**Filter columns in DataFrame.** We can filter for one or more columns by calling the select() method:

```
In [7]: df.select('Date', 'Location', 'Temp9am').show(5)
```

```
+-----+-----+-----+
|      Date|Location|Temp9am|
+-----+-----+-----+
|2008-12-01| Albury|   16.9|
|2008-12-02| Albury|   17.2|
|2008-12-03| Albury|    21|
|2008-12-04| Albury|   18.1|
|2008-12-05| Albury|   17.8|
+-----+-----+-----+
only showing top 5 rows
```

**Filter rows based on criteria.** We can also filter for rows that match a specific criteria using filter():

```
In [8]: df.filter(df['Location']=='Brisbane').select('Date', 'Location', 'Temp9am').show(5)
```

```
+-----+-----+-----+
|      Date|Location|Temp9am|
+-----+-----+-----+
|2008-07-01|Brisbane|   14.9|
|2008-07-02|Brisbane|   16.2|
|2008-07-03|Brisbane|   15.4|
|2008-07-04|Brisbane|   14.1|
|2008-07-05|Brisbane|   16.1|
+-----+-----+-----+
only showing top 5 rows
```

The arguments to filter() are a Column, in this case specified as df["Location"], and the condition, which is the city of Brisbane. The remainder of the commander selects only the Date, Location and temperature columns and shows the first five rows.

**Group by a column and count.** The groupBy() method groups the values of column(s). We can calculate how many records we have for each location by grouping by 'Location' column and counting the result:

```
In [10]: df.groupby('Location').count().show()
```

| Location         | count |
|------------------|-------|
| Mildura          | 3007  |
| Wollongong       | 2983  |
| Sale             | 3000  |
| Adelaide         | 3090  |
| Moree            | 2854  |
| Watsonia         | 2999  |
| Dartmoor         | 2943  |
| Woomera          | 2990  |
| Newcastle        | 2955  |
| Hobart           | 3188  |
| SalmonGums       | 2955  |
| Albury           | 3011  |
| Darwin           | 3192  |
| PerthAirport     | 3009  |
| NorahHead        | 2929  |
| Katherine        | 1559  |
| NorfolkIsland    | 2964  |
| Bendigo          | 3034  |
| MelbourneAirport | 3009  |
| Launceston       | 3028  |

only showing top 20 rows

**Calculate average and sum.** Aggregate operations can be performed on columns of DataFrames. First, let's import the Python libraries for the aggregate operations. Next, we can calculate the minimum, average and total values by specifying the min, mean and sum aggregations for specific columns, respectively:

```
In [15]: df.groupby('Location').agg({'MinTemp': 'min', 'Temp9am': 'mean', 'Rainfall': 'sum'}).show()
```

| Location         | min(MinTemp) | sum(Rainfall)      | avg(Temp9am)       |
|------------------|--------------|--------------------|--------------------|
| Mildura          | -0.1         | 2839.800000000001  | 15.530871590153033 |
| Wollongong       | 10           | 10562.800000000008 | 18.208679500168824 |
| Sale             | -0.1         | 4525.899999999975  | 13.406070713809209 |
| Adelaide         | 0.7          | 4747.999999999996  | 16.973192868719593 |
| Moree            | -0.1         | 4474.799999999994  | 18.80441485634196  |
| Watsonia         | -0.1         | 5524.599999999983  | 13.760147354320177 |
| Dartmoor         | -0.1         | 6314.599999999962  | 12.587963277796646 |
| Woomera          | 0.7          | 1462.0000000000027 | 17.874874539979896 |
| Newcastle        | 10           | 9235.200000000013  | 18.10284172661872  |
| Hobart           | -0.4         | 5106.3999999999805 | 12.464072795732664 |
| SalmonGums       | -0.1         | 3035.999999999994  | 17.04334575008465  |
| Albury           | -0.1         | 5759.7999999999765 | 14.348619886930468 |
| Darwin           | 12.5         | 16260.200000000019 | 27.186434837092783 |
| PerthAirport     | -0.2         | 5300.7999999999965 | 18.773479561316044 |
| NorahHead        | 10           | 9768.600000000008  | 18.489388736263734 |
| Katherine        | 10           | 4844.499999999999  | 26.640499671268884 |
| NorfolkIsland    | 10.3         | 9236.999999999984  | 19.74535315985128  |
| Bendigo          | -0.1         | 4912.999999999967  | 13.844788918205824 |
| MelbourneAirport | -0.2         | 4368.999999999999  | 14.124925224327027 |
| Launceston       | -0.1         | 6076.899999999988  | 12.027347883597878 |

only showing top 20 rows

## 4. Using SQL queries on DataFrames

There are two main ways to manipulate DataFrames:

### 1. Imperative manipulation

Using python methods such as `.select` and `.groupby`.

- Advantage: order of operations is specified.
- Disadvantage: You need to describe both the result you want and how to get it.

### 2. Declarative Manipulation (SQL like)

- Advantage: You need to describe only what kind of result you want.
- Disadvantage: SQL does not have primitives for common analysis operations such as covariance

Spark supports a subset of the Hive SQL query language. For example, you can use Hive select syntax to select a subset of the rows in a dataframe. To use sql on a dataframe you need to register it as a TempTable first. Then you can issue an SQL query to get the result:

```
In [24]: df.registerTempTable('weather')

aggregates = sqlsc.sql(
    'SELECT Location, min(MinTemp) as minT, avg(Temp9am) as avgT, sum(Rainfall) as Rain from weather group by Location')
print(aggregates)
count = 0
tbl = aggregates.collect()
for agg in tbl:
    print(['Location'], agg['minT'], agg['avgT'], agg['Rain'])
    count += 1
    if count > 20: # do not print more than 20 results
        break

DataFrame[Location: string, minT: string, avgT: double, Rain: double]
[('Location') -0.1 15.530871590153033 2839.800000000001
 ('Location') 10.18.208679500168824 10562.800000000008
 ('Location') -0.1 13.406070713809209 4525.899999999975
 ('Location') 0.7 16.973192868719593 4747.99999999996
 ('Location') -0.1 18.80441485634196 4474.79999999994
 ('Location') -0.1 13.760147354320177 5524.599999999983
 ('Location') -0.1 12.587963277796646 6314.599999999962
 ('Location') 0.7 17.874874539979896 1462.000000000027
 ('Location') 10.18.10284172661872 9235.200000000013
 ('Location') -0.4 12.464072795732664 5106.3999999999805
 ('Location') -0.1 17.04334575008465 3035.99999999994
 ('Location') -0.1 14.348619886930468 5759.7999999999765
 ('Location') 12.5 27.186434837092783 16260.200000000019
 ('Location') -0.2 18.773479561316044 5300.7999999999965
 ('Location') 10.18.489388736263734 9768.600000000008
 ('Location') 10.26.640499671268884 4844.49999999999
 ('Location') 10.3 19.74535315985128 9236.99999999994
 ('Location') -0.1 13.844788918205824 4912.999999999967
 ('Location') -0.2 14.124925224327027 4368.99999999999
 ('Location') -0.1 12.027347883597878 6076.899999999988
 ('Location') 10.25.63128915265414 10560.0]
```

```
df.registerTempTable('weather')

aggregates = sqlsc.sql(
    'SELECT Location, min(MinTemp) as minT, avg(Temp9am) as avgT,
sum(Rainfall) as Rain from weather group by Location')
print(aggregates)
count = 0
tbl = aggregates.collect()
for agg in tbl:
    print(['Location'], agg['minT'], agg['avgT'], agg['Rain'])
    count += 1
    if count > 20: # do not print more than 20 results
        break
```

## 5. Load data using predefined schema

If you know the schema of the file ahead and do **not** want to use the `inferSchema` option for column names and types, use user-defined custom column names and type using `schema` option:

```
In [35]: from pyspark.sql.types import Row, StructField, StructType, StringType, IntegerType, DoubleType, DateType
```

```
In [55]: schema = StructType([StructField("Date", DateType(), False),
                           StructField("Location", StringType(), False),
                           StructField("MinTemp", DoubleType(), False),
                           StructField("MaxTemp", DoubleType(), False),
                           StructField("Rainfall", DoubleType(), False),
                           StructField("Evaporation", StringType(), False),
                           StructField("Sunshine", StringType(), False),
                           StructField("WindGustDir", StringType(), False),
                           StructField("WindGustSpeed", DoubleType(), False),
                           StructField("WindDir9am", StringType(), False),
                           StructField("WindDir3pm", StringType(), False),
                           StructField("WindSpeed9am", DoubleType(), False),
                           StructField("WindSpeed3pm", DoubleType(), False),
                           StructField("Humidity9am", DoubleType(), False),
                           StructField("Humidity3pm", DoubleType(), False),
                           StructField("Pressure9am", DoubleType(), False),
                           StructField("Pressure3pm", DoubleType(), False),
                           StructField("Cloud9am", IntegerType(), False),
                           StructField("Cloud3pm", IntegerType(), False),
                           StructField("Temp9am", DoubleType(), False),
                           StructField("Temp3pm", DoubleType(), False),
                           StructField("RainToday", StringType(), False),
                           StructField("RISK_MM", DoubleType(), False),
                           StructField("RainTomorrow", StringType(), False)])
```

```
In [56]: df_1 = sqlsc.read.format('csv').options(header='true', nullValues='NA', dateFormat='yyyy-MM-dd').schema(schema).load(
          '/user/cloudera/data/weatherAUS.csv')
df_1.printSchema()
```

```
root
 |-- Date: date (nullable = false)
 |-- Location: string (nullable = false)
 |-- MinTemp: double (nullable = false)
 |-- MaxTemp: double (nullable = false)
 |-- Rainfall: double (nullable = false)
 |-- Evaporation: string (nullable = false)
 |-- Sunshine: string (nullable = false)
 |-- WindGustDir: string (nullable = false)
 |-- WindGustSpeed: double (nullable = false)
 |-- WindDir9am: string (nullable = false)
 |-- WindDir3pm: string (nullable = false)
 |-- WindSpeed9am: double (nullable = false)
 |-- WindSpeed3pm: double (nullable = false)
 |-- Humidity9am: double (nullable = false)
 |-- Humidity3pm: double (nullable = false)
 |-- Pressure9am: double (nullable = false)
 |-- Pressure3pm: double (nullable = false)
 |-- Cloud9am: integer (nullable = false)
 |-- Cloud3pm: integer (nullable = false)
 |-- Temp9am: double (nullable = false)
 |-- Temp3pm: double (nullable = false)
 |-- RainToday: string (nullable = false)
 |-- RISK_MM: double (nullable = false)
 |-- RainTomorrow: string (nullable = false)
```

```
In [59]: df_1.select('Date', 'Location', 'Rainfall').show(5)
```

| Date       | Location | Rainfall |
|------------|----------|----------|
| 2008-12-01 | Albury   | 0.6      |
| 2008-12-02 | Albury   | 0.0      |
| 2008-12-03 | Albury   | 0.0      |
| 2008-12-04 | Albury   | 0.0      |
| 2008-12-05 | Albury   | 1.0      |

```
from pyspark.sql.types import Row, StructField, StructType,
StringType, IntegerType, DoubleType, DateType
```

```

schema = StructType([StructField("Date", DateType(), False),
                     StructField("Location", StringType(), False),
                     StructField("MinTemp", DoubleType(), False),
                     StructField("MaxTemp", DoubleType(), False),
                     StructField("Rainfall", DoubleType(), False),
                     StructField("Evaporation", StringType(), False),
                     StructField("Sunshine", StringType(), False),
                     StructField("WindGustDir", StringType(), False),
                     StructField("WindGustSpeed", DoubleType(),
                     False),
                     StructField("WindDir9am", StringType(), False),
                     StructField("WindDir3pm", StringType(), False),
                     StructField("WindSpeed9am", DoubleType(), False),
                     StructField("WindSpeed3pm", DoubleType(), False),
                     StructField("Humidity9am", DoubleType(), False),
                     StructField("Humidity3pm", DoubleType(), False),
                     StructField("Pressure9am", DoubleType(), False),
                     StructField("Pressure3pm", DoubleType(), False),
                     StructField("Cloud9am", IntegerType(), False),
                     StructField("Cloud3pm", IntegerType(), False),
                     StructField("Temp9am", DoubleType(), False),
                     StructField("Temp3pm", DoubleType(), False),
                     StructField("RainToday", StringType(), False),
                     StructField("RISK_MM", DoubleType(), False),
                     StructField("RainTomorrow", StringType(),
                     False)])
df_1 = sqlsc.read.format('csv').options(header='true',
nullValues='NA', dateFormat='yyyy-MM-dd').schema(schema).load(
    '/user/cloudera/data/weatherAUS.csv')

df_1.printSchema()

df_1.select('Date', 'Location', 'Rainfall').show(5)

```

## Congratulations!

You have successfully completed your seminar on Spark SQL. Now you know how to load csv data from file and query the data in Spark. As a continuation

to learning Spark, try to do the additional seminar and learn how to load data to Spark from a relational database using JDBC.