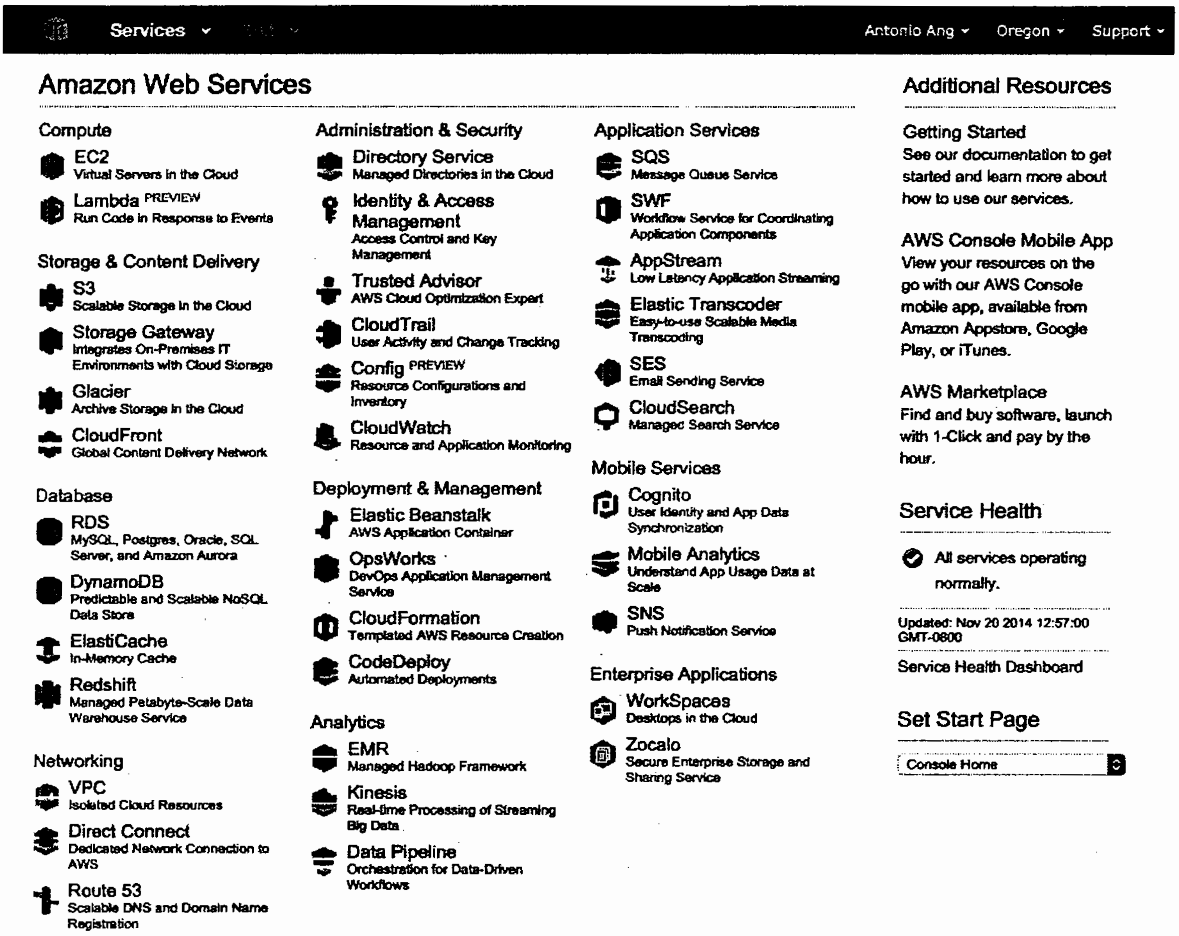
Lab 9

CREATE YOUR FIRST AMAZON RDS DATABASE

STEP 1: Log In to the Amazon Web Service Console

This laboratory experience is about Amazon Web Services and you will use the AWS Management Console in order to complete all the lab steps.



Services ▾ | Antonio Ang ▾ | Oregon ▾ | Support ▾

Amazon Web Services

Compute
- EC2 – Virtual Servers in the Cloud
- Lambda PREVIEW – Run Code in Response to Events

Storage & Content Delivery
- S3 – Scalable Storage in the Cloud
- Storage Gateway – Integrates On-Premises IT Environments with Cloud Storage
- Glacier – Archive Storage in the Cloud
- CloudFront – Global Content Delivery Network

Database
- RDS – MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
- DynamoDB – Predictable and Scalable NoSQL Data Store
- ElastiCache – In-Memory Cache
- Redshift – Managed Petabyte-Scale Data Warehouse Service

Networking
- VPC – Isolated Cloud Resources
- Direct Connect – Dedicated Network Connection to AWS
- Route 53 – Scalable DNS and Domain Name Registration

Administration & Security
- Directory Service – Managed Directories in the Cloud
- Identity & Access Management – Access Control and Key Management
- Trusted Advisor – AWS Cloud Optimization Expert
- CloudTrail – User Activity and Change Tracking
- Config PREVIEW – Resource Configurations and Inventory
- CloudWatch – Resource and Application Monitoring

Deployment & Management
- Elastic Beanstalk – AWS Application Container
- OpsWorks – DevOps Application Management Service
- CloudFormation – Templated AWS Resource Creation
- CodeDeploy – Automated Deployments

Analytics
- EMR – Managed Hadoop Framework
- Kinesis – Real-time Processing of Streaming Big Data
- Data Pipeline – Orchestration for Data-Driven Workflows

Application Services
- SQS – Message Queue Service
- SWF – Workflow Service for Coordinating Application Components
- AppStream – Low Latency Application Streaming
- Elastic Transcoder – Easy-to-use Scalable Media Transcoding
- SES – Email Sending Service
- CloudSearch – Managed Search Service

Mobile Services
- Cognito – User Identity and App Data Synchronization
- Mobile Analytics – Understand App Usage Data at Scale
- SNS – Push Notification Service

Enterprise Applications
- WorkSpaces – Desktops in the Cloud
- Zocalo – Secure Enterprise Storage and Sharing Service

Additional Resources

Getting Started
See our documentation to get started and learn more about how to use our services.

AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.

Service Health

✓ All services operating normally.

Updated: Nov 20 2014 12:57:00 GMT-0800

Service Health Dashboard

Set Start Page

Console Home ▾

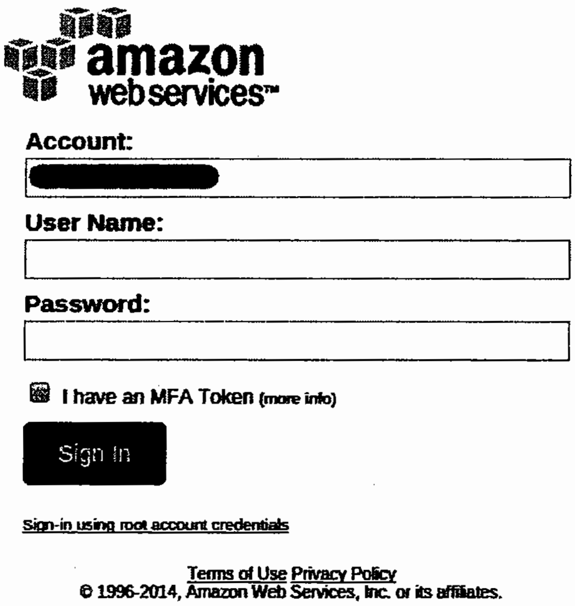
The AWS Management Console is a web control panel for managing all your AWS resources, from EC2 instances to SNS topics. The console enables cloud management for all aspects of the AWS account, including managing security credentials, or even setting up new IAM Users.

Log in to the AWS Management Console

In order to start the laboratory experience, open the Amazon Console by clicking this button:

Open AWS Console

Log in with the username **xxxx** and the password **xxxx** .



amazon web services™

Account:

[redacted]

User Name:

Password:

☒ I have an MFA Token (more info)

Sign In

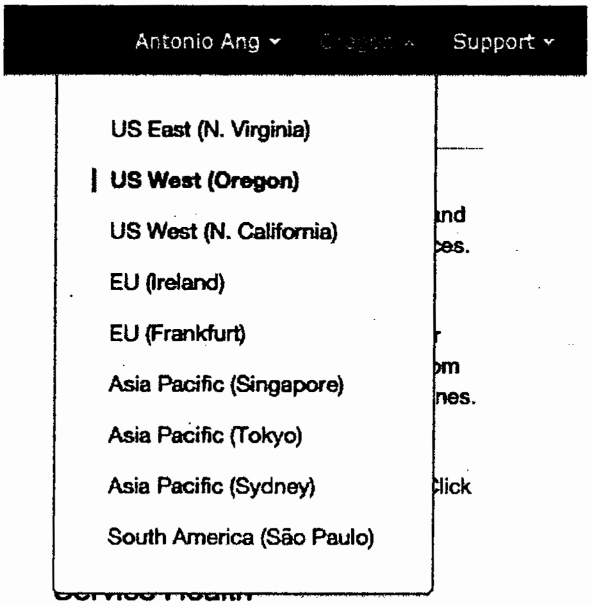
Sign-in using root account credentials

Terms of Use Privacy Policy
© 1996-2014, Amazon Web Services, Inc. or its affiliates.

Select the right AWS Region

Amazon Web Services is available in different regions all over the world, and the console lets you provision resources across multiple regions. You usually choose a region that best suits your business needs to optimize your customer's experience, but you must use the region **US West (Oregon)** for this laboratory.

You can select the **US West (Oregon)** region using the upper right dropdown menu on the AWS Console page.



STEP 2: Create an RDS Subnet Group

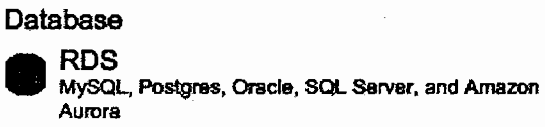
Amazon Relational Database Service (**Amazon RDS**) makes it easy to set up, operate, and scale a relational database in the cloud. Before launching actual RDS instances, you need to configure a **DB Subnet Group**.

Subnets are segments of a VPC's IP address range that allow you to group your resources based on security and operational needs. A **DB Subnet Group** is a collection of subnets (typically private) that you create in a VPC and designate for your DB instances. Each DB subnet group should have subnets in at least two Availability Zones in a given region. Note that SQL Server Mirroring with a SQL Server DB instance requires at least 3 subnets in distinct Availability Zones.

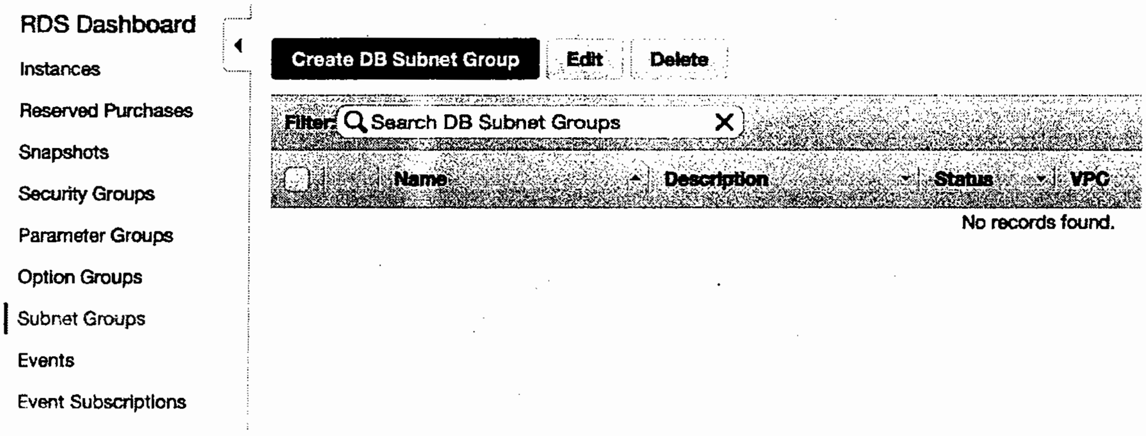
When creating a DB instance in a VPC, you must select a DB subnet group. Amazon RDS uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet to associate with your DB instance. When Amazon RDS creates a DB instance in a VPC, it assigns a network interface to your DB instance by using an IP address selected from your DB Subnet Group. If the primary DB instance of a Multi-AZ deployment fails, Amazon RDS can promote the corresponding standby and subsequently create a new standby using an IP address from an assigned subnet in one of the other Availability Zones.

You can create an RDS Subnet Group using the RDS launch wizard.

Select the RDS service from the Management Console dashboard:



From the RDS dashboard, click **Subnet Groups** from the left-hand menu.

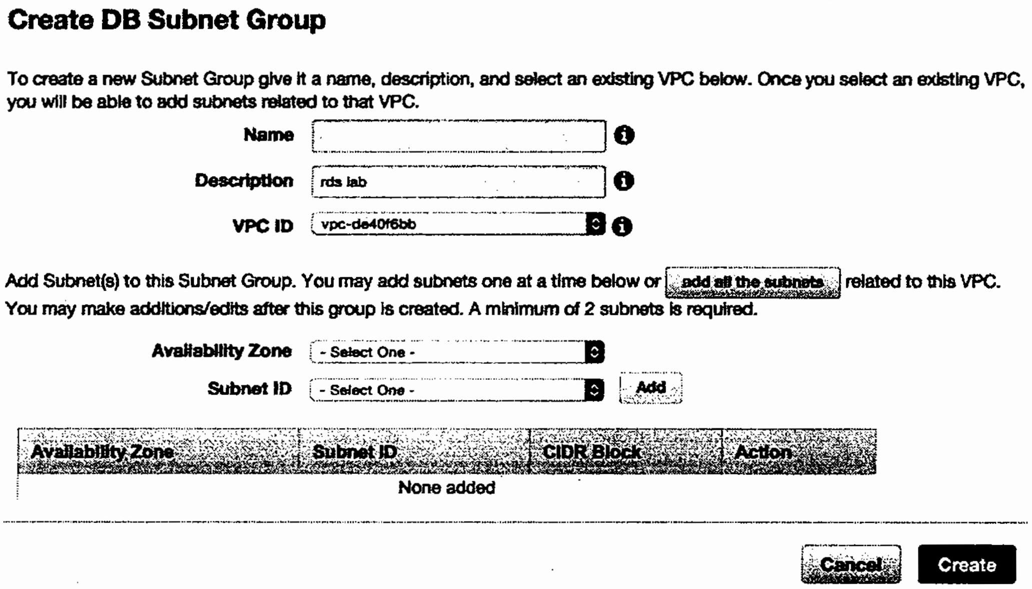


No records found.

Click **Create DB Subnet Group** to open the creation wizard.

You must fill the form using the following data:

- ✓ **Name: vepsun**
- ✓ **Description: rds lab**
- ✓ **VPC ID: select the available one**



Create DB Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

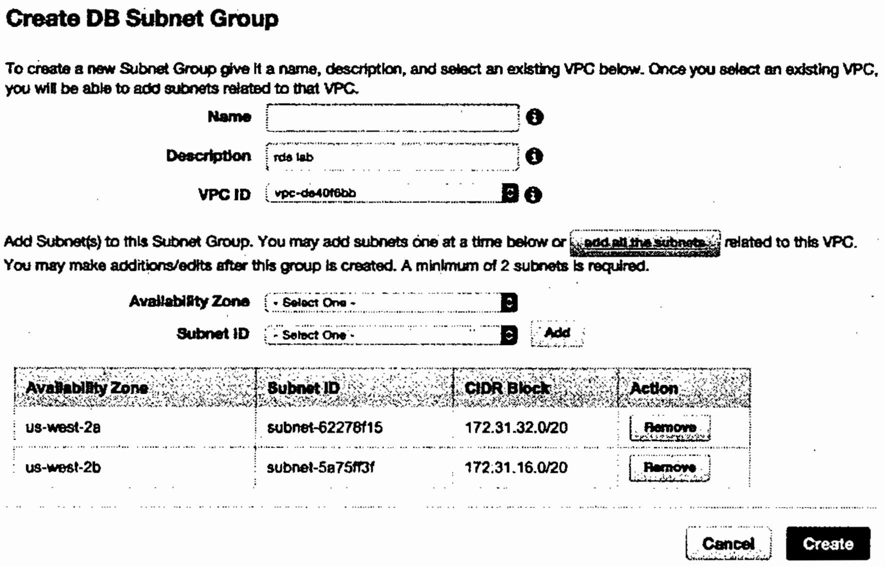
Name	
Description	rds lab
VPC ID	vpc-de40f6bb

Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Availability Zone	- Select One -
Subnet ID	- Select One -

Availability Zone	Subnet ID	CIDR Block	Action
None added			

You must add the available subnets of the selected VPC. Click **add all the subnets** button for filling the subnet list and then click **Create**.



Create DB Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

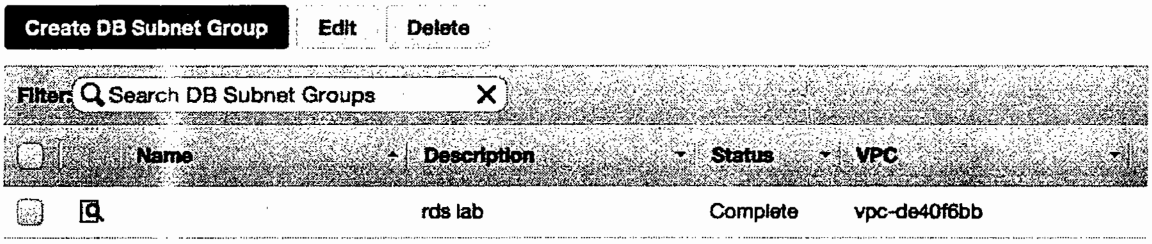
Name	
Description	rds lab
VPC ID	vpc-de40f6bb

Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Availability Zone	- Select One -
Subnet ID	- Select One -

Availability Zone	Subnet ID	CIDR Block	Action
us-west-2a	subnet-62278f15	172.31.32.0/20	Remove
us-west-2b	subnet-5a75ff3f	172.31.16.0/20	Remove

After a few seconds, your DB Subnet Group will be available and ready for use.



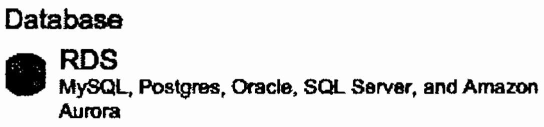
□	Name	Description	Status	VPC
□		rds lab	Complete	vpc-de40f6bb

STEP 3: Create a database cluster using RDS

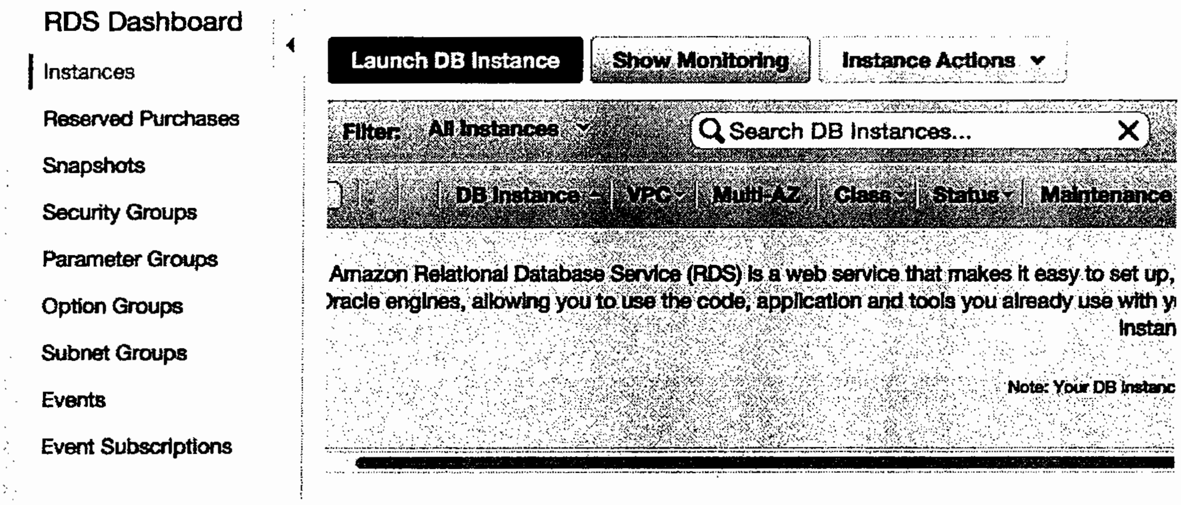
Amazon Relational Database Service (**Amazon RDS**) makes it easy to set up, operate, and scale a relational database in the cloud. Amazon RDS is designed for developers or businesses who require the full features and capabilities of a relational database or who wish to migrate existing applications and tools that utilize a relational database. It gives you access to the capabilities of a **MySQL**, **Oracle**, **Microsoft SQL Server**, **MariaDB**, **Amazon Aurora**, or **PostgreSQL** database engine.

The RDS service is fully managed by Amazon. RDS will make sure that the database software stays up-to-date with the latest patches and any faulty compute instance powering your database deployment will be automatically replaced in the event of a hardware failure. You can automatically or manually create database snapshots and easily scale your infrastructure up or down using the AWS Management Console.

You can launch an RDS instance using the RDS launch wizard. Select the RDS service from the Management Console dashboard:



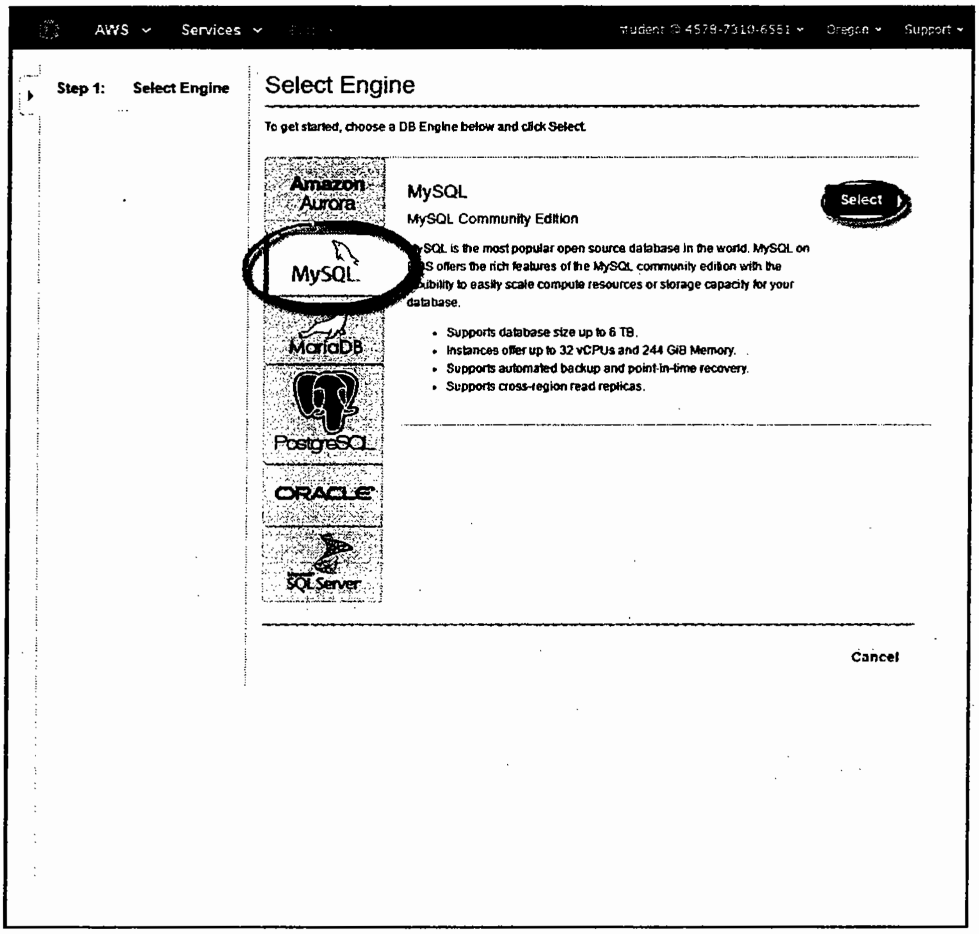
Select **Instances** from the left menu and click **Launch DB Instance.**



Amazon Relational Database Service (RDS) is a web service that makes it easy to set up,
)racle engines, allowing you to use the code, application and tools you already use with y
Instan

Note: Your DB instanc

The **Launch DB Instance Wizard** appears and you must select the database engine. Choose the **MySQL** database engine and click the **Select** button.



Select Engine

To get started, choose a DB Engine below and click Select.

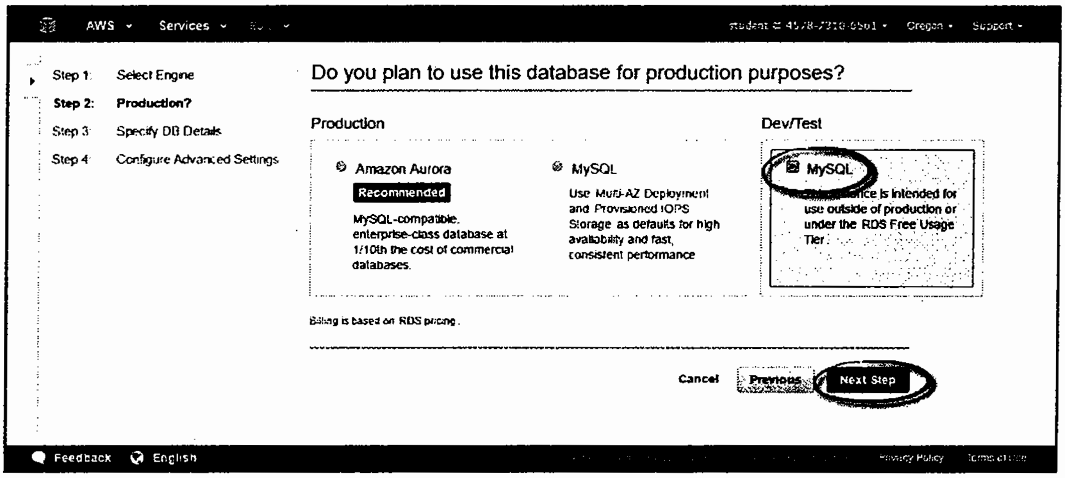
MySQL

MySQL Community Edition

ySQL is the most popular open source database in the world. MySQL on S offers the rich features of the MySQL community edition with the xibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 6 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

The next step asks if you are planning to use the DB instance you are creating for production purposes. By selecting Yes, the failover option, Multi-AZ and the Provisioned IOPS storage option will be preselected in the following step. For this exercise, please select **Dev/Test MySQL**. Then click **Next Step**.



Step 1: Select Engine
Step 2: Production?
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

Do you plan to use this database for production purposes?

Production

○ Amazon Aurora
Recommended
MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases.

○ MySQL
Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance

Dev/Test

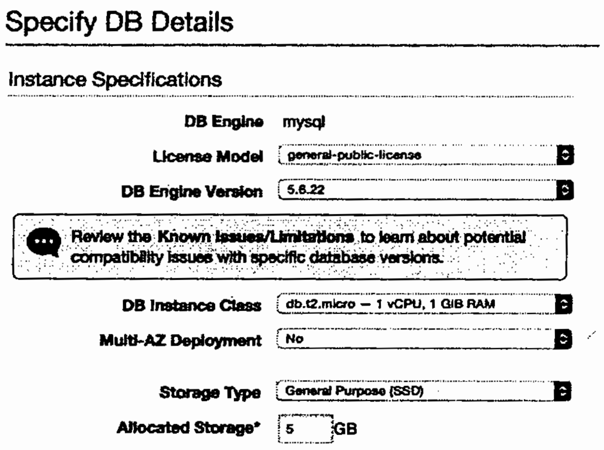
● MySQL
ce is intended for use outside of production or under the RDS Free Usage Tier.

Billing is based on RDS pricing.

Cancel Previous Next Step

On the **Specify DB Details** page, you should select the following options:

- ✓ **License Model: general-public-license**
- ✓ **DB Engine Version: select the latest one**
- ✓ **DB Instance Class: db.t2.micro**
- ✓ **Multi-AZ Deployment: No**
- ✓ **Storage Type: General Purpose (SSD)**
- ✓ **Allocated Storage: 5GB**



Specify DB Details

Instance Specifications

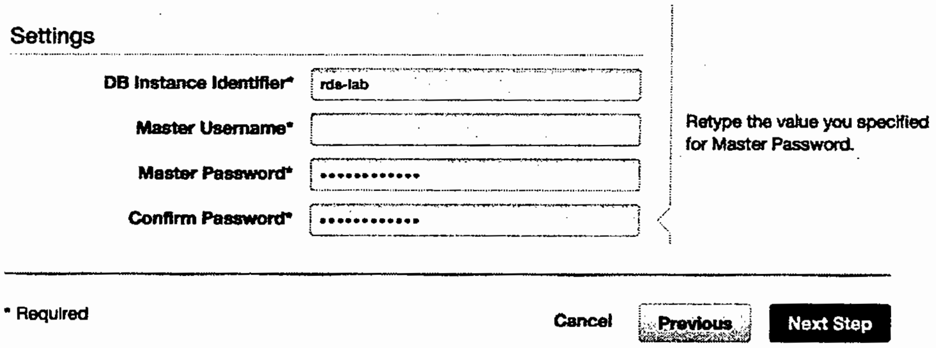
DB Engine	mysql
License Model	general-public-license
DB Engine Version	5.6.22

Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

DB Instance Class	db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment	No
Storage Type	General Purpose (SSD)
Allocated Storage*	5 GB

Scroll down the page and continue configuring your RDS Instance by filling-in the following fields:

- ✓ **DB Instance Identifier**: **rds-lab**
- ✓ **Master Username**: **vepsun**
- ✓ **Master Password**: **myStrongRDSpwd!**



Settings

DB Instance Identifier*	rds-lab
Master Username*	
Master Password*	••••••••••••
Confirm Password*	••••••••••••

Retype the value you specified for Master Password.

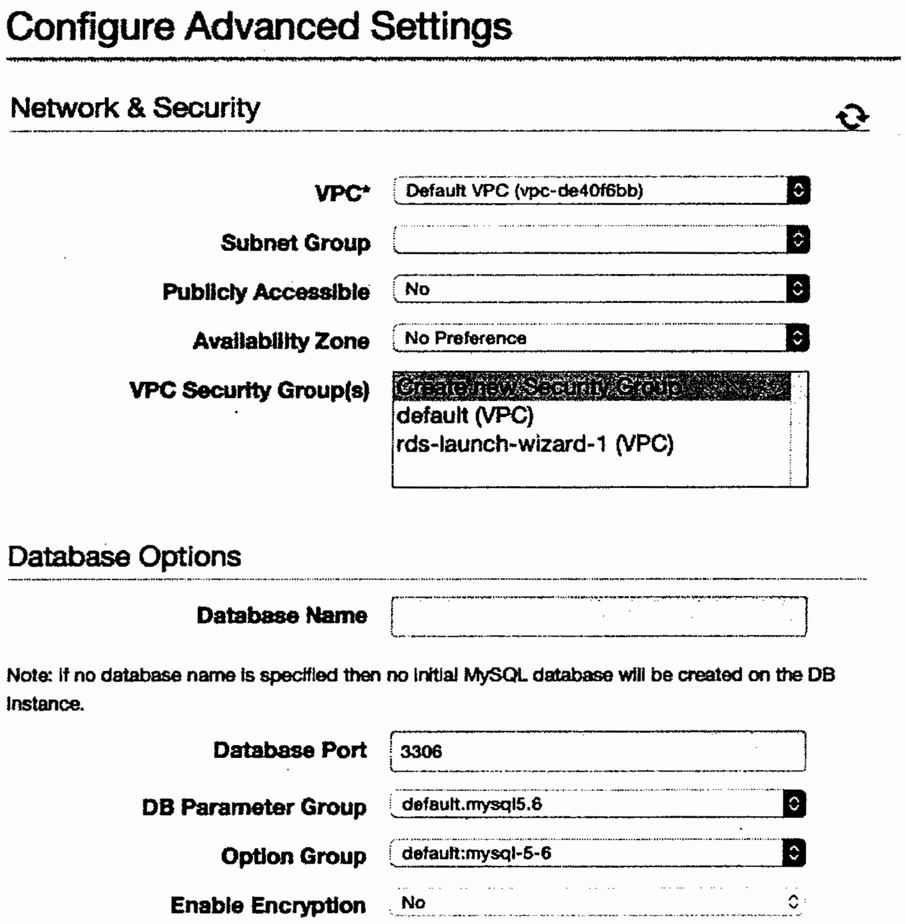
Click **Next Step,** and start configuring the Advanced Settings.

On the **Configure Advanced Settings** page, provide additional information that RDS needs to launch the MySQL DB instance.

You need to select the following options:

- ✓ **VPC**: **select the available one**
- ✓ **Subnet Group**: **vepsun**

- ✓ **Publicly Accessible**: **No**
- ✓ **Availability Zone**: **us-west-2a**
- ✓ **VPC Security Group(s)**: **Create new Security Group**
- ✓ **Database Name**: **rdsappdb**
- ✓ **Database Port**: **3306**
- ✓ **DB Parameter Group**: **default**
- ✓ **Option Group**: **default**
- ✓ **Enable Encryption**: **No**



Configure Advanced Settings

Network & Security

VPC* Default VPC (vpc-de40f6bb)

Subnet Group

Publicly Accessible No

Availability Zone No Preference

VPC Security Group(s) Create new Security Group
default (VPC)
rds-launch-wizard-1 (VPC)

Database Options

Database Name

Note: If no database name is specified then no initial MySQL database will be created on the DB instance.

Database Port 3306

DB Parameter Group default.mysql5.6

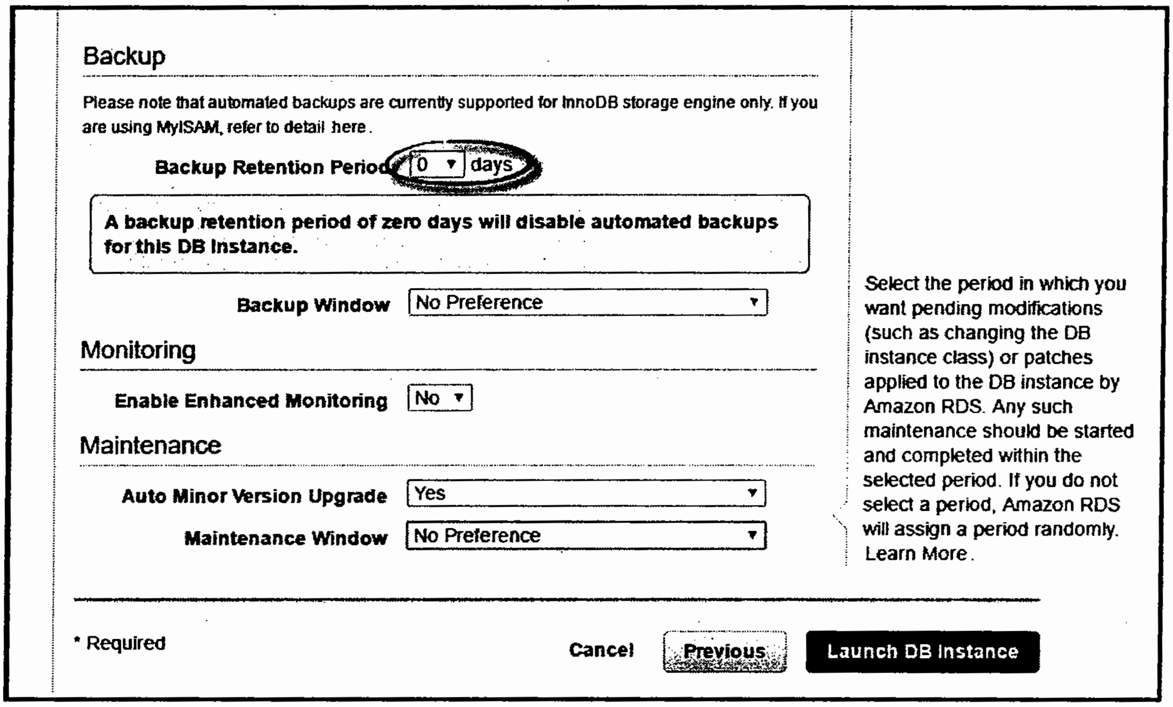
Option Group default:mysql-5-6

Enable Encryption No

Continue specifying the following settings for the **Backup** and **Maintenance** options:

- ✓ **Backup Retention Period**: **0 days**

- ✓ **Backup Window: No preference**
- ✓ **Auto Minor Version Upgrade: Yes**
- ✓ **Maintenance Window: No preference**



Backup

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail here.

Backup Retention Period 0 days

A backup retention period of zero days will disable automated backups for this DB Instance.

Backup Window No Preference

Monitoring

Enable Enhanced Monitoring No

Maintenance

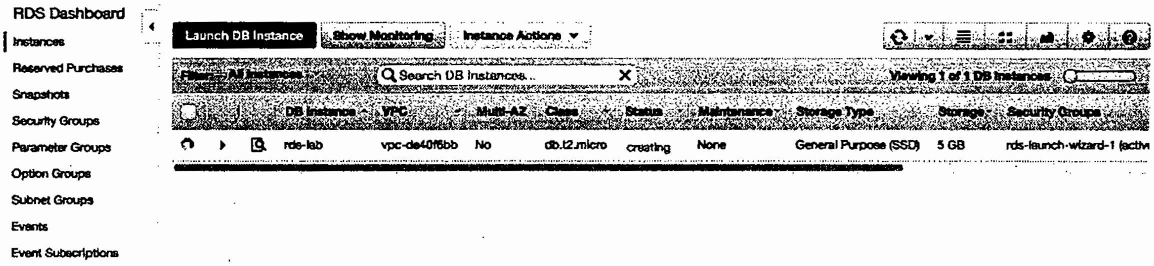
Auto Minor Version Upgrade Yes

Maintenance Window No Preference

Select the period in which you want pending modifications (such as changing the DB instance class) or patches applied to the DB instance by Amazon RDS. Any such maintenance should be started and completed within the selected period. If you do not select a period, Amazon RDS will assign a period randomly. Learn More.

* Required

Click **Launch DB Instance,** and then click **View Your DB Instances**.



DB Instance	VPC	Multi-AZ	Class	Status	Maintenance	Storage Type	Storage	Security Groups
rds-lab	vpc-de40f6bb	No	db.t2.micro	creating	None	General Purpose (SSD)	5 GB	rds-launch-wizard-1 (activ

N.B., RDS instance creation requires **up to 10 minutes** for completion. Wait until its status becomes *available*.

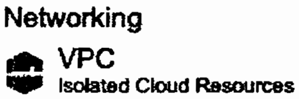
STEP 4: Setup security group rules for connecting to the RDS instance

In order to use the previously created RDS instance inside the VPC, you need to **add an inbound rule to the VPC Security Group** created during the RDS instance creation.

The rules of a **Security Group** control the inbound traffic that's allowed to reach the instances that are associated with the security group and the outbound traffic that's allowed to leave them. By default, security groups allow all outbound traffic and deny all inbound traffic.

You can add new rules to a **VPC Security Group** using the AWS Management Console.

Select the VPC service from the Management Console dashboard:



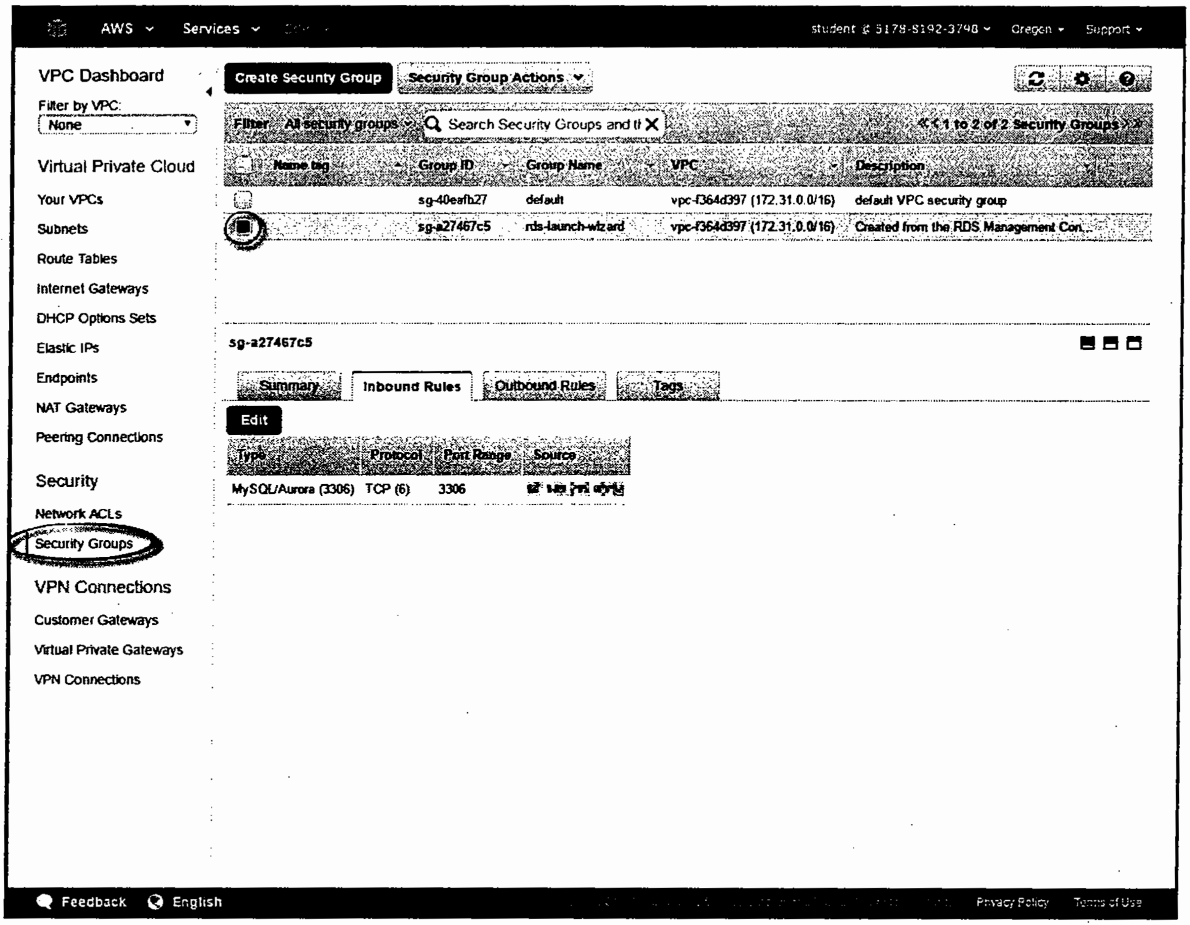
Networking

VPC

Isolated Cloud Resources

Note: You've been working in the RDS console. Make sure you go back to the main console and select the **VPC** console.

In the navigation pane, click **Security Groups.** Locate and click the **rds-launch-wizard** security group.

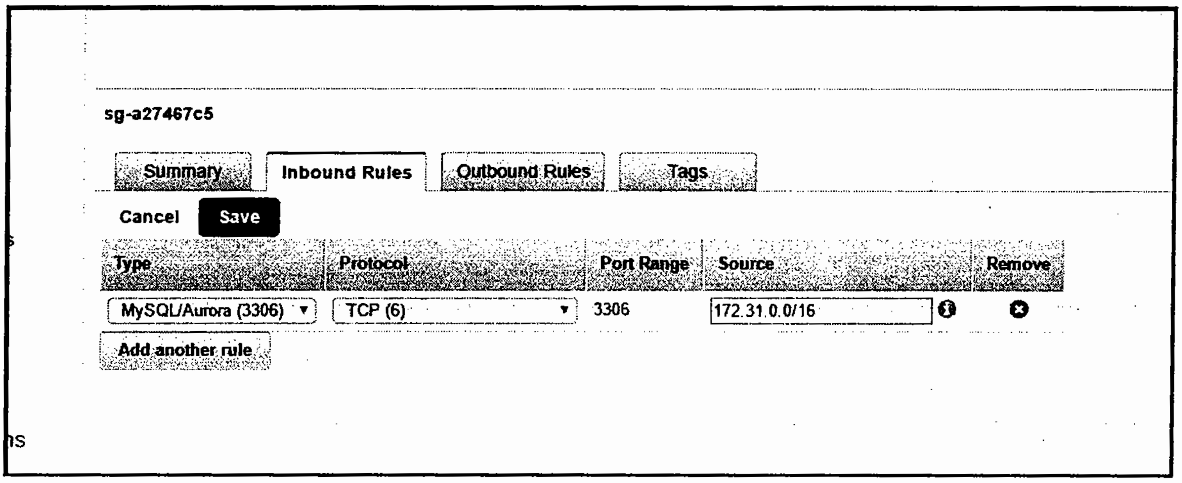


Name tag	Group ID	Group Name	VPC	Description
	sg-40eafb27	default	vpc-f364d397 (172.31.0.0/16)	default VPC security group
	sg-a27467c5	rds-launch-wizard	vpc-f364d397 (172.31.0.0/16)	Created from the RDS Management Con...

sg-a27467c5

Type	Protocol	Port Range	Source
MySQL/Aurora (3306)	TCP (6)	3306	[illegible]

On the **Inbound Rules** tab, click **Edit**.



sg-a27467c5

Type	Protocol	Port Range	Source	Remove
MySQL/Aurora (3306)	TCP (6)	3306	172.31.0.0/16	

Ensure that the rule is completed using the following information:

- ✓ **Type**: MYSQL
- ✓ **Protocol**: TCP
- ✓ **Port**: 3306
- ✓ **Source**: 172.31.0.0/16

Click **Save**, and you will be ready to connect to your RDS instance inside the VPC.

STEP 5: Create an EC2 instance

You can launch an EC2 instance using the EC2 launch wizard.

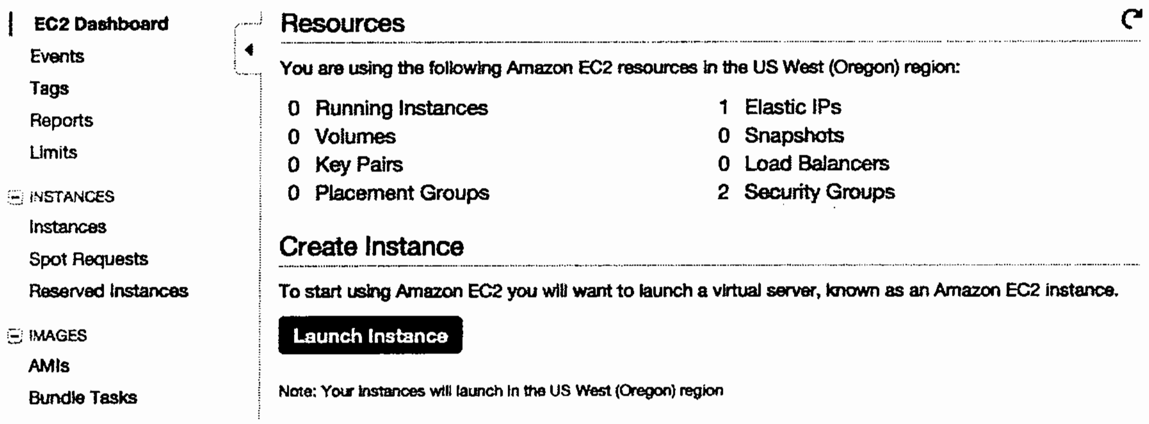
Select the EC2 service from the Management Console dashboard:

Compute

EC2

Virtual Servers in the Cloud

From the EC2 dashboard, click **Launch Instance**.



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

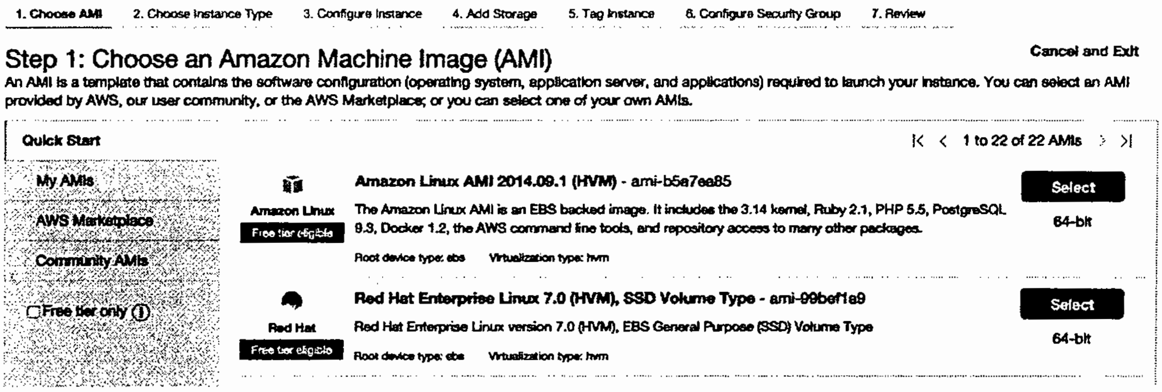
0 Running Instances
0 Volumes
0 Key Pairs
0 Placement Groups
1 Elastic IPs
0 Snapshots
0 Load Balancers
2 Security Groups

Create Instance

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

Note: Your instances will launch in the US West (Oregon) region

The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called **Amazon Machine Images (AMIs)** that serve as templates for your instance. Select the first listed 64-bit **Amazon Linux AMI**.



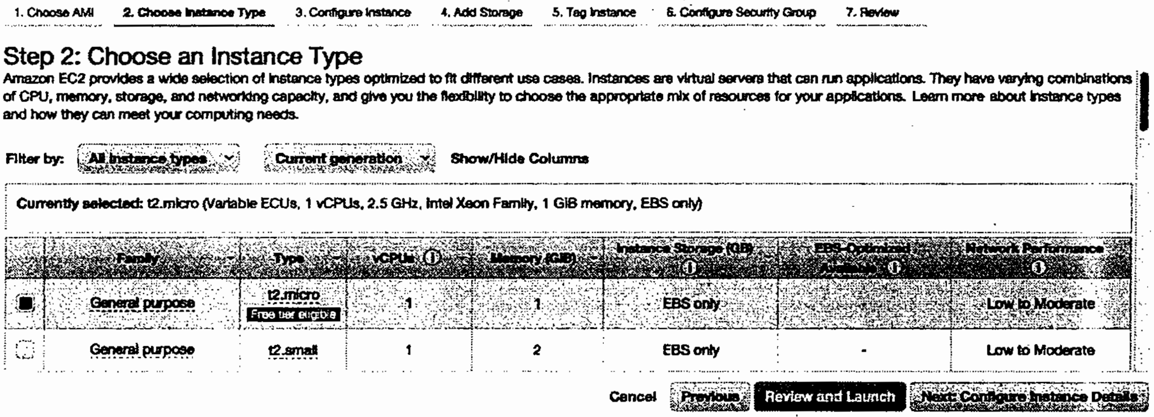
Step 1: Choose an Amazon Machine Image (AMI)

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.

Amazon Linux AMI 2014.09.1 (HVM) - ami-b5a7ea85
Amazon Linux – Free tier eligible – 64-bit
The Amazon Linux AMI is an EBS backed image. It includes the 3.14 kernel, Ruby 2.1, PHP 5.5, PostgreSQL 9.3, Docker 1.2, the AWS command line tools, and repository access to many other packages.
Root device type: ebs   Virtualization type: hvm

Red Hat Enterprise Linux 7.0 (HVM), SSD Volume Type - ami-99bef1a9
Red Hat – Free tier eligible – 64-bit
Red Hat Enterprise Linux version 7.0 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs   Virtualization type: hvm

On the **Choose an Instance Type** page, do **not** change any options and click **Next: Configure Instance Details.**



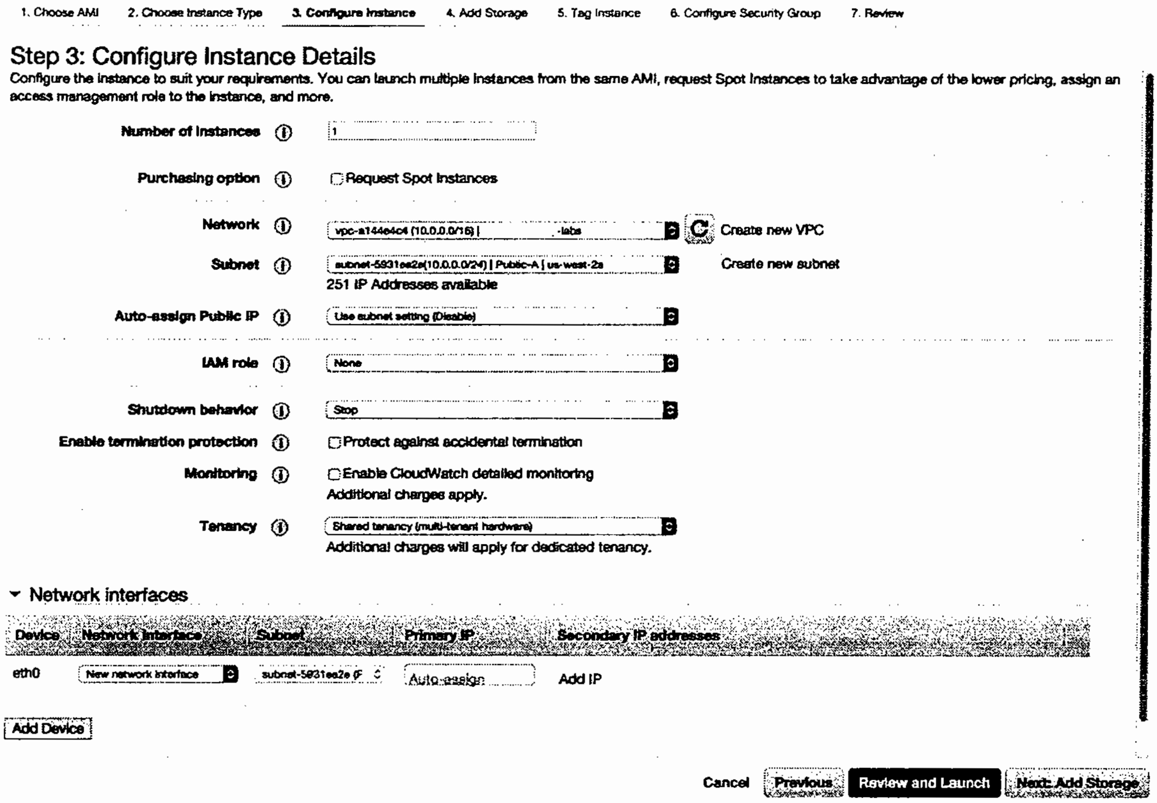
Step 2: Choose an Instance Type

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.

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
■	General purpose	t2.micro Free tier eligible	1	1	EBS only		Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate

On the **Configure Instance Details** tab, check the selected **Network (VPC)** and **Subnet**. Change them, if needed, and then click **Next: Add Storage**.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances: 1

Purchasing option: □ Request Spot Instances

Network: vpc-a144e4c4 (10.0.0.0/16) | -labs — Create new VPC

Subnet: subnet-5931ea2a(10.0.0.0/24) | Public-A | us-west-2a — Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None

Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared tenancy (multi-tenant hardware)
Additional charges will apply for dedicated tenancy.

Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses
eth0	New network interface	subnet-5931ea2a (F	Auto-assign	Add IP

On the **Add Storage** tab, do **not** change any options, and click the **Review and Launch** button.

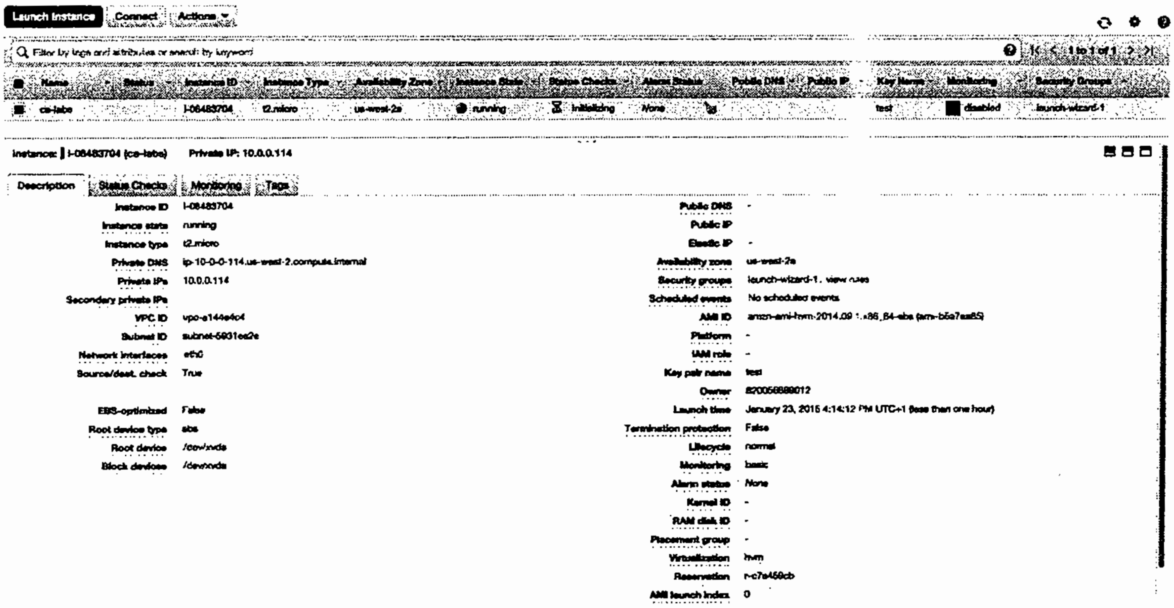
On the Review Instance Launch page, click **Launch**.

In the **Select an existing key pair or create a new key pair** dialog box, select **Create a new key pair**, then type a KeyPair name (e.g., "TestKeys") and download it.

Select the acknowledgment checkbox, and then click **Launch Instances**.

A confirmation page will let you know that your instance is launching. Click **View Instances** to close the confirmation page and return to the console.

On the Instances Screen, you can view the status of your instance. It will take a short time for your instance to be launched. When you launch an instance, its initial state defaults to *pending*. After the instance starts, its Instance State changes to *running*, and it receives a public DNS name.



Name	Status	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Security Groups
cs-labs		i-08483704	t2.micro	us-west-2a	running	Initializing	None			test	disabled	launch-wizard-1

Instance: i-08483704 (cs-labs) Private IP: 10.0.0.114

Instance ID	i-08483704	Public DNS	-
Instance state	running	Public IP	
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-10-0-0-114.us-west-2.compute.internal	Availability zone	us-west-2a
Private IPs	10.0.0.114	Security groups	launch-wizard-1. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-a144e4c4	AMI ID	amzn-ami-hvm-2014.09.1.x86_64-ebs (ami-b5a7ea85)
Subnet ID	subnet-5931ea2e	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	test
		Owner	820056889012
EBS-optimized	False	Launch time	January 23, 2015 4:14:12 PM UTC+1 (less than one hour)
Root device type	ebs	Termination protection	False
Root device	/dev/xvda	Lifecycle	normal
Block devices	/dev/xvda	Monitoring	basic
		Alarm status	None
		Kernel ID	-
		RAM disk ID	-
		Placement group	-
		Virtualization	hvm
		Reservation	r-c7e459cb
		AMI launch index	0

STEP 6: Connect to a remote shell using an SSH connection

In order to manage a remote Linux server, you must employ an **SSH Client**. Secure Shell (SSH) is a cryptographic network protocol for securing data communication. It establishes a secure channel over an insecure network. Common applications include remote command-line login and remote command execution.

Connect using Linux / Mac OS

Linux distributions and Mac OS are shipped with a fully working SSH client that accepts standard PEM Keys.

Starting a remote SSH session is easy:

- ✓ Open your **Terminal** application
- ✓ Write and run the following command: **`ssh -i /path/to/your/keypair.pem user@server-ip`**.

`server-ip` is the Public IP of your server, you can find it in the EC2 instance details

`user` is the remote system user that will be used for the remote authentication

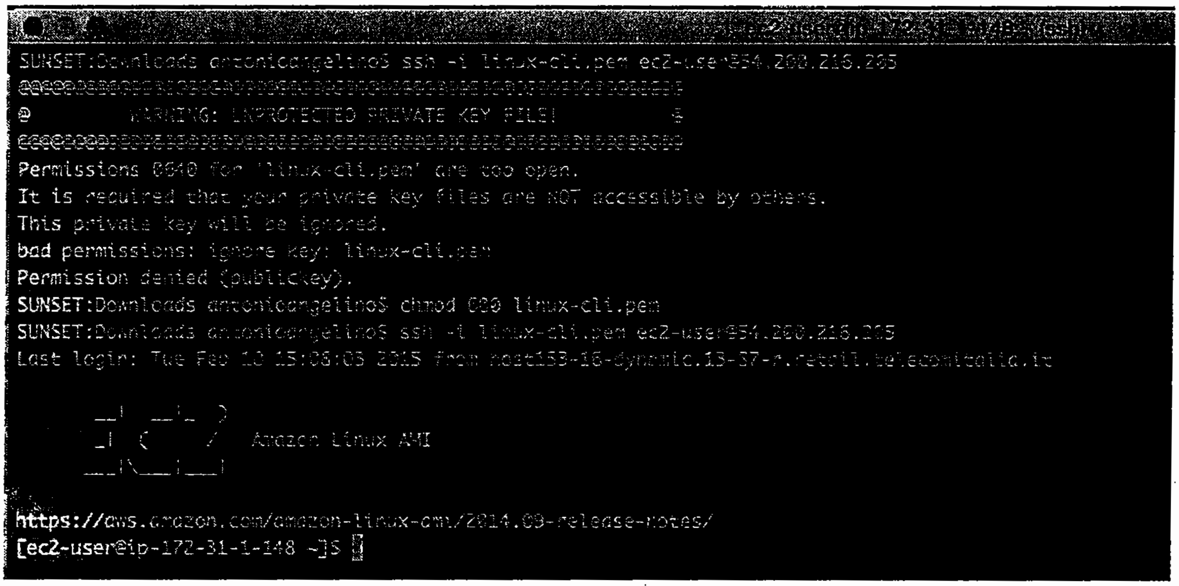
Amazon Linux AMIs typically use **`ec2-user`** as username.

Ubuntu AMIs login user is **ubuntu**, Debian AMIs use **admin** instead.

Assuming that you selected the Amazon Linux AMI, your assigned public IP is 123.123.123.123, and your keypair (named "keypair.pem") is stored in /home/youruser/keypair.pem, the right command to run is: **ssh -i /home/youruser/keypair.pem ec2-user@123.123.123.123**

Note: your SSH Client may refuse to start the connection, warning that the key file is unprotected. You should deny the file access to any other system user by changing its permissions. Issue the following command and then try again:

chmod 600 /home/youruser/keypair.pem



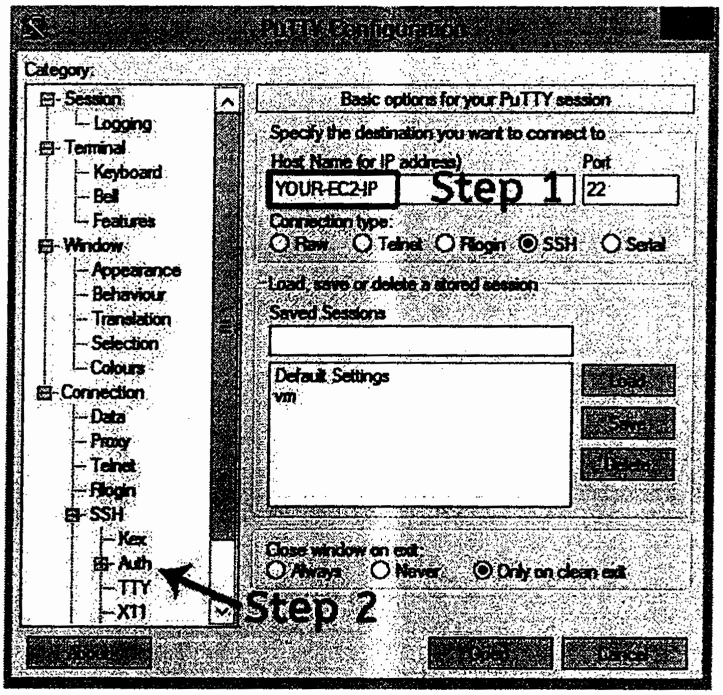
```
SUNSET:Downloads antonioangelino$ ssh -i linux-cli.pem ec2-user@54.200.216.205
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for 'linux-cli.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: linux-cli.pem
Permission denied (publickey).
SUNSET:Downloads antonioangelino$ chmod 600 linux-cli.pem
SUNSET:Downloads antonioangelino$ ssh -i linux-cli.pem ec2-user@54.200.216.205
Last login: Tue Feb 10 15:06:03 2015 from host153-16-dynamic.13-37-r.retail.telecomitalia.it

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
[ec2-user@ip-172-31-1-148 ~]$
```

Connect using Windows

Windows has no SSH client, so you must use PuTTY and convert the PEM key to PPK using PuTTYgen.

Starting a remote SSH session using PuTTY is easy:

- ✓ Open PuTTY and insert the EC2 instance IP Address in the Host Name field.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): YOUR-EC2-IP Step 1 Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

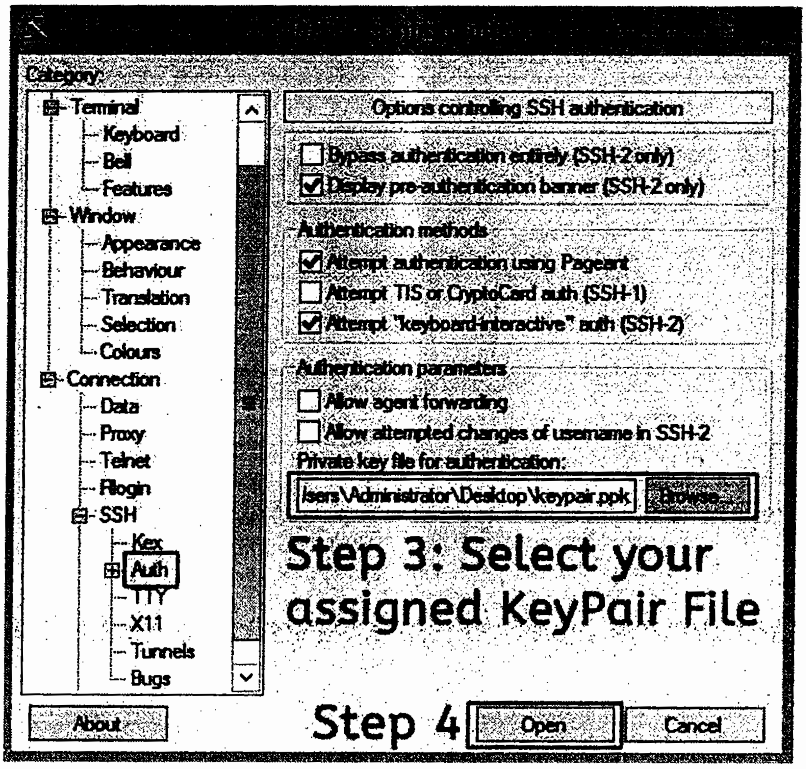
Saved Sessions

Default Settings
vm

Close window on exit: ○ Always ○ Never ● Only on clean exit

Step 2

- ✓ Select **Connection > SSH > Auth** section and then select the downloaded Keypair that you previously converted to PPK format.



Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:
lsers\Administrator\Desktop\keypair.ppk

Step 3: Select your assigned KeyPair File

Step 4

✓ After some seconds, you will see the authentication form. **Login as** **ec2-user** and you will see the EC2 server welcome banner.

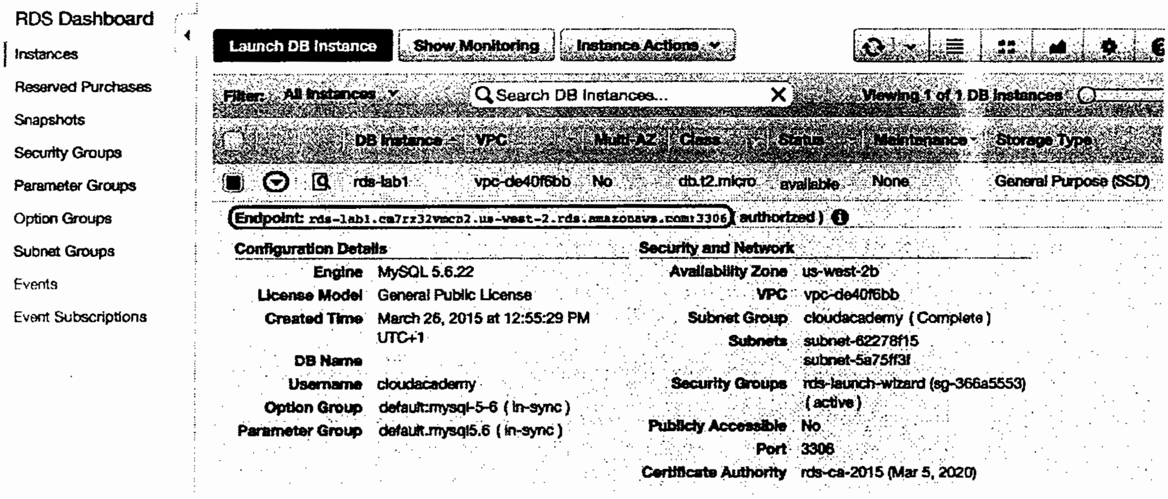
STEP 7: Connect to RDS and create a database table

Your RDS instance is ready and accessible from any EC2 instance created within the same VPC, so you can connect to the previously created EC2 instance and use the database.

The MySQL client software isn't usually installed in the Linux Amazon AMI, but you can download and install it using the YUM package manager.

Execute the following command: **sudo yum -y install mysql**

You are ready to connect to your RDS instance using the **RDS Endpoint URL**. It's automatically generated during the RDS instance creation and you can find it in the RDS instance details box.



DB Instance	VPC	Multi-AZ	Class	Status	Maintenance	Storage Type
rds-lab1	vpc-de40f6bb	No	db.t2.micro	available	None	General Purpose (SSD)

Endpoint: rds-lab1.ca7rz32vmcn2.us-west-2.rds.amazonaws.com:3306 (authorized)

Configuration Details

Engine	MySQL 5.6.22
License Model	General Public License
Created Time	March 26, 2015 at 12:55:29 PM UTC+1
DB Name	
Username	cloudacademy
Option Group	default:mysql-5-6 (in-sync)
Parameter Group	default.mysql5.6 (in-sync)

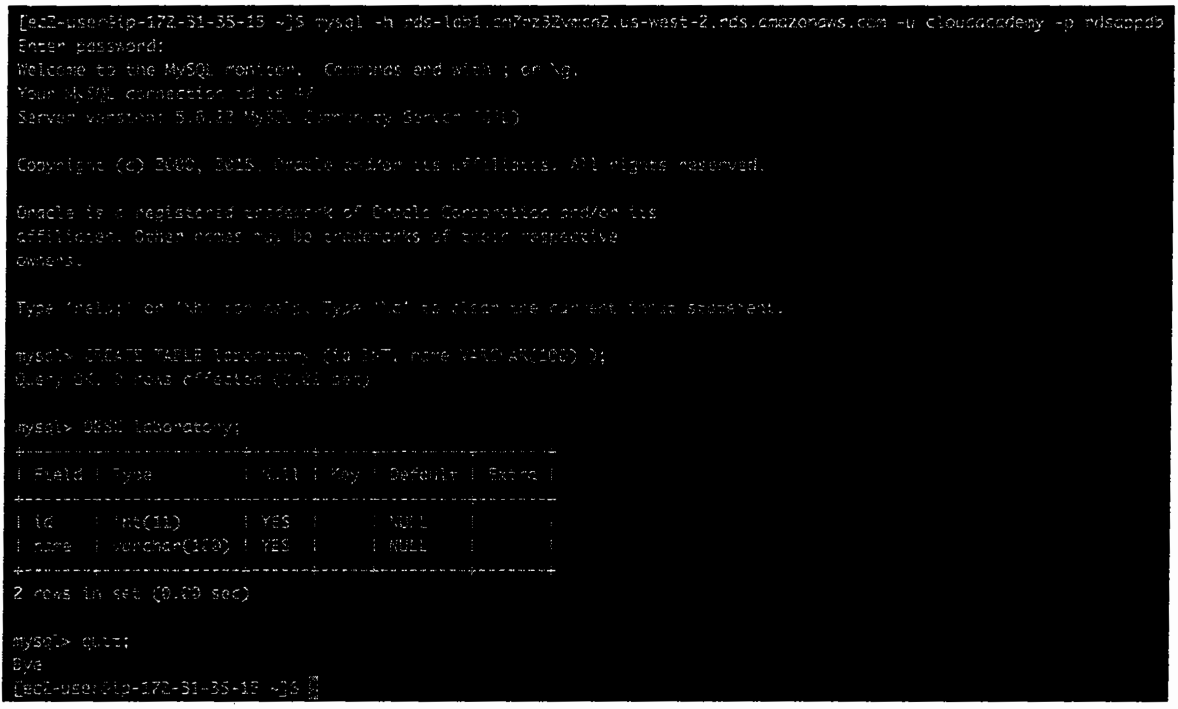
Security and Network

Availability Zone	us-west-2b
VPC	vpc-de40f6bb
Subnet Group	cloudacademy (Complete)
Subnets	subnet-62278f15 subnet-5a75ff3f
Security Groups	rds-launch-wizard (sg-366a5553) (active)
Publicly Accessible	No
Port	3306
Certificate Authority	rds-ca-2015 (Mar 5, 2020)

In order to complete this lab step, you must connect to your instance, using the RDS Endpoint URL, and create a table named "laboratory" in the database "rdsappdb".
Connect to your RDS instance by using the mysql client and replace *your.endpoint.aws.com* with your real endpoint URL (w/o including the host port): **mysql -h your.endpoint.aws.com -u vepsun -p rdsappdb**

Insert the DB Master Password (**myStrongRDSpwd!**) and the MySQL console will be ready to accept SQL queries.

Note: remember **NOT** to include the port number of your endpoint URL (e.g. ":3306") or you will not be able to connect to the remote host.



```
[ec2-user@ip-172-31-35-15 ~]$ mysql -h rds-lab1.cn7rz32vmcn2.us-west-2.rds.amazonaws.com -u cloudacademy -p rdsappdb
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.6.23 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE TABLE laboratory (id INT, name VARCHAR(100) );
Query OK, 0 rows affected (0.02 sec)

mysql> DESC laboratory;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| id    | int(11)      | YES  |     | NULL    |       |
| name  | varchar(100) | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> quit;
Bye
[ec2-user@ip-172-31-35-15 ~]$
```

Create a new table in the "rdsappdb" database writing **CREATE TABLE laboratory (id INT, name VARCHAR(100));** and then press ENTER.

You can check if the table was successfully created using the SQL command: **DESC laboratory;**

Close your database connection writing the command: **quit;**

STEP 8: Destroy an RDS instance

You can delete an RDS Instance using the AWS Management Console.

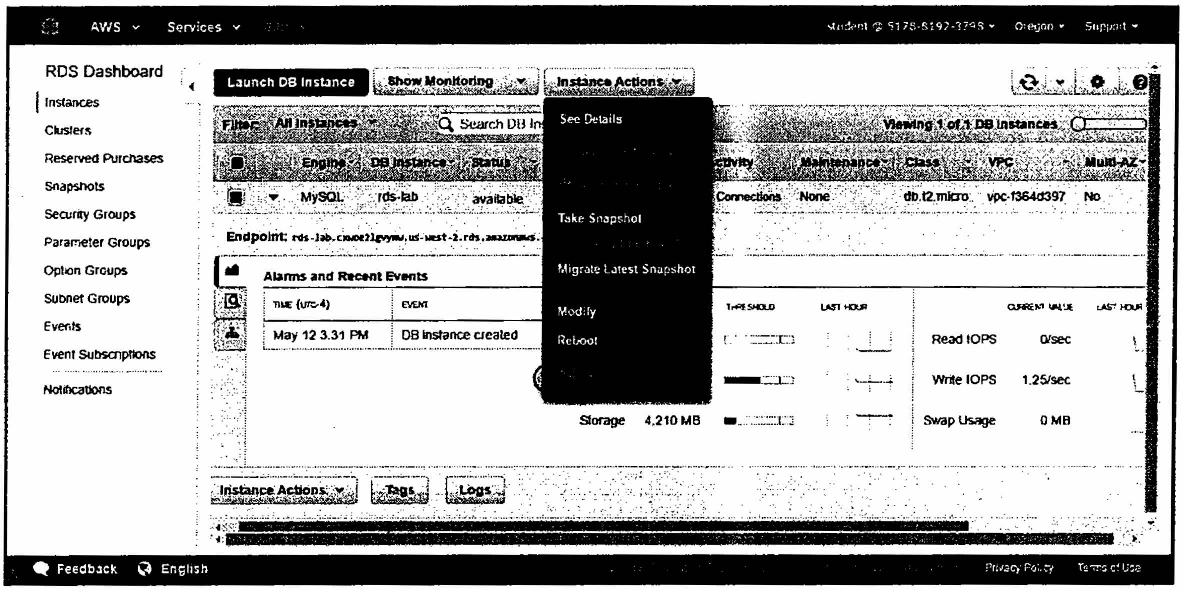
Select the RDS service from the Management Console dashboard:

Database

RDS
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora

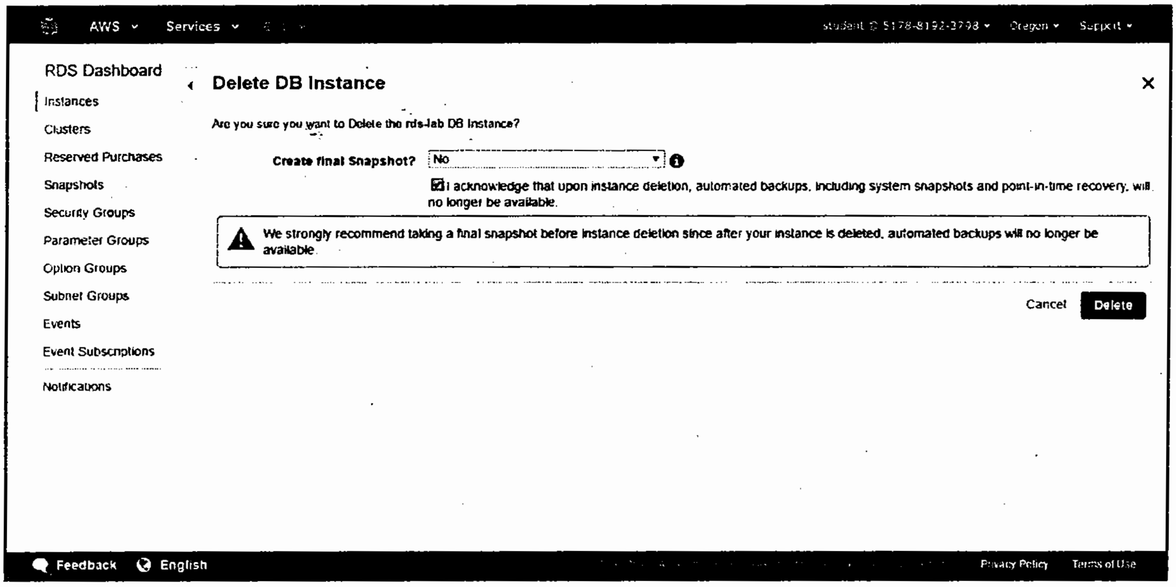
From the RDS dashboard, click **Instances** from the left menu.

Select rds-lab from the RDS Instances list. Click on the **Instance Actions** button, then select**Delete**.



TIME (UTC-4)	EVENT
May 12 3.31 PM	DB instance created

On the Delete DB Instance page, answer **No** to the *Create final Snaphot?* query. Check the acknowledgement checkbox, and click **Delete**.



Delete DB Instance

Are you sure you want to Delete the rds-lab DB Instance?

Create final Snapshot? No

☑ I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.

We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.

Your RDS instance is now in the **deleting** status, which will take a few minutes to complete.