



Database Week

Hands-On Lab: Getting Started with Amazon RDS

June 2019

Table of Contents

| | |
|---|-----------|
| Table of Contents | 2 |
| Overview | 3 |
| Prerequisites | 3 |
| <i>Create an EC2 Key Pair</i> | 3 |
| <i>Launch an EC2 Instance as the Web Server</i> | 5 |
| <i>Create VPC Security Group for the DB Instance</i> | 9 |
| Launch an RDS DB Instance | 11 |
| Connect the RDS DB Instance to the Web Server | 15 |
| Working with RDS DB Instances | 16 |
| <i>Backup and Restore using RDS Snapshots</i> | 16 |
| <i>Scale up the Compute Capacity for an RDS DB Instance</i> | 18 |
| <i>Monitoring RDS DB Instances</i> | 21 |
| Extra Credit | 23 |
| Cleaning Up | 23 |

Overview

Amazon RDS is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.

In this hands-on lab, we will create an Amazon RDS database instance, connect to it using an example web application and learn how to perform basic operations such as snapshots and scaling compute.

The lab includes a few setup steps to ensure all prerequisites are met for you to be able to successfully launch the database instance.

Prerequisites

In order to successfully provision and use a DB instance there are a few minimum prerequisite configurations and resources that you need to set up.

Create an EC2 Key Pair



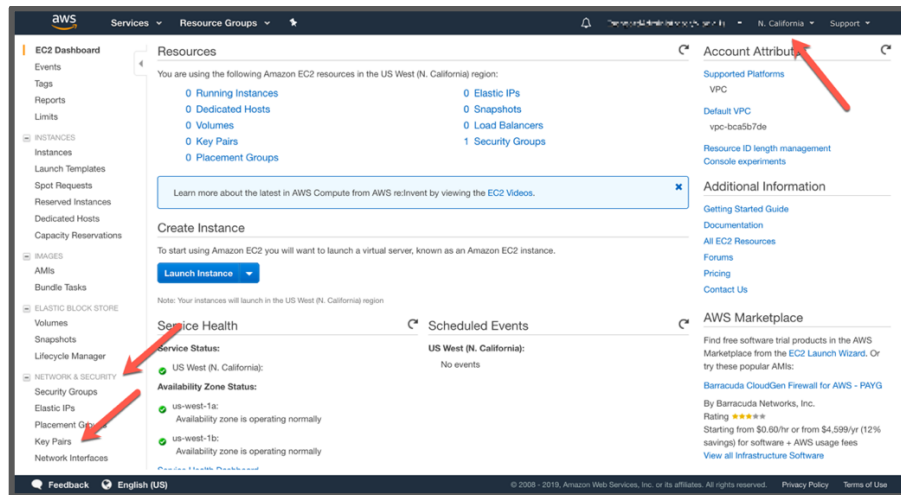
If you have previously completed the **Getting Started with EC2 Linux** lab and still have the key pair available, you may re-use it.

EC2 Key Pairs are used to connect securely to your EC2 Linux-based instances using SSH.

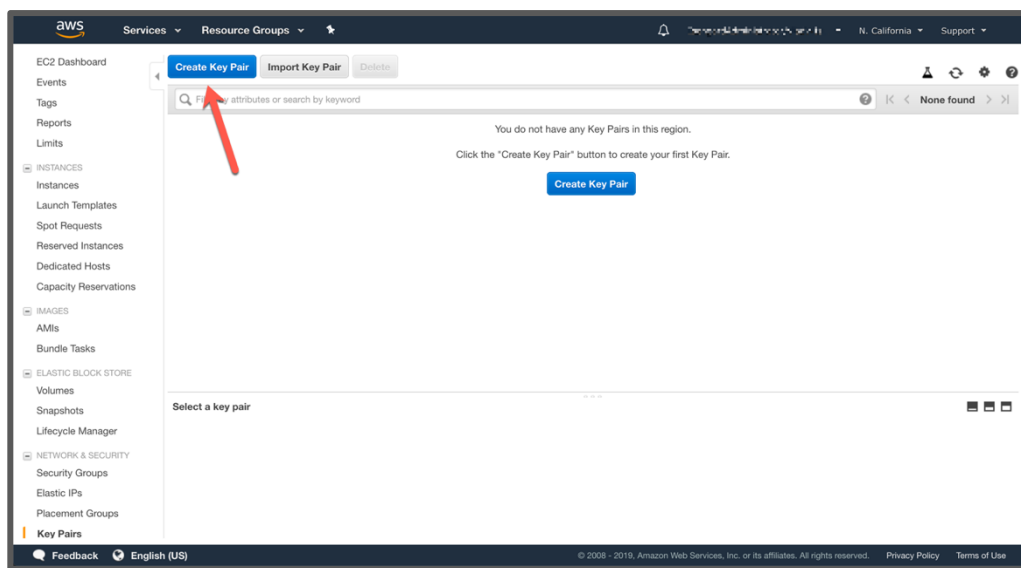
Sign into the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2>.

In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

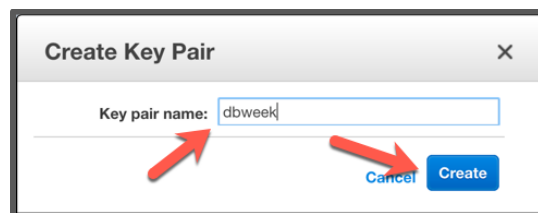
Click on **Key Pairs** in the **NETWORK & SECURITY** section near the bottom of the left-hand menu. This will display a page to manage your SSH key pairs.



Click Create Key Pair.



Name the key pair “dbweek”, or another memorable name, then click **Create** and download the file with the same name (e.g. **dbweek.pem**) to your computer, save it in a memorable location like your desktop.



Launch an EC2 Instance as the Web Server



If you have previously completed the **Getting Started with EC2 Linux lab** and still have the EC2 instance running, you may re-use it.

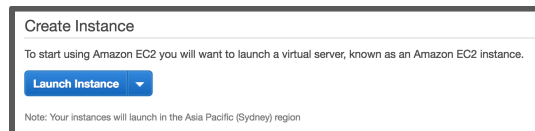
In this section we will launch an Amazon Linux 2 EC2 instance, bootstrap Apache/PHP, and install a basic web application.

EC2 instances are launched within a [virtual private network](#) (VPC), configured with the correct topology in terms of network subnets, routing tables, gateways and other network resources. Setting up a VPC is beyond the scope of this exercise. Each AWS account is automatically created with a **Default VPC** in each region, containing a basic network configuration, where resources can be provisioned in any of that region's availability zones, and can have direct access to the Internet. In rare circumstances, customers can re-purpose these default VPCs, or delete them entirely. If you cannot find the default VPC in your account and region while following the steps in this section, please contact a lab assistant.

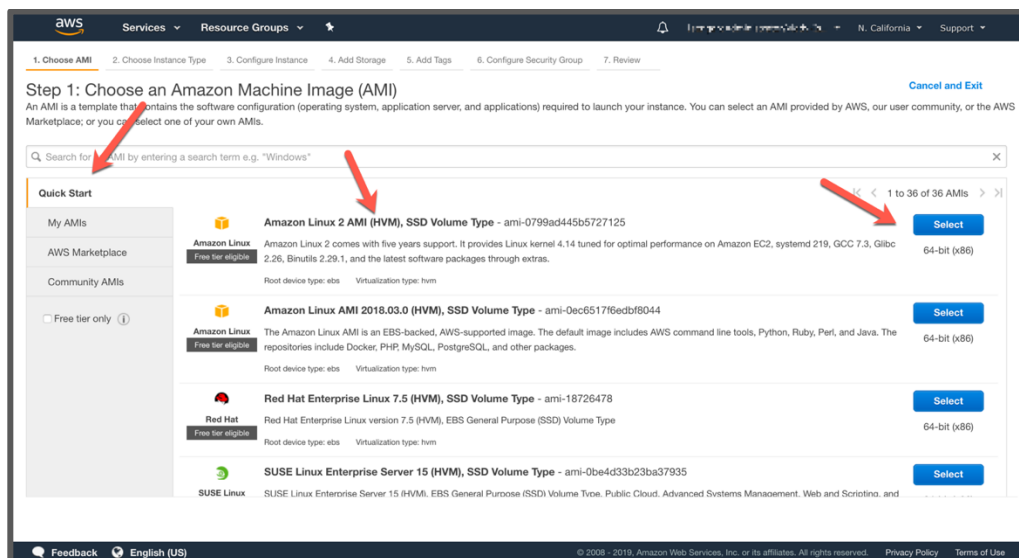
Sign into the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2>, if the console is not already open

In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

Navigate in the left-hand menu to **EC2 Dashboard** or **Instances**, then click **Launch Instance**.



In the Quick Start section of Step 1: Choose an Amazon Machine Image (AMI), select the first Amazon Linux 2 AMI (HVM), SSD Volume Type option and click Select.



On the **Step 2: Choose Instance Type** screen, select the **t2.micro** instance size and click **Next: Configure Instance Details** in the bottom right corner.



If it isn't labeled **Free Tier Eligible** you may incur a charge!

On the **Step 3: Configure Instance Details** screen, accept the defaults, but scroll down and expand the **Advanced Details** section. Copy/paste the script below into the **User Data** field (this shell script will install Apache & PHP, start the web service, and deploy a simple web page). Click **Next: Add Storage** in the bottom right corner.



'User data' is a method for bootstrapping your instance. Any code placed here will be executed the first time an instance is launched.

```
#include  
https://s3.amazonaws.com/immersionday-labs/bootstrap.sh
```

On the **Step 4: Add Storage** screen, you have the ability to modify or add storage and disk drives to the instance. For this lab, we will simply accept the storage defaults and click **Next: Add Tags** in the bottom right corner.

On the **Step 5: Add Tags** screen, you can name your EC2 instance by creating a **Name** tag. This Name will appear in the Management Console once the instance launches. It makes it easy to keep track of running machines in a complex environment. Click the **Add Tag** button, write **Name** under the **Key** column, and write a memorable name like “DB Week Web Server” in the **Value** column. Click **Next: Configure Security Group** in the bottom right corner.

The screenshot shows the AWS Management Console interface for 'Step 5: Add Tags'. The breadcrumb trail at the top indicates the following steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (current step), 6. Configure Security Group, and 7. Review. The main heading is 'Step 5: Add Tags'. Below it, a description states: 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.' The interface features a table with four columns: 'Key' (127 characters maximum), 'Value' (255 characters maximum), 'Instances' (1), and 'Volumes' (1). A single row is present with 'Name' in the 'Key' column and 'DB Week Web Server' in the 'Value' column. Below the table is a button labeled 'Add another tag' with the text '(Up to 50 tags maximum)'. At the bottom of the console, there are four buttons: 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'. A red arrow points to the 'Next: Configure Security Group' button.

On the **Step 6: Configure Security Group** screen, you will be prompted to create a new **Security Group**, which will contain your firewall rules. Since we are building out a Web server, name your new security group “DB Week Web Tier” or a similarly memorable name, and confirm an existing SSH rule exists which allows TCP port 22 from Anywhere (or 0.0.0.0/0). Click the **Add Rule** button, to add

an additional rule.

Select **HTTP** from the **Type** dropdown menu, and confirm **TCP** port **80** is allowed from **Anywhere** (you'll notice, that "Anywhere is the same as '0.0.0.0/0'). Click **Review and Launch** in the bottom right corner.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, you can add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

| Type | Protocol | Port Range | Source | Description |
|------|----------|------------|--------------------|----------------------------|
| SSH | TCP | 22 | Custom 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| HTTP | TCP | 80 | Anywhere 0.0.0.0/0 | e.g. SSH for Admin Desktop |

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Review your configuration and choices, and then click **Launch** in the bottom right corner.

At the **Select an existing key pair...** prompt make sure to choose the option **Choose and existing key pair** in the first dropdown. Select the key pair that you created in the beginning of this lab from the second drop-down and check the "I acknowledge [...]" checkbox. Then click the **Launch Instances** button.

Select an existing key pair or create a new key pair

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.

☒ 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.](#)

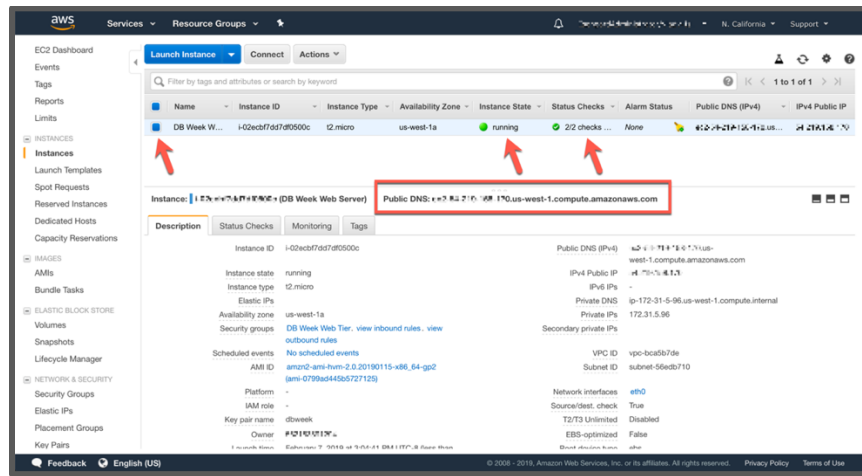
Choose an existing key pair

Select a key pair

☒ I acknowledge that I have access to the selected private key file (dbweek.pem), and that without this file, I won't be able to log into my instance.

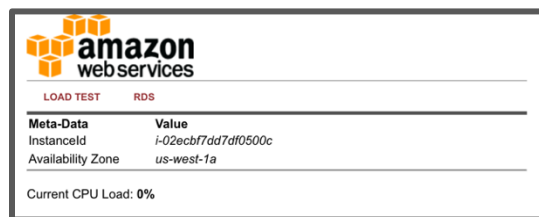
Click the **View Instances** button in the lower righthand portion of the screen to view the list of EC2 instances. Once your instance has launched, you will see your web server as well as the Availability Zone the instance is in, and the publicly routable DNS name.

Click the checkbox next to your web server to view details about this EC2 instance.



Wait for the **Instance state** to change to **running** and to show **2/2 checks passed** in the **Status Checks** column.

Open a new browser tab and navigate to the web server interfaces by entering the EC2 instance's **Public DNS** name into the browser. The EC2 instance's Public DNS name can be found in the console by reviewing the Public DNS name line highlighted in the preceding screenshot. You should see a website that looks like the example below:



Create VPC Security Group for the DB Instance

The RDS servers have the same security model as Amazon EC2: trust nothing. A common use of an RDS instance in a VPC is to share data with an application server running in an EC2 instance in the same VPC. In this lab, the web server EC2 instance you just created, can be accessed directly over the Internet, and that web server will then initiate database connections to the RDS DB instance.

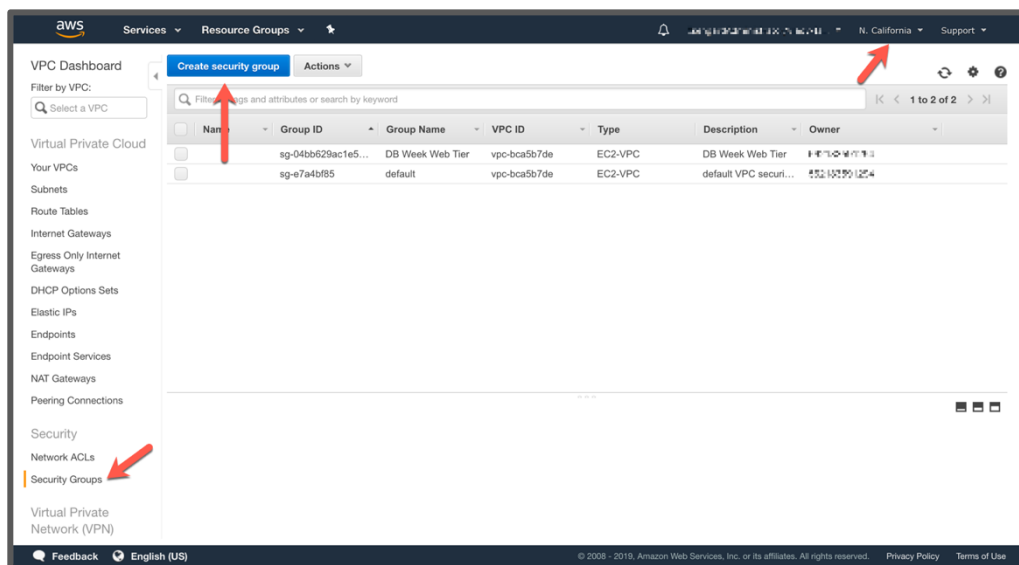
To this end, we'll need to utilize a VPC security group to permit the EC2 instance to access the RDS DB instance. Because workloads in AWS are elastic, and the number of actual EC2 instances typically can change, we will reference the security group of the web server as a permitted source of traffic to the database, instead of the IP address of the web server itself. This ensures that if we decide later to add more web server EC2 instances (or remove some), and we configure them to use the same web tier security group, we do not have to change the database security group to allow access. Access will be granted automatically by the membership in the web tier security group.

Let's create a new VPC security group for our database tier that only allows traffic from our web server (the EC2 instance we created previously).

Sign into the AWS Management Console and open the Amazon VPC console at <https://console.aws.amazon.com/vpc>, if the console is not already open, or showing a different service console.

In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

In the VPC dashboard, click **Security Groups**, then the **Create Security Group** button.



Set the **Security group name** and **Description** to a memorable name “DB Week Database”. Under **VPC**, keep the VPC setting to the same VPC you’ve launched your EC2 instance in (typically the *Default VPC* which may be unnamed). Then click **Create**.

Security Groups > Create security group

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group fill in the fields below.

Security group name* DB Week Database

Description* DB Week Database

VPC vpc-bca5b7de

* Required

Cancel Create

After your VPC security group is created (click **Close** on the confirmation screen), you'll see it listed along with the other security groups in your account. Check the box next to it, to see the details of it in the lower pane on the screen.

VPC Dashboard

Filter by VPC: Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Virtual Private Network (VPN)

Create security group Actions

Filter by tags and attributes or search by keyword

| Name | Group ID | Group Name | VPC ID | Type | Description | Owner |
|-------------------------------------|----------------------|------------------|--------------|---------|-----------------------|--------------|
| <input checked="" type="checkbox"/> | sg-02d2cf47759a3a129 | DB Week Database | vpc-bca5b7de | EC2-VPC | DB Week Database | 111111111111 |
| <input type="checkbox"/> | sg-04bb629ac1e5ee649 | DB Week Web Tier | vpc-bca5b7de | EC2-VPC | DB Week Web Tier | 111111111111 |
| <input type="checkbox"/> | sg-e7a4bfb5 | default | vpc-bca5b7de | EC2-VPC | default VPC securi... | 111111111111 |

Security Group: sg-02d2cf47759a3a129

Description Inbound Rules Outbound Rules Tags

Edit rules

| Type | Protocol | Port Range | Source | Description |
|----------------------------------|----------|------------|--------|-------------|
| This security group has no rules | | | | |

Click **Inbound Rules**, then the **Edit rules** button in that lower pane.

Add a new inbound rule for the EC2 server(s) in our web tier by clicking the **Add Rule** button. The **Type** should be **MySQL/Aurora (3306)** which auto-selects protocol **TCP** and port **3306**. In the **Source** text box, start typing "sg-", while you're typing, a list of security group(s) that match should be presented to you. Select your web tier security group, then click the **Save rules** button.

Security Groups > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

| Type | Protocol | Port Range | Source | Description |
|--------------|----------|------------|--------------|----------------------------|
| MySQL/Aurora | TCP | 3306 | Custom sg-04 | e.g. SSH for Admin Desktop |

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

* Required

Cancel Save rules

Click **Close** on the confirmation screen.

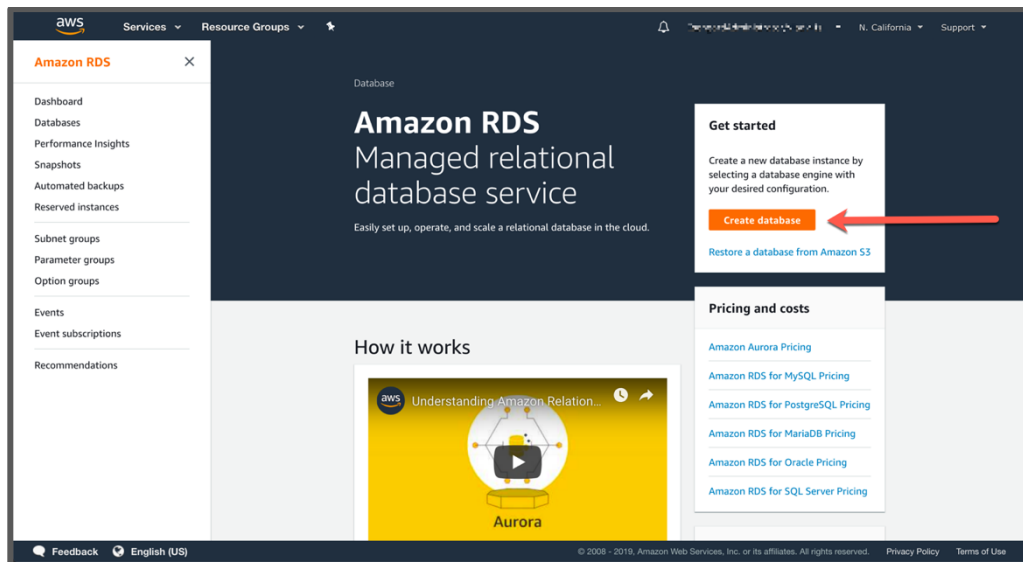
Launch an RDS DB Instance

Now that all prerequisites are met, you can configure and launch a MySQL RDS DB Instance. For the purposes of this lab, you will use the default configurations where possible.

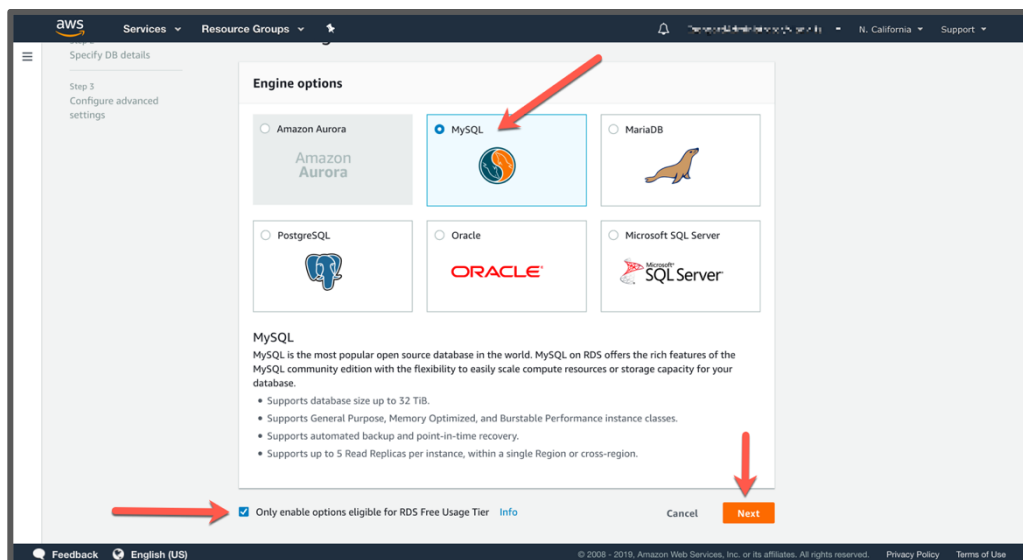
Sign into the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds>, if the console is not already open, or showing a different service console.

In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

1. Click on **Create database**.



At **Step1 Select engine**, choose **MySQL** from the available engine options. Check **Only enable options eligible for RDS Free Usage Tier**, at the bottom of the page. This option is not recommended for production databases, as it will disable options such as Multi-AZ deployments or read replicas, but it is safe for the purposes of this lab. Click **Next** in the bottom right area of the screen.



At **Step2 DB details**, fill out the DB Instance details with the following information and then click **Next** in the bottom right area of the screen.

- DB engine version: MySQL 5.7.25 (or newer 5.7 minor version)
- DB instance class: db.t2.micro
- Storage type: General Purpose (SSD)
- Allocated storage: 20 GB
- DB instance identifier: dbweek
- Master username: awsuser (or another memorable username)
- Master password: DBweek19 (or another memorable password)

The screenshot shows the 'Specify DB details' step in the AWS Management Console. The interface includes a sidebar with navigation links for 'Step 1: Select engine', 'Step 2: Specify DB details' (active), and 'Step 3: Configure advanced settings'. The main content area is titled 'Specify DB details' and contains two sections: 'Instance specifications' and 'Settings'.

Instance specifications:

- DB engine:** MySQL Community Edition
- License model:** general-public-license
- DB engine version:** MySQL 5.7.23
- Known Issues/Limitations:** Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.
- Free tier:** The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GiB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions here.
- DB instance class:** db.t2.micro — 1 vCPU, 1 GiB RAM
- Multi-AZ deployment:** Create replica in different zone (selected)
- Storage type:** General Purpose (SSD)
- Allocated storage:** 20 GiB

Settings:

- DB instance identifier:** dbweek
- Master username:** awsuser
- Master password:** [masked]
- Confirm password:** [masked]

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next' (highlighted with a red arrow).

At **Step 3 Configure advanced**, fill out the **Network & Security** section with the following information:

- Virtual Private Cloud (VPC): *Select the one named Default VPC*
- Subnet group: default
- Public accessibility: No
- Availability zone: No preference

- VPC security groups: Select **Choose existing security group**, then pick the one you have created previously for the database. Remove the one named **default** by clicking on the × symbol.

In the **Database options** section, provide a memorable database name, such as “dbweek” (you will need it later).

Step 1: Select engine

Step 2: Specify DB details

Step 3: Configure advanced settings

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) [Info](#)
VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-bca0b7de) [Info](#)

Only VPCs with a corresponding DB subnet group are listed.

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

default [Info](#)

Public accessibility [Info](#)
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instance. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

☐ Yes

☒ No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone [Info](#)
No preference [Info](#)

VPC security groups
Security groups from which authorizing connections from all the EC2 instances and devices that need to access the DB instance.

☐ Create new VPC security group

☒ Choose existing VPC security groups

Choose VPC security groups

DB Week Database [X](#)

Database options

Database name [Info](#)
dbweek

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

Port [Info](#)
TCP/IP port the DB instance will use for application connections.

3306

DB parameter group [Info](#)
default-mysql-8-0

Option group [Info](#)
default-mysql-8-0

IAM DB authentication [Info](#)
Manage your database user credentials through AWS IAM users and roles.

☐ Enable IAM DB authentication

☒ Disable

Encryption

Encryption

☒ Enable encryption [Learn more](#)

Select to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the Key Management Service (KMS) console.

☐ Disable encryption

[The selected engine or DB instance class does not support storage encryption.](#)

Backup

[Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MySQL, refer to detail here.](#)

Backup retention period [Info](#)
Select the number of days that Amazon RDS should retain automatic backups of this DB instance.

7 days

Backup window [Info](#)
☐ Select window

☒ No preference

☒ Copy tags to snapshots

Monitoring

Enhanced monitoring

☐ Enable enhanced monitoring
Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

☒ Disable enhanced monitoring

Log exports

Select the log types to publish to Amazon CloudWatch Logs

☐ Error log

☐ General log

☐ Slow query log

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS Service Linked Role

[Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default.](#)

[Learn more](#)

Maintenance

Auto minor version upgrade [Info](#)
☒ Enable auto minor version upgrade
Enables automatic upgrades to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.

☐ Disable auto minor version upgrade

Maintenance window [Info](#)
Select the period in which you want pending modifications or patches applied to the DB instance by Amazon RDS.

☐ Select 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.

Cancel Previous **Create database**

Use the default settings for the Encryption, Backup, Monitoring, Log exports and Maintenance sections.

Under **Deletion protection**, uncheck the box labeled **Enable deletion protection**. This is usually not recommended in a production deployment, but will simplify your lab deployment.

Review your settings and click **Create database** in the bottom right area of the screen.

On the confirmation screen, click **View DB instance details** to monitor the provisioning process.



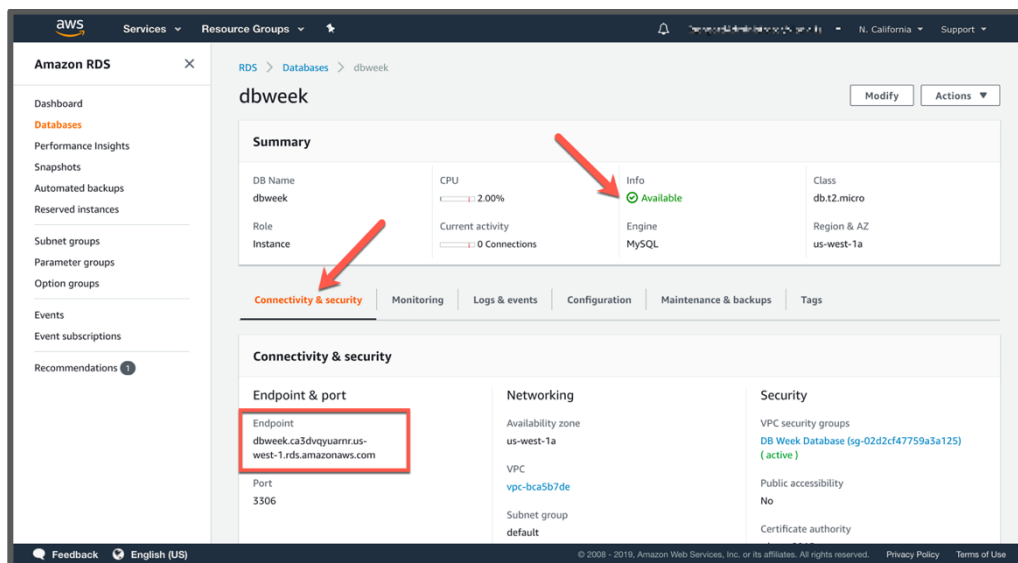
This may take up to 5 minutes as the database is being created, configured, then backed up for the first time.

Connect the RDS DB Instance to the Web Server

The web server previously created contains a script to deploy an example database table and sample data for creating a simple address book. Next, you will connect the web server to the RDS DB instance and deploy that example.

While still on the DB instance detail page of the database we recently created, refresh the browser window to ensure the DB instance status is listed as **Available**.

Click on the **Connectivity & security** tab if it is not already selected, and check the value under **Endpoint**. This is the DNS name of your database, and you will need it to connect to the database from clients. Copy that value, as you will need it shortly.



Navigate to the browser tab or window where you have the web server interface previously created open. Click on the **RDS** link.

You should see a prompt to enter the DB endpoint previously copied (do NOT include :3306 at the end of the DB endpoint), as well as the database “dbweek”, username “awsuser” and password “DBweek19”, or the custom values you specified when creating the database. Click the **Submit** button.

amazon web services

LOAD TEST RDS

Endpoint: dbweek.ca3dvquarnr.us-west-1.rds.amazonaws.com

Database: dbweek

Username: awsuser

Password:

Submit

When complete, you will be redirected to a simple page displaying all of the information from the database you just created.

amazon web services

LOAD TEST RDS

Address Book

| Name | Phone | Email | Admin |
|-------|--------------|---------------------|-------------|
| Alice | 571-555-4875 | alice@address2.us | Edit Remove |
| Bob | 630-555-1254 | bob@fakeaddress.com | Edit Remove |

Add Contact

This is a very basic example of a simple address book interacting with a MySQL database managed by AWS. RDS can support much more complicated relational database scenarios, but we hope this simple example will suffice to demonstrate the point.

Feel free to play around with the address book and add/edit/remove content from your RDS database by using the **Add Contact**, **Edit**, and **Remove** links in the Address Book.

Working with RDS DB Instances

Backup and Restore using RDS Snapshots

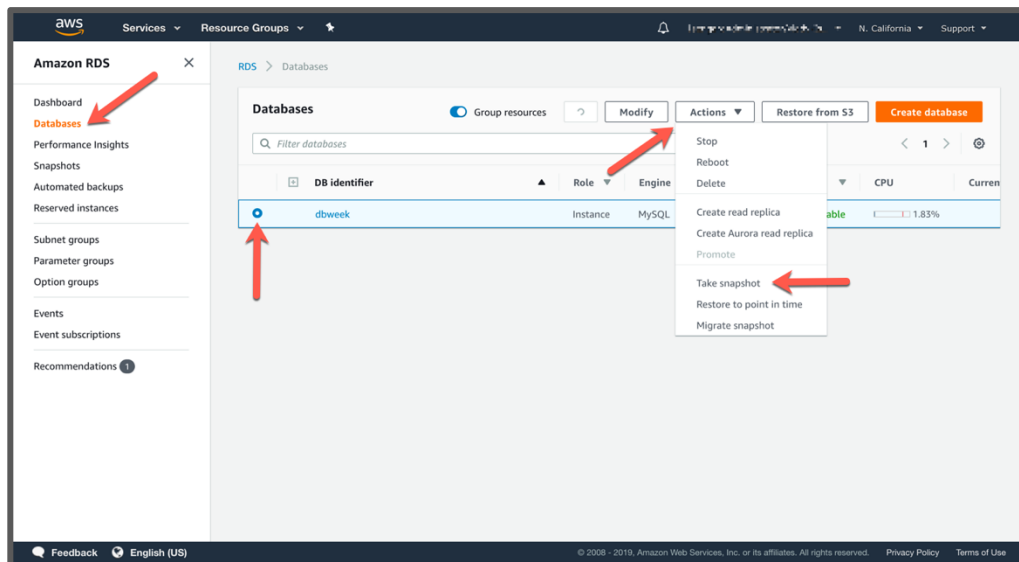
Now is a good time to take a snapshot of your RDS database. Taking a snapshot enables you to back up your DB Instance in a known state as frequently as you wish, and then restore to that specific state at any time.

Sign into the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds>, if the console is not already open, or showing a different service console.

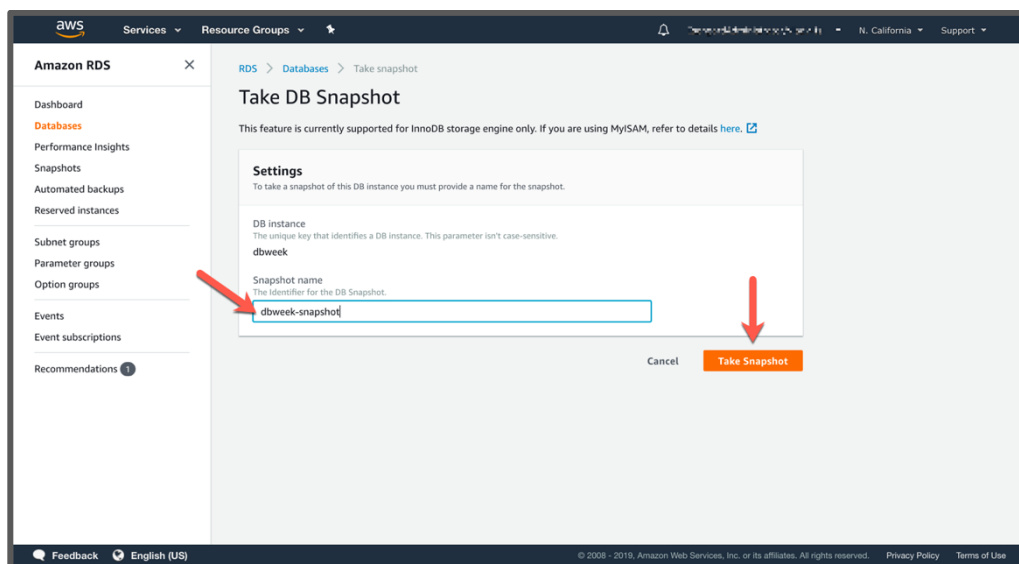
In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

In the left-hand side navigation panel (the panel may be collapsed, click the three horizontal bars at the top of it to expand it), click on **Databases**, then select the radio box of the row corresponding to your DB instance.

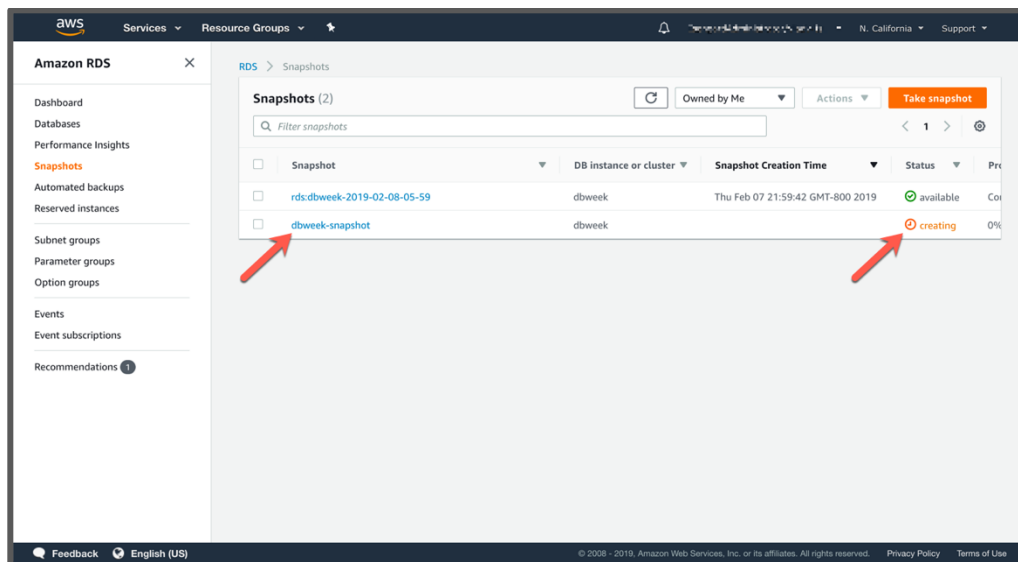
From the **Actions** menu button at the top of the DB instance listing, choose **Take snapshot**.



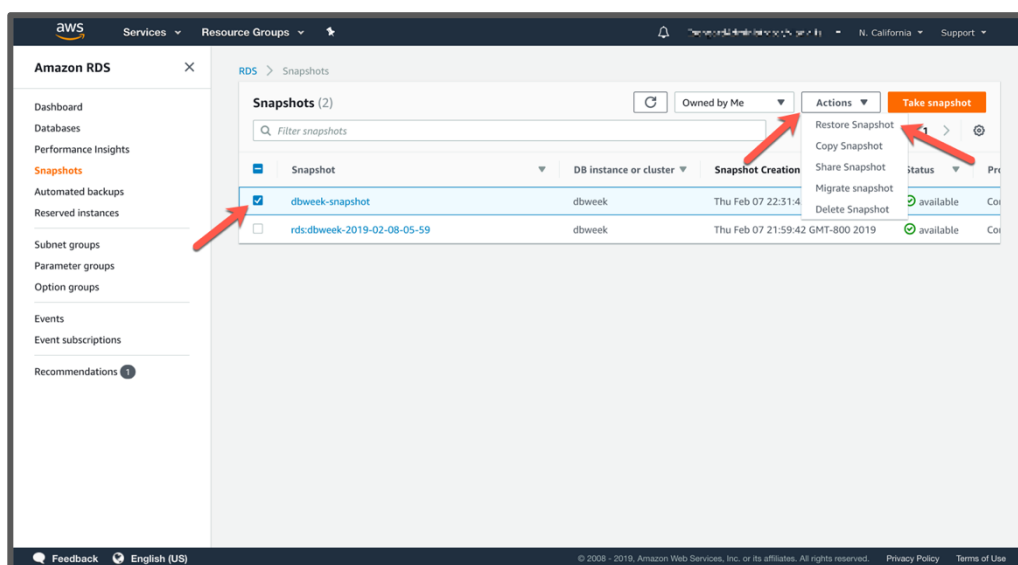
Give the snapshot a memorable name, such as “dbweek-snapshot” and click **Take Snapshot**.



You can track the backup progress in the **Snapshots** section of the RDS console. When the snapshot completes it will be listed as **available**. Notice the automated snapshot in the listing. It was enabled automatically when the default of 7 days backup retention was configured at DB instance launch time. You can leverage the automated backups to execute point in time restore operations, up to the last 5 minutes prior to the present time.



Once available, you can use that snapshot to restore the database. RDS does not replace your existing DB instance with the restored one, instead it will create a new DB instance using the data set of the snapshot. To initiate a restore operation, select the desired snapshot, then from the **Action** menu button at the top of the snapshot listing. Choose **Restore Snapshot** then follow the configuration screens. Notice that you can easily launch new RDS instances from any previous snapshot!



Launching a new DB instance from a snapshot may incur additional costs beyond the Free Tier.

Scale up the Compute Capacity for an RDS DB Instance

Scaling up and down with RDS is simple via the AWS Management Console. You can change the underlying server size, by selecting a new DB instance class, and you can modify the storage characteristics, by changing the storage type and growing the size of the storage to accommodate usage growth. You cannot

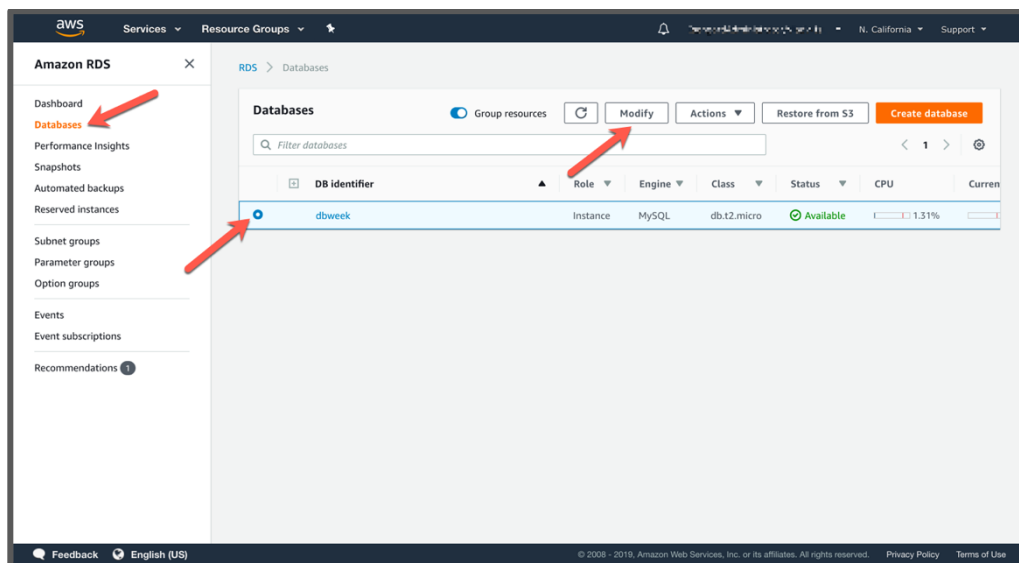
shrink the size of the storage, however, if it turns out you have overprovisioned it. It is always recommended to start with the right amount of storage you need in the immediate future, and scale the storage as your workload grows.

Sign into the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds>, if the console is not already open, or showing a different service console.

In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

In the left-hand side navigation panel (the panel may be collapsed, click the three horizontal bars at the top of it to expand it), click on **Databases**, then select the radio box of the row corresponding to your DB instance.

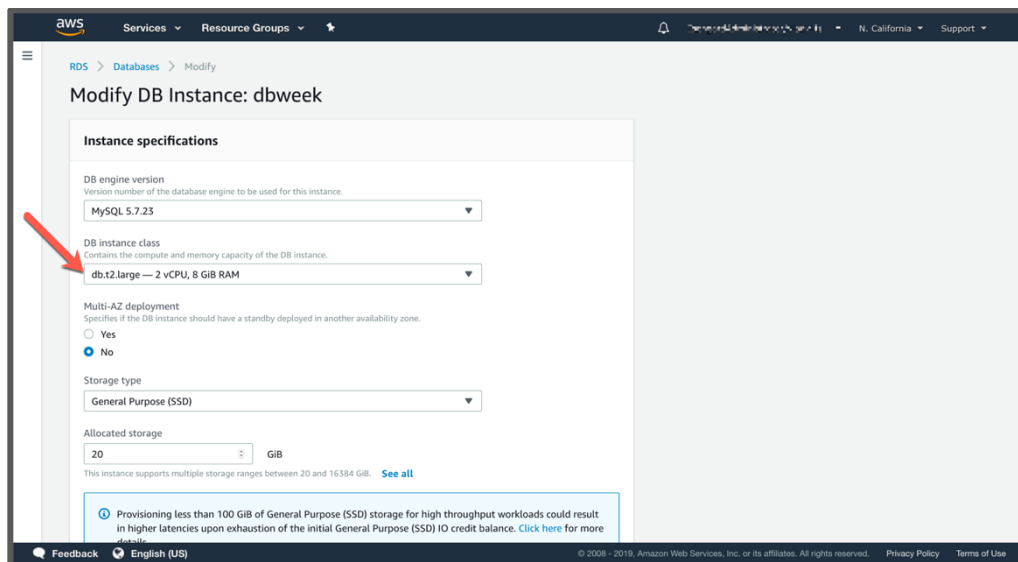
Click the **Modify** button at the top of the DB instance listing.



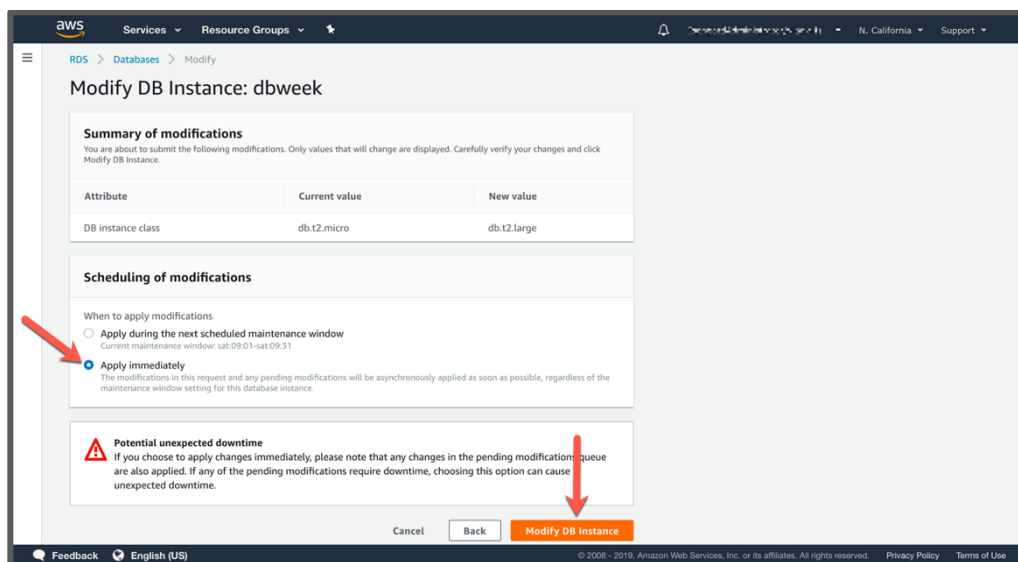
On the **Modify DB Instance** screen, try changing to a **t2.large** DB instance class, and if you want, also grow the database at the same time. Click **Next**.



Changing the DB instance class to a larger one, or allocating more storage may incur additional costs beyond the Free Tier.

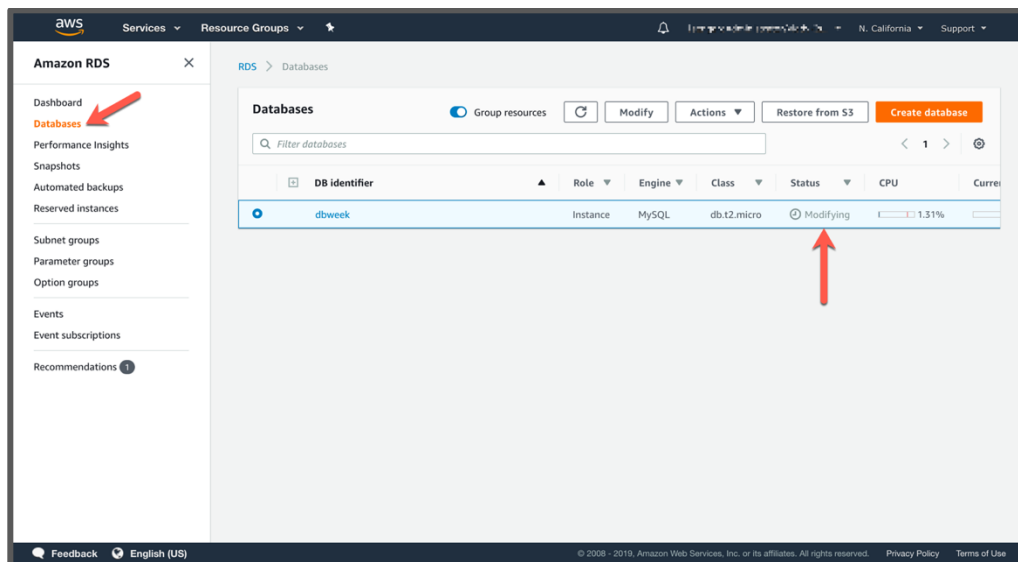


On the next screen, don't forget to choose **Apply immediately**. Then click **Modify DB Instance**. Otherwise changes will be scheduled to be applied during the next maintenance window. Scheduling changes is preferred for production workloads, as some modifications, like changing the DB instance class, require a restart (or failover in case of Multi-AZ DB instances). Applying such changes during a period of low utilization mitigates the corresponding temporary loss of availability.



This may take up to 5 minutes as the database is being reconfigured.

Back on the **Databases** listing page in the console, you can track the status of the operation. While the changes are performed, your DB instance will be listed in a **Modifying** state.



Monitoring RDS DB Instances

You can monitor the health and performance of your RDS DB instances directly from the console. Amazon RDS leverages Amazon CloudWatch to expose various relevant health and performance metrics. You can use these metrics to monitor, create alarms, troubleshoot or make capacity planning decisions.

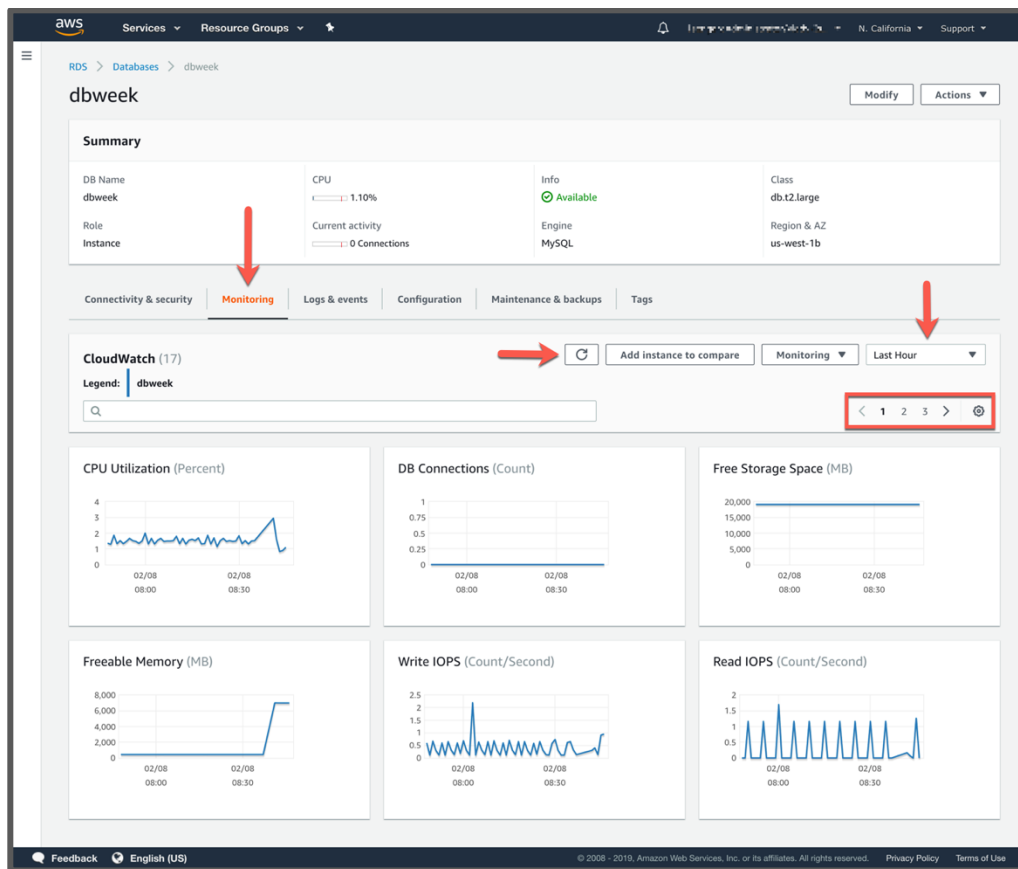
Sign into the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds>, if the console is not already open, or showing a different service console.

In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

In the left-hand side navigation panel (the panel may be collapsed, click the three horizontal bars at the top of it to expand it), click on **Databases**, then select the radio box of the row corresponding to your DB instance.

Click on the **DB identifier** of the desired database in the list.

Click on the **Monitoring** tab on the database detail page.



Use the **Refresh** button (circle with arrow) to repopulate the graphs with new data. You can also change the time period of the graphs from **Last Hour** to other relevant ones.

Monitoring metrics are paginated due to the large number of options, use the pagination controls to cycle through them, or change the number of graphs displayed on one page.

Extra Credit

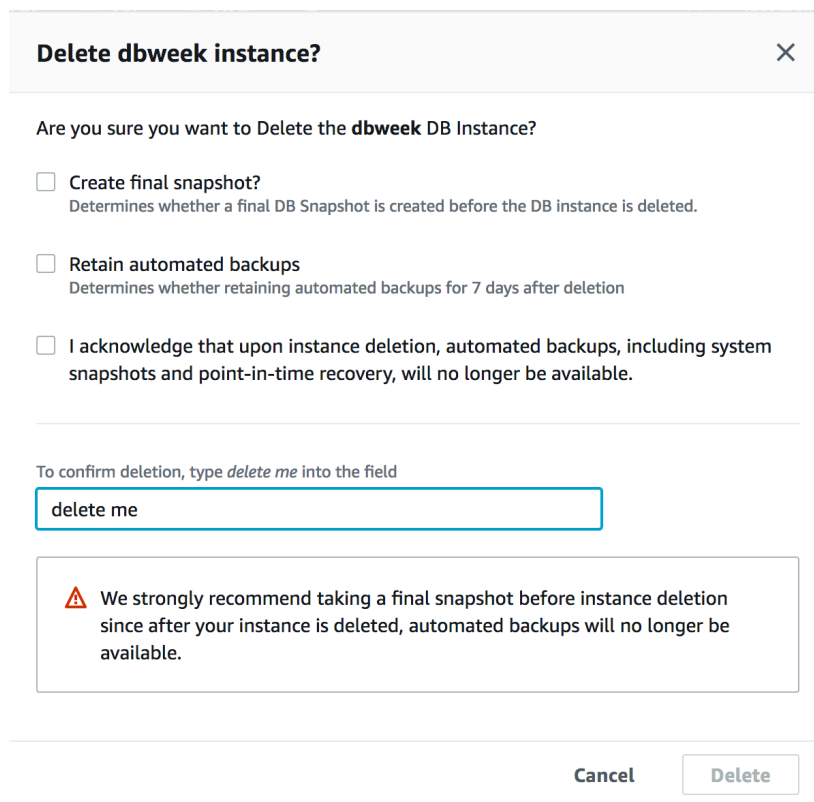
If you have additional time, try to do the following:

- Modify your original database instance size to **db.m5.large** (\$0.171/hr; scaling operation to take about 10 minutes and will result in downtime), enable **Performance Insights**. Once turned on, return to the web app interface to insert, update, or delete records. Wait a few minutes and take a peek into RDS Performance Insights for metrics on database level performance. Click on the settings (gear icon) and add additional metrics like `Select_scan`, `Innodb_rows_read`, `Innodb_rows_updated`, and `Innodb_rows_deleted`. What do you see within the Performance Insights metrics?

Cleaning Up

Once finished, cleanup your resources created during this workshop to avoid incurring any running charges. Remember all running EC2 and RDS resources will incur an hourly on-demand charge.

1. Delete all RDS database instances created during the workshop. In the RDS Console, for each of the databases created, click on **Actions**, then **Delete**. Uncheck “**Create final snapshot**” and “**Retain automated backups**” (Normally for an actual database you might want to create the final snapshot in the event you need to restore the database). Click on “**I acknowledge**” checkbox and type in “**delete me**” in the field. Then press the **Delete** button.



The screenshot shows the 'Delete dbweek instance?' dialog box in the AWS RDS console. The dialog has a title bar with a close button (X). The main content area asks 'Are you sure you want to Delete the **dbweek** DB Instance?'. Below this are three checkboxes: 'Create final snapshot?' (unchecked), 'Retain automated backups' (unchecked), and 'I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.' (unchecked). Below the checkboxes is a text input field with the placeholder 'To confirm deletion, type *delete me* into the field'. The input field contains the text 'delete me'. Below the input field is a warning box with a yellow triangle icon and the text: 'We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.' At the bottom of the dialog are two buttons: 'Cancel' and 'Delete'.

2. Delete the EC2 instance created during the workshop. In the EC2 Console main page, click on **Running Instances**; or click on **Instances** on the left menu. Click on the checkbox next to the EC2 instance created for the web app. Then click on **Actions**, then **Instance State**, then **Terminate**.

Confirm on **Yes, Terminate** when prompted.

3. Delete the EC2 keypair. Click on **Key Pairs** on the left menu under the EC2 Console. Click on the checkbox next to the keypair created. Then **Delete**.
4. Delete the VPC Security Groups. Click on **Security Group** on the left menu under the EC2 Console. Highlight all the Security Groups created during the workshop (they should start with *dbweek** if you followed the naming conventions in the instructions) and then click on **Delete Security Group**.