Database Week

Hands-On Lab: Getting Started with Amazon RDS

October 2019

Table of Contents

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

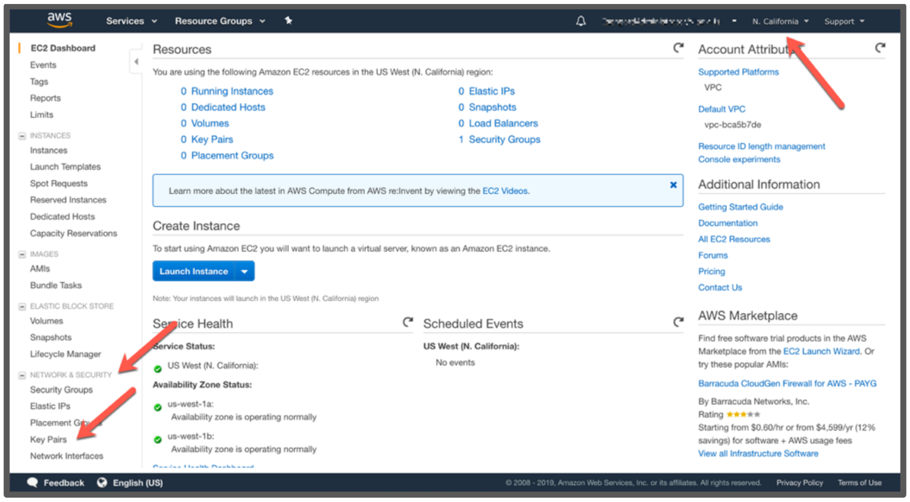
> If you have previously completed the **Getting Started with EC2 Linux** lab and still have the key pair available, you may re-use it.

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.



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

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

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 (N. California) region

Service Status:
US West (N. California):
Availability Zone Status:
us-west-1a:
Availability zone is operating normally
us-west-1b:
Availability zone is operating normally

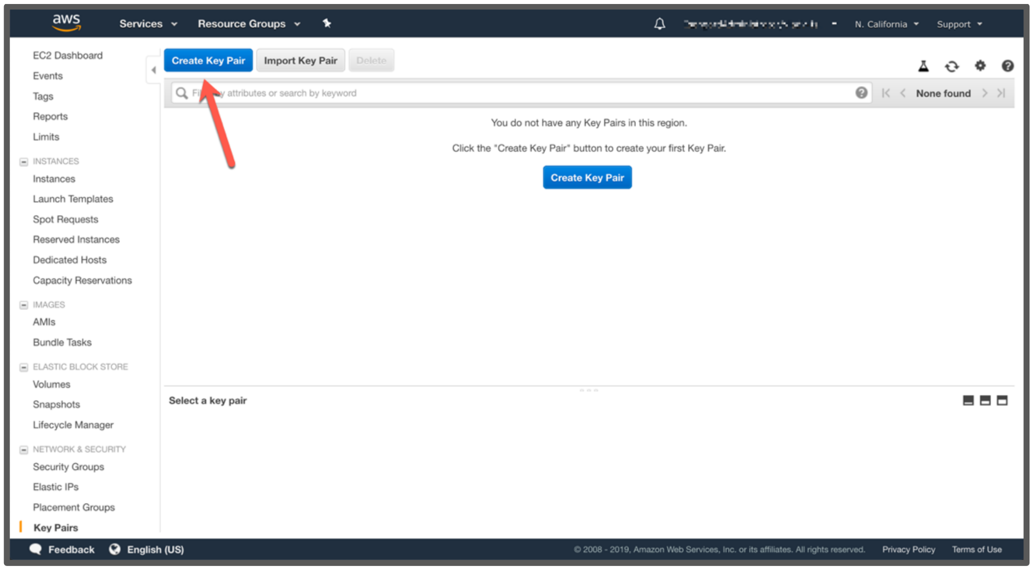
US West (N. California):
No events

Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:

Barracuda CloudGen Firewall for AWS - PAYG
By Barracuda Networks, Inc.
Rating ★★★★★
Starting from $0.60/hr or from $4,599/yr (12% savings) for software + AWS usage fees
View all Infrastructure Software

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click Create Key Pair.

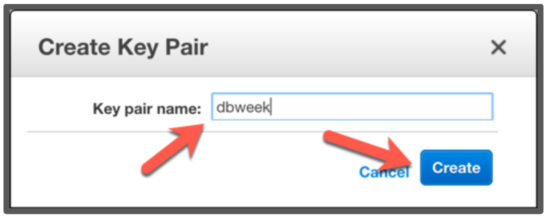


You do not have any Key Pairs in this region.

Click the "Create Key Pair" button to create your first Key Pair.

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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.



Key pair name: dbweek

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.

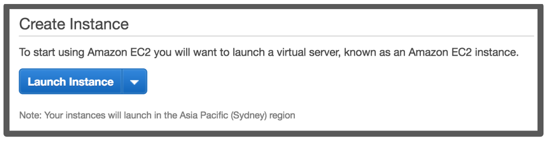
In this section we will launch an Amazon Linux 2 EC2 instance, bootstrap Apache/PHP, and install a basic web application.

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**.

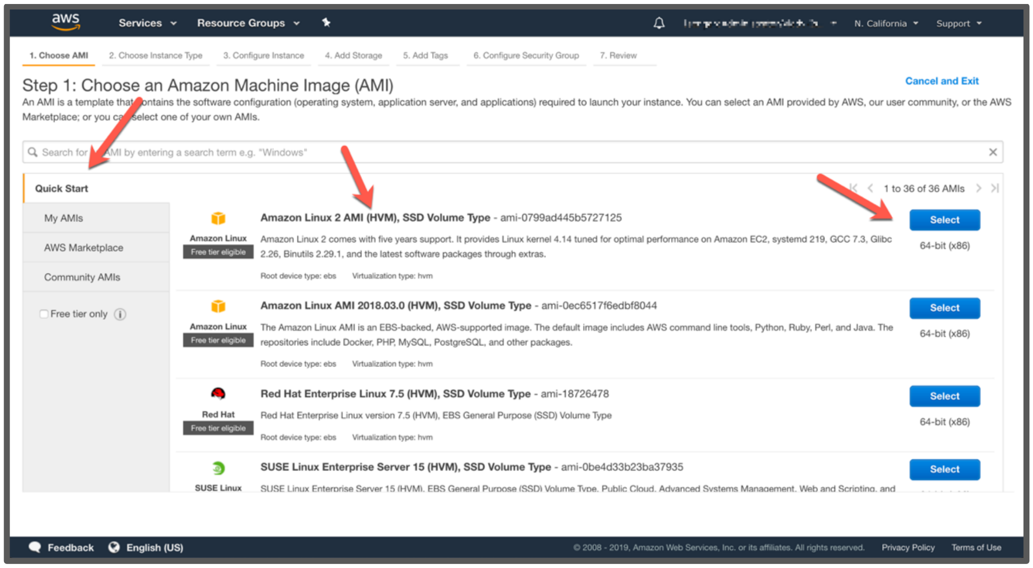


Create Instance

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 Asia Pacific (Sydney) region

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.



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 2 AMI (HVM), SSD Volume Type - ami-0799ad445b5727125

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm

64-bit (x86)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0ec6517f6edbf8044

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

64-bit (x86)

Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-18726478

Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

64-bit (x86)

SUSE Linux Enterprise Server 15 (HVM), SSD Volume Type - ami-0be4d33b23ba37935

SUSE Linux Enterprise Server 15 (HVM), EBS General Purpose (SSD) Volume Type, Public Cloud, Advanced Systems Management, Web and Scripting, and

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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!

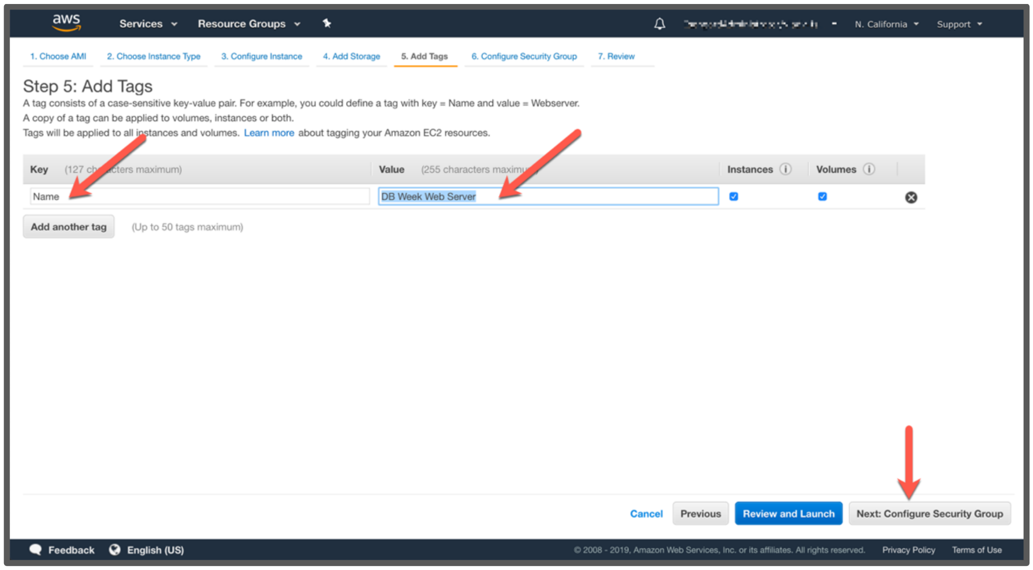
On the **Step 3: Configure Instance Details** screen, accept the defaults, but scroll down and expand the **Advanced Details** section. Copy/paste the script below into the **User Data** field (this shell script will install Apache & PHP, start the web service, and deploy a simple web page). Click **Next: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



Step 5: Add Tags

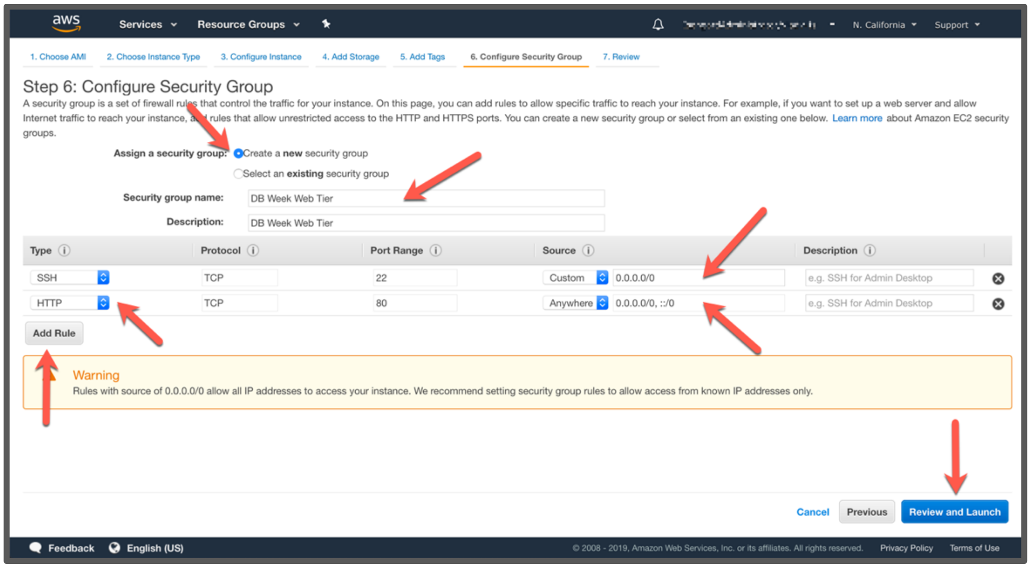
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	DB Week Web Server	☑	☑

(Up to 50 tags maximum)

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, 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: DB Week Web Tier

Description: DB Week Web Tier

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, ::/0	e.g. SSH for Admin Desktop

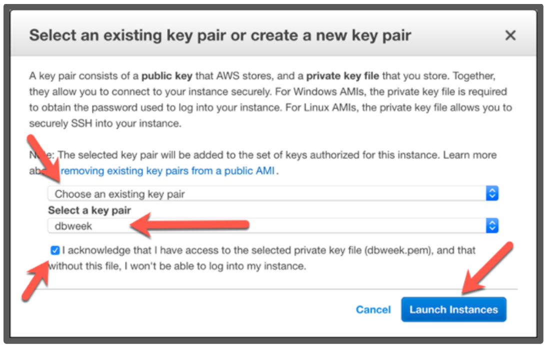
Add Rule

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.

Cancel
Previous
Review and Launch
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Review your cofiguration 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.

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.

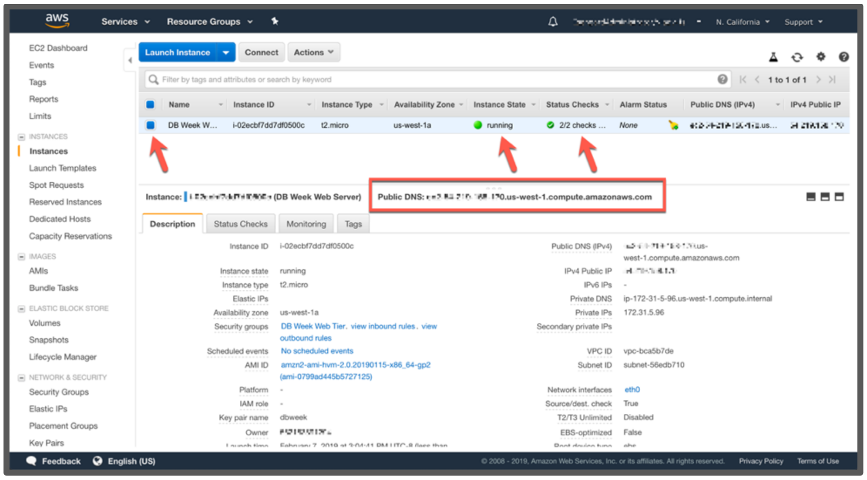
Choose an existing key pair

Select a key pair: dbweek

☑ 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.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
DB Week W...	i-02ecbf7dd7df0500c	t2.micro	us-west-1a	running	2/2 checks ...	None	[redacted]	[redacted]

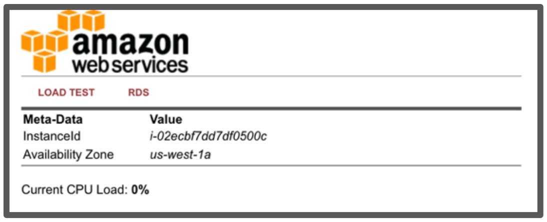
Instance: [redacted] (DB Week Web Server)

Public DNS: [redacted]0.us-west-1.compute.amazonaws.com

Instance ID	i-02ecbf7dd7df0500c	Public DNS (IPv4)	[redacted]us-west-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	[redacted]
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-5-96.us-west-1.compute.internal
Availability zone	us-west-1a	Private IPs	172.31.5.96
Security groups	DB Week Web Tier. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-bca5b7de
AMI ID	amzn2-ami-hvm-2.0.20190115-x86_64-gp2 (ami-0799ad445b5727125)	Subnet ID	subnet-56edb710
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	dbweek	T2/T3 Unlimited	Disabled
Owner	[redacted]	EBS-optimized	False
Launch time	February 7, 2019 at 3:04:41 PM UTC-8 (less than	Root device type	ebs

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:



Meta-Data	Value
InstanceId	*i-02ecbf7dd7df0500c*
Availability Zone	*us-west-1a*

Current CPU Load: **0%**

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.

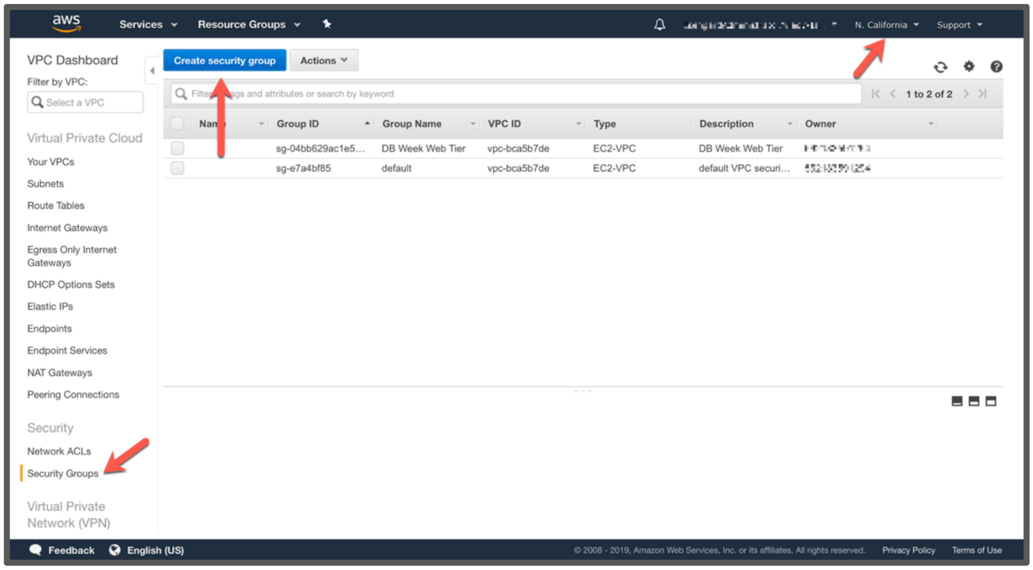
To this end, we'll need to utilize a VPC security group to permit the EC2 instance to access the RDS DB instance. Because workloads in AWS are elastic, and the number of actual EC2 instances typically can change, we will reference the security group of the web server as a permitted source of traffic 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.

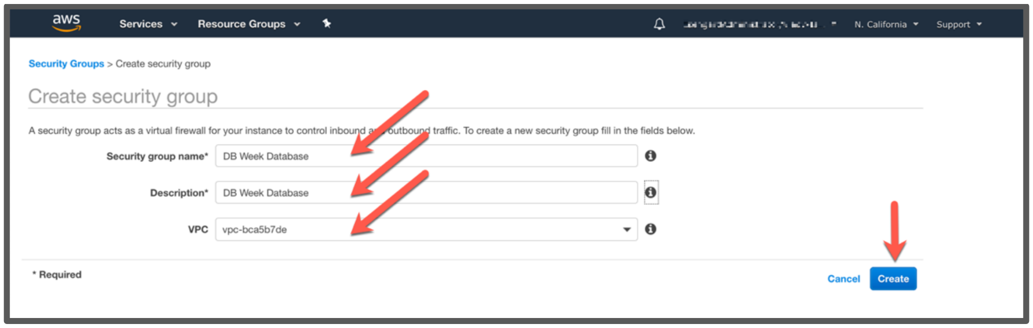
In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

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



Name	Group ID	Group Name	VPC ID	Type	Description	Owner
	sg-04bb629ac1e5...	DB Week Web Tier	vpc-bca5b7de	EC2-VPC	DB Week Web Tier	[redacted]
	sg-e7a4bf85	default	vpc-bca5b7de	EC2-VPC	default VPC securi...	[redacted]

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**.



[redacted]

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.

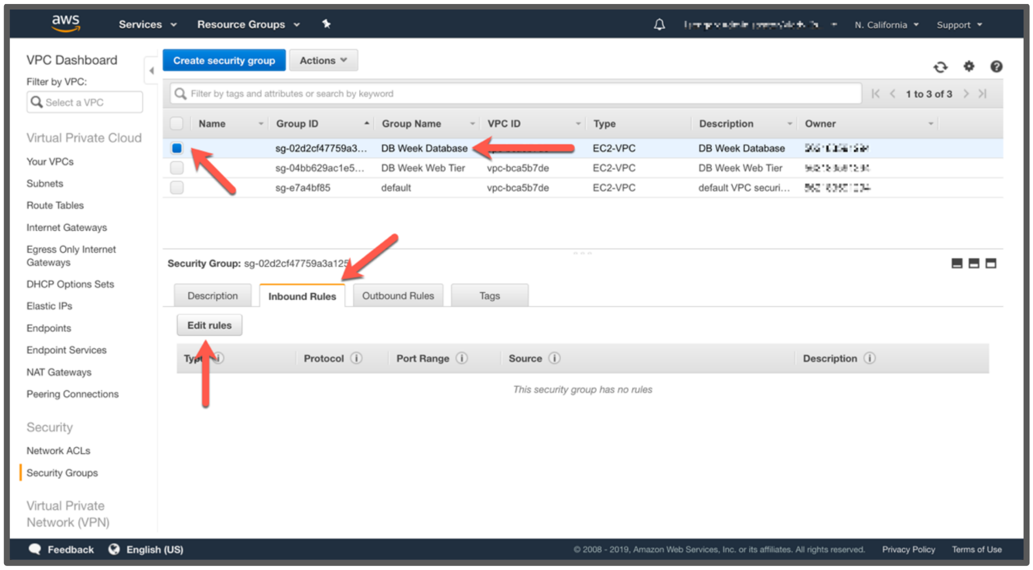
Security group name* DB Week Database

Description* DB Week Database

VPC vpc-bca5b7de

* Required

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.



[redacted]

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
	sg-02d2cf47759a3...	DB Week Database		EC2-VPC	DB Week Database	[redacted]
	sg-04bb629ac1e5...	DB Week Web Tier	vpc-bca5b7de	EC2-VPC	DB Week Web Tier	[redacted]
	sg-e7a4bf85	default	vpc-bca5b7de	EC2-VPC	default VPC securi...	[redacted]

Security Group: sg-02d2cf47759a3a125

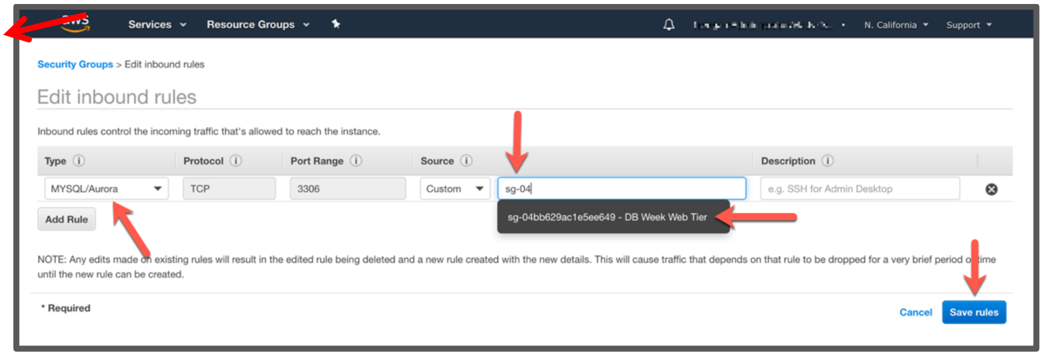
Type	Protocol	Port Range	Source	Description

This security group has no rules

Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

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.



[redacted]

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

sg-04bb629ac1e5ee649 - DB Week Web Tier

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

Click **Close** on the confirmation screen.

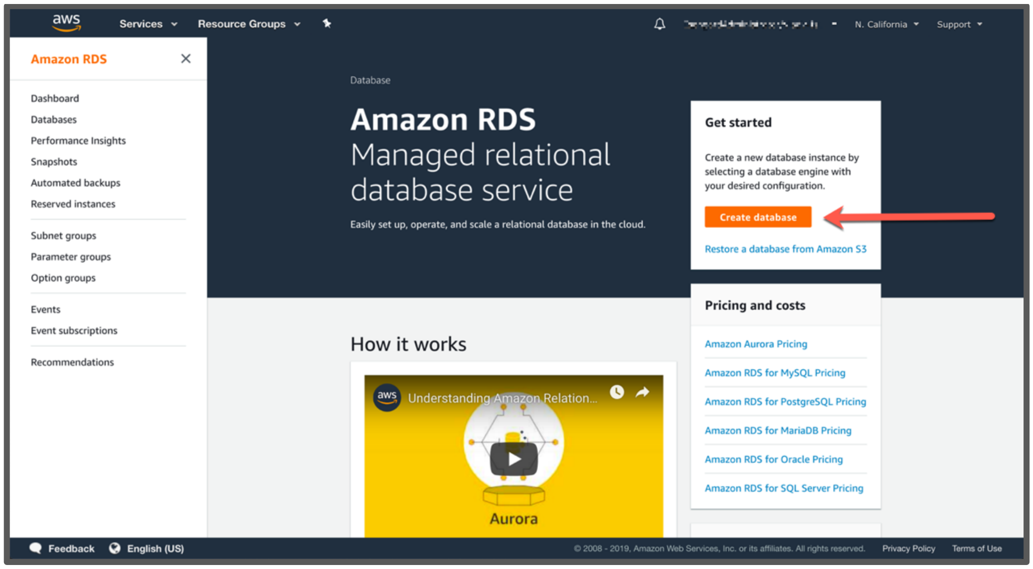
Launch an RDS DB Instance

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.

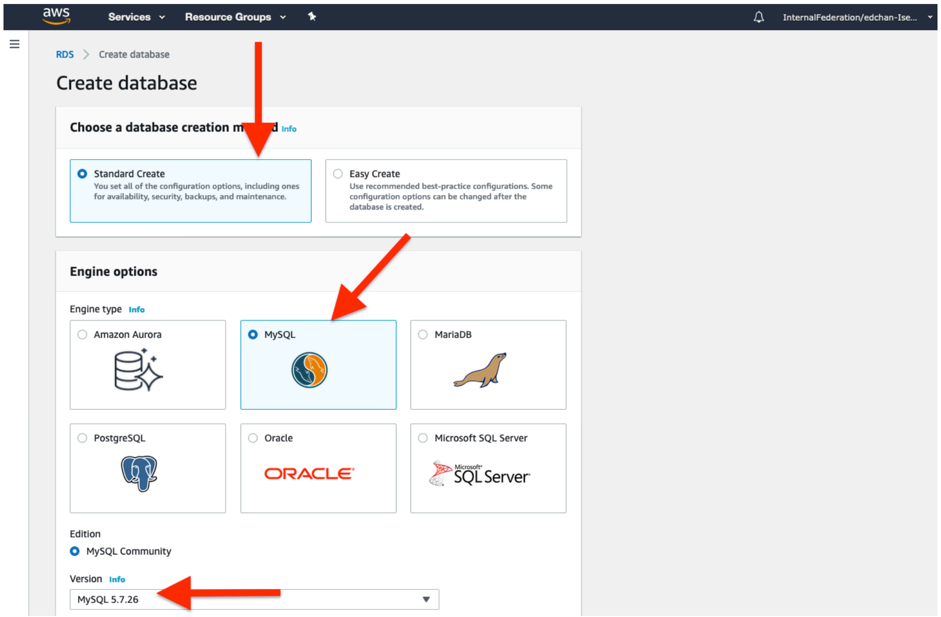
Sign into the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds, if the console is not already open, or showing a different service console.

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**.


aws
Services
Resource Groups
N. California
Support
Amazon RDS
Dashboard
Databases
Performance Insights
Snapshots
Automated backups
Reserved instances
Subnet groups
Parameter groups
Option groups
Events
Event subscriptions
Recommendations
Database
Amazon RDS
Managed relational database service
Easily set up, operate, and scale a relational database in the cloud.
Get started
Create a new database instance by selecting a database engine with your desired configuration.
Create database
Restore a database from Amazon S3
Pricing and costs
Amazon Aurora Pricing
Amazon RDS for MySQL Pricing
Amazon RDS for PostgreSQL Pricing
Amazon RDS for MariaDB Pricing
Amazon RDS for Oracle Pricing
Amazon RDS for SQL Server Pricing
How it works
Understanding Amazon Relation...
Aurora
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

At **Create Database** page**,** choose Standard Create to see most options for the purpose of this workshop. Under Engine options, select **MySQL** from the available engine types. Select MySQL 5.7.26 for Engine Version.


aws
Services
Resource Groups
InternalFederation/edchan-Ise...
RDS
Create database
Create database
Choose a database creation method Info
Easy Create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Amazon Aurora
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
SQL Server
Version Info

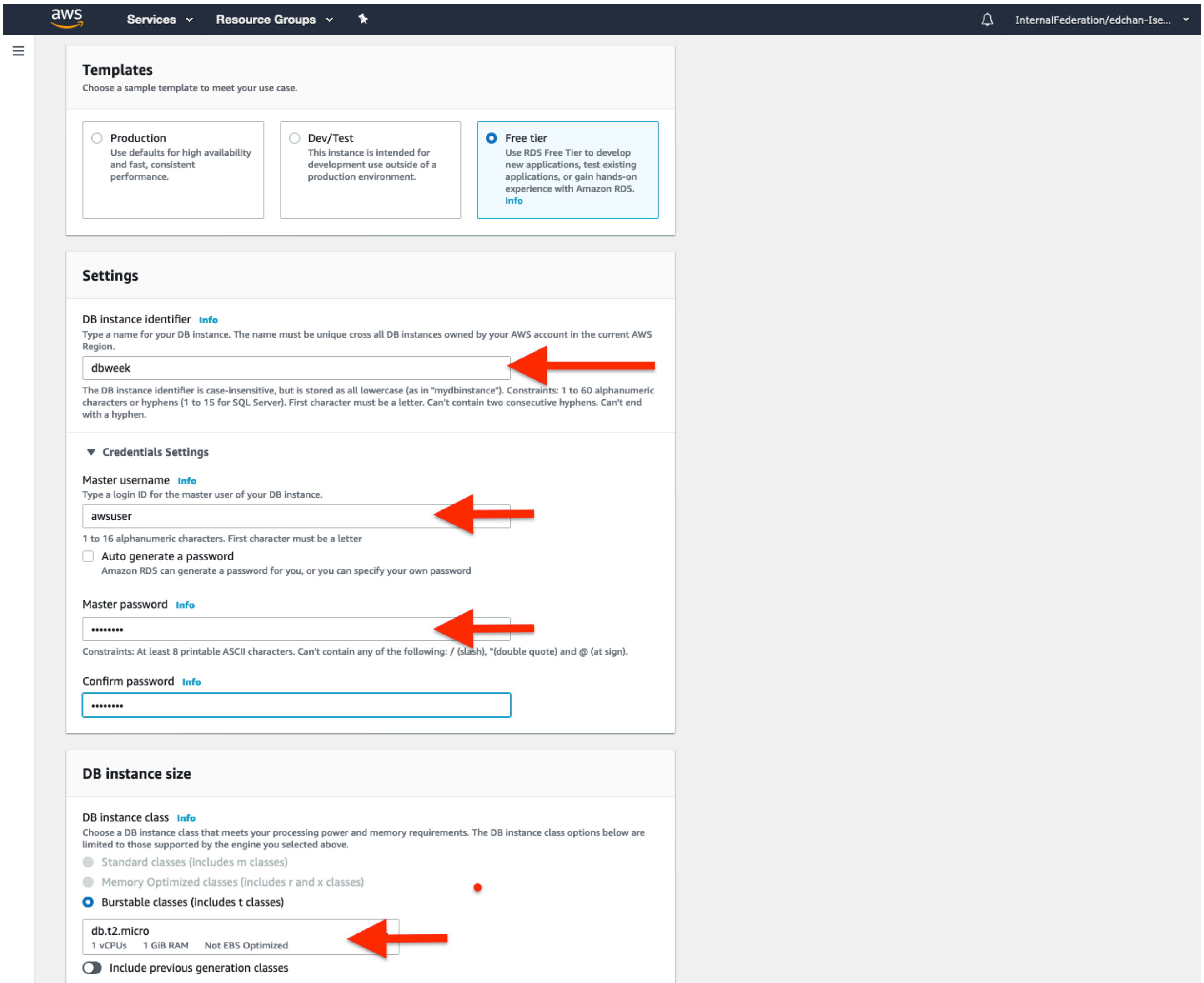
● Standard Create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

● MySQL

Edition
● MySQL Community

Version
MySQL 5.7.26

Continue to scroll down, Under Templates, select **Free Tier** (normally you would select Production or Dev/Test to prefill some additional options). Free Tier will disable options such as Multi-AZ deployments or read replicas, but it is safe for the purposes of this lab. The Free Tier will also bound your instance to ***db.t2.micro*** for DB instance size.



Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

Settings

DB instance identifier Info
Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

dbweek

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.

awsuser

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), "(double quote) and @ (at sign).

Confirm password Info

••••••••

DB instance size

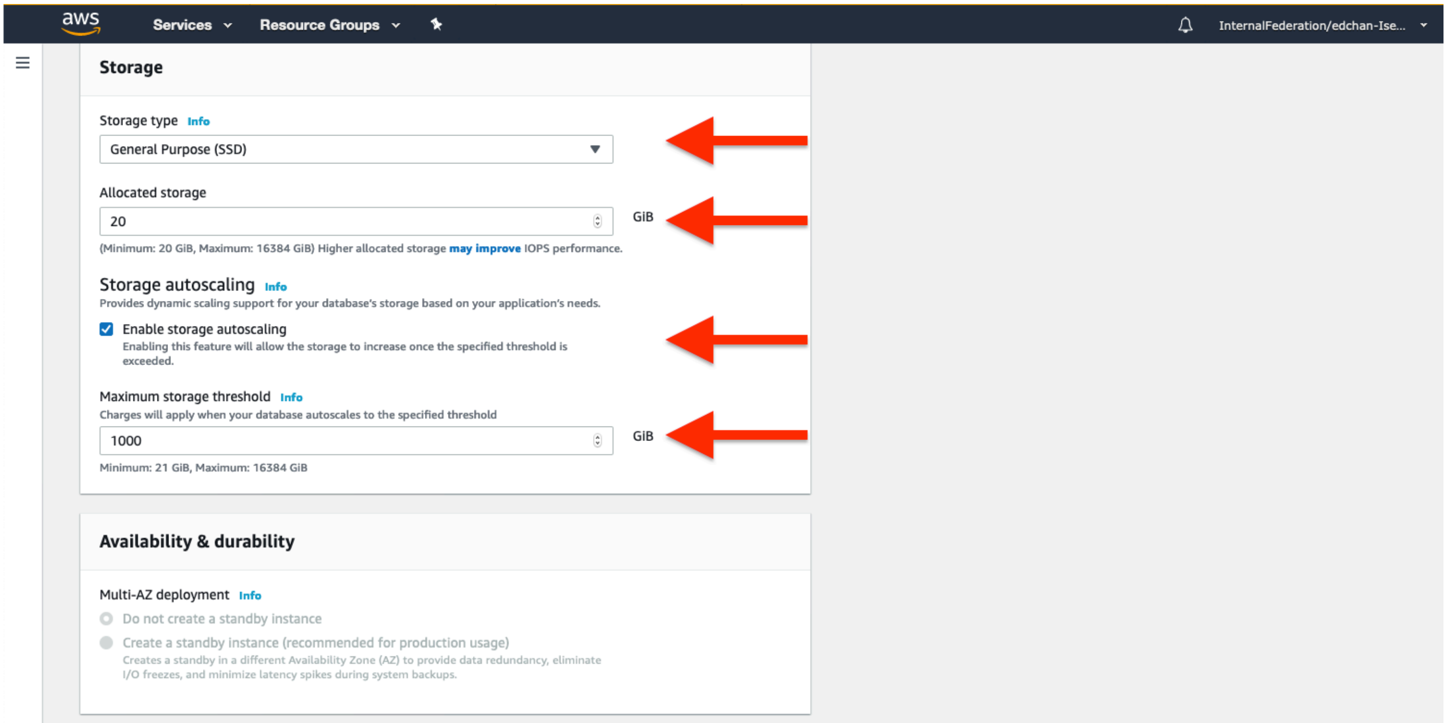
DB instance class Info
Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

○ Standard classes (includes m classes)
○ Memory Optimized classes (includes r and x classes)
● Burstable classes (includes t classes)

db.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized

Include previous generation classes

Set Storage to **General Purpose (SSD)** for Storage Type, and leave the defaults of **20GiB** Allocation Storage and Maximum threshold of **1000GiB**, and check **Enable storage autoscaling** on. As Free Tier is being used, this database instance will be forced to disable Multi-AZ deployment (**Do not create a standby instance**) under Availability & durability. Under Connectivity, select the **Default VPC**. Then click on **Additional connectivity configuration** to expand more options. Ensure that **Publicly accessible** is set to **No**, in order to disable a public IP from being assigned and remove public connectivity. Under **VPC security group**, we will select the previously created security group named ***DBweek Database*** and remove the ***default*** security group with the X button. Set Availability zone to No Preference, and Database port as default value of 3306, and utilize Password authentication for database authentication.



Storage

Storage type Info

General Purpose (SSD)

Allocated storage

20 GiB

(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage **may 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 once the specified threshold is exceeded.

Maximum storage threshold Info

Charges will apply when your database autoscales to the specified threshold

1000 GiB

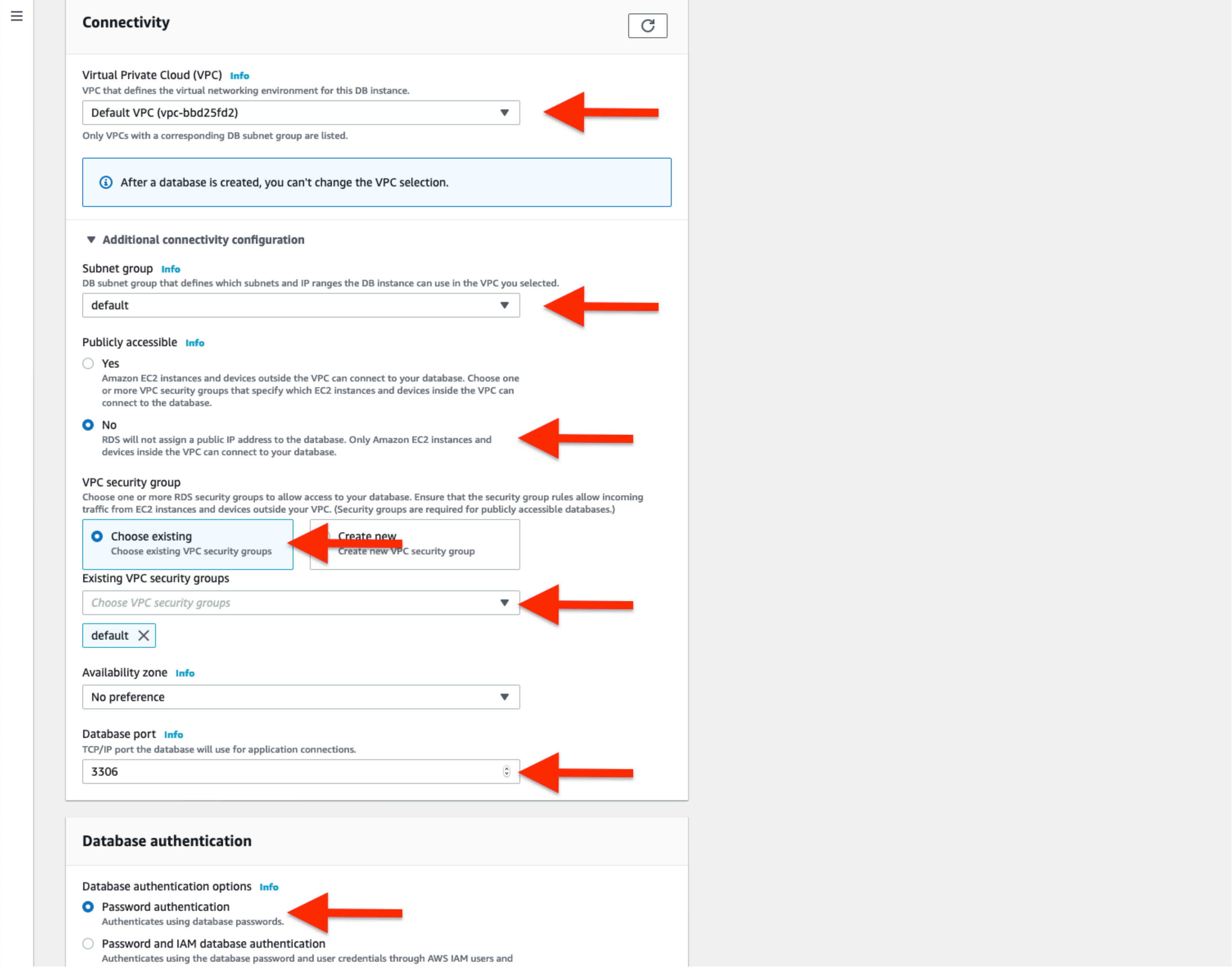
Minimum: 21 GiB, Maximum: 16384 GiB

Availability & durability

Multi-AZ deployment Info

○ Do not create a standby instance

● Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.



Connectivity

Virtual Private Cloud (VPC) Info

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-bbd25fd2)

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group Info

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default

Publicly accessible Info

○ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

● No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups

default ✕

Availability zone Info

No preference

Database port Info

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

3306

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

Your input values should resemble these values:

Database creation method:	Standard Create
Engine type/edition:	MySQL / MySQL Community
DB engine version:	MySQL 5.7.26 (or newer 5.7 minor version)
Templates:	Free Tier
DB instance identifier:	`dbweek`
Master username:	`awsuser` (or another memorable username)
Master password / Confirm:	`DBweek19` (or another memorable password)
DB instance class:	`db.t2.micro` (Automatically Selected)
Storage type:	`General Purpose (SSD)`
Allocated storage:	`20 GiB`
Storage Autoscaling:	`On`
Maximum storage threshold:	`1000 GiB`
Multi-AZ:	Do not create a standby instance
Virtual Private Cloud (VPC):	*Select the one named* **`Default VPC`**
Subnet group:	`default`
Public accessibility:	No
Availability zone:	`No preference`
Database port:	`3306`
VPC security groups:	*Select the one named* **`DB Week Database`**; *remove* **`Default`**
Database authentication:	Password authentication

Use the default settings for Additional configuration such as the Encryption, Backup, Backtrack, Enhanced Monitoring, Logs, Maintenance and Delete protection sections.

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

This will bring you back to the main RDS console Databases page and you can monitor the creation process of your RDS database instance.

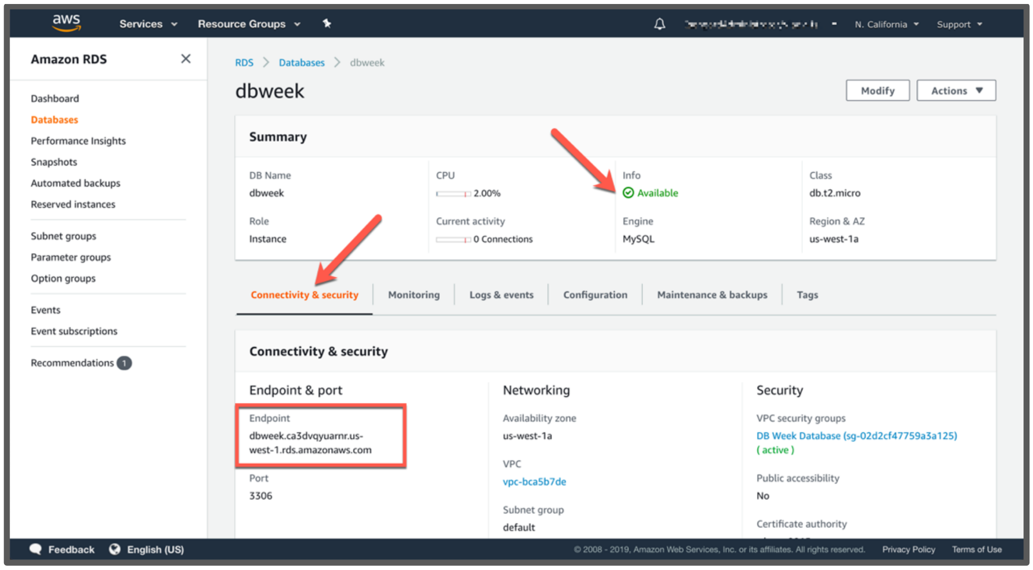
This may take up to 5 minutes as the database is being created, configured, then backed up for the first time.

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.



dbweek

Summary

DB Name: dbweek
CPU: 2.00%
Info: Available
Class: db.t2.micro
Role: Instance
Current activity: 0 Connections
Engine: MySQL
Region & AZ: us-west-1a

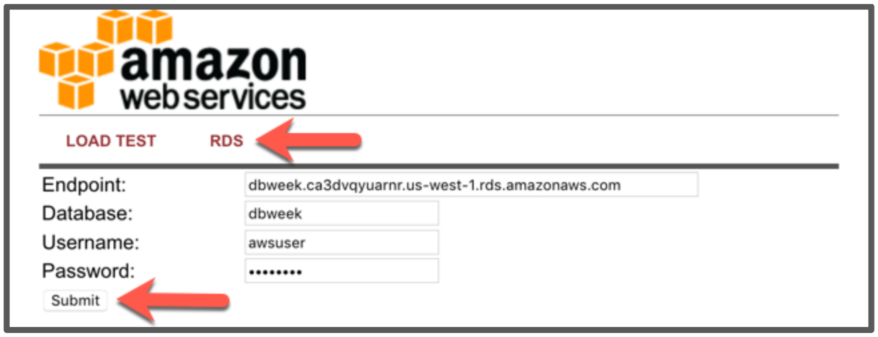
Connectivity & security

Endpoint & port
Endpoint: dbweek.ca3dvqyuarnr.us-west-1.rds.amazonaws.com
Port: 3306

Networking
Availability zone: us-west-1a
VPC: vpc-bca5b7de
Subnet group: default

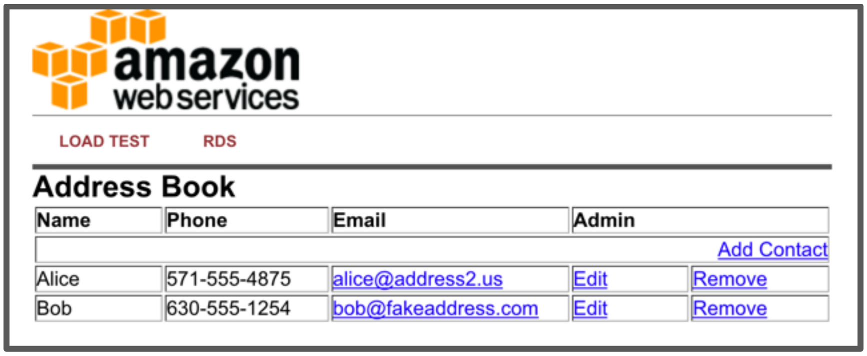
Security
VPC security groups: DB Week Database (sg-02d2cf47759a3a125) (active)
Public accessibility: No
Certificate authority

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.



Endpoint: dbweek.ca3dvqyuarnr.us-west-1.rds.amazonaws.com
Database: dbweek
Username: awsuser
Password: ••••••••

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



Address Book

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

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

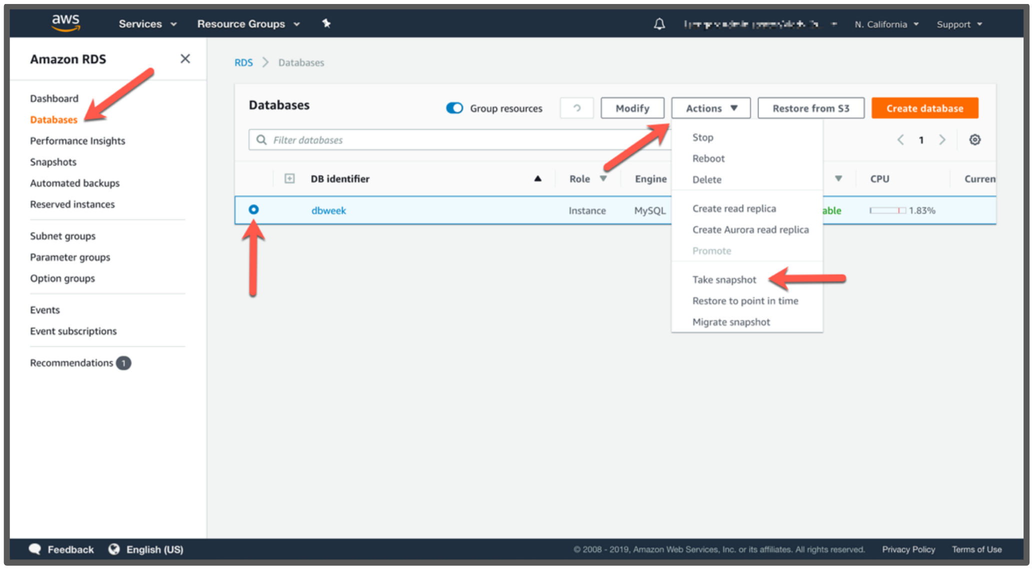
Now is a good time to take a snapshot of your RDS database. Taking a snapshot enables you to back up your DB Instance in a known state as frequently as you wish, and then restore to that specific state at any time.

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.

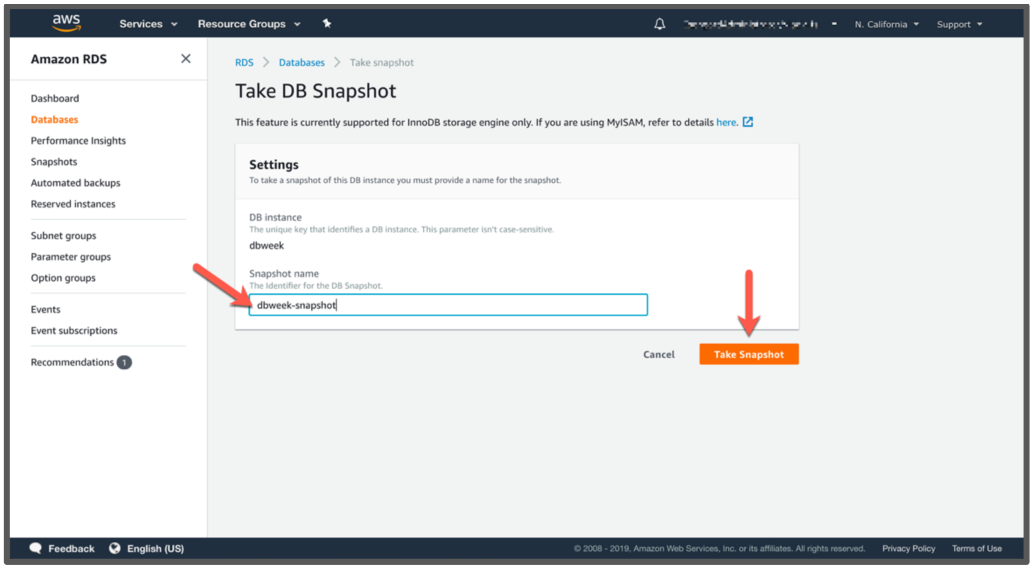
In the upper-right corner of the AWS Management Console, confirm you are in the desired AWS region (e.g., Oregon).

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**.



Take DB Snapshot

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.

Settings

To take a snapshot of this DB instance you must provide a name for the snapshot.

DB instance

The unique key that identifies a DB instance. This parameter isn't case-sensitive.

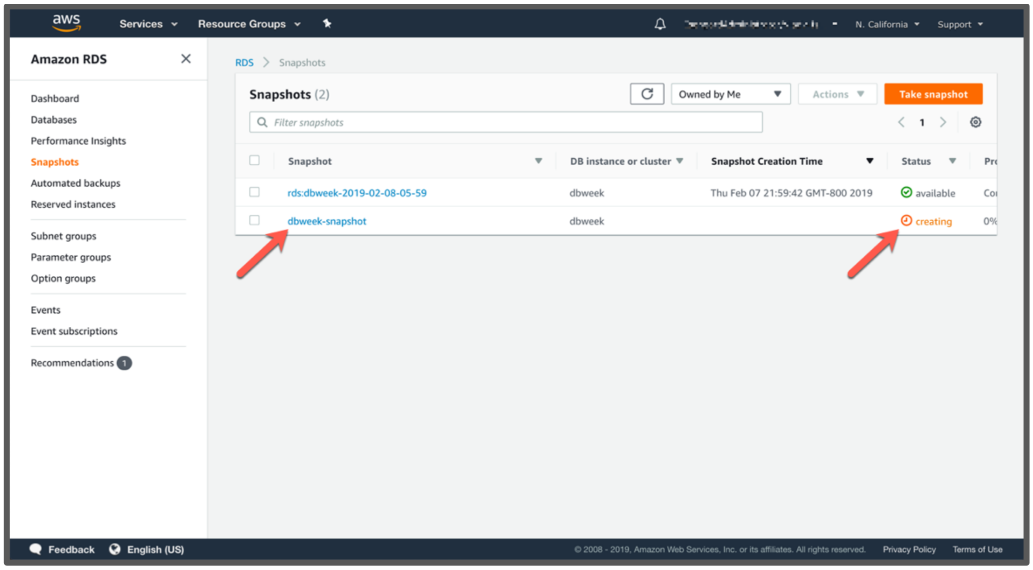
dbweek

Snapshot name

The identifier for the DB Snapshot.

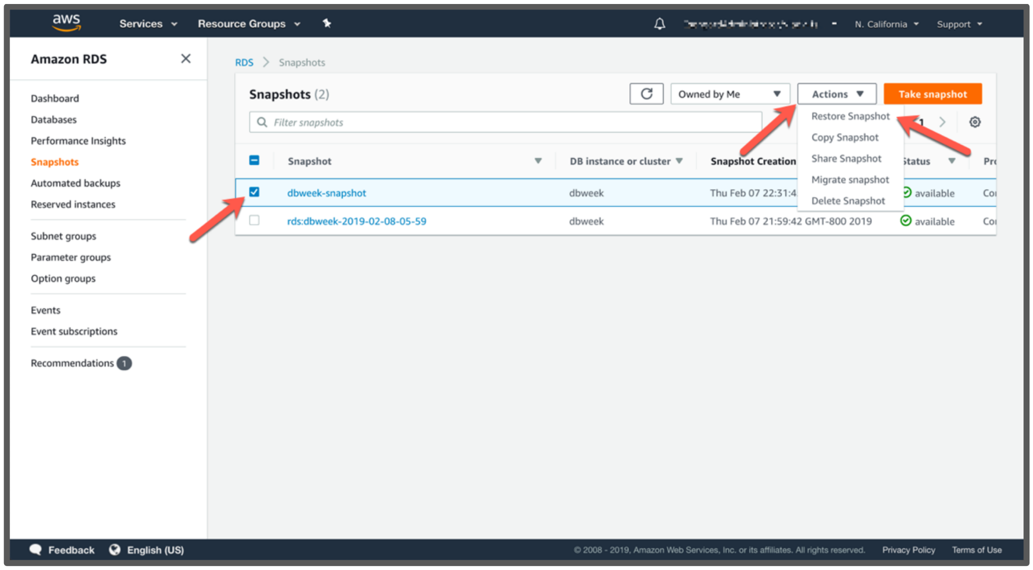
dbweek-snapshot

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.



☐	Snapshot	DB instance or cluster	Snapshot Creation Time	Status	Pr
☐	rds:dbweek-2019-02-08-05-59	dbweek	Thu Feb 07 21:59:42 GMT-800 2019	✓ available	Co
☐	dbweek-snapshot	dbweek		creating	0%

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!



☐	Snapshot	DB instance or cluster	Snapshot Creation	Status	Pr
☑	dbweek-snapshot	dbweek	Thu Feb 07 22:31:4	✓ available	Co
☐	rds:dbweek-2019-02-08-05-59	dbweek	Thu Feb 07 21:59:42 GMT-800 2019	✓ available	Co

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

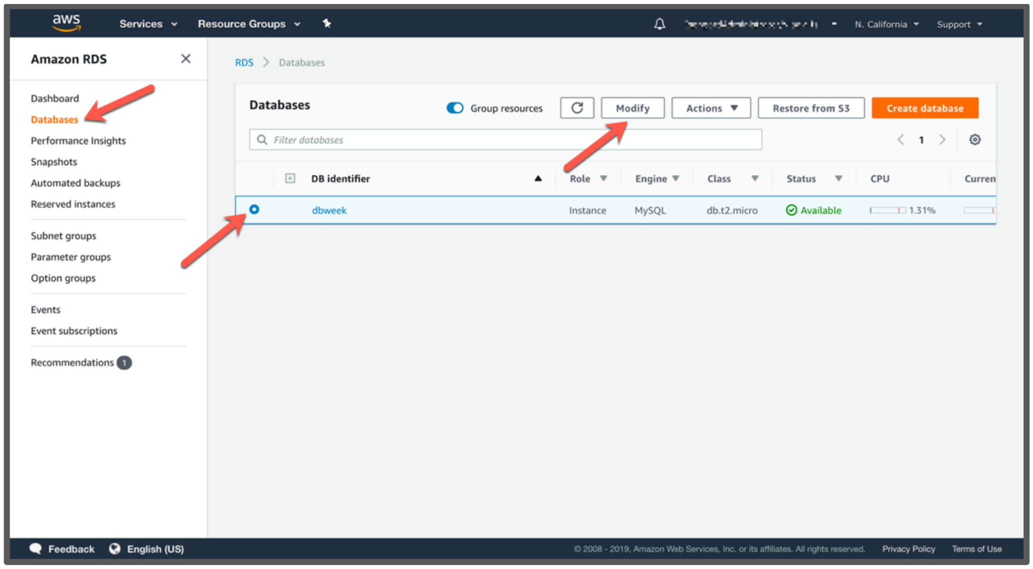
Scaling up and down with RDS is simple via the AWS Management Console. You can change the underlying server size, by selecting a new DB instance class, and you can modify the storage characteristics, by changing the storage type and growing the size of the storage to accommodate usage growth. You cannot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

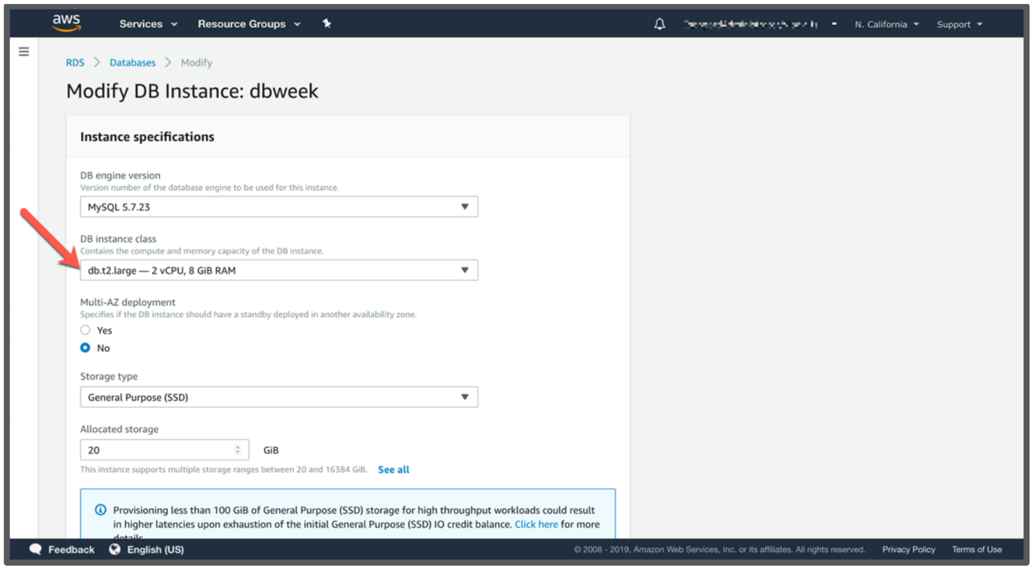
Click the **Modify** button at the top of the DB instance listing.



	DB identifier	Role	Engine	Class	Status	CPU	Curren
●	dbweek	Instance	MySQL	db.t2.micro	Available	1.31%	

On the **Modify DB Instance** screen, try changing to a **t2.large** DB instance class, and if you want, also grow the database allocated storage at the same time to 40GiB. Click **Next**.

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



Modify DB Instance: dbweek

Instance specifications

DB engine version
Version number of the database engine to be used for this instance.
MySQL 5.7.23 ▼

DB instance class
Contains the compute and memory capacity of the DB instance.
db.t2.large — 2 vCPU, 8 GiB RAM ▼

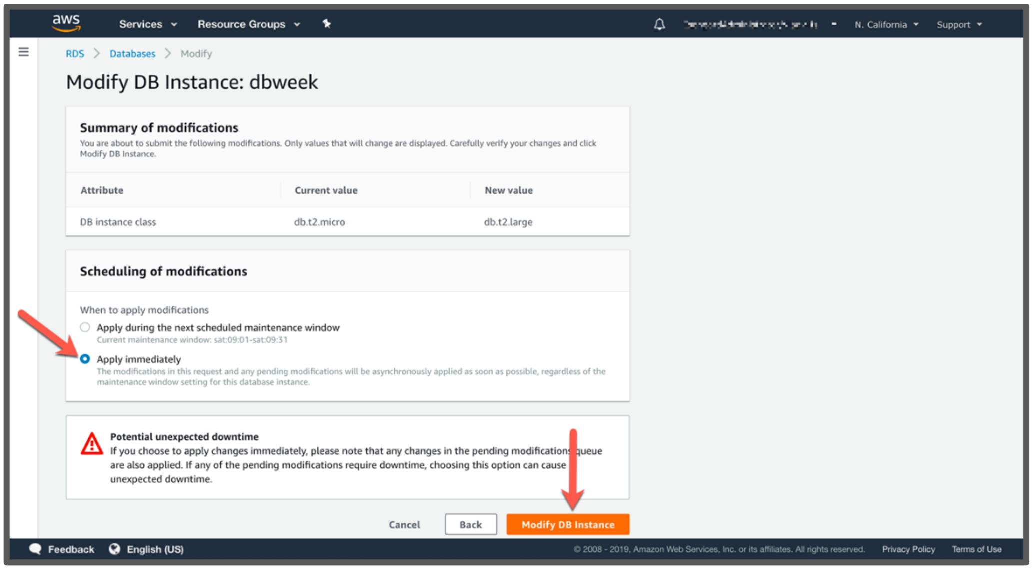
Multi-AZ deployment
Specifies if the DB instance should have a standby deployed in another availability zone.
○ Yes
● No

Storage type
General Purpose (SSD) ▼

Allocated storage
20 GiB
This instance supports multiple storage ranges between 20 and 16384 GiB. See all

Provisioning less than 100 GiB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Click here for more details.

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.



Modify DB Instance: dbweek

Summary of modifications

You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify DB Instance.

Attribute	Current value	New value
DB instance class	db.t2.micro	db.t2.large

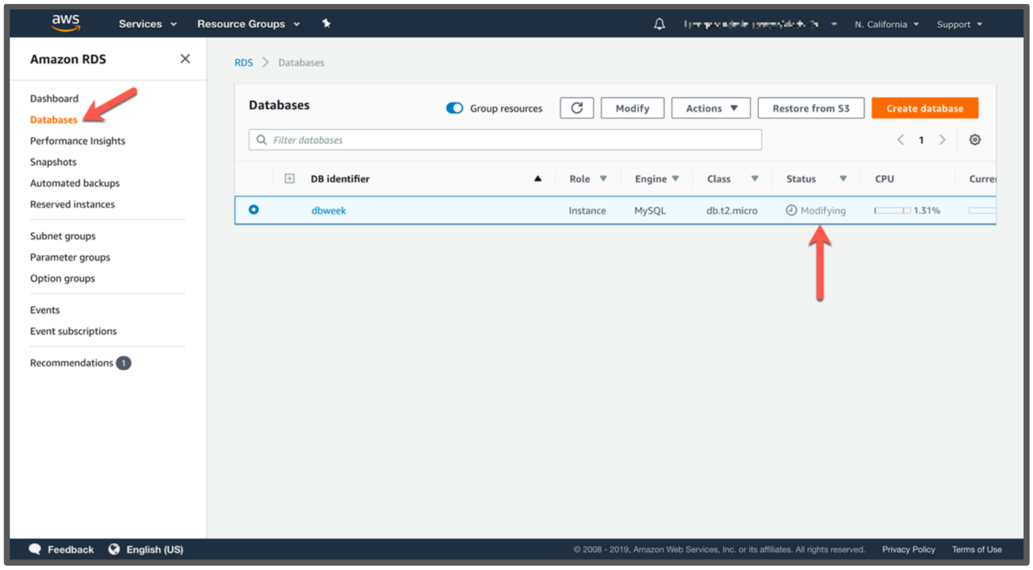
Scheduling of modifications

When to apply modifications
○ Apply during the next scheduled maintenance window
Current maintenance window: sat:09:01-sat:09:31
● Apply immediately
The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

Potential unexpected downtime
If you choose to apply changes immediately, please note that any changes in the pending modifications queue are also applied. If any of the pending modifications require downtime, choosing this option can cause unexpected downtime.

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.



DB identifier	Role	Engine	Class	Status	CPU	Curre
dbweek	Instance	MySQL	db.t2.micro	Modifying	1.31%	

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.

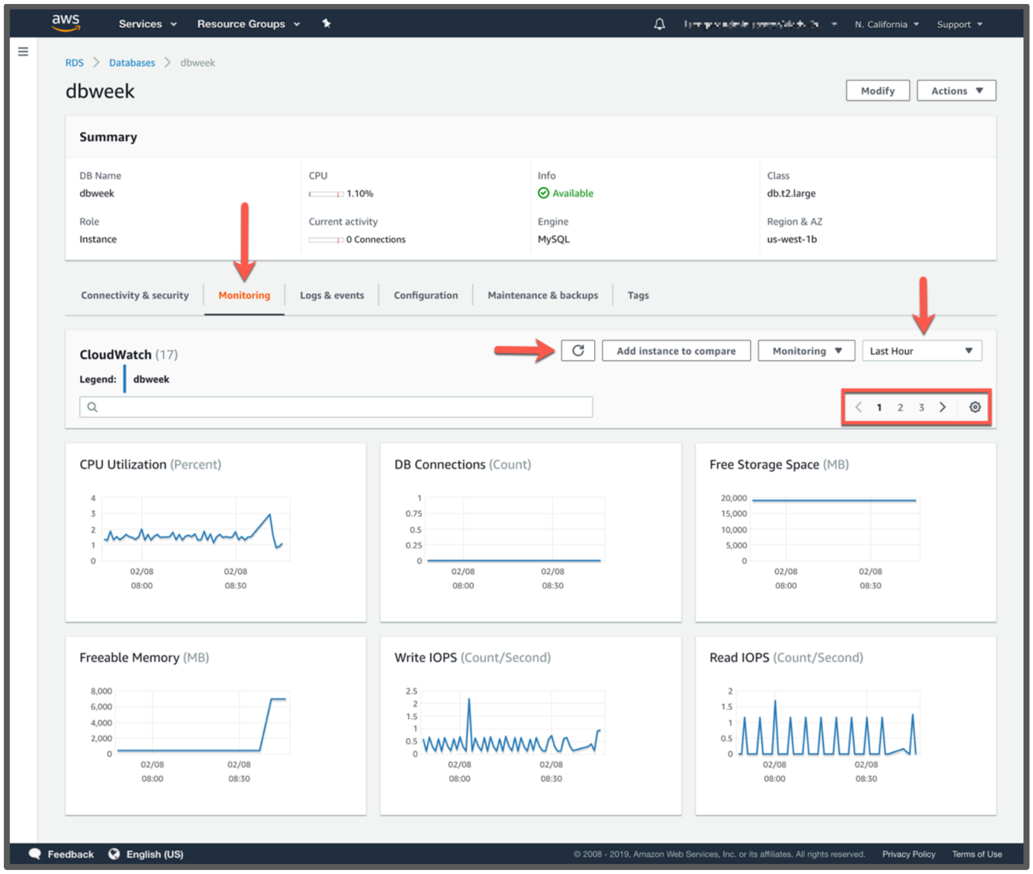
Sign into the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds, if the console is not already open, or showing a different service console.

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.



dbweek

Summary

DB Name: dbweek
CPU: 1.10%
Info: Available
Class: db.t2.large
Role: Instance
Current activity: 0 Connections
Engine: MySQL
Region & AZ: us-west-1b

CloudWatch (17)

Legend: dbweek

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

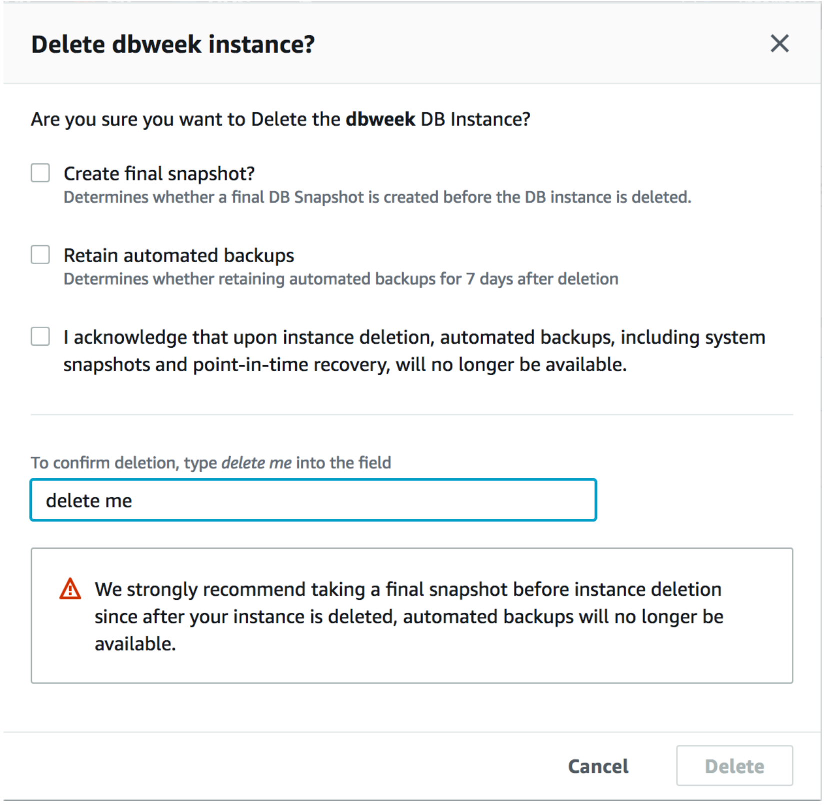
If you have additional time, try to do the following:

- 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.



Delete dbweek instance? ✕

Are you sure you want to Delete the **dbweek** DB Instance?

□ Create final snapshot?
Determines whether a final DB Snapshot is created before the DB instance is deleted.

□ Retain automated backups
Determines whether retaining automated backups for 7 days after deletion

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

To confirm deletion, type *delete me* into the field

delete me

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

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**.