

AWS Academy Cloud Architecting 2.x - Capstone Project

[Environment Overview](#)
[Environment Navigation](#)
[Access the AWS Management Console](#)
[Project Overview](#)
[Introducing the Example Social Research Organization](#)
[Solution requirements](#)
[Project deliverables](#)
[Assets for completing the project](#)

Environment Overview

This environment is long-lived. When the session timer runs to 0:00, the session will end, but any data and resources that you created in the AWS account will be retained. Running EC2 instances will be stopped for you if you did not already stop them before the session ends.

If you later launch a new session (for example, the next day), you will find that your work is still in the lab environment. Any existing EC2 instances may be started up again when you start the new session, but may be assigned new IPv4 public IP addresses.

You can continue to develop your solution as you progress through the course materials.

Preserving your budget

IMPORTANT: Monitor your lab budget in the lab interface above. Whenever you have an active lab session, the latest known remaining budget information will display at the top of this screen. This data comes from AWS Budgets which typically updates every 8 to 12 hours. Therefore *the remaining budget that you see may not reflect your most recent account activity*. **If you exceed your lab budget your lab account will be disabled and all progress and resources will be lost**. Therefore, it is important for you to manage your spending.

Suggestions to avoid overspending:

- Launch only the number of EC2 instances you need, sized to your requirements.
- Delete no longer needed resources and data.
- Typically, it is compute-type resources, such as EC2, RDS, Cloud9, NAT Gateway instances, that you leave running that most quickly use up your budget. Turn these off when no longer needed, or better yet, delete them.
- Stop any NAT gateways, EC2 instances, and RDS instances when not in use. If you will be building this solution over the course of a few weeks or months, consider deleting these resources entirely to save budget. You can always recreate them again later.
 - Be aware that if you do stop an RDS instance or cluster and leave it stopped for seven days, AWS may start it again automatically, which will increase the cost impact.
 - NAT Gateway resources if left in your account, will also use up your budget.
- Use the [AWS Simple Monthly Calculator](#) or [AWS Pricing Calculator](#) to estimate cost. *Example:* the estimate shown in the screenshot below calculated the cost of running the following resources for a month:
 - One t3.medium size Linux EC2 instance running 6 hours per day for a month in the us-east-1 Region.
 - One db.t2.small size MySQL RDS database with 20GB of storage, left running for a month in the us-east-1 Region.
 - One NAT Gateway left running, processing 1GB per month in the us-east-1 Region

Services	Estimate of Your Monthly Bill (\$ 65.50)			
Estimate of Your Monthly Bill				
<input checked="" type="checkbox"/> Show First Month's Bill (include all one-time fees, if any)				
Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.				
<input type="button" value="Export to CSV"/>	<input type="button" value="Save and Share"/>			
<input checked="" type="checkbox"/> Amazon EC2 Service (US East (N. Virginia))	\$ 7.62			
Compute:	\$ 7.62			
<input checked="" type="checkbox"/> Amazon RDS Service (US East (N. Virginia))	\$ 27.19			
DB instances:	\$ 24.89			
Storage:	\$ 2.30			
<input checked="" type="checkbox"/> Amazon VPC Service (US East (N. Virginia))	\$ 32.99			
NAT Gateway	\$ 32.99			
<input checked="" type="checkbox"/> AWS Support (Basic)	\$ 0.00			
Free Tier Discount:	\$ -2.30			
Total Monthly Payment:	\$ 65.50			

Pricing is subject to change. The calculation above is just an example from a point in time.

Environment Navigation

Use the **Readme** link above to return to these instructions at any time.

Use the **AWS Details** link above to access information about your environment.

After you start the lab, the AWS Details panel will display and you will need to choose **Readme** to return to these instructions.

Tip: you can resize this panel at anytime by dragging the bar to the left of these instructions to make it wider or narrower.

Use the **Reset** link above if you ever want to reset your AWS account back to the way it was in the beginning, before you ever ran sessions of this lab environment. **CAUTION:** if you choose reset and then choose Yes to confirm that you do want to reset, you

will **permanently delete** everything that you have created or stored in the AWS account.

The terminal window to the left of these instructions can be used to run AWS CLI commands or code such as AWS SDK for Python code (details provided below).

Access the AWS Management Console

1. At the top of these instructions, choose

Start Lab to start the lab session.

- The lab session is started and session information is displayed.
- A timer above shows the time remaining in the session.

Tip: You can refresh the session length at any time by choosing Start Lab again before the timer reaches 0:00.

2. Choose the [Readme link](#) to return to these instructions.

5. Connect to the AWS Management Console by choosing the **AWS** link above the terminal window.

- You should be connected to the AWS Management Console.

Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with the message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop-ups**.

Tip: if you are interested in interacting with the AWS account programmatically, read the [Configuring and using the terminal in the browser](#) section below for details.

Project overview

This project provides you with an opportunity to demonstrate the solution design skills that you develop throughout this course. Your assignment is to design and deploy a solution for the following case.

By the end of this project, you should be able to apply the architectural design principles that you learned in this course to:

- Deploy a PHP application that runs on an Amazon Elastic Compute Cloud (Amazon EC2) instance
- Create a database instance that the PHP application can query
- Create a MySQL database from a structured query language (SQL) dump file
- Update application parameters in an AWS Systems Manager Parameter Store
- Secure the application to prevent public access to backend systems

Introducing the Example Social Research Organization

Example Social Research Organization is a (fictitious) nonprofit organization that provides a website for social science researchers to obtain global development statistics. For example, visitors to the site can look up various data, such as the life expectancy for any country in the world over the past 10 years.

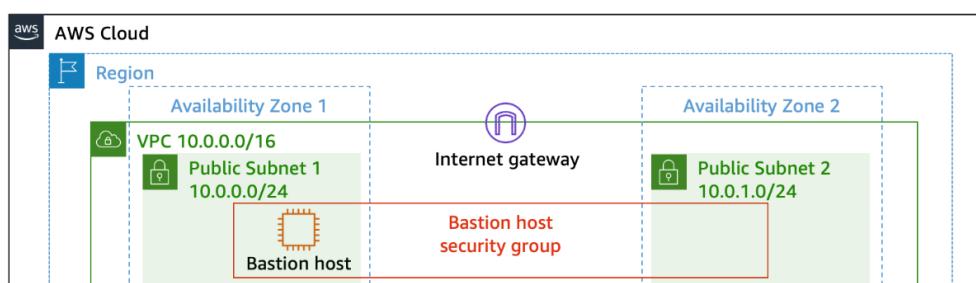
Shirley Rodriguez, a researcher at the organization, developed the website. She thought it would be valuable to share the data that she had gathered with other researchers. Shirley stores the data in a MySQL database, and the data is available through a PHP website that she built. She initially published the site through a commercial hosting company that provides limited support for technical issues and security.

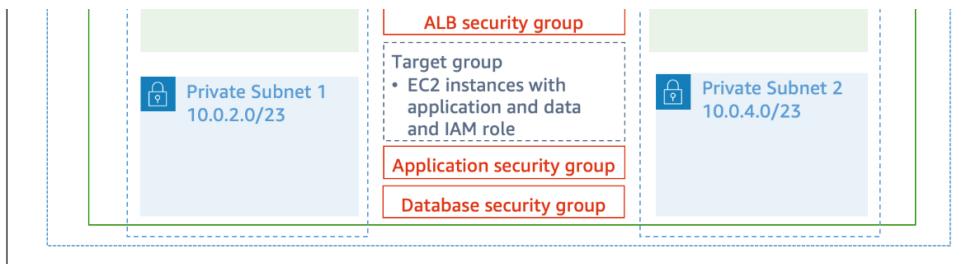
Over the past year, Shirley's website has grown in popularity. As a result of increased traffic, she started receiving complaints that the site is not as responsive as it used to be. She also experienced an attempted ransomware security breach. The security breach was unsuccessful, but her supervisor, Mateo Jackson, suggested that Shirley investigate new ways to host the website.

Shirley heard about Amazon Web Services (AWS), and initially moved her website and database to an EC2 instance that runs in a public subnet. She also runs an instance of MySQL on the same EC2 instance.

Shirley approached your team to make sure that her current design follows best practices. She wants to make sure that she has a robust and secure website. One of your colleagues started the process of migrating the site to a more secure implementation, but they were reassigned to another project. Your tasks are to complete the implementation, make sure that the website is secure, and confirm that the website returns data from the query page.

The following summary lists the solution requirements, and provides a diagram of the current environment.





Solution requirements

- Provide secure hosting of the MySQL database
- Provide secure access for an administrative user
- Provide anonymous access to web users
- Run the website on a t2.micro EC2 instance, and provide Secure Shell (SSH) access to administrators
- Provide high availability to the website through a load balancer
- Store database connection information in the AWS Systems Manager Parameter Store
- Provide automatic scaling that uses a launch template

The following parameters are used by the PHP application to connect to the database:

- /example/endpoint
- /example/username
- /example/password
- /example/database

These parameter values are case sensitive.

Project deliverables

To complete this assignment, you must:

- Deploy a PHP application that meets the system requirements outlined above
- Submit a diagram that illustrates your solution
- Submit a written summary of the design decisions that you made to achieve the result

Assets for completing the project

You can use the following assets for this project:

- [A SQL dump file that contains sample data](#)
- [A zip file that contains the PHP and image files for the Example Social Research Organization website](#)

Note: An EC2 launch template named Example-LT is provided in the AWS account. It already has both the SQL dump file and the unzipped PHP application resources built into it.