

CS3204 | Cloud Infrastructure and Services

Lab 2: Web Services in the Cloud

Content

- Goals & Objectives
- Benefits of the AWS

Lab Work Content

- Task 1: Cloud Web Instance
 - AWS & Its €1 Fee
 - Rest-API & Fake Data
 - Finding Salvation in Mining
- Task 2: Cloud Web Database
 - Form Submission in PHP
 - The choice of PHP & working with MyAdmin
- Experience gained from using, testing and debugging new software

Goals & Objectives

My goal for this 3ish week lab is to put into practice the theoretical knowledge I have acquired from lectures in relation to working with the Cloud environment and how different companies use this service to speed up initial development of their projects.

I will accomplish this by documenting my progression through the set-up of one of Amazon's Web Services ([AWS](#)) called [Elastic Beanstalk](#), extracting data from an API source, creating my own database and providing a web-based method to upload files to it.

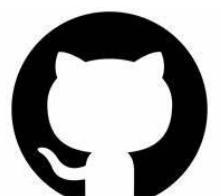
By the end of this Lab work, I hope to have enhanced my understanding of tools that can be used to speed up development of website building as well as to familiarise myself with services such as AWS & Elastic Beanstalk.



All code can be found in my GitHub Repository by following this link:

<https://github.com/Passe-Sleeper/UNI-Public/tree/main/Year3/CS3204/Lab2>

GitHub



GitHub

Benefits of AWS

Amazon Web Services is a nextensive provider of cloud based tools that allow various people to work on projects and web applications without the need for having their own server. This leads to many benefits which include: Scalability, Flexibility, Cost-effectiveness, Reliability, etc.



A bit of detail on each of these:

Scalability: AWS allows you to easily scale your infrastructure up or down based on your needs. You can quickly add or remove resources as your business demands change, without any major disruptions.

Flexibility: AWS offers a wide range of services that cater to various business requirements. Whether you need computing power, storage, databases, machine learning, or analytics, AWS has a service to fulfil your needs.

Cost-effectiveness: With AWS, you only pay for what you use. There are no long-term contracts or upfront costs. This pay-as-you-go model can help you save costs by eliminating the need for investing in expensive hardware and infrastructure.

Reliability: AWS has a global infrastructure that is designed to provide high availability and reliability. They offer multiple data centres worldwide, ensuring that your applications and data remain accessible even in the event of a failure.



There are also a couple of disadvantages that are also associated with similar benefits, such as: Complexity, Dependency, Cost, Security, etc.



A bit on each one of these:

Cost: While AWS offers a pay-as-you-go pricing model, costs can add up quickly. If you don't carefully monitor and manage your usage, you may end up with unexpectedly high bills. Additionally, certain services and features may incur additional charges, leading to hidden costs.

Complexity: AWS is a complex platform with a steep learning curve. It requires a solid understanding of various services, configurations, and management practices, which can be overwhelming for beginners or small organisations with limited technical expertise.

Security: As with any cloud service provider, there are security concerns when using AWS.

Dependency: By using AWS, you become dependent on their services and infrastructure. This can be a disadvantage if AWS experiences outages or disruptions, which could impact your business operations. It's important to have contingency plans in place to mitigate such risks.



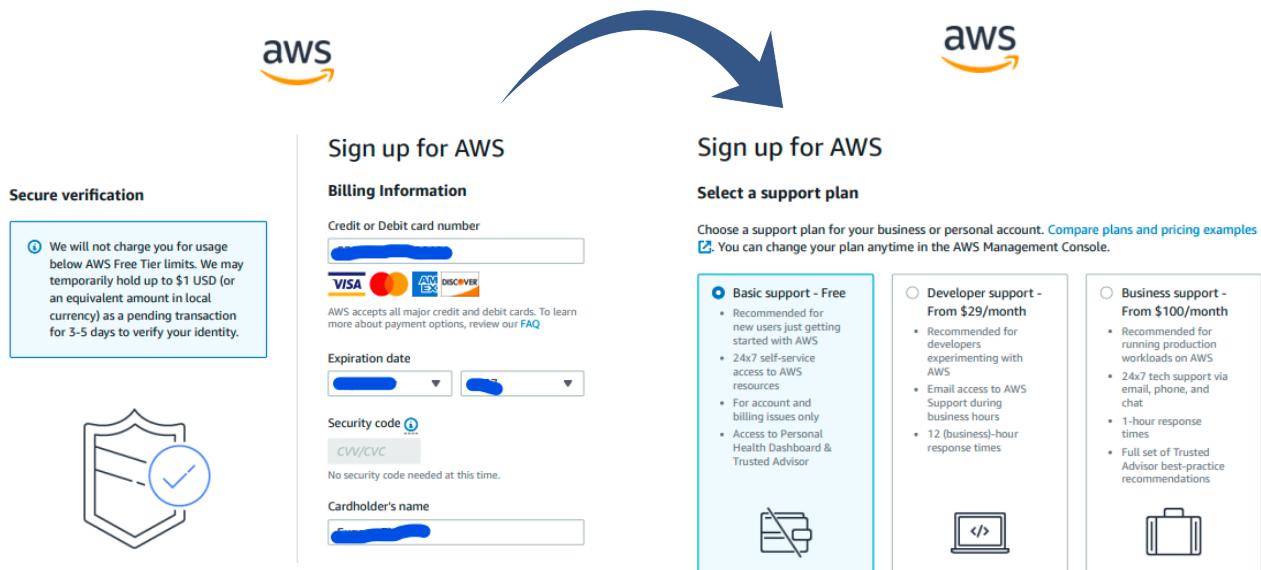
Overall, AWS offers a very wide branching service that takes time to master, is great for start-ups, but once the company grows, it is probably in their interest to start up their own website & create their own hosting services for security and cost reasons.

Lab Work Content

Task 1: Cloud Web Instance

AWS & Its €1 Fee

When initially following the lab instructions of signing up to AWS services, I was forced to create an account and even pay a deposit of €1! I was quite surprised at this, but, since it was just €1, it should be fine...



After Registration of my Account, I continued to follow the same procedures as stated in the notes, that being to create my own Web-based application and set the Region!

Region	Endpoint
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1

Configure environment

Environment tier

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment Run a website, web application, or web API that serves HTTP requests. Learn more [\[Link\]](#)
- Worker environment Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more [\[Link\]](#)

Application information

Application name FormSubmission Maximum length of 100 characters.

▶ Application tags (optional)

NOTE: I choose to code this Assignment in PHP to further develop my own skills!

aws Services Search

Step 1 Configure environment

Step 2 Configure service access

Step 3 - optional

Set up networking, database, and tags

Step 4 - optional

Configure instance traffic and scaling

Step 5 - optional

Configure updates, monitoring, and logging

Step 6 Review

Environment description

Platform

Platform type Managed platform Platforms published and maintained by Amazon Elastic Beanstalk. Learn more [\[Link\]](#)

Custom platform Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform PHP

Platform branch PHP 8.2 running on 64bit Amazon Linux 2023

Platform version 4.0.3 (Recommended)

Application code

Sample application

Existing version Application versions that you have uploaded.

Upload your code Upload a source bundle from your computer or copy one from Amazon S3.

Presets

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets Single instance (free tier eligible) Single instance (using spot instance) High availability High availability (using spot and on-demand instances) Custom configuration

Roles (2) Info

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

Role name	Trusted entities
AWSServiceRoleForSupport	AWS Service: support (Service-Linked -)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked -)

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSFlasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy
AWSElasticBeanstalkWebTier	AWS managed	Permissions policy
AWSElasticBeanstalkWorkerTier	AWS managed	Permissions policy

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more [\[Link\]](#)

Service role

Create and use new service role Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

admin	<input type="button" value="Edit"/>
-------	-------------------------------------

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more [\[Link\]](#)

Choose a key pair	<input type="button" value="Edit"/>
-------------------	-------------------------------------

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

admin	<input type="button" value="Edit"/>
-------	-------------------------------------

View permission details

Cancel Skip to review Previous Next

When configuring the environment, I had to Name the Application, to Set-up the Domain Name, and the opportunity to upload my own Initial Code (for the 1st attempt I stuck with the Sample code).

I had to also create an 'admin' role for the EC2 service access, this allowed me to use the web service application at a later stage!

aws Services Search

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Set up networking, database, and tags

Step 4 - optional
Configure instance traffic and scaling

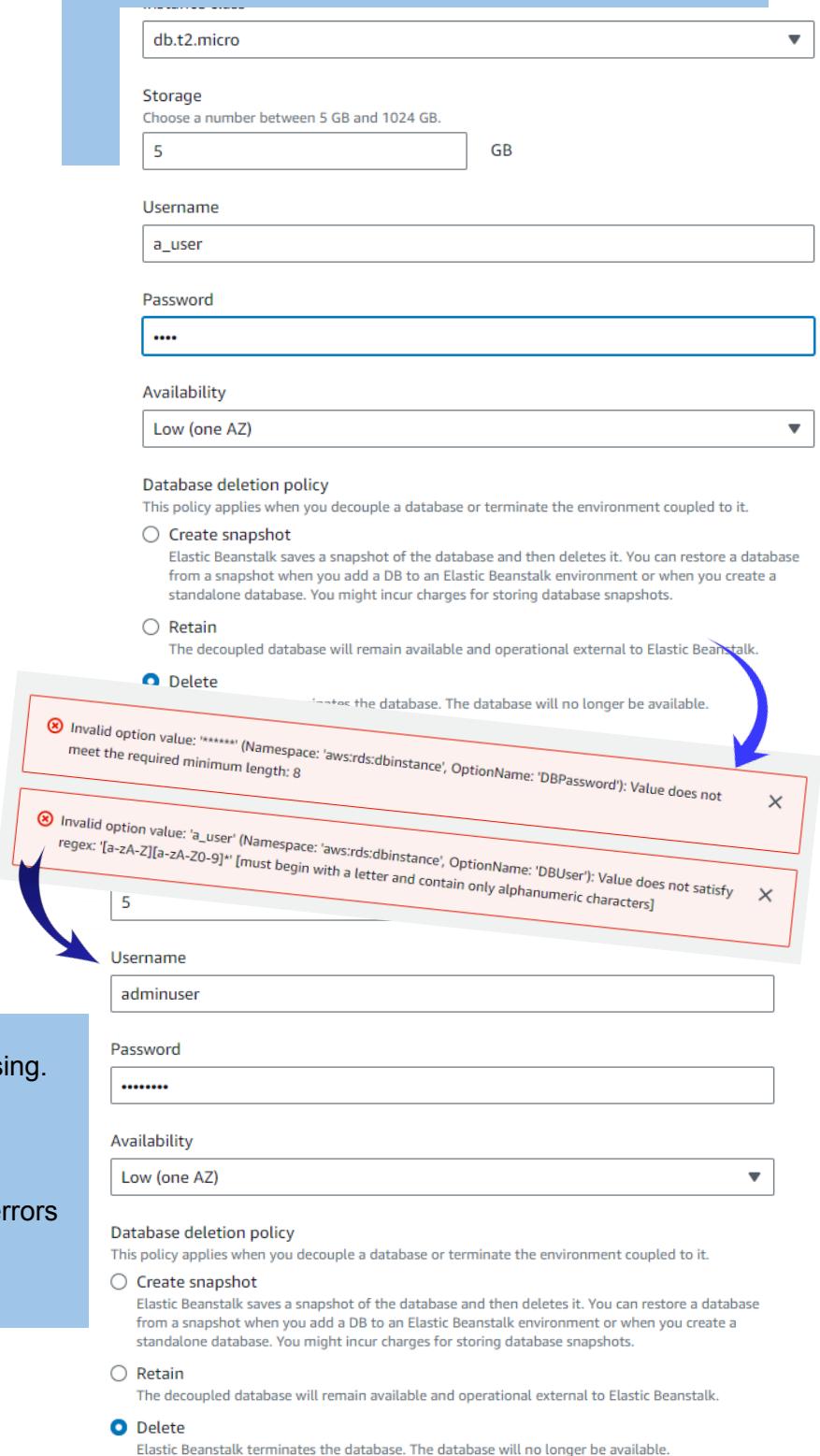
Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

I added the Database that I would be using. Created a Username & Password as required of me.

Of course, coming across a few minor errors in Username & Password length - ouch!

Next I had to Set-up the Database connection as this would be used Task 2!



db.t2.micro

Storage
Choose a number between 5 GB and 1024 GB.
5 GB

Username
a_user

Password

Availability
Low (one AZ)

Database deletion policy
This policy applies when you decouple a database or terminate the environment coupled to it.

Create snapshot
Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.

Retain
The decoupled database will remain available and operational external to Elastic Beanstalk.

Delete
Terminates the database. The database will no longer be available.

Invalid option value: '***' (Namespace: 'aws:rds:dbinstance', OptionName: 'DBPassword'): Value does not meet the required minimum length: 8**

Invalid option value: 'a_user' (Namespace: 'aws:rds:dbinstance', OptionName: 'DBUser'): Value does not satisfy regex: '[a-zA-Z][a-zA-Z0-9]*' [must begin with a letter and contain only alphanumeric characters]

Username
adminuser

Password

Availability
Low (one AZ)

Database deletion policy
This policy applies when you decouple a database or terminate the environment coupled to it.

Create snapshot
Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.

Retain
The decoupled database will remain available and operational external to Elastic Beanstalk.

Delete
Elastic Beanstalk terminates the database. The database will no longer be available.

Steps 4 & 5, were just skipped as they were not necessary for my project, but I watched a [tutorial](#) to help me understand what they were for.

Before we continue, let us perform a quick **Review** of what we have created/set-up!

Step 1: Configure environment

[Edit](#)

Environment information

Environment tier	Application name
Web server environment	FormSubmission
Environment name	Application code
FormSubmission-env	Sample application
Platform	
arn:aws:elasticbeanstalk:eu-west-1::platform/PHP 8.2	
running on 64bit Amazon Linux 2023/4.0.3	

Step 2: Configure service access

[Edit](#)

Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::492743373237:role/admin	admin
min	

Step 3: Set up networking, database, and tags

[Edit](#)

Networking, database, and tags [Info](#)

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Database

Database availability	Has coupled database	Database deletion policy
false	true	Delete
Database engine	Database engine version	Database instance class
mysql	8.0.33	db.t2.micro
Database password	Database storage	Database username
*****	5	adminuser

Sample Application Code

After the Initial set-up, the Web Environment was ready to launch the application.

The screenshot shows the AWS Elastic Beanstalk environment configuration page for 'FormSubmission-env'. At the top, a blue header bar indicates 'Elastic Beanstalk is launching your environment. This will take a few minutes.' Below this, the navigation path is 'Elastic Beanstalk > Environments > FormSubmission-env'. The main content area is divided into two columns: 'Environment overview' and 'Platform'. The 'Environment overview' column contains fields for Health (Unknown), Domain (phpbasedformsubmission.eu-west-1.elasticbeanstalk.com), Environment ID (e-2xieykrj5), and Application name (FormSubmission). A note at the bottom of this column states: 'Note: the Website links do-not work due to being deleted after the assignment was completion. [90-free]'. The 'Platform' column shows PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3, with a running version of '-' and a supported platform state. A green banner at the bottom left says 'Environment successfully launched.' On the right side of the page, there are two sections: 'What's Next?' and 'AWS SDK for PHP'.

Environment overview

- Health: Unknown
- Domain: phpbasedformsubmission.eu-west-1.elasticbeanstalk.com
- Environment ID: e-2xieykrj5
- Application name: FormSubmission

Note: the Website links do-not work due to being deleted after the assignment was completion. [90-free]

Platform

- Platform: PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3
- Running version: -
- Platform state: Supported

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git](#)
- [Using Amazon RDS with PHP](#)
- [Customizing the Software on EC2 Instances](#)
- [Customizing Environment Resources](#)

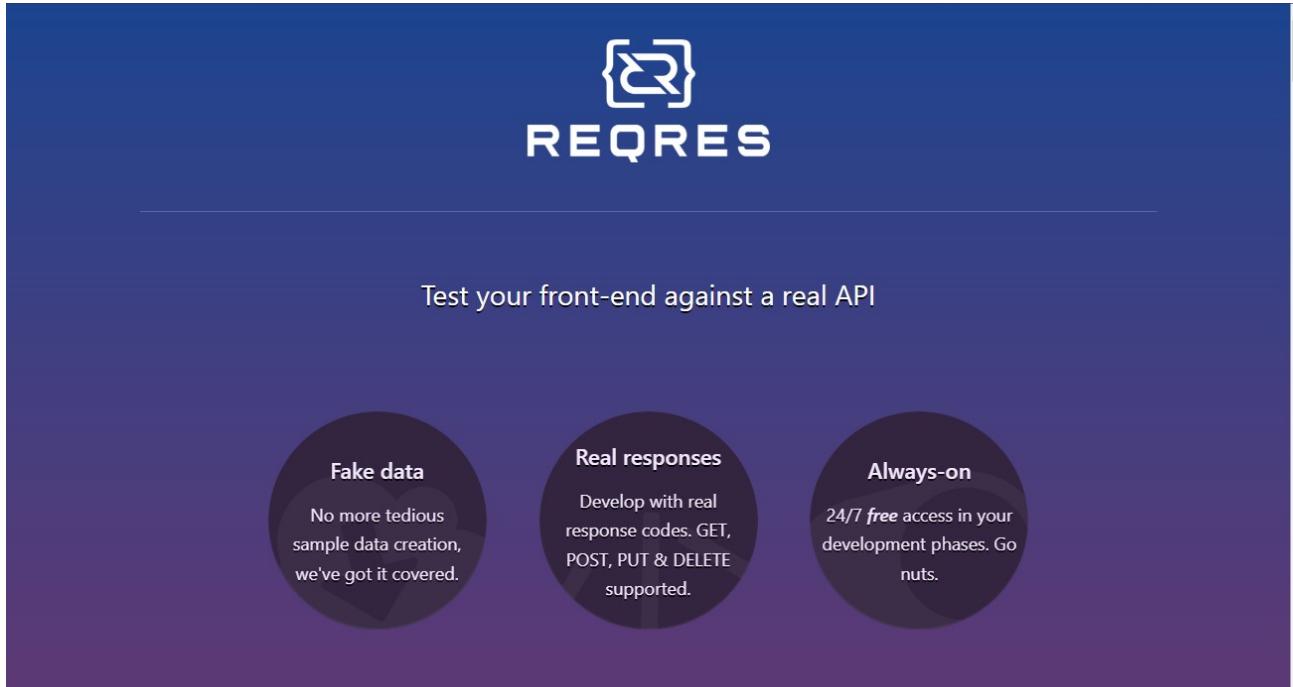
AWS SDK for PHP

- [AWS SDK for PHP home](#)
- [PHP developer center](#)
- [AWS SDK for PHP on GitHub](#)

Excellent, that procedure was relatively straightforward. It is now time to upload my own project onto the Cloud to see how it goes!

Rest-API & Fake Data

Initially I wished to import data from a website called [Reqres.in](https://reqres.in), and in order to do so, I followed some online tutorials on how the Rest-API worked. I even coded up a small application to extract the code from the database and to write out everything it collected. The result was consistent with what other user's of this dataset had done in the past.



I wished to get access to the following JSON layout data-set:

A screenshot of a browser window showing a JSON response. The URL in the address bar is https://reqres.in/api/users?page=2. The JSON data is as follows:

```
{"page":2,"per_page":6,"total":12,"total_pages":2,"data": [{"id":7,"email":"michael.lawson@reqres.in","first_name":"Michael","last_name":"Lawson","avatar":"https://reqres.in/img/faces/7-image.jpg"}, {"id":8,"email":"lindsay.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":"https://reqres.in/img/faces/8-image.jpg"}, {"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":"https://reqres.in/img/faces/9-image.jpg"}, {"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":"https://reqres.in/img/faces/10-image.jpg"}, {"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":"https://reqres.in/img/faces/11-image.jpg"}, {"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://reqres.in/img/faces/12-image.jpg"}], "support": {"url": "https://reqres.in/#support-heading", "text": "To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

This task was simple in nature, but required a lot of background work to accomplish. The reason for that was my choice in programming language - PHP.

My choice | PHP

Why pick PHP and not stick to Python? This module, Cloud Infrastructure and Services, teaches you about various web services; How they work, Why they work and most importantly: Why we need them. This module teaches you in what direction the current modern (cloud) world is moving in, as well as the history of what came before!

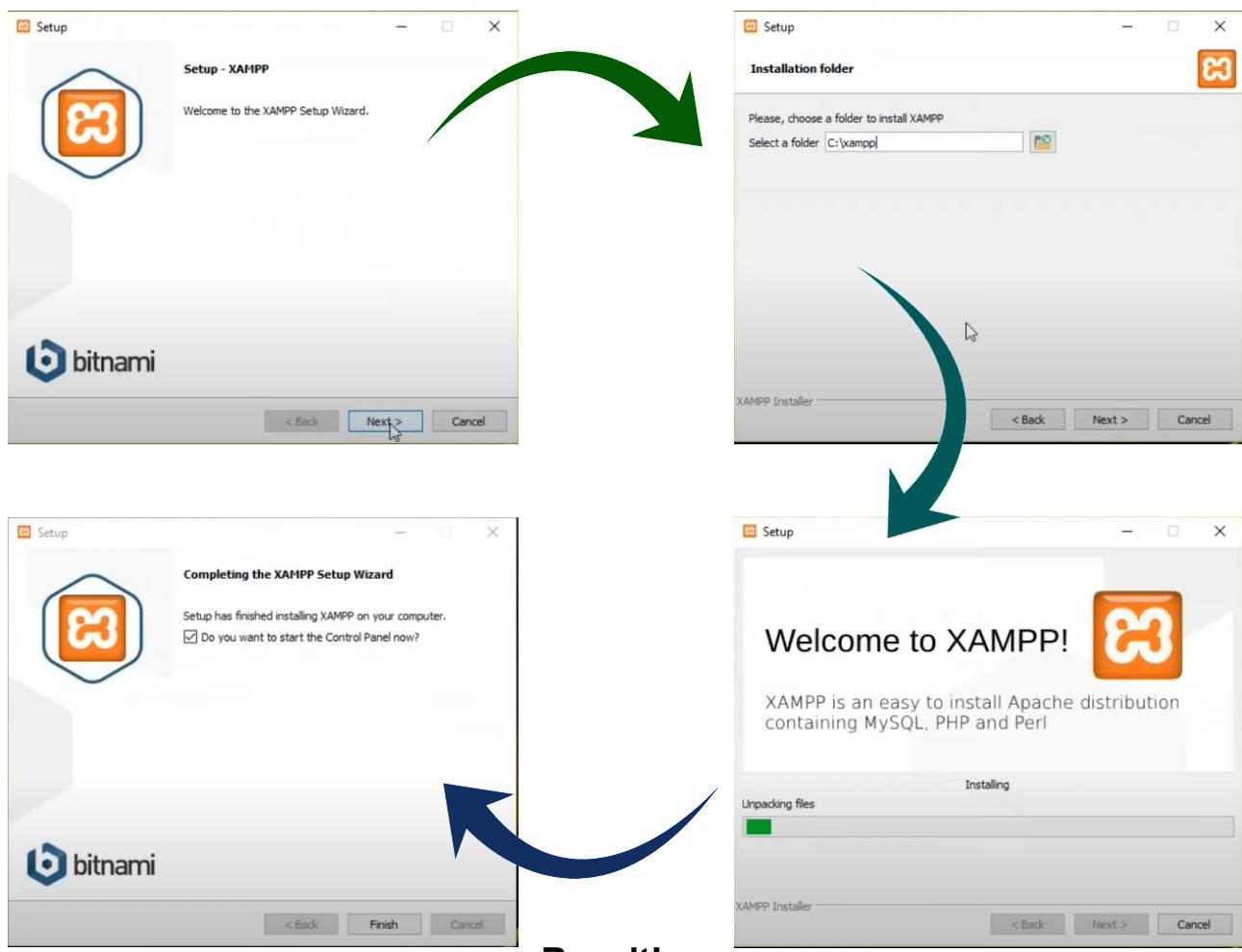
PHP as a programming language is old and has its disadvantages: Poor Security Features, Limited Debugging Tools, Inability to Modify Core Behaviour, Not as Versatile and that there are easier languages to learn (due to complex writing).

However, I believe that this assignment is the perfect opportunity for me to step into learning this new language! A tutorial here & a tutorial there, I should be able to write a simple website with it!

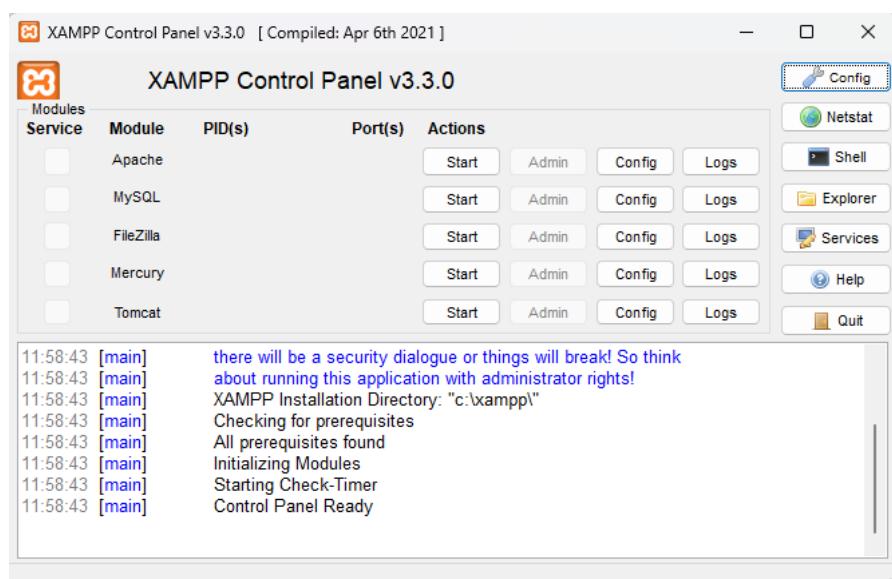
Local Testing with XAMPP

While doing my research online, I found suggestions/recommendations, for if you are using PHP for Deployment purposes, it is suggested to run an Apache server and then run your application using it rather than run your code via terminal (as with Python).

To test and consequently learn all of this for myself, I went through a XAMPP Installation Guide which can be found [here](#) and watched a Youtube [video](#) to confirm.



Result!



After writing the generic lines "Hello World!" and starting the base Apache server, I was able to launch my localised Web-page!

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
	Apache	15180 26088	80, 443	<button>Stop</button> <button>Add</button>
	MySQL			<button>Start</button> <button>Add</button>
	FileZilla			<button>Start</button> <button>Add</button>
	Mercury			<button>Start</button> <button>Add</button>
	Tomcat			<button>Start</button> <button>Add</button>

localhost/demo/index.php

Hello World

[main] XAMPP Installation Directory: "c:\xampp\"
[main] Checking for prerequisites
[main] All prerequisites found
[main] Initializing Modules
[main] Starting Check-Timer
[main] Control Panel Ready
[Apache] Attempting to start Apache app...
[Apache] Status change detected: running

After the initial success, I began writing software to work with the Resque Database.

```
{"page":2,"per_page":6,"total":12,"total_pages":2,"data": [{"id":7,"email":"michael.lawson@reqres.in","first_name":"Michael","last_name":"Lawson","avatar":"https://reqres.in/img/faces/7-image.jpg"}, {"id":8,"email":"lindsay.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":"https://reqres.in/img/faces/8-image.jpg"}, {"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":"https://reqres.in/img/faces/9-image.jpg"}, {"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":"https://reqres.in/img/faces/10-image.jpg"}, {"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":"https://reqres.in/img/faces/11-image.jpg"}, {"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://reqres.in/img/faces/12-image.jpg"}],"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}
```

The result was the following code & output:

```
fetch.php
1 <?php
2
3 $ch = curl_init();
4
5 $url = "https://reqres.in/api/users?page=2";
6
7 curl_setopt($ch, CURLOPT_URL, $url);
8 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
9
10 $resp = curl_exec($ch);
11
```

```
$resp = curl_exec($ch);
if ($e = curl_error($ch)) {
    echo $e;
} else {
    $decoded = json_decode($resp);
    print_r($decoded);
}
curl_close($ch);
?>
```

```
stdClass Object ( [page] => 2 [per_page] => 6 [total] => 12 [total_pages] => 2 [data] => Array ( [0] => stdClass Object ( [id] => 7 [email] => michael.lawson@reqres.in [first_name] => Michael [last_name] => Lawson [avatar] => https://reqres.in/img/faces/7-image.jpg ) [1] => stdClass Object ( [id] => 8 [email] => lindsay.ferguson@reqres.in [first_name] => Lindsay [last_name] => Ferguson [avatar] => https://reqres.in/img/faces/8-image.jpg ) [2] => stdClass Object ( [id] => 9 [email] => tobias.funke@reqres.in [first_name] => Tobias [last_name] => Funke [avatar] => https://reqres.in/img/faces/9-image.jpg ) [3] => stdClass Object ( [id] => 10 [email] => byron.fields@reqres.in [first_name] => Byron [last_name] => Fields [avatar] => https://reqres.in/img/faces/10-image.jpg ) [4] => stdClass Object ( [id] => 11 [email] => george.edwards@reqres.in [first_name] => George [last_name] => Edwards [avatar] => https://reqres.in/img/faces/11-image.jpg ) [5] => stdClass Object ( [id] => 12 [email] => rachel.howell@reqres.in [first_name] => Rachel [last_name] => Howell [avatar] => https://reqres.in/img/faces/12-image.jpg ) [support] => stdClass Object ( [url] => https://reqres.in/#support-heading [text] => To keep ReqRes free, contributions towards server costs are appreciated! ) )
```

A Break from AWS

I attempted to bring this same file to the AWS server. However, when attempting to run it on the servers, I encountered a lot of errors in relation to the file health being Degraded.

The screenshot shows the 'Environment overview' section of the AWS Elastic Beanstalk console. It includes details like Health (Degraded), Environment ID (e-2xieyjkrj5), Application name (FormSubmission), and Platform (PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3). A deployment log at the top shows three successful steps: launching the environment, uploading the file, and completing the update. The status bar at the bottom indicates the environment is currently 'Degraded'.

Environment overview		Platform
Health ☒ Degraded - View causes	Environment ID e-2xieyjkrj5	Platform PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3
Domain phpbasedformsubmission.eu-west-1.elasticbeanstalk.com	Application name FormSubmission	Running version -
		Platform state ☒ Supported

I tried to resolve the issue both via my own efforts and by googling excessively. Unfortunately, nothing worked.

Instance ID	Status
<p>▼ Severe</p>	
i-022e6798e2...	<ul style="list-style-type: none">Application deployment failed at 2023-11-20T14:03:08Z with exit status 1 and error: Engine execution has encountered an error.Incorrect application version "FormSubmission-version-2" (deployment 3). Expected version "Sample" (deployment 1).

The conclusion was that the selection of Sample code instead of my own, at the very start of the Environment creation process led to these errors!

I recreated the Environment and uploaded my own file instead of using the provided Sample file, however, I only changed the error from Degraded to Warning status.

The screenshot shows the AWS Elastic Beanstalk Environment Overview page for 'GoldGraph-env'. At the top, a green banner indicates 'Environment successfully launched.' Below the banner, the navigation path is 'Elastic Beanstalk > Environments > GoldGraph-env'. The main content area is divided into two sections: 'Environment overview' and 'Platform'. The 'Environment overview' section contains details like Health (Warning), Domain (goldgraph.eu-west-1.elasticbeanstalk.com), Environment ID (e-ghcrmxiea3), and Application name (Gold Graph). The 'Platform' section shows PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3, Running version myversion, and Platform state Supported.

Even worse, when I tried accessing the website, nothing was working!!

This site can't be reached

goldgraph.eu-west-1.elasticbeanstalk.com refused to connect.

Try:

- Checking the connection
- [Checking the proxy and the firewall](#)

ERR_CONNECTION_REFUSED

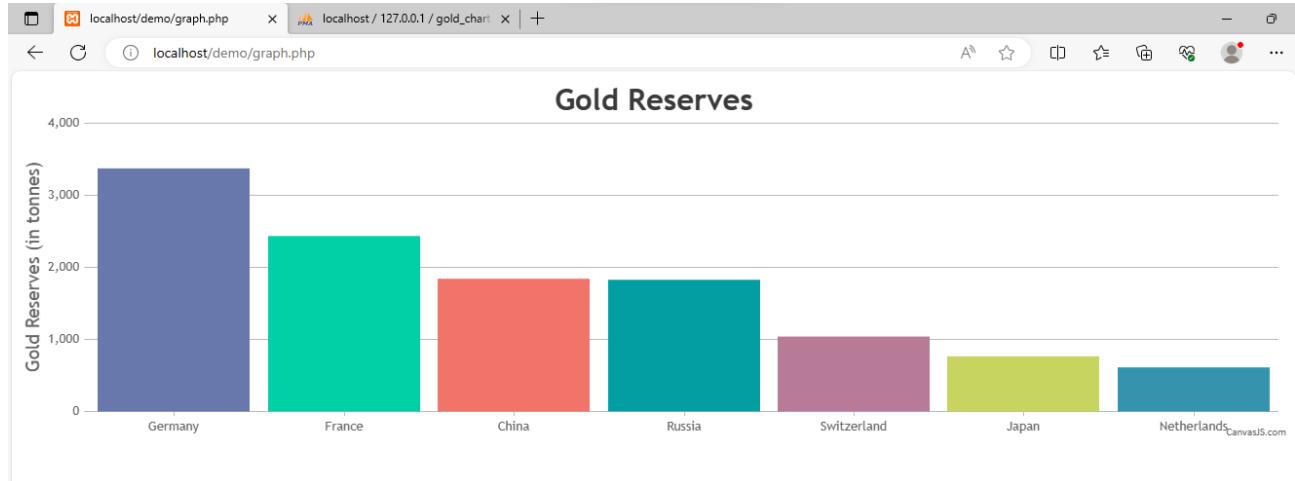
[Reload](#) [Details](#)

As none of this was working, I decided to not get frustrated with it and to return back to the Local Host Apache server and continue working on my Project.

Who knows.. Maybe I will come across a solution later...?

Finding Salvation in Mining

As AWS had given up on me, I went and learned how to create graphs and draw data points on them. This was a very fun and stress relieving exercise for me! I found a cool library online called [Canvas.js](#) that contains heaps of different graphs and the style sheets needed to run them. I built a static model of the Gold Reserves of a couple of countries! This was the result:



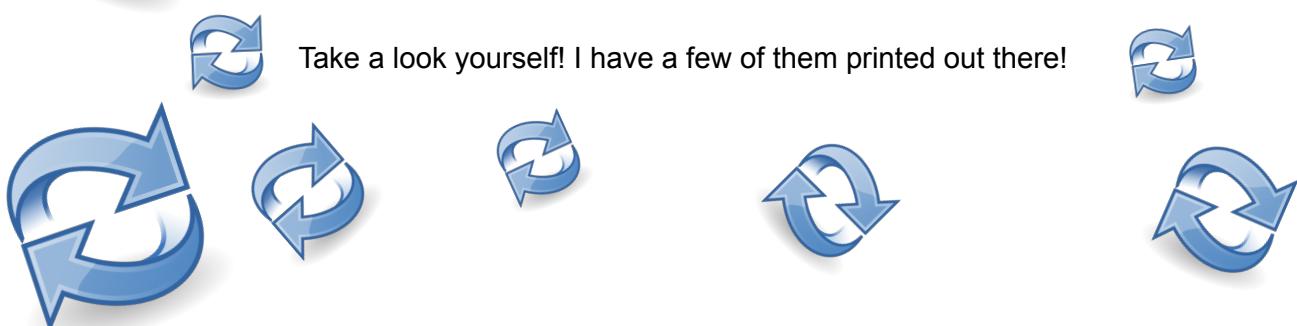
It looked very neat & I liked the colour scheme, however, it was quite boring to refresh the page and see the same result every single time! To make it more entertaining, I modified my data points to instead provide me with randomised results to my processes!

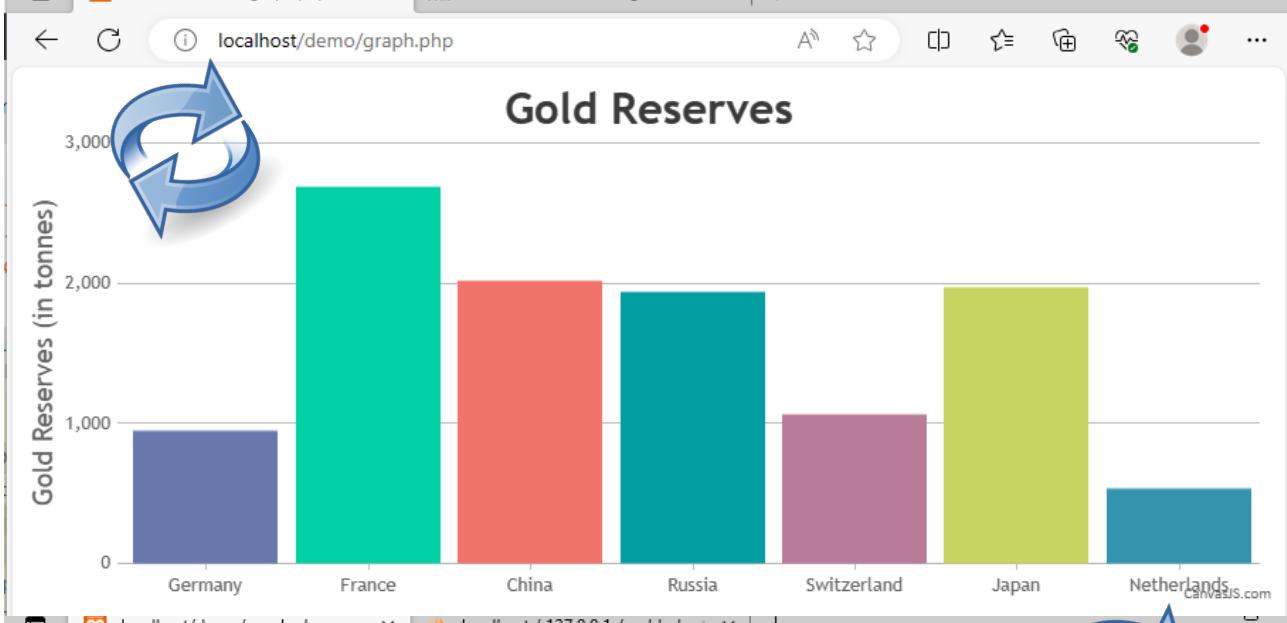
The screenshot shows a website for learning PHP. On the left, there's a sidebar with a list of PHP functions: is_infinite(), is_nan(), lcg_value(), log(), log10(), log1p(), max(), min(), mt_getrandmax(), mt_rand(), mt_srand(), octdec(), pi(), pow(), rad2deg(), and rand(). The "rand()" function is highlighted in green. The main content area has a title "PHP rand() Function" and a sub-section "PHP Math Reference". Below that is an "Example" section with the code:

```
<?php  
echo(rand());  
echo(rand());  
echo(rand(10,100));  
?>
```

 A "Try it Yourself" button is present. To the right, there's a code editor window with the following PHP code:

```
$dataPointsRandom = array(  
    array("y" => rand(123.47, 3691.43), "label" => "Germany" ),  
    array("y" => rand(1435.21, 2739.43), "label" => "France" ),  
    array("y" => rand(1942.43, 2042.23), "label" => "China" ),  
    array("y" => rand(1942.43, 2042.23), "label" => "Russia" ),  
    array("y" => rand(800.12, 1285.39), "label" => "Switzerland" ),  
    array("y" => rand(1265.25, 2017.30), "label" => "Japan" ),  
    array("y" => rand(412.453, 803.02), "label" => "Netherlands" )  
);
```





I also added a button that would Act the same way as a Refresh!



After working out this little side project, I returned back to AWS.

Finding Solutions in Mining

After a bit of online research and searching through the [AWS Elastic Beanstalk Documentation](#), I managed to find another means by which to connect to Elastic Beanstalk and that was using the [EB CLI](#). I followed the documentation and successfully installed the package `awsebcli`.

```
PS C:\xampp\htdocs\demo> pip install awsebcli
WARNING: Skipping C:\Users\Admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip-22.3.1.dist-info due to invalid metadata entry 'name'
WARNING: Skipping C:\Users\Admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip-22.3.1.dist-info due to invalid metadata entry 'name'
Requirement already satisfied: awsebcli in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (3.20.10)
Requirement already satisfied: botocore<1.32.0,>1.23.41 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from awsebcli) (1.31.85)
Requirement already satisfied: cement==2.8.2 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from awsebcli) (2.8.2)
Requirement already satisfied: colorama<0.4.4,>=0.2.5 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from awsebcli) (0.4.3)
Requirement already satisfied: pathspec==0.10.1 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages (from awsebcli) (0.10.1)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\admin\appdata\local\programs\python\python311\lib\site-packages
```

This package allows me to connect to existing environments, as well as to create new ones (on the spot). I did need to Google a couple more commands, such as the `eb deploy` command to package up anything in your directory to be sent to the AWS EB server, there was also `eb init` which was used to first step 'into' the repository for that application!

5) eu-central-1 : EU (Frankfurt)
 6) ap-south-1 : Asia Pacific (Mumbai)
 7) ap-southeast-1 : Asia Pacific (Singapore)
 8) ap-southeast-2 : Asia Pacific (Sydney)
 9) ap-northeast-1 : Asia Pacific (Tokyo)
 10) ap-northeast-2 : Asia Pacific (Seoul)
 11) sa-east-1 : South America (Sao Paulo)
 12) cn-north-1 : China (Beijing)
 13) cn-northwest-1 : China (Ningxia)
 14) us-east-2 : US East (Ohio)
 15) ca-central-1 : Canada (Central)
 16) eu-west-2 : EU (London)
 17) eu-west-3 : EU (Paris)
 18) eu-north-1 : EU (Stockholm)
 19) eu-south-1 : EU (Milano)
 20) ap-east-1 : Asia Pacific (Hong Kong)
 21) me-south-1 : Middle East (Bahrain)
 22) il-central-1 : Middle East (Israel)
 23) af-south-1 : Africa (Cape Town)
 24) ap-southeast-3 : Asia Pacific (Jakarta)
 25) ap-northeast-3 : Asia Pacific (Osaka)
 (default is 3): 4

In this screenshot, you can see the list of server locations you can go to & I chose Number 4 as that was eu-west-1 (Ireland).

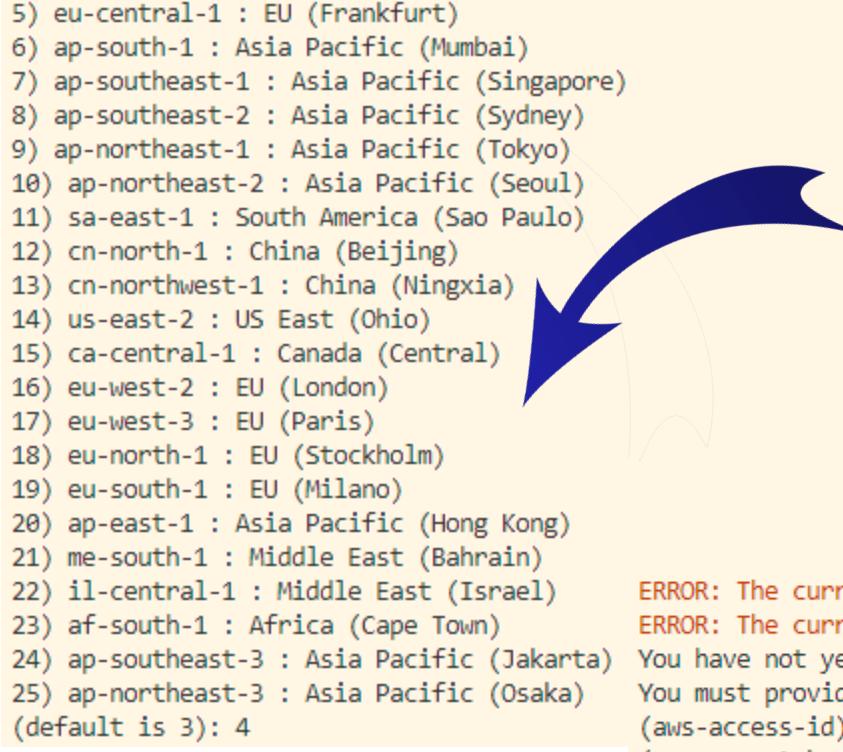
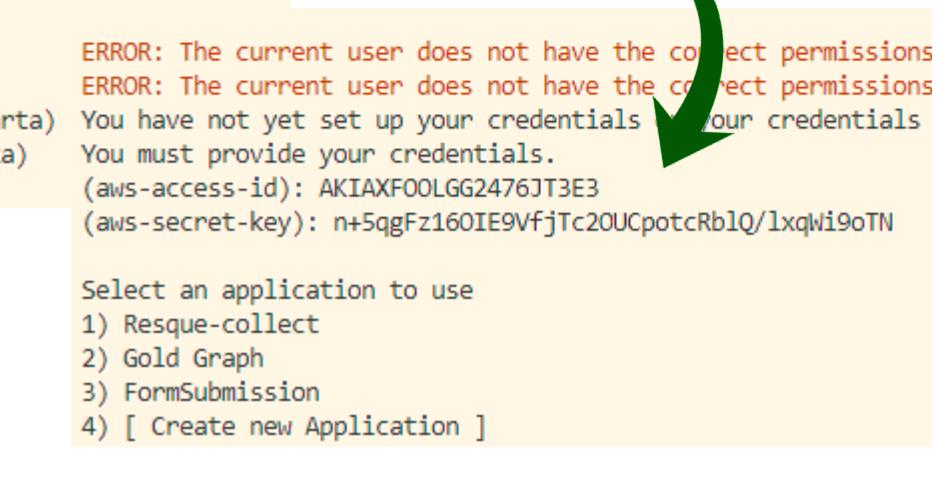
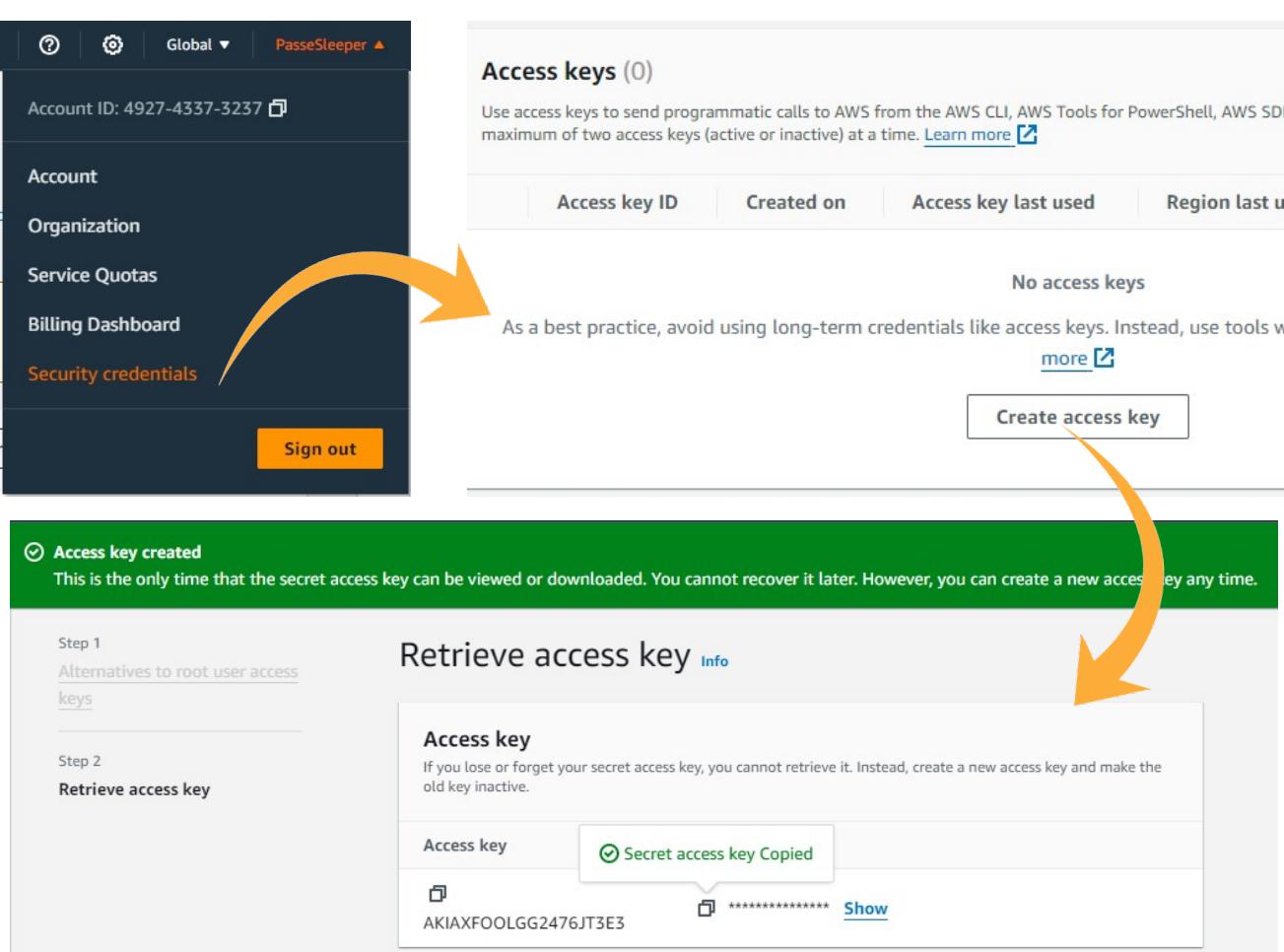
After connecting, you were required to input an aws-access-id & aws-secret-key.

ERROR: The current user does not have the correct permissions
 ERROR: The current user does not have the correct permissions
 You have not yet set up your credentials
 You must provide your credentials.
 (aws-access-id): AKIAXFOOLGG2476JT3E3
 (aws-secret-key): n+5qgFz160IE9VfjTc20UCpotcRblQ/lxqWi9oTN

Select an application to use

- 1) Resque-collect
- 2) Gold Graph
- 3) FormSubmission
- 4) [Create new Application]

Those can be obtained by following the steps (can also be found [here](#)):

The screenshots show the process of selecting a server location (eu-west-1), creating an access key, and retrieving the secret access key.

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or directly via the AWS API. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

Access key ID	Created on	Access key last used	Region last used
No access keys			

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide temporary credentials. [more](#)

[Create access key](#)

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
[Alternatives to root user access keys](#)

Step 2
Retrieve access key

Access key
 If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key (Copied)

AKIAXFOOLGG2476JT3E3

Show

The AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a [Learn more](#)

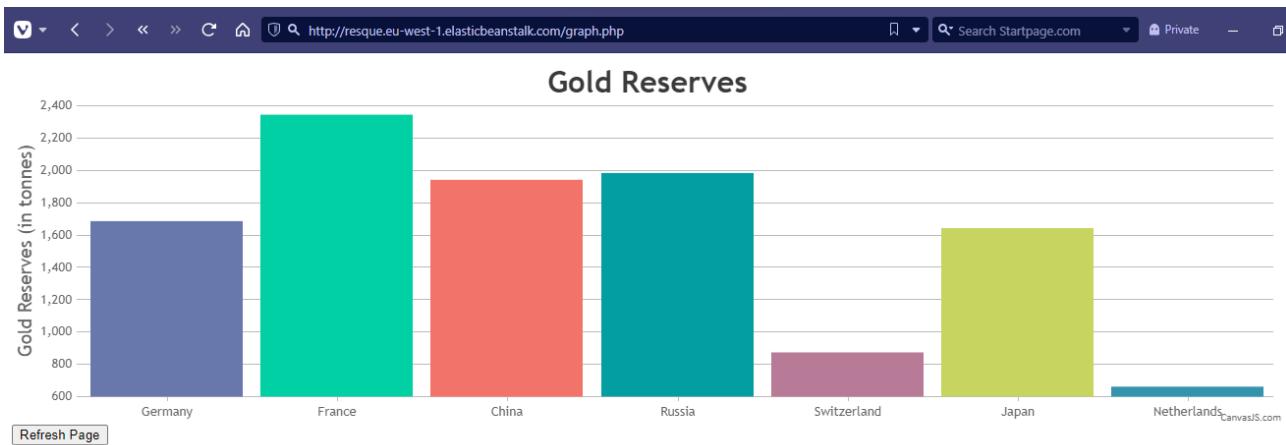
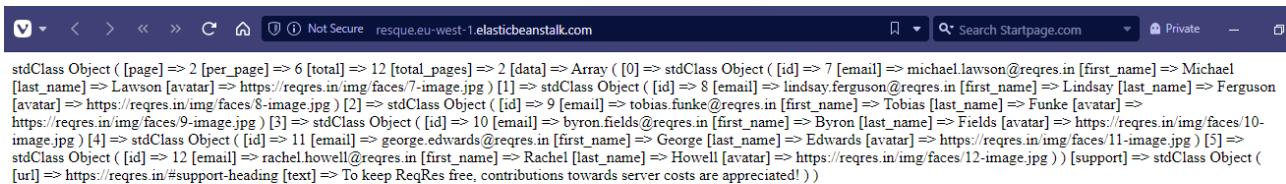
Access key last used	Region last used	Service last used	Status
3 minutes ago	eu-west-1	elasticbeanstalk	Active

If it was created successfully, after using it for the 1st time, the last `Access key last used` should be visible!



Hello World

I tried this out on the generic “Hello World” approach first, before plugging in my Gold Reserve & Resque PHP files. But just like that, everything worked successfully!



Task 2: Cloud Web Database

Data Reserves

To accomplish the task of connecting and making use of a database in my lab work, I decided to use my Gold Reserve website instead of the Resque form. Originally, the Resque form website was supposed to be a Form Submission page that upon completion gets automatically added to a Database, but In the limited time frame, I decided to build something simpler. I decided to repurpose my Gold Reserve website to allow a User to add & remove points from a graph, they should also be capable of naming the point.

In order to accomplish this, I created more supplementary webpages for the Gold Reserve site.

I had the Main Page.



I had the Database Page.

```
id: 1 - Name: A - Amount: 100
id: 2 - Name: B - Amount: 200
id: 3 - Name: C - Amount: 250
id: 4 - Name: D - Amount: 300
id: 5 - Name: E - Amount: 700
```

Add to Database Remove from Database Return to Graph

The Add Entry Page & Remove Entry Page.

Add Data Point

Label of Data Point

Value

Submit

Remove Data Point

Label of Data Point

Submit

I also set-up and connected to the Database using the previously created `admin` and `password`. To find the hostname, I had to go to RDS and find it there.

The screenshot shows the AWS Management Console interface. On the left, a sidebar lists various services under 'Recently visited' and 'Favorites'. A yellow arrow points from the 'RDS' link in the sidebar to a modal window titled 'Recently visited' which displays the 'RDS' service.

The main content area shows the 'Resources' section for Amazon RDS, indicating 3 DB Instances, 0 DB Clusters, and 0 Reserved instances. Below this, the 'Connectivity & security' tab is selected, showing detailed configuration for an endpoint:

Endpoint & port	Networking	Security
Endpoint awseb-e-22krcvmr22-stack-awsebrdsdatabase-fssjzs2lfxvd.cg8j0mlhknbk.eu-west-1.rds.amazonaws.com	Availability Zone eu-west-1a VPC vpc-09e637b082d10ad6a	VPC security groups rds-awseb-e-22krcvmr22-stack-awsebrdsdbsecuritygroup-18r7i3aia49ka-svh (sg-03a42b1468ad7fdb5) <input checked="" type="checkbox"/> Active
Port 3306	Subnet group default Subnets subnet-0b949a5d65e4d8177 subnet-0656063ab6e3a8b35 subnet-01dc80795f92907	Publicly accessible Yes Certificate authority Info rds-ca-2019

A note at the bottom left of the 'Connectivity & security' section states: 'NOTE: None of these links work as the account has been deleted.'

After collecting all the information, I was able to connect the AWS Database & display my website.



Conclusions & Experiences Gained

Throughout this Lab Assignment, I was given ample opportunity to work on new things while developing my cloud based web application. One such thing was the choice of programming language & I took the chance to learn a new one - PHP!

I am thankful for this module for allowing me this occasion. It was fun, sometimes frustrating, but a very unique experience. I loved that I could use the knowledge gained from the module to help me in understanding why I was doing what I was doing, and how AWS & Cloud Services in general help improve the lives of other people!

The most difficult parts of this assignment were definitely anything related to working on the AWS platform, as it does not provide enough material to begin right away, instead I was forced to research the proper means by which to upload my files (when using PHP, the default file is ALWAYS index.php), how to actually connect with AWS Database (very complicated for a newbie!).

I can't wait to begin using this type of code in my future, I might even build a website of my own someday!