

CS3204 | Cloud Infrastructure and Services

Lab2: 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 dubbing new software

Goals & Objectives

My goal for this 2ish 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/Lab02>



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:

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.



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.



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.

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

Overall, AWS offers a very wide branched 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 websites & 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...

The screenshot shows the AWS Sign up for AWS page. The left sidebar has a 'Secure verification' section with a shield icon and a message: 'We will not charge you for usage below AWS Free Tier limits. We may temporarily hold up to \$1 USD (or an equivalent amount in local currency) as a pending transaction for 3-5 days to verify your identity.' The main content area is titled 'Sign up for AWS' and contains two sections: 'Billing Information' and 'Select a support plan'.

Billing Information

Credit or Debit card number
[Redacted]

VISA MASTERCARD AMEX DISCOVER

AWS accepts all major credit and debit cards. To learn more about payment options, review our [FAQ](#)

Expiration date
[Redacted] [Redacted]

Security code [icon]
[Redacted]

No security code needed at this time.

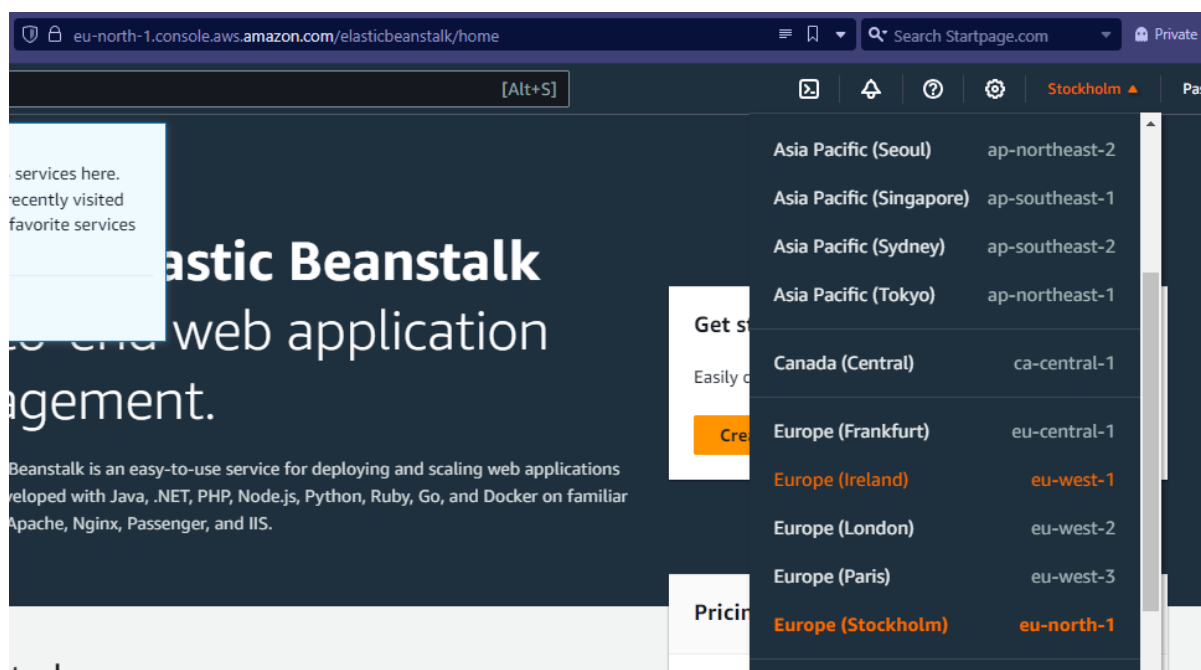
Cardholder's name
[Redacted]

Select a support plan

Choose a support plan for your business or personal account. [Compare plans and pricing examples](#)
[icon] You can change your plan anytime in the AWS Management Console.

- ☒ **Basic support - Free**
 - Recommended for new users just getting started with AWS
 - 24x7 self-service access to AWS resources
 - For account and billing issues only
 - Access to Personal Health Dashboard & Trusted Advisor
- ☐ **Developer support - From \$29/month**
 - Recommended for developers experimenting with AWS
 - Email access to AWS Support during business hours
 - 12 (business)-hour response times
- ☐ **Business support - From \$100/month**
 - Recommended for running production workloads on AWS
 - 24x7 tech support via email, phone, and chat
 - 1-hour response times
 - Full set of Trusted Advisor best-practice recommendations

Then I continued to follow the same procedures as stated in the notes..



Services

Search

[Alt+S]

Search Startpage.com

Private

Compute

Amazon Elastic Beanstalk

End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Create application

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. [Learn more](#)

Application code [Info](#)

- ☐ 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.


Version label


Unique name for this version of your application code.

Source code origin. Maximum size 500 MB

- ☒ Local file

Upload application

 Choose file

 File name: **graph.php**

File must be less than 500MB max file size

- ☐ Public S3 URL

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

Engine version

8.0.33

Instance class

db.t2.micro

Storage

Choose a number between 5 GB and 1024 GB.

5

GB

Username

dbadmin

Password

.....

Availability

Low (one AZ)

Review [Info](#)

Step 1: Configure environment

Edit

Environment information

Environment tier	Application name
Web server environment	Gold Graph
Environment name	Application code
GoldGraph-env	graph.php
Platform	
arn:aws:elasticbeanstalk:eu-west-1::platform/PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3	

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

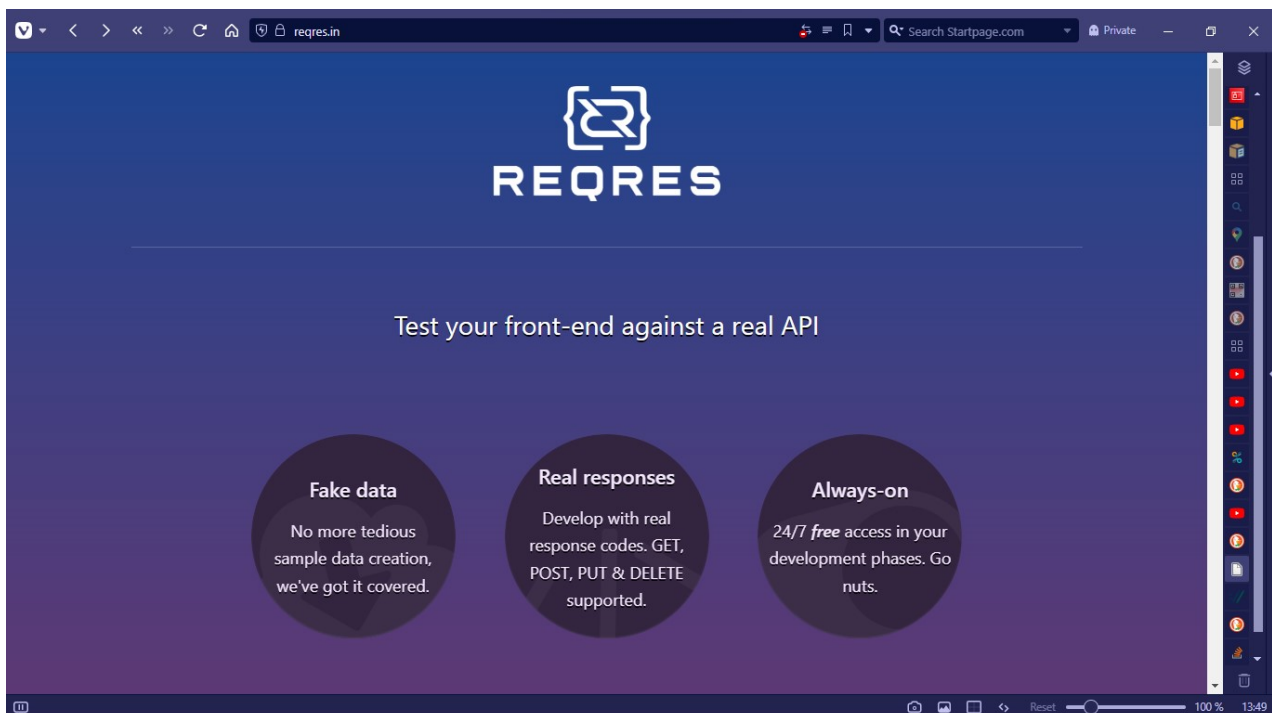
Database availability	Has coupled database	Database deletion policy
false	true	Delete
Database engine	Database engine version	Database instance class
mysql	8.0.33	db.t2.micro
Database password	Database storage	Database username
*****	5	dbadmin

Tags

Key	Value
No tags	
There are no tags defined	

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 follow 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 others had done too!



Browser screenshot showing the URL `https://reqres.in/api/users?page=2` and the JSON response:

```
{
  "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!"
  }
}
```

Code editor showing the PHP script `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
12 if ($e = curl_error($ch)) {
13     echo $e;
14 }
15 else {
16     $decoded = json_decode($resp);
17     print_r($decoded);
18 }
19
20 curl_close($ch);
21
22 ?>
```



Using XAMPP, I was able to test on the local host before running it on the Cloud.

Browser screenshot showing the local host URL `localhost/demo/fetch.php` and the output of the PHP script:

```
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! ) )
```

However, when attempting to run it on the AWS servers, I encountered a lot of errors in relation to the file health being Degraded cloud

Environment successfully launched.

Successfully uploaded file fetch.php to S3, created application version and started deployment with new application version

Environment update successfully completed.

Elastic Beanstalk > Environments > FormSubmission-env

FormSubmission-env Info

Environment overview

Health

Degraded - View causes

Domain

phpbasedformsubmission.eu-west-1.elasticbeanstalk.com

Environment ID

e-2xiejykrj5

Application name

FormSubmission

Platform

Change version

Platform

PHP 8.2 running on 64bit Amazon Linux 2023/4.0.3

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
	Severe
<div><div></div><div>i-022e6798e2...</div></div>	<div><div>Application deployment failed at 2023-11-20T14:03:08Z with exit status 1 and error: Engine execution has encountered an error.</div><div>Incorrect application version "FormSubmission-version-2" (deployment 3). Expected version "Sample" (deployment 1).</div></div>

I recreated the Environment and this time managed to bring down the Health Degradation to just a Warning status.




Environment successfully launched.

[Elastic Beanstalk](#) > [Environments](#) > GoldGraph-env

GoldGraph-env Info


 Actions ▾ Upload and deploy

Environment overview

Health  Warning	Environment ID  e-ghcrmxiea3
Domain goldgraph.eu-west-1.elasticbeanstalk.com 	Application name Gold Graph

Platform

Change version

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

Unfortunately, when attempted to access my website I got an error stating:

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 returned back to the LocalHost Apache server and continued working on my Project

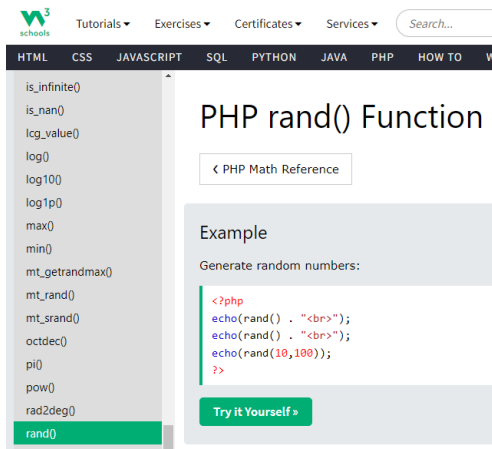
Finding Salvation in Mining

As AWS had given up on me, I decided to modify my website and make it more interactive and a bit more interesting to look at each time!

Initially I had built a static model using [Canvas.js](#) - a very handy tool for creating simple graphs for websites! This was the result:



It looked very neat & I liked the Result, 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!



```
$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" )
);
```

This allowed me to refresh the page and get a different result every time!

Take a look yourself! I have a few of them printed out there!

S-----

