James Witts - S14139866
Ahmed Ismail - S15120305
Cheng Jiacheng - S16153760

CMP6203 - Cloud Computing: AWS Report

This report describes **six** main stages in a project using *Amazon Web Services (AWS)* from initial development on a local machine through deployment, testing and concluding with some remarks for further development

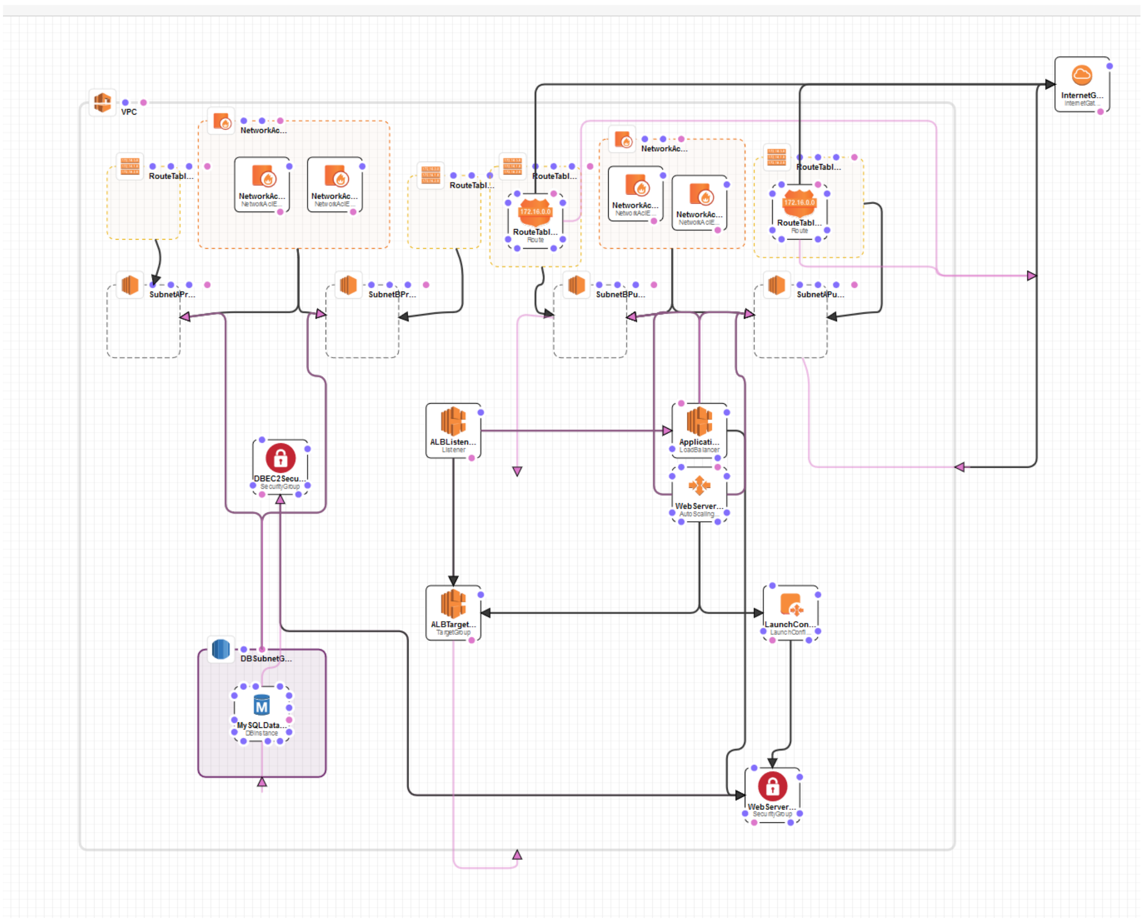
STAGE ONE: Local development

To start with we developed our application on a windows 8 laptop using XAMPP Lamp stack from Bitnami that uses a MySQL database running on port 3306. The application is a php web app that allows a user to insert data into one of three different tables; users, bookmarks or tags. The default engine for the MySQL database is MyISAM. In the future it might be better to changes this to Innodb since the “tags” table we created uses a foreign key dependent on user data, which will be replaced in Innodb by a unique identifier when running from an Amazon RDS MySQL database.

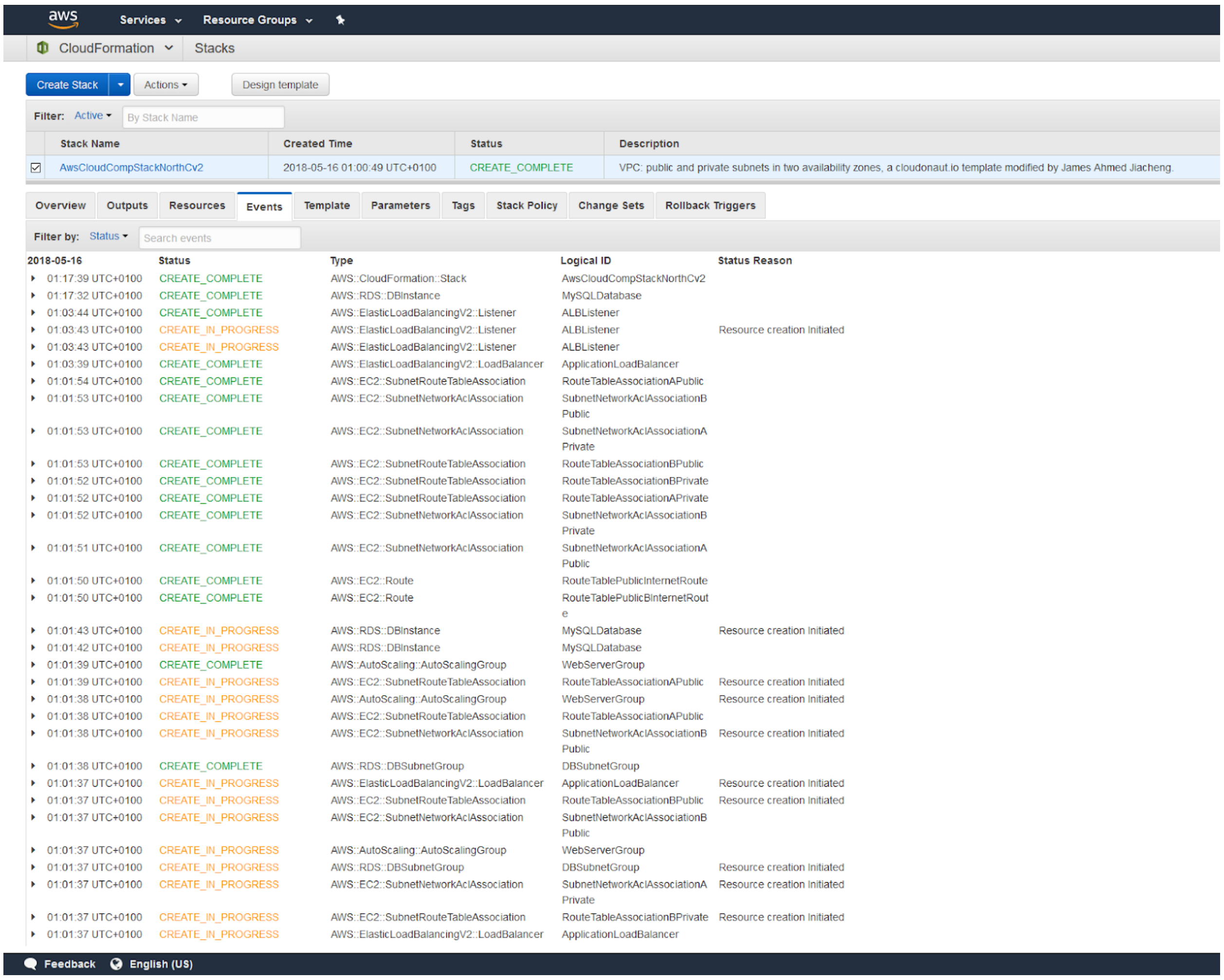
STAGE TWO: Deployment - North California region

When moving the XAMPP stack to Amazon we first prepared all the resources we needed by using a CloudFormation stack template which we made in CloudFormation Designer by merging two existing templates together from two different sources. One came from a git repository and the other a sample template provided by CloudFormation Designer. When merging the two together the workflow was mostly trial and error and we eventually got a stack that created most of the resources we needed.

A schematic overview of the full design in given in Appendix I.



The following diagram shows the output from creating a stack using the *Cloudformation* tool:



Stack Name	Created Time	Status	Description
AwsCloudCompStackNorthCv2	2018-05-16 01:00:49 UTC+0100	CREATE_COMPLETE	VPC: public and private subnets in two availability zones, a cloudonaut.io template modified by James Ahmed Jiacheng.

2018-05-16	Status	Type	Logical ID	Status Reason
01:17:39 UTC+0100	CREATE_COMPLETE	AWS::CloudFormation::Stack	AwsCloudCompStackNorthCv2	
01:17:32 UTC+0100	CREATE_COMPLETE	AWS::RDS::DBInstance	MySQLDatabase	
01:03:44 UTC+0100	CREATE_COMPLETE	AWS::ElasticLoadBalancingV2::Listener	ALBListener	
01:03:43 UTC+0100	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancingV2::Listener	ALBListener	Resource creation Initiated
01:03:43 UTC+0100	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancingV2::Listener	ALBListener	
01:03:39 UTC+0100	CREATE_COMPLETE	AWS::ElasticLoadBalancingV2::LoadBalancer	ApplicationLoadBalancer	
01:01:54 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationAPublic	
01:01:53 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationBPublic	
01:01:53 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationAPrivate	
01:01:53 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationBPublic	
01:01:52 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationBPrivate	
01:01:52 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationAPrivate	
01:01:52 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationBPrivate	
01:01:51 UTC+0100	CREATE_COMPLETE	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationAPublic	
01:01:50 UTC+0100	CREATE_COMPLETE	AWS::EC2::Route	RouteTablePublicInternetRoute	
01:01:50 UTC+0100	CREATE_COMPLETE	AWS::EC2::Route	RouteTablePublicBInternetRoute	
01:01:43 UTC+0100	CREATE_IN_PROGRESS	AWS::RDS::DBInstance	MySQLDatabase	Resource creation Initiated
01:01:42 UTC+0100	CREATE_IN_PROGRESS	AWS::RDS::DBInstance	MySQLDatabase	
01:01:39 UTC+0100	CREATE_COMPLETE	AWS::AutoScaling::AutoScalingGroup	WebServerGroup	
01:01:39 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationAPublic	Resource creation Initiated
01:01:38 UTC+0100	CREATE_IN_PROGRESS	AWS::AutoScaling::AutoScalingGroup	WebServerGroup	Resource creation Initiated
01:01:38 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationAPublic	
01:01:38 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationBPublic	Resource creation Initiated
01:01:38 UTC+0100	CREATE_COMPLETE	AWS::RDS::DBSubnetGroup	DBSubnetGroup	
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancingV2::LoadBalancer	ApplicationLoadBalancer	Resource creation Initiated
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationBPublic	Resource creation Initiated
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationBPublic	
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::AutoScaling::AutoScalingGroup	WebServerGroup	
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::RDS::DBSubnetGroup	DBSubnetGroup	Resource creation Initiated
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetNetworkAclAssociation	SubnetNetworkAclAssociationAPrivate	Resource creation Initiated
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	RouteTableAssociationBPrivate	Resource creation Initiated
01:01:37 UTC+0100	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancingV2::LoadBalancer	ApplicationLoadBalancer	

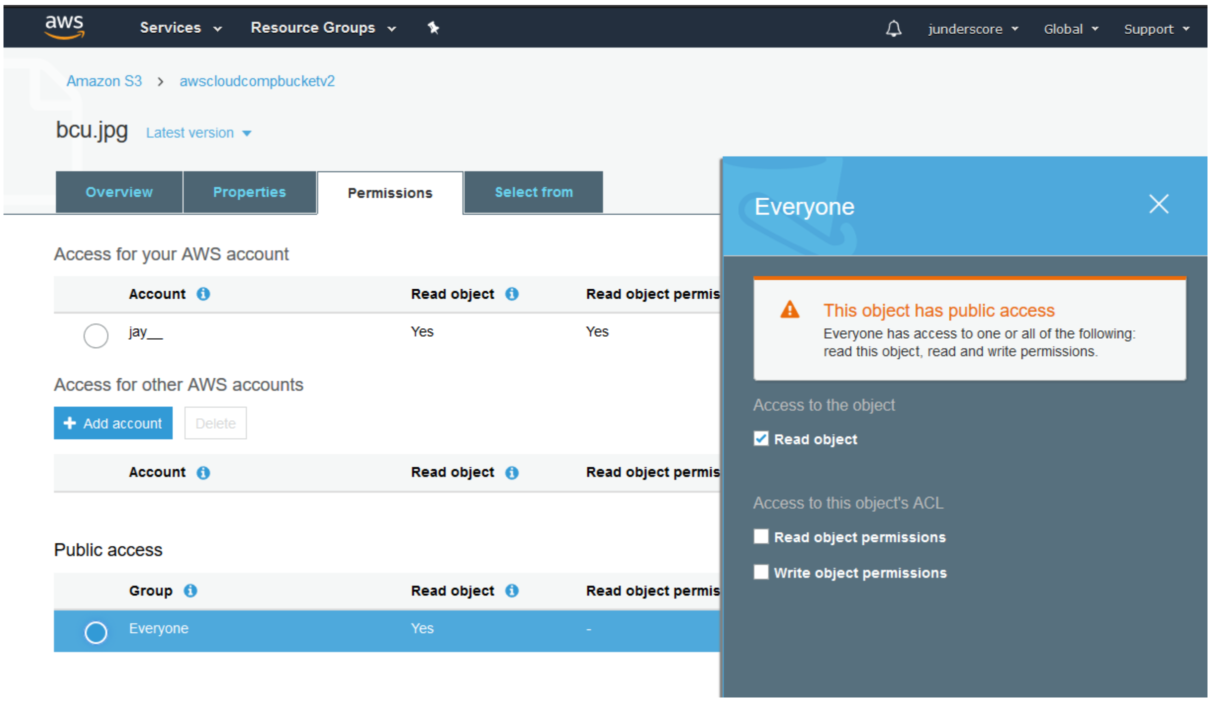
The following modifications were made after cloudformation stack completion from the *AWS console*:

Security

- Added the inbound rule on webservers from port 3306 for the DBEC2SecurityGroup
- Modified DBInstance to the DBEC2SecurityGroup and removed the default one.
- DBEC2SecurityGroup ingress rule to allow all trafic from webservers. (to get it to work)

S3 storage

- Created an s3 bucket in the current region
- Moved the "bcu.jpg" into the S3 bucket.
- Gave read-only permissions for the public for the "bcu.jpg" image.



Access for your AWS account

Account	Read object	Read object permis
jay__	Yes	Yes

Access for other AWS accounts

Add account Delete

Account	Read object	Read object permis

Public access

Group	Read object	Read object permis
Everyone	Yes	-

Everyone

This object has public access
Everyone has access to one or all of the following: read this object, read and write permissions.

Access to the object

[✓] Read object

Access to this object's ACL

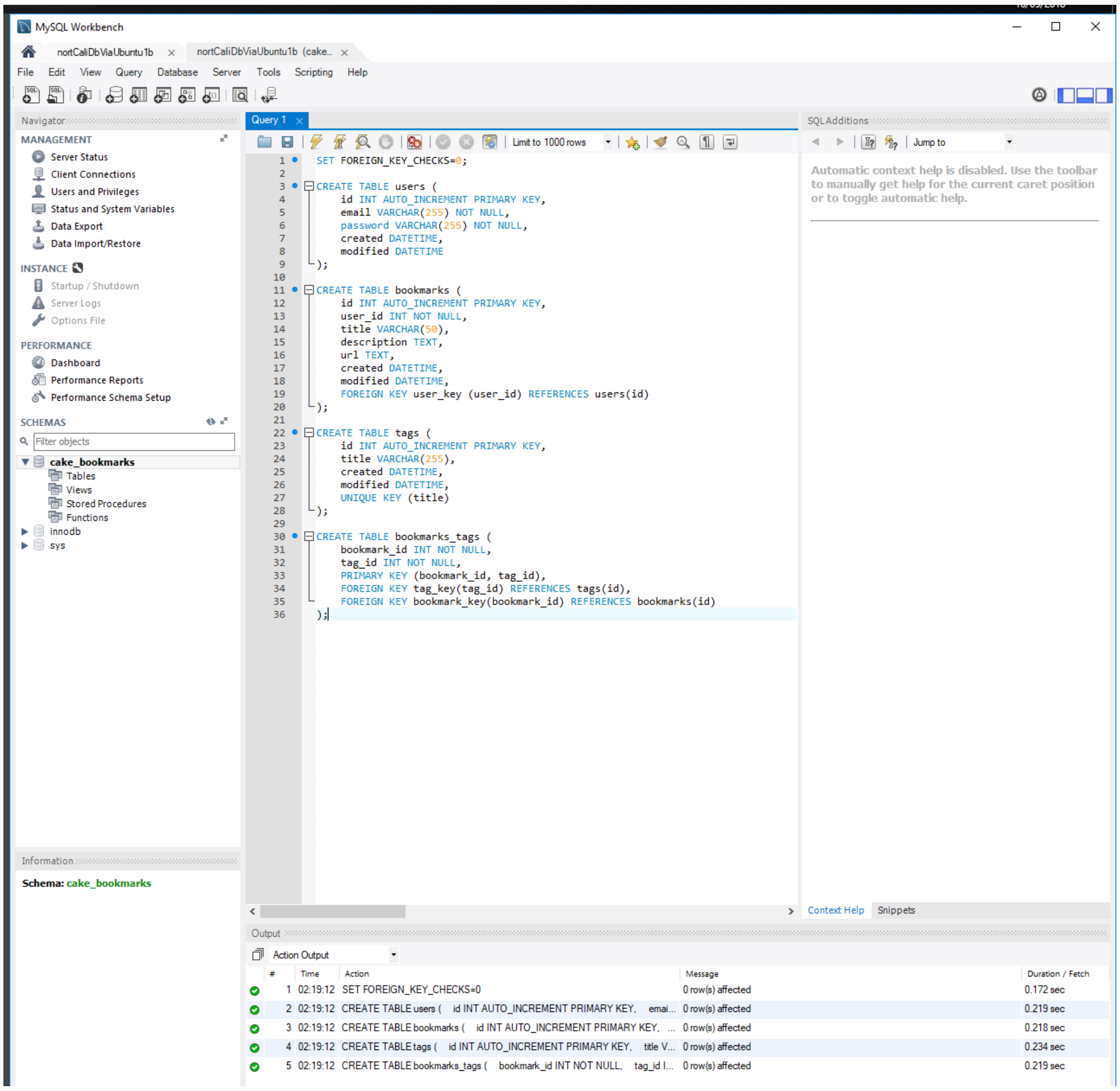
[ ] Read object permissions

[ ] Write object permissions

Illustration 1: Screenshot of S3 Storage Interface

MySQL scripting using Workbench

- Created a connection to the database
- Created the tables with the "cake_bookmarks_createTables.sql" file.



```sql
SET FOREIGN_KEY_CHECKS=0;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    created DATETIME,
    modified DATETIME
);

CREATE TABLE bookmarks (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    title VARCHAR(50),
    description TEXT,
    url TEXT,
    created DATETIME,
    modified DATETIME,
    FOREIGN KEY user_key (user_id) REFERENCES users(id)
);

CREATE TABLE tags (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255),
    created DATETIME,
    modified DATETIME,
    UNIQUE KEY (title)
);

CREATE TABLE bookmarks_tags (
    bookmark_id INT NOT NULL,
    tag_id INT NOT NULL,
    PRIMARY KEY (bookmark_id, tag_id),
    FOREIGN KEY tag_key(tag_id) REFERENCES tags(id),
    FOREIGN KEY bookmark_key(bookmark_id) REFERENCES bookmarks(id)
);
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

	#	Time	Action	Message	Duration / Fetch
✓	1	02:19:12	SET FOREIGN_KEY_CHECKS=0	0 row(s) affected	0.172 sec
✓	2	02:19:12	CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, emai...	0 row(s) affected	0.219 sec
✓	3	02:19:12	CREATE TABLE bookmarks (id INT AUTO_INCREMENT PRIMARY KEY, ...	0 row(s) affected	0.218 sec
✓	4	02:19:12	CREATE TABLE tags (id INT AUTO_INCREMENT PRIMARY KEY, title V...	0 row(s) affected	0.234 sec
✓	5	02:19:12	CREATE TABLE bookmarks_tags (bookmark_id INT NOT NULL, tag_id I...	0 row(s) affected	0.219 sec

Illustration 2: Screenshot of MySQL Workbench Interface

Configured Webserver using WinSCP and Putty

This procedure was carried out for both ubuntu EC2 instances that were created:

- Connect via ssh client and execute the following

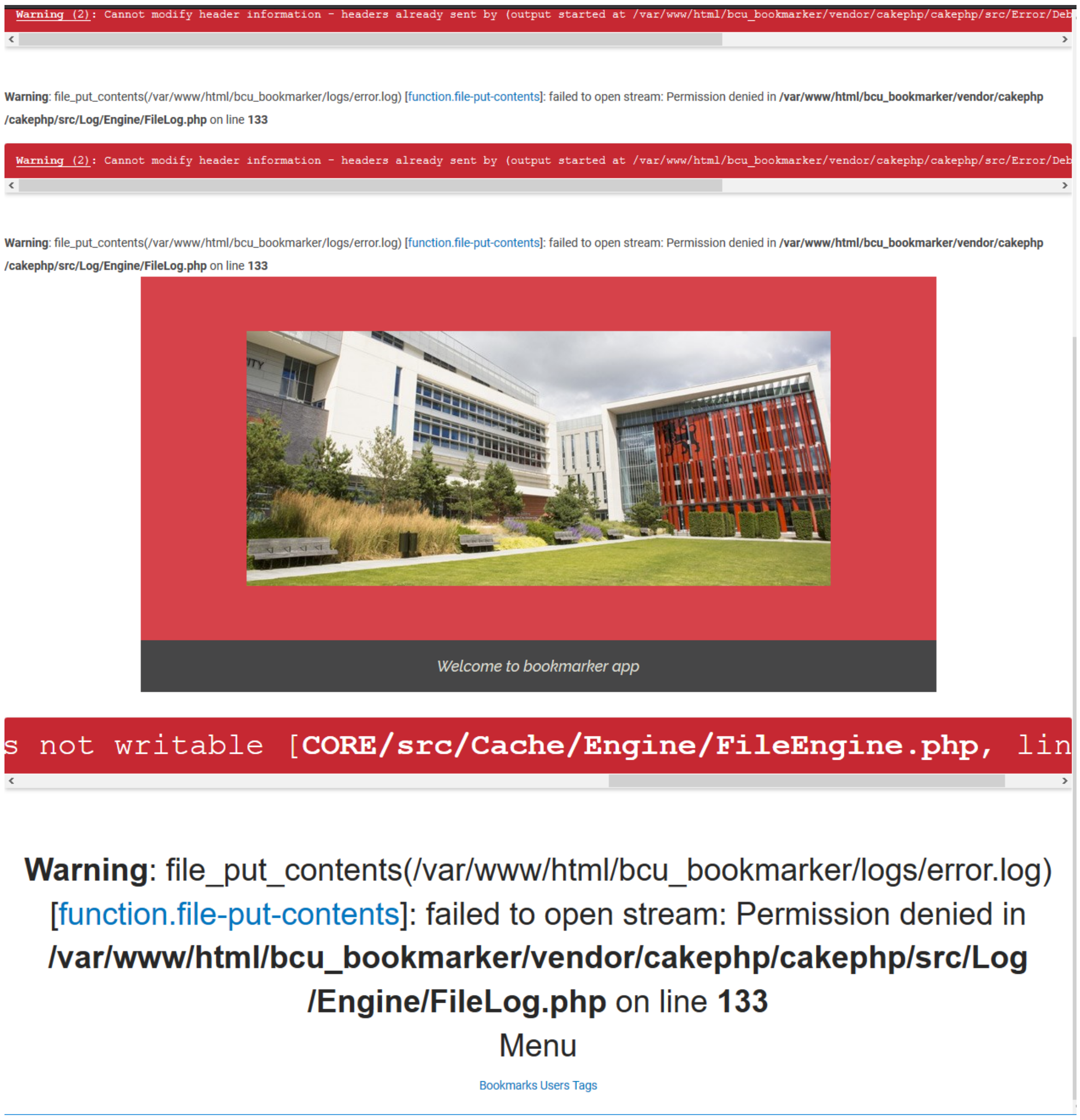
```
sudo su
apt-get update && apt-get upgrade -y
usermod -a -G root ubuntu
chown -R ubuntu:ubuntu /var/www/html
chmod -R 755 /var/www/html
a2enmod rewrite
sudo nano /etc/apache2/sites-available/000-default.conf
```

- Added to webserver configuration file:

```
<VirtualHost *:80>
     <Directory /var/www/html>
     Options Indexes FollowSymLinks MultiViews
     AllowOverride All
     Require all granted
     </Directory>
     . . .
</VirtualHost>
```

- Modified *document root*
  `/var/www/html/bcu_bookmarker`
  <<Ctrl+o>> then <<Ctrl+x>> to save the changes.

- *`sudo nano /var/www/html/.htaccess`*
  `RewriteEngine on`
  <<Ctrl+o>> then <<Ctrl+x>> to save the changes.
  `systemctl restart apache2`
- Dragged and dropped "bcu_bookmarker" client application into `/var/www/html` folder. (Long wait)



```
Warning (2): Cannot modify header information - headers already sent by (output started at /var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Error/Deb
```

Warning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in **/var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Log/Engine/FileLog.php** on line **133**

```
Warning (2): Cannot modify header information - headers already sent by (output started at /var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Error/Deb
```

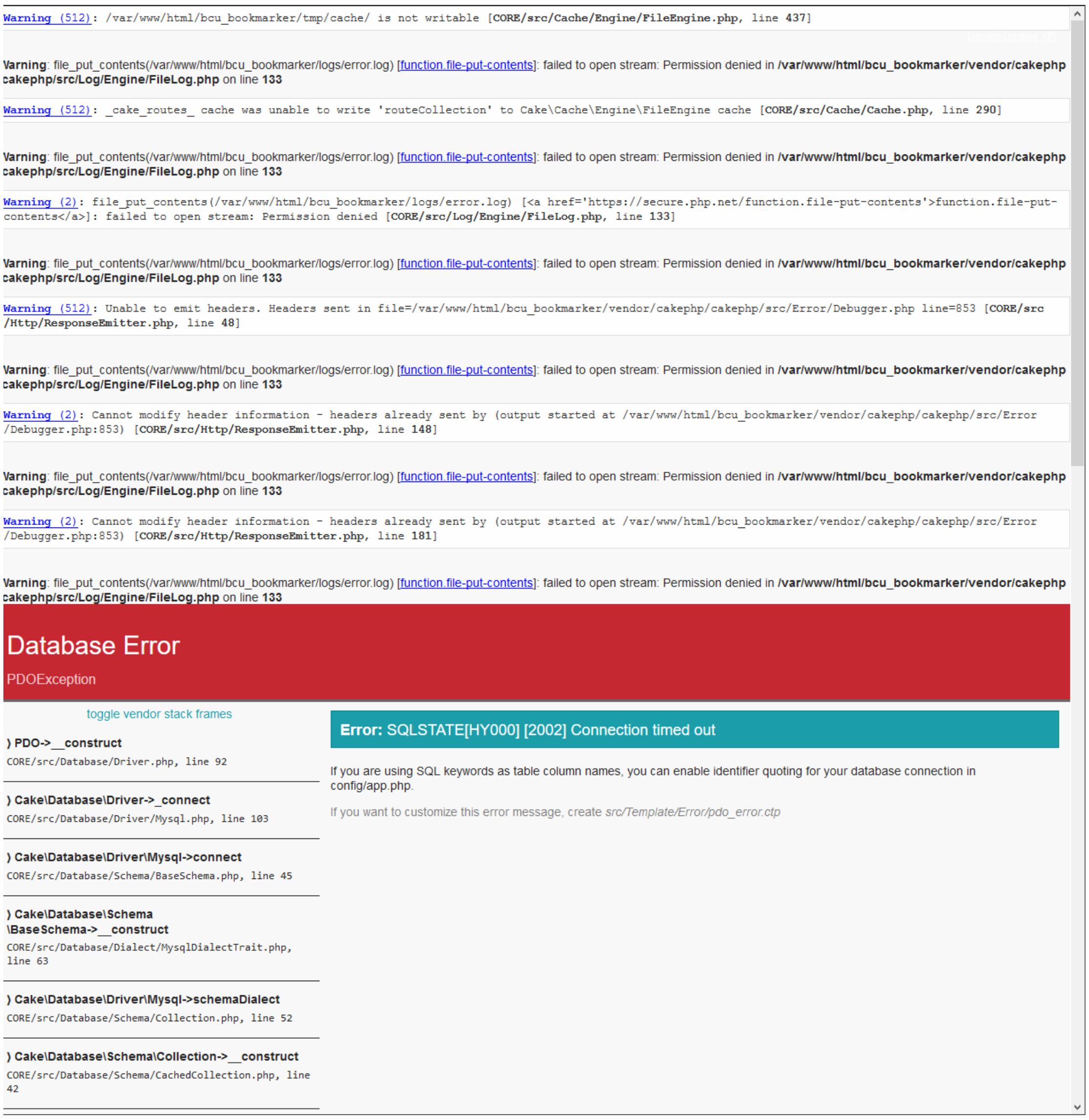
Warning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in **/var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Log/Engine/FileLog.php** on line **133**

Welcome to bookmarker app

```
s not writable [CORE/src/Cache/Engine/FileEngine.php, lin
```

Warning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in **/var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Log/Engine/FileLog.php** on line **133**

Illustration 3: Screenshot of Web App in error reporting mode



```
Warning (512): /var/www/html/bcu_bookmarker/tmp/cache/ is not writable [CORE/src/Cache/Engine/FileEngine.php, line 437]

Varning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in /var/www/html/bcu_bookmarker/vendor/cakephp
cakephp/src/Log/Engine/FileLog.php on line 133

Warning (512): _cake_routes_ cache was unable to write 'routeCollection' to Cake\Cache\Engine\FileEngine cache [CORE/src/Cache/Cache.php, line 290]

Varning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in /var/www/html/bcu_bookmarker/vendor/cakephp
cakephp/src/Log/Engine/FileLog.php on line 133

Warning (2): file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [<a href='https://secure.php.net/function.file-put-contents'>function.file-put-
contents</a>]: failed to open stream: Permission denied [CORE/src/Log/Engine/FileLog.php, line 133]

Varning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in /var/www/html/bcu_bookmarker/vendor/cakephp
cakephp/src/Log/Engine/FileLog.php on line 133

Warning (512): Unable to emit headers. Headers sent in file=/var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Error/Debugger.php line=853 [CORE/src
/Http/ResponseEmitter.php, line 48]

Varning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in /var/www/html/bcu_bookmarker/vendor/cakephp
cakephp/src/Log/Engine/FileLog.php on line 133

Warning (2): Cannot modify header information - headers already sent by (output started at /var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Error
/Debugger.php:853) [CORE/src/Http/ResponseEmitter.php, line 148]

Varning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in /var/www/html/bcu_bookmarker/vendor/cakephp
cakephp/src/Log/Engine/FileLog.php on line 133

Warning (2): Cannot modify header information - headers already sent by (output started at /var/www/html/bcu_bookmarker/vendor/cakephp/cakephp/src/Error
/Debugger.php:853) [CORE/src/Http/ResponseEmitter.php, line 181]

Varning: file_put_contents(/var/www/html/bcu_bookmarker/logs/error.log) [function.file-put-contents]: failed to open stream: Permission denied in /var/www/html/bcu_bookmarker/vendor/cakephp
cakephp/src/Log/Engine/FileLog.php on line 133

Database Error
PDOException

toggle vendor stack frames

) PDO->__construct
CORE/src/Database/Driver.php, line 92

) Cake\Database\Driver->_connect
CORE/src/Database/Driver/Mysql.php, line 103

) Cake\Database\Driver\Mysql->connect
CORE/src/Database/Schema/BaseSchema.php, line 45

) Cake\Database\Schema
\BaseSchema->__construct
CORE/src/Database/Dialect/MysqlDialectTrait.php,
line 63

) Cake\Database\Driver\Mysql->schemaDialect
CORE/src/Database/Schema/Collection.php, line 52

) Cake\Database\Schema\Collection->__construct
CORE/src/Database/Schema/CachedCollection.php, line
42

Error: SQLSTATE[HY000] [2002] Connection timed out

If you are using SQL keywords as table column names, you can enable identifier quoting for your database connection in
config/app.php.

If you want to customize this error message, create src/Template/Error/pdo_error.ctp
```

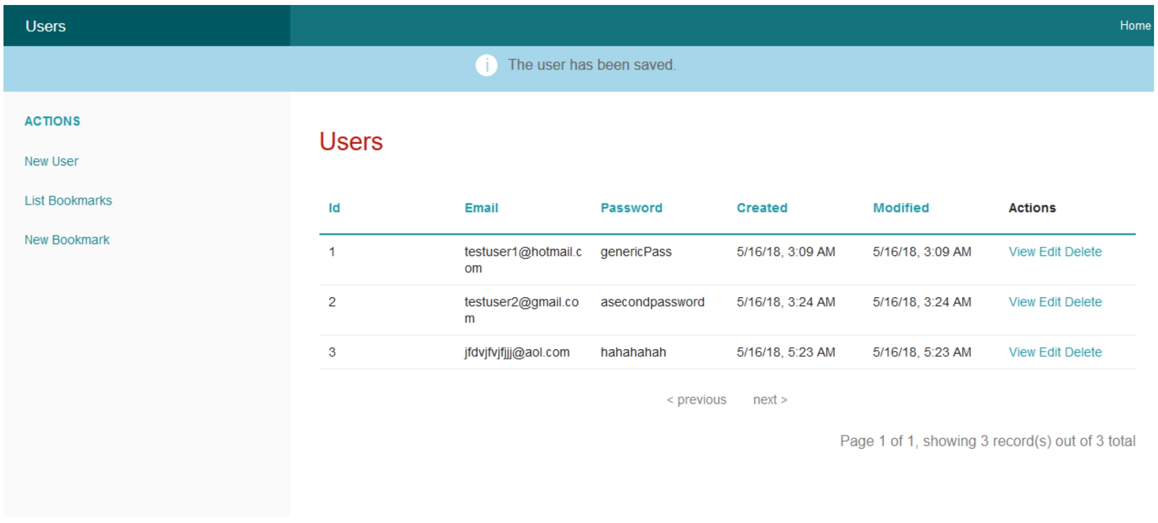
Illustration 4: More errors reported during debugging

In app `config`

- We set the `debug` variable to `false` to hide the errors
- Entered the db host as the rds endpoint.
  - `amtzcbah63q2ln.ccot9al0nlqj.us-west-1.rds.amazonaws.com`
- Modified `security salt` variable
- In the `home.ctp` we linked to s3 bucket in home.ctp img tag src:
  - https://s3-us-west-1.amazonaws.com/awscloudcompbucketv2/bcu.jpg

This completed the initial deployment of the app.

The next step was to **populate database with test data:**



Users

Home

The user has been saved.

ACTIONS

New User

List Bookmarks

New Bookmark

Users

Id	Email	Password	Created	Modified	Actions
1	testuser1@hotmail.com	genericPass	5/16/18, 3:09 AM	5/16/18, 3:09 AM	View Edit Delete
2	testuser2@gmail.com	asecondpassword	5/16/18, 3:24 AM	5/16/18, 3:24 AM	View Edit Delete
3	jfdvjfvjfjjj@aol.com	hahahahah	5/16/18, 5:23 AM	5/16/18, 5:23 AM	View Edit Delete

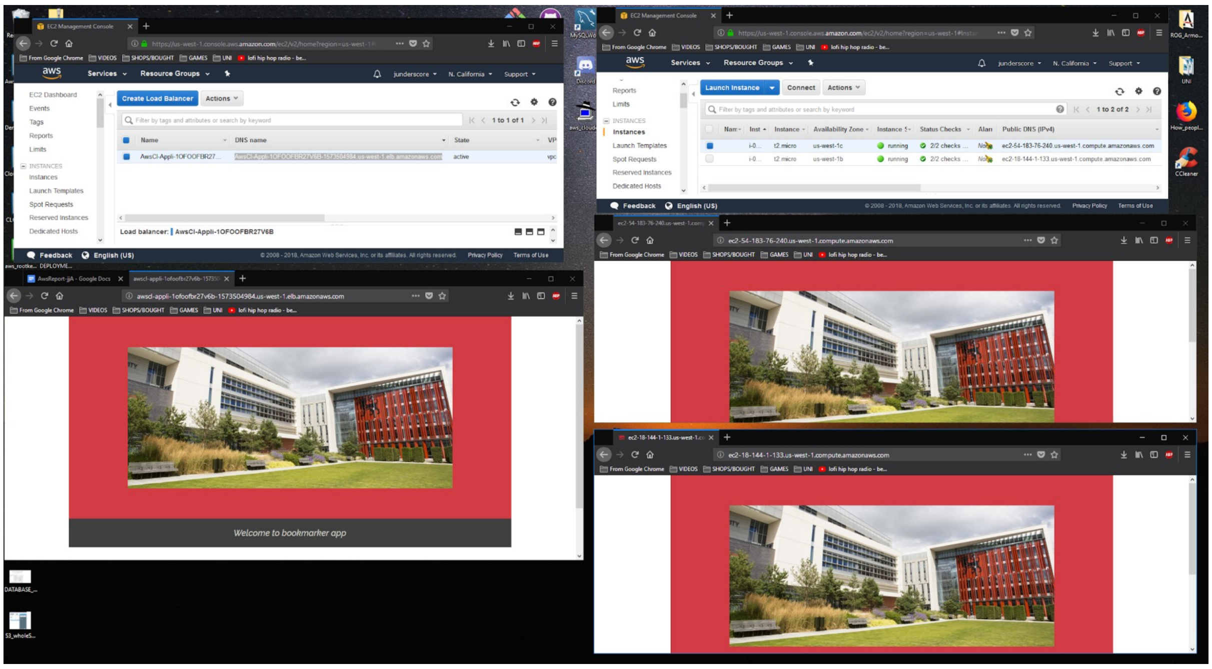
< previous next >

Page 1 of 1, showing 3 record(s) out of 3 total

Illustration 5: Screenshot showing test data has been added

STAGE THREE: Load balancing tests (North California)

We have two Ubuntu EC2 instances on the right side of the screen running in N.Cali region, amazon zone 1b and 1c. Both of them can be accessed from the load balancer on the left side of the screen.



EC2 Management Console

https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1

Name	DNS name	State
AwsCl-Appli-1OFOOFBR27...	AwsCl-Appli-1OFOOFBR27V6B-1573504984.us-west-1.elb.amazonaws.com	active

Load balancer: AwsCl-Appli-1OFOOFBR27V6B

awscl-appli-1ofoofbr27v6b-1573504984.us-west-1.elb.amazonaws.com

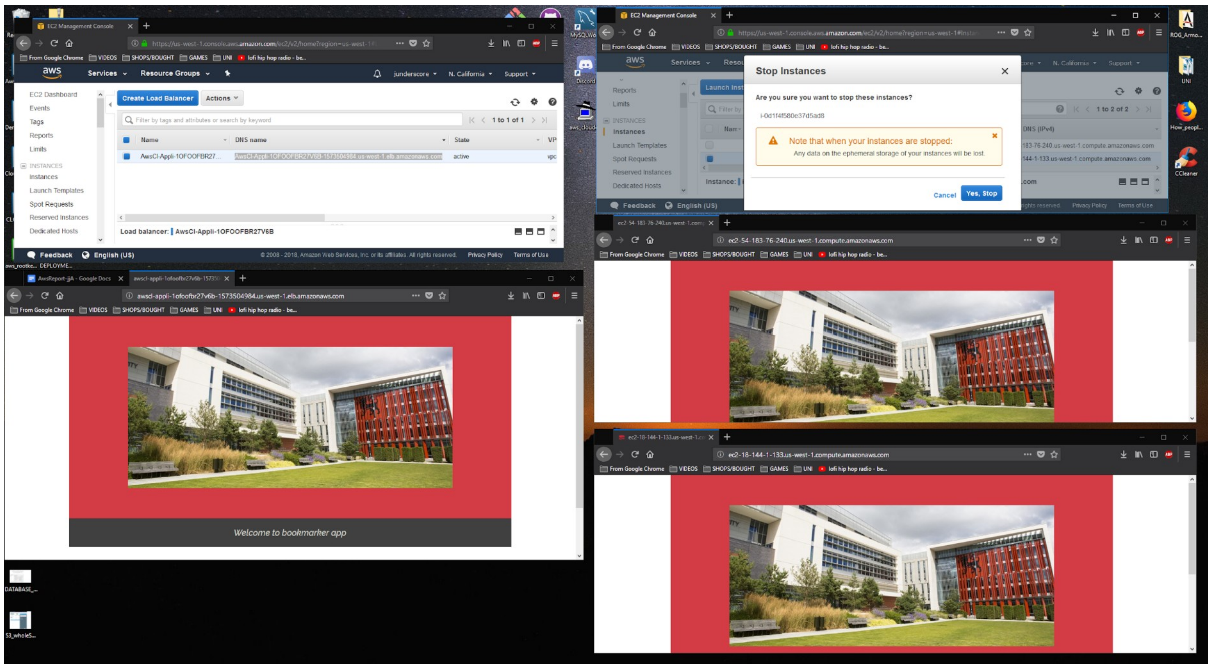
Welcome to bookmarker app

Inst	Instance	Availability Zone	Instance S	Status Checks	Alarm	Public DNS (IPv4)
i-0...	t2.micro	us-west-1c	running	2/2 checks ...	None	ec2-54-183-76-240.us-west-1.compute.amazonaws.com
i-0...	t2.micro	us-west-1b	running	2/2 checks ...	None	ec2-18-144-1-133.us-west-1.compute.amazonaws.com

ec2-54-183-76-240.us-west-1.compute.amazonaws.com

ec2-18-144-1-133.us-west-1.compute.amazonaws.com

Stopping 1b gives us this:



Name	DNS name	State
AwsCl-Appl-1OFOOFBR27...	AwsCl-Appl-1OFOOFBR27V6B-1573504984.us-west-1.elb.amazonaws.com	active

Load balancer: AwsCl-Appl-1OFOOFBR27V6B

awsd-appl-1ofoofbr27v6b-1573504984.us-west-1.elb.amazonaws.com

Welcome to bookmarker app

Stop Instances

Are you sure you want to stop these instances?

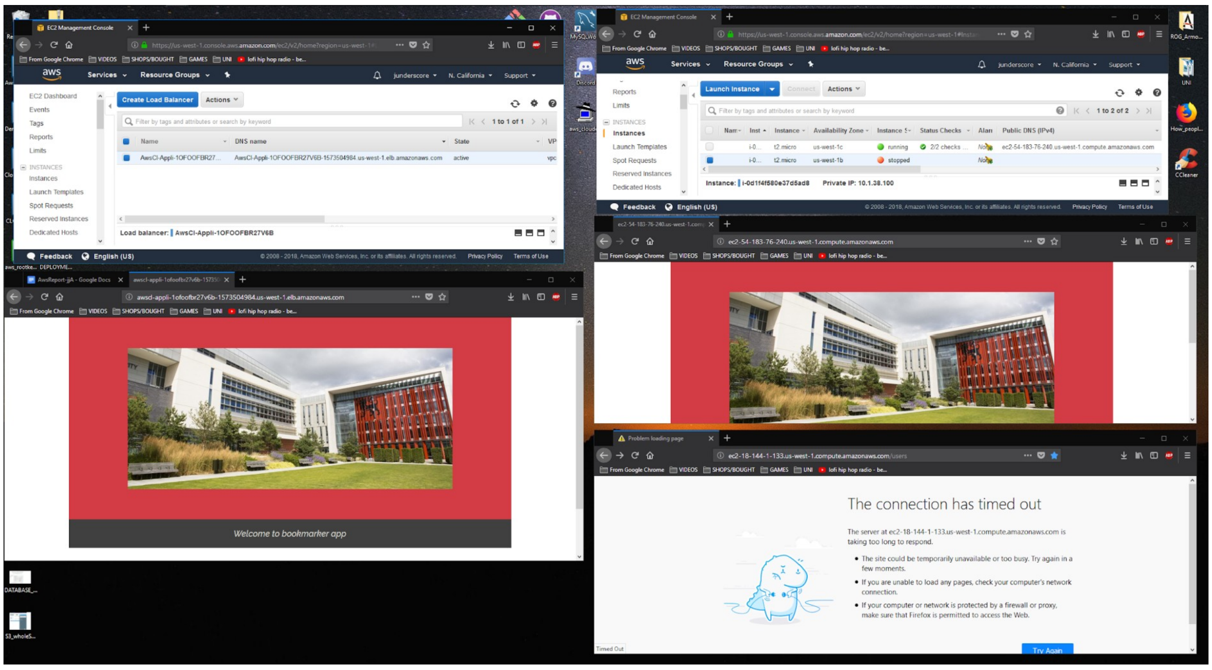
i-0d1f4f580e37d5ad8

Note that when your instances are stopped:
Any data on the ephemeral storage of your instances will be lost.

Cancel | Yes, Stop

ec2-54-183-76-240.us-west-1.compute.amazonaws.com

ec2-18-144-1-133.us-west-1.compute.amazonaws.com



Nam	Inst	Instance	Availability Zone	Instance S	Status Checks	Aları	Public DNS (IPv4)
	i-0...	t2.micro	us-west-1c	running	2/2 checks ...	None	ec2-54-183-76-240.us-west-1.compute.amazonaws.com
	i-0...	t2.micro	us-west-1b	stopped		None	

Instance: i-0d1f4f580e37d5ad8 Private IP: 10.1.38.100

ec2-18-144-1-133.us-west-1.compute.amazonaws.com/users

The connection has timed out

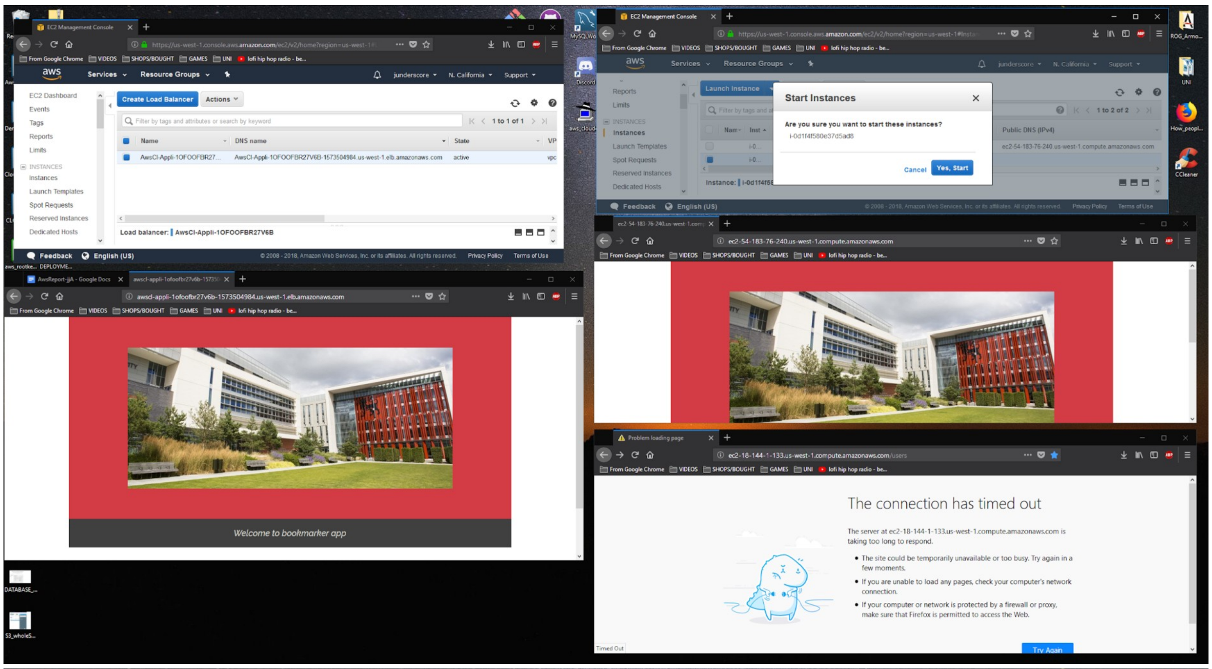
The server at ec2-18-144-1-133.us-west-1.compute.amazonaws.com is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

This shows us that the load balancer is still able to access the app through ubuntu EC2 instance in AZ 1c.

Restarting the instance gives this:



Start Instances

Are you sure you want to start these instances?

i-0d1f4f580e37d5ad8

Cancel Yes, Start

Load balancer: AwsCl-Appli-1OFOOFBR27V6B

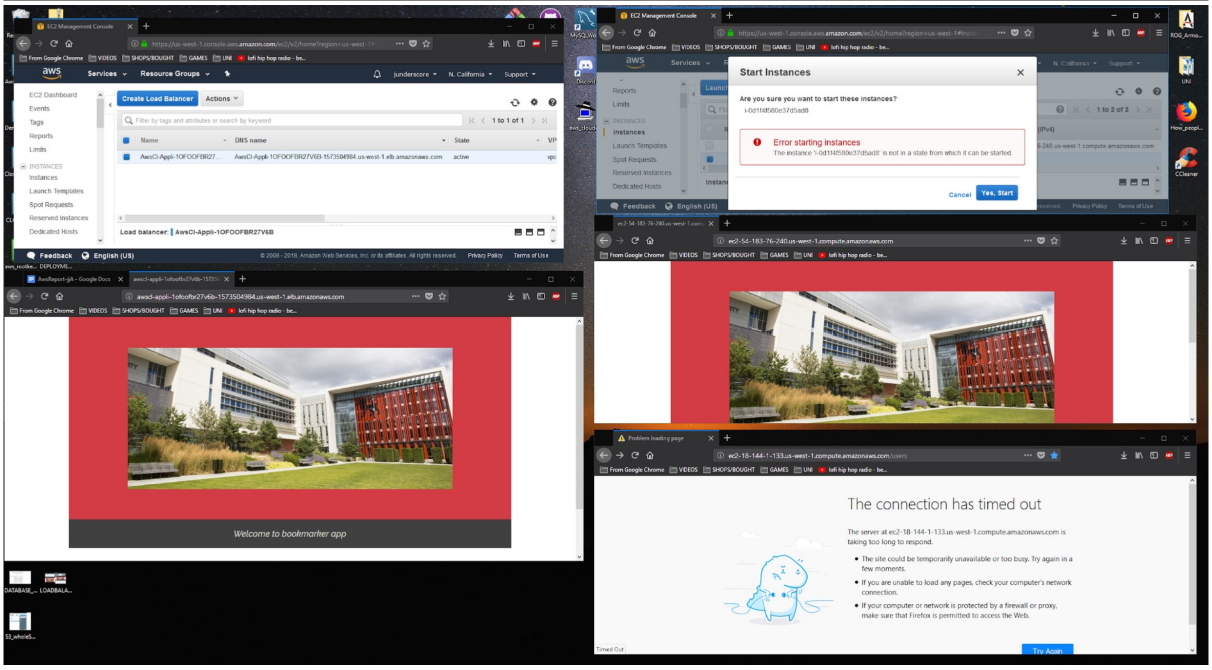
AwsCl-Appli-1OFOOFBR27... AwsCl-Appli-1OFOOFBR27V6B-1573504984.us-west-1.elb.amazonaws.com active

Welcome to bookmarker app

The connection has timed out

The server at ec2-18-144-1-133.us-west-1.compute.amazonaws.com is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.



Start Instances

Are you sure you want to start these instances?

i-0d1f4f580e37d5ad8

Error starting instances

The instance 'i-0d1f4f580e37d5ad8' is not in a state from which it can be started.

Cancel Yes, Start

Welcome to bookmarker app

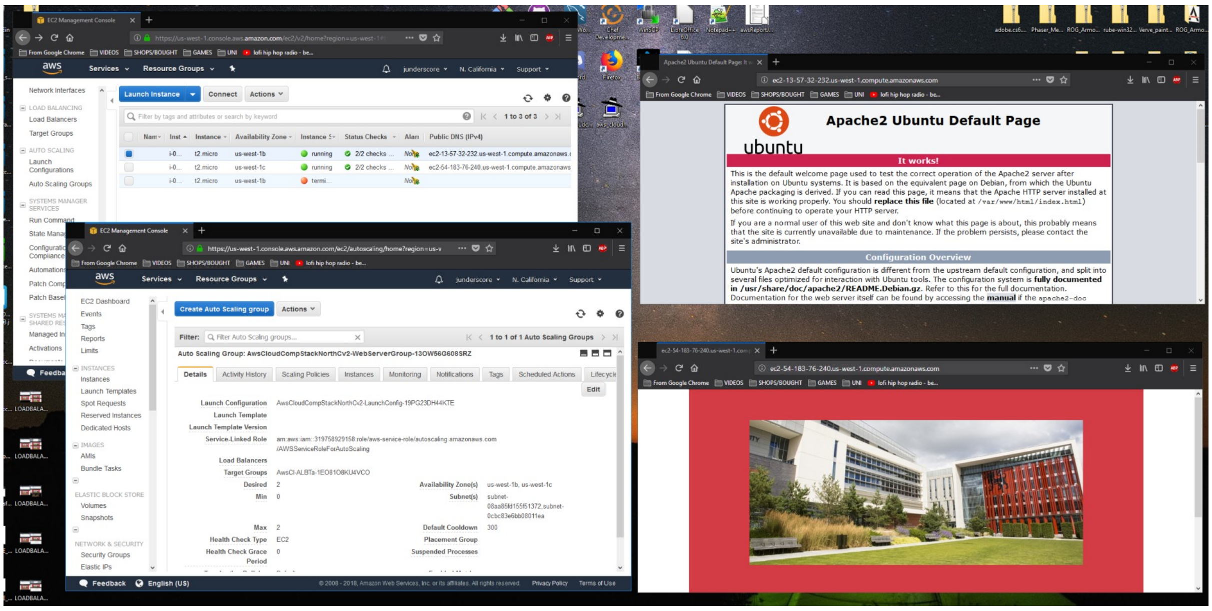
The connection has timed out

The server at ec2-18-144-1-133.us-west-1.compute.amazonaws.com is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

(Instances can’t be recovered once stopped.)

The additional features we have included like the autoscaling group almost fixes this problem as it “defaults” to two running instances and boots another ubuntu EC2 instance ready for app deployment:



https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1

Filter by tags and attributes or search by keyword — 1 to 3 of 3

	Nam	Inst	Instance	Availability Zone	Instance S	Status Checks	Alarm	Public DNS (IPv4)
		i-0...	t2.micro	us-west-1b	running	2/2 checks ...	None	ec2-13-57-32-232.us-west-1.compute.amazonaws.
		i-0...	t2.micro	us-west-1c	running	2/2 checks ...	None	ec2-54-183-76-240.us-west-1.compute.amazonaws
		i-0...	t2.micro	us-west-1b	termi...		None	

https://us-west-1.console.aws.amazon.com/ec2/autoscaling/home?region=us-v

Filter: Filter Auto Scaling groups... — 1 to 1 of 1 Auto Scaling Groups

Auto Scaling Group: AwsCloudCompStackNorthCv2-WebServerGroup-13OW56G608SRZ

Launch Configuration	AwsCloudCompStackNorthCv2-LaunchConfig-19PG23DH44KTE		
Launch Template			
Launch Template Version			
Service-Linked Role	arn:aws:iam::319758929158:role/aws-service-role/autoscaling.amazonaws.com /AWSServiceRoleForAutoScaling		
Load Balancers			
Target Groups	AwsCl-ALBTa-1EO81O8KU4VCO		
Desired	2	Availability Zone(s)	us-west-1b, us-west-1c
Min	0	Subnet(s)	subnet-08aa85fd155f51372,subnet-0cbc83e6bb08011ea
Max	2	Default Cooldown	300
Health Check Type	EC2	Placement Group	
Health Check Grace Period	0	Suspended Processes	

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

ec2-13-57-32-232.us-west-1.compute.amazonaws.com

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc`

ec2-54-183-76-240.us-west-1.compute.amazonaws.com

STAGE FOUR: Security practices (Oregon region)

N.B. All the current setups and tests done from the VPC in Oregon could be continued over into the VPC in North California so are still valid.

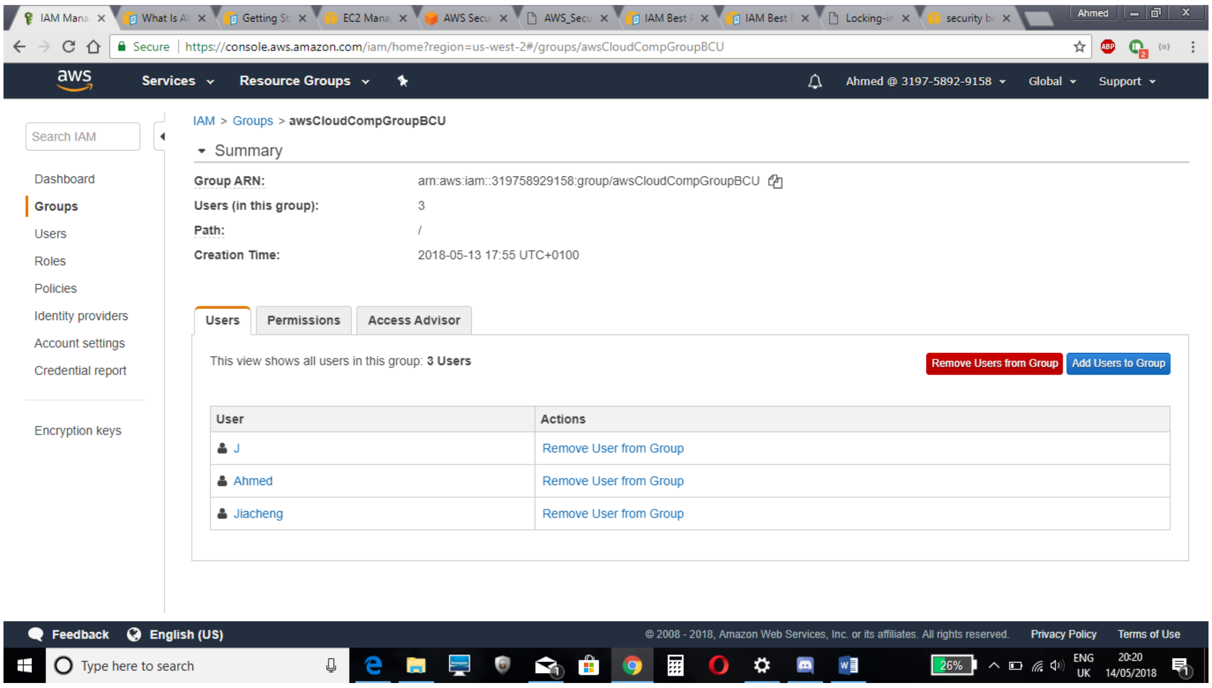
IAM service

One of the security features that were used in this project is the creation of an IAM service. The IAM service is one of the services within AWS. With IAM it's possible to manage users and their security credentials like their passwords, permissions and access keys, also it controls which AWS service the users account can use.

When a user creates an AWS account, the account has access to the AWS management console, it's possible to create access keys which contain an access ID and a secret access key. This is used to make programmatic calls to AWS using different features like the command line interface (CLI).

The IAM allows the creation of individual users with that AWS account, they each get their own username, access keys and password. Each account is tied to a URL link which allows the user to access their account. All the activities that are done within these account are billed to the main account. There are 4 access levels which a user can have which are list, read, write and permissions management. The list and read allows the user to just view and list different services like for example on S3 services the user with these access levels would only be able to view and list buckets. With the write access the user can delete or add objects in the buckets and with the permission management they have full access to all the services and its features such as manage bucket policies on S3.

As a best practice 3 different user accounts where created within 1 group for each of the team members, including the main accounts holder. In this case this was done on James's AWS account. What was done was James's user account had all the access levels since he's the administrator, whereas Ahmed's and cheng's account has the list, read and write access which limited.



https://console.aws.amazon.com/iam/home?region=us-west-2#/groups/awsCloudCompGroupBCU

IAM > Groups > awsCloudCompGroupBCU

Summary

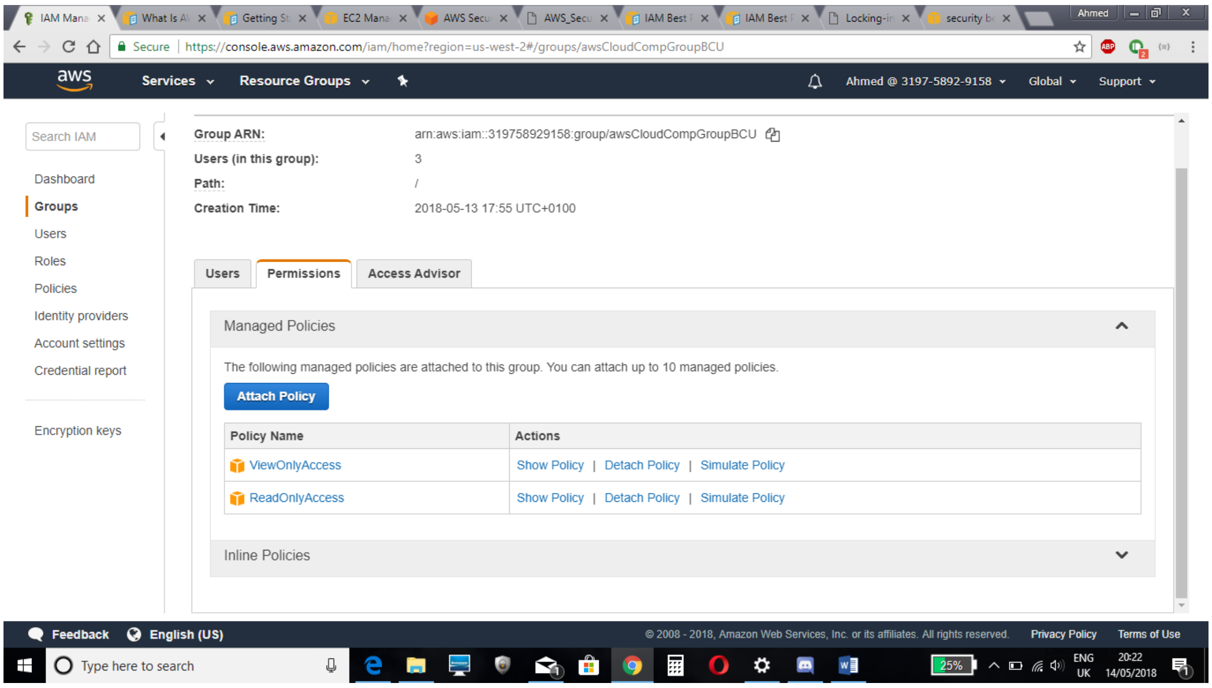
Group ARN:	arn:aws:iam::319758929158:group/awsCloudCompGroupBCU
Users (in this group):	3
Path:	/
Creation Time:	2018-05-13 17:55 UTC+0100

This view shows all users in this group: **3 Users**

User	Actions
J	Remove User from Group
Ahmed	Remove User from Group
Jiacheng	Remove User from Group

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 1: the IAM group with all the users contained.



https://console.aws.amazon.com/iam/home?region=us-west-2#/groups/awsCloudCompGroupBCU

Group ARN:	arn:aws:iam::319758929158:group/awsCloudCompGroupBCU
Users (in this group):	3
Path:	/
Creation Time:	2018-05-13 17:55 UTC+0100

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

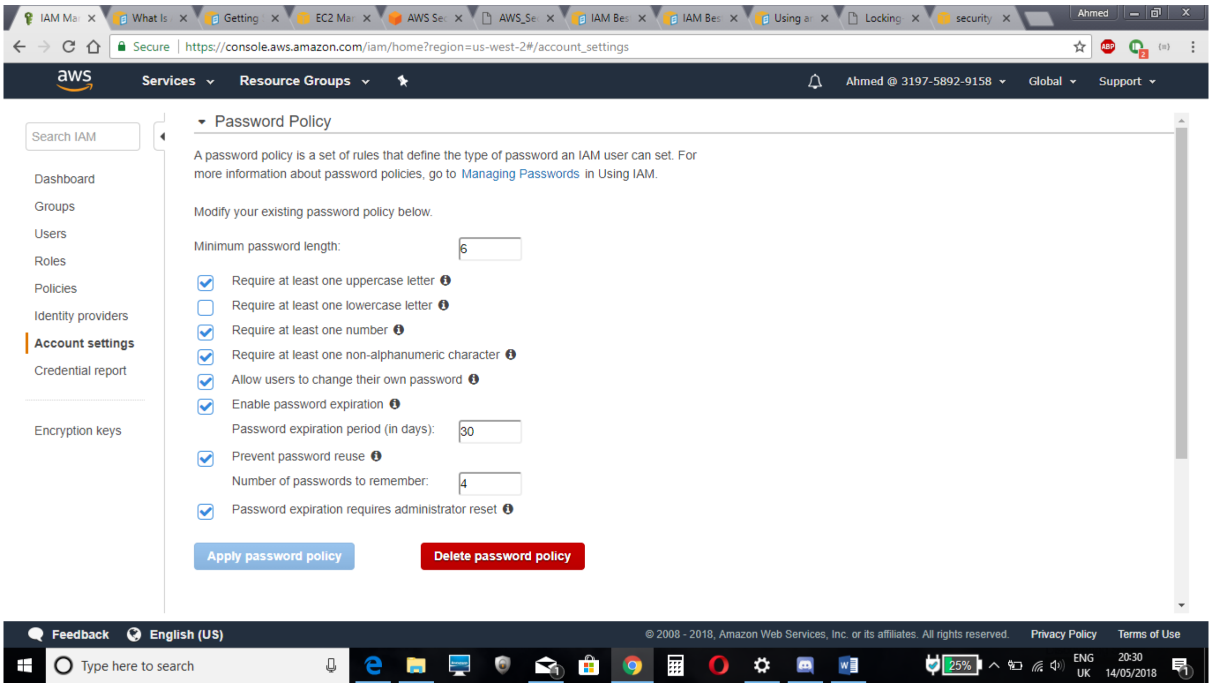
Policy Name	Actions
ViewOnlyAccess	Show Policy | Detach Policy | Simulate Policy
ReadOnlyAccess	Show Policy | Detach Policy | Simulate Policy

Inline Policies

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 2: permissions included in the group

Another feature with the IAM service was an IAM password policy allowing the user to change their password. What was done was when one user wants to change their password they would need to create a strong password. The screenshot below shows what policies was added.



Password Policy

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to Managing Passwords in Using IAM.

Modify your existing password policy below.

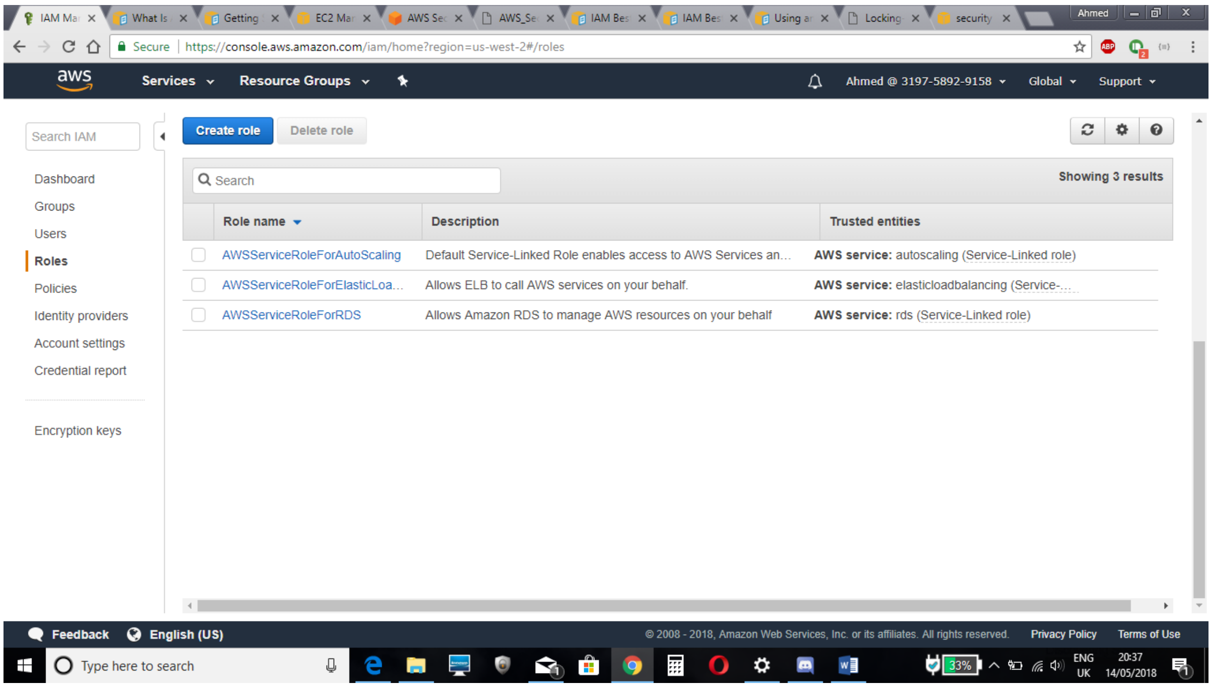
Minimum password length: 6

- ✓ Require at least one uppercase letter
- ☐ Require at least one lowercase letter
- ✓ Require at least one number
- ✓ Require at least one non-alphanumeric character
- ✓ Allow users to change their own password
- ✓ Enable password expiration
  - Password expiration period (in days): 30
- ✓ Prevent password reuse
  - Number of passwords to remember: 4
- ✓ Password expiration requires administrator reset

Figure 3: password policy.

IAM roles

When an EC2 instance is running applications that are run on it need certain IDs to make use of the AWS services they need. A role is basically a unit with its own set of permissions. Using IAM roles allows the EC2 instance to temporarily gain these credentials to allow them to use the services they need. An example can be application code running on an EC2 instance that needs to perform actions on AWS resources. IAM roles distribute keys that are valid for a short time, thus ensuring they are a more secure way to grant access. Below is a screen shot of the roles that are included in the EC2 instances within the account.

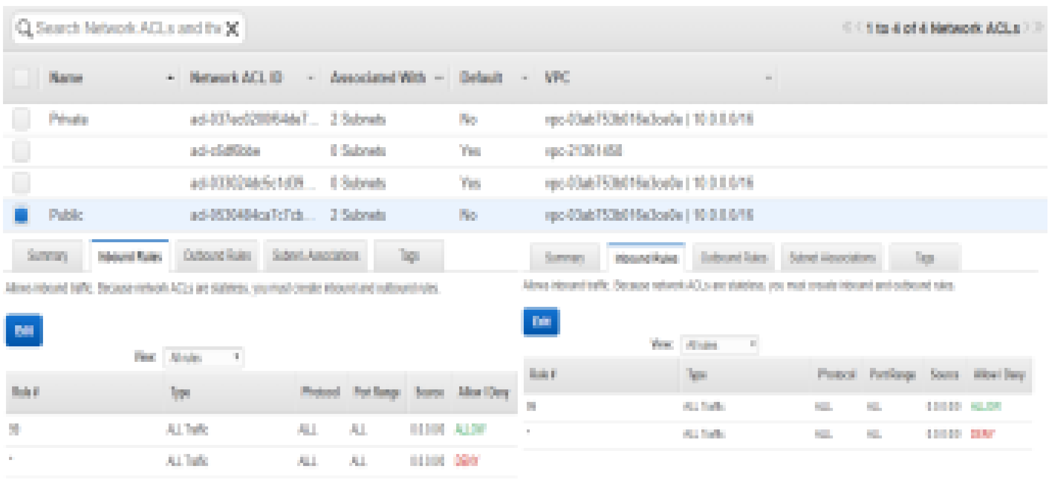


	Role name	Description	Trusted entities
☐	AWSServiceRoleForAutoScaling	Default Service-Linked Role enables access to AWS Services an...	**AWS service:** autoscaling (Service-Linked role)
☐	AWSServiceRoleForElasticLoa...	Allows ELB to call AWS services on your behalf.	**AWS service:** elasticloadbalancing (Service-...
☐	AWSServiceRoleForRDS	Allows Amazon RDS to manage AWS resources on your behalf	**AWS service:** rds (Service-Linked role)

Figure 4: IAM Roles

Network ACLs

Network access control list (ACL) is an optional layer of security for the VPC which essentially acts as a firewall for controlling traffic in and out of one or more subnets. As a best practice the network ACLs that were created have the same outbound rules as the security groups that were created. The security groups have more specific inbound rules. The resemblance in the rules add extra layer of security to the VPC. Each subnet is related to a network ACL A network ACL has a numbered list of rules are valued in order from lowest to highest, to control if traffic is allowed in or out of any subnet linked with the network ACL. The highest number that can used for a rule is 32766.



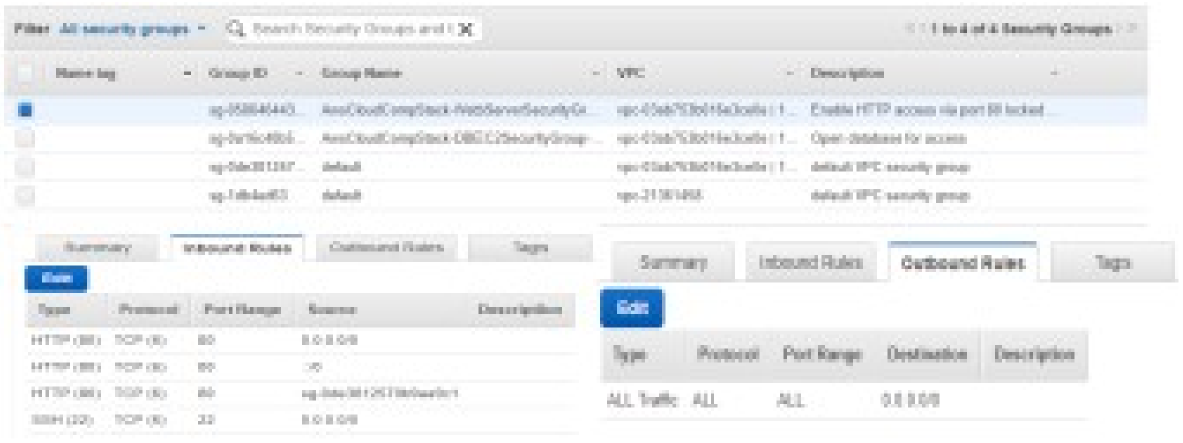
	Name	Network ACL ID	Associated With	Default	VPC
☐	Private	[illegible]	2 Subnets	No	[illegible]
☐		[illegible]	0 Subnets	Yes	[illegible]
☐		[illegible]	0 Subnets	Yes	[illegible]
■	Public	[illegible]	2 Subnets	No	[illegible]

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Figure 5: ACLs with inbound and outbound rules

Security groups

A security groups actions is being a simulated firewall that controls the traffic for one or more instances. When an instance is launched, it is good practice to add one or more security groups to that instance. adding rules to each security group allows traffic to or from its related instances. The rules of a security group control the inbound traffic that's allowed to get to the instances which link to the security group and the outbound traffic that's allowed to leave them. As a best secrutiy practice these security groups implemment the least premissive rules.


Filter: All security groups   1 to 4 of 4 Security Groups

	Name tag	Group ID	Group Name	VPC	Description
■		sg-05864443...	AwsCloudCompStack-WebServerSecurityGr...	vpc-03ab7f3b014a3ce8e | 1...	Enable HTTP access via port 80 locked...
☐		sg-0e16c48b5...	AwsCloudCompStack-DBEC2SecurityGroup-...	vpc-03ab7f3b014a3ce8e | 1...	Open database for access
☐		sg-0de381297...	default	vpc-03ab7f3b014a3ce8e | 1...	default VPC security group
☐		sg-1db4a453	default	vpc-2130149	default VPC security group

Summary | Inbound Rules | Outbound Rules | Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	0.0.0.0/0	
HTTP (80)	TCP (6)	80	::/0	
HTTP (80)	TCP (6)	80	sg-0de381297de0ae0c1	
SSH (22)	TCP (6)	22	0.0.0.0/0	

Summary | Inbound Rules | Outbound Rules | Tags

Edit

Type	Protocol	Port Range	Destination	Description
ALL Traffic	ALL	ALL	0.0.0.0/0	

Figure 6: security groups with inbound and outbound rules

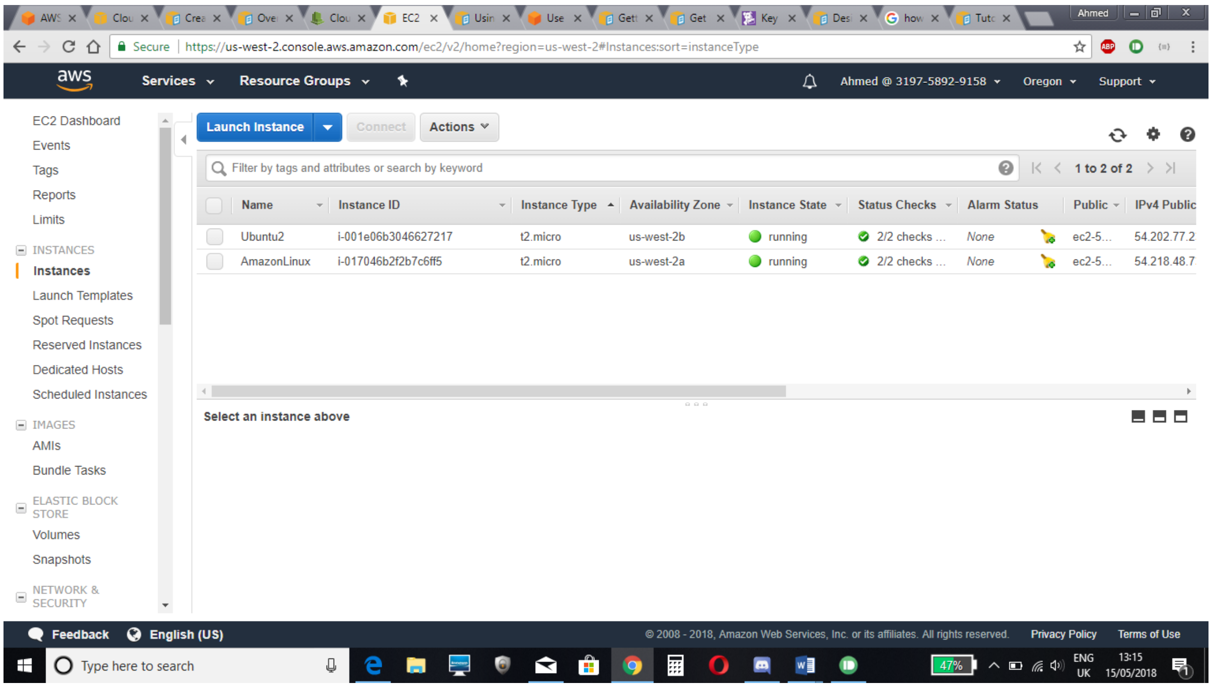
Additional options

There are other security options that can be added, one such security element is a multi-factor authentication. For extra security, mufti-factor authentication can be enabled for the users who have all the access levels as they are allowed to access delicate resources. The user can obtain a device that would generate a onetime unique passcode, they would then need to sign in with both their user account password and this one time passcode.

Demonstrates high availability at different levels

Instances:

There were 2 instances created and with high availability in mind, these instances have different availability zones. By having them in different availability zones it improves the fault tolerance in the application. If on availability zone is not working because of an outage, all traffic is routed to the other availability zone.



https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:sort=instanceType

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public	IPv4 Public
Ubuntu2	i-001e06b3046627217	t2.micro	us-west-2b	● running	✓ 2/2 checks ...	None	ec2-5...	54.202.77.2
AmazonLinux	i-017046b2f2b7c6ff5	t2.micro	us-west-2a	● running	✓ 2/2 checks ...	None	ec2-5...	54.218.48.7

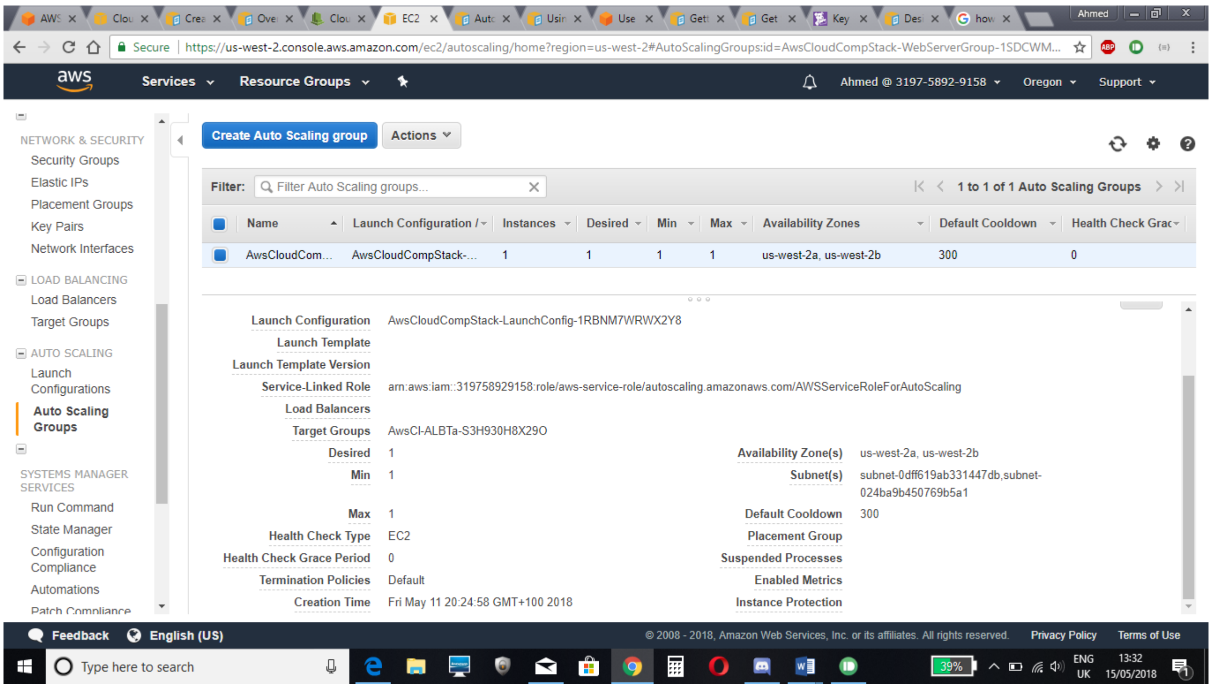
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Figure 1: instances with different availability zones.

Auto scaling groups - transferred to North California.

This contains the EC2 instances that have a link between them, like in our example where the 2 instances that we created, one for the web application and another for the PHP application.

What the auto scaling group does it launches the necessary number of instances to meet the requirements we need which is for it to run a web application that can edit a RDS database. For our case one was enough. The auto scaling group keeps this one instance running and keeps tabs on that instance like doing health check, if that specific instance fails the auto scaling group terminates that instance in the group and launches the other instance that is available so that it switches it out.



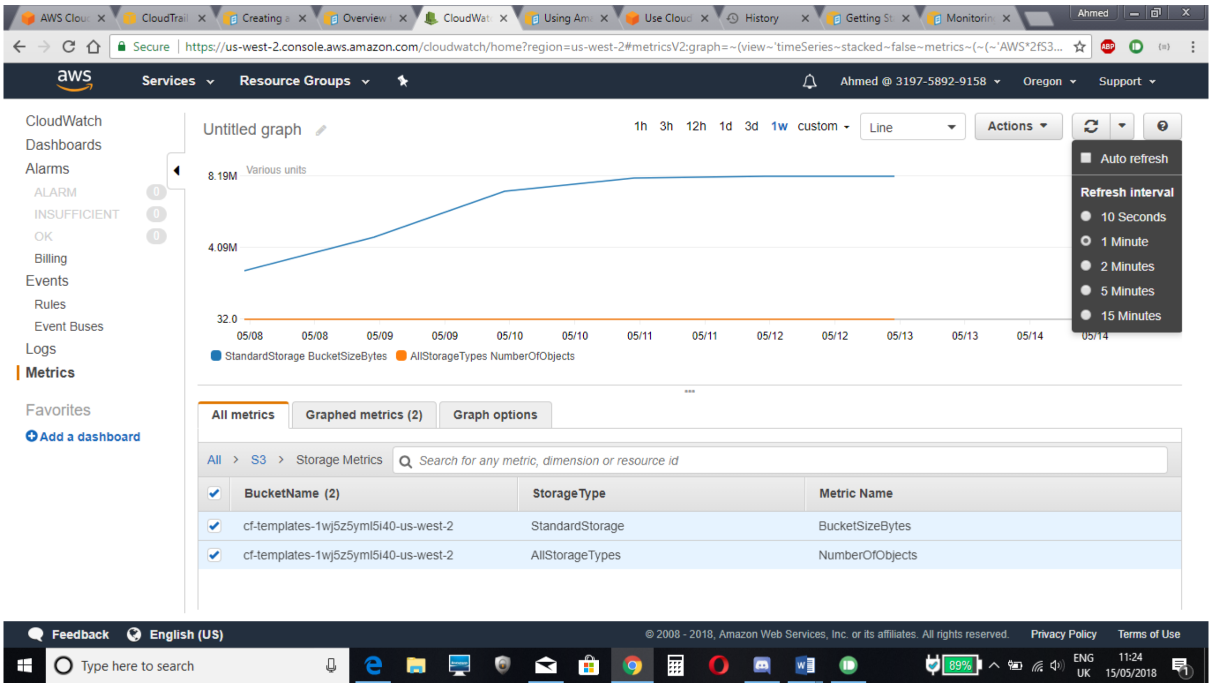
Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
AwsCloudCom...	AwsCloudCompStack-...	1	1	1	1	us-west-2a, us-west-2b	300	0

Launch Configuration	AwsCloudCompStack-LaunchConfig-1RBNM7WRWX2Y8		
Launch Template			
Launch Template Version			
Service-Linked Role	arn:aws:iam::319758929158:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling		
Load Balancers			
Target Groups	AwsCl-ALBTa-S3H930H8X29O		
Desired	1	Availability Zone(s)	us-west-2a, us-west-2b
Min	1	Subnet(s)	subnet-0dff619ab331447db,subnet-024ba9b450769b5a1
Max	1	Default Cooldown	300
Health Check Type	EC2	Placement Group	
Health Check Grace Period	0	Suspended Processes	
Termination Policies	Default	Enabled Metrics	
Creation Time	Fri May 11 20:24:58 GMT+100 2018	Instance Protection	

Figure 3: auto scaling for 2 different instances.

STAGE FIVE: Monitoring Instances Using CloudWatch

With CloudWatch it's possible to monitor the instances within the AWS account. The data that is collected with is in raw form is processed from EC2 instance and converted into readable, real-time metrics. These data help in having advanced perception on how the web application or the different services that are running within the instance is performing.
By default, Amazon EC2 sends metric data to CloudWatch in 5-minute periods. Our group configured this option to send data in 1 minute periods. This allows more detailed monitoring on the instances created.



BucketName (2)	StorageType	Metric Name
cf-templates-1wj5z5yml5i40-us-west-2	StandardStorage	BucketSizeBytes
cf-templates-1wj5z5yml5i40-us-west-2	AllStorageTypes	NumberOfObjects

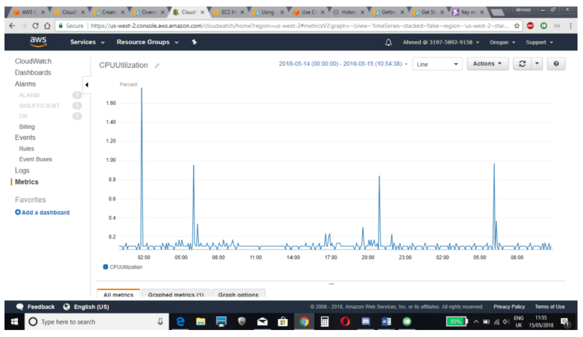
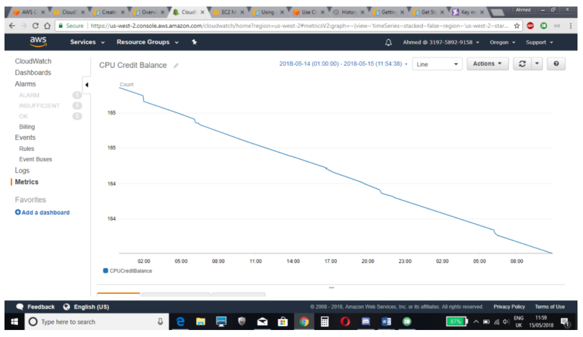
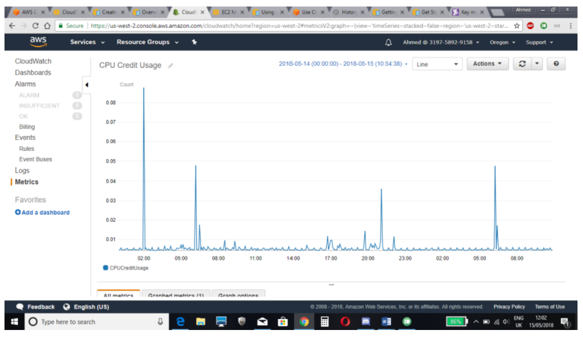
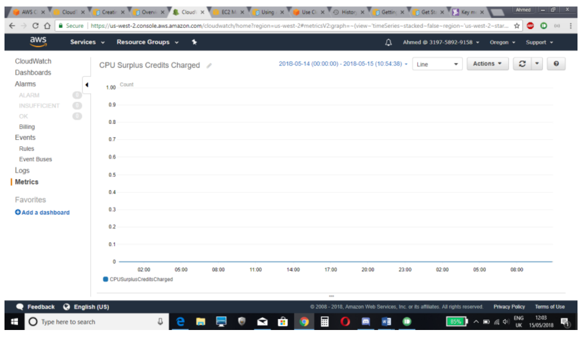
Figure 1: metric data refresh intervals

When monitoring the instances it's important to monitor the basic system-level metrics which are the core setup of the instances to make user they are healthy. Other things to consider tracking are the resources being used. For our project we covered 3 metrics types.
CPU
There are several CPU configurations within ECS instances, tracking the CPU usage allows us to know that the instance is appropriately sized for the job that it's configured for.
5 different metrics fall under this category which are:

1. CPU Utilization
2. CPU Credit Balance
3. CPU Credit Usage
4. CPU Surplus Credits Charged
5. CPU Surplus Credit Balance

The table below shows the graphs for these metrics. With description on what these metrics are and screenshots of the graphs corresponding with them. There will a table for each instance. The graphs show the usage between 2018/05/14 at 00:00:00 and 2018/05/15 at 10:54:38

Table 1: Ubuntu 2

Test	Screenshot of graph
CPUUtilization: Percentage of allocated EC2 compute units that are currently in use on the instance.	 CPUUtilization 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) Percent CPUUtilization
CPUCreditBalance: Number of CPU credits that an instance has accumulated.	 CPU Credit Balance 2018-05-14 (01:00:00) - 2018-05-15 (11:54:38) Count CPUCreditBalance
CPUCreditUsage Number of CPU credits consumed.	 CPU Credit Usage 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) Count CPUCreditUsage
CPUSurplusCreditsCharged Number of surplus credits not offset by earned CPU credits and that will incur charges	 CPU Surplus Credits Charged 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) Count CPUSurplusCreditsCharged

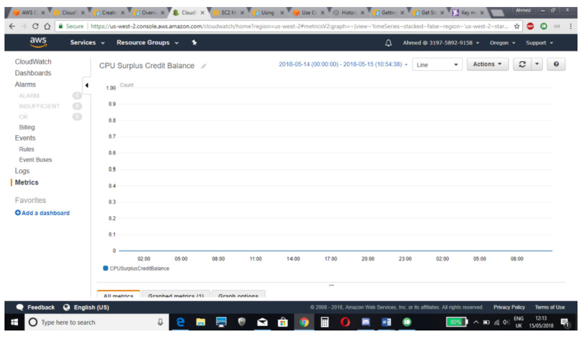
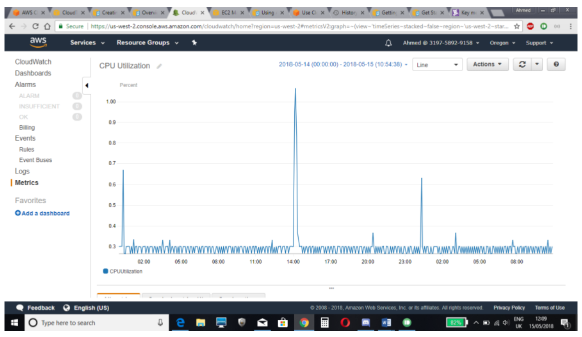
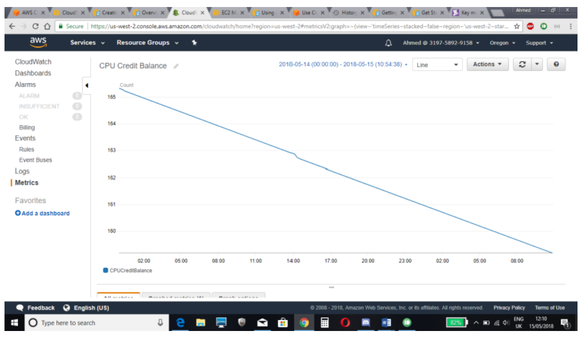
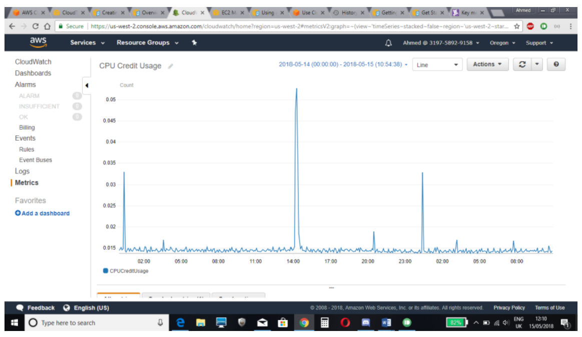
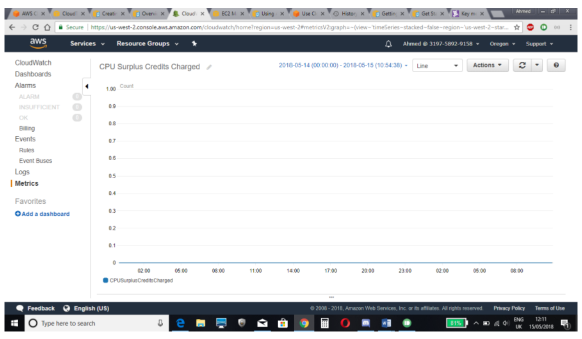
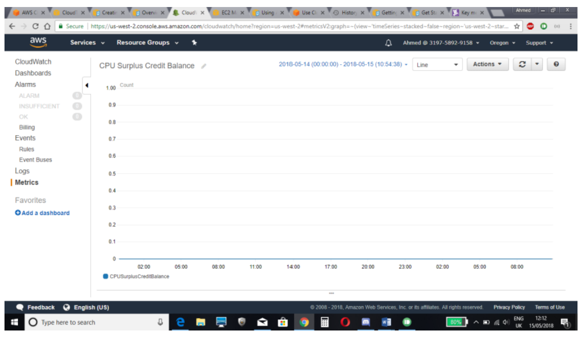
CPUSurplusCreditBalance Number of credits consumed after CPU credit balance has reached 0.	 CPU Surplus Credit Balance 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) CPUSurplusCreditBalance

Table 2: Amazon Linux

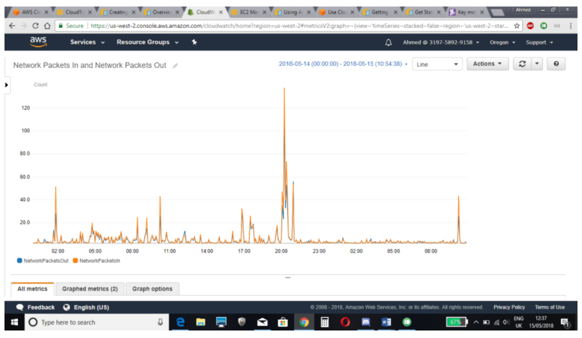
Test	Screenshot of graph
CPUUtilization: Percentage of allocated EC2 compute units that are currently in use on the instance.	 CPU Utilization 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) CPUUtilization
CPUCreditBalance: Number of CPU credits that an instance has accumulated.	 CPU Credit Balance 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) CPUCreditBalance
CPUCreditUsage Number of CPU credits consumed.	 CPU Credit Usage 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) CPUCreditUsage

CPUSurplusCreditsCharged Number of surplus credits not offset by earned CPU credits and that will incur charges	 CPU Surplus Credits Charged 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) CPUSurplusCreditsCharged
CPUSurplusCreditBalance Number of credits consumed after CPU credit balance has reached 0.	 CPU Surplus Credit Balance 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) CPUSurplusCreditBalance

Network
The main reason behind the importance of network metrics are their importance for EC2 since it depend on solid network connections, which might be spread across various availability zones.
4 different metrics fall under this category which are:

1. Network packs in
2. Network packs out
3. Network in
4. Network out

The table below shows the graphs for these metrics. With description on what these metrics are and screenshots of the graphs corresponding with them. There will a table for each instance. The graphs show the usage between 2018/05/14 at 00:00:00 and 2018/05/15 at 10:54:38
Table 1: Ubuntu 2

Test	
NetworkPacketsIn/NetworkPacketsOut: Number of packets received/sent out on all network interfaces by the instance. Only available at five-minute resolution.	 Network Packets In and Network Packets Out 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) NetworkPacketsOut NetworkPacketsIn

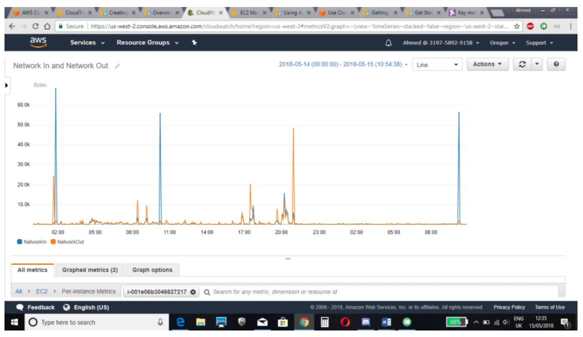
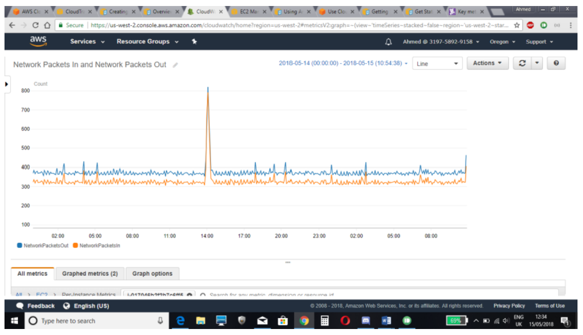
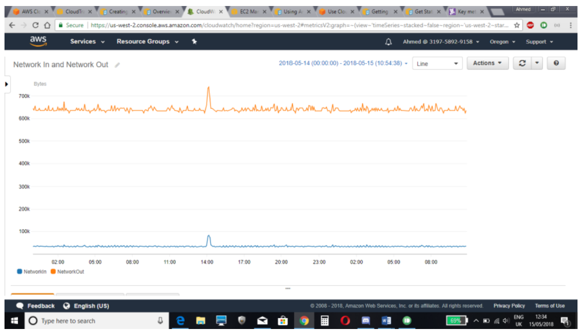
NetworkIn NetworkOut Number of bytes received/sent out on all network interfaces by the instance.	 Network In and Network Out 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) NetworkIn NetworkOut

Table 2: Amazon Linux

Test	
NetworkPacketsIn/NetworkPacketsOut: Number of packets received/sent out on all network interfaces by the instance. Only available at five-minute resolution.	 Network Packets In and Network Packets Out 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) NetworkPacketsOut NetworkPacketsIn
NetworkIn NetworkOut Number of bytes received/sent out on all network interfaces by the instance.	 Network In and Network Out 2018-05-14 (00:00:00) - 2018-05-15 (10:54:38) NetworkIn NetworkOut

Status checks

These checks are simply what is used to check the status of the instance and the systems within it. 2 different metrics fall under this category which are:

1. Status Check Failed_System
2. Status Check Failed_Instance

The table below shows the graphs for these metrics. With description on what these metrics are and screenshots of the graphs corresponding with them. There will a table for each instance. The graphs show the usage between 2018/05/14 at 00:00:00 and 2018/05/15 at 10:54:38

Table 1: Ubuntu 2

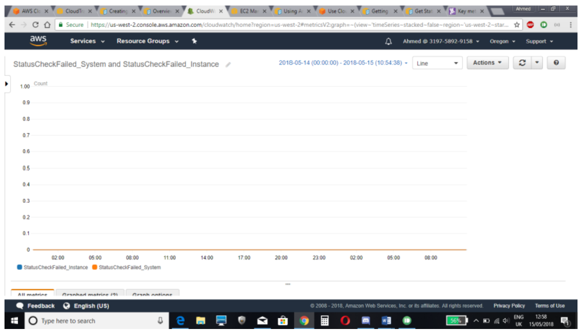
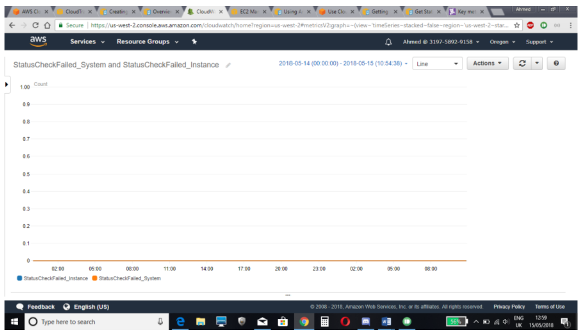
StatusCheckFailed_System StatusCheckFailed_Instance Returns 1 if the instance has failed EC2's system/instance status check.	

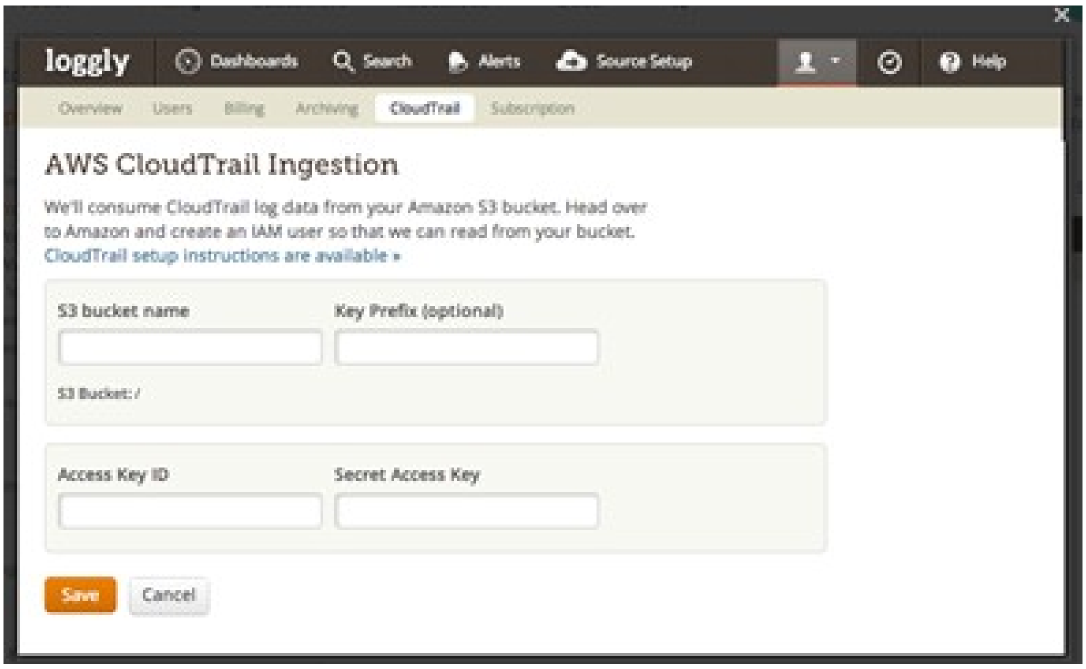
Table 2: Amazon Linux

StatusCheckFailed_System StatusCheckFailed_Instance Returns 1 if the instance has failed EC2's system/instance status check.	

Cloud trail

This is basically an API (application program interface) which creates log files and sends it to the S3 bucket for storage. They provide an audit trail of changes to the AWS instances and the interactions that happen within them.

We have used a third party website called Loogly to create graphs that represent different findings from these log files. What we did was we gave the site the S3 bucket name and an IAM user's access key ID and secret access key. With that user having the right permissions which are list buckets and het objects we managed to get all the S3 data into the site.



AWS CloudTrail Ingestion

We'll consume CloudTrail log data from your Amazon S3 bucket. Head over to Amazon and create an IAM user so that we can read from your bucket.
CloudTrail setup instructions are available »

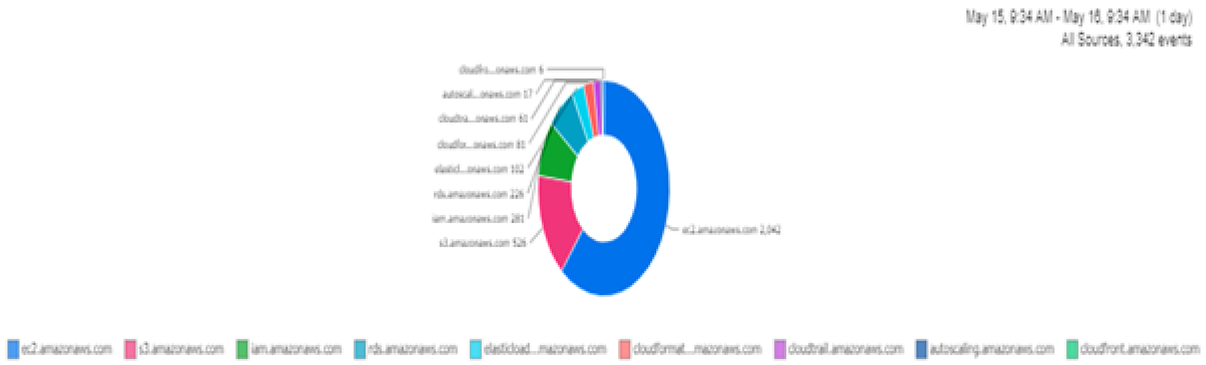
S3 bucket name
Key Prefix (optional)
S3 Bucket: /

Access Key ID
Secret Access Key

Save Cancel

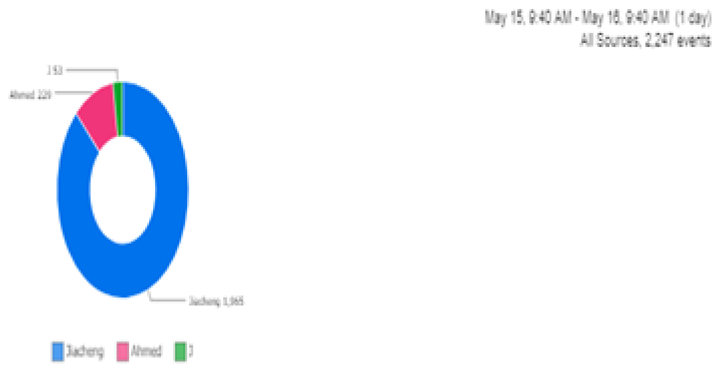
The graphs below show different charts and graphs that we were able to pull from the CloudTrail

Top AWS CloudTrail Event Sources in the Last Day.



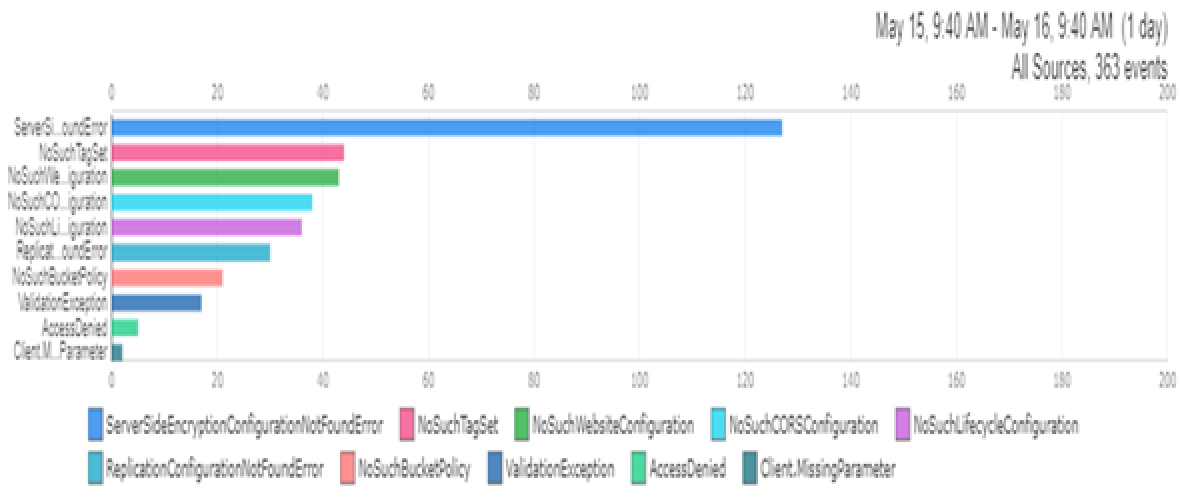
This chart shows which of the AWS services were modified by the admin. This can be used to check for unexpected activity

AWS CloudTrail Top Users in the Last Day



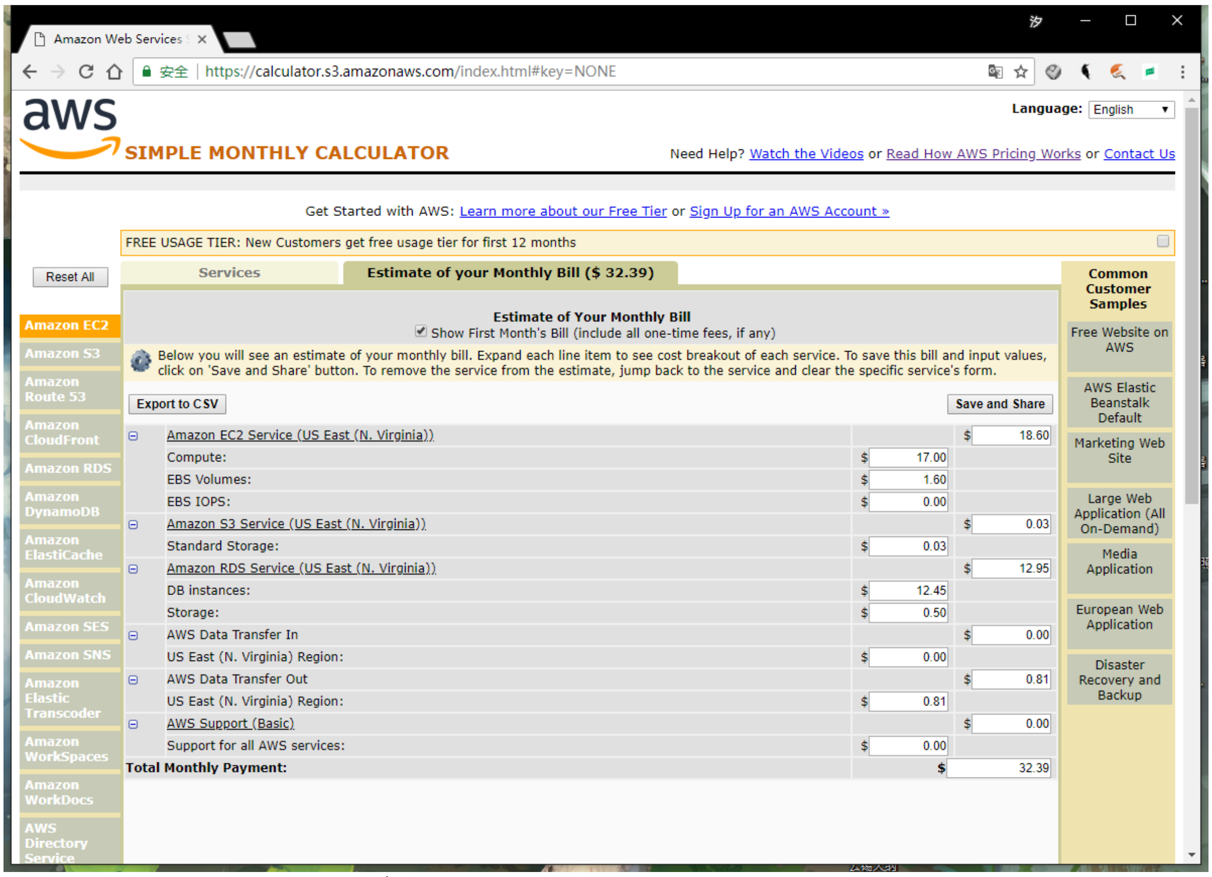
This chart shows the activities the users within the AWS account. Unknown usernames can show a security breach

AWS CloudTrail Top Error Messages in the Last Day



This chart shows all the error messages that have come up from all sources.

Cost breakdown for our current setup:
If one person use data 50MB every month, 100 user will use 5GB every month.



Get Started with AWS: Learn more about our Free Tier or Sign Up for an AWS Account »

FREE USAGE TIER: New Customers get free usage tier for first 12 months

Estimate of your Monthly Bill ($ 32.39)

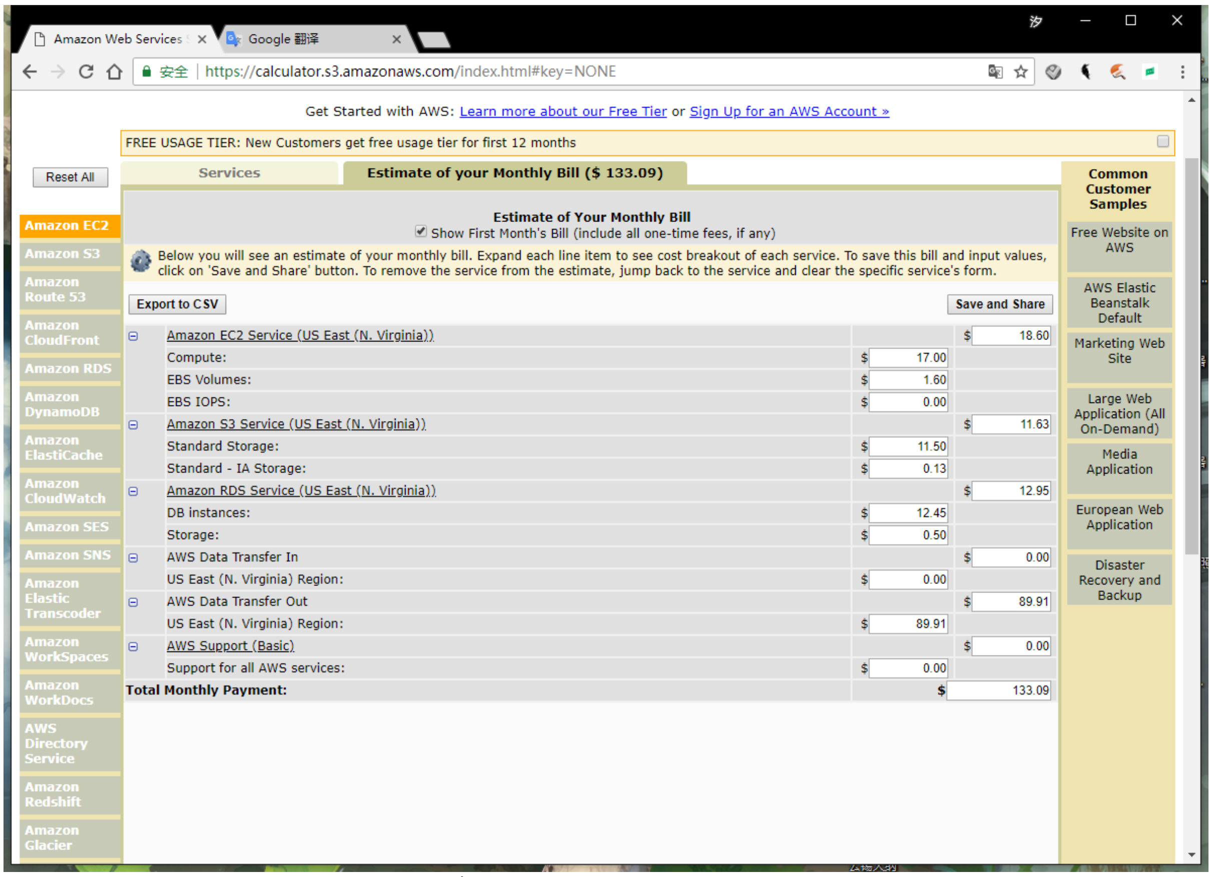
Estimate of Your Monthly Bill

☑ 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.

Amazon EC2 Service (US East (N. Virginia))		$ 18.60
Compute:	$ 17.00	
EBS Volumes:	$ 1.60	
EBS IOPS:	$ 0.00	
Amazon S3 Service (US East (N. Virginia))		$ 0.03
Standard Storage:	$ 0.03	
Amazon RDS Service (US East (N. Virginia))		$ 12.95
DB instances:	$ 12.45	
Storage:	$ 0.50	
AWS Data Transfer In		$ 0.00
US East (N. Virginia) Region:	$ 0.00	
AWS Data Transfer Out		$ 0.81
US East (N. Virginia) Region:	$ 0.81	
AWS Support (Basic)		$ 0.00
Support for all AWS services:	$ 0.00	
Total Monthly Payment:		$ 32.39

5GB data translate will need $32.39every month.
10 thousand user will be 500GB/month



Get Started with AWS: Learn more about our Free Tier or Sign Up for an AWS Account »

FREE USAGE TIER: New Customers get free usage tier for first 12 months

Estimate of your Monthly Bill ($ 133.09)

Estimate of Your Monthly Bill

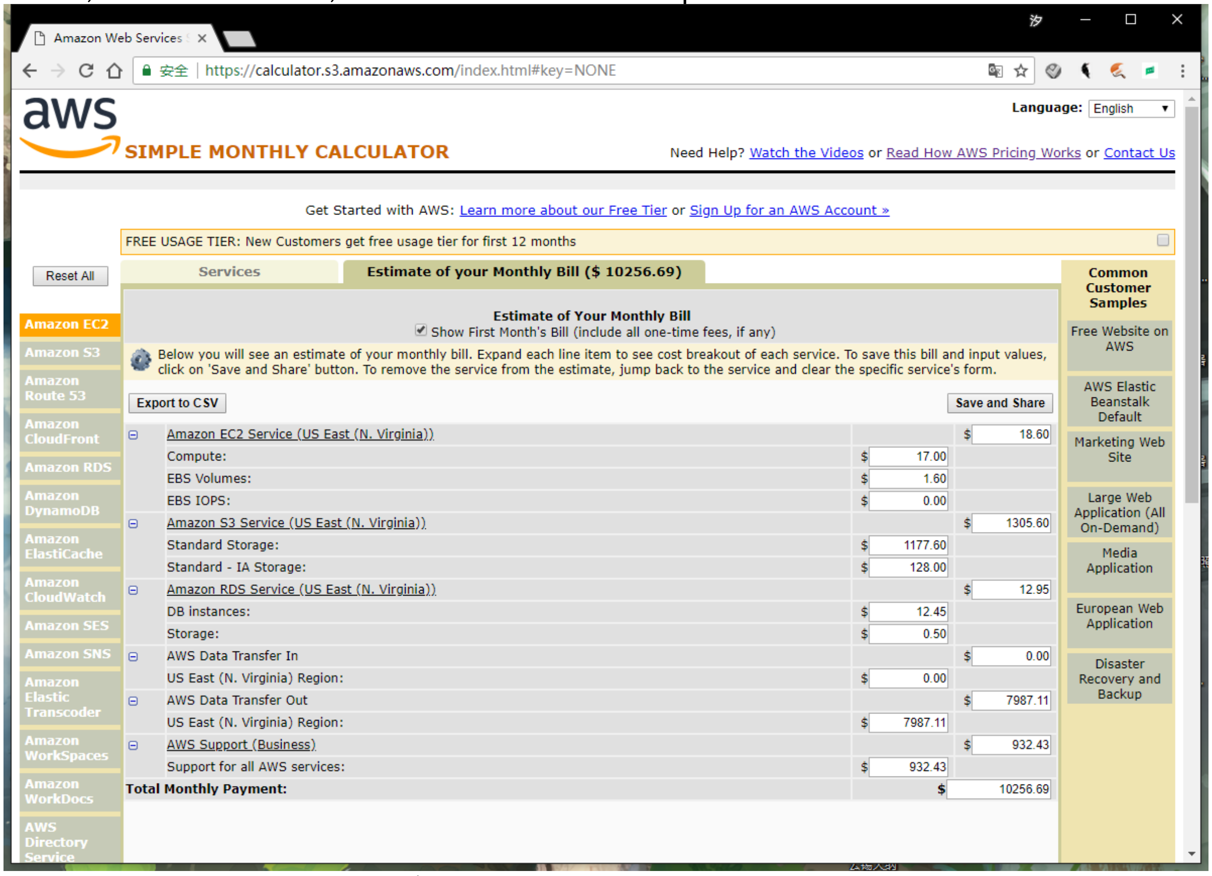
☑ Show First Month's Bill (include all one-time fees, if any)

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.

Amazon EC2 Service (US East (N. Virginia))		$ 18.60
Compute:	$ 17.00	
EBS Volumes:	$ 1.60	
EBS IOPS:	$ 0.00	
Amazon S3 Service (US East (N. Virginia))		$ 11.63
Standard Storage:	$ 11.50	
Standard - IA Storage:	$ 0.13	
Amazon RDS Service (US East (N. Virginia))		$ 12.95
DB instances:	$ 12.45	
Storage:	$ 0.50	
AWS Data Transfer In		$ 0.00
US East (N. Virginia) Region:	$ 0.00	
AWS Data Transfer Out		$ 89.91
US East (N. Virginia) Region:	$ 89.91	
AWS Support (Basic)		$ 0.00
Support for all AWS services:	$ 0.00	
Total Monthly Payment:		$ 133.09

About 500GB/mouth, it will be use $133.09

At last, if one million user, the data transfer is 50 TB per month



Need Help? Watch the Videos or Read How AWS Pricing Works or Contact Us

Get Started with AWS: Learn more about our Free Tier or Sign Up for an AWS Account »

FREE USAGE TIER: New Customers get free usage tier for first 12 months

Estimate of your Monthly Bill ($ 10256.69)

Estimate of Your Monthly Bill

✓ 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.

Amazon EC2 Service (US East (N. Virginia))		$ 18.60
Compute:	$ 17.00	
EBS Volumes:	$ 1.60	
EBS IOPS:	$ 0.00	
Amazon S3 Service (US East (N. Virginia))		$ 1305.60
Standard Storage:	$ 1177.60	
Standard - IA Storage:	$ 128.00	
Amazon RDS Service (US East (N. Virginia))		$ 12.95
DB instances:	$ 12.45	
Storage:	$ 0.50	
AWS Data Transfer In		$ 0.00
US East (N. Virginia) Region:	$ 0.00	
AWS Data Transfer Out		$ 7987.11
US East (N. Virginia) Region:	$ 7987.11	
AWS Support (Business)		$ 932.43
Support for all AWS services:	$ 932.43	
Total Monthly Payment:		$ 10256.69

That Estimate of monthly bill is $10256.69

STAGE SIX: Remarks about project improvements

AWS systems Manager

Systems manager is Powerful tools. Systems Manager provides a unified user interface that lets you view the operational data of multiple AWS services and automate operational tasks on AWS resources. With Systems Manager, you can group resources (for example, Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances) by application, view operational data for monitoring and troubleshooting, and act on resource groups. Systems Manager simplifies resource and application management, reduces the time required to detect and resolve operational problems, and enables you to run and manage your infrastructure on a large scale with ease and security.(Mathew, 2014a)

AWS Config

AWS Config is a service that evaluates, reviews, and evaluates the configuration of AWS resources. Config continuously monitors and records AWS resource configurations and supports automatic configuration of records based on configuration requirements. And you can view the relationship between configuration changes and AWS resources, drill down into detailed resource configuration history, and determine if the configuration meets the configuration requirements specified in the internal guide. Absconding basically has the following features, configurable and customizable rules, configuration history of AWS resources, software configuration history, configuration snapshots, resource relationship tracking, cloud management control panel, partner solution ecosystem. As a result, compliance audits, security analysis, change management, and operational troubleshooting will simplify these features.(Mathew, 2014b)

References

Ref - AWS CloudFormation [WWW Document], n.d. URL https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference-ref.html (accessed 5.16.18).

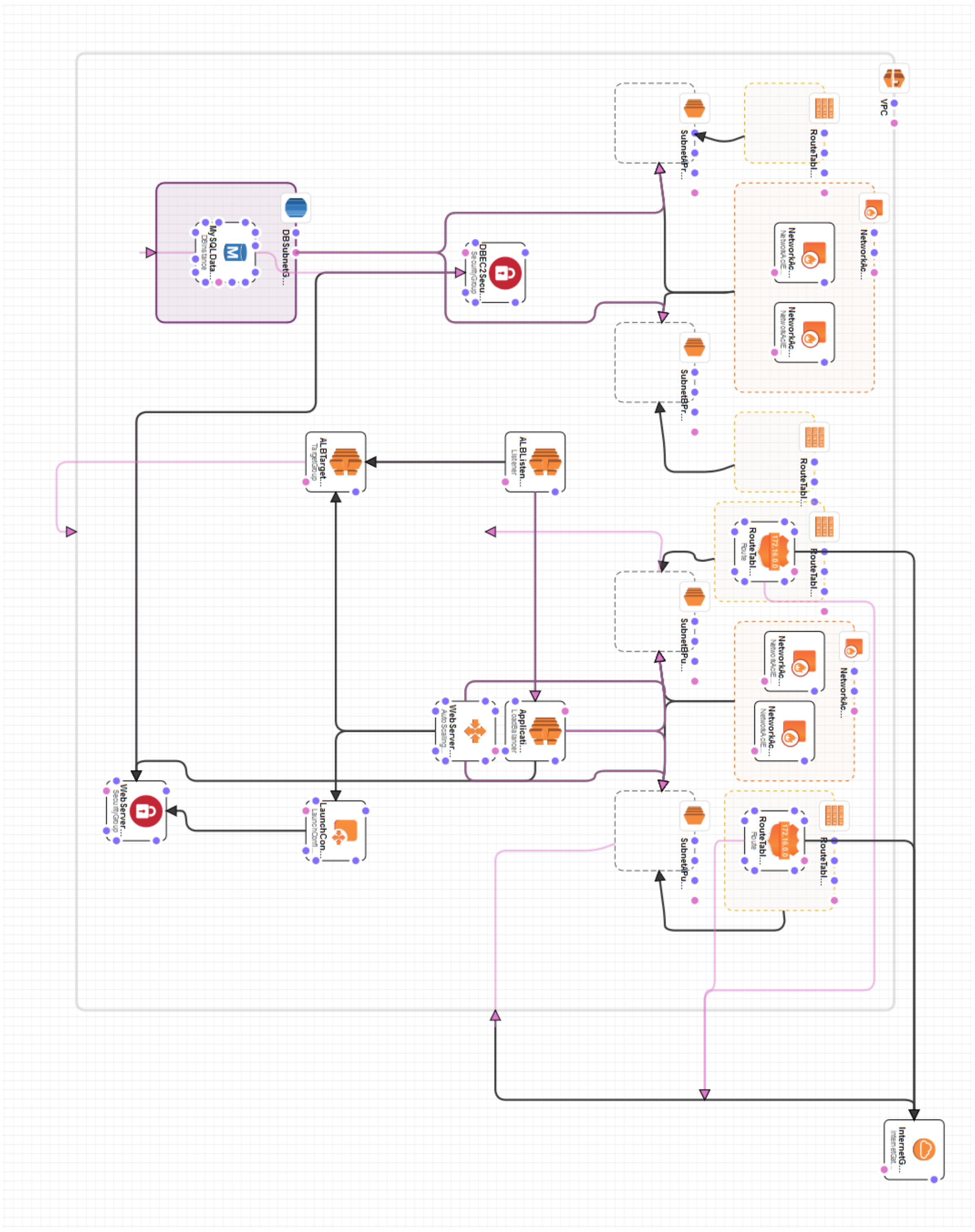
“Ref - AWS CloudFormation.” Accessed May 16, 2018. https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function-reference-ref.html.

Mathew, S. (2014). Overview of Amazon Web Services. [online] Ajsnetworking.com. Available at: http://www.ajsnetworking.com/wp-content/uploads/2015/01/aws-overview.pdf [Accessed 16 May 2018].

Docs.aws.amazon.com. (2018). Designing for High Availability - Exchange Server on AWS. [online] Available at: https://docs.aws.amazon.com/quickstart/latest/exchange/design-ha.html [Accessed 14 May 2018].

Docs.aws.amazon.com. (2018). Creating a Trail - AWS CloudTrail. [online] Available at: https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-a-trail-using-the-console-first-time.html?icmpid=docs_cloudtrail_console [Accessed 14 May 2018].

Appendix I: Cloud Formation Design Diagram


VPC
RouteTabl...
SubnetPr...
NetworkAc...
NetworkAc...
NetworkAcl
NetworkAc...
NetworkAcl
DBSubnetG...
MySQLData...
DBInstance
DBEC2Secu...
SecurityGroup
SubnetBPr...
RouteTabl...
ALBTarget...
TargetGroup
ALBListen...
Listener
RouteTabl...
RouteTabl...
Route
172.16.0.0
SubnetBPu...
NetworkAc...
NetworkAc...
NetworkAcl
NetworkAc...
NetworkAcl
Web Server...
AutoScaling
Applicati...
LoadBalancer
Web Server...
SecurityGroup
LaunchCon...
LaunchConfi...
SubnetPu...
RouteTabl...
RouteTabl...
Route
172.16.0.0
InternetG...
InternetGat...