



London
Stock Exchange Group

Assignment

Part 01

Documentation

Methmi Kaveesha Thantrige

Contents

List of Figures.....	iii
List of Abbreviations	iv
1. Introduction	1
2. System Setup.....	2
2.1. Creating a New Server	2
2.1.1. AWS EC2 Instance Creation	2
2.1.2. SSH into the Public Server	3
2.2. Running a Web Server	5
2.3. Creating the Script for Web Server Status	5
2.4. Create Bash script to Collect Log Files	9
2.4.1. Collect log files and Content of Web Server and Compress	9

List of Figures

Figure 1: Web Sever 6 Instance	2
Figure 2: Instance Summary-Details.....	3
Figure 3: ssh.sh	3
Figure 4: Public server terminal.....	3
Figure 5: Database	4
Figure 6: Status Check with code 200	5
Figure 7: Web Server Enable and Start.....	6
Figure 8: Content Validation.....	6
Figure 9: Save Status to DB	7
Figure 10: WEB_SERVER table	8
Figure 11: Copying log files	9
Figure 12: Compressing log Files	9
Figure 13: Uploading log file to S3 bucket.....	10
Figure 14: S3 Storage.....	10

List of Abbreviations

Table 1: List of Abbreviations

Acronyms	Definition
AWS	Amazon Web Server
EC2	Elastic Compute Cloud
ECU	EC2 Compute Unit
CPU	Central Processing Unit
AMI	Amazon Machine Image
SSH	Secure Shell
HTTP	Hyper Text Transfer Protocol
RDS	Relational Database Service
SSD	Solid State Drive
VPC	Virtual Private Cloud
S3	Simple Storage Service

1. Introduction

As a member of the Capital Markets Application Support/DevOps team I have been tasked to setup following in a cloud based environment,

- Creating a new server which should be publicly accessible.
- Run a web server which is freely available to serve and deploy some content.
- Prepare a script which can be executed externally to the web server to SSH into the public server and check whether the web server is running and server the expected content (in case web server is not running script should start the service.), save the results of the script in a suitable database service along with the timestamp and if any error occurred script should notify it to the App Support team via an email with a suitable error message.
- Prepare another script to run externally and collect log files and content of the web server daily and compress it to a single file and move it to the location where the script is running, upload the compressed file to a storage service accessible via internet or local network, if the file is successfully uploaded remove it from script running location, if not successful notify it to the App Support team via an email.
- Configure proper logging mechanism for the created script or setup.

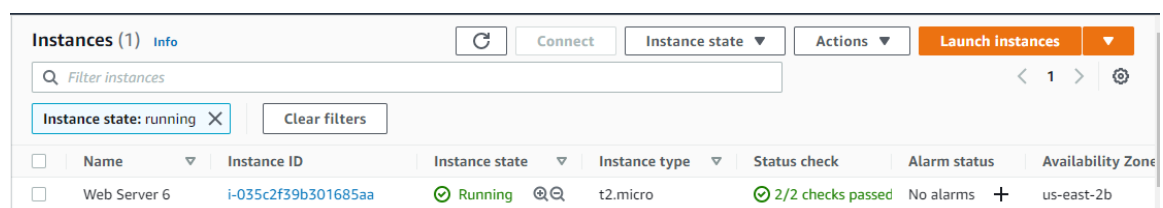
2. System Setup

2.1. Creating a New Server

2.1.1. AWS EC2 Instance Creation

After creating an account in AWS EC2 service was taken under Compute Services. Once Launch Instance was clicked it redirected to the,

- **Step 1:** Required Amazon Machine Image (AMI) should be selected. In this setup Amazon Linux 2 AMI is used as a public server under EC2 compute service of AWS.
- **Step 2:** Instance Type should be selected according to the requirement. t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only) was selected as the instance type.
- **Step 3:** Instance Details should be configured. As default configuration details appeared to be fine, no modifications needed.
- **Step 4:** Storage should be added. Default storage device settings was taken where it allocate 8Gib.
- **Step 5:** Add Tags. One tag was added Key = Name and Value = Web Server 6.
- **Step 6:** Security group should be configured. New security group was created by adding SSH and HTTP as types and made it publicly accessible by setting Source to Anywhere.
- **Step 7:** Review Instance Launch. As the final step all the configurations can be reviewed and modifications can be done if needed. As all appeared to be fine Launch button was clicked.
- While launching the instance it asked for a key pair. As there was no existing key pair clicked on Create new key pair (Web-Test5) and downloaded it. Once all done clicked on Launch Instance and the created instance appeared as in below figure 1.



Instances (1) Info		Refresh	Connect	Instance state ▼	Actions ▼	Launch instances	▼
Filter instances							
Instance state: running X		Clear filters					
<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Web Server 6	i-035c2f39b301685aa	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b

Figure 1: Web Sever 6 Instance

▼ Instance summary Info		
Instance ID i-035c2f39b301685aa (Web Server 6)	Public IPv4 address 18.220.98.155 open address	Private IPv4 addresses 172.31.26.141
Instance state Running	Public IPv4 DNS ec2-18-220-98-155.us-east-2.compute.amazonaws.com open address	Private IPv4 DNS ip-172-31-26-141.us-east-2.compute.internal
Instance type t2.micro	Elastic IP addresses -	VPC ID vpc-76fe621d
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for	IAM Role -	Subnet ID subnet-5528da28

Figure 2: Instance Summary-Details

2.1.2. SSH into the Public Server

Once the instance (virtual server) created, bash script (ssh.sh) was created to make a SSH connection to the Web Server 6 using its public ip and private key from kali Linux machine.

Public IP: 18.220.98.155

Private Key file: Web-Test5.pem

```
#!/bin/bash
webtest.sh

PRIV_KEY=Desktop/Web-Test5.pem
PUB_IP="18.220.98.155"

ssh -i $PRIV_KEY ec2-user@$PUB_IP
```

Figure 3: ssh.sh

```
(root@kali)-[~]
# ./ssh.sh
Last login: Thu Jun  3 08:02:03 2021 from 123.231.106.18

 _ _ | _ _ |
-| ( _ _ /  Amazon Linux 2 AMI
 _| \ _ _ |

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 17 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$
```

Figure 4: Public server terminal

2.1.3. Creating a Database – RDS

Database was created using RDS under database service in AWS. Once click on RDS it redirect to RDS console where we can create new database.

Engine: MySQL

Version: MySQL 8.0.20

Database Instance Identifier: web-database-1

Master Username: admin1

Master Password: awbdb123#\$

Storage Type: General Purpose (SSD)

Allocated Storage: 20Gib

Virtual Private Cloud (VPC): Default as EC2 Instance

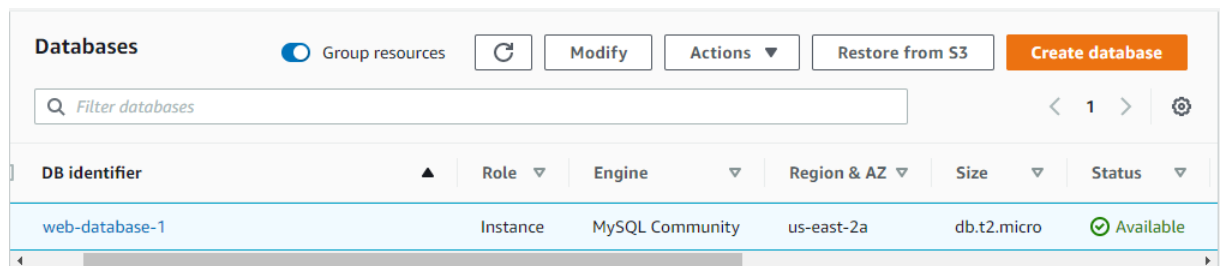
Subnet Group: Default as EC2 Instance

Security Group: Choose Existing one (This security group was edited and newly added rule MySQL-)

Database Authentication: Password Authentication

Initial Database Name: ApplicationDB

Once the above values were set and clicked on Create database it is appeared as follows,



The screenshot shows the AWS RDS 'Databases' console. At the top, there are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and a prominent orange 'Create database' button. Below these is a search bar labeled 'Filter databases'. The main area contains a table with the following columns: 'DB identifier', 'Role', 'Engine', 'Region & AZ', 'Size', and 'Status'. A single database instance is listed with the identifier 'web-database-1', role 'Instance', engine 'MySQL Community', region 'us-east-2a', size 'db.t2.micro', and status 'Available' (indicated by a green checkmark icon).

DB identifier	Role	Engine	Region & AZ	Size	Status
web-database-1	Instance	MySQL Community	us-east-2a	db.t2.micro	Available

Figure 5: Database

2.2. Running a Web Server

Apache is used as a web server.

2.2.1. Install and Start Apache service

- Using yum command apache was installed in Amazon Linux server as super user (root).

```
sudo yum -y httpd
```

- Enabling the httpd service

```
systemctl enable httpd.service
```

- Starting the httpd service

```
systemctl start httpd.service
```

2.2.2. Deploying the Content

In here simple text message 'WELCOME' was put in the index.html file.

```
echo "WELCOME" > /var/www/html/index.html
```

Web page is accessible by the public IP (18.220.98.155) of the server.

2.3. Creating the Script for Web Server Status

2.3.1. Status Check

HTTP status code 200 (OK The request has succeed) was used to check whether the web server is up and running.

Using curl command the response header value was taken, status code was extracted and match it with 200. If matches it will print Server is Up and Running. Meantime new variable called SERVER_STATUS was declared and it can have two values 'up and Running', 'Down'. Relevant value will be assigned inside content validate segment or else segment.

```
#!/bin/bash
PUB_DNS=ec2-18-220-98-155.us-east-2.compute.amazonaws.com
OK="$(curl -Is $PUB_DNS | head -1)"

VALIDATE=( $OK )
DEFAULT_CONTENT=$(cat content.txt)

#Web Server status check
if [ ${VALIDATE[-2]} = "200" ]
then
    echo "Server is Up And Running"
```

Figure 6: Status Check with code 200

If does not match it will enable the Apache service and start the service.

```
else
    echo "Server is Down"
    SERVER_STATUS="Down"
    CONTENT_VALIDATE="Unable to Validate"
    echo "Server is starting..."
    sudo systemctl enable httpd.service
    sudo systemctl start httpd.service
```

Figure 7: Web Server Enable and Start

2.3.2. Content Validate

Validation will be done only if the connection is up and running.

The original content of the web page was taken in to a file called Results.txt and cat that content of the file it in to a variable called DEFAULT_CONTENT.

Current content was taken in to a variable called CONTENT using curl command. Both CONTENT and DEFAULT_CONTENT was matched.

New variable called CONTENT_VALIDATE was declared where it can have three values 'Successful', 'Not Successful' and 'Unable to Validate'.

If both content matched, value Successful was assigned to CONTENT_VALIDATE or else Not Successful. In case of server is down CONTENT_VALIDATE will have the value 'Unable to Validate'.

```
#Content Validate
CONTENT=$(curl -L $PUB_DNS)

if [[ "$CONTENT" == "$DEFAULT_CONTENT" ]]
then
    SERVER_STATUS="Up and Running"
    CONTENT_VALIDATE="Successful"
else
    SERVER_STATUS="Up and Running"
    CONTENT_VALIDATE="Not Successful"
```

Figure 8: Content Validation

2.3.3. Saving the Results to Database

MySQL was installed in Amazon Linux server using yum command

```
yum install -y mysql
```

Once the Database was created (ApplicationDB) as mentioned in 2.1.3. connected to it via Amazon Linux server by giving the password.

```
mysql -h web-database-1.ccswmhkmtl8y.us-east-2.rds.amazonaws.com -  
P 3306 -u admin1 -p
```

In MySQL console WEB_STATUS table was created which has ID, STATUS, DATE_TIME and CONTENT_VALIDATE as columns.

The above declared variable values (SERVER_STATUS and CONTENT_VALIDATE) is inserted to the STATUS and CONTENT_VALIDATE columns respectively. DATE_TIME values is inserted using now() function in MySQL.

```
#Save Status to the DB  
  
RDS_MYSQL_ENDPOINT="web-database-1.ccswmhkmtl8y.us-east-2.rds.amazonaws.com"  
RDS_MYSQL_PORT=3306  
RDS_MYSQL_USER="admin1"  
RDS_MYSQL_PASS="awsdb123#"  
RDS_MYSQL_BASE="ApplicationDB"  
TABLE="WEB_STATUS"  
  
mysql -h $RDS_MYSQL_ENDPOINT -P $RDS_MYSQL_PORT -u $RDS_MYSQL_USER -p$RDS_MYSQL_PASS -D $RDS_MYSQL_BASE -e "INSERT INTO $TABLE (`STATUS`, `DATE_TIME`, `CONTENT_VALIDATE`) VALUES ('$SERVER_STATUS', NOW(), '$CONTENT_VALIDATE')";  
  
DB_STATUS=$?  
  
if [[ $DB_STATUS -eq 0 ]]  
then  
    echo "Status Successfully Saved to the DB"  
else  
    echo "DB Operation: Fail"  
    echo "Cannot write to the Database. Status is not saved." | mail -v -s "Database Operation Fail" mtkaveesha@gmail.com  
fi
```

Figure 9: Save Status to DB

Sample output of WEB_SERVER table

```
MySQL [ApplicationDB]> select * from WEB_STATUS;
```

ID	STATUS	DATE_TIME	CONTENT_VALIDATE
1	Server is up and running	2021-06-01 16:21:40	
2		2021-06-01 16:42:57	
3		2021-06-01 16:55:07	
4		2021-06-01 16:56:41	
5		2021-06-01 16:57:24	
6	Server is Up And Running	2021-06-01 16:58:07	
7	Server is Up And Running	2021-06-01 17:00:32	
8	Up and Running	2021-06-01 23:23:38	Not Successful
9	Up and Running	2021-06-01 23:33:38	Not Successful
10	Up and Running	2021-06-01 23:37:08	Not Successful
11	Up and Running	2021-06-01 23:41:41	Not Successful
12	Up and Running	2021-06-01 23:47:30	Not Successful
13	Up and Running	2021-06-01 23:51:26	Not Successful
14	Up and Running	2021-06-01 23:57:13	Successful
15	Up and Running	2021-06-02 00:05:38	Not Successful
16	Down	2021-06-02 00:19:57	
17	Up and Running	2021-06-02 01:11:48	Not Successful
18	Up and Running	2021-06-02 01:15:19	Successful
19	Up and Running	2021-06-03 08:04:43	Successful
20	Up and Running	2021-06-03 11:23:41	Successful
21	Up and Running	2021-06-03 11:31:46	Successful

21 rows in set (0.00 sec)

Figure 10: WEB_SERVER table

2.3.4. Notify Errors via email

In order to send mails in error detection mail command was used.

Installed mailx package using yum command.

```
yum install -y mailx
```

Once it is installed it was required to configure mail.rc file with below configurations.

```
set smtp=smtps://smtp.gmail.com:465
```

```
set smtp-auth=login
```

```
set smtp-auth-user=methmit25@gmail.com
```

```
set smtp-auth-password=Methmi123!@#
```

```
set ssl-verify=ignore
```

```
set nss-config-dir=/etc/pki/nssdb/
```

Once configuration completed mail can be sent with error message using

‘mail’ command as below

```
echo "Web Server is Down. Service Start comand executed" | mail -v -s
```

```
"Web Server Unavailable" mtkaveesha@gmail.com
```

2.4. Create Bash script to Collect Log Files

2.4.1. Collect log files and Content of Web Server and Compress

There are two main log files in apache named access log and error log. These files were copied in to new files which is named as (timestamp).accessLog.txt and (timestamp).errorLog.txt.

Further content of web server was taken using 'curl' command and put it in a new file named (timestamp).contentLog.txt.

```
#copy apache access log to a new file locate in script running location
sudo cp /var/log/httpd/access_log $LOCATION/$CUR_TIME.accessLog.txt
ACCESS_LOG=$?

#copy apache error log to a new file locate in script running location
sudo cp /var/log/httpd/error_log $LOCATION/$CUR_TIME.errorLog.txt
ERROR_LOG=$?

#Get the content of web and to a new file locate in script running location
curl -L ec2-18-220-98-155.us-east-2.compute.amazonaws.com > $LOCATION/$CUR_TIME.contentLog.txt
CONTENT=$?
```

Figure 11: Copying log files

If these all the command successfully executed then these three files were compressed using 'tar'.

If compressing process was success then (timestamp).accessLog.txt, (timestamp).errorLog.txt and (timestamp).contentLog.txt files were removed.

```
if (( $ACCESS_LOG == 0 && $ERROR_LOG == 0 && $CONTENT == 0 ))
then
    #compress into a single file
    tar -czvf $CUR_TIME.Log_Files.tar.gz $CUR_TIME.accessLog.txt $CUR_TIME.errorLog.txt $CUR_TIME.contentLog.txt
    if [ $? = 0 ]
    then
        echo "$CUR_TIME: $SCRIPT: Log files successfully compressed" >> scriptlog
        rm -f $CUR_TIME.accessLog.txt $CUR_TIME.errorLog.txt $CUR_TIME.contentLog.txt
        #Upload compress file to S3 bucket
        aws s3 cp $LOCATION/$CUR_TIME.Log_Files.tar.gz s3://webservers3/WebServer_LogFiles/
    fi
fi
```

Figure 12: Compressing log Files

2.4.2. Upload the Log file to S3 bucket

Then the compressed file is uploaded to the S3 bucket which is another storage service provided by AWS. Once the S3 bucket called webserver3 is created, a folder called WebServer_LogFiles was created. All the compressed log files will be uploaded to that. Files can be uploaded to it using aws console commands. If the uploading process is successful the compressed file will be removed. If not error notifying email will be sent.

```
#Upload compress file to S3 bucket
aws s3 cp $LOCATION/$CUR_TIME.Log_Files.tar.gz s3://webserver3/WebServer_LogFiles/

if [ $? = 0 ]
then
    echo "Log Files Successfully uploaded."
    rm -f $LOCATION/$CUR_TIME.Log_Files.tar.gz
    echo "$CUR_TIME: $SCRIPT: Log Files Successfully uploaded:Compressed file removed" >> scriptlog
else
    #send notify email
    echo "Unable to upload Log Files." | mail -v -s "Log File Upload Fail" mtkaveesha@gmail.com
    echo "$CUR_TIME: $SCRIPT: Unable to upload Log Files:Notifying email sent" >> scriptlog
```

Figure 13: Uploading log file to S3 bucket

Objects (35)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	2021.06.02-19.29.20.Log_Files.tar.gz	gz	June 3, 2021, 00:59:22 (UTC+05:30)	12.1 KB	Standard
<input type="checkbox"/>	2021.06.02-19.49.09.Log_Files.tar.gz	gz	June 3, 2021, 01:19:10 (UTC+05:30)	12.2 KB	Standard
<input type="checkbox"/>	2021.06.02-19.56.47.Log_Files.tar.gz	gz	June 3, 2021, 01:26:49 (UTC+05:30)	12.2 KB	Standard
<input type="checkbox"/>	2021.06.02-21.02.01.Log_Files.tar.gz	gz	June 3, 2021, 02:32:02 (UTC+05:30)	12.3 KB	Standard
<input type="checkbox"/>	2021.06.02-21.08.04.Log_Files.tar.gz	gz	June 3, 2021, 02:38:05 (UTC+05:30)	12.3 KB	Standard
<input type="checkbox"/>	2021.06.02-21.10.36.Log_Files.tar.gz	gz	June 3, 2021, 02:40:37 (UTC+05:30)	12.3 KB	Standard
<input type="checkbox"/>	2021.06.02-21.11.47.Log_Files.tar.gz	gz	June 3, 2021, 02:41:49 (UTC+05:30)	12.3 KB	Standard
<input type="checkbox"/>	2021.06.02-21.13.00.Log_Files.tar.gz	gz	June 3, 2021, 02:43:01 (UTC+05:30)	12.3 KB	Standard
<input type="checkbox"/>	2021.06.02-21.14.43.Log_Files.tar.gz	gz	June 3, 2021, 02:44:44 (UTC+05:30)	12.3 KB	Standard

Figure 14: S3 Storage

10

In order to perform above operations daily this script was set to execute in cron jobs.

Open the crontab file

```
Crontab -e
```

Inside the file job was set to execute the script daily midnight.

```
0 0 * * * /home/ec2-user/logcollect.sh
```

2.5. Logging Mechanism for Script/Setup

The results of each script is echoed to a file called scriptlog along with timestamp and script name.

Format of log record is as follows,

Timestamp: Script Name: Message

Sample Command:

```
echo "$CUR_TIME: $SCRIPT: Log Files Successfully uploaded:Compressed  
file removed" >> scriptlog
```

Sample Log Record:

```
2021.06.03-11.41.33: logcollect.sh: Log Files Successfully uploaded:  
Compressed file removed
```

3. Assumption

If Web server http status code 200 only the server is up and running. If it is other than 200 the server is down.