|  |  |
| --- | --- |
| 30-05-2022 | **Assignment part 1 - Scripting** |

|  |  |
| --- | --- |
| **Madushan Kulathilaka**  94 77 333 2093  [madu8k@gmail.com](mailto:madu8k@gmail.com) |  |

**Table of Contents**

[**1.** **Architecture Diagram**](#_Toc104826531)

[**2.** **AWS EC2**](#_Toc104826532)

[**3.** **Docker environment**](#_Toc104826533)

[**4.** **Mariadb DataBase**](#_Toc104826534)

[**5.** **Nginx**](#_Toc104826535)

[**6.** **Script**](#_Toc104826536)

[**7.** **Email server.**](#_Toc104826537)

[**8.** **Logging Mechanism for Script**](#_Toc104826538)

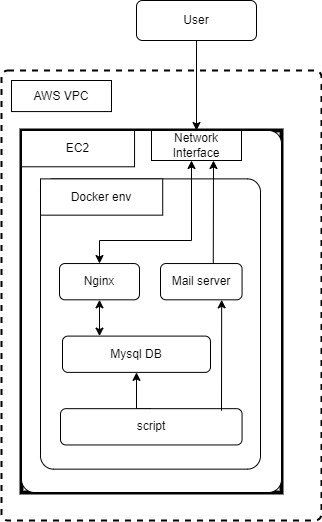
[**9.** **Crontab configuration**](#_Toc104826539)

[**10.** **Monitoring Areas**](#_Toc104826540)

[**11.**  **Summery**](#_Toc104826541)

# Architecture Diagram

This script run in AWS cloud Containerized Environment.



# AWS EC2

AWS is the most convenient cloud provider for this system. After implementing this setup, can create an AWS EC2 Image using Beanstalk and CloudFormation. It will useful to rebuild this setup if any failure.

1st Step is creating AWS EC2 considering below.

* Amazon Machine Image (AMI) is Red Hat Enterprise Linux 8
* Create security group and Allow HTTP/HTTPs traffic from the internet

# Docker environment

Docker is a software platform for building applications based on containers.

Follow these steps in RHEL 8 flatform

* *Sudo su*
* *yum -y update*
* *yum-config-manager --add-repo* [*https://download.docker.com/linux/centos/docker-ce.repo*](https://download.docker.com/linux/centos/docker-ce.repo)
* *yum install -y docker-ce docker-ce-cli containerd.io*
* *systemctl start docker*
* *systemctl enable docker*

# Mariadb DataBase

Mariadb use as the database. These are the steps for install Mariadb in container and create database

* *docker pull mariadb*
* *docker run -e MYSQL\_ROOT\_PASSWORD=test --name mydbcontainer -p 3306:3306 -d mariadb*
* *docker exec -it mydbcontainer mysql -u root -ptest*
* *CREATE database server\_datetime;*

For access database need to install Mariadb client, use below command for check database host ip. Default ip is 172.17.0.2

*docker inspect mydbcontainer*

These steps for config YUM package repository.

* *wget* [*https://downloads.mariadb.com/MariaDB/mariadb\_repo\_setup*](https://downloads.mariadb.com/MariaDB/mariadb_repo_setup)
* *echo "b9e90cde27affc2a44f9fc60e302ccfcacf71f4ae02071f30d570e6048c28597 mariadb\_repo\_setup" \*

*| sha256sum -c –*

* *chmod +x mariadb\_repo\_setup*
* *sudo ./mariadb\_repo\_setup \*

*--mariadb-server-version="mariadb-10.6"*

Run these commands for check access and create database and table. If default host IP address is different, can be changed by below 2nd command.

* *yum install -y MariaDB-client*
* *mysql -h 172.17.0.2 -u root -ptest*
* *USE server\_datetime;*
* *CREATE TABLE sdlog (*

*ID int NOT NULL AUTO\_INCREMENT PRIMARY KEY,*

*error\_code varchar(255),*

*error\_time DATETIME );*

# Nginx

Nginx is the hosting server for this program

*docker run -d --publish 80:80 -v '/root/webfile:/usr/share/nginx/html' -v '/var/log//nginx/:/var/log/nginx/' --name webhost1 nginx*

# Script

There are two scripts in this program which called as Webhosting and error logging

**Webhosting script**

This is html script.it should add to ‘*/root/webfile’* path.

Html file attached here with.

**Error logging script**

This is bash script. Follow these steps for create this script.

* *Cd /home/ec2-user/*
* *touch update\_table.sh*
* *vi update\_table.sh*
* copy this below script and run script using *./update\_table.sh &*

# ! /bin/bash

while [ TRUE ]

do

ERR\_TIME=`cat  /var/log/nginx/access.log | tail -1  | cut -d " " -f4 | cut -d ":" -f2,3,4`

SRV\_TIME=`date +'%H:%M:%S'`

FC=`cat /var/log/nginx/access.log | wc -l`

echo $FC

ERROR=`cat  /var/log/nginx/access.log  | cut -d " " -f9 | tail -1`

if [ "$FC" != "$NC" ] && [ "$ERROR" != "200" ]

    then

    ERROR=`cat  /var/log/nginx/access.log  | cut -d " " -f9 | tail -1`

    echo $ERROR

    TIME=`date +'%y%m%d %H%M%S'`

    echo $TIME

    mysql -h 172.17.0.2 -u root -ptest server\_datetime -e 'insert into sdatetime (error\_code,error\_time) values ('${ERROR}','${TIME}')'

else

    echo "Log not updating"

fi

NC=`cat /var/log/nginx/access.log | wc -l`

echo $NC

done

# Email server.

The mail command in Linux will make use of the local SMTP server running on port 25 for sending mails. But if you want to send mails using external SMTP server such as smtp.gmail.com or SMTP server of your organization, then you can use mailx command. Using mailx command is quite easy and all you need to do is install mailx package and pass few SMTP related arguments to the command

* *yum install mailx*
* *echo "The actual message goes here" | mailx -v -r "user@domain.com" -s "The actual subject line goes here" -S smtp="smtp.domain.com:587" -S smtp-use-starttls -S smtp-auth=login -S smtp-auth-user="user@domain.com" -S smtp-auth-password="password123" -S ssl-verify=ignore the\_recipient\_email@domain.com*

# Logging Mechanism for Script

We can use API call in JavaScript for logging this script. In this program, I create password alert window.

Password is: 1234

# Crontab configuration

For availability of website in required time period (9 AM to 6 PM), need to create job in crontab.

(Server clock run UTC time frame. So, need to get business location time zone and should match it)

*crontab -e*

and update this and save

30 3 \* \* \* docker run nginx

30 12 \* \* \* docker stop nginx

# Monitoring Areas

* Ec2 health
* Container logs
* Cron logs

# 11. Summery

Cloud computing is probably the most cost-effective method to use, maintain, and upgrade your IT infrastructures. There are many pay-as-you-go packages and other scalable options available, which make it very reasonable for businesses of any size to switch from traditional on-premises hardware to the cloud.

Amazon Web Services (AWS) owns a large portion of the market share for cloud computing and offers a large feature set, including support for analytics, storage, developer tools, and security.

Docker technology is also more controllable, more granular and is a microservices-based method focused on efficiency. It can develop lots of server in one EC2. Maintain can be done without offline whole application.