1. **Set up the aws cloud environment**
   1. Create two ec2-server for the admin server and the webserver. Both have public access.
   2. Admin server- To run shell scripts created. (Installed git using user data section when creating the instance)
   3. Webserver- To run apache. (Installed mysql, ssmtp using user data section when creating the instance).
   4. Deployed a rds mysl instance to save results of this script in the database service along with the timestamps.
   5. Deployed a S3 bucket ([apache-webserver-logs](https://s3.console.aws.amazon.com/s3/buckets/apache-webserver-logs?region=us-east-1) )to upload the compressed file.
   6. Additional configurations are added to user data files.(to install when instances creating)
   7. Create an iam role called”webserver-s3-access” and attach “AmazonS3FullAccess” policy.(use ec2 as aws service)
2. **Created suitable shell scripts to achieve following tasks**
3. check\_webserver\_content.sh - Check if the server is serving the expected content (part1)
4. check\_webserver\_status.sh - check if Web server is running and start it if it is not (part1)
5. creating\_compressed\_file.sh - Collect log files and content of the web server daily and create one compressed file (part2)
6. move\_compressed\_file.sh - Move the compressed file to the same location as the script is running(part2)
7. uploading\_to\_s3.sh - Upload the file to a S3 bucket (part2)

**main\_1.sh - To achieve the task in part1**

* check\_webserver\_content.sh
* check\_webserver\_status.sh

**main\_2.sh - To achieve the tasks in part2**

* creating\_compressed\_file.sh
* move\_compressed\_file.sh
* uploading\_to\_s3.sh

1. **Created ansible two ansible scripts and saved locally. (Ansible should be installed in your local pc)**
   1. copy\_key.playbook.yml – To copy the pem key to admin server.(one time only)
   2. web\_playbook.yml – To install apache on webserver
   3. scripts\_playbook.yml – To Run shell scripts in admin bash server.
2. **Created a table in mysql database with three columns**

**Table\_name-** web\_details

**Columns**- created\_at, status, status\_code

Table

Description automatically generated

1. **Autoscaling has been used to ensure availability and reliability of this setup.**

**Launch configurations -** webserver-launchtemplate, adminserver-launchtemplate

**Auto Scaling groups -** webserver-auto-scaling-group, adminserver-auto-scaling-group

1. **Ports opened in security groups**

Webserver- 22, 80

Admin – 22

Mysql database- 3306

1. **Instructions to run scripts**

Add webserver ip and admin server ip in **inventory** file.

Edit the main\_1.sh and main\_2.sh scripts and add your **webserver ip** address in ssh part.

All the shell scripts are exectuted through ansible server. (from your local pc). Ansible scripts should run locally.

Install ansible on your local pc and run below two playbooks from your local pc. Please check aws infrastructure has been deployed correctly before running the two playbooks.

You need to run **web\_playbook.yml** first.

Secondly you need to run the **scripts\_playbook.yml**

**Note - copy\_key.playbook.yml** executed only one time.(when creating a new environment)

You can use following commands to run ansible scripts

**$***ansible-playbook web\_playbook.yml*

$*ansible-playbook* scripts\_playbook.yml

1. **Instructions to set up in a new environment**

1.Edit the **copy\_key.playbook.yml** according to your details.

Add your **pem key** path in the **src field** as shown in the below figure. Do not change the destination file name and the file path.

**Text

Description automatically generated**

2. Edit user data files

Add your local pc public key(where you run ansible scripts) to the user data files accordingly. Two separate user data files are available in the git hub repo for admin server and the webserver.

3. Create **webserver-launchtemplate** and **adminserver-launchtemplate** in your aws environment.

Add followings to **webserver-launchtemplate**

* User data file- user-data-file-webserver (available in GitHub)
* Pem key - (your pem key file)
* security group with open ports 22,80
* Attach **webserver-s3-access** iam roles

Add followings to **adminserver-launchtemplate**

* User data file- user-data-file-adminserver
* Pem key - (your pem key file)
* security group with open ports 22

Create two auto scalling groups **-** webserver-auto-scaling-group, adminserver-auto-scaling-group

Finally you can runweb\_playbook.yml and scripts\_playbook.yml separately.

**Note- If accidently instances got terminated, new instances will be automatically launched. ( Auto scaling grops have been created).**

**After a termination of instances, you have to run the three ansible playbooks.**

**Edit the main\_1.sh, main\_2.sh and add new webserver ip address.**