

1. Set up the aws cloud environment

- a. Create two ec2-server for the admin server and the webserver. Both have public access.
- b. Admin server- To run shell scripts created. (Installed git using user data section when creating the instance)
- c. Webserver- To run apache. (Installed mysql, ssmtp using user data section when creating the instance).
- d. Deployed a rds mysql instance to save results of this script in the database service along with the timestamps.
- e. Deployed a S3 bucket (apache-webserver-logs)to upload the compressed file.
- f. Additional configurations are added to user data files.(to install when instances creating)
- g. 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

- I. `check_webserver_content.sh` - Check if the server is serving the expected content (part1)
- II. `check_webserver_status.sh` - check if Web server is running and start it if it is not (part1)
- III. `creating_compressed_file.sh` - Collect log files and content of the web server daily and create one compressed file (part2)
- IV. `move_compressed_file.sh` - Move the compressed file to the same location as the script is running(part2)
- V. `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`

3. Created ansible two ansible scripts and saved locally. (Ansible should be installed in your local pc)

- a. `copy_key.playbook.yml` – To copy the pem key to admin server.(one time only)
- b. `web_playbook.yml` – To install apache on webserver
- c. `scripts_playbook.yml` – To Run shell scripts in admin bash server.

4. Created a table in mysql database with three columns

Table_name- web_details

Columns- created_at, status, status_code

	created_at	status	status_code	
	2021-03-08 09:04:46	running	200	
	2021-03-08 08:58:29	running	200	
	2021-03-07 20:10:41	running	200	
▶	2021-03-07 20:02:46	running	200	
	2021-03-07 06:57:16	running	200	
	2021-03-07 06:56:24	running	200	
	2021-03-07 06:46:10	running	200	
	2021-03-06 18:18:40	running	200	
	2021-03-06 18:18:20	running	200	

5. 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

6. Ports opened in security groups

Webserver- 22, 80

Admin – 22

Mysql database- 3306

7. 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 executed 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
```

8. 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.

```
[jalitha@MacBook-Pro keys % vi copy_key.playbook.yml
--
- name: install pre-requirities on webserver
  hosts: admin
  remote_user: ec2-user
  become: yes
  tasks:
    - name:
      copy:
        src: /Users/jalitha/Documents/keys/lseg.pem
        dest: /home/ec2-user/lseg
~
```

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 run web_playbook.yml and scripts_playbook.yml separately.

Note- If accidently instances got terminated, new instances will be automatically launched. (Auto scaling groups 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.

