**Q. Certain web pages are loading slow in user’s browser for our live web application. What steps will you take to resolve the issue?**

**Solution:**

There could be several issues regarding live web application slowness on user’s browser.

First of all, issues can be on Web Application Server, where web application server unable to serve requests from user’s, if there is continuous requests from random user’s.

In that case, Autoscaling of web server can be implemented in case of AWS, GSuite etc. Where, if web applications usage limit (CPU Utilization, Network Utilization) crosses as per configured policy, web application will automatically scale up to its desired configuration. If usage reduces, web application will automatically scale down to its minimum configuration. Also, integrated health-check will identify failed node, and automatically replica web application server will be created.

Secondly, issues can be on web application server geographic location. In that case, caching solution can be implemented, where if web server is in asia and user from Europe access the web application server, then the content will be cached to Europe nearest POP of AWS or GSuite. When any other user will browse the same content, the content will be transferred from cache location rather than backend web application server.

Thirdly, Only Load-Balancing solution can be implemented with target groups with several compute instances.

For this scenario, feasible solution will be **Autoscaling** as per my personal opinion.

**Q. Imagine a scenario where a web application is serving from a single web server to the internet. What are the problems in this scenario? Design and architect a solution that will mitigate these problems? Or How would you design a scalable architecture with resiliency in mind for the following situations:**

**a. if a service is resource intensive b. a service needs to be low latency c. if parts of a service need to be restricted to certain geographical boundaries**

Solution:

Problems of the scenario are:

1. Single point of failure.
2. High Resource utilization

To resolve above problems, Load-Balancer can be deployed at front of backend application. Where multiple server will serve the same application under single application group or multiple application group. User’s of the application will able to reach server via single frontend ip.

Also Auto-scaling can be implemented for the same purpose. Where, backend servers will autoscale to its minimum and maximum resource limit as per the utilization. Here, auto-scaling policy can be based on Resource utilization or Latency.

Aside, for the certain geographical location, auto-scaling groups can be set to specific geographical location. If parts of service like database can be implemented in certain geographical location with auto-scaling policy implemented.

**Q. Currently there’s no monitoring in place for the above single web server. How and what application will you use to monitor the resources/process in your new design?**

Solution:

For monitoring, if application is running in AWS, Cloud Watch can be used for Monitoring.

If application is running in Google Cloud, Stackdriver can be used for Monitoring.

**Q. In our server we want to create a user who can only view logs using `less` from this path /var/log. Please explain how to achieve this.**

**Solution:**

**Create a copy of /bin/bash as rbash:**

- cp /bin/bash /bin/rbash

**First create a group named as log:**

- groupadd log

**Create a user named as test in group log:**

- useradd -s /bin/rbash test -g log; passwd test

**Change the /var/log ownership to root:log:**

- chown -R root:log log/

**Create a directory under /var/log:**

- cd /var/log; mkdir cmd

**Change the user test .bash\_profile PATH variable as follows:**

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

# User specific environment and startup programs

PATH=$HOME/cmd

export PATH

**Change the test user home directory to /var/log:**

- usermod -d /var/log/ test

**Change the /var/log/ permission to 650:**

- chmod -R 650 /var/log/

**Copy contents of /home/test/ into /var/log/**

- cp -f /home/test/.bash\* /var/log

**Create a softlink of /usr/bin/less into /var/log/cmd**

- ln -s /usr/bin/less /var/log/cmd

Above process will restrict user test to use only less command under its home directory /var/log. As test user’s .bash\_profile PATH variable only set to /var/log/cmd directory. Where it will only get the command less.

Aside, test user under log group. Where log group has read and execution permission into /var/log directory. So, user test can read any file under /var/log.

1. **Explain how you can ssh into a private server from the internet.**

Solution:

If private server under a VPN gateway (PPTP or other secure VPN Gateway), by connect to that VPN gateway securely, we can access a private server.

On the other hand, disable the password authentication of the server and using a secure key pair user’s can access the server via ssh. Only user with secure key pair can access the server.

Aside, a seperate jumphost can be created for this scenario. If ssh to any server, users need to use VPN connection to access jumphost. From jumphost, user can ssh to any private server which is accessible from jumphost.

1. **Write a bash function that will find all occurrences of an IPv4 from a given file.**

**Solution:**

#!/bin/bash

read -p "Enter File Location:" -e FILE

#echo $FILE

grep -E -o "([0-9]{1,3}[\.]){3}[0-9]{1,3}" $FILE

**Q . Share with us a steps to run a web service container on 80 port.**

**Solution:**

- docker pull nginx

#This will print out pulled docker image.

- docker images

# Update repository and install necessary application and start nginx service

- docker run nginx /bin/bash -c “apt update; apt install net-tools; netstat -tulnp; /etc/init.d/nginx start; ifconfig”

- curl [container IP]

- docker ps -a

- docker commit [CONTAINER ID] docker-image-name

Or,

**Search for similar images for httpd:**

- docker search httpd

NAME DESCRIPTION STARS OFFICIAL AUTOMATED

httpd The Apache HTTP Server Project 2713 [OK]

centos/httpd 26 [OK]

....................

**Pull Httpd image from docker :**

- docker pull httpd

**Checked pulled docker image:**

**-** docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

httpd latest d3017f59d5e2 4 days ago 165MB

- docker run httpd /bin/bash -c “pwd”

- docker run httpd /bin/bash -c “/usr/local/apache2/bin/httpd &”

- curl [Container\_IP]