Sr.SRE Question and Answers

Linux USE CASE Interview Question and Answers :

1. **Can you describe a time when you had to troubleshoot a production issue and how you approached it?**

- In my previous role at company X, we experienced a spike in latency across our entire system. I began by reviewing server and application logs, as well as monitoring metrics, to determine the source of the issue. I discovered that one of our database servers was running low on disk space, causing a bottleneck. I quickly resolved the issue by adding more storage to the server and implementing a process for monitoring disk space on a regular basis.

2. **How would you go about improving the reliability of a service?**

- I would first start by identifying the current reliability metric of the service and then establish a baseline to measure improvements against. After that, I would conduct a thorough analysis of the service to identify potential bottlenecks and failure points. Once I have identified these areas, I would implement appropriate strategies such as increasing redundancy, implementing circuit breakers, and implementing proactive monitoring.

3. **How would you go about troubleshooting performance issues in a distributed system?**

- To troubleshoot performance issues in a distributed system, I would first begin by identifying the specific service or component that is experiencing poor performance. I would then gather system and application-level metrics to understand the resources that are being consumed, such as CPU, memory, and network usage. Once I have identified the resources that are being consumed, I would work to isolate the cause of the issue. If it is a network issue I would work with network team, if it is a database issue I would work with DBA team. After identifying the root cause, I would work to implement a solution, such as adding more resources or implementing a caching layer.

4. **How do you approach capacity planning for a service?**

- I would begin by understanding the current usage patterns and growth trends of the service. This would include gathering data on traffic, resource usage, and response times. I would then use this data to create a model of expected future usage. Based on the results of the model, I would make recommendations for adding resources, such as servers or bandwidth, in advance of reaching capacity. Finally, I would implement monitoring and alerting to notify when the capacity usage is close to the predefined threshold.

5. **How do you deal with unplanned outages?**

- When an unplanned outage occurs, my first priority is to quickly assess the situation and gather information about the cause of the issue. I would then work with the appropriate team members to implement a fix as soon as possible. At the same time, I would also

communicate with stakeholders to keep them informed about the status of the issue, and implement any mitigations to minimize the impact of the outage. Once the issue is resolved, I would conduct a post-mortem analysis to understand the root cause and identify opportunities for prevention in the future.

6. **How do you troubleshoot high CPU usage on a Linux server?**

- To troubleshoot high CPU usage on a Linux server, I would begin by using the top or htop command to identify the processes that are consuming the most CPU. I would then use the ps command with the -o option to display the process in a user-defined format to get the PID of the process. Then I would use the strace or ltrace command to trace the system calls made by the process and identify the source of the issue. If the issue is related to a particular application, I would then investigate the application's log files and configuration settings to identify any problems.

7. **How do you optimize Linux server performance?**

- There are many ways to optimize Linux server performance, but some of the common ones are:
- Tuning the kernel parameters and sysctl settings
- Optimizing file system mount options
- Enabling transparent huge pages
- Decreasing swappiness
- Turning off unnecessary services
- Configuring the correct I/O scheduler
- Setting up a RAM disk for frequently accessed files
- Providing enough RAM and ensure appropriate usage of swap space

8. **How do you troubleshoot slow disk I/O on a Linux server?**

- To troubleshoot slow disk I/O on a Linux server, I would first use the iostat command to monitor disk I/O statistics and identify which disk or partition is experiencing slow I/O. I would then use the dstat command to get the detailed statistics about disk usage and find out the cause of the problem whether it's a high number of read/write requests, low throughput, or high latency. I would also use the top or htop command to identify any processes that are consuming a large amount of I/O. Based on the findings, I would take appropriate actions like tuning the filesystem mount options, changing the I/O scheduler, checking for hardware issues, or optimizing the workload to decrease the I/O contention.

9. **How do you monitor and optimize network performance on a Linux server?**

- To monitor and optimize network performance on a Linux server, I would use the netstat command to check the network connections and view the statistics of the network interfaces. I would use the ifconfig or ip addr show command to check the status of the network interfaces, including their current IP address, netmask, and MTU. I would also use the tc command to configure and view the traffic control settings and iptables or nftables to configure and view the firewall rules. I would also use tools like tcpdump or wireshark to capture and analyze network packets. Based on the findings, I would take

appropriate actions to optimize the network performance, such as adjusting the MTU, tweaking the TCP settings, or reconfiguring the firewall rules.

10. How do you setup a high availability cluster in Linux?

- To set up a high availability cluster in Linux, I would use a cluster management software like Pacemaker or Corosync, which provides a set of tools to manage the resources and failover of a cluster. The basic steps of setting up a high availability cluster would include:
- Installing and configuring the cluster management software
- Creating a shared storage for the cluster
- Creating a virtual IP address
- Config

11. How would you go about troubleshooting a Linux server that's running out of memory?

- To troubleshoot a Linux server that's running out of memory, I would begin by using the free or vmstat command to check the server's current memory usage and see if it is indeed running out of memory. I would then use the top or htop command to identify which processes are consuming the most memory. I would use the ps command with -o option to display the process in a user-defined format to get the PID of the process. I would then use strace or ltrace command to trace the system calls made by the process and identify the source of the issue. I would also look into swapping or page faults with the command vmstat 1, and check for any out of memory killer. Based on the findings, I would take appropriate actions to address the issue, such as killing the memory-hungry processes, adjusting memory limits, or adding more memory to the server.

12. How would you go about setting up a Linux-based firewall and configure rules to limit incoming traffic?

- To set up a Linux-based firewall and configure rules to limit incoming traffic, I would use the iptables or nftables command-line utilities. The basic steps of setting up a Linux-based firewall would include:
- Creating a new firewall chain for incoming traffic
- Setting the default policy for the chain to DROP
- Creating rules to allow specific incoming traffic based on IP addresses, ports, and protocols
- Using the -A option to append the rules to the chain
- Saving the firewall rules to make them persistent across reboot.

13. How would you handle a situation in which there is a sudden increase in traffic on a website?

- In the event of a sudden increase in traffic on a website, my first step would be to gather data about the traffic to understand the scale of the increase and identify any patterns or anomalies. I would then use this information to quickly scale up resources such as adding more servers, increasing CPU and memory, or optimizing the database

configurations. I would also work on improving the website's caching and content delivery network(CDN) to offload traffic and improve response time. I would also use monitoring tools to check resource utilization and detect bottlenecks. Once the traffic increase has been addressed, I would also work on implementing long-term solutions to prevent similar issues in the future.

14. **How would you go about troubleshooting a network connectivity issue on a Linux server?**

- To troubleshoot a network connectivity issue on a Linux server, I would begin by using the ping command to check the server's connectivity to other hosts. I would also use the traceroute command to check the path that packets take to reach a specific host and identify any routing issues. I would also use the netstat command to check the status of the network connections and view the statistics of the network interfaces. I would also use the ifconfig or ip addr show command to check the status of the network interfaces. I would also check the firewall rules using the iptables or nftables command. Based on the findings, I would take appropriate actions to address the issue, such as reconfiguring the network settings, adjusting the firewall rules or replacing faulty hardware.

15. **How do you manage software updates and package installations on a Linux server?**

- To manage software updates and package installations on a Linux server, I would use a package manager like apt-get, yum, or dnf, depending on the Linux distribution. I would also use tools such as apt-get upgrade, yum update, dnf upgrade, etc to update all the packages to the latest version. I would also use tools like apt-get dist-upgrade to update packages to new releases. I would also use tools such as apt-get install, yum install, or dnf install to install new packages and tools. I would also make sure to test any updates and new packages on a test or staging server before deploying them to production.

16. **How do you perform security hardening on a Linux server?**

- To perform security hardening on a Linux server, there are several steps I would take such as:
- Remove unneeded services and packages
- Use a firewall, like iptables or nftables, to block unwanted incoming traffic
- Disable unnecessary users and groups
- Set up and configure intrusion detection and prevention systems (IDS/IPS)
- Regularly update the operating system and installed software
- Implement appropriate file permissions and ownership
- Configure SSH for key-based authentication
- Use SELinux or AppArmor for additional security layer
- Use tools like tripwire, aide, or samhain to detect any changes in the file system
- Regularly monitor logs for any signs of intrusion or suspicious activity.

Monitoring tools : Based on resume we can ask below :

1. **Can you explain your experience with monitoring tools such as Prometheus and Grafana?**

- I have experience using Prometheus and Grafana for monitoring systems and applications. I have set up Prometheus instances to collect metrics from various systems and services, and configured alerts based on specific thresholds. I have also used Grafana to visualize the metrics collected by Prometheus and create dashboards for real-time monitoring and alerting. I have also used Grafana to create alerts based on the metric and notify the respective team to take action.

2. **How do you use Nagios or Zabbix for monitoring and alerting?**

- I have experience using Nagios or Zabbix for monitoring and alerting. I have set up Nagios or Zabbix instances to monitor various systems and services, and configured alerts based on specific thresholds. I have also written custom plugins to monitor services that are not natively supported by Nagios or Zabbix. Additionally, I have also set up notifications to alert the respective team on critical issues.

3. **How do you use Logstash, Elasticsearch and Kibana (ELK Stack) for log analysis and troubleshooting?**

- I have experience using the ELK Stack (Elasticsearch, Logstash, and Kibana) for log analysis and troubleshooting. I have set up Logstash instances to collect logs from various systems and services, and configured Elasticsearch to store and index the logs. I have also used Kibana to create visualizations and dashboards for analyzing and identifying patterns in the logs. Additionally, I have used Kibana to create alerts based on the log data, and to notify the respective team to take action.

4. **Can you explain your experience with APM (Application Performance Management) tools such as New Relic and Datadog?**

- I have experience using APM (Application Performance Management) tools such as New Relic and Datadog. I have set up New Relic or Datadog instances to monitor various applications, and have used the tools to identify and troubleshoot performance issues such as slow requests, high error rates, and high CPU or memory usage. I have also used these tools to create alerts based on specific thresholds, and to notify the respective team to take action.

5. **How do you use monitoring and observability tools like Splunk, Prometheus and Grafana together?**

- I have experience using a combination of monitoring and observability tools such as Splunk, Prometheus, and Grafana. I have set up Splunk to collect and index logs, Prometheus to collect metrics, and Grafana to create visualizations and dashboards. By using all three tools together, I can correlate metrics and logs to get a complete.

Application troubleshooting few questions:

1. **What steps do you take when a web page is not loading?**

- First, I would check to see if the page is down for everyone or just me by using a website such as IsItDownRightNow.com. Next, I would check the browser's developer console for any error messages. If there are none, I would check to see if the page is being blocked by a firewall or proxy. If the problem persists, I would look at the web server's error logs to see if there is any indication of the issue.

2. **How do you troubleshoot a slow web page?**

- To troubleshoot a slow web page, I would first check the page's load time using a tool such as Google's PageSpeed Insights. Next, I would check the network tab in the browser's developer console to see if there are any large or slow resources being loaded. I would also check to see if the web server is under heavy load. If so, I would look into optimizing the database queries and implementing caching mechanisms to improve performance.

3. **How do you troubleshoot a "404 Error" on a web page?**

- A "404 Error" means that the page cannot be found on the server. To troubleshoot this error, I would first check to make sure that the URL entered is correct. Next, I would check the server's error logs to see if there is any information about the missing page. If the page should exist, I would check the server's configuration to make sure that the correct mapping for the URL is set up. If all else fails, I would reach out to the website owner/maintainer to see if they are aware of the issue or if they can point me to the right direction.

4. **How do you troubleshoot a "500 Error" on a web pag**e?

- A "500 Error" usually means there is a server-side error. To troubleshoot this error, I would first check the server's error logs for more information about the issue. I would also check the browser's developer console for any error messages. If there is a problem with a specific script, I would check the script for syntax errors or other issues. In case the error persist, I would make sure that all server components such as web server, database server and other services are running as expected.

5. **How do you debug a cross-browser compatibility issue?**

- To debug a cross-browser compatibility issue, I would first reproduce the issue in as many browsers as possible to determine which ones are affected. Next, I would use browser developer tools to examine the HTML, CSS, and JavaScript of the page in each affected browser. I would then look for any differences between the implementations in each browser that could be causing the issue. Finally, I would test the page using browser compatibility testing tools, and make necessary changes to ensure that the page works as expected across all browsers.

Shell Scripting Questions (If he/she is Good in Scripting you can ask one or two from below:

1. **Write a shell script that takes a file name as an argument and renames it by adding the current date and time to the file name.**

```
#!/bin/bash

file=$1

if [ -f $file ]; then
  timestamp=$(date +%Y%m%d_%H%M%S)
  new_file="${file%.*}_$timestamp.${file##*.}"
  mv $file $new_file
  echo "File $file has been renamed to $new_file"
else
  echo "File $file does not exist"
fi
```

2. Write a shell script that takes a directory path as an argument and prints all the files in the directory and its subdirectories.

```
#!/bin/bash

path=$1

if [ -d $path ]; then
  for file in $(find $path -type f); do
    echo $file
  done
else
  echo "Directory $path does not exist"
Fi
```

3. Write a shell script that takes a URL as an argument and checks if the website is up or down

```
#!/bin/bash

url=$1

status_code=$(curl -s -o /dev/null -w "%{http_code}" $url)

if [ $status_code -eq 200 ]; then
  echo "Website $url is up"
else
  echo "Website $url is down"
fi
```

4. Write a shell script that takes a process name as an argument and checks if the process is running or not.

```
#!/bin/bash

process=$1
```

```
if pgrep -x "$process" > /dev/null; then
  echo "Process $process is running"
else
  echo "
```

If a Candidate is Good in Python Scripting you can ask any one or two from below:

1. Write a Python program to check if a given string is a palindrome.

```
def is_palindrome(string):
    return string == string[::-1]

print(is_palindrome("racecar")) # True
print(is_palindrome("hello")) # False
```

2. Write a Python program to find the common elements between two lists.

```
list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]

def common_elements(list1, list2):
    return list(set(list1) & set(list2))

print(common_elements(list1, list2)) # [4, 5]
```

3. Write a Python program to count the occurrences of each word in a given sentence.

```
from collections import Counter

def word_count(sentence):
    words = sentence.split()
    return dict(Counter(words))

print(word_count("this is a test sentence with multiple words"))
```

4. Write a Python program to remove duplicates from a list.

```
def remove_duplicates(lst):
    return list(set(lst))

print(remove_duplicates([1, 2, 3, 4, 4, 5, 5])) # [1, 2, 3, 4, 5]
```