1. What OS would you recommend for Server usage and why?

I recommend using Linux for many reasons. It's very secure, stable, scalable and easy to use. The fact of it´s opensource helps to the community and the users to change this OS. The most used tecnologies nowadays like the containers (ex:docker), orchestration (ex:kubernetes), CI/CD (ex:jenkins), automation(ex:ansible, terraform) are well prepared and were born in linux. Of course, the customer's environment is important, and each situation and deployment is different.

2. You've noticed that a server is low on free space, and proceed to clean up the disk. But after the procedure, you are still getting errors that the file system is full. The df command reports that there is 40% free space. Explain what is going on?

It's possible the deleted files did not release the space locked to them.

3. You've set up a new web server, but i t seems to be unreachable. How do you check that the web server i s properly accepting external traffic on port 80?

Check if the service is running:

```
ps all
```

Check if the http port is listening:

```
ss -l |grep http
```

4. You reboot a server and the docker service is online but not accessible, how would you debug the access of a docker app?

On the host run the command: curl localhost:9000 (or the port published)

Try to run the same command from another host: curl 10.0.0.6:9000 (ip and port from host)

Check if the firewall is blocking the port.

5. What monitoring tools do you know? Talk briefly about how it was used, the good and bad parts of it, how it could be improved.

I used NAGIOS monitoring tools to check the health status of servers. Basically checking the CPU, Disk, Network usage and connectivity. This tool is easy to use and has many resources. Since I´m not, at this moment using it, the improvements should be linked to each environment and the needs.

6. You are being DDoSed; how do you mitigate or stop the attack? What kind of further precautions would you take so that a DDoS will not happen again?

Managing a firewall solution with rules added to deny all incoming traffic from the attackers, based on protocols, ports or the originating IP addresses. A DDS, IPS could help as well. But since this type of attack can be of multiple types, many precautions would be recommended. A good application front end hardware is a really robust good choice to help in the future.

7. You've noticed that a server has little free disk space left, and you proceed to remove some unneeded files. After the procedure, however, the df tool still reports the same small amount of free disk space left. How would you diagnose the issue, and what tools would you use for that?

Check with:

```
df -i
```

Take a look if you have deleted the files but still opened:

```
lsof -nP | grep '(deleted)'
```

8. CloudHosting Provider e-mails you about a maintenance requiring a reboot on one of our apps, what steps would you do to ensure the server is rebootable?
I´ll schedule a job to reboot the server.

9. You get in the office and a client contacts us that his system is not accessible, you try to login on the server and get a timeout. Monit says the server is online and all services are running. What would you do?
Start to get sure there is no problem on the network level, like a switch, router, hub. After that, check if there is a firewall stopping the traffic in between. If no problem was detected, the logs can be checked.

10. You have a database server running Postgresql, and you need to backup all of the databases nightly using a shell script. Please provide a minimalistic shell or Perl script that would achieve that, and provide for error notifications and rotation of the backups. Please specify how you would set up that script to run every night in a Linux distribution.

Script: postgresql_bkp.sh

```
# Database name
db_name=avidity

# Backup storage directory
backupfolder=~/postgresql_bkp

# Notification email address
recipient_email=dbowner@avidity.com

# Number of days to store the backup
keep_day=30

sqlfile=$backupfolder/all-database-$(date +%d-%m-%Y_%H-%M-%S).sql
zipfile=$backupfolder/all-database-$(date +%d-%m-%Y_%H-%M-%S).zip

#create backup folder
mkdir -p $backupfolder

# Create a backup

if pg_dump $db_name > $sqlfile ; then
  echo 'Sql dump created'
else
  echo 'pg_dump return non-zero code' | mailx -s 'No backup was created!' $recipient_email
  exit
```

```
fi

# Compress backup
if gzip -c $sqlfile > $zipfile; then
  echo 'The backup was successfully compressed'
else
  echo 'Error compressing backup' | mailx -s 'Backup was not created!' $recipient_email
  exit
fi

rm $sqlfile
echo $zipfile | mailx -s 'Backup was successfully created' $recipient_email

# Delete old backups
find $backupfolder -mtime +$keep_day -delete
```

Crontab

Install the postgresql user

```
su - postgres
```

Create a backup storage folder

```
mkdir -p ~/postgresql_bkp
```

Open crontab by running the command

```
crontab -e
```

Add this line at the end

```
0 0 23 * * * /postgresql_bkp.sh
```