Linux

- Yum
    - **sudo yum install tree**
        - tree /etc
    - **sudo yum mlocate**
        - **sudo updatedb**
        - locate <name>

- Get info about file
    - ls, file, stat
        - **file /etc/passwd**
        - **stat /etc/passwd**
- CD
    - **cd /**
    - **cd ~**
    - **cd -**
- MKDIR
    - **mkdir -p /parent/child**
    - **mkdir {dir1,dir2,dir3}**
    - mkdir dir{1,2,3}
    - touch file{1,2,3}.txt
        - no space between dir1
- CP
    - cp -i <source path/file> <destination path>
        - -i
            - provide warning message
        - -r
            - recursive
        - -u
            - only copies if newer
        - -a
            - preserves all metadata
- MV
    - mv <source file path> <destination path/file>
- RM & RMDIR
    - touch file{a,b,c,d}.txt
    - rm file[abcd].txt

- ln
    - ln file.txt fileLink.txt
    - ln -s file.txt fileLink.txt
- 1>, 2>
- tee - *redirect STDOUT to a file and to the screen*
- sed?
- find?
- ps
    - ps -e —format uid,pid,tty,%cpu,cmd —sort %cpu
    - ps -U ec2-user —format %mem | aws '{memory +=$1};' END'{print memory}'

cat >> tmp.txt

streaming input
>> append
> overwrite

```
# compressed archive file of a dir
tar -csvpzf backup.FILE.tar.gz FILE
#alias
alias = backup='tar -cvzf'
backup backup.FILE.tar.gz FILE

# ??appended "message"
echo "message" | tee FILE

# overwritten with new "message"
echo "message" > FILE

#display only onlyusername
awk -F: '{ print $1}' /etc/passwd
#display group
```

```
cat /etc/group

#add user
sudo useradd -d /usr/USER -c "COMMENT" -g ec2-user -p secret -s /usr/bin/bash USER
#find user
cat /etc/passwd | grep USER




#add -a OR -d(remove) user togroup
gpasswd -a USER GROUP
#add multi-user
gpasswd -M USER1, USER2 GROUP


#change ownership
sudo chown -R USER:GROUP DIR
```

---

scp -r -i webInstance.pem ec2-user@54.226.235.151:/home/ec2-user/lab /Users/zhanxink/Downloads

---

Module 11

- details for network interfaces
  - cat /etc/sysconfig/network-scripts/ifcfg-eth0
-
  - ip `-o link show | grep -v lo`
    - □ for interfaces
  - `ip route | grep default`
    - □ for gateway
  - `ethtool -i eth0 | grep driver`
    - □ for driver

Module 12 - 13

Module 14

- Monitoring file system space utilization commands
  - **Volume level**
  - **File system level**
    - **Mounting**
  - **directory level**

Module 15

1. All the files that end with .cfg under the /etc directory
   a. ls /etc/*.cfg
   b. find <filename> -name “*.cfg”
   c. find . -name “addresses.dat” -exec grep “.net” ‘{}’ \
2. All hidden files in the /home/ec2-user directory
   a. ls -a /home/ec2-user/.*
3. The directory names that exist under the /var directory
   a. ls -d /var/*
4. All the files that start with the letter “a” under the /bin directory
   a. ls /bin/a*
5. All the files that have exactly three letters in their filename in the /bin directory
   a. ls /bin/???
6. All files that have exactly three letters in their filename and end with either the letter “t” or the letter “h” in the /bin directory
   a. ls /bin/??[th]

   b. [^ht]: not include h and t
   c. [a-z0-9]: a to z, 0-9
7. grep addresses.dat -E -e ‘[a-z]+@+[a-z]+.(net|org)’
8. a.

- Section 13
  - Netowkring Basics
    - IP - internet protocol, routes packets from one machine to another
      - 127.x.x.x, denotes a fictitious network that has no real hardware interface and only one host.
    - ICMP - internet control message protocol, defines several kinds of low-level support(error message, routing assistance, debugging help) for IP
    - ARP - address resolution protocol. translate IP to hardware address
    - UDP - user datagram protocol, implement unverified, one way data delivery

- TCP - transmission control protocol, implements reliable, full duplex, flow-controlled, error-corrected conversations
- IPv4 - 4 byte IP address
- Ipv6 - 16 bytes
- Packet
  - data travel on a network in form of packets
    - header - indicate where the packet came from and where it go, also include checksum, protocol-specific info, or other handling instructions
    - payload - data
- Address types
  - Unicast – addresses that refer to a single network interface
  - Multicast – addresses that simultaneously target a group of hosts
  - Broadcast – addresses that include all hosts on the local subnet
  - Anycast – addresses that resolve to any one of a group of hosts

---

Hello there,

Thank you for contacting AWS Premium Support today. My name is Jason and it's a pleasure working with you. I understand that you are experiencing issues with Linux instances performing slow and unresponsive. There are different factors that cause this performance issue, like CPU, memory, and disk. For the first step we can check our CPU utilization.

To check the utilization of resources:
You can go to your EC2 instance console and check your utilization of resources from the “Monitoring” tab. (EC2 → Your Linux Instance → Monitoring tab). You can check the “CPU Utilization” metric, which might cause your instances become slow and unresponsive. If you CPU usgae is always too high then you need to double check if the current instance type from the “Detail tab“. If the instance type is the one you expect it then you may consider modify your instance type to have better performance. For further investigate the process utilization of your Linux instance, you can use top command in Linux instance checking if there any anomalies process is running.

Also, you can check the utilization of Linux instance directly from CloudWatch which collect and processes data from EC2 into near real-time metrics. To prevent the CPU utilization hit 100% and affects your business, you can create a CloudWatch alarm to notify you when the CPU metric reaches a certain percentage. Before your instance becomes slow and unresponsive, you have enough time to make changes.

After above operations, you should have good overview of performance of your Linux instance and the potential issues cause your instance become slow and unresponsive.

Please do reach out to me if you have any questions. If possible, we can come on a call to discuss further.

Your feedback is important to us, please share your experience by rating this response. You will find a link to the AWS Support Center at the end of this correspondence to rate us.

Yeah. So what You could pro probably do is, um, in the very beginning, you can put in all the major factors that cause this performance issues. So basically CPU memory, uh, dis dis usage. Um, and they provide them a way to check that from the EC2 instance. So that will be the easy two instance metrics and the Met and the monitoring tools inside the OS because

Reference links:
[1] https://docs.aws.amazon.com/cli/latest/reference/cloudwatch/get-metric-statistics.html
[2] https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-types.html
[3] https://www.geeksforgeeks.org/top-command-in-linux-with-examples/
[4] https://www.geeksforgeeks.org/kill-command-in-linux-with-examples/?ref=lbp

---

Hello there,

Thank you for contacting AWS Premium Support today. My name is Jason, and it's a pleasure to work with you. I understand that you're experiencing performance issues with your Linux instances, which are slow and unresponsive. Several factors, such as CPU, memory, and disk, can contribute to these issues. Let's start by checking our CPU utilization.

Steps to Check Resource Utilization:
option 1: Via EC2 Instance Console:
Go to the EC2 console.
Navigate to your Linux instance and select the “Monitoring” tab (EC2 → Your Linux Instance → Monitoring tab).
Check the “CPU Utilization” metric, which might be the cause of the slowness and unresponsiveness.
If the CPU usage is consistently high, verify the current instance type from the “Detail” tab. If the instance type is as expected but performance issues persist, consider upgrading to a more powerful instance type[1]. For deeper analysis, use the `top` command on your Linux instance to check for any anomalous processes consuming excessive resources[2].

Option 2: Using CloudWatch:
CloudWatch collects and processes data from EC2 instances into near real-time metrics and you can check the utilization metrics for your Linux instance directly from CloudWatch[3]. To prevent CPU utilization from reaching 100% and affecting your business, create a CloudWatch alarm to notify you when the CPU metric hits a certain threshold[4]. This will give you enough time to make necessary adjustments before the instance becomes slow and unresponsive.

By following these steps, you should get a good overview of your Linux instance's performance and identify potential issues causing the slowdown.

Please feel free to reach out if you have any questions. If needed, we can arrange a call to discuss further.

Your feedback is important to us. Please share your experience by rating this response. You will find a link to the AWS Support Center at the end of this correspondence to rate us.

Best regards,
Jason Zhang
AWS Support Engineer

Reference links:
[1] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-resize.html
[2]https://www.geeksforgeeks.org/top-command-in-linux-with-examples/
[3] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch.html
[4] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-createalarm.html