**DevOps/Deployment – Key Topics**

**Operating Systems (OS)**

* **Definition**: Software that manages hardware resources and provides services for running applications.
* **Types**:
  + **Windows**, **Linux**, **macOS**
* **Sample Command (Linux)**:

bash

Copy code

uname -a # Shows OS details

**Linux**

* **Definition**: Open-source Unix-like operating system used for servers and development.
* **Sample Command**:

bash

Copy code

ls -la # List files with permissions

**Windows**

* **Definition**: Microsoft's proprietary OS, commonly used on PCs and servers.
* **Sample Command (PowerShell)**:

powershell

Copy code

Get-EventLog -LogName System # View system logs

**Troubleshooting**

* **Definition**: Identifying and resolving issues affecting system performance or availability.
* **Sample Commands**:
  + **Linux**:

bash

Copy code

top # View active processes and resource usage

* + **Windows**:

powershell

Copy code

Get-Process # View active processes

**Networking**

* **Definition**: Connecting multiple devices to communicate and share resources.
* **Sample Command**:

bash

Copy code

ip a # Display network interfaces in Linux

**Containerization**

* **Definition**: Encapsulation of an application and its dependencies into containers for portability.
* **Sample Command**:

bash

Copy code

docker run nginx # Run a Docker container

**Docker**

* **Definition**: A platform that uses containerization to package and run applications.
* **Sample Command**:

bash

Copy code

docker ps # List running containers

**Kubernetes**

* **Definition**: An orchestration system for managing containerized applications across a cluster.
* **Sample Command**:

bash

Copy code

kubectl get pods # List all running pods in Kubernetes

**Continuous Integration/Continuous Deployment (CI/CD)**

* **Definition**: Practice of automating code integration, testing, and deployment.
* **Sample Tool**:
  + **Jenkins** for automating CI/CD pipelines.

**Infrastructure as Code (IaC)**

* **Definition**: Managing and provisioning computing infrastructure through code.
* **Sample Tool**:
  + **Terraform**:

bash

Copy code

terraform apply # Deploy infrastructure defined in code

**DevOps/Deployment – Example Questions**

**Operating Systems (OS)**

1. **Types of Operating Systems**: Linux, Windows, macOS, Unix.
2. **Troubleshooting**: Use top, dmesg in Linux or Event Viewer in Windows.

**Containerization**

1. **Types of Containers**: Docker, LXC.
2. **What is Docker**: A platform to build, run, and manage containers.
3. **What is Kubernetes**: Orchestration tool for deploying and scaling containerized applications.
4. **How to Resolve Container Crashes**: Check logs using docker logs <container-id>, inspect resource usage.
5. **Microservices**: Small, independently deployable services communicating via REST or messaging protocols.

**CI/CD**

1. **What is CI/CD**: Automating code testing, building, and deployment.
2. **Deployment Policies**: Blue/Green, Rolling, Canary deployments.
3. **Infrastructure as Code**: Frameworks like Terraform, AWS CloudFormation to manage infrastructure through code.

**Networking – Key Topics**

**IP Addresses**

* **Public vs. Private Networks**: Public IP is globally routable, private IP is for internal networks.

**Protocols**

* **TCP**: Reliable, connection-oriented protocol.
* **UDP**: Unreliable, connectionless protocol.
* **Sample Commands**:
  + **Linux**:

bash

Copy code

nc -zv <ip> <port> # Check if port is open (TCP)

**Domain Name System (DNS)**

* **DNS Record Types**: A, AAAA, CNAME, MX.
* **DNS Resolution Flow**: Resolves domain names into IP addresses via a recursive process.
* **Sample Command**:

bash

Copy code

nslookup example.com # Query DNS

**DHCP**

* **Definition**: Assigns IP addresses automatically to devices on a network.

**Networking – Practice Questions**

**IP Addresses**

1. **Public vs. Private Networks**: Public IPs are globally accessible, private IPs are used within local networks.
2. **How Devices Communicate**: Through IP addressing and protocols like TCP/UDP.

**Protocols**

1. **TCP and UDP in IPv4/IPv6**: TCP is reliable, UDP is faster but less reliable.

**DNS**

1. **DNS Record Types**: A (IPv4 address), AAAA (IPv6), MX (Mail Exchange), CNAME (Alias).
2. **DNS Resolution Flow**: A recursive process where the DNS server resolves a domain to an IP.
3. **Troubleshoot DNS Issues**: Use nslookup, dig to query DNS servers.

**DHCP**

1. **IP Address Allocation**: DHCP automatically assigns IP addresses to devices in a network.

**Operating Systems – Key Topics**

**Linux:**

* **Networking Essentials**: Commands like ip, ifconfig.
* **Port Numbers**: Common ports like 80 (HTTP), 22 (SSH).
* **Package Management**: apt, yum.

**Windows:**

* **PowerShell**:

powershell

Copy code

Get-Service # List running services

* **System Logs**: Use **Event Viewer** to analyze logs.

**Operating Systems – Example Questions**

**Windows:**

1. **PowerShell Execution**: Run scripts using ./script.ps1.
2. **System Logs**: Access via Event Viewer.
3. **Windows Tools for Troubleshooting**: Use tools like **Task Manager** and **Resource Monitor**.
4. **Memory Management**: Use Get-Process to find resource-heavy processes.

Here’s a more detailed breakdown of definitions and how to troubleshoot networking issues related to topics like \*\*TCP\*\*, \*\*UDP\*\*, \*\*DNS\*\*, and \*\*DHCP\*\*, as well as methods to troubleshoot networking issues on both \*\*Linux\*\* and \*\*Windows\*\*:

---

### \*\*Networking – Key Subtopics\*\*

#### \*\*1. IP Addresses\*\*

- \*\*Public vs. Private Networks\*\*:

- \*\*Public IP Addresses\*\*: Globally routable and used to connect devices over the internet.

- \*\*Private IP Addresses\*\*: Used within internal networks (e.g., 192.168.x.x, 10.x.x.x) and cannot be accessed from outside the local network without a router or NAT device.

#### \*\*2. TCP (Transmission Control Protocol)\*\*

- \*\*Definition\*\*: TCP is a \*\*connection-oriented\*\* protocol that ensures reliable communication between two devices. It breaks down data into packets, ensures they are transmitted in order, and confirms delivery.

- \*\*Use Cases\*\*: Web browsing (HTTP/HTTPS), email (SMTP), file transfers (FTP).

- \*\*How to Troubleshoot\*\*:

- Use the \*\*`telnet`\*\* or \*\*`nc`\*\* (Netcat) command to check if a TCP port is open and listening:

```bash

telnet <server-ip> 80

```

```bash

nc -zv <server-ip> 80

```

#### \*\*3. UDP (User Datagram Protocol)\*\*

- \*\*Definition\*\*: UDP is a \*\*connectionless\*\* protocol that sends data without verifying delivery or packet order. It's faster than TCP but less reliable.

- \*\*Use Cases\*\*: Video streaming, online gaming, DNS queries.

- \*\*How to Troubleshoot\*\*:

- Use \*\*`nc`\*\* to test UDP connectivity:

```bash

nc -uzv <server-ip> 53 # Check UDP port (53 for DNS)

```

#### \*\*4. Domain Name System (DNS)\*\*

- \*\*Definition\*\*: DNS is the system that translates domain names (e.g., `example.com`) into IP addresses.

- \*\*DNS Record Types\*\*:

- \*\*A\*\*: Maps a domain to an IPv4 address.

- \*\*AAAA\*\*: Maps a domain to an IPv6 address.

- \*\*CNAME\*\*: Alias of one domain name to another.

- \*\*MX\*\*: Mail Exchange records, directing email to servers.

- \*\*DNS Resolution Flow\*\*: The process of querying a series of DNS servers (recursive DNS resolver, root server, TLD server, authoritative DNS server) to resolve a domain into an IP address.

- \*\*How to Troubleshoot\*\*:

- Use \*\*`nslookup`\*\* or \*\*`dig`\*\* to troubleshoot DNS issues:

```bash

nslookup example.com

dig example.com

```

#### \*\*5. DHCP (Dynamic Host Configuration Protocol)\*\*

- \*\*Definition\*\*: DHCP is used to automatically assign IP addresses to devices on a network. It eliminates the need for manual IP address configuration.

- \*\*How to Troubleshoot\*\*:

- \*\*Linux\*\*:

- Check if the system is receiving a DHCP-assigned IP address:

```bash

ip a

```

- Restart the DHCP service:

```bash

sudo dhclient -r # Release current IP

sudo dhclient # Request new IP

```

- \*\*Windows\*\*:

- Use the following commands to release and renew the IP address:

```powershell

ipconfig /release

ipconfig /renew

```

---

### \*\*How to Troubleshoot Networking Issues on Linux\*\*

1. \*\*Check Network Interfaces\*\*:

- Use \*\*`ip a`\*\* or \*\*`ifconfig`\*\* to display network interfaces and their status.

```bash

ip a

ifconfig

```

2. \*\*Check Network Routes\*\*:

- Use \*\*`ip route`\*\* or \*\*`route -n`\*\* to view the routing table and ensure traffic is being directed correctly.

```bash

ip route

route -n

```

3. \*\*Ping to Check Connectivity\*\*:

- Ping the gateway or another machine to ensure basic connectivity.

```bash

ping <ip-address>

```

4. \*\*Test DNS Resolution\*\*:

- Use \*\*`nslookup`\*\* or \*\*`dig`\*\* to verify DNS is resolving correctly.

```bash

nslookup example.com

dig example.com

```

5. \*\*Check Open Ports and Services\*\*:

- Use \*\*`netstat -tuln`\*\* or \*\*`ss -tuln`\*\* to check listening services and open ports.

```bash

netstat -tuln

ss -tuln

```

6. \*\*Traceroute to Identify Routing Issues\*\*:

- Use \*\*`traceroute`\*\* to track the packet's route to a destination and identify network bottlenecks.

```bash

traceroute example.com

```

7. \*\*TCP and UDP Port Testing\*\*:

- Use \*\*`nc`\*\* (Netcat) to check TCP or UDP connectivity.

```bash

nc -zv <ip-address> <port>

nc -uzv <ip-address> <port>

```

---

### \*\*How to Troubleshoot Networking Issues on Windows\*\*

1. \*\*Check Network Interfaces\*\*:

- Use \*\*`ipconfig`\*\* to display the current network configuration and assigned IP address.

```powershell

ipconfig

```

2. \*\*Check DNS Resolution\*\*:

- Use \*\*`nslookup`\*\* to verify DNS queries.

```powershell

nslookup example.com

```

3. \*\*Ping for Connectivity\*\*:

- Use \*\*`ping`\*\* to test connectivity to another machine or server.

```powershell

ping <ip-address>

```

4. \*\*Traceroute to Identify Network Hops\*\*:

- Use \*\*`tracert`\*\* to trace the route a packet takes to reach its destination.

```powershell

tracert example.com

```

5. \*\*Test Open Ports\*\*:

- Use \*\*`telnet`\*\* to test if a service is listening on a specific port.

```powershell

telnet <ip-address> <port>

```

6. \*\*Check Listening Services\*\*:

- Use \*\*`netstat -an`\*\* to list all open and listening ports.

```powershell

netstat -an

```

7. \*\*Release and Renew IP Address (DHCP)\*\*:

- Use \*\*`ipconfig /release`\*\* and \*\*`ipconfig /renew`\*\* to reset your network connection.

```powershell

ipconfig /release

ipconfig /renew

```

---

### \*\*Networking Practice Questions\*\*

#### \*\*1. IP Addresses\*\*

- \*\*What is the difference between public and private networks?\*\*

- \*\*Public IPs\*\* are routable over the internet; \*\*private IPs\*\* are confined to internal networks.

- \*\*How do devices communicate?\*\*

- Devices communicate over networks using protocols like \*\*TCP/IP\*\*, relying on unique IP addresses to send and receive data.

#### \*\*2. TCP and UDP\*\*

- \*\*How do you use protocols like TCP and UDP in both IPv4 and IPv6?\*\*

- TCP is used when reliability is needed (e.g., web pages, emails), while UDP is used when speed is critical (e.g., video streaming).

#### \*\*3. DNS\*\*

- \*\*What types of DNS records are there?\*\*

- \*\*A\*\*, \*\*AAAA\*\*, \*\*CNAME\*\*, \*\*MX\*\*, \*\*TXT\*\*, \*\*NS\*\*.

- \*\*What is the DNS resolution flow?\*\*

- DNS queries start with a recursive resolver, move through the root server, TLD server, and eventually to the authoritative DNS server.

- \*\*How do you troubleshoot DNS resolution issues?\*\*

- Use `nslookup` or `dig` to check if DNS queries are resolving correctly.

#### \*\*4. DHCP\*\*

- \*\*How do devices get their IP addresses via DHCP?\*\*

- A device sends a DHCP request to a DHCP server, which assigns it an IP address, subnet mask, gateway, and DNS information.

---

### \*\*Operating Systems Practice Questions\*\*

#### \*\*Windows\*\*

- \*\*How do you execute PowerShell scripts?\*\*

- You can run a PowerShell script by typing `./script.ps1` in the PowerShell console.

- \*\*How do you check system logs?\*\*

- Use \*\*Event Viewer\*\* to browse logs under \*\*Windows Logs > System\*\*.

- \*\*How do you troubleshoot memory issues?\*\*

- Use \*\*Task Manager\*\* to monitor resource usage, and free up memory by stopping processes.

---

This breakdown should help you better understand the \*\*networking concepts\*\* and \*\*commands\*\* and prepare for troubleshooting questions during your interview for both \*\*Linux\*\* and \*\*Windows\*\* environments.