Contents

[**🔍 Networking Concepts** 3](#_Toc195263720)

[1. What is DNS? Why is it used? 3](#_Toc195263721)

[2. What is NAT? Why is it used? 3](#_Toc195263722)

[3. Difference between TCP and UDP? 3](#_Toc195263723)

[4. What is the OSI Model? 3](#_Toc195263724)

[5. What is a Firewall? 4](#_Toc195263725)

[6. What are ports? Common ports? 4](#_Toc195263726)

[7. What is IP Subnetting? 4](#_Toc195263727)

[8. What is CIDR notation? 4](#_Toc195263728)

[9. What is Routing? 4](#_Toc195263729)

[10. What is VPN? Why is it used? 4](#_Toc195263730)

[11. What is ARP? 5](#_Toc195263731)

[12. What is a VLAN? 5](#_Toc195263732)

[13. What are protocols? Key examples? 5](#_Toc195263733)

[14. IPv4 vs. IPv6: 5](#_Toc195263734)

[**🧰 Command-line Networking Tools** 5](#_Toc195263735)

[15. ping – Purpose and example usage (ping google.com) 5](#_Toc195263736)

[16. traceroute / tracert – Route tracing and delay checking (traceroute google.com) 5](#_Toc195263737)

[17. netstat – Show active connections and routing tables (netstat -a) 5](#_Toc195263738)

[18. nmap – Scan ports and detect open services (nmap -p 1-1000 target) 5](#_Toc195263739)

[19. tcpdump – Capture packets on an interface (tcpdump -i eth0) 5](#_Toc195263740)

[20. ipconfig / ifconfig – View IP configuration (ifconfig, ipconfig) 5](#_Toc195263741)

[21. dig – DNS queries (dig google.com) 5](#_Toc195263742)

[22. nslookup / host – Get DNS info (host google.com) 5](#_Toc195263743)

[23. wireshark – GUI tool for capturing and analyzing packets 5](#_Toc195263744)

[24. iperf – Measure bandwidth between two systems 5](#_Toc195263745)

[**☁️ Cloud Networking (AWS/Azure/Kubernetes)** 6](#_Toc195263746)

[25. What is a VPC (Virtual Private Cloud)? 6](#_Toc195263747)

[26. What is a Subnet in cloud platforms? 6](#_Toc195263748)

[27. What are Security Groups vs. Network ACLs? 6](#_Toc195263749)

[28. What is an Internet Gateway and NAT Gateway? 7](#_Toc195263750)

[29. What is Service Discovery in Kubernetes? 7](#_Toc195263751)

[30. What is a Kubernetes Service and what types exist (ClusterIP, NodePort, LoadBalancer)? 7](#_Toc195263752)

[31. How do pods communicate in Kubernetes? 8](#_Toc195263753)

[32. What is Ingress and how is it used in Kubernetes? 8](#_Toc195263754)

[33. How does Kubernetes networking differ from traditional networking? 8](#_Toc195263755)

[**🔐 Security and Best Practices** 9](#_Toc195263756)

[34. What is HTTPS and How Does TLS/SSL Work? 9](#_Toc195263757)

[35. What is Port Forwarding and How is it Configured? 9](#_Toc195263758)

[36. Common Network Vulnerabilities & Mitigations 10](#_Toc195263759)

[37. What is Zero Trust Networking? 10](#_Toc195263760)

[**🧪 Advanced & Troubleshooting Scenarios** 11](#_Toc195263761)

[38. How to Debug a DNS Resolution Failure 11](#_Toc195263762)

[39. How to Check if a Port is Open or Blocked 12](#_Toc195263763)

[40. How to Monitor Real-Time Network Traffic on a Server 13](#_Toc195263764)

[41. How to Troubleshoot Slow Application Performance Due to Networking 13](#_Toc195263765)

[42. How to Debug Pod-to-Pod Network Issues in Kubernetes 14](#_Toc195263766)

**🔍** Networking Concepts

### What is DNS? Why is it used?

DNS (Domain Name System) translates human-readable domain names (e.g., google.com) into machine-readable IP addresses (e.g., 172.217.0.142).

Used to simplify accessing resources on the internet without memorizing IP addresses.

### What is NAT? Why is it used?

NAT (Network Address Translation) is a method used in networking to allow multiple devices on a private network to access the internet using a single public IP address.

Simple Breakdown:

Why it's used: Because public IPv4 addresses are limited.

How it works: NAT changes the private IP address of a device to a public IP address when sending data to the internet, and then reverses it when receiving data.

### Difference between TCP and UDP?

|  |
| --- |
|  |

|  |
| --- |
| **TCP (Transmission Control Protocol)** |

Connection-oriented, reliable (ensures delivery), ordered (e.g., HTTP, FTP), web browsing

| **UDP (User Datagram Protocol)** |
| --- |

|  |
| --- |
|  |

Connectionless, unreliable (no delivery guarantees), faster (e.g., VoIP, DNS), video call

### What is the OSI Model?

A 7-layer conceptual framework standardizing network communication functions (from physical transmission to application-level interactions).

**7 Layers of the OSI Model:**

Physical (cables, signals)

Data Link (MAC, switches)

Network (IP, routers)

Transport (TCP/UDP)

Session (connections)

Presentation (data formatting)

Application (HTTP, FTP)

**TCP/IP Model:**

A simplified 4-layer model:

Network Interface (Physical + Data Link)

Internet (IP, routing)

Transport (TCP/UDP)

Application (HTTP, SSH, etc.).

### What is a Firewall?

A security device/software that filters incoming/outgoing traffic based on rules to block unauthorized access.

### What are ports? Common ports?

Virtual endpoints for communication (0–65535).

Common ports: 80 (HTTP), 443 (HTTPS), 22 (SSH), 21 (FTP), 53 (DNS).

### What is IP Subnetting?

Dividing a network into smaller subnets to improve efficiency, security, and reduce congestion.

### What is CIDR notation?

CIDR (Classless Inter-Domain Routing): Notation like 192.168.1.0/24 where /24 indicates the subnet mask (255.255.255.0).

Used to allocate IP addresses flexibly and reduce wastage.

### What is Routing?

The process of forwarding packets between networks using routers, which use routing tables to determine paths.

### What is VPN? Why is it used?

VPN (Virtual Private Network): Encrypts internet traffic and masks the user’s IP.

Used for privacy, bypassing geo-restrictions, and secure remote access.

Public vs. Private IP:

Public IP: Globally routable (assigned by ISPs).

Private IP: Used internally (e.g., 192.168.x.x, 10.x.x.x), not routable on the internet.

### What is ARP?

ARP (Address Resolution Protocol): Maps IP addresses to MAC addresses on a LAN.

Function: Hosts broadcast ARP requests to find MACs for local communication.

### What is a VLAN?

VLAN (Virtual LAN): Logically segments a physical network into isolated broadcast domains.

Useful for security, traffic management, and reducing collisions.

### What are protocols? Key examples?

Rules governing data communication.

Examples: HTTP (web), FTP (file transfer), SSH (secure remote login), DHCP (IP assignment), ICMP (ping).

### IPv4 vs. IPv6:

IPv4: 32-bit address (4.3 billion addresses), uses NAT.

IPv6: 128-bit address (340 undecillion addresses), built-in security (IPsec), no NAT needed.

**🧰** Command-line Networking Tools

### ping – Purpose and example usage (ping google.com)

### traceroute / tracert – Route tracing and delay checking (traceroute google.com)

### netstat – Show active connections and routing tables (netstat -a)

### nmap – Scan ports and detect open services (nmap -p 1-1000 target)

### tcpdump – Capture packets on an interface (tcpdump -i eth0)

### ipconfig / ifconfig – View IP configuration (ifconfig, ipconfig)

### dig – DNS queries (dig google.com)

### nslookup / host – Get DNS info (host google.com)

### wireshark – GUI tool for capturing and analyzing packets

### iperf – Measure bandwidth between two systems

**☁️** Cloud Networking (AWS/Azure/Kubernetes)

### What is a VPC (Virtual Private Cloud)?

A Virtual Private Cloud (VPC) is an isolated, logically segmented private cloud environment within a public cloud (AWS, GCP, Azure). It allows users to define their own virtual network, including IP ranges, subnets, routing, and security policies.

Key Features:

Provides network isolation for security.

Allows custom IP addressing (CIDR blocks).

Supports subnets, route tables, and gateways.

Enables hybrid cloud connectivity via VPN or Direct Connect.

### What is a Subnet in cloud platforms?

A subnet is a segmented range of IP addresses within a VPC, deployed in a specific Availability Zone (AZ) for fault tolerance.

Types:

Public Subnet: Has a route to the internet (via an Internet Gateway).

Private Su**bnet:** No direct internet access (uses NAT Gateway for outbound traffic).

### What are Security Groups vs. Network ACLs?

| **Feature** | **Security Group** | **Network ACL** |
| --- | --- | --- |
| **Scope** | Instance-level (stateful) | Subnet-level (stateless) |
| **Rules** | Allow-only | Allow/Deny |
| **Evaluation** | All rules evaluated before allowing traffic | Rules evaluated in order (top-down) |
| **Example Use** | Allow SSH (port 22) to an EC2 instance | Block an IP range from accessing a subnet |

### What is an Internet Gateway and NAT Gateway?

| Feature | Internet Gateway (IGW) | NAT Gateway |
| --- | --- | --- |
| Purpose | Enables public subnets to access the internet (bidirectional). | Allows private subnets to access the internet (outbound-only). |
| Public IP | Required for instances. | Uses a public IP but hides private instances. |
| Use Case | Public-facing web servers. | Private databases needing updates from the internet. |

### What is Service Discovery in Kubernetes?

Service Discovery allows Kubernetes services/pods to locate and communicate with each other dynamically, using:

DNS-based (e.g., my-service.namespace.svc.cluster.local).

Environment variables (deprecated).

Tools:

CoreDNS (default in Kubernetes).

External DNS (for cloud providers).

### What is a Kubernetes Service and what types exist (ClusterIP, NodePort, LoadBalancer)?

| **Type** | **Description** | **Example Use** |
| --- | --- | --- |
| **ClusterIP** (default) | Internal IP, accessible only within the cluster. | Database services. |
| **NodePort** | Exposes a port on each node (30000–32767). | Testing/debugging. |
| **LoadBalancer** | Cloud-provisioned external LB (AWS ALB, GCP LB). | Public-facing apps. |
| **ExternalName** | Maps to an external DNS (e.g., AWS RDS). | Legacy integration. |

### How do pods communicate in Kubernetes?

Same Node: Via the local bridge (cni0).

Different Nodes: Via overlay networks (e.g., Flannel, Calico) or cloud-native networking (AWS VPC CNI).

Services: Use kube-proxy (iptables/IPVS) to route traffic.

### What is Ingress and how is it used in Kubernetes?

An Ingress is an API object that manages external HTTP/S access to services (Layer 7 routing). It uses an Ingress Controller (e.g., Nginx, AWS ALB) to enforce rules.

### How does Kubernetes networking differ from traditional networking?

| **Feature** | **Kubernetes Networking** | **Traditional Networking** |
| --- | --- | --- |
| **IP Allocation** | Dynamic (per-pod IPs). | Static/DHCP (per VM/host). |
| **Service Discovery** | Built-in DNS (CoreDNS). | Manual (hosts files, DNS). |
| **Load Balancing** | Native (Service/Ingress). | External (HAProxy, F5). |
| **Security** | Network Policies (Calico). | Firewalls/VLANs. |
| **Scalability** | Self-healing pods. | Manual scaling. |

**🔐** Security and Best Practices

### What is HTTPS and How Does TLS/SSL Work?

HTTPS (HyperText Transfer Protocol Secure)

An encrypted version of HTTP that ensures secure communication between a client (browser) and a server.

Uses TLS/SSL (Transport Layer Security / Secure Sockets Layer) to encrypt data in transit.

How TLS/SSL Works (TLS Handshake)

Client Hello → Browser sends supported cipher suites & a random number.

Server Hello → Server responds with chosen cipher suite, its SSL certificate, and a random number.

Certificate Validation → Browser checks the certificate (issued by a trusted CA).

Key Exchange →

Symmetric Encryption: Client and server agree on a session key (via asymmetric encryption first).

Asymmetric Encryption: Used initially (e.g., RSA, ECC) to securely exchange the symmetric key.

Secure Session → All further communication uses symmetric encryption (faster).

Why HTTPS?

Prevents eavesdropping, tampering, and spoofing (MITM attacks).

Required for SEO, PCI compliance, and modern web security.

### What is Port Forwarding and How is it Configured?

Port Forwarding (NAT Forwarding)

Redirects traffic from a public IP/port to a private IP/port inside a local network.

Used for:

Hosting a web server behind a router.

Remote access (SSH, RDP) to internal devices.

Configuration (Example: Home Router)

Access router admin panel (192.168.1.1).

Navigate to Port Forwarding settings.

Add a rule:

External Port: 80 (HTTP)

Internal IP: 192.168.1.100 (local server)

Internal Port: 80

Protocol: TCP

Security Risks:

Exposes internal services to the internet (ensure firewall rules restrict access).

### Common Network Vulnerabilities & Mitigations

| Vulnerability | Mitigation |
| --- | --- |
| Man-in-the-Middle (MITM) | Use HTTPS (TLS), VPNs, certificate pinning. |
| DDoS Attacks | Rate limiting, WAF (Cloudflare), CDN. |
| IP Spoofing | Ingress/Egress filtering (BCP38), cryptographic auth. |
| DNS Cache Poisoning | DNSSEC, use trusted DNS (Google/Cloudflare). |
| ARP Spoofing | Static ARP entries, DHCP snooping (switch security). |
| Port Scanning | Firewalls, IDS/IPS (Fail2ban), disable unused ports. |
| Weak Authentication | MFA, strong passwords, SSH keys (disable password login). |

Best Practices:

Zero Trust (verify all requests).

Network Segmentation (VLANs, micro-segmentation).

Regular Patching (OS, firmware, IoT devices).

### What is Zero Trust Networking?

Zero Trust Model

A security framework where no entity (user/device) is trusted by default, even inside the network.

Follows "Never Trust, Always Verify" principles.

Core Principles:

Least Privilege Access → Grant minimal required permissions.

Micro-Segmentation → Isolate workloads (e.g., Kubernetes namespaces).

Continuous Authentication → Verify users/devices dynamically (MFA, behavioral analytics).

Encrypt Everything → TLS, VPNs, encrypted DNS.

Key Technologies:

IAM (Identity & Access Management) → Okta, Azure AD.

SDP (Software-Defined Perimeter) → BeyondCorp (Google).

ZTNA (Zero Trust Network Access) → Replace VPNs with app-level access (Cloudflare Access).

Example (Zero Trust Flow):

User requests access → Verified via MFA + device health check.

Access granted only to the specific app (not the whole network).

Traffic is logged & inspected continuously.

Why Zero Trust?

Prevents lateral movement (e.g., ransomware spread).

Supports remote work & hybrid cloud securely.

Summary

HTTPS/TLS → Encrypts web traffic via asymmetric + symmetric encryption.

Port Forwarding → Maps external ports to internal services (use cautiously).

Network Vulnerabilities → MITM, DDoS, spoofing (mitigate with encryption, firewalls, segmentation).

Zero Trust → Verify every request, least privilege, micro-segmentation.

**🧪** Advanced & Troubleshooting Scenarios

### How to Debug a DNS Resolution Failure

Step-by-Step Debugging:

Check Basic Connectivity:

Try to ping an external IP like 8.8.8.8 to verify internet access.

If ping fails, check the network interface (ip a or ifconfig), gateway, and firewall settings.

Test DNS Resolution:

Use nslookup, dig, or host to query a domain (e.g., google.com).

If DNS resolution fails, try querying a public DNS directly (e.g., dig @8.8.8.8 google.com).

Check Local DNS Configuration:

On Linux/macOS, view /etc/resolv.conf to see the configured DNS servers.

On Windows, use ipconfig /all to verify DNS settings.

Ensure DNS servers like 8.8.8.8 or 1.1.1.1 are correctly configured.

Verify DNS Resolver Services:

Check the status of DNS-related services like systemd-resolved or dnsmasq.

Restart the service if it’s inactive or malfunctioning.

Check Firewall Rules:

Ensure that DNS traffic (UDP/TCP on port 53) is allowed by firewall tools like ufw or iptables.

Test with a Different DNS Server:

Temporarily set a public DNS by modifying /etc/resolv.conf and pointing to 8.8.8.8.

### How to Check if a Port is Open or Blocked

Methods:

Using Telnet or Netcat:

Use telnet or nc to test if a specific port (e.g., 80 or 443) is open on a target server.

If the connection is successful → the port is open.

If you get "connection refused" → the port is closed.

If it times out → the port is likely blocked by a firewall.

Using Nmap (Advanced Scanning):

Use nmap to scan specific ports or run a TCP connect scan to detect open/closed ports.

Check Local Listening Ports:

Use tools like netstat or lsof to check if your system is listening on the required port.

Test from a Remote Host:

From a different server, use nc to check if a port (like SSH on port 22) is reachable on the target.

### How to Monitor Real-Time Network Traffic on a Server

Tools & Commands:

tcpdump (Packet Capture):

Captures packets for a specific port or protocol (like HTTP or ICMP).

You can save the capture to a file and analyze it in Wireshark.

iftop (Bandwidth Usage):

Shows real-time bandwidth usage per connection on a network interface.

nload (Interface-Level Traffic):

Displays real-time incoming and outgoing traffic per interface.

netstat or ss (Active Connections):

View all active TCP/UDP connections and the associated processes.

Wireshark (GUI Tool):

A powerful network analyzer for deep packet inspection. Requires a graphical interface.

### How to Troubleshoot Slow Application Performance Due to Networking

Debugging Steps:

Check Latency and Packet Loss:

Use ping or mtr to check network latency and detect packet drops.

Test Bandwidth:

Use iperf3 to test bandwidth between two systems (requires iperf3 installed on both ends).

Inspect DNS Delays:

Use curl with the -I option and time to measure response delays.

Use dig +trace to analyze DNS resolution steps.

Check TCP Retransmissions:

Use ss -ti to view TCP stats like retransmissions and congestion.

Check for Proxy/CDN Issues:

Bypass caches using headers like Cache-Control: no-cache in curl.

Check for Firewall or Throttling Issues:

Use tcptraceroute to test if any firewall or middlebox is affecting traffic.

### How to Debug Pod-to-Pod Network Issues in Kubernetes

Step-by-Step Debugging:

Check Pod Network Status:

Use kubectl get pods -o wide to verify IP addresses and which nodes the pods are running on.

Test Connectivity Between Pods:

Use kubectl exec to enter one pod and ping or curl another pod’s IP or service port.

Check Service DNS Resolution:

From a pod, use nslookup or similar tools to check if service names resolve properly.

Inspect Network Policies:

Use kubectl get networkpolicies to check if any network policies are restricting traffic.

Check CNI Plugin Logs:

Check the status of the CNI plugin (e.g., Calico, Flannel) by inspecting logs in the kube-system namespace.

Node-Level Debugging:

On the node, check iptables NAT rules and interface configurations like cni0.

Use kubectl describe:

Run kubectl describe pod to view detailed pod status and events that might indicate network errors.