**Networking**

Contents

[Networking Basics for Beginners 2](#_Toc205188872)

[Understanding Network Fundamentals 2](#_Toc205188873)

[What is Networking? 2](#_Toc205188874)

[Key Components of Network 2](#_Toc205188875)

[Types of Networks 2](#_Toc205188876)

[Key Concepts in Network 2](#_Toc205188877)

[Networking in a Distributed System 2](#_Toc205188878)

[Tools and Commands 2](#_Toc205188879)

[Linux Network Basics 3](#_Toc205188880)

[Network Interfaces 3](#_Toc205188881)

[Essential Network Commands 3](#_Toc205188882)

[Linux Networking Features 3](#_Toc205188883)

[IP Address Management 3](#_Toc205188884)

[Network in Containers 3](#_Toc205188885)

[Practical Networking Scenarios 3](#_Toc205188886)

# Networking Basics for Beginners

## Understanding Network Fundamentals

### What is Networking?

Refers to practice of connecting computers and other devices to share data/application/hardware. Backbone of distributed systems like Kubernetes.

### Key Components of Network

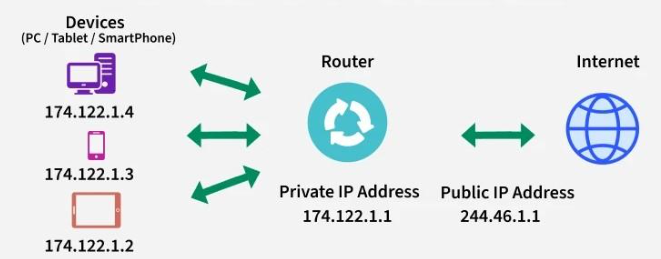
* IP Address 🡪 Unique identifier assigned to devices on a network to enable communication
  + IPv4: Uses 32-bit address (128.0.0.1) supports 4.3 billion unique address may get over in 2035
  + IPv6: Uses 128-bit address (abcd:ff0a:8293:2026:5051:9999) provides unlimited address space.
* Classless Inter-Domain Routing (CIDR) 🡪 Method to define IP Address ranges. CIDR uses notation like 192.1.1.0/24 indicates number of bits for network prefix allowing flexible subnet.
* Ports 🡪 Logical endpoints for specific applications or services running on a device.
* Protocols 🡪 Rules for communication between devices. Include
  + HTTP/HTTPS: Used for web traffic.
  + TCP/UDP: Transport layer protocols for reliable (TCP) and fast, connectionless (UDP) communication.
  + DNS (Domain Name System): Translates human-readable domain names into IP addresses.

### Types of Networks

* LAN 🡪 Network to a small area like an office or home
* WAN (Wide Area Network) 🡪 Network that spans large geographical areas, like internet
* Virtual Network 🡪 Software defined network created within systems like Cloud or Kubernetes Clusters

### Key Concepts in Network

* Subnets 🡪 Small section of large network with unique range of IP Address
* Routing 🡪 Process of determining how data moves from one to another device across network
* Firewall 🡪 Security systems that monitor and control incoming/outgoing traffic.
* Load Balancers 🡪 Distribute traffic across multiple servers to ensure high availability and reliability
* Network Address Translation (NAT) 🡪 Allows multiple devices to use the same public IP address and access the Internet. NAT is a process in which one or more local IP addresses are translated into one or more Global IP addresses and vice versa to provide Internet access to the local hosts.



### Networking in a Distributed System

Distributed systems rely on network to enable communication between services, applications and users. In such systems, concepts like micro services, service discovery and network policies are critical.

### Tools and Commands

* Ping 🡪 Test connectivity between 2 devices.
* Traceroute 🡪 Shows data path to reach destination
* Netstat 🡪 Display network connections, routing tables and more.
* Curl 🡪 Command-line tool for testing HTTP/HTTPs Requests.

## Linux Network Basics

### Network Interfaces

* Physical Interfaces
  + eth0, eth1 – Represent physical network interfaces connected to the system.
  + lo – Loopback interface used for internal communication with IP
* Virtual Interfaces
  + Veth – Virtual Ethernet used in containerized environments
  + Bro – connect multiple interfaces at Layer2 enabling communication between VM or containers

### Essential Network Commands

* Ip addr 🡪 Display or modifies IP Address and properties of network interfaces. Example – ip addr show shows all interfaces and their assigned IPs
* Ip route 🡪 Displays/modifies routing table. Ip route add default via 192.168.1.1 sets default gateway.
* Ping, traceroute, netstat

### Linux Networking Features

* Bridge Network 🡪 Used in Docker and Kubernetes to enable container communication
* Host Network 🡪 Containers or VMs share host network stake.
* Network Namespace 🡪 Isolate network stacks for different process or containers. Each NS has its own interfaces, routes and rules

### IP Address Management

* Dynamic IP Assignment 🡪 Dynamic Host Configuration Protocol (DHCP) automatically assigns IPs to devices.
* Static IP Assignment 🡪 Fixed IPs in /etc/network/interfaces or equivalent configuration files.

### Network in Containers

* Container Network Models
  + Bridge Mode 🡪 Share bridge network with NAT
  + Host Mode 🡪 Use host’s network directly
  + None Mode 🡪 No networking; useful for isolating containers
* Docker Network
  + Default bridge network connects containers with private IPs
  + Custom network allows more control over communication and IP allocation

### Practical Networking Scenarios

* Troubleshooting connectivity
  + Verify connections 🡪 ping, traceroute or curl
  + Interface status 🡪 ip addr, ethtool
* Packet Capture
  + Inspect traffic for debugging 🡪 tcpdump or wireshark
* Firewall Rules
  + Allow/Restrict traffic 🡪 iptables or firewalld

# Kubernetes Network Basics

## Network in Kubernetes

Enables communication between containers, pods, and external resources.

## Container Network Modes

Bridge Network 🡪 Containers connect through a bridge network and communicate using NAT.