Popular Protocols Used in IT and Networking:

HTTP - 80, HTTPS - 443, FTP - 21, SMTP - 25, SSH - 22, Telnet - 23, DNS - 53, Jenkins - 8080, Apache Tomcat - 8080, GitLab - 8081, Bitbucket - 7990, Datadog Agent - 8125, Splunk - 8000, Prometheus - 9090, Grafana - 3000, Docker Daemon - 2375, Kubernetes API Server - 6443, Containerd - 10010, RabbitMQ - 5672

1. Web Protocols & Communication

- **HTTP – 80**
- **HTTPS - 443**
- **FTP - 21**
- **SFTP - 22**
- **SSH - 22**
- **Telnet - 23**
- **DNS - 53**
- **SMTP - 25, 587, 465**
- **POP3 - 110, 995**
- **IMAP - 143, 993**
- **SNMP - 161 (UDP)**
- **RDP (Remote Desktop Protocol) - 3389**

2. DevOps & CI/CD Tools

- **Jenkins - 8080**
- **GitLab CI/CD - 8081**
- **Bitbucket - 7990**
- **SonarQube - 9000**
- **Nexus Repository Manager - 8081**
- **Artifactory - 8082**
- **TeamCity - 8111**
- **Travis CI - Uses dynamic ports**

3. Monitoring & log

- **Datadog agent – 8125(udp) , 8126 (TCP0)**
- **Datadog Logs – 10514**
- **Prometheus – 9090**
- **Grafana – 3000**
- **Thannos - 10901**
- **Splunk - 8000**

4. Containerization & Orchestration

- **Docker Daemon - 2375 (Non-SSL), 2376 (SSL)**
- **Kubernetes API Server - 6443**
- **Kubelet API - 10250**
- **Kube Scheduler - 10251**
- **Kube Controller Manager - 10252**
- **Kubernetes etcd - 2379, 2380**
- **Containerd - 10010**
- **CRIO - 10010**

6. Databases & Storage

- **MySQL - 3306**
- **PostgreSQL - 5432**
- **MongoDB - 27017**
- **Cassandra - 9042**
- **Elasticsearch - 9200**
- **Zookeeper - 2181**
- **Consul - 8500**
- **Vault - 8200**
- **Etcd - 2379, 2380**
- **MinIO - 9000, 9001**

5. Message Brokers & Streaming

- **Kafka - 9092 (Broker), 2181 (Zookeeper)**
- **RabbitMQ - 5672 (AMQP), 15672 (Management UI)**
- **ActiveMQ - 61616 (OpenWire), 8161 (Web Console)**
- **NATS - 4222**

7. Security & Authentication

- **Keycloak - 8080**
- **OpenVPN - 1194**
- **WireGuard - 51820**
- **LDAP - 389, 636 (SSL)**

8. DevOps Tools

- **Nginx - 80, 443**
- **Apache Tomcat - 8080**
- **Metabase - 3000**
- **Celery Flower - 5555**
- **Django Dev Server - 8000**
- **Flask Dev Server - 5000**
- **NC (Virtual Network Computing - VNC) - 5900**
- **NFS (Network File System) - 2049**

- **Same PORT :- 8080**

  Jenkins - 8080 && Apache Tomcat – 8080

Story - *Jenkins and Apache Tomcat both use port 8080 by default, but if they run on the same system, this creates a conflict since only one service can occupy a port at a time. To resolve this, one of them must be reconfigured to use a different port, ensuring smooth operation without interference.*

1. Changing the Default Port

- *Jenkins: You can change the default port by modifying the JENKINS_PORT variable in the jenkins.xml file or by running Jenkins with --httpPort=9090.*
- *Apache Tomcat: The port can be changed in server.xml (inside the Tomcat conf directory) by modifying the Connector port (<Connector port="8080" ... />).*

2. Running on Different Machines/Containers

- *If Jenkins and Tomcat are running on different servers or inside separate Docker containers, they can both use port 8080 without conflict.*

3. Using a Reverse Proxy

- *A reverse proxy (like Nginx or Apache HTTP Server) can route requests to different services based on the URL path. For example:*
  - *http://yourdomain.com/jenkins → Jenkins (8080)*
  - *http://yourdomain.com/tomcat → Tomcat (8081 or another port)*

Protocols relevance to DevOps workflows: -

The most commonly used protocol is **SSH (port 22),** which is developed for secure remote access. One of the most widely used protocols worldwide is **HTTP/HTTPS (ports 80/443),** which facilitates communication between users and servers. **DNS (port 53)**, or the Domain Name System, is used to resolve domain names to IP addresses. **FTP (port 21)** is used for file transfers. **Jenkins (port 8080)** is a web-based automation server primarily used for CI/CD

Key points:

Docker API: Uses HTTP/HTTPS (REST API) for communication.

Networking: Uses TCP, UDP, VXLAN, and DNS for container communication.

Security: Uses TLS/SSL, OAuth, and JWT for secure access.

Container Communication: Uses HTTP, gRPC, MQTT, and AMQP for service interactions.

Orchestration: Uses Raft & etcd for state management in clusters.