Untitled

🔥 **Essential Port Numbers for DevOps Engineers (with Explanations & Debugging Tips)**

As a **DevOps Engineer**, knowing key port numbers is essential for **networking, security, and troubleshooting**. Here's a **detailed breakdown** of commonly used ports and how to debug them!

🔹 Web & HTTP(S)

- **80 (HTTP)** → Default port for web traffic, used for serving web pages.
- **443 (HTTPS)** → Secure version of HTTP using SSL/TLS encryption.
- **8080 (Alternate HTTP)** → Often used for local development or proxy servers.
- **8443 (Alternate HTTPS)** → HTTPS alternative, sometimes used for secure applications.

Debugging Commands:

```
curl -I http://your-server-ip:80 # Check if HTTP service is running
curl -I https://your-server-ip:443 # Check HTTPS connection
netstat -tulnp | grep :80 # Check if a process is using port 80
```

🔹 SSH & Remote Access

- **22 (SSH)** → Secure Shell for remote server access.
- **3389 (RDP)** → Remote Desktop Protocol for Windows remote access.

Debugging Commands:

```
ssh user@your-server-ip # Test SSH connection
telnet your-server-ip 22 # Check if SSH is reachable
netstat -tulnp | grep :22 # Verify SSH service is running
```

🔹 Database Ports

- **3306 (MySQL/MariaDB)** → Default MySQL/MariaDB database port.
- **5432 (PostgreSQL)** → Default port for PostgreSQL database.
- **1433 (SQL Server)** → Used by Microsoft SQL Server.
- **1521 (Oracle Database)** → Oracle Database Listener port.
- **27017 (MongoDB)** → Default MongoDB connection port.

Debugging Commands:

```
mysql -h your-server-ip -P 3306 -u user -p # Connect to MySQL
pg_isready -h your-server-ip -p 5432 # Check PostgreSQL status
netstat -tulnp | grep :3306 # Verify if MySQL is running
```

◆ DevOps & CI/CD Tools

- **8081 (Nexus Repository)** → Used for artifact management.
- **8082 (JFrog Artifactory)** → JFrog's binary repository manager.
- **5000 (Docker Registry)** → Used for private Docker registries.

Debugging Commands:

```
curl http://your-server-ip:8081 # Check Nexus Repository
curl http://your-server-ip:5000/v2/ # Check Docker Registry
```

◆ Kubernetes & Containers

- **6443 (Kubernetes API Server)** → Controls Kubernetes cluster operations.
- **10250 (Kubelet API)** → Communication between nodes and API server.
- **2379-2380 (etcd Cluster)** → Kubernetes' key-value store backend.

Debugging Commands:

```
kubectl cluster-info # Verify API Server is running
kubectl get nodes # Check node status
etcdctl endpoint status # Verify etcd health
```

◆ Monitoring & Logging

- **9090 (Prometheus)** → Used for metrics scraping and monitoring.
- **3000 (Grafana)** → Web-based visualization dashboard.
- **5601 (Kibana)** → Elasticsearch dashboard for log analysis.
- **9200 (Elasticsearch)** → RESTful search and analytics engine.
- **1514 (Syslog)** → Collects system logs from network devices.

Debugging Commands:

```
curl http://your-server-ip:9090 # Check Prometheus
curl http://your-server-ip:3000 # Check Grafana
curl -X GET http://your-server-ip:9200 # Check Elasticsearch health
```

◆ Cloud & DevOps Services

- **4505-4506 (SaltStack)** → Used by SaltStack for remote execution.

- **9418 (Git)** → Used by Git daemon for repository access.
- **2375-2376 (Docker Daemon)** → Unsecured and secured Docker API ports.

Debugging Commands:

```
docker -H tcp://your-server-ip:2375 info # Test Docker daemon
```

🔹 VPN & Proxy Services

- **1194 (OpenVPN)** → Secure VPN service port.
- **3128 (Squid Proxy)** → Used for caching and filtering proxy servers.
- **1080 (SOCKS Proxy)** → Used for tunneling traffic via proxy.

Debugging Commands:

```
netstat -tulnp | grep :1194 # Verify OpenVPN is running
```

🔹 Message Brokers

- **5672 (RabbitMQ - AMQP)** → Advanced Message Queuing Protocol.
- **1883 (MQTT)** → Lightweight messaging protocol for IoT.
- **61616 (ActiveMQ)** → Message broker service.

Debugging Commands:

```
rabbitmqctl status # Check RabbitMQ status
```

🔹 File Transfer & Storage

- **21 (FTP)** → File Transfer Protocol (insecure).
- **22 (SFTP)** → Secure FTP using SSH.
- **2049 (NFS)** → Network File System for file sharing.
- **445 (SMB)** → Windows file sharing.

Debugging Commands:

```
ftp your-server-ip # Check FTP connectivity
mount -t nfs your-server-ip:/path /mnt # Mount NFS share
```

💡 **Pro Tip:** 🔒 **Restrict unnecessary open ports** using **firewalls, security groups, and network policies** to **minimize attack surfaces**.

Commands to check open ports:

```
netstat -tulnp # List all listening ports
ss -tulnp # Alternative to netstat
```

```
nmap -p 1-65535 your-server-ip # Scan all ports
```