# Chat Transcript

## User

How do you chock running processes in Linux?  
  
What's the difference between 'top', 'ktep; and 'ps?  
  
How to schedule acron job every 15 minutes?  
  
What is the difference between hard link and soft link?  
  
How to find which process is using high memory  
  
How do you resolve merge confiicts?  
  
Difference between git pull', 'git fetch', and 'git clone'?  
  
Use case of 'git stash  
  
What is the gitignore' file and how does it work?  
  
Whats the difference between 'git rebase' and 'git merge  
  
What is the difference between an image and a container?  
  
How to persist data across container restarts?  
  
What is the use of ocker-com-pose?  
  
How do you check logs of a specific container?  
  
How to expose container to the outside world?

## Assistant

Great set of questions! Here's a concise breakdown for each:  
  
🔧 Linux & System Monitoring  
  
1. How do you check running processes in Linux?  
- Use commands like:  
 - ps aux – Lists all running processes.  
 - top – Real-time view of processes.  
 - htop – Enhanced version of top (requires installation).  
 - pgrep <name> – Finds process IDs by name.  
  
2. Difference between top, htop, and ps:  
| Command | Description | Interactive? | Real-time? |  
|--------|-------------|--------------|------------|  
| ps | Snapshot of processes | No | No |  
| top | Real-time process monitor | Yes | Yes |  
| htop | Enhanced top with UI | Yes | Yes |  
  
⏰ Cron Jobs  
  
3. How to schedule a cron job every 15 minutes?  
\*/15 \* \* \* \* /path/to/your/script.sh  
- Add this line using crontab -e.  
  
🔗 File Links in Linux  
  
4. Difference between hard link and soft link:  
| Type | Points to | Breaks if original is deleted? | Cross-filesystem? |  
|------|-----------|-------------------------------|-------------------|  
| Hard Link | File data | No | No |  
| Soft Link (symlink) | File path | Yes | Yes |  
  
📈 Memory Usage  
  
5. How to find which process is using high memory?  
- Use:  
 ps aux --sort=-%mem | head  
 or  
 top  
  
🔀 Git & Version Control  
  
6. How do you resolve merge conflicts?  
- Open conflicting files.  
- Look for markers like:  
 <<<<<<< HEAD  
 Your changes  
 =======  
 Incoming changes  
 >>>>>>> branch-name  
- Edit, test, then:  
 git add <file>  
 git commit  
  
7. Difference between git pull, git fetch, and git clone:  
| Command | Purpose |  
|--------|---------|  
| git clone | Copies a remote repo locally |  
| git fetch | Gets latest changes but doesn't merge |  
| git pull | Fetch + merge |  
  
8. Use case of git stash:  
- Temporarily save uncommitted changes:  
 git stash  
- Useful when switching branches without committing.  
  
9. What is .gitignore and how does it work?  
- A file listing patterns of files/folders Git should ignore.  
- Example:  
 \*.log  
 node\_modules/  
  
10. Difference between git rebase and git merge:  
| Command | Behavior | History |  
|---------|----------|---------|  
| merge | Combines branches with a merge commit | Preserves history |  
| rebase | Moves commits to a new base | Linear history |  
  
🐳 Docker & Containers  
  
11. Difference between an image and a container:  
| Term | Description |  
|------|-------------|  
| Image | Blueprint (read-only) |  
| Container | Running instance of an image |  
  
12. How to persist data across container restarts?  
- Use volumes:  
 docker run -v /host/path:/container/path ...  
  
13. Use of docker-compose:  
- Define and run multi-container Docker apps using docker-compose.yml.  
  
14. How to check logs of a specific container?  
docker logs <container\_name\_or\_id>  
  
15. How to expose container to the outside world?  
docker run -p 8080:80 <image>  
- Maps host port 8080 to container port 80.  
  
Would you like a visual cheat sheet or a downloadable summary of these?

**1. Troubleshooting SSH Issues in EC2**

- Check Security Group: Ensure port 22 is open for your IP.  
- Verify Key Pair: Use the correct `.pem` file and permissions (`chmod 400`).  
- Instance State: Confirm the instance is running.  
- Public IP/DNS: Use the correct address.  
- User Name: Use `ec2-user`, `ubuntu`, etc., depending on the AMI.  
- Check NACLs: Ensure inbound/outbound rules allow SSH.

**2. Difference Between EBS, S3, and EFS**

| Feature | EBS | S3 | EFS |  
|--------|-----|----|-----|  
| Type | Block storage | Object storage | File storage |  
| Use Case | EC2 volumes | Backup, static assets | Shared file system |  
| Access | Single EC2 | HTTP/S | Multiple EC2s |  
| Performance | High IOPS | Varies | Scalable throughput |

**3. IAM Policy vs IAM Role**

- IAM Policy: Defines permissions (what actions are allowed).  
- IAM Role: An identity with permissions that can be assumed by users, services, or applications.

**4. Security Group vs NACL**

| Feature | Security Group | NACL |  
|--------|----------------|------|  
| Level | Instance-level | Subnet-level |  
| State | Stateful | Stateless |  
| Rules | Allow only | Allow & Deny |  
| Use Case | EC2 access control | Subnet traffic filtering |

**5. Setting Up Auto-Scaling**

1. Create a Launch Template.  
2. Define Auto Scaling Group.  
3. Set scaling policies (CPU, memory, etc.).  
4. Attach to Load Balancer (optional).  
5. Monitor with CloudWatch.

**Kubernetes Topics**

**6. Troubleshooting CrashLoopBackOff**

- Run `kubectl describe pod <pod-name>` and `kubectl logs <pod-name>`.  
- Common causes:  
 - Misconfigured environment variables.  
 - Missing dependencies.  
 - Application crash or exit code ≠ 0.  
 - Resource limits too low.

**7. Deployment vs StatefulSet**

| Feature | Deployment | StatefulSet |  
|--------|------------|--------------|  
| Pod Identity | Stateless | Persistent identity |  
| Storage | Ephemeral | Persistent (PVC) |  
| Use Case | Web apps | Databases, queues |

**8. DaemonSet Use Case**

- Ensures a pod runs on every node.  
- Use cases: logging agents, monitoring tools (e.g., Fluentd, Prometheus Node Exporter).

**9. Kubernetes Service**

- Abstracts access to pods.  
- Types: ClusterIP (internal), NodePort, LoadBalancer, ExternalName.  
- Enables load balancing and stable networking.

**10. ConfigMap vs Secret**

| Feature | ConfigMap | Secret |  
|--------|-----------|--------|  
| Data Type | Non-sensitive | Sensitive |  
| Encoding | Plain text | Base64 |  
| Use Case | App configs | Passwords, tokens |

**11. Taints and Tolerations**

- Taints: Prevent pods from being scheduled on nodes.  
- Tolerations: Allow pods to override taints.  
- Use case: dedicate nodes for specific workloads.

**Git Topics**

**12. Resolving Merge Conflicts**

- Use `git status` to identify conflicts.  
- Edit conflicting files manually.  
- Use `git add <file>` to mark resolved.  
- Commit with `git commit`.

**13. Git Pull vs Fetch vs Clone**

| Command | Description |  
|---------|-------------|  
| git clone | Copies a remote repo locally |  
| git fetch | Downloads changes but doesn’t merge |  
| git pull | Fetch + merge changes into current branch |

**14. Use Case of `git stash`**

- Temporarily save changes not ready to commit.  
- Useful when switching branches without committing.

**15. `.gitignore` File**

- Specifies files/folders Git should ignore.  
- Common entries: `node\_modules/`, `.env`, `\*.log`.

# Transcript: AWS, Git, and Kubernetes Concepts

## AWS Topics

### 1. Troubleshooting SSH Issues in EC2

- Check Security Group: Ensure port 22 is open for your IP.  
- Verify Key Pair: Use the correct `.pem` file and permissions (`chmod 400`).  
- Instance State: Confirm the instance is running.  
- Public IP/DNS: Use the correct address.  
- User Name: Use `ec2-user`, `ubuntu`, etc., depending on the AMI.  
- Check NACLs: Ensure inbound/outbound rules allow SSH.

### 2. Difference Between EBS, S3, and EFS

| Feature | EBS | S3 | EFS |  
|--------|-----|----|-----|  
| Type | Block storage | Object storage | File storage |  
| Use Case | EC2 volumes | Backup, static assets | Shared file system |  
| Access | Single EC2 | HTTP/S | Multiple EC2s |  
| Performance | High IOPS | Varies | Scalable throughput |

### 3. IAM Policy vs IAM Role

- IAM Policy: Defines permissions (what actions are allowed).  
- IAM Role: An identity with permissions that can be assumed by users, services, or applications.

### 4. Security Group vs NACL

| Feature | Security Group | NACL |  
|--------|----------------|------|  
| Level | Instance-level | Subnet-level |  
| State | Stateful | Stateless |  
| Rules | Allow only | Allow & Deny |  
| Use Case | EC2 access control | Subnet traffic filtering |

### 5. Setting Up Auto-Scaling

1. Create a Launch Template.  
2. Define Auto Scaling Group.  
3. Set scaling policies (CPU, memory, etc.).  
4. Attach to Load Balancer (optional).  
5. Monitor with CloudWatch.

## Kubernetes Topics

### 6. Troubleshooting CrashLoopBackOff

- Run `kubectl describe pod <pod-name>` and `kubectl logs <pod-name>`.  
- Common causes:  
 - Misconfigured environment variables.  
 - Missing dependencies.  
 - Application crash or exit code ≠ 0.  
 - Resource limits too low.

### 7. Deployment vs StatefulSet

| Feature | Deployment | StatefulSet |  
|--------|------------|--------------|  
| Pod Identity | Stateless | Persistent identity |  
| Storage | Ephemeral | Persistent (PVC) |  
| Use Case | Web apps | Databases, queues |

### 8. DaemonSet Use Case

- Ensures a pod runs on every node.  
- Use cases: logging agents, monitoring tools (e.g., Fluentd, Prometheus Node Exporter).

### 9. Kubernetes Service

- Abstracts access to pods.  
- Types: ClusterIP (internal), NodePort, LoadBalancer, ExternalName.  
- Enables load balancing and stable networking.

### 10. ConfigMap vs Secret

| Feature | ConfigMap | Secret |  
|--------|-----------|--------|  
| Data Type | Non-sensitive | Sensitive |  
| Encoding | Plain text | Base64 |  
| Use Case | App configs | Passwords, tokens |

### 11. Taints and Tolerations

- Taints: Prevent pods from being scheduled on nodes.  
- Tolerations: Allow pods to override taints.  
- Use case: dedicate nodes for specific workloads.

## Git Topics

### 12. Resolving Merge Conflicts

- Use `git status` to identify conflicts.  
- Edit conflicting files manually.  
- Use `git add <file>` to mark resolved.  
- Commit with `git commit`.

### 13. Git Pull vs Fetch vs Clone

| Command | Description |  
|---------|-------------|  
| git clone | Copies a remote repo locally |  
| git fetch | Downloads changes but doesn’t merge |  
| git pull | Fetch + merge changes into current branch |

### 14. Use Case of `git stash`

- Temporarily save changes not ready to commit.  
- Useful when switching branches without committing.

### 15. `.gitignore` File

- Specifies files/folders Git should ignore.  
- Common entries: `node\_modules/`, `.env`, `\*.log`.