# Chat Transcript

## User:

• How do you check running processes in Linux?  
 • What’s the difference between top, htop, and ps?  
 • How to schedule a cron job every 15 minutes?  
 • What is the difference between hard link and soft link?  
 • How to find which process is using high memory?What is the difference between an image and a container?  
 • How to persist data across container restarts?  
 • What is the use of docker-compose?  
 • How do you check logs of a specific container?  
 • How to expose a container to the outside world?

## Assistant:

Great set of questions! Here's a concise breakdown for each:  
  
---  
  
🖥️ Linux Process Management  
  
• How do you check running processes in Linux?  
You can use:  
- ps aux — Lists all running processes.  
- top — Real-time view of system processes.  
- htop — Enhanced version of top with a user-friendly interface (requires installation).  
- pgrep <process\_name> — Finds processes by name.  
  
• What’s the difference between top, htop, and ps?  
Command | Description | Pros | Cons  
--------|-------------|------|-----  
ps | Snapshot of current processes | Simple, script-friendly | Not real-time  
top | Real-time process viewer | Built-in, dynamic | Less intuitive  
htop | Interactive process viewer | Colorful, easy to use, supports mouse | Needs installation  
  
---  
  
⏱️ Cron Jobs  
  
• How to schedule a cron job every 15 minutes?  
Edit the crontab using crontab -e and add:  
\*/15 \* \* \* \* /path/to/your/script.sh  
This runs the script every 15 minutes.

---  
  
🔗 Linux Links  
  
• What is the difference between hard link and soft link?  
Type | Description | Key Traits  
-----|-------------|------------  
Hard Link | Direct pointer to file data | Same inode, file persists even if original is deleted  
Soft Link (Symlink) | Pointer to file path | Different inode, breaks if original is deleted  
  
---  
  
📊 Memory Usage  
  
• How to find which process is using high memory?  
Use:  
- top or htop — Sort by memory usage.  
- ps aux --sort=-%mem | head — Lists top memory-consuming processes.  
  
---  
  
🐳 Docker Basics  
  
• What is the difference between an image and a container?  
Term | Description  
-----|------------  
Image | Blueprint for a container (includes code, dependencies)  
Container | Running instance of an image (isolated environment)  
  
• How to persist data across container restarts?  
Use volumes:  
docker run -v /host/path:/container/path my-image  
Or define volumes in docker-compose.yml.  
  
• What is the use of docker-compose?  
docker-compose helps define and run multi-container Docker applications using a YAML file. It simplifies:  
- Networking  
- Volume management  
- Service orchestration  
  
• How do you check logs of a specific container?  
docker logs <container\_name\_or\_id>  
  
• How to expose a container to the outside world?  
Use the -p flag:  
docker run -p 8080:80 my-image  
This maps port 80 inside the container to port 8080 on the host.  
  
---  
  
Would you like a visual cheat sheet for these commands or help setting up a Docker project or cron job?

# Chat Transcript

## User:

Explain a typical CI/CD pipeline you’ve worked on.  
 • Difference between Declarative and Scripted pipeline?  
 • How to trigger a Jenkins job when code is pushed to GitHub?  
 • How do you handle secrets in Jenkins pipelines?  
 • How to pass parameters between stages?How do you resolve merge conflicts?  
 • Difference between git pull, git fetch, and git clone?  
 • Use case of git stash.  
 • What is the .gitignore file and how does it work?  
 • What’s the difference between git rebase and git merge?

## Assistant:

Here’s a comprehensive breakdown of your questions related to CI/CD pipelines, Git, and Jenkins:  
  
---  
  
✅ Typical CI/CD Pipeline  
  
A typical CI/CD pipeline includes the following stages:  
  
1. Source Stage: Code is pushed to a version control system like GitHub.  
2. Build Stage: Code is compiled and dependencies are installed.  
3. Test Stage: Unit, integration, and other tests are run.  
4. Artifact Packaging: Build artifacts are created (e.g., JAR, Docker image).  
5. Deployment Stage: Artifacts are deployed to staging or production environments.  
6. Notification Stage: Success/failure notifications are sent (Slack, email, etc.).  
  
Tools often used: Jenkins, GitHub Actions, GitLab CI, CircleCI, Docker, Kubernetes, Helm.

---  
  
🔄 Declarative vs Scripted Jenkins Pipelines  
  
| Feature | Declarative Pipeline | Scripted Pipeline |  
|----------------------|-----------------------------------------------|--------------------------------------------|  
| Syntax | Simplified, structured with `pipeline {}` | Groovy-based, more flexible |  
| Error Handling | Built-in | Manual |  
| Readability | Easier to read and maintain | More complex |  
| Use Case | Preferred for most CI/CD tasks | Used for complex logic or legacy scripts |

---  
  
🚀 Trigger Jenkins Job on GitHub Push  
  
To trigger a Jenkins job when code is pushed to GitHub:  
  
1. Install GitHub plugin in Jenkins.  
2. Configure webhook in GitHub:  
 - Go to repo → Settings → Webhooks → Add webhook.  
 - URL: `http://<jenkins-url>/github-webhook/`  
3. Enable GitHub hook trigger in Jenkins job:  
 - In job config → Build Triggers → Check “GitHub hook trigger for GITScm polling”.

---  
  
🔐 Handling Secrets in Jenkins Pipelines  
  
Best practices:  
  
- Use Jenkins Credentials Plugin.  
- Store secrets in Jenkins Credentials Manager.  
- Access them in pipeline using:  
  
withCredentials([string(credentialsId: 'my-secret', variable: 'SECRET')]) {  
 sh "echo $SECRET"  
}  
  
Avoid hardcoding secrets in scripts or environment variables.

---  
  
🔄 Passing Parameters Between Stages  
  
Use environment variables or shared variables:  
  
def myVar  
  
stage('Stage 1') {  
 steps {  
 script {  
 myVar = "Hello"  
 }  
 }  
}  
  
stage('Stage 2') {  
 steps {  
 echo "Value from Stage 1: ${myVar}"  
 }  
}

---  
  
⚔️ Resolving Merge Conflicts  
  
Steps:  
  
1. Run `git pull` or `git merge` and encounter conflict.  
2. Open conflicting files and manually resolve differences.  
3. Mark resolved files: `git add <file>`  
4. Commit the merge: `git commit`  
  
Use tools like VS Code, GitKraken, or `git mergetool` for easier resolution.

---  
  
🔁 Git Pull vs Fetch vs Clone  
  
| Command | Description |  
|-------------|-------------|  
| `git pull` | Fetches and merges changes from remote to local branch |  
| `git fetch` | Fetches changes but does \*\*not\*\* merge them |  
| `git clone` | Copies entire remote repository to local machine |

---  
  
📦 Use Case of `git stash`  
  
Temporarily saves changes that are not ready to commit:  
  
git stash  
  
Use case: You’re working on a feature but need to switch branches without committing.  
  
Retrieve with:  
  
git stash pop

---  
  
📁 What is `.gitignore`?  
  
A file that tells Git which files/folders to ignore:  
  
Example:  
  
node\_modules/  
\*.log  
.env  
  
Useful for excluding sensitive files, build artifacts, or OS-specific files.

---  
  
🔀 Git Rebase vs Merge  
  
| Feature | `git merge` | `git rebase` |  
|---------------|--------------------------------------|---------------------------------------|  
| History | Preserves full history | Rewrites history |  
| Use Case | Collaborative work, safe | Clean linear history, advanced users |  
| Command | `git merge branch-name` | `git rebase branch-name` |  
  
Rebase is cleaner but riskier if used incorrectly (especially on shared branches).

# Conversation Transcript

## User:

How does Terraform maintain the state of resources?  
 • What are Terraform modules?  
 • How to manage sensitive variables in Terraform?  
 • What is the purpose of terraform validate and terraform fmt?  
 • How do you handle provisioning in different environments (dev/stage/prod)?  
  
🔹 AWS & Cloud  
 • How to troubleshoot SSH issues in an EC2 instance?  
 • What is the difference between EBS, S3, and EFS?  
 • How does IAM policy differ from IAM role?  
 • Difference between security group and NACL?  
 • How to set up auto-scaling for an application?  
  
🔹 Kubernetes (K8s)  
 • Difference between Deployment and StatefulSet?  
 • What is a DaemonSet used for?  
 • How does a Service in Kubernetes work?  
 • What is a ConfigMap vs Secret?  
 • What are taints and tolerations?  
 • How do liveness and readiness probes work?  
 • How to troubleshoot a pod stuck in CrashLoopBackOff?

## Assistant:

You've asked a great set of questions across \*\*Terraform\*\*, \*\*AWS & Cloud\*\*, and \*\*Kubernetes\*\*. Here's a concise breakdown for each topic:  
  
---  
  
### 🔹 \*\*Terraform\*\*  
  
\*\*1. How does Terraform maintain the state of resources?\*\*   
Terraform uses a \*\*state file (`terraform.tfstate`)\*\* to keep track of the infrastructure it manages. This file records resource metadata and dependencies, enabling Terraform to determine what changes need to be applied. For team collaboration, it's common to store this state remotely (e.g., in \*\*Terraform Cloud\*\*, \*\*S3 with DynamoDB for locking\*\*).  
  
\*\*2. What are Terraform modules?\*\*   
Modules are reusable containers for Terraform configurations. A module can include resources, input/output variables, and other modules. They help organize code and promote reuse across environments or projects.  
  
\*\*3. How to manage sensitive variables in Terraform?\*\*   
- Use `variable` blocks with `sensitive = true`.  
- Store secrets in secure backends like \*\*AWS Secrets Manager\*\*, \*\*Vault\*\*, or \*\*environment variables\*\*.  
- Avoid committing `.tfstate` files to version control if they contain sensitive data.  
- Use remote state with encryption.  
  
\*\*4. What is the purpose of `terraform validate` and `terraform fmt`?\*\*   
- `terraform validate`: Checks whether the configuration is syntactically valid.  
- `terraform fmt`: Formats Terraform code according to standard style guidelines.  
  
\*\*5. How do you handle provisioning in different environments (dev/stage/prod)?\*\*   
- Use \*\*workspaces\*\* (`terraform workspace`) or separate directories.  
- Parameterize environment-specific values using \*\*variables\*\* and \*\*backend configurations\*\*.  
- Use \*\*modules\*\* to share common infrastructure logic.  
  
---  
  
### 🔹 \*\*AWS & Cloud\*\*  
  
\*\*1. How to troubleshoot SSH issues in an EC2 instance?\*\*   
- Check \*\*security group rules\*\* (port 22 open).  
- Verify \*\*key pair\*\* and correct private key permissions.  
- Ensure the \*\*EC2 instance is running\*\* and has a public IP.  
- Confirm \*\*network ACLs\*\* and \*\*route tables\*\* allow traffic.  
- Use \*\*EC2 serial console\*\* or \*\*SSM Session Manager\*\* if SSH fails.  
  
\*\*2. Difference between EBS, S3, and EFS?\*\*   
- \*\*EBS\*\*: Block storage for EC2, persistent and fast.  
- \*\*S3\*\*: Object storage, scalable and ideal for backups, media, etc.  
- \*\*EFS\*\*: Network file system, shared across multiple EC2s.  
  
\*\*3. How does IAM policy differ from IAM role?\*\*   
- \*\*IAM Policy\*\*: Defines permissions.  
- \*\*IAM Role\*\*: An identity with permissions that can be assumed by users, services, or applications.  
  
\*\*4. Difference between Security Group and NACL?\*\*   
- \*\*Security Group\*\*: Stateful, instance-level firewall.  
- \*\*NACL (Network ACL)\*\*: Stateless, subnet-level firewall.  
  
\*\*5. How to set up auto-scaling for an application?\*\*   
- Create an \*\*Auto Scaling Group (ASG)\*\*.  
- Define \*\*launch template/configuration\*\*.  
- Set \*\*scaling policies\*\* (CPU, memory, etc.).  
- Attach to \*\*Elastic Load Balancer (ELB)\*\* for traffic distribution.  
  
---  
  
### 🔹 \*\*Kubernetes (K8s)\*\*  
  
\*\*1. Difference between Deployment and StatefulSet?\*\*   
- \*\*Deployment\*\*: Manages stateless apps.  
- \*\*StatefulSet\*\*: Manages stateful apps with persistent identity and storage.  
  
\*\*2. What is a DaemonSet used for?\*\*   
Ensures a pod runs on \*\*every node\*\* (or selected nodes). Used for logging, monitoring agents, etc.  
  
\*\*3. How does a Service in Kubernetes work?\*\*   
Abstracts access to pods. Provides \*\*stable IP\*\* and \*\*DNS name\*\*. Types include:  
- ClusterIP (default)  
- NodePort  
- LoadBalancer  
- ExternalName  
  
\*\*4. What is a ConfigMap vs Secret?\*\*   
- \*\*ConfigMap\*\*: Stores non-sensitive config data.  
- \*\*Secret\*\*: Stores sensitive data (base64 encoded).  
  
\*\*5. What are taints and tolerations?\*\*   
- \*\*Taints\*\*: Prevent pods from being scheduled on nodes.  
- \*\*Tolerations\*\*: Allow pods to override taints.  
  
\*\*6. How do liveness and readiness probes work?\*\*   
- \*\*Liveness Probe\*\*: Checks if the app is alive; restarts if it fails.  
- \*\*Readiness Probe\*\*: Checks if the app is ready to serve traffic.  
  
\*\*7. How to troubleshoot a pod stuck in CrashLoopBackOff?\*\*   
- Check logs: `kubectl logs <pod-name>`  
- Describe pod: `kubectl describe pod <pod-name>`  
- Look for misconfigurations, missing dependencies, or failed health checks.  
  
---  
  
Would you like this compiled into a PDF or cheat sheet for quick reference? Or should we dive deeper into any specific topic?