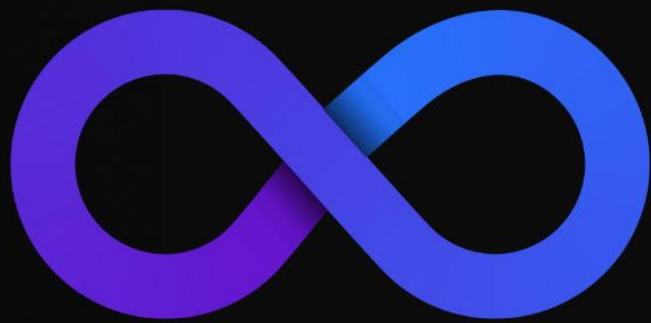




DevOps Cheat Sheet Commands



By DevOps Shack

[Click here for DevSecOps & Cloud DevOps Course](#)

DevOps Shack

Corporate DevOps Workbook

Introduction

Welcome to the **Corporate DevOps Workbook**—your go-to resource for mastering daily DevOps operations. Whether you're a **DevOps engineer, SRE, or system administrator**, this guide provides a **comprehensive reference** for managing infrastructure, automating deployments, and troubleshooting issues.

This workbook covers critical **commands, workflows, and best practices** across industry-standard tools such as:

- Git** – Version Control & Collaboration
- Docker** – Containerization & Image Management
- Kubernetes** – Container Orchestration & Scaling
- Terraform** – Infrastructure as Code (IaC)
- Azure DevOps** – CI/CD Pipelines & Automation
- Linux** – System Administration & Networking

This Workbook?

- ◊ **Quick & Easy Access** – A single source for the most used DevOps commands.
- ◊ **Practical Use Cases** – Commands are structured with real-world applications.
- ◊ **Troubleshooting & Optimization** – Common issues and solutions for DevOps workflows.
- ◊ **Security & Best Practices** – Safe usage guidelines for each tool to avoid critical mistakes.

Below is a **quick reference table** featuring the **Top 40 Most Used DevOps** Whether you're working on **deployments, infrastructure provisioning, or troubleshooting**, this workbook will help you **increase efficiency and reduce downtime**. Let's dive in! 

Top 40 Most Used DevOps Commands (Quick Reference)

◊ Git (Version Control)

| Command | Description |
|--|--|
| <code>git status</code> | Check the status of working directory |
| <code>git pull origin <branch></code> | Fetch and merge latest changes from remote |
| <code>git add .</code> | Stage all modified files for commit |
| <code>git commit -m "message"</code> | Commit staged changes with a message |
| <code>git push origin <branch></code> | Push local commits to remote repository |
| <code>git checkout -b <branch></code> | Create and switch to a new branch |
| <code>git merge <branch></code> | Merge specified branch into the current branch |
| <code>git rebase <branch></code> | Reapply commits on top of another branch |
| <code>git reset --soft <commit></code> | Undo commits but keep changes staged |
| <code>git reset --hard <commit></code> | WARNING: Reset to a previous commit, losing all changes |

Docker (Containers & Images)

| Command | Description |
|---|--|
| <code>docker ps</code> | List running containers |
| <code>docker ps -a</code> | List all containers (running & stopped) |
| <code>docker images</code> | List all available Docker images |
| <code>docker run -d -p 8080:80 <image></code> | Run a container in detached mode with port mapping |
| <code>docker exec -it <containerid> bash</code> | Open shell inside a running container |
| <code>docker logs <container-id></code> | View logs of a running container |
| <code>docker stop <containerid></code> | Stop a running container |
| <code>docker rm <container-id></code> | WARNING: Remove a stopped container |
| <code>docker rmi <image></code> | WARNING: Delete a Docker image |
| <code>docker system prune -a</code> | WARNING: Remove unused images, containers, and networks |



Kubernetes (K8s)

| Command | Description |
|---|---|
| <code>kubectl get pods</code> | List all running pods |
| <code>kubectl describe pod <podname></code> | Get detailed information about a pod |
| <code>kubectl logs <pod-name></code> | View logs of a pod |
| <code>kubectl get deployments</code> | List all deployments |
| <code>kubectl scale deployment <name> --replicas=3</code> | Scale deployment to 3 replicas |
| <code>kubectl rollout status deployment <name></code> | Check deployment rollout status |
| <code>kubectl exec -it <pod-name> -- /bin/sh</code> | Access a running pod's shell |
| <code>kubectl delete pod <podname></code> | WARNING: Delete a specific pod |
| <code>kubectl delete deployment <name></code> | WARNING: Remove a deployment |
| <code>kubectl drain <node></code> | WARNING: Prepare a node for maintenance by evicting pods |

🌐 Terraform (Infrastructure as Code)

| Command | Description |
|---|--|
| terraform init | Initialize Terraform working directory |
| terraform fmt | Format Terraform configuration files |
| terraform validate | Validate Terraform configuration files |
| terraform plan | Preview changes before applying them |
| terraform apply | Apply the Terraform configuration |
| terraform refresh | Update Terraform state file with real infrastructure data |
| terraform destroy | WARNING: Destroy all Terraform-managed resources |
| terraform state list | List all managed resources |
| terraform state show <resource> | Show details of a specific resource |
| terraform force-unlock <id> | WARNING: Manually unlock Terraform state (use with caution) |

Linux & Shell Commands

| Command | Description |
|--|---|
| <code>ls -la</code> | List files and directories with detailed information |
| <code>cd <directory></code> | Change directory |
| <code>mkdir <directory></code> | Create a new directory |
| <code>rm -rf <directory></code> | WARNING: Remove a directory and its contents permanently |
| <code>chmod +x <file></code> | Change file permissions to executable |
| <code>chown user:group <file></code> | Change file ownership |
| <code>ps aux</code> | List running processes |
| <code>kill -9 <PID></code> | WARNING: Forcefully terminate a process |
| <code>netstat -tulnp</code> | Show active network connections |
| <code>tail -f /var/log/syslog</code> | View system logs in real-time |

Next Set of DevOps Commands

Introduction

Now that we've covered the Top 40 Most Used DevOps Commands, let's dive deeper into specific tools and workflows.

In the next sections, you'll find essential daily commands for:

- Git – Version Control
- Docker – Container Management
- Kubernetes – Orchestration
- Terraform – Infrastructure as Code
- Azure DevOps – CI/CD & Pipelines
- Linux – System Administration

Each section includes:

-  Frequently Used Commands
-  Real-World Use Cases
-  Troubleshooting Tips

These commands will serve as a quick reference guide for DevOps engineers to efficiently manage deployments, infrastructure, and automation. Let's get started! 

1. Git & Version Control

Basic Commands

| Command | Description |
|----------------------|----------------------------------|
| git init | Initialize a new Git repository |
| git clone <repo-url> | Clone an existing repository |
| git status | Show status of working directory |
| Command | Description |

| | |
|--------------------------|---|
| git add <file> | Stage changes for commit |
| git commit -m "message" | Commit staged changes |
| git push origin <branch> | Push commits to a remote repository |
| git pull origin <branch> | Fetch and merge changes from remote |
| git log --oneline | Show commit history in short format |
| git diff | Show differences in modified files |
| git stash | Temporarily save changes without committing |

Branching & Merging

| Command | Description |
|--------------------------|--|
| git branch | List all branches |
| git checkout -b <branch> | Create and switch to a new branch |
| git merge <branch> | Merge specified branch into current branch |
| git rebase <branch> | Reapply commits on top of another branch |
| git branch -d <branch> | Delete a local branch |

Reverting & Resetting

| Command | Description |
|---------------------------|--|
| git reset --hard <commit> | Reset repository to a specific commit |
| git revert <commit> | Undo changes by creating a new commit |
| git checkout -- <file> | Discard changes in a working directory |

2. Docker & Containerization

Basic Commands

| Command | Description |
|-------------------------------------|--|
| docker --version | Show Docker version |
| docker ps | List running containers |
| docker ps -a | List all containers (running & stopped) |
| docker images | List all available images |
| docker build -t <image-name> . | Build a Docker image from Dockerfile |
| docker run -d -p 8080:80 <image> | Run a container in detached mode with port mapping |
| docker stop <container-id> | Stop a running container |
| docker restart <container-id> | Restart a container |
| docker logs <container-id> | View logs of a running container |
| docker exec -it <container-id> bash | Access a running container's shell |

3. Kubernetes (K8s)

Pod Management

| Command | Description |
|--|------------------------------|
| kubectl get pods | List all running pods |
| kubectl describe pod <pod-name> | Show details of a pod |
| kubectl logs <pod-name> | Fetch logs from a pod |
| kubectl delete pod <pod-name> | Delete a pod |
| kubectl exec -it <pod-name> -- /bin/sh | Access a running pod's shell |

Deployments & Scaling

| Command | Description |
|---|---------------------------------|
| kubectl get deployments | List all deployments |
| kubectl create deployment <name> -image=<image> | Create a deployment |
| kubectl scale deployment <name> --replicas=3 | Scale deployment to 3 replicas |
| kubectl rollout status deployment <name> | Check deployment rollout status |
| kubectl delete deployment <name> | Delete a deployment |

4. Terraform (IaC - Infrastructure as Code)

| Command | Description |
|---------------------------------|--|
| terraform init | Initialize Terraform working directory |
| terraform fmt | Format Terraform files |
| terraform validate | Validate Terraform configuration |
| terraform plan | Show execution plan before applying |
| terraform apply | Apply the Terraform configuration |
| terraform destroy | Destroy all Terraform-managed infrastructure |
| terraform state list | List all managed resources |
| terraform state show <resource> | Show details of a specific resource |
| terraform output | Show Terraform outputs |
| terraform refresh | Sync state with real infrastructure |

5. Azure DevOps & CI/CD Pipelines

Repositories

| Command | Description |
|--|-------------------------|
| az repos list | List all repositories |
| az repos create --name <repository name> | Create a new repository |

git push --set-upstream origin <branch> Push a new branch to Azure Repos

Pipelines & Releases

| Command | Description |
|---|--------------------------|
| az pipelines list | List all pipelines |
| az pipelines run --name <pipeline name> | Run a specific pipeline |
| az artifacts list | List stored artifacts |
| az pipelines releases list | List release pipelines |
| az pipelines variable-group list | List all variable groups |

6. Linux & Shell Scripting

File & Directory Management

| Command | Description |
|--------------------|-----------------------------------|
| ls -la | List files with details |
| cd <directory> | Change directory |
| mkdir <directory> | Create a new directory |
| rm -rf <directory> | Remove directory and its contents |

User & Permission Management

| Command | Description |
|-------------------------|-------------------------|
| whoami | Show current user |
| chmod +x <file> | Change file permissions |
| chown user:group <file> | Change file ownership |

Process & Networking

| Command | Description |
|----------------|---------------------------------|
| ps aux | List running processes |
| kill -9 <PID> | Terminate a process |
| netstat -tulnp | Show active network connections |

7. Monitoring & Logging

Prometheus & Grafana

| Command | Description |
|--|----------------------------|
| kubectl get pods -n monitoring | List monitoring stack pods |
| kubectl logs <pod-name> -n monitoring | View Prometheus logs |
| kubectl port-forward svc/grafana 3000:3000 -n monitoring | Access Grafana |

Log Management with ELK Stack

| Command | Description |
|--|------------------------------|
| curl -XGET "http://localhost:9200/_cat/indices?v" | List Elasticsearch indices |
| tail -f /var/log/syslog | View system logs in realtime |

8. Database & SQL Operations

Basic Commands

| Command | Description |
|--|--------------------------------------|
| mysql -u root -p | Login to MySQL database |
| SHOW DATABASES; | List all databases |
| USE <database>; | Select a database |
| SHOW TABLES; | List all tables in the database |
| SELECT * FROM <table>; | Retrieve data from a table |
| mysqldump -u user -p database > backup.sql | Backup a MySQL database |
| psql -U postgres -d mydb | Connect to PostgreSQL |
| SELECT COUNT(*) FROM <table>; | Count records in a table |
| DROP DATABASE <database>; | Delete a database |
| ALTER TABLE <table> ADD COLUMN <column> TYPE; | Add a column to an existing table |

Conclusion

The **Corporate DevOps Workbook** serves as a **comprehensive guide** for navigating daily DevOps operations efficiently. From **Git version control** to **container management with Docker and Kubernetes**, **infrastructure automation with Terraform**, and **CI/CD pipelines in Azure DevOps**, this resource equips engineers with **critical commands, troubleshooting techniques, and best practices** to streamline workflows.

Key Takeaways:

- Efficiency Boost** – A single reference to execute DevOps tasks faster and with greater confidence.
- Reduced Errors** – Color-coded safety indicators help prevent critical mistakes.
- Troubleshooting Ready** – Includes solutions to common issues across multiple DevOps tools.
- Security & Best Practices** – Guidelines to enhance security, automation, and operational resilience.

As DevOps continues to evolve, so should your skill set. Keep this workbook handy, update it with new findings, and use it as a **living document** to adapt to emerging technologies and best practices.