

AWS 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
- **Jenkins** – Automation Server for CI/CD Pipelines
- **Ansible** – Configuration Management & Automation
- **Terraform** – Infrastructure as Code (IaC)
- **AWS DevOps** – CI/CD Pipelines & Automation
- **Linux** – System Administration & Networking

Why This Workbook?

- ◆ **Quick & Easy Access** – A compact, go-to collection of the most frequently used DevOps commands.
- ◆ **Practical Use Cases** – Commands are presented with context-driven use cases and tips.
- ◆ **Troubleshooting & Optimization** Covers common errors and practical solutions to keep systems running smoothly.
- ◆ **Security & Best Practices** – Ensures safe usage of powerful tools to avoid costly mistakes in production environments.

Below is a **quick reference table** featuring the **Top 40 Most Used DevOps Commands** categorized by tool, with **color-coded indicators** for command safety:

AWS DevOps Workbook

- ☐ **Safe Commands** – Everyday usage, minimal risk.
- ☐ **Caution Commands** – Requires attention, potential risk.
- ☒ **Destructive Commands** – High risk, irreversible actions.

Whether you're deploying applications, provisioning infrastructure, or resolving system issues, this workbook is your guide to working smarter, faster, and with fewer errors.

Top 60 Most Used DevOps Commands (Quick Reference)

◆ Git (Version Control)

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

AWS DevOps Workbook



Docker (Containers & Images)

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

AWS DevOps Workbook






Kubernetes (K8s)

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

AWS DevOps Workbook

✿ Jenkins (CI/CD Automation)

Command	Description	Safety
<code>jenkins --version</code>	Show installed Jenkins version	<input type="checkbox"/>
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ list-jobs</code>	List all Jenkins jobs using CLI	<input type="checkbox"/>
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ build <job-name></code>	Trigger a Jenkins job manually via CLI	<input type="checkbox"/>
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ create-job <job-name> < job.xml</code>	Create a new Jenkins job from XML config	<input type="checkbox"/>
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ delete-job <job-name></code>	■ Delete an existing Jenkins job	
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ reload-configuration</code>	Reload Jenkins job configurations without restart	<input type="checkbox"/>
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ safe-restart</code>	Restart Jenkins safely after finishing builds	<input type="checkbox"/>
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ disable-job <job-name></code>	Disable a job temporarily	
<code>java -jar jenkins-cli.jar -s http://localhost:8080/ enable-job <job-name></code>	Enable a previously disabled job	

AWS DevOps Workbook





Ansible (Automation & Configuration Management)

Command	Description	Safety
<code>ansible --version</code>	Show installed Ansible version	<input type="checkbox"/>
<code>ansible all -m ping</code>	Ping all hosts to check connectivity	<input type="checkbox"/>
<code>ansible-playbook playbook.yml</code>	Run a playbook on target hosts	<input type="checkbox"/>
<code>ansible-inventory --list</code>	List hosts from dynamic/static inventory	<input type="checkbox"/>
<code>ansible all -m shell -a "uptime"</code>	Run shell command on all hosts	<input type="checkbox"/>
<code>ansible webserver -a "rm -rf /tmp/*"</code>	Run a destructive command on a group	
<code>ansible-galaxy install <role></code>	Install roles from Ansible Galaxy	<input type="checkbox"/>
<code>ansible-vault encrypt secrets.yml</code>	Encrypt sensitive YAML files	<input type="checkbox"/>
<code>ansible-vault decrypt secrets.yml</code>	Decrypt previously encrypted files	<input type="checkbox"/>
<code>ansible --version</code>	Show installed Ansible version	<input type="checkbox"/>

AWS DevOps Workbook

Terraform (Infrastructure as Code)

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

AWS DevOps Workbook

Linux & Shell Commands

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

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

AWS DevOps Workbook

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

AWS DevOps Workbook

2. Docker & Containerization

Basic Commands

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

AWS DevOps Workbook

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

AWS DevOps Workbook

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

AWS DevOps Workbook

5. AWS DevOps & CI/CD Pipelines

Repositories (AWS CodeCommit)

Command	Description
<code>aws codecommit list-repositories</code>	List all CodeCommit repositories
<code>aws codecommit create-repository --repository-name <repo></code>	Create a new repository
<code>git remote add origin <repo-URL></code>	Connect your local repo to CodeCommit
<code>git push -u origin <branch></code>	Push a new branch to CodeCommit

Build & Deployment (AWS CodeBuild & CodeDeploy)

Command	Description
<code>aws codebuild list-projects</code>	List all CodeBuild projects
<code>aws codebuild start-build --project-name <project></code>	Start a CodeBuild build
<code>aws deploy list-applications</code>	List CodeDeploy applications
<code>aws deploy create-deployment --application-name <app> --deployment-group-name <group> --s3-location bucket=<bucket>,key=<key>,bundleType=zip</code>	Trigger a deployment via CodeDeploy

AWS DevOps Workbook

Pipelines & Releases

Command	Description
<code>aws codepipeline list-pipelines</code>	List all CodePipeline pipelines
<code>aws codepipeline start-pipeline-execution --name <pipeline></code>	Trigger a pipeline run
<code>aws codepipeline get-pipeline-state --name <pipeline></code>	Check current status of a pipeline
<code>aws codepipeline disable-stage-transition --pipeline-name <pipeline> --stage-name <stage> --transition-type Inbound --reason "Reason"</code>	Disable stage transition (useful for pause/debug)

AWS DevOps Workbook

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

AWS DevOps Workbook

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 real-time

AWS DevOps Workbook

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

AWS DevOps Workbook

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 AWS 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.