

DEVOPS

INTERMEDIATE

3 DAYS

Docker and Jenkins CI/CD Bootcamp

Build a strong foundation in container delivery using Docker and CI/CD automation with Jenkins, from Dockerfile best practices to pipeline-as-code. Learn how to build, test, tag, publish, and deploy container images reliably with secure credentials and real troubleshooting of common CI/CD failures.

Why Docker + Jenkins CI/CD?

Faster CI/CD delivery

Automate builds, tests, and deployments with repeatable pipelines.

Scalable automation

Jenkins pipelines scale across teams and services with shared pipeline libraries.

Consistent environments

Docker ensures the same runtime across developer, test, and production stages.

Improved release velocity

Ship more frequently with stable, automated build-and-release workflows.

Reduced deployment failures

Containerized builds minimize “works on my machine” issues.

What you'll be able to do

- ✓ Understand Docker fundamentals and container-based packaging.
- ✓ Build Docker images using best practices including Dockerfile structure, multi-stage builds, and tagging strategy.
- ✓ Run and troubleshoot containers locally using core Docker commands.
- ✓ Understand Jenkins fundamentals including pipelines, agents, jobs, credentials, and plugins.
- ✓ Build CI pipelines using Jenkins Pipeline (Jenkinsfile) with declarative pipeline structure.
- ✓ Automate Docker image build and push workflows using Jenkins.
- ✓ Apply pipeline best practices including reusable stages, artifacts, environment variables, and parameters.
- ✓ Implement secure automation patterns including Jenkins credentials binding and least privilege practices.
- ✓ Deliver an end-to-end Docker-based CI/CD workflow as a capstone project.

Who this is for

- DevOps Engineers
- Build and Release Engineers
- Cloud Engineers
- Backend Developers working with deployment pipelines
- SRE teams supporting CI/CD platforms

Curriculum

01 Docker Fundamentals (Container Packaging Basics)

- What containers are and how they differ from VMs
- Docker architecture (images, containers, registry concepts)
- Core Docker workflows (build, run, pull, push)
- Container lifecycle and common use cases
- **Hands-on:** Lab: Run a containerized sample app locally and inspect logs and processes

02 Building Docker Images with Best Practices

- Dockerfile structure and layer caching fundamentals
- Multi-stage builds for smaller, faster images
- Tagging strategy (semantic versions, git-sha tags, avoid latest)
- Image hardening basics (minimal base images, non-root user concepts)
- **Hands-on:** Lab: Build a production-ready multi-stage Docker image and compare image size and startup time

03 Running and Troubleshooting Containers Locally

- Core docker commands (ps, logs, exec, inspect)
- Networking basics (ports, bridge networks)
- Volumes basics (persist data, mount configs)
- Troubleshooting patterns (port conflicts, env issues, crashing containers)
- **Hands-on:** Lab: Debug common container issues (wrong port, missing env var, permission issue)

04 Jenkins Fundamentals (Pipelines, Agents, Jobs, Credentials)

- Jenkins architecture (controller, agents, executors)
- Jobs vs Pipelines and when to use each
- Credentials management and secure secret handling
- Plugins overview and common pipeline extensions
- **Hands-on:** Lab: Setup Jenkins pipeline job and configure credentials securely

05 Jenkins Pipeline (Jenkinsfile) Declarative Fundamentals

- Declarative pipeline structure (pipeline, agent, stages, steps)
- Environment variables, parameters, and options
- Post actions (always, success, failure) and notifications concepts
- Artifacts handling and workspace basics
- **Hands-on:** Lab: Create a Jenkinsfile that builds and tests an application with parameterized runs

06 Automating Docker Build and Push with Jenkins

- Docker build in pipelines (agent requirements and Docker daemon access)
- Registry authentication and push workflow
- Image versioning strategy and build metadata
- Publishing artifacts and tagging releases
- **Hands-on:** Lab: Build and push Docker images to a registry from Jenkins using secure credentials

07 Pipeline Best Practices (Reusable Stages, Artifacts, Parameters)

- Reusable pipeline patterns (shared libraries concept, stage reuse)
- Artifacts and dependencies between stages
- Environment variables, parameters, and secrets-safe usage
- Efficient pipelines (caching concepts, parallel stages overview)
- **Hands-on: Lab: Refactor a pipeline to add reusable stages, parameters, and artifact publishing**

08 Secure Automation Patterns in Jenkins (Least Privilege)

- Credentials binding best practices (never echo secrets)
- Least privilege access for registries and SCM
- Securing agents and restricting permissions
- Auditability and pipeline governance basics
- **Hands-on: Lab: Implement secure credentials binding and validate secrets are masked in logs**

09 Capstone Project (Docker-Based CI/CD Workflow with Jenkins)

- Capstone goal: Deliver an end-to-end Docker build and release pipeline
- Build Docker image using production-ready Dockerfile (multi-stage, tags)
- Run tests and package artifacts
- Push image to registry with secure credentials
- Add parameters for versioning and environment selection
- **Hands-on: Capstone Lab: Present the working Jenkins pipeline from commit to pushed image with evidence**