CareerByteCode
Learning Made simple

CareerByteCode Admissions
support@careerbytecode.in
www.careerbytecode.in



CAREERBYTECODE PROGRAM

CyberForge: DevSecOps & Cloud Security Engineering

The CareerByteCode CyberForge - DevSecOps & Cloud Security Engineering is an eight-week practical program for learners with basic DevOps knowledge who want to integrate security into the software delivery lifecycle.

Build secure pipelines.
Ship with confidence.

Mentors: Jenefer Rexee George & Daisy Grace Thomas

8 weeks · 24 live classes · 48 instructor-led hours · 4 hands-on projects · Hybrid: classroom + online

Course Overview

LEARNING MADE SIMPLE

The CareerByteCode CyberForge - DevSecOps & Cloud Security Engineering is an eight-week practical program for learners with basic DevOps knowledge who want to integrate security into the software delivery lifecycle. The course demonstrates how to identify, prioritise and remediate security risks in source code, open-source dependencies, secrets, CI/CD pipelines, APIs, container images, Kubernetes workloads and Infrastructure as Code.

Learners will work with GitHub Actions and Jenkins, use commonly adopted security tools, practise secure pipeline design and complete four guided projects. The cloud-security component will use one cloud provider selected for the batch (AWS or Azure), rather than promising hands-on delivery on both platforms.

8	24	48	16	4
WEEKS	LIVE CLASSES	HOURS	MODULES	PROJECTS

Your Mentors

LEARN FROM PRACTITIONERS



Jenefer Rexee George

AI ML DATA CONSULTANT

Jenefer Rexee George is an AI and ML Data Consultant and Software Engineer (AI/ML) who builds intelligent systems that combine data, machine learning and modern AI to solve real-world problems.



Daisy Grace Thomas

AI CONSULTANT · AI PRODUCT & AUTOMATION ENGINEER

Daisy Grace Thomas is an AI Product and Automation Engineer and Business Analyst with 9+ years delivering enterprise solutions across IT services, EdTech, finance and aerospace.

CareerByteCode
Learning Made simple

Table of Contents

CYBERFORGE DEVSECOPS

01 Program at a Glance

02 Who This Course Is for

03 Before You Start: Prerequisites

04 What You'll Achieve: Learning Outcomes

05 Week-Wise Curriculum Overview

06 Four Major Hands-On Projects

07 Tools and Technologies

08 Assessment and Evaluation

09 Training Methodology and Delivery

10 Career Support and Portfolio

11 Commercial Details

12 Important Academic and Lab Safety Notes

How every class works

Each class combines concept explanation, a live instructor demonstration, guided practice and a short recap. You practise between classes and finish four projects for your portfolio. Delivery: Hybrid, classroom + online. Next batch dates are shared on enrolment.

CareerByteCode

Learning Made simple

Program at a Glance

CYBERFORGE DEVSECOPS

Duration	8 weeks
Schedule	3 classes per week; 24 classes total
Class duration	2 hours
Instructor-led learning	48 hours
Target audience	Students, freshers, working IT professionals, DevOps/Cloud Engineers and cybersecurity learners
Entry level	Basic DevOps knowledge; limited security knowledge
Delivery	Hybrid: classroom + online
CI/CD platforms	GitHub Actions and Jenkins
Cloud component	One cloud provider per batch: AWS or Azure (selected before delivery)
Security scope	SAST, DAST, SCA, secrets scanning, container, Kubernetes and Terraform security
Projects	4 major hands-on projects
Course fee	₹40,000 + applicable GST

Who This Course Is for

CYBERFORGE DEVSECOPS

CareerByteCode
Learning Made simple

Students, freshers, working IT professionals, DevOps/Cloud Engineers and cybersecurity learners

Before You Start: Prerequisites

CYBERFORGE DEVSECOPS

- Basic understanding of the software development or DevOps lifecycle.
- Basic familiarity with Linux commands and working in a terminal.
- Basic Git knowledge: clone, add, commit, push, pull and branches.
- A conceptual understanding of CI/CD pipelines and build/deployment stages.
- Basic familiarity with Docker is helpful; container fundamentals will be revised before security labs.
- No prior professional cybersecurity experience is required.
- A laptop with internet access and permission to install or run course tools; cloud account requirements will be communicated before the relevant lab.

What You'll Achieve: Learning Outcomes

CYBERFORGE DEVSECOPS

By the end of the program, learners should be able to:

- Explain DevSecOps principles and identify security activities across the software delivery lifecycle.
- Recognise common application security risks and use threat-modelling concepts to identify potential attack paths.
- Integrate SAST, SCA and secret scanning into GitHub Actions and Jenkins pipelines.
- Run introductory DAST checks against an authorised test application and interpret findings.

- Triage vulnerabilities by severity, exploitability, business context and remediation priority.
- Build and scan Docker images, reduce image attack surface and understand SBOM and image-signing concepts.
- Apply Kubernetes security fundamentals including RBAC, namespaces, pod security, network policies and configuration checks.
- Scan Terraform code for insecure configurations and apply safer Infrastructure as Code practices.
- Understand cloud shared responsibility, IAM least privilege, secure networking, storage and logging using one selected cloud provider.
- Explain security gates, exception handling, evidence collection and continuous improvement in a secure delivery workflow.
- Demonstrate an end-to-end secure pipeline through a portfolio-ready capstone.

CareerByteCode
Learning Made simple

Week-Wise Curriculum Overview

CYBERFORGE DEVSECOPS

WEEK	FOCUS	MODULES	CLASSES
1	DevSecOps Foundations and Security Fundamentals	2	Classes 1-3
2	Secure Source Code, SAST, SCA and Secrets	2	Classes 4-6
3	Secure CI/CD with GitHub Actions and Jenkins	2	Classes 7-9
4	DAST, API Security and Vulnerability Remediation	2	Classes 10-12
5	Docker, Container Security and Software Supply Chain	2	Classes 13-15
6	Kubernetes Security and Runtime Fundamentals	2	Classes 16-18
7	Terraform, IaC Security and Cloud Security	2	Classes 19-21
8	Security Governance, Integrated Capstone and Career Readiness	2	Classes 22-24

The course contains 16 modules (2 per week) and 24 two-hour classes. The sequence moves from foundational security concepts to pipeline integration, workload security, infrastructure security and an end-to-end capstone.

WEEK 01 — DevSecOps Foundations and Security Fundamentals

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 1: DevSecOps Concepts

- What is DevSecOps and why security must be continuous
- Secure Software Development Lifecycle (SSDLC)
- Security roles, developer ownership and security champions
- DevOps vs DevSecOps; shift-left and shared responsibility
- People, process and technology in DevSecOps
- Overview of the software delivery lifecycle and common control points

Module 2: Security Fundamentals and Threat Modelling

- Confidentiality, Integrity and Availability (CIA)
- Vulnerability, threat, exploit, risk and impact
- Basic threat modelling: assets, trust boundaries, entry points and abuse cases
- Authentication, authorisation and least privilege
- OWASP Top 10 overview and common web application risks
- Security findings, severity and remediation lifecycle

DETAILED CLASS PLAN

Class 1: DevSecOps Overview and Secure Delivery Lifecycle

- DevOps and DevSecOps principles
- Where security fits into plan, code, build, test, release and deploy
- Security responsibilities and shift-left practices

✓ Hands-on lab: Map security checks to a sample application delivery lifecycle.

Class 2: Security Fundamentals and OWASP Risks

- CIA triad, identity and access concepts
- Common web risks such as injection, broken access control and insecure configuration
- Vulnerability lifecycle and risk-based prioritisation

✓ Hands-on lab: Review deliberately vulnerable training examples and identify risks.

Class 3: Threat Modelling, Git and Lab Setup

- Assets, trust boundaries, data flows and attack surfaces
- Threat scenarios and mitigation brainstorming
- Prepare Git repository, Linux environment and course lab tools

✓ Hands-on lab: Create a simple threat model for a sample web application.

WEEK 02 — Secure Source Code, SAST, SCA and Secrets

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 3: Secure Coding and Static Application Security Testing (SAST)

- Secure coding principles and input validation
- SAST purpose, strengths and limitations
- Understanding rules, findings, severity and false positives
- Common coding weaknesses: injection, insecure deserialisation and hard-coded credentials
- Static analysis with SonarQube and/or Semgrep
- Fixing findings and validating the correction

Module 4: Software Composition Analysis (SCA) and Secret Scanning

- Open-source dependencies and software supply-chain risk
- SCA and dependency vulnerability scanning with OWASP Dependency-Check or an equivalent
- Secret scanning with Gitleaks
- Direct vs transitive dependencies
- Lockfiles, dependency pinning and safe upgrades
- Credential rotation, revocation and prevention of future exposure

DETAILED CLASS PLAN

Class 4: Secure Coding and SAST Hands-on

- Input validation and secure coding examples
- SAST workflow and scanning a sample repository
- Review findings, prioritise and remediate a vulnerable code sample

✓ Hands-on lab: Run a static scan, inspect findings and commit a safe fix.

Class 5: Dependency and Open-Source Security

- Dependency trees, CVEs and vulnerable package versions
- Run dependency scanning and interpret reports
- Upgrade or replace a vulnerable dependency

✓ Hands-on lab: Scan a sample application and document dependency remediation.

Class 6: Secrets Management and Project 1

- Detect exposed keys and tokens with Gitleaks
- Remove secrets from code and understand history exposure
- Rotate/revoke exposed credentials; use environment variables or a secret store

✓ Hands-on lab: Project 1: Secure Repository Baseline - implement SAST, SCA and secret scanning, fix selected findings and produce a scan summary.

WEEK 03 — Secure CI/CD with GitHub Actions and Jenkins

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 5: GitHub Actions Pipeline Security

- Workflow structure, triggers, jobs, steps and runners
- Protecting workflow permissions and using least privilege
- Adding SAST, SCA and secret-scanning jobs
- Secure repository and branch settings
- Secure handling of secrets and environment variables
- Pull-request checks, status checks and security gates

Module 6: Jenkins Pipeline Security

- Jenkins architecture and pipeline-as-code basics
- Securing agents, plugins and build permissions
- Pipeline failure thresholds and exception handling
- Credentials store and safe credential binding
- Integrating scanning tools into Jenkins stages
- Comparing GitHub Actions and Jenkins security patterns

DETAILED CLASS PLAN

Class 7: GitHub Actions Pipeline Fundamentals and Hardening

- Workflow YAML, events, jobs and runners
- Repository permissions, protected branches and pull-request checks
- Secrets, token permissions and safe workflow patterns

✓ Hands-on lab: Create a pipeline with a secure checkout/build process and least-privilege permissions.

Class 8: Integrating Security Scans into GitHub Actions

- Add SAST, SCA and secret scanning stages
- Publish and interpret reports
- Set quality gates and fail a build for agreed critical findings

✓ Hands-on lab: Automate security checks on pull requests and demonstrate a blocked insecure change.

Class 9: Jenkins Secure Pipeline and Project 2

- Jenkinsfile, stages, agents and credentials binding
- Integrate scanners and secure pipeline configuration
- Compare pipeline controls and document remediation workflow

✓ Hands-on lab: Project 2: Secure CI/CD Pipeline - build a pipeline in GitHub Actions and a guided equivalent in Jenkins, with automated security checks and controlled release gates.

WEEK 04 — DAST, API Security and Vulnerability Remediation

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 7: Dynamic Application Security Testing (DAST)

- SAST vs DAST vs SCA: what each detects
- Safe use of OWASP ZAP against intentionally vulnerable lab applications
- Understanding alerts, evidence, confidence and false positives
- How DAST interacts with a running application
- Baseline and active scan concepts
- Retesting after remediation

Module 8: API Security and Vulnerability Management

- API basics: endpoints, methods, headers, authentication and authorisation
- Testing authentication, access control and input handling in a lab
- Risk prioritisation, remediation tickets and retesting
- Common API security risks and OWASP API Security Top 10 overview
- Security testing in pre-production environments
- Responsible testing boundaries and written authorisation

DETAILED CLASS PLAN

Class 10: DAST Fundamentals and OWASP ZAP

- DAST workflow and scan types
- Configure OWASP ZAP for a permitted test target
- Interpret alerts and understand scan limitations

✓ Hands-on lab: Run a baseline scan against a local training application.

Class 11: API Security Testing Fundamentals

- API endpoints, tokens, authentication and access control
- Common API misconfigurations and insecure object access concepts
- Use a safe lab API to inspect requests and responses

✓ Hands-on lab: Test an authorised demo API and record observations.

Class 12: Remediation, Retesting and Project 2 Extension

- Triage findings and separate confirmed issues from false positives
- Fix a selected application or API issue
- Retest and record before/after evidence

✓ Hands-on lab: Extend Project 2 with a DAST stage and a concise vulnerability remediation report.

WEEK 05 — Docker, Container Security and Software Supply Chain

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 9: Secure Docker and Container Images

- Container security risks and image layers
- Dockerfile security: non-root users, pinned versions and reduced packages
- Image misconfiguration and vulnerability remediation
- Trusted base images and minimal images
- Image scanning with Trivy
- Runtime configuration, read-only filesystems and resource limits

Module 10: Software Supply-Chain Security

- Build provenance and software supply-chain concepts
- Generate SBOMs using Syft or an equivalent tool
- Registry access, image tags and immutable digest references
- Software Bill of Materials (SBOM) and why it matters
- Image signing and verification concepts using Cosign
- Integrating image scanning and supply-chain checks into CI/CD

DETAILED CLASS PLAN

Class 13: Docker Security Fundamentals

- Container isolation and common risks
- Secure Dockerfile patterns and least-privilege runtime
- Image layers, base image selection and patching

✓ Hands-on lab: Harden a sample Dockerfile and rebuild the image.

Class 14: Container Scanning and SBOM

- Scan images with Trivy
- Interpret OS and application package findings
- Generate and inspect an SBOM with Syft

✓ Hands-on lab: Scan a sample image, fix selected findings and attach an SBOM.

Class 15: Image Integrity and Project 3 Start

- Image tags vs digests and registry permissions
- Image signing and verification concepts
- Add image security checks to a build workflow

✓ Hands-on lab: Project 3: Secure Container Build - produce a hardened image, scan it, generate an SBOM and demonstrate a verification/signing workflow where lab access permits.

WEEK 06 — Kubernetes Security and Runtime Fundamentals

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 11: Kubernetes Workload and Cluster Security

- Kubernetes architecture and security boundaries
- RBAC: roles, role bindings and least privilege
- Pod security context, non-root containers and capabilities
- Namespaces and workload isolation
- Service accounts and avoiding unnecessary privileges
- Secrets handling and configuration hygiene

Module 12: Kubernetes Network, Admission and Runtime Security

- NetworkPolicy concepts and traffic restriction
- Cluster configuration checks with kube-bench (guided lab)
- Runtime threat detection overview with Falco
- Image and manifest scanning with Trivy or Kubescape
- Admission control and policy concepts
- Logging, alert triage and secure deployment practices

DETAILED CLASS PLAN

Class 16: Kubernetes Security Architecture and RBAC

- Control plane, nodes, pods and service accounts
- RBAC roles, role bindings and least privilege
- Namespace boundaries and access review

✓ Hands-on lab: Create a restricted service account and validate permissions.

Class 17: Pod, Secrets and Network Security

- Security contexts, non-root users and Linux capabilities
- Secret handling and workload configuration
- NetworkPolicy concepts and traffic isolation

✓ Hands-on lab: Harden a Kubernetes deployment manifest and apply a basic network policy.

Class 18: Cluster Scanning, Runtime Concepts and Project 3 Completion

- Scan manifests and cluster configurations
- kube-bench/Kubescape guided checks
- Falco runtime detection concepts and response

✓ Hands-on lab: Complete Project 3 by deploying the secured container workload to a training cluster and documenting security checks.

WEEK 07 — Terraform, IaC Security and Cloud Security

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 13: Terraform and Infrastructure as Code Security

- IaC security principles and infrastructure drift
- Common cloud misconfigurations in IaC
- Secure storage, network and IAM configuration patterns
- Terraform files, providers, variables, outputs and state sensitivity
- Scan Terraform with Checkov or an equivalent tool
- Secrets in Terraform and safe state handling

Module 14: Cloud Security Fundamentals (One Provider Per Batch)

- Cloud shared responsibility model
- Secure networking and public exposure review
- Logging, monitoring and security posture checks
- IAM users, roles, policies and least privilege
- Storage access controls and encryption concepts
- Integrate IaC scanning and cloud configuration checks into CI/CD

DETAILED CLASS PLAN

Class 19: Terraform Security and Misconfiguration Scanning

- Terraform structure and sensitive values
- Common insecure resource configurations
- Scan Terraform with Checkov and interpret results

✓ Hands-on lab: Find and fix intentionally insecure Terraform examples.

Class 20: Cloud Security Fundamentals – Selected Provider

- AWS or Azure account and identity fundamentals
- IAM roles/policies and least privilege
- Secure networking, storage and logging fundamentals

✓ Hands-on lab: Complete a guided configuration review in the selected provider's sandbox or training environment.

Class 21: Policy Checks and Project 4 Build

- Add IaC scanning to a CI/CD workflow
- Security gates, approvals and exception records
- Review cloud configuration evidence

✓ Hands-on lab: Start Project 4: Secure Infrastructure Pipeline - scan Terraform, block selected misconfigurations and demonstrate safer cloud configuration in the chosen provider.

CareerByteCode Learning Management Portal

WEEK 08 **Governance, Integrated Capstone and Career Readiness**

2 MODULES I 3 CLASSES I 6 INSTRUCTOR-LED HOURS

Module 15: Security Governance, Monitoring and Continuous Improvement

- Security gates and risk-based release decisions
- Policy-as-code and compliance-as-code concepts
- Security exceptions, approvals and compensating controls
- Vulnerability SLAs, ownership and remediation tracking
- Audit evidence, scan reports and traceability
- Continuous improvement and metrics for DevSecOps

Module 16: Integrated Capstone and Career Preparation

- Connect source, dependencies, secrets, DAST, container and IaC checks
- Review findings, remediation and evidence
- Present a technical walkthrough to a business audience
- Secure release workflow and deployment verification
- Document architecture and security controls
- Resume, portfolio, interview and project storytelling

DETAILED CLASS PLAN

Class 22: Governance, Reporting and Security Metrics

- Security gates, severity and risk acceptance
- Vulnerability ageing, remediation SLAs and reporting
- Evidence, audit trail and policy-as-code overview

✓ Hands-on lab: Create a simple security dashboard/report from pipeline scan outputs.

Class 23: Integrated Capstone Build and Validation

- Connect pipeline security stages
- Run end-to-end scans and verify release controls
- Resolve priority findings and document exceptions

✓ Hands-on lab: Complete the integrated secure delivery workflow and prepare the project demonstration.

Class 24: Capstone Presentation and Interview Preparation

- Present architecture, pipeline and security decisions
- Explain findings, remediation, trade-offs and residual risk
- Portfolio and resume project descriptions
- Mock interview and feedback

✓ Hands-on lab: Demonstrate Project 4 and receive structured instructor feedback.

yteCode
ade simple

Four Major Hands-On Projects

CYBERFORGE DEVSECOPS

P1 **Secure Repository Baseline**

Schedule: Week 2

Secure a sample application repository by running SAST, SCA and secret scanning. Review findings, fix selected issues, document dependency updates and produce a short security summary.

Tools: Semgrep or SonarQube, OWASP Dependency-Check, Gitleaks, Git

P2 **Secure CI/CD Pipeline**

Schedule: Weeks 3-4

Build a GitHub Actions pipeline and a guided Jenkins equivalent. Integrate code, dependency, secret and DAST checks. Configure security gates and demonstrate how a risky change can be blocked before release.

Tools: GitHub Actions, Jenkins, Semgrep/SonarQube, Gitleaks, OWASP ZAP

P3 **Secure Container and Kubernetes Workload**

Schedule: Weeks 5-6

Harden a Docker image, scan it, generate an SBOM and deploy it to a training Kubernetes cluster with restricted RBAC, pod security settings and network controls. Document findings and mitigations.

Tools: Docker, Trivy, Syft, Cosign concepts, Kubernetes, Kubescape/kube-bench, Falco overview

P4 **Secure IaC and Cloud Delivery Capstone**

Schedule: Weeks 7-8

Create or review Terraform infrastructure, scan for misconfigurations, integrate checks into CI/CD and demonstrate safer IAM, network, storage and logging configurations in one selected cloud provider. Present the end-to-end security workflow.

Tools: Terraform, Checkov, GitHub Actions/Jenkins, AWS or Azure (one selected provider)

Tools and Technologies

CYBERFORGE DEVSECOPS

AREA	TOOLS / PLATFORMS
Source control and CI/CD	Git, GitHub, GitHub Actions, Jenkins
SAST	Semgrep; SonarQube Community or an available equivalent edition
SCA / dependency scanning	OWASP Dependency-Check; package-manager audit capabilities
Secrets scanning	Gitleaks
DAST / API testing	OWASP ZAP; safe demo API tooling
Container security	Docker, Trivy
SBOM and image integrity	Syft, Cosign (guided introduction)
Kubernetes security	Kubernetes, RBAC, NetworkPolicy, Kubescape, kube-bench, Falco (introductory/guided)
IaC security	Terraform, Checkov
Cloud security	AWS or Azure - one provider selected per batch
Reporting and collaboration	GitHub Issues/Projects or equivalent, Markdown, scan reports

Tool versions, editions and feature availability can change. CareerByteCode should confirm the selected versions and licensing before each batch. Where a commercial feature is unavailable in a free/community edition, the instructor should use an equivalent accessible lab or clearly label the feature as a demonstration.

Assessment and Evaluation

CYBERFORGE DEVSECOPS

ASSESSMENT	WEIGHT	EVALUATION FOCUS
Class labs and practical exercises	20%	Hands-on completion, safe tool usage and ability to explain results.

ASSESSMENT	WEIGHT	EVALUATION FOCUS
Weekly quizzes and assignments	15%	Concept understanding, pipeline configuration and security terminology.
Projects 1-3	30%	Correct scanning, meaningful remediation, secure configuration and clear evidence.
Final Project 4 capstone	25%	End-to-end workflow, security gates, infrastructure controls, validation and presentation.
Participation and technical communication	10%	Collaboration, troubleshooting, documentation and explanation of decisions.

Suggested completion requirements: attend required number of sessions, complete all four projects and meet the published passing threshold. CareerByteCode should define the exact attendance and pass criteria before enrolment.

Training Methodology and Delivery

CYBERFORGE DEVSECOPS

- Hybrid delivery with consistent learning outcomes, lab instructions and assignment expectations for classroom and online learners.
- Recommended two-hour class structure: 20-30 minutes concepts, 35-45 minutes instructor demonstration, 35-45 minutes guided hands-on practice and 5-10 minutes recap.
- Use deliberately vulnerable applications, local containers and isolated training environments for security exercises.
- Provide lab guides, sample repositories, pipeline templates, test datasets/applications, remediation notes and project rubrics.
- Learners should use only systems they own or have explicit permission to test. DAST, API testing and vulnerability exercises must remain within authorised lab targets.
- Cloud labs should use sandbox accounts, least-privilege permissions, budget limits and cleanup instructions. One cloud provider (AWS or Azure) will be selected for each batch.
- Provide a defined doubt-clearing process and catch-up materials where operationally possible.

Career Support and Portfolio

CYBERFORGE DEVSECOPS

- Course completion certificate subject to published attendance and assessment criteria.
- Resume guidance focused on actual tools, pipelines and project contributions completed during training.
- Portfolio containing repository links, pipeline YAML/Jenkinsfiles, scan summaries, remediation evidence and architecture diagrams.
- Mock interviews covering DevSecOps concepts, CI/CD security, SAST/DAST/SCA, containers, Kubernetes, Terraform and cloud fundamentals.
- Interview preparation with troubleshooting scenarios and project walkthrough practice.
- Guidance on describing findings and remediation honestly, without claiming production experience that the learner has not gained.

Commercial Details

CYBERFORGE DEVSECOPS

Course fee: ₹40,000 + applicable GST per learner.

Duration: 8 weeks | 24 classes | 48 instructor-led hours.

Delivery: Hybrid: classroom + online.

Included: Structured classes, guided labs, four project briefs, assessments, certificate subject to completion criteria, resume guidance, portfolio support, mock interviews and interview preparation.

Important Academic and Lab Safety Notes

CYBERFORGE DEVSECOPS

- The 48 instructor-led hours are intensive. Learners should budget additional self-study and lab practice between classes.
- All scanning and testing must be conducted against local, intentionally vulnerable or explicitly authorised systems. No scanning of public targets without permission.
- Never commit real credentials, private keys, tokens or production secrets to class repositories. Use disposable lab credentials and rotate any accidentally exposed secrets.
- Use isolated cloud accounts or sandbox environments with budget alerts and teardown steps. Avoid deploying intentionally vulnerable workloads to public production environments.
- The selected cloud provider (AWS or Azure) must be announced before the batch; the course fee does not imply full certification preparation or deep training on both providers.
- Advanced topics such as advanced threat hunting, full penetration testing, service mesh security, advanced policy-as-code, enterprise SIEM engineering and production compliance implementation are outside the core eight-week scope.

CareerByteCode
Learning Made simple

CareerByteCode
Learning Made simple

THANK YOU

Join the next batch, learn from practitioners, and leave with projects you can show and explain.

CareerByteCode Admissions · support@careerbytecode.in · www.careerbytecode.in/courses/cyberforge-devsecops/

CareerByteCode is a brand of Cloudnloud Technologies Pvt Ltd, Chennai. CareerByteCode Team.