DevOpsSchool

Course curriculum · November 2026

DEVOPS · TRAINING PROGRAM

Dot Net Devops

A practical, hands-on Dot Net Devops curriculum: 539 topics. Learn live online in a small batch, as a private corporate batch, or self-paced with one year of LMS access.

LIVE ONLINE **Max 8-10 hours**	CORPORATE **Private team batch, on demand**	SELF-PACED VIDEO **1-year LMS access**
BATCH SIZE **Max 10 learners**	NEXT BATCHES **1st of every month · weekends: first Saturday**	CERTIFICATE **Verifiable certificate of completion**

INSIDE THIS CURRICULUM

Who should attend	Objectives, audience, prerequisites
Agenda	539 topics
Formats and fees	Live, 1:1, self-paced, corporate
Trainer and reviews	Rajesh Kumar
Enrolment	Batches, payment, contacts

WHY DEVOPSSCHOOL

- Rated 4.6 on Google
- 1,500+ verifiable certificates issued in the public registry
- Trainer Rajesh Kumar: 20 years in industry
- Small batches: max 10 learners, every session recorded

Course page: www.devopsschool.com/courses/devops/devops-training-course-dot-net.html
Talk to us: WhatsApp +91 99057 40781 · +1 (469) 756-6329 · contact@DevOpsSchool.com · reply within 1 hour

COURSE FIT

Who this program is for

Use this page to decide whether Dot Net Devops matches your role and experience.

1

Objectives

Understand the core concepts and usage model, apply practical workflows through examples and guided tasks, and prepare for implementation, certification or team adoption.

2

Who should attend

Developers, DevOps engineers, administrators, SREs, security engineers, platform teams, release teams, technical leads and corporate teams working with this topic.

3

Before you start

Basic computer literacy and awareness of software, operations, cloud, DevOps or security workflows is helpful. Deep programming experience is not needed unless the topic requires it.

AGENDA AT A GLANCE · 539 TOPICS

Core curriculum

539 topics · Tools Covered

AGENDA · 539 TOPICS

Dot Net Devops agenda

Live online batches run for max 8-10 hours; corporate batches are planned with your team.

CORE CURRICULUM 539 topics

1. Tools Covered
2. Instructor-led, Live & Interactive Sessions
3. PRICE
4. Let’s Understand about Software Development Model
5. Overview of Waterfall Development Model
6. Challenges of Waterfall Development Model
7. Overview of Agile Development Model
8. Challenges of Agile Development Model
9. Requirement of New Software Development Model
10. Understanding an existing Pain and Waste in Current Software Development Model
11. What is DevOps?
12. Transition in Software development model
13. Waterfall -> Agile -> CI/CD -> DevOps -> DevSecOps
14. Understand DevOps values and principles
15. Culture and organizational considerations
16. Communication and collaboration practices
17. Improve your effectiveness and productivity
18. DevOps Automation practices and technology considerations
19. DevOps Adoption considerations in an enterprise environment
20. Challenges, risks and critical success factors
21. What is DevSecOps?
22. Let’s Understand DevSecOps Practices and Toolsets.
23. What is SRE?
24. Let’s Understand SRE Practices and Toolsets.
25. List of Tools to become Full Stack Developer/QA/SRE/DevOps/DevSecOps
26. Microservices Fundamentals
27. Microservices Patterns
28. Choreographing Services
29. Presentation components
30. Business Logic
31. Database access logic
32. Application Integration
33. Modelling Microservices
34. Integrating multiple Microservices
35. Keeping it simple
36. Avoiding Breaking Changes

37. Choosing the right protocols
38. Sync & Async
39. Dealing with legacy systems
40. Testing
41. What and When to test
42. Preparing for deployment
43. Monitoring Microservice Performance
44. Tools used for Microservices Demo using container
45. Centos,Ubuntu,VirtualBox,Vagrant, Bash Scripting - Concept
46. Ubuntu
47. Installing CentOS7 and Ubuntu
48. Accessing Servers with SSH
49. Working at the Command Line
50. Reading Files
51. Using the vi Text Editor
52. Piping and Redirection
53. Archiving Files
54. Accessing Command Line Help
55. Understanding File Permissions
56. Accessing the Root Account
57. Using Screen and Script
58. Overview of Hypervisor
59. Introduction of VirtualBox
60. Install VirtualBox and Creating CentOS7 and Ubuntu Vms
61. Vagrant
62. Understanding Vagrant
63. Basic Vagrant Workflow
64. Advance Vagrant Workflow
65. Working with Vagrant VMs
66. The Vagrantfile
67. Installing Nginx
68. Provisioning
69. Networking
70. Sharing and Versioning Web Site Files
71. Vagrant Share
72. Vagrant Status
73. Sharing and Versioning Nginx Config Files
74. Configuring Synced Folders
75. Cloud Computing using AWS Foundation Services
76. Introduction of AWS
77. Understanding AWS infrastructure

78. Understanding AWS Free Tier
79. IAM: Understanding IAM Concepts
80. IAM: A Walkthrough IAM
81. IAM: Demo & Lab
82. Computing:EC2: Understanding EC2 Concepts
83. Computing:EC2: A Walkthrough EC2
84. Computing:EC2: Demo & Lab
85. Storage:EBS: Understanding EBS Concepts
86. Storage:EBS: A Walkthrough EBS
87. Storage:EBS: Demo & Lab
88. Storage:S3: Understanding S3 Concepts
89. Storage:S3: A Walkthrough S3
90. Storage:S3: Demo & Lab
91. Storage:EFS: Understanding EFS Concepts
92. Storage:EFS: A Walkthrough EFS
93. Storage:EFS: Demo & Lab
94. Database:RDS: Understanding RDS MySql Concepts
95. Database:RDS: A Walkthrough RDS MySql
96. Database:RDS: Demo & Lab
97. ELB: Elastic Load Balancer Concepts
98. ELB: Elastic Load Balancer Implementation
99. ELB: Elastic Load Balancer: Demo & Lab
100. Networking:VPC: Understanding VPC Concepts
101. Networking:VPC: Understanding VPC components
102. Networking:VPC: Demo & Lab
103. Container using Docker & Docker Compose
104. What is Containerization?
105. Why Containerization?
106. How Docker is good fit for Containerization?
107. How Docker works?
108. Docker Architecture
109. Docker Installations & Configurations
110. Docker Components
111. Docker Engine
112. Docker Image
113. Docker Containers
114. Docker Registry
115. Docker Basic Workflow
116. Managing Docker Containers
117. Creating our First Image
118. Understading Docker Images

119. Creating Images using Dockerfile
120. Managing Docker Images
121. Using Docker Hub registry
122. Docker Networking
123. Docker Volumes
124. Deepdive into Docker Images
125. Deepdive into Dockerfile
126. Deepdive into Docker Containers
127. Deepdive into Docker Networks
128. Deepdive into Docker Volumes
129. Deepdive into Docker Volume
130. Deepdive into Docker CPU and RAM allocations
131. Deepdive into Docker Config
132. Docker Compose Overview
133. Install & Configure Compose
134. Understanding Docker Compose Workflow
135. Understanding Docker Compose Services
136. Writing Docker Compose Yaml file
137. Using Docker Compose Commands
138. Coding using Python & Django
139. Planning - Discuss some of the Small Project Requirement which include
140. Login/Registertration with Some Students records CRUD operations.
141. Design a Method --> Classes -> Interface using Core Python
142. Fundamental of Core Python with Hello-world Program with Method --> Classes
143. Coding in Flask using HTMl - CSS - JS - MySql
144. Fundamental of Flask Tutorial of Hello-World APP
145. UT - 2 Sample unit Testing using Pythontest
146. Package a Python App
147. AT - 2 Sample unit Testing using Selenium
148. Technology Demonstration
149. Software Planning and Designing using JAVA
150. Core Python
151. Flask
152. mySql
153. pytest
154. Selenium
155. HTMl
156. CSS
157. Js.
158. Coding using Java & Springboot

159. Planning - Discuss some of the Small Project Requirement which include Login/Registertration with Some Students records CRUD operations.
160. Design a Method --> Classes -> Interface using Core Java
161. Fundamental of Core Java with Hello-world Program with Method --> Classes -> Interface
162. Coding in SpringBoot using HTMl - CSS - JS - MySql
163. Fundamental of SpringBoot Tutorial of Hello-World APP
164. UT - 2 Sample unit Testing using Junit - Use Gradle or Maven
165. Package - War file - Use Gradle or Maven
166. AT - 2 Sample unit Testing using Selenium - Use Gradle or Maven
167. Core Java
168. SpringBoot
169. Junit
170. Planning and Designing - Jira + Confluence
171. Overview of Jira
172. Use cases of Jira
173. Architecture of Jira
174. Installation and Configuraration of Jira in Linux
175. Installation and Configuraration of Jira in Windows
176. Jira Terminologies
177. Understanding Types of Jira Projects
178. Working with Projects
179. Working with Jira Issues
180. Adding Project Components and Versions
181. Use Subtasks to Better Manage and Structure Your Issues
182. Link Issues to Other Resources
183. Working in an Agile project
184. Working with Issues Types by Adding/Editing/Deleting
185. Working with Custom Fields by Adding/Editing/Deleting
186. Working with Screens by Adding/Editing/Deleting
187. Searching and Filtering Issues
188. Working with Workflow basic
189. Introduction of Jira Plugins and Addons.
190. Jira Integration with Github
191. Exploring Confluence benefits and resources
192. Configuring Confluence
193. Navigating the dashboard, spaces, and pages
194. Creating users and groups
195. Creating pages from templates and blueprints
196. Importing, updating, and removing content
197. Giving content feedback
198. Watching pages, spaces, and blogs

199. Managing tasks and notifications
200. Backing up and restoring a site
201. Admin tasks
202. Add/Edit/Delete new users
203. Adding group and setting permissions
204. Managing user permissions
205. Managing addons or plugins
206. Customizing confluence site
207. Installing Confluence
208. Evaluation options for Confluence
209. Supported platforms
210. Installing Confluence on Windows
211. Activating Confluence trial license
212. Finalizing Confluence Installation
213. Source Code Versioning - Git using Github & GitOps
214. Introduction of Git
215. Installing Git
216. Configuring Git
217. Git Concepts and Architecture
218. How Git works?
219. The Git workflow
220. Working with Files in Git
221. Adding files
222. Editing files
223. Viewing changes with diff
224. Viewing only staged changes
225. Deleting files
226. Moving and renaming files
227. Making Changes to Files
228. Undoing Changes
229. Reset
230. Revert
231. Amending commits
232. Ignoring Files
233. Branching and Merging using Git
234. Working with Conflict Resolution
235. Comparing commits, branches and workspace
236. Working with Remote Git repo using Github
237. Push - Pull - Fetch using Github
238. Tagging with Git
239. Code Analysis & Securing Code (SAST) - SonarQube + SonarQube OWASP + Nginx

240. What is SonarQube?
241. Benefits of SonarQube?
242. Alternative of SonarQube
243. Understanding Various License of SonarQube
244. Architecture of SonarQube
245. How SonarQube works?
246. Components of SonarQube
247. SonarQube runtime requirements
248. Installing and configuring SonarQube in Linux
249. Basic Workflow in SonarQube using Command line
250. Working with Issues in SonarQube
251. Working with Rules in SonarQube
252. Working with Quality Profiles in SonarQube
253. Working with Quality Gates in SonarQube
254. Deep Dive into SonarQube Dashboard
255. Understanding Seven Axis of SonarQube Quality
256. Workflow in SonarQube with Maven Project
257. Workflow in SonarQube with Gradle Project
258. OWASP Top 10 with SonarQube
259. Build Management - Maven + Gradle
260. Maven
261. Introduction to Apache Maven
262. Advantage of Apache Maven over other build tools
263. Understanding the Maven Lifecycle and Phase
264. Understanding the Maven Goals
265. Understanding the Maven Plugins
266. Understanding the Maven Repository
267. Understanding and Maven Release and Version
268. Prerequisite and Installing Apache Maven
269. Understanding and using Maven Archetypes
270. Understanding Pom.xml and Setting.xml
271. Playing with multiples Maven Goals
272. Introducing Maven Dependencies
273. Introducing Maven Properties
274. Introducing Maven Modules
275. Introducing Maven Profile
276. Introducing Maven Plugins
277. How can Maven benefit my development process?
278. How do I setup Maven?
279. How do I make my first Maven project?
280. How do I compile my application sources?

281. How do I compile my test sources and run my unit tests?
282. How do I create a JAR and install it in my local repository?
283. How do I use plugins?
284. How do I add resources to my JAR?
285. How do I filter resource files?
286. How do I use external dependencies?
287. How do I deploy my jar in my remote repository?
288. How do I create documentation?
289. How do I build other types of projects?
290. How do I build more than one project at once?
291. Gradle
292. What is Gradle?
293. Why Gradle?
294. Installing and Configuring Gradle
295. Build Java Project with Gradle
296. Build C++ Project with Gradle
297. Build Python Project with Gradle
298. Dependency Management in Gradle
299. Project Structure in Gradle
300. Gradle Tasks
301. Gradle Profile and Cloud
302. Gradle Properties
303. Gradle Plugins
304. Package Management - Packer + Artifactory + JFrog XRay
305. Artifactory
306. Artifactory Overview
307. Understanding a role of Artifactory in DevOps
308. System Requirements
309. Installing Artifactory in Linux
310. Using Artifactory
311. Getting Started
312. General Information
313. Artifactory Terminology
314. Artifactory Repository Types
315. Artifactory Authentication
316. Deploying Artifacts using Maven
317. Download Artifacts using Maven
318. Browsing Artifactory
319. Viewing Packages
320. Searching for Artifacts
321. Manipulating Artifacts

322. Packer
323. Introduction to Selenium
324. Components of Selenium
325. Selenium IDE
326. Selenium Web driver
327. Selenium Grid
328. Installing and Configuring Selenium
329. Working with Selenium IDE
330. Working With Selenium Web driver with Java Test Case Setup and Working with Selenium Grid
331. Setup and Working with Selenium Grid
332. Jmeter
333. What Is Performance Testing?
334. Introducing JMeter
335. Configuring a Test Plan
336. Recording a Test
337. Assertions in Jmeter
338. Configuring a Dynamic Test
339. Collecting and Analyzing Test Results
340. Http cookie Managerto capture sessions
341. Jmeter controllers for Modules wise Metrics
342. Timers in Jmeter
343. Regular Expression in Jmeter
344. Data Driven testing in Jmeter
345. Handling Dynamic Responses
346. Jmeter NON GUI mode
347. Stress and Load Testing using JMeter
348. Scalability and Security Testing
349. Integration of JMeter with Selenium
350. Configuration & Deployment Management - Ansible
351. Overflow of Configuration Management
352. Introduction of Ansible
353. Ansible Architecture
354. Let’s get startted with Ansible
355. Ansible Authentication & Authorization
356. Let’s start with Ansible Adhoc commands
357. Let’s write Ansible Inventory
358. Let’s write Ansible Playbook
359. Working with Popular Modules in Ansible
360. Deep Dive into Ansible Playbooks
361. Working with Ansible Variables
362. Working with Ansible Template

363. Working with Ansible Handlers
364. Roles in Ansible
365. Ansible Galaxy
366. Infrastructure Coding - Terraform
367. Deploying Your First Terraform Configuration
368. Introduction
369. What's the Scenario?
370. Terraform Components
371. Updating Your Configuration with More Resources
372. Terraform State and Update
373. Data Type and Security Groups
374. Configuring Resources After Creation
375. Terraform Provisioners
376. Terraform Syntax
377. Adding a New Provider to Your Configuration
378. Terraform Providers
379. Terraform Functions
380. Intro and Variable
381. Resource Creation
382. Deployment and Terraform Console
383. Updated Deployment and Terraform Commands
384. Container Orchestration - Kubernetes & Helm Introduction
385. Understanding the Need of Kubernetes
386. Understanding Kubernetes Architecture
387. Understanding Kubernetes Concepts
388. Kubernetes and Microservices
389. Understanding Kubernetes Masters and its Component
390. kube-apiserver
391. etcd
392. kube-scheduler
393. kube-controller-manager
394. Understanding Kubernetes Nodes and its Component
395. kubelet
396. kube-proxy
397. Container Runtime
398. Understanding Kubernetes Addons
399. DNS
400. Web UI (Dashboard)
401. Container Resource Monitoring
402. Cluster-level Logging
403. Understand Kubernetes Terminology

404. Kubernetes Pod Overview
405. Kubernetes Replication Controller Overview
406. Kubernetes Deployment Overview
407. Kubernetes Service Overview
408. Understanding Kubernetes running environment options
409. Working with first Pods
410. Working with first Replication Controller
411. Working with first Deployment
412. Working with first Services
413. Introducing Helm
414. Basic working with Helm
415. Continuous Integration - Jenkins
416. Lets understand Continuous Integration
417. What is Continuous Integration
418. Benefits of Continuous Integration
419. What is Continuous Delivery
420. What is Continuous Deployment
421. Continuous Integration Tools
422. Integrations
423. Infrastructure
424. Host Map
425. Events
426. Dashboards
427. Develop an Instrumentation Strategy
428. Design and implement logging
429. Design and implement telemetry
430. IIntegrate logging and monitoring solutions
431. Develop a Site Reliability Engineering (SRE) strategy
432. Develop an actionable alerting strategy
433. Design a failure prediction strategy
434. Design and implement a health check
435. Design an authentication and authorization strategy
436. Design a sensitive information management strategy
437. Develop security and compliance
438. Design governance enforcement mechanisms
439. Manage source control
440. Develop a modern source control strategy
441. Plan and implement branching strategies for the source code
442. Configure repositories
443. Integrate source control with tools
444. Facilitate communication and collaboration

445. Communicate deployment and release information with business stakeholders
446. Generate DevOps process documentation
447. Automate communication with team members
448. Define and implement continuous integration
449. Design build automation
450. Design a package management strategy
451. Design an application infrastructure management strategy
452. Implement a build strategy
453. Maintain build strategy
454. Design a process for standardizing builds across organization
455. Define and implement a continuous delivery and release management strategy
456. Develop deployment scripts and templates
457. Implement an orchestration automation solution
458. Plan the deployment environment strategy
459. Log Monitoring Tool 1 - Splunk
460. What Is Splunk?
461. Overview
462. Machine Data
463. Splunk Architecture
464. Careers in Splunk
465. Introduction and Overview of NewRelic
466. What is Application Performance Management?
467. Understanding a need of APM
468. Understanding transaction traces
469. What is Application Performance?
470. APM Benefits
471. APM Selection Criteria
472. Why NewRelic is best for APM?
473. What is NewRelic APM?
474. How does NewRelic APM work?
475. NewRelic Architecture
476. NewRelic Terminology
477. Installing and Configuring NewRelic APM Agents for Application
478. Register a Newrelic Trial account
479. Installing a JAVA Agent to Monitor your Java Application
480. Installing a PHP Agent to Monitor your PHP Application
481. Installing New Relic Agent for .NET Framework Application
482. Installing a Docker based Agent to Monitor your Docker based Application
483. Understanding of NewRelic Configration settings of newrelic.yml
484. Understanding of NewRelic Agent Configration settings
485. Working with NewRelic Dashboard

486. Understanding a transactions
487. Understanding Apdex and Calculating and Setting Apdex Threshold
488. Understanding Circuit break
489. Understanding Throughput
490. Newrelic default graphs
491. Understanding and Configuring Service Maps
492. Understanding and Configuring JVM
493. Understanding Error Analytics
494. Understanding Violations
495. Understanding and Configuring Deployments
496. Understanding and Configuring Thread Profiler
497. Depp Dive into Transaction Traces
498. Profiling with New Relic
499. Creating and managing Alerts
500. Working with Incidents
501. Sending NewRelic Alerts to Slack
502. Assessing the quality of application deployments
503. Monitoring using Newrelic
504. View your applications index
505. APM Overview page
506. New Relic APM data in Infrastructure
507. Transactions page
508. Databases and slow queries
509. Viewing slow query details
510. External services page
511. Agent-specific UI
512. Viewing the transaction map
513. Istio Introduction - Concept
514. Environment Setup (Kubernetes cluster and Istio(1.7.X/1.8.0)) -
515. Istio Architecture
516. Traffic Management
517. Security -
518. Telemetry, Observability
519. Ingress Gateway: HTTP
520. Ingress Gateway: HTTPS/TLS
521. Ingress Gateway: mTLS, ciphersuites, TLS version - Concept
522. VirtualService
523. DestinationRule
524. Fault Injection
525. Request Timeout
526. Traffic Mirroring

527. Gateway Routing
528. Load Balancing
529. Egress HTTP
530. Egress TLS, mTLS
531. Service Entry
532. AuthorizationPolicy
533. Request Authentication
534. Peer Authentication
535. Envoy Filters
536. External Authorization
537. Rate Limiting
538. Egress Mongo Setup
539. Debugging with istioctl

HOW YOU CAN LEARN

Formats and fees

The same agenda in four formats. Fees are in Indian rupees as of November 2026; the course page shows them in your currency.

FORMAT	DURATION	WHAT YOU GET	FEE
Live & interactive (online)	Max 8-10 hours	Instructor-led public batch of max 10 learners with demos, labs, Q&A and assignment review. Weekday batches start on the 1st of every month, weekend batches on the first Saturday of every month.	**INR 34,999**
1:1 live training (online)	Max 8-10 hours, one-to-one	The same agenda, taught one-to-one by the trainer. Starts immediately, without waiting for the next batch; sessions planned around your working hours, at your pace.	**INR 99,999**
Self-paced video	1-year LMS access	This program plus 20+ more courses and 50+ tools, at your own pace.	**INR 833/month** billed yearly at INR 9,996
Corporate	Planned with your team	A private batch for your team, on demand, with examples mapped to your platforms.	**Quote on request**

ASSESSMENT AND CERTIFICATE

Your progress is checked through hands-on tasks and assignments. On completion you receive a DevOpsSchool certificate of completion with a unique credential ID that anyone can verify at devopsschool.com/certificates.

INCLUDED WITH EVERY LIVE BATCH

- Every live session is recorded and stays in your LMS for a year.
- Can't attend your batch? Move to next month's batch.
- Free demo class before you pay.
- Free 10-15 minute call with an advisor or the trainer to check the course fits you.

YOUR TRAINER

Learn from Rajesh Kumar

Principal Architect, trainer and mentor in DevOps, SRE, DevSecOps and platform engineering, with 20 years across 8+ software MNCs, including Staff DevOps/SRE Engineer for a leading Japanese mobile-payment platform and Principal DevOps Architect and Manager in Europe.

Profile: www.rajeshkumar.xyz · Videos: youtube.com/@TheDevOpsSchool

20 years in industry	**200+** corporate trainings
10,000+ engineers mentored	**70+** organisations coached

REVIEWS

What learners say

Rated 4.6 on Google · more reviews at devopsschool.com/testimonials and in our video testimonials.

> “Rajesh is very good trainer. Rajesh was able to resolve our queries and question effectively. We really liked the hands-on examples covered during this training program.”
>
> Indrayani, India

> “The training was very useful and interactive. Rajesh helped develop the confidence of all.”
>
> Abhinav Gupta, Pune

START YOUR TRAINING

Join the next batch

Weekday batches start on the 1st of every month and weekend batches on the first Saturday of every month, with max 10 learners each. 1:1 training can start immediately; corporate batches run on demand. We reply within 1 hour.

Pay by UPI, credit and debit cards, net banking or PayPal. Before you pay, read our refund policy. All courses: www.devopsschool.com/courses/

ENROL OR RESERVE A SEAT
Dot Net Devops

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com

More DevOps training: www.devopsschool.com/courses/devops/