

**Introduction to Chaos Mesh**

**Course Number:** DVOP-178WA  
**Duration:** 1 day

**Overview**

This Chaos Mesh training course introduces learners to the principles of Chaos Engineering using Chaos Mesh, an open-source chaos testing platform for Kubernetes. Attendees gain practical skills to inject, monitor, and mitigate failures in distributed systems, ensuring system resilience and reliability.

**Prerequisites**

All learners should have taken [Intro to Docker or Kubernetes](https://www.exitcertified.com/it-training/programming/intro-docker-kub-67154-detail.html)or have equivalent knowledge.

**Materials**

All Chaos Mesh training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

**Objectives**

* Recognize the role of controlled failure injection in system resilience
* Learn the architecture and components of Chaos Mesh
* Install and configure Chaos Mesh in a Kubernetes environment
* Create and execute Chaos Experiments for common failure scenarios
* Analyze the impact of chaos experiments on system performance
* Use monitoring tools to observe system behavior during experiments
* Evaluate metrics and logs to identify potential system improvements
* Develop strategies to enhance resilience based on chaos experiments
* Incorporate Chaos Engineering into CI/CD pipelines for proactive failure testing

**Outline**

* Introduction to Chaos Engineering
  + What is Chaos Engineering?
  + Goals and benefits of controlled failure testing
  + Chaos Engineering vs. traditional testing
* Introduction to Chaos Mesh
  + Overview of Chaos Mesh and its architecture.
  + Installing Chaos Mesh in a Kubernetes environment.
* Running Chaos Experiments
  + Understanding Chaos Experiments and fault injection
  + Pod failures
  + Network delays or disconnections
  + CPU and memory stress tests
  + DNS disruptions
* Monitoring and Analyzing Results
  + Observing system behavior using tools like Prometheus and Grafana
  + Analyzing experiment results with Chaos Dashboard
* Improving Resilience and Automating Chaos
  + Strategies for addressing discovered weaknesses
  + Incorporating Chaos Mesh into CI/CD pipelines
* Conclusion