COURSE TABLE OF CONTENTS

Locust Performance Testing

OEM: Open Source • Duration: 3 Days (18 hrs)

COURSE MODULES & TOPICS

Module 1: Performance Testing Fundamentals

- Fundamentals of performance testing
- Importance in software applications
- Types: Load, Stress, Spike, Endurance, Scalability
- Performance testing lifecycle
- Key metrics: Response Time, Throughput, Latency, Error Rate
- Overview of performance testing tools
- Locust vs JMeter and others

Module 2: Getting Started with Locust

- Overview and features of Locust
- Locust architecture and principles
- Python fundamentals for Locust
- Locust installation and setup
- Installing Locust using pip
- Locust project structure
- Running the first Locust test
- Locust command-line interface

Module 3: Developing Locust Test Scripts

- Locust test script basics
- HttpUser class overview
- Defining tasks with @task decorator
- Task execution and weighting
- Configuring wait_time
- Managing user behavior
- Defining host and target application
- HTTP GET and POST requests
- Handling headers and parameters

Module 4: Locust Web Interface and Basic Execution

- Starting Locust Web UI
- Configuring concurrent users
- Defining user spawn rate
- Starting and stopping load tests
- Monitoring real-time statistics
- Understanding request statistics and response times
- Downloading test results

Module 5: Advanced User Behavior and Task Management

- Creating multiple user classes

- TaskSet and task organization
- Task weighting and execution sequence
- Managing user wait times
- Sequential task execution
- User lifecycle methods: on_start(), on_stop()
- Custom user behavior simulation
- Managing user sessions and cookies

Module 6: API Performance Testing with Locust

- REST API performance testing introduction
- Testing GET, POST, PUT, DELETE endpoints
- Sending JSON payloads
- Authentication and authorization handling
- Token-based authentication
- Managing request headers
- Parameterized API testing
- Response validation and assertions
- Handling HTTP errors and exceptions
- Measuring API response performance

Module 7: Advanced Load Testing Scenarios

- Simulating real-world user journeys
- Configuring user load patterns
- Gradual ramp-up and ramp-down
- Constant and variable load testing
- Stress and spike testing
- Endurance and soak testing
- Testing multiple API endpoints
- Managing distributed user workloads

Module 8: Test Data Management and Customization

- Managing external test data
- Reading data from CSV and JSON files
- Dynamic test data generation
- Parameterizing test execution
- Custom request naming and grouping
- Validating responses during execution
- Managing test data for concurrent users

Module 9: Distributed Load Testing with Locust

- Introduction to distributed load testing
- Locust Master and Worker architecture
- Understanding distributed execution
- Configuring master and worker nodes
- Running tests across multiple machines
- Scaling load generation
- Managing worker connections
- Executing large-scale load tests

Module 10: Monitoring and Performance Analysis

- Locust performance statistics overview
- Analyzing response time and latency
- Requests per second (RPS)
- Throughput and failure rate analysis
- Identifying performance bottlenecks
- Monitoring CPU and memory utilization
- Integrating system monitoring tools
- Analyzing application performance under load

Module 11: Reporting and Test Result Analysis

- Locust real-time statistics
- Generating CSV reports
- Exporting request statistics
- Analyzing response time percentiles
- Understanding failures and exceptions
- Comparing test runs
- Creating summary reports
- Identifying scalability issues

Module 12: CI/CD Integration and Best Practices

- Running Locust via command-line
- Headless test execution
- Integrating Locust with Jenkins
- Automating tests in CI/CD pipelines
- Managing test configurations
- Performance testing best practices
- Designing reusable Locust scripts
- Common troubleshooting techniques

Module 13: Capstone Project: End-to-End Performance Testing

- Identify performance testing requirements
- Develop Locust scripts for REST APIs
- Configure concurrent virtual users
- Execute load, stress, and endurance tests
- Perform distributed load testing
- Monitor application performance
- Analyze test results and bottlenecks
- Prepare final performance testing report