

## Red Hat Training: DevOps Culture and Practice Enablement (TL500)

**Duration: 5 Days**    **Course Code: TL500**

### Overview:

**Experience the possibilities of DevOps through proven open culture and practices used by Red Hat to support customer innovation**

Red Hat Training: DevOps Culture and Practice Enablement (TL500) is a five-day, immersive class offering students an opportunity to experience and implement cultural shifts that are utilized in many successful DevOps adoption journeys.

Many agile training offerings focus on a particular framework, delivery mechanism, or technology.

Instead, DevOps Culture & Practice combines the best tools from many leading frameworks to blend continuous discovery and continuous delivery with cultural and technical practices into a unique, highly-engaging experience simulating real-world scenarios and applications.

To achieve the learning objectives, participants should include multiple roles from an organization. Business product owners, architects, developers, and site reliability engineers will gain the experience of working outside of their traditional silos.

The daily routine simulates a real-world delivery team, where cross-functional teams learn how collaboration breeds innovation. Armed with shared experiences and best practices, the team can apply what it has learned to help the organization's culture and mission succeed in the pursuit of new projects and improved processes.

You will know how to deliver continuous value to customers through an enhanced understanding of changing customer needs and the ability to more rapidly get new ideas to market. Our instructors will share experiences and best practices learned from engaging directly with customers during Red Hat services engagements.

How is this course different from any other agile or DevOps technical training? Most agile training focuses on a particular framework, or only on delivery or technology, but this course combines the best tools from many leading frameworks. This course blends continuous discovery and continuous delivery with cultural and technical practices into a unique, highly-engaging experience, packed with real-world applications. You will learn both the individual practices and how each builds upon the next to improve alignment and delivery efficiency. Some DevOps technical training focuses solely on implementation of technology to perform specific tasks, not on how that technology integrates and enables delivery of the many features and bug fixes that customers need. This class encourages a systems thinking approach to delivery.

This course includes a copy of the Red Hat Training: Open Practices for your DevOps Journey course book, which can be used as a resource as students take the learnings from this course and apply them to other real world scenarios.

This course is based on Red Hat OpenShift Container Platform, Red Hat® Ansible Engine, and Red Hat® Enterprise Linux®.

**Note:** Starting January 2026 this course only exists in CR (classroom) if scheduled or Closed course modalities. No RHLS-Course for this course.

Updated Jan2026

### Company Events

These events can be delivered exclusively for your company at our locations or yours, specifically for your delegates and your needs. The Company Events can be tailored or standard course deliveries.

### Target Audience:

This experience demonstrates how individuals across different roles must learn to share, collaborate, and work toward a common goal to achieve positive outcomes and drive innovation. Business product owners, developers, site reliability engineers, and engineers will especially benefit from experiencing this scenario, which includes some technical aspects that rely on working with software systems.

### Objectives:

- After this course participants should be able to:
- Understand and implement multiple open practices
- Deploy a small multi-tiered application to an OpenShift cluster
- Work as an effective member of an agile team
- Discover, prioritize, and document desired software features and functionality
- Gain and apply experience with more than 30 practices, including:
- Use of cultural tools such as social contracts, real-time retrospectives, and team sentiments to align teams.
- Manage powerful planning practices such as impact mapping, event storming, and metrics-based process mapping.
- Discover iterative and incremental delivery practices, including scrum and kanban.

- Develop software using pair and mob programming styles

- Use technical programming practices, including continuous integration, continuous delivery, infrastructure as code, test automation, and test-driven development.

- Discover other open practices that are used in Open Innovation Labs

---

## Prerequisites:

- Knowledge of agile practices is helpful
  - Experience using agile practices and methodologies such as scrum is beneficial
- Take Red Hat free assessment to gauge whether this offering is the best fit for your skills [Red Hat Skills Assessment](#)

## Testing and Certification

- 

---

## Follow-on-Courses:

- Container Adoption Journey Bootcamp (DO700)
  - Container Adoption Boot Camp for Developers (DO720)
  - Red Hat OpenShift skill paths
  - Open Innovation Labs residency (contact Red Hat)
-

## Content:

|                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| What is DevOps?                                                                                                                                                                                                                                                                | Prioritization and pivoting                                                                                                                                                                                          | Continuous integration, deployment, and delivery                                                                                                                                                                                                            |
| Brainstorm and explore what principles, practices, and cultural elements make up a DevOps model for software design and development.                                                                                                                                           | Experience the collection of ideas, aligning them to target outcomes, and using economic prioritization practices and value slicing to build product backlogs that can deliver incremental value.                    | Explore the foundational practices of continuous integration, continuous deployment, and continuous delivery.                                                                                                                                               |
| Collaborative practices to establish culture and shared understanding                                                                                                                                                                                                          | Agile practices                                                                                                                                                                                                      | Non-functional requirements                                                                                                                                                                                                                                 |
| Learn and experience practices that facilitate great conversation and alignment across stakeholder groups such as priority sliders, pair programming, mob programming, conducting retrospectives, visualizing work, assessing team sentiment, and performing agile estimation. | Cover agile delivery practices, including Kanban, Scrum, sprint planning, daily standup, showcase, retrospective, and backlog refinement.                                                                            | Learn how to elaborate non-functional areas that are unlikely to be captured by using practices primarily focused on the functional aspects of a solution.                                                                                                  |
| Understanding the Why and Who of software delivery                                                                                                                                                                                                                             | Design of experiments                                                                                                                                                                                                | Testing                                                                                                                                                                                                                                                     |
| Use the impact mapping discovery practice to connect deliverables to measurable impact. Learn how to use human-centered design, design thinking, and Lean UX to develop empathy with users and stakeholders.                                                                   | Set up, execute, and measure the results of experiments by utilizing platform's advanced deployment features, including A/B Testing, Blue/Green Deployments, Feature Toggles, Dark Launches, and Canary Deployments. | Develop an understanding of test-driven development and business-driven development foundational practices, often called automated testing.                                                                                                                 |
| Domain-driven design and storytelling                                                                                                                                                                                                                                          | Value stream and process mapping                                                                                                                                                                                     | Everything as code and GitOps                                                                                                                                                                                                                               |
| Learn and practice the powerful Event Storming tool to visualize and map event-driven systems to produce emergent architectures for iterative and incremental delivery.                                                                                                        | Delve into the practices of value stream mapping and metric-based process mapping to establish non-functional improvements that you can make to product delivery and execution of value streams.                     | Explore continuous integration/continuous delivery pipelines using Jenkins and Tekton and sing a GitOps approach to codify everything for repeatability. Experience how to extend pipelines to cover non-functional testing, monitoring, and observability. |

---

## Additional Information:

Official course book provided to participants

---

## Further Information:

For More information, or to book your course, please call us on Head Office Tel.: +974 40316639

[training@globalknowledge.qa](mailto:training@globalknowledge.qa)

[www.globalknowledge.com/en-qa/](http://www.globalknowledge.com/en-qa/)

Global Knowledge, Qatar Financial Center, Burj Doha, Level 21, P.O.Box 27110, West Bay, Doha, Qatar