

- Learn practical DevOps techniques using Docker
- Understand what Containers are and why they are critical to DevOps Implementation
- Understand the management and orchestration of containers
- Learn how to network containers and share files and data between containers
- Discover how to secure containers

As organizations look to improve the speed with which they deliver software, they increasingly turn to microservices and infrastructure-as-code software architecture and delivery techniques to help leverage value from their DevOps adoptions. Docker, an industry standard containerization tool, facilitates moving processes into isolated environments that can be frozen into images, with an ecosystem that helps developers across organizations build and share these containers. This allows a software team to break their application down into small, platform-agnostic components that can be stitched together to provide the complete application capability. Containers are easily shared through a Continuous Integration/Continuous Delivery (CI/CD) pipeline and reduce provisioning, deployment, and startup times, driving run and test cycle times down. This allows software teams to use testing time more efficiently and minimizes the difference between environments.

This course is an extension to our [Foundations of DevOps—ICAgile Certification](#) [1] course and will teach you practical DevOps techniques using Docker, a DevOps tool of choice. Upon completion of the course, students will understand and have hands-on experience with critical DevOps concepts and techniques including:

- Containers - what they are and why they are critical to DevOps implementation
- Management and orchestration of Containers
- Networking Containers and sharing files and data between Containers
- Securing Containers

In addition, we'll talk about critical Docker best practices that will ensure success and avoid pitfalls:

- Container rules and service design
- Managing and limiting volumes
- Managing images to minimize disk space usage
- Reducing Build Time by minimizing steps in Dockerfiles
- Improving Container Security
- Critical Container metrics

## Hands-on Exercises

In this one-day hands-on workshop, students will construct their own containers and deploy a web application into those containers using a provided virtual machine image and will also learn to:

- Construct and run containers
- Share files and data between running containers and the host machine
- Network containers securely
- Share containers across teams and the overall organization
- Analyze and inspect running container health
- Use Kubernetes to simplify management and orchestration of containers for your application

## Laptop Required

This is a hands-on course. With their laptops, participants will construct, experiment with, and orchestrate their own containers gaining valuable experience on the hows and whys of containers as well as potential implementation pitfalls. Participants will be provided a virtual machine image to work within, minimizing prep time outside of class. Participants will need to install Oracle VirtualBox and download the (large) image file prior to the start of class.

## Who Should Attend

This course is especially appropriate for both Developers and Operations Engineers. Developers will gain a way to control and manage the dependencies of their application and drastically shorten the code-build-test cycle. Operations Engineers will learn a new way to deploy and orchestrate applications with their contexts, minimizing differences in environments from local development through production. Basic familiarity with the Linux command line interface is assumed.

## Course Outline

---

### Introduction to Containers

- What are Containers?
- Running an Application in Docker
- Reaching through Container Boundaries

### Building Containers

- Writing a Dockerfile for your Application
- Sharing Resources between Containers
- Accessing Containerized Services
- The Single Process Rule
- Designing Secure Containers

### Docker-in-Depth

- Understanding, Managing, and Securing Images
- Running, Configuring, and Operating Containers
- Volume Quirks
- Networking Containers Together
- Sharing Images with your Team

### Container Orchestration and Operation with Kubernetes

- What is Kubernetes?
- Pods, Services, Deployments, and Controllers
- Monitoring, the Kube Dashboard, and Metrics
- Tying in Configuration Management