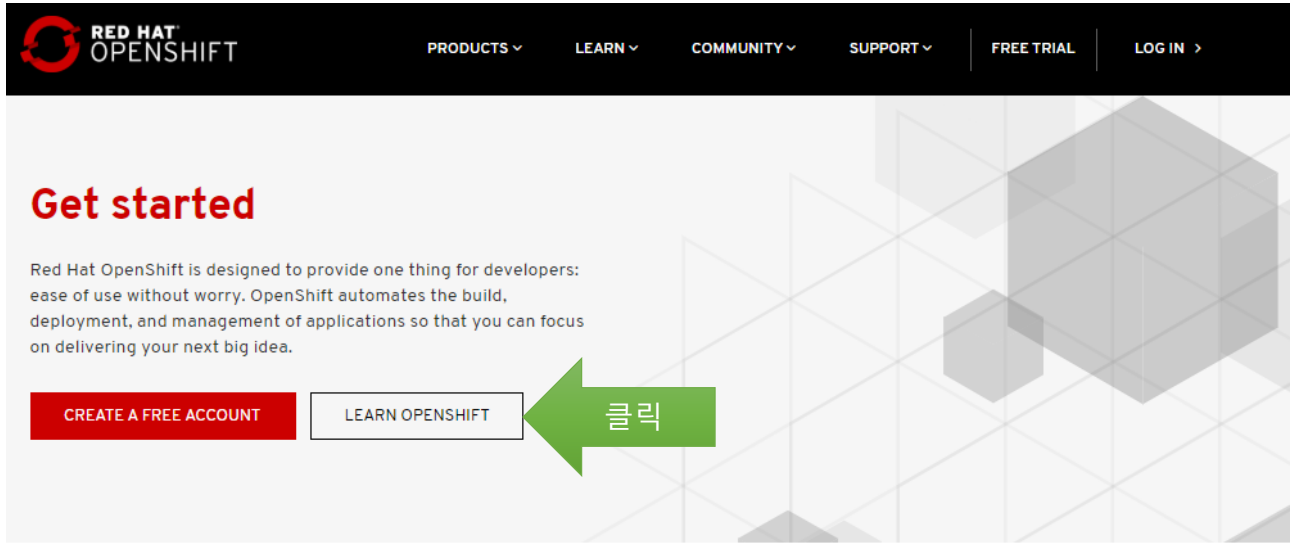
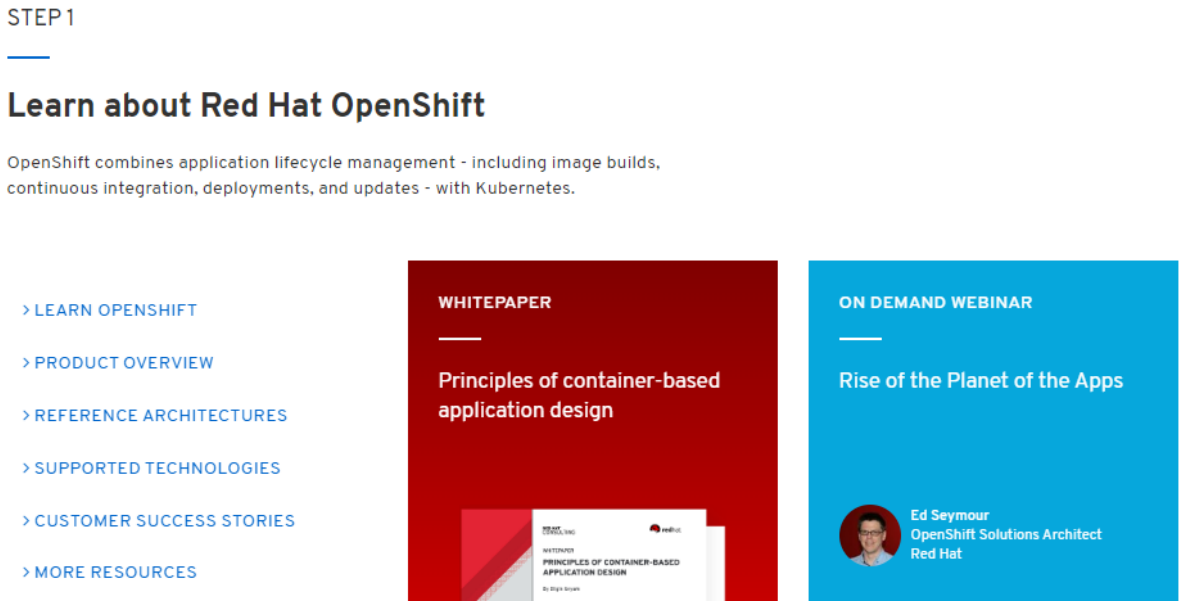
4장. 고가용성과 신뢰성

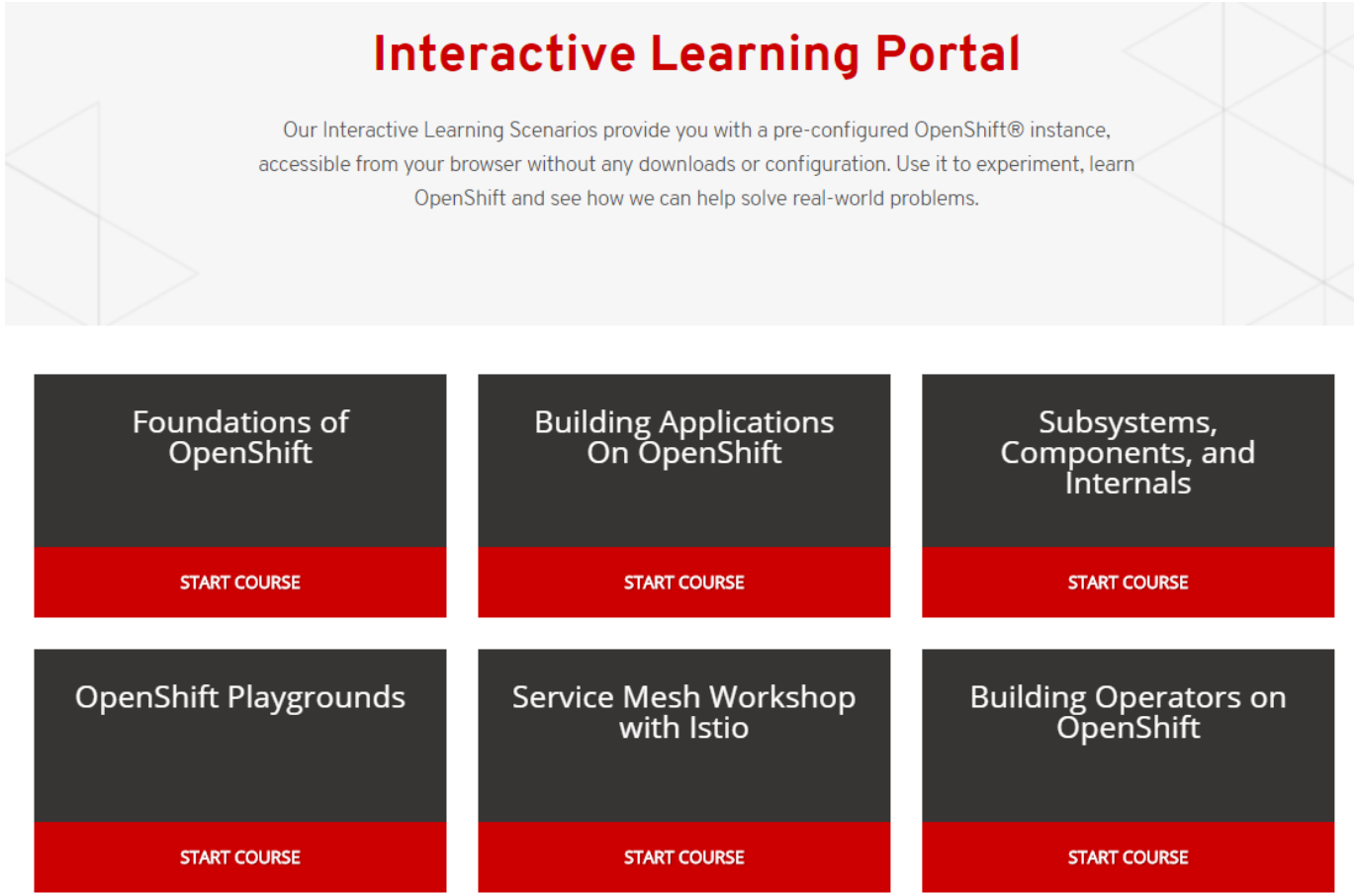
Mastering kubernetes

https://www.openshift.com/learn/get-started/





실습 환경


Interactive Learning Portal
Our Interactive Learning Scenarios provide you with a pre-configured OpenShift® instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.
Foundations of OpenShift
START COURSE
Building Applications On OpenShift
START COURSE
Subsystems, Components, and Internals
START COURSE
OpenShift Playgrounds
START COURSE
Service Mesh Workshop with Istio
START COURSE
Building Operators on OpenShift
START COURSE

학습 내용

- **Foundations of OpenShift**
  - Getting Started with OpenShift for Developers
  - Logging in to an OpenShift Cluster
  - Developing with odo
  - Deploying Applications From Images
  - Deploying Applications From Source
  - Using the CLI to Manage Resource Objects
  - Connecting to a Database Using Port Forwarding
  - Transferring Files in and out of Containers
  - Exploring and using metrics and HPAs
  - Introduction to Federation V2
- **Building Applications On OpenShift**
  - Developing with Quarkus
  - Eclipse Vert.x development
  - Spring and Spring Boot development
  - Java EE 8 Development
  - Application Messaging with Openshift
  - Thorntail development
  - Red Hat Data Grid development
  - Node.js development
  - JBoss BRMS Loan Application demo
  - Red Hat Decision Manager Loan Application demo
  - Red Hat Decision Manager DMN Introduction
  - Java EE Batch Processing with OpenShift, WildFly & JBeret
  - Debezium deployment
  - Hello ! Fuse - Getting Started
- **Subsystems, Components, and Internals**
  - Linux Container Internals 2.0 - Lab 1: Introduction to Containers
  - Linux Container Internals 2.0 - Lab 2: Container Images
  - Linux Container Internals 2.0 - Lab 3: Container Registries
  - Linux Container Internals 2.0 - Lab 4: Container Hosts
  - Linux Container Internals 2.0 - Lab 5: Container Orchestration
  - Linux Container Internals 2.0 - Lab 6: Container Standards
  - Linux Container Internals 2.0 - Lab 7: Container Tools Ecosystem
- **OpenShift Playgrounds**
  - OpenShift 3.6 Playground
  - OpenShift 3.9 Playground
  - OpenShift 3.11 Playground
- **Service Mesh Workshop with Istio**
  - Istio 1.0.x workshop: Istio Introduction
  - Istio 1.0.x workshop: Deploy microservices
  - Istio 1.0.x workshop: Monitoring and Tracing
  - Istio 1.0.x workshop: Simple Routing
  - Istio 1.0.x workshop: Advanced RouteRules
  - Istio 1.0.x workshop: Fault Injection
  - Istio 1.0.x workshop: Circuit Breaker
  - Istio 1.0.x workshop: Egress
  - Istio 1.0.x Advanced: Observing with Kiali
  - Istio 1.0.x Advanced: Mutual TLS
- **Building Operators on OpenShift**
  - Kubernetes API Fundamentals
  - Etcd Operator
  - Operator SDK with Go (PodSet)
  - Operator Lifecycle Manager
  - Ansible Refresher
  - Ansible Kubernetes Modules
  - Ansible Operator Overview
  - Operator SDK with Helm

Welcome!

openshift - Getting Started with OpenShift for Developers

★ Difficulty: beginner

Estimated Time: 20-30 minutes

In this self paced tutorial you will learn how to use the OpenShift Container Platform to build and deploy applications using both containers and orchestration.

Let's get started

If you are not familiar with the OpenShift Container Platform, it's worth taking a few minutes to understand the basics of the platform as well as the environment that you will be using for this self paced tutorial.

The goal of OpenShift is to provide a great experience for both Developers and System Administrators to develop, deploy, and run containerized applications. Developers should love using OpenShift because it enables them to take advantage of both containerized applications and orchestration without having to the know the details. Developers are free to focus on their code instead of spending time writing Dockerfiles and running docker builds.

OpenShift is a full platform that incorporates several upstream projects while also providing additional features and functionality to make those upstream projects easier to consume. The core of the platform is containers and orchestration. For the container side of the house, the platform uses images based upon the docker image format. For the orchestration side, we have put a lot of work into the upstream Kubernetes project. Beyond these two upstream projects, we have created a set of additional Kubernetes objects such as routes and deployment configs that we will learn how to use during this course.

Both Developers and Operators communicate with the OpenShift Platform via one of the following methods:

Command Line Interface

The command line tool that we will be using as part of this training is called the *oc* tool. This tool is written in the Go programming language and is a single executable that is provided for Windows, OS X, and the Linux Operating Systems.

Web Console

Getting Started with OpenShift for Developers

Step 1 of 6

Step 1 - Exploring The Command Line

The OpenShift CLI is accessed using the command *oc*. From here, you can administrate the entire OpenShift cluster and deploy new applications.

The CLI exposes the underlying Kubernetes orchestration system with the enhancements made by OpenShift. Users familiar with Kubernetes will be able to adapt to OpenShift quickly. The CLI is ideal in situations where you are:

1) Working directly with project source code.

2) Scripting OpenShift operations.

3) Restricted by bandwidth resources and cannot use the web console.

For this section, our task is going to be creating our first project.

What is a project? Why does it matter?

The goal of this scenario is to get a project created and running, which you'll be doing with the **web console** in the next section.

OpenShift is often referred to as a container application platform in that it is a platform designed for the development and deployment of containers.

To contain your application, we use projects. The reason for having a project to contain your application is to allow for controlled access and quotas for developers or teams.

More technically, it's a visualization of the Kubernetes namespace based on the developer access controls.

Terminal | Dashboard

```
Your Interactive Learning Environment Bash Terminal

$ ~/.launch.sh
Starting OpenShift
Waiting for OpenShift to start... This may take a couple of moments
OpenShift started.
Configuring... OpenShift Ready
$
```

Getting Started with OpenShift for Developers

Step 1 of 6

Step 1 - Exploring The Command Line

The OpenShift CLI is accessed using the command *oc*. From here, you can administrate the entire OpenShift cluster and deploy new applications.

The CLI exposes the underlying Kubernetes orchestration system with the enhancements made by OpenShift. Users familiar with Kubernetes will be able to adapt to OpenShift quickly. The CLI is ideal in situations where you are:

1) Working directly with project source code.

2) Scripting OpenShift operations.

3) Restricted by bandwidth resources and cannot use the web console.

For this section, our task is going to be creating our first project.

What is a project? Why does it matter?

The goal of this scenario is to get a project created and running, which you'll be doing with the **web console** in the next section.

OpenShift is often referred to as a container application platform in that it is a platform designed for the development and deployment of containers.

To contain your application, we use projects. The reason for having a project to contain your application is to allow for controlled access and quotas for developers or teams.

More technically, it's a visualization of the Kubernetes namespace based on the developer access controls.