

## Day - 1

- **Operator Introduction**

- Kubernetes application basics
- Kubernetes controllers and Custom Resource Definitions (CRD)
- Operator architecture
- CoreOS Operator framework
- Operator pattern
- Working with operatorhub.io
- Deploying Operators
- Using an Operator

- **Automating Stateful Applications with Kubernetes Operators Length**

- Kubernetes automation primitives and orchestration of stateless applications
- The Problem: Most apps have state. It would be nice to automate their scaling, recovery, with common methods
- Operators manage the lifecycle of complex, stateful applications by extending the Kubernetes API to store, and the Kubernetes control plane to manage, internal, application-specific state
- Demo: Etcd Operator managing an etcd cluster on Kubernetes
- Q&A

- **Operator Introduction**

- deploying an application on demand
- taking and restoring backups of that application's state
- handling upgrades of the application code alongside related changes such as database schemas
- publishing a Service to applications that don't support Kubernetes APIs to discover them
- simulating failure in all or part of your cluster to test its resilience
- choosing a leader for a distributed application without an internal member election process

- **The Operator Framework and SDK Length**

- SDK: Tools for Go, Ansible, Helm, ...
- Operator Lifecycle Manager & distributing Operators
- Exercise: Deploying from Operator Hub

- **The Operator Framework and SDK Length**

- Operator-sdk tool
- Operand: the example application
- Generating Operator scaffolding
- Exercise: Create, build, test, and deploy an Operator

- **Operators are software SREs Length**

- SRE principles
- Table stakes automation: Deploy, replication, lifecycle
- Operator Maturity Model
- The Pager Loop: Identifying advanced automation targets