Applied Containers & Orchestration

Overview

Containers have completely revolutionized the way that modern applications are deployed. They provide flexibility and adaptability that simply did not exist before. Deploying applications using containers can make systems more reliable, predictable, and portable. Understanding containers and container orchestrators are increasingly becoming a requirement for many software engineering disciplines. Container orchestration tools like Kubernetes make deploying large scale systems in predictable and reliable ways easier than ever before.
In this course we'll start by exploring the internals of containers, storage drivers, container networking, building images, and the security implications of all of it. Along the way you'll learn how to build, run, optimize, share and deploy modern container applications. Later we'll dive into how Kubernetes allows for deploying, scaling, and coordinating applications that run on containers. Have you ever wanted to know how to design a modern application that can handle hundreds or even thousands of clients at any given moment? Have you ever wanted to know how to design systems that can automatically resize to respond to your clients needs? Then this is the class you have been waiting for.

Prerequisites

- Strong Programming Background
- Operating Systems (6033 or 3224)

Grading

- 30% Weekly quizzes (drop lowest 2) (maybe a few multiple choice and a short answer)
- 10% Participation (lectures or forum)
- 60% Project (split up into 3 deadlines of varying weight)

Recommended Reading

- Docker: Up & Running (Kane & Matthias)
- Kubernetes: Up and Running (Burns, Beda & Hightower)
- Designing Data Intensive Applications Systems (Kleppmann)
- Networking & Kubernetes (Strong & Lancey)
- Container Security (Rice)

Semester Project

The semester long project will be to build a fully containerized key-value store system. The system should consist of an api, web service, and database. The api should be stateless REST api with endpoints for reading and writing key value pairs. The web service will simply be for serving website static files (html, css, js, etc.). To interact with your system, you will open your website, and manage your key value pairs.

Throughout the semester there will be several deadlines where we will be building up the functionality of this system from a very simple bare deployment to a production grade system.

Course Schedule (Tentative)

Theme	**Topics**	**Reading**
Container Basics	Container vs VM	
	Namespaces & Cgroups	
	Container Runtimes	
	Packaging Applications	
	Docker-compose & Swarm	
Components of containers	Images & Buildkit	
	Storage Drivers	
	Caching & Optimization	
	Linuxkit	
Using Containers Effectively	Sidecars, Ambassadors and Adapters	
	Modernizing legacy services	
	Abstracting complex APIs simply	
	Keeping your applications up	
Container Networking	Local network types & DNS	
	Container Networking Interfaces	

	Multi-node networking	
	Securing container networks	
	Spreading load, automatically	
Container Security	Isolation	
	Linux Permissions (root, capabilities, selinux, & privileged permissions)	
	Handling Credentials	
	Thinking through attack surfaces & Identifying potential blast radius	
	Common mistakes	
Kubernetes Basics	Components of internals	
	Pods vs Containers	
	Deployments & resource hierarchy	
	How to debug kubernetes applications (minikube)	
	Kubernetes vs docker-compose vs swarm	
	Types of Clusters (Managed vs Self-Hosted)	
Effective Kubernetes	Deployments advanced	
	Networking & Discovery	
	Managing Data with Persistent Volumes	
	Deploying Application Configuration Effectively	
	Keeping services up (probes, & rollouts)	
Advanced Kubernetes	Vertical vs Horizontal Scaling	
	Scaling,Scheduling and Nodes	

	(Horizontal Pod Autoscalers & smart scheduling)	
	RBAC & Interacting with Kubernetes API	
	Optimizations for faster scheduling and scaling	
	Packaging applications with helm	
Mastering Kubernetes	Maintenance done easy	
	Observability	
	Security first applications	
	Multi-User clusters	