Docker-Kubernetes Training

Kubernetes is an open-source orchestration system for automating management, placement, scaling and routing of containers. It provides an API to control how and where the containers would run.

Docker is also an open-source container-file format for automating the deployment of applications as portable, self-sufficient containers that can run in the cloud or on-premises.

Together, Kubernetes and Docker have become hugely popular among developers, especially in the DevOps world.

Table of content:

Docker

- Why Use Docker?
- What is Docker?
- Docker for Mac/Windows
- Installing Docker on MacOS
- Installing Docker with WSL2 on Windows 10 Home and Pro
- Installing Docker for Windows Professional with HyperV
- More Windows Professional Setup with HyperV
- One Last Piece of Windows Professional Setup with HyperV
- Installing Docker on Linux
- Using the Docker Client
- But Really...What's a Container?
- How's Docker Running on Your Computer?

Manipulating Containers with the Docker Client

- Docker Run in Detail
- Overriding Default Commands
- Listing Running Containers
- Container Lifecycle
- Restarting Stopped Containers
- Removing Stopped Containers
- Retrieving Log Outputs
- Stopping Containers
- Multi-Command Containers
- Executing Commands in Running Containers
- The Purpose of the IT Flag
- Getting a Command Prompt in a Container
- Starting with a Shell
- Container Isolation

Building Custom Images Through Docker Server

- Creating Docker Images

- Buildkit for Docker Desktop
- Building a Dockerfile
- Dockerfile Teardown
- What's a Base Image?
- The Build Process in Detail
- A Brief Recap
- Rebuilds with Cache
- Tagging an Image
- Manual Image Generation with Docker Commit

Making Real Projects with Docker

- Project Outline
- Node Server Setup
- A Few Planned Errors
- Required Node Base Image Version
- Base Image Issues
- A Few Missing Files
- Copying Build Files
- Container Port Mapping
- Specifying a Working Directory
- Unnecessary Rebuilds
- Minimizing Cache Busting and Rebuilds

Docker Compose with Multiple Local Containers

- App Overview
- App Server Starter Code
- Assembling a Dockerfile
- Introducing Docker Compose
- Docker Compose Files
- Networking with Docker Compose
- Docker Compose Commands
- Stopping Docker Compose Containers
- Container Maintenance with Compose
- Automatic Container Restarts
- Container Status with Docker Compose

Creating a Production-Grade Workflow

- Development Workflow
- Flow Specifics
- Docker's Purpose
- Project Generation
- Create React App Generation
- More on Project Generation
- Necessary Commands
- Creating the Dev Dockerfile
- Duplicating Dependencies
- Starting the Container
- Docker Volumes

- WSL2 and Windows Users Must Read Before Next Lecture
- Bookmarking Volumes
- Shorthand with Docker Compose
- Overriding Dockerfile Selection
- Do We Need Copy?
- Executing Tests
- Live Updating Tests
- Docker Compose for Running Tests
- Shortcomings on Testing
- Need for Nginx
- Multi-Step Docker Builds
- Implementing Multi-Step Builds
- Running Nginx

Continuous Integration and Deployment with AWS

- Services Overview
- GitHub Setup
- Important Info About Travis and Account Registration
- Travis CI Setup
- Travis YML File Configuration
- Required Travis Script Updates
- A Touch More Travis Setup
- Automatic Build Creation
- Required Updates for Amazon Linux 2 Platform - DO NOT SKIP
- AWS Elastic Beanstalk
- More on Elastic Beanstalk
- Travis Config for Deployment
- Automated Deployments
- Exposing Ports Through the Dockerfile
- Workflow With Github
- Redeploy on Pull Request Merge
- Deployment Wrapup
- Environment Cleanup
- AWS Configuration Cheat Sheet
- Finished Project Code with Updates Applied

Building a Multi-Container Application

- Single Container Deployment Issues
- Application Overview
- Application Architecture
- Worker Process Setup
- Express API Setup
- Important Update for Table Query
- Connecting to Postgres
- More Express API Setup
- Create React App Generation
- Generating the React App
- Fetching Data in the React App

- Rendering Logic in the App
- Exporting the Fib Class
- Routing in the React App

"Dockerizing" Multiple Services

- Checkpoint Files
- Checkpoint Catchup
- Expected Output for Latest Create React App Versions
- Dockerizing a React App - Again!
- Dockerizing Generic Node Apps
- Adding Postgres as a Service
- Docker-compose Config
- Postgres Database Required Fixes and Updates
- Environment Variables with Docker Compose
- Required Worker Environment Variables
- The Worker and Client Services
- Nginx Path Routing
- Routing with Nginx
- Building a Custom Nginx Image
- Starting Up Docker Compose
- Nginx connect () failed - Connection refused while connecting to upstream
- Troubleshooting Start-up Bugs
- WebSocket connection to 'ws://localhost:3000/ws' failed
- Opening WebSocket Connections

A Continuous Integration Workflow for Multiple Images

- Production Multi-Container Deployments
- Production Dockerfile
- Multiple Nginx Instances
- Nginx fix for React Router
- Altering Nginx's Listen Port
- Cleaning Up Tests
- GitHub and Travis CI Setup
- Fix for Failing Travis Builds
- Travis Configuration Setup
- Pushing Images to Docker Hub
- Successful Image Building

Multi-Container Deployments to AWS

- Important Info About Multi Containers on Amazon Linux 2 Platform
- Multi-Container Definition Files
- Finding Docs on Container Definitions
- Adding Container Definitions to DockerRun
- More Container Definitions
- Forming Container Links
- Using the Amazon Linux 2 Platform
- AWS Configuration Cheat Sheet - Updated for new UI
- Creating the EB Environment

- Managed Data Service Providers
- Overview of AWS VPC's and Security Groups
- RDS Database Creation
- ElastiCache Redis Creation
- Creating a Custom Security Group
- Applying Security Groups to Resources
- Setting Environment Variables
- IAM Keys for Deployment
- Travis Keys Update
- Travis Deploy Script
- Container Memory Allocations
- Verifying Deployment
- Making Changes
- Cleaning Up AWS Resources
- AWS Configuration Cheat Sheet
- Finished Project Code with Updates Applied

Onwards to Kubernetes!

- The Why's and What's of Kubernetes
- Kubernetes in Development and Production
- Docker Desktop's Kubernetes Setup and Installation - macOS
- Docker Desktop's Kubernetes Setup and Installation - Windows
- Updated Minikube Install and Setup Info - macOS
- Minikube Setup on MacOS
- Minikube Setup on Windows
- Minikube Setup on Linux
- Mapping Existing Knowledge
- Adding Configuration Files
- Object Types and API Versions
- Running Containers in Pods
- Service Config Files in Depth
- Connecting to Running Containers
- The Entire Deployment Flow
- Imperative vs Declarative Deployments

Maintaining Sets of Containers with Deployments

- Updating Existing Objects
- Declarative Updates in Action
- Limitations in Config Updates
- Running Containers with Deployments
- Deployment Configuration Files
- Walking Through the Deployment Config
- Applying a Deployment
- Why Use Services?
- Scaling and Changing Deployments
- Updating Deployment Images
- Rebuilding the Client Image
- Triggering Deployment Updates

- Imperatively Updating a Deployment's Image
- Reminder for Docker Desktop's Kubernetes Users
- Multiple Docker Installations
- Reconfiguring Docker CLI
- Why Mess with Docker in the Node?

A Multi-Container App with Kubernetes

- The Path to Production
- Checkpoint Files
- Recreating the Deployment
- NodePort vs ClusterIP Services
- The ClusterIP Config
- Applying Multiple Files with Kubectl
- Express API Deployment Config
- Cluster IP for the Express API
- Combining Config Into Single Files
- The Worker Deployment
- Reapplying a Batch of Config Files
- Creating and Applying Redis Config
- Important Note about Expected Postgres Error
- Last Set of Boring Config!
- The Need for Volumes with Databases
- Kubernetes Volumes
- Volumes vs Persistent Volumes
- Persistent Volumes vs Persistent Volume Claims
- Claim Config Files
- Persistent Volume Access Modes
- Where Does Kubernetes Allocate Persistent Volumes?
- Designating a PVC in a Pod Template
- Applying a PVC
- Defining Environment Variables
- Adding Environment Variables to Config
- Creating an Encoded Secret
- Postgres Environment Variable Fix
- Passing Secrets as Environment Variables
- Environment Variables as Strings

Handling Traffic with Ingress Controllers

- Load Balancer Services
- Behind the Scenes of Ingress
- More Behind the Scenes of Ingress
- Optional Reading on Ingress Nginx
- Docker Driver and Ingress - IMPORTANT
- Important - DO NOT SKIP - Ingress Nginx Installation Info
- Setting up Ingress Locally with Minikube
- Ingress v1 API Update + "this.state.seenIndexes.map..." + 404 errors
- Creating the Ingress Configuration
- Testing Ingress Locally

Pioneer Business Solutions

- The Minikube Dashboard
- Docker Desktop's Kubernetes Dashboard

Kubernetes Production Deployment

- The Deployment Process
- Google Cloud vs AWS for Kubernetes
- Creating a Git Repo
- Linking the Github Repo to Travis
- Free Google Cloud Credits
- Creating a Google Cloud Project
- Linking a Billing Account
- Updated GKE creation steps for new Google Cloud UI
- Kubernetes Engine Init
- Creating a Cluster with Google Cloud
- Don't Forget to Cleanup!
- Kubernetes Dashboard on Google Cloud
- Travis Deployment Overview
- Installing the Google Cloud SDK
- Updated Service Account steps for new GCP UI
- Generating a Service Account
- Ruby Version Fix
- Running Travis CLI in a Container
- Travis Login Issues, "iv undefined" or "repository not known"
- Encrypting a Service Account File
- More Google Cloud CLI Config
- Fix For Failing Travis Builds
- Running Tests with Travis
- Custom Deployment Providers
- Unique Deployment Images
- Unique Tags for Built Images
- Updating the Deployment Script
- Configuring the GCloud CLI on Cloud Console
- Creating a Secret on Google Cloud
- Helm v3 Update
- Helm Setup

Kubernetes Security with RBAC

- Assigning Tiller, a Service Account
- Ingress-Nginx with Helm
- The Result of Ingress-Nginx
- Finally - Deployment!
- Did I Really Type That?
- Verifying Deployment
- A Workflow for Changing in Prod
- Merging a PR for Deployment
- That's It! What's Next?
- Completed Code For Google Cloud Deployment

Pioneer Business Solutions

HTTPS Setup with Kubernetes

- HTTPS Setup Overview
- Domain Purchase
- Domain Name Setup
- Required Updates for Cert Manager Install
- Cert Manager Install
- How to Wire Up Cert Manager
- Required Update for Issuer
- Issuer Config File
- Required Update for the Certificate
- Certificate Config File
- Deploying Changes
- No Resources Found?
- Verifying the Certificate
- Required Update for the HTTPS Ingress
- Ingress Config for HTTPS
- It Worked!
- Google Cloud Clean-up
- Local Environment Clean-up

Pioneer Business Solutions