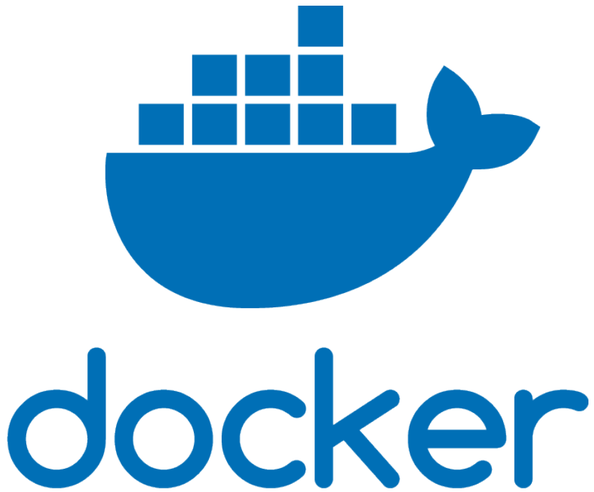
Docker in Production using AWS

COURSE INTRODUCTION

Justin Menga
FULL STACK TECHNOLOGIST
@jmenga pseudo.co.de

Docker in Production Using AWS



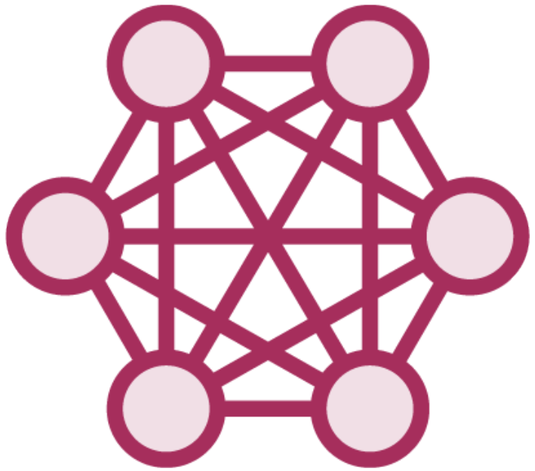
Docker

Amazon Web Services

Course Goals

Course Goals

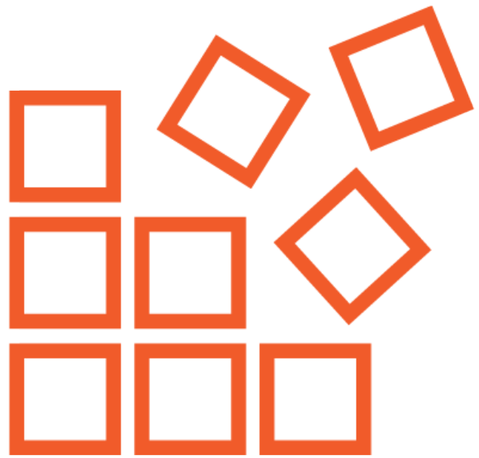
Build a production-class Docker application platform



Learn how to run Microservices on AWS

Course Goals

Adopt native AWS services



Learn how to use the
EC2 Container Service

Integrate with Native AWS Services

Route 53

CodeBuild

IAM

Load Balancing

KMS

RDS

S3

Auto Scaling

Lambda

VPC

CloudFormation

EC2

CodePipeline

ECS

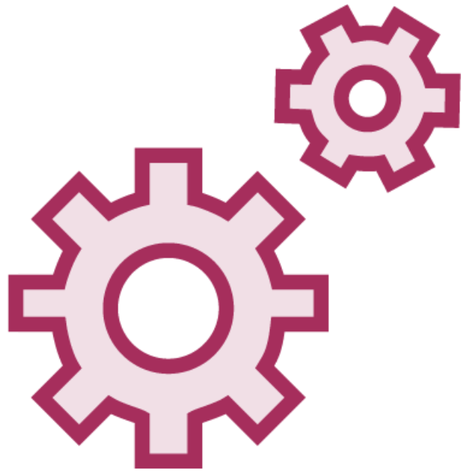
CloudWatch

Course Goals

Infrastructure as code and automated deployments

Deploy environments using Ansible and CloudFormation

Course Goals

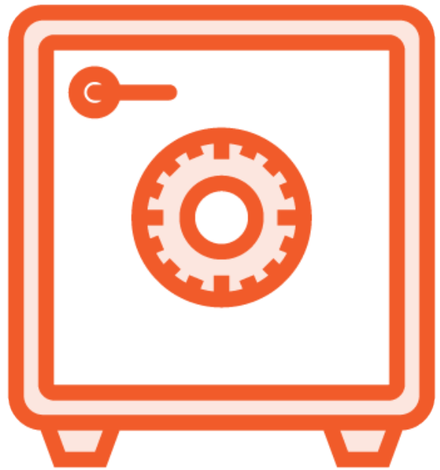


Custom provisioning tasks using AWS Lambda

Build and deploy Lambda functions written in Python

Course Goals

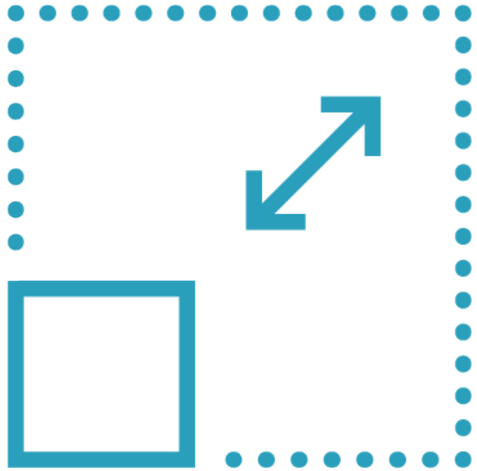
Secure secrets management



Create a secure store for storing encrypted secrets

Course Goals

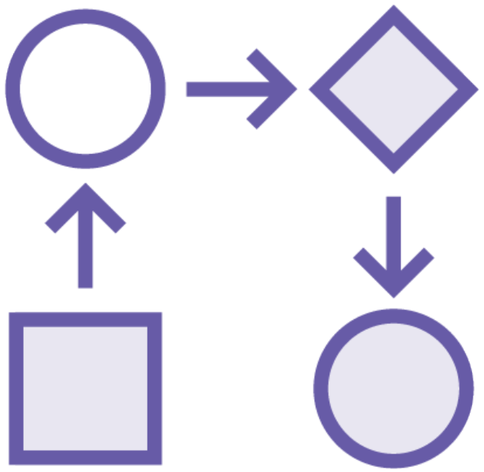
Infrastructure lifecycle management



Auto scaling ECS applications

Course Goals

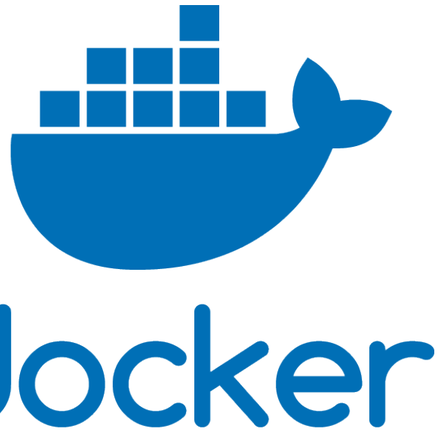
Create a continuous delivery pipeline



Test, build, publish and deploy

Course Prerequisites

No experience required



Beginner to Intermediate

Beginner to Intermediate

Beginner

Beginner

Beginner

Course Audience

Operations

Build a production container platform

Solve key operational challenges

Auto scale your infrastructure

Infrastructure as code

Developers

Overcome the challenges of running distributed Docker applications in AWS

Learn how to test, build and publish your application in Docker images

Leverage AWS services

Build a continuous delivery pipeline

Continuously deploy to a development environment

Architects

Leverage cloud native services

Reusable blueprint for building and deploying AWS services

Immutable Infrastructure

Infrastructure as Code

Course Tour

Installing the Sample Application

Install Sample Application

Test and Build Application

Run Application

Creating Docker Release Images

Build a Docker Image

Run local Docker Workflow

Introduce Workflow Tooling

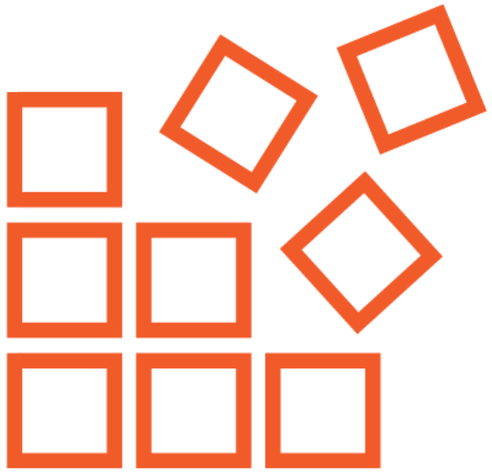
Setting up AWS Access

Establish AWS Account

Identity Access & Management

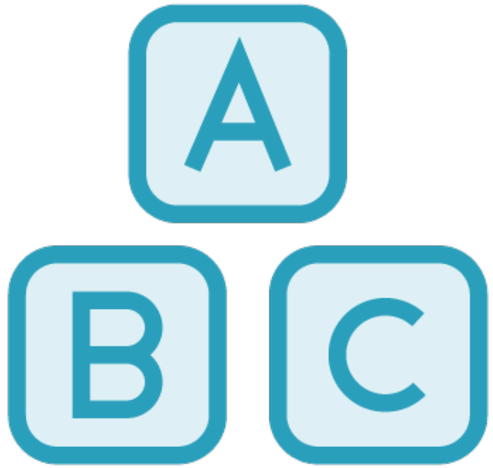
Multi Factor Authentication

Running Docker Applications Using ECS



EC2 Container Service

Publish to EC2 Container Registry



Create an ECS Cluster

Customizing ECS Container Instances

Create an Amazon Machine Image

ECS Agent

CloudWatch Logs Agent

Deploying AWS Infrastructure Using Ansible and CloudFormation

Create Ansible Playbooks

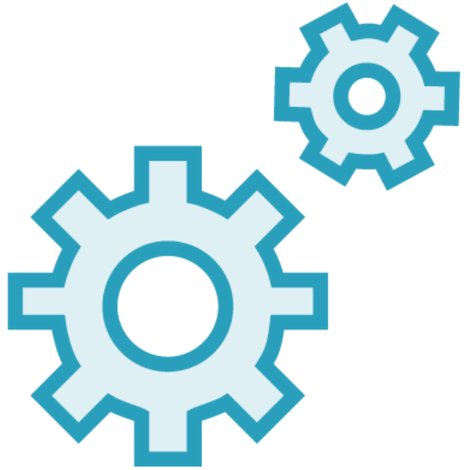
Integrate with CloudFormation

Establish Shared AWS Resources

Architecting and Preparing Applications for ECS

Application Cluster Discovery on AWS

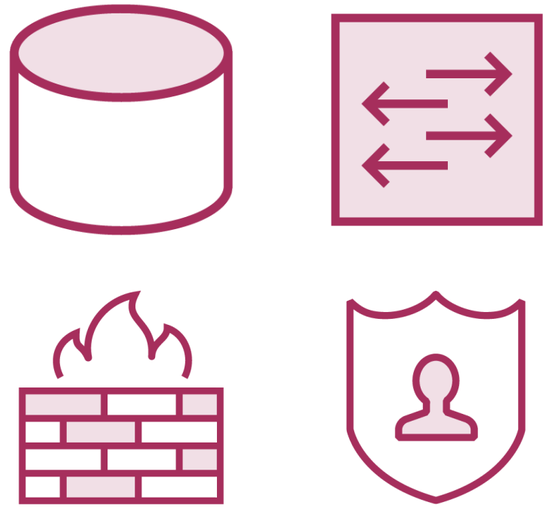
Customize Container Startup



Generate Config Files Dynamically

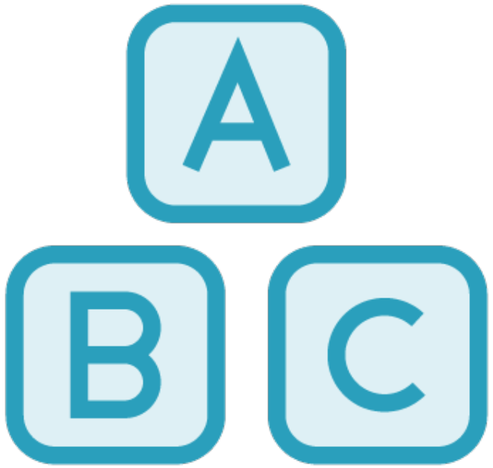
Defining ECS Applications Using Ansible and CloudFormation

Create CloudFormation Stack for the Application



Configure Supporting Resources

Deploying ECS Applications Using Ansible and CloudFormation



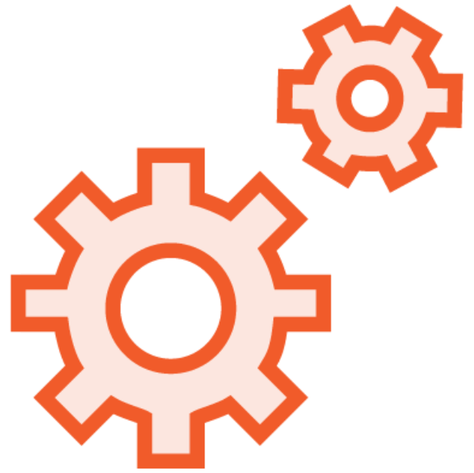
Define ECS Resources Using CloudFormation

Deploy Application to Development Environment

Creating CloudFormation Custom Resources

CloudFormation Custom Resources

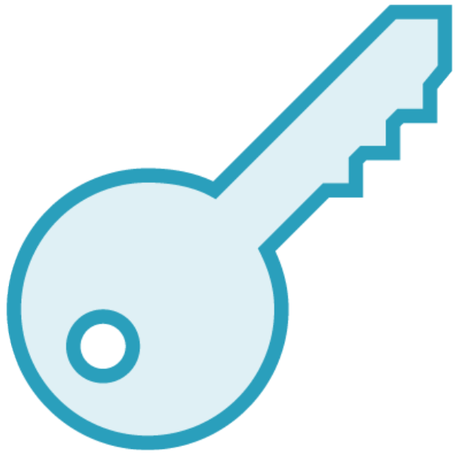
AWS Lambda



Create an ECS Task Runner

Managing Secrets in AWS

Secrets Management Solution



AWS KMS + EC2 Parameter Store

Inject Secrets into Containers

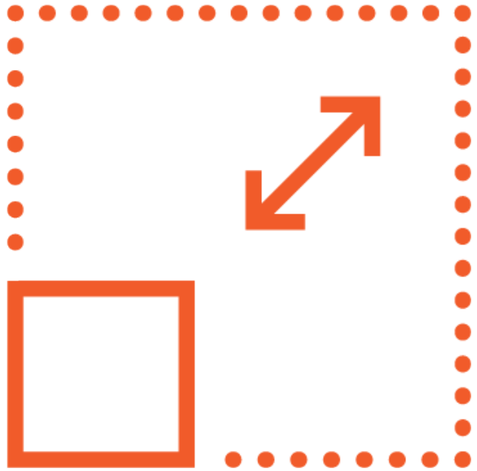
Managing ECS Infrastructure Lifecycle

Lifecycle Management

EC2 Lifecycle Hooks

Lifecycle Hook Lambda Function

Auto Scaling ECS Applications



ECS
Auto Scaling

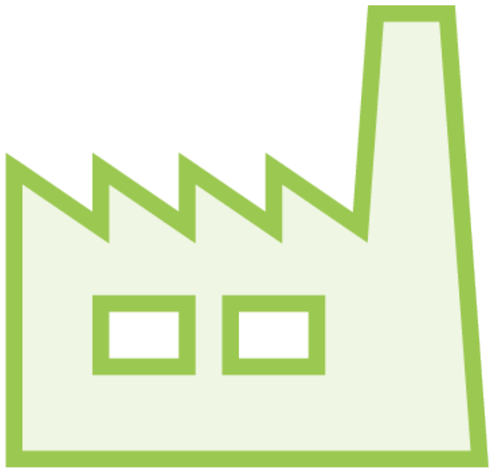
ECS Capacity
Management

Application
Auto Scaling

Continuous Delivery Using CodePipeline

Create a Continuous Delivery Pipeline

Continuously Deploy to Development



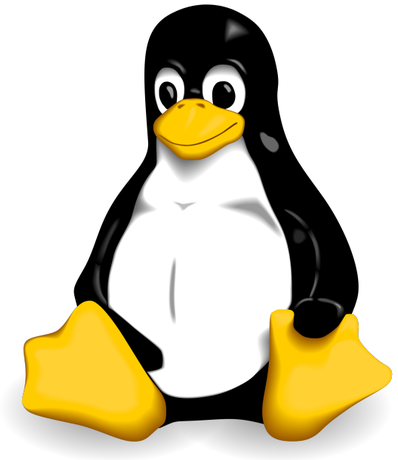
Deploy to Production (with approval)

Course Setup

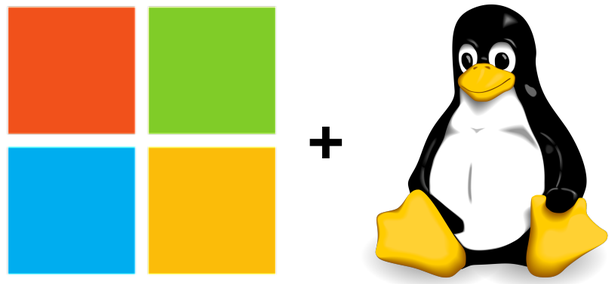
Local Development Environment

macOS

Linux



Windows + Linux VM



Demo

Preparing your Environment

- Installing Docker
- Installing Brew
- Installing Java
- Installing Ansible
- Other Recommended Tools
- Setting Up Required Services
- Creating the Course Root Folder

Installing Docker

Installing Brew

Installing Java

Installing Ansible

Other Recommended Tools

Setting up Required Services

GitHub and AWS

GitHub
github.com

AWS
aws.amazon.com/free

Avoiding Excessive AWS Costs

Use Free Tier services

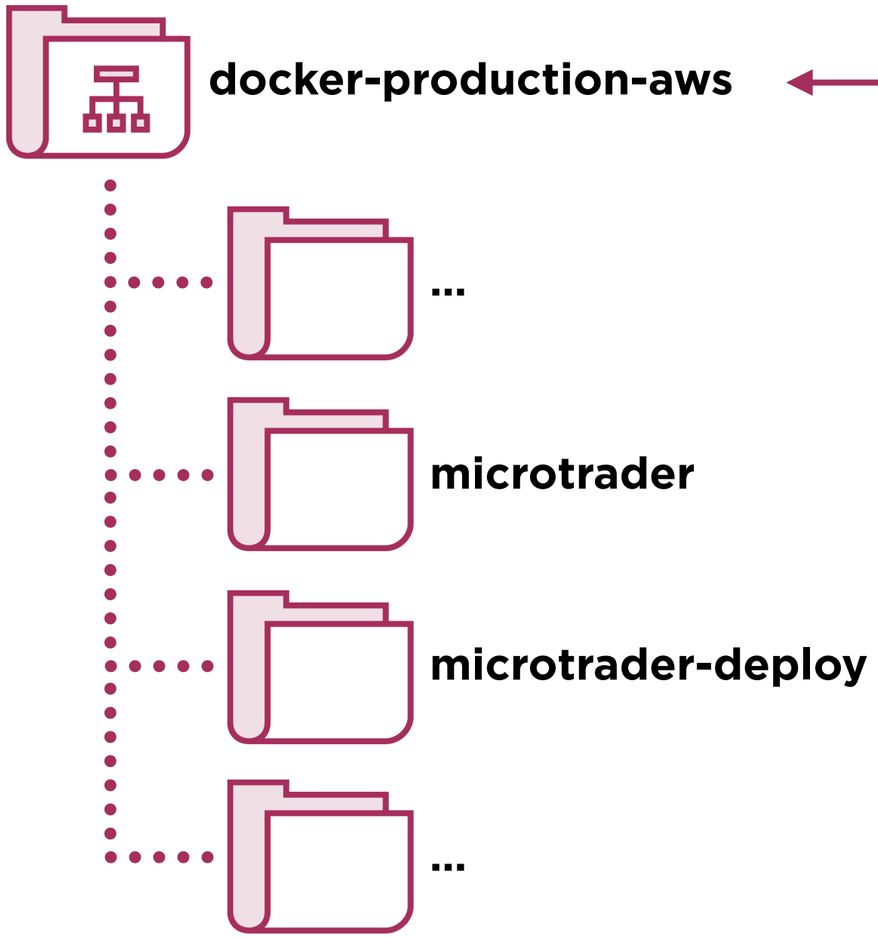
See aws.amazon.com/free for full details

Use CloudFormation to destroy stacks with running EC2 or RDS instances when not in use

Review and clean up any manually created resources after each module

Setting up a Course Root Folder

Course Root Folder Structure



Course Root Folder

Repositories

All repositories are published at
github.com/docker-production-aws

Summary

Course Introduction

- Deploy a Microservices application to production using Docker and AWS
- Leverage AWS services
    - EC2 Container Service
    - CloudFormation
    - and many more...
- Infrastructure as code
- Ansible + CloudFormation for fully automated deployments

Summary

Course Introduction

- Solve operational challenges
  - Secrets management
  - Infrastructure lifecycle management
  - Dynamic auto scaling
- Create a continuous delivery pipeline
  - CodePipeline + CodeBuild
  - CloudFormation
  - GitHub integration

Summary

Course Introduction

- Course setup
  - Docker
  - Brew Package Manager
  - Python + Ansible
  - Java Development Kit
- Required services
  - GitHub
  - Amazon Web Services
- Course root folder