Welcome to my GitHub repository showcasing my DevOps projects. Below are the descriptions and details of the projects I've worked on during my career as a Junior DevOps Engineer.

**Table of Contents**

**- [Automated Deployment Pipeline](#automated-deployment-pipeline)**

**- [Cloud Cost Optimization Tool](#cloud-cost-optimization-tool)**

**- [Containerized Application Deployment](#containerized-application-deployment)**

**Automated Deployment Pipeline**

**Overview**

This project involves the creation of a Continuous Integration and Continuous Deployment (CI/CD) pipeline using Jenkins, Docker, and AWS. The goal was to automate the deployment process, reduce manual interventions, and enhance the overall deployment efficiency.

**Features**

- Automated build and deployment process.

- Integration with Jenkins for CI/CD.

- Docker for containerization of applications.

- Deployment on AWS infrastructure.

**Technologies Used**

- Jenkins

- Docker

- AWS (EC2, S3, RDS)

- GitHub

**Impact**

- Reduced deployment time by 40%.

- Increased deployment consistency and reliability.

**Cloud Cost Optimization Tool**

**Overview**

This project focuses on optimizing cloud costs by building a tool to analyze and manage AWS expenditures. The tool leverages AWS SDK and Python to provide insights into cloud usage and suggest optimizations.

**Features**

- Analysis of AWS cost and usage data.

- Recommendations for cost-saving opportunities.

- Detailed reports on cloud spending.

**Technologies Used**

- Python

- AWS SDK (Boto3)

- AWS Cost Explorer

**Impact**

- Achieved a 25% reduction in monthly cloud costs.

- Improved visibility into cloud spending.

**Containerized Application Deployment**

**Overview**

This project aimed to improve the scalability and maintainability of a legacy application by containerizing it using Docker and deploying it on a Kubernetes cluster.

**Features**

- Containerization of legacy application using Docker.

- Deployment and orchestration with Kubernetes.

- Improved scalability and maintainability of the application.

**Technologies Used**

- Docker

- Kubernetes

- Helm

- AWS EKS

**Impact**

- Enhanced scalability and ease of management.

- Improved resource utilization and application performance.

Getting Started

**Prerequisites**

To run these projects, you will need the following:

- Docker

- Jenkins

- AWS account

- Python (for the Cloud Cost Optimization Tool)

- Kubernetes cluster (for Containerized Application Deployment)