AWS Infrastructure with TypeScript: Getting Started

CREATING A TYPESCRIPT CDK APP

David Tucker
TECHNICAL ARCHITECT & CTO CONSULTANT
@_davidtucker_ davidtucker.net

Globomantics

Ellen, CTO

Globomantics is a global manufacturing company

Last year Ellen led a transition to AWS and away from their own data centers

Cloud infrastructure is created as needed, much like in their data centers

The organization has run into multiple challenges since this transition

Globomantics

Josh
Cloud Architect

Recently joined Globomantics

Has experience building applications on Amazon Web Services

Has worked with CloudFormation, but has not worked with the CDK

Will be in charge of the document management proof of concept

Overview

Reviewing the architecture of the solution that will be built in this course

Creating a new TypeScript CDK app using the CLI

Examining the project configuration generated by the CLI

Managing CDK packages and dependencies

Reviewing the different types of constructs in the CDK

What We Will Build

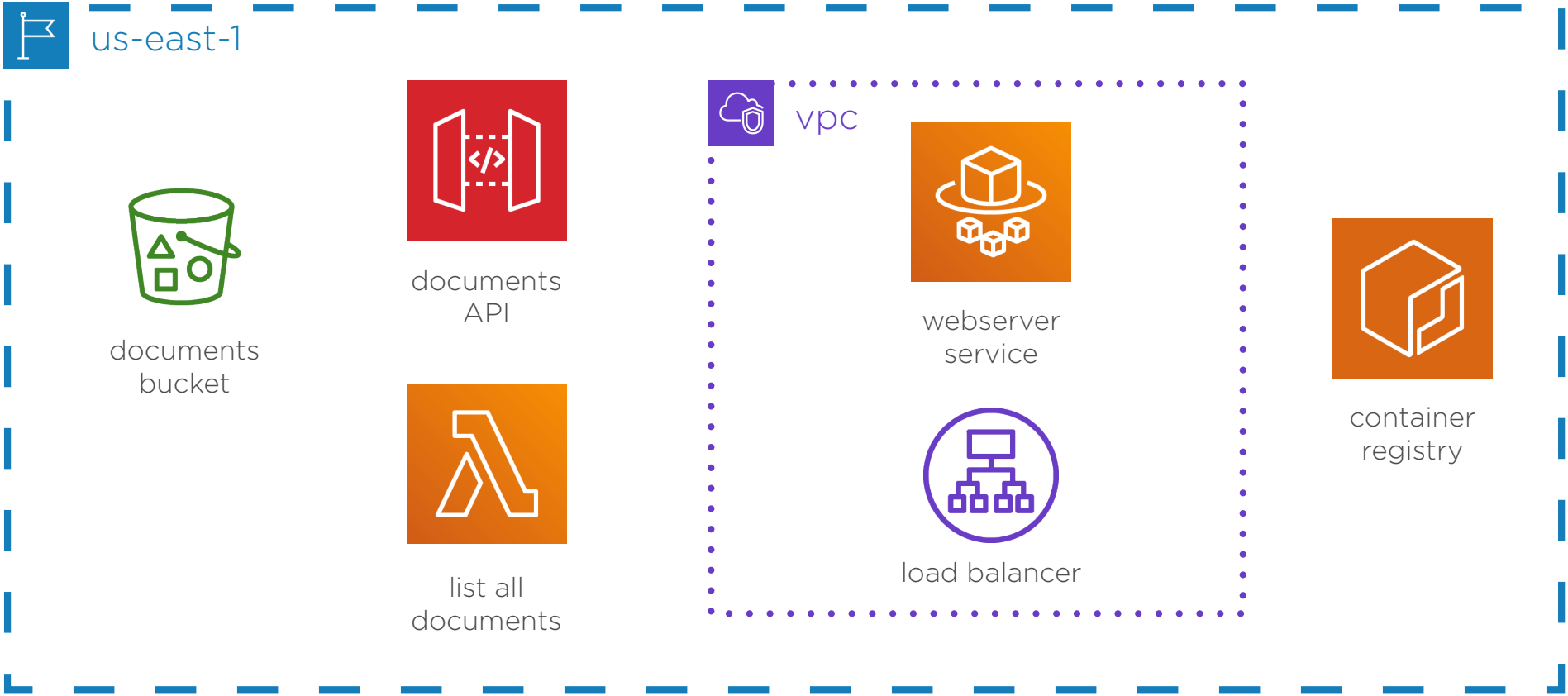
Course Goals

- Be able to create a new TypeScript application using the CDK CLI
- Be able to deploy your CDK application to an environment
- Be able to install and manage service specific packages
- Be able to create a Lambda-based API using local assets
- Be able to launch a Fargate-based web server using local assets
- Be able to manage permissions and network access with the CDK

Document Management

The use case for this course will be an enterprise document management system. This will be just a bare bones proof of concept utilizing the CDK. This could be a starting point for a real solution.

Sample Architecture



Notes on Approach

We will be leveraging NPM for managing application dependencies

I will be using a Mac, but all concepts should transfer to Windows

Visual Studio Code will be the IDE that is used in the course

The application use case is not designed to be a production application

Infrastructure as Code

Managing infrastructure should be as easy as managing your application code.

Benefits of Development Workflow

Version Control

Collaborative Review

Automated Testing

Continuous Delivery

Infrastructure as Code

The approach of managing infrastructure and its configuration in human readable definition files instead of manually launching and configuring infrastructure.

Revisiting Challenges

Manual Infrastructure

Separate deployment workflows for infrastructure and code

Manual processes are error-prone

Environments can easily get out of sync

Environment configuration drifts over time

Standing up new environments is time consuming and error-prone

Infrastructure as Code

Deployment workflow can be unified between infrastructure and code

There is no need for manual processes

Since workflow is unified and manual processes eliminated, it stays in sync

Many Infrastructure as Code solutions provide drift detection

Launching a new environment is quick and easy

Use of TypeScript

Throughout this course, TypeScript will be leveraged for infrastructure as code, as well as for the code for the API's and webserver. It will be used as an end to end solution.

Generating a New TypeScript CDK App

Demo

Verifying CDK installation

Utilizing the CDK to generate a new app

Reviewing the CDK TypeScript project structure

Managing CDK Dependencies

Demo

- Accessing the CDK documentation
- Installing service-specific packages for use with the CDK
- Reviewing best practices for managing dependencies for a project
- Configuring an S3 bucket with the CDK

Examining Construct Levels in the CDK

Demo

Reviewing L1 and L2 constructs within the CDK documentation

Utilizing an L1 construct