



DEPLOYING A PRISM  API IN AWS

BY: CALEB BRIGGS

ABOUT ME

- ▶ I am the owner of Tyc00n Software where we use the latest cutting edge technology to make our partners vision a reality
- ▶ Any more I mostly just entertain my wife and two girls one 3 years old and one 8 months old but also enjoy grilling out, hunting, fishing, golf, drones/rc cars, games, and travel besides programming of course

DISCLAIMER

- ▶ Everything we will be covering today can be found somewhere in the prisma docs (I have found them to be really good)
- ▶ <https://www.prisma.io/docs>

PRISMA

- ▶ Prisma is an open source service. It is a *database abstraction layer* that turns your databases into GraphQL APIs with CRUD operations and realtime capabilities. It is the glue between your database and GraphQL server.
- ▶ It can create and update database schema with graphql SDL
- ▶ Has robust graphql api do all crud operations and do much of heavy lifting with querying and creating/updating complex data structures

DESCRIPTION

- ▶ Today I am going to show you how you can very quickly and simply get a graphql api setup in AWS using the following tools:
 - ▶ AWS cloud formation template
 - ▶ AWS Fargate
 - ▶ AWS Elastic Container Service
 - ▶ Prisma/PrismaCLI (you may need to create an account but this is an OSS)
 - ▶ Bring your own DB (Mysql, Postgres, or Mongo for now, many more coming)

AWS CLOUD FORMATION TEMPLATE

- ▶ <https://github.com/prismagraphql/prisma-templates/blob/master/aws/fargate.yml>
- ▶ Demo AWS Cloud Formation

AWSTemplateFormatVersion: '2010-09-09'

Description: AWS Fargate Prisma stack.

Parameters:

PrismaVersion:

Type: String

Default: 1.11.0

AllowedValues:

- 1.11.0

- 1.10.2

- 1.9.0

- 1.8.4

- 1.8.3

- 1.7.4

DbHost:

Type: String

DbPort:

Type: Number

Default: 3306

DbUser:

Type: String

DbPassword:

Type: String

NoEcho: true

DbConnector:

Type: String

Default: mysql

Cpu:

Type: String

Description: The CPU units for the container. Must adhere to [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ecs-task-definition.html)

Default: 1024

Memory:

Description: The memory reservation for the container. Must adhere to [https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ecs-task-definition.html)

Type: String

Default: 2GB

AWS FARGATE

- ▶ AWS Fargate is a compute engine for Amazon ECS and EKS* that allows you to run containers without having to manage servers or clusters. With AWS Fargate, you no longer have to provision, configure, and scale clusters of virtual machines to run containers. This removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing. AWS Fargate removes the need for you to interact with or think about servers or clusters.

AWS ECS

- ▶ AWS Elastic Container service has two main components that I want to touch on:
 - ▶ ECR -> Elastic container registry where you can put your private docker containers then spin them up in ECS
 - ▶ ECS -> Put your service in a cluster and let AWS scale your services entirely with Fargate or take a more granular approach and choose your own EC2 instances and networking

DEMO

- ▶ Code - prisma-test
- ▶ Show docker-compose file for local use
- ▶ <http://prism-publi-1nvv6ltys2z98-573029174.us-west-2.elb.amazonaws.com/prisma/devops>
- ▶ Prisma generate
- ▶ Node test

TEXT

QUESTIONS

