Introduction

100 XP

5 minutes

Imagine you're an IT engineer at Fruit Smoothies, a nationwide chain of smoothie shops. The company's development team developed a new website that allows users to rate the company's different smoothy flavors. Fruit Smoothies has outlets worldwide with a large follower base and the company expects many fans to visit the new website. You want to make sure when you deploy the ratings website, you can scale the site quickly when needed.

The ratings website consists of several components. There's a web frontend, a document database that stores captured data, and a RESTful API. The API allows the web frontend to communicate with the database. Since the ratings website may store sensitive data, an additional requirement is to protect the site with an HTTPS certificate.

Fruit Smoothies wants to use Kubernetes as their compute platform. The development teams already use containers for application development and deployment, and using an orchestration platform will introduce many benefits. Kubernetes is a portable, extensible, open-source platform for automating the deployment, scaling, and management of containerized workloads. Kubernetes abstracts away complex container management and provides declarative configuration to orchestrate containers in different compute environments. This orchestration platform give the same ease of use and flexibility as with Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) offerings.

In this module, you'll go through tasks to deploy a multicontainer application to Kubernetes on Azure Kubernetes Service (AKS).

You'll use your Azure subscription to deploy the resources in this workshop. To estimate the expected costs for these resources, see the preconfigured Azure Calculator estimate of the resources that you'll deploy.

Learning objectives

In this module, you will:

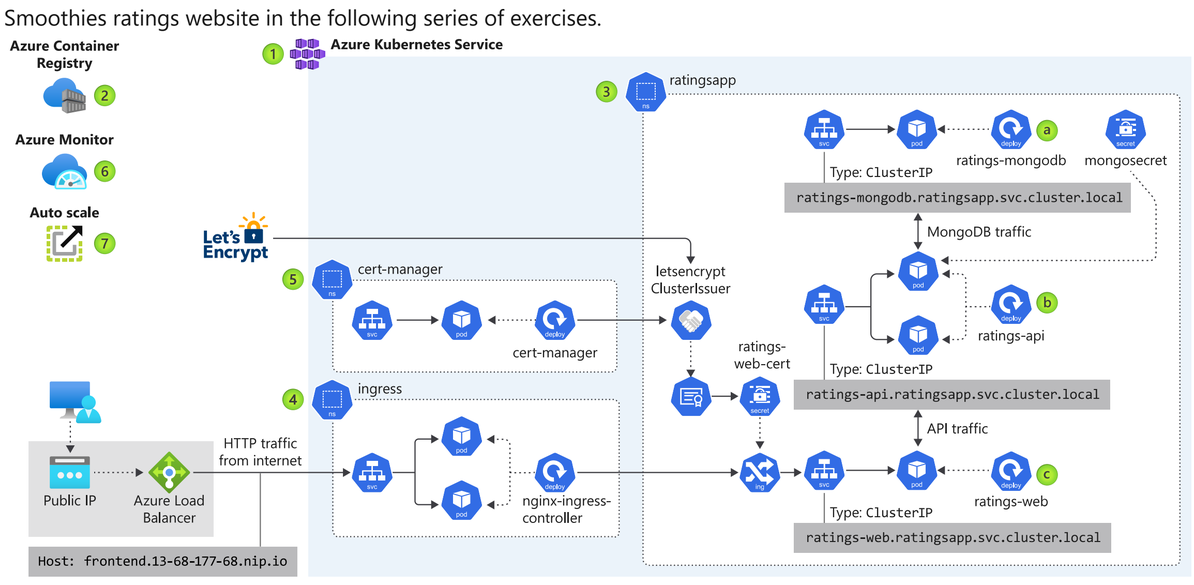
- Create an Azure Kubernetes Service cluster
- Choose the best deployment options for you Pods
- Expose Pods to internal and external network users
- Configure SSL/TLS for Azure Kubernetes Service ingress
- Monitor the health of an Azure Kubernetes Service cluster
- Scale your application in an Azure Kubernetes Service cluster

Prerequisites

- Familiarity with Kubernetes and its concepts. If you're new to Kubernetes, start with the basics of Kubernetes.
- An Azure subscription to deploy resources in.
- Familiarity with Azure Cloud Shell.
- A GitHub account.

Application architecture

Our goal is to deploy an Azure managed Kubernetes service, Azure Kubernetes Service (AKS), that runs the Fruit Smoothies ratings website in the following series of exercises.



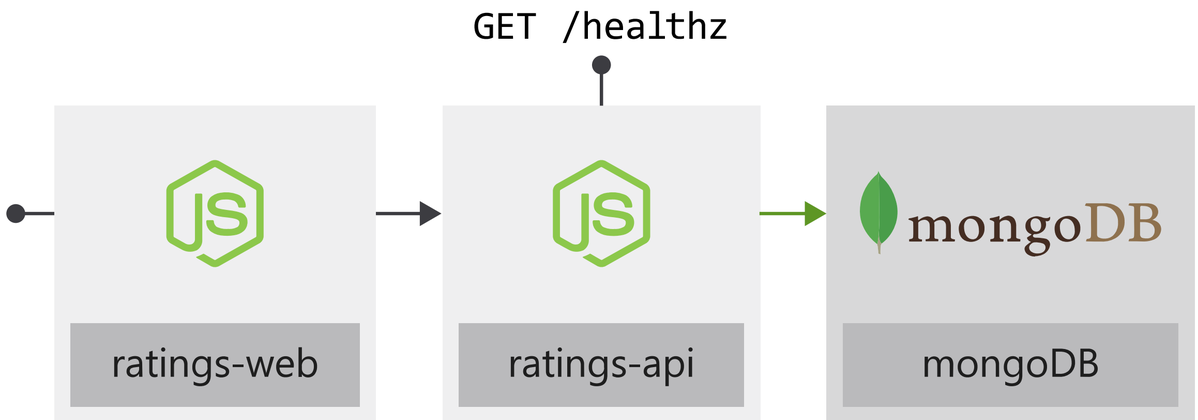
There are several tasks that you'll complete to show how Kubernetes abstracts away complex container management and provides you with declarative configuration to orchestrate containers.

1. Use AKS to deploy a Kubernetes cluster.
2. Configure an Azure Container Registry to store application container images.
3. Deploy the three ratings application components.
   a. Deploy the Fruit Smoothies website document database using Helm 3.
   b. Deploy the Fruit smoothies RESTFul API using deployment manifests.
   c. Deploy the Fruit smoothies website frontend using deployment manifests.
4. Deploy Azure Kubernetes ingress using Helm 3.
5. Configure SSL/TLS on the controller using `cert-manager`.
6. Configure Azure Monitor for containers to monitor the Fruit Smoothies website deployment.
7. Configure cluster autoscaler and horizontal pod autoscaler for the Fruit Smoothies cluster.

Source code

The application consists of two components: the API and the front end. Both components are written in Node.js. The API stores data in a MongoDB database.

Component	Link
An API `ratings-api`	GitHub repo
A front end `ratings-web`	GitHub repo



Next unit: Exercise - Deploy Kubernetes with Azure Kubernetes Service

Continue

© Microsoft 2020