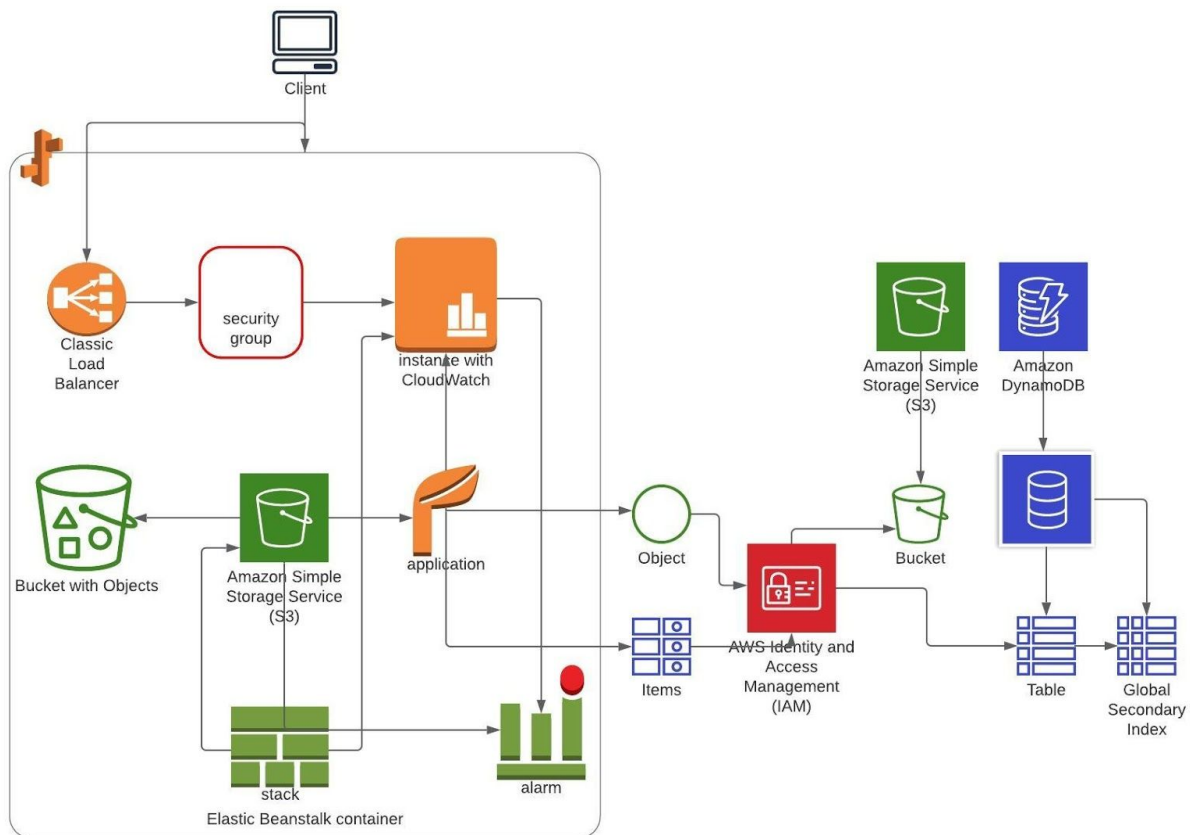


Program 4

Design Diagram:



Amazon elastic beanstalk automatically creates an environment that includes “An Amazon Elastic Compute Cloud (Amazon EC2) instance (virtual machine), An Amazon EC2 security group, An Amazon Simple Storage Service (Amazon S3) bucket, Amazon CloudWatch alarms,

An AWS CloudFormation stack, A domain name” the architecture of the web application is as follows:

When a client accesses my website from the url, An Amazon EC2 security group listens to and allows incoming traffic on port 80, it then lets HTTP traffic from the load balancer reach the EC2 instance running my web app. The source code and other dependencies that the application runs on are stored in an amazon s3 bucket within the elastic beanstalk “environment”. All of these resources are launched on AWS cloud formation.

The application itself makes calls to my own S3 bucket as well as my own dynamoDB that exist on their own outside of the elastic beanstalk environment.

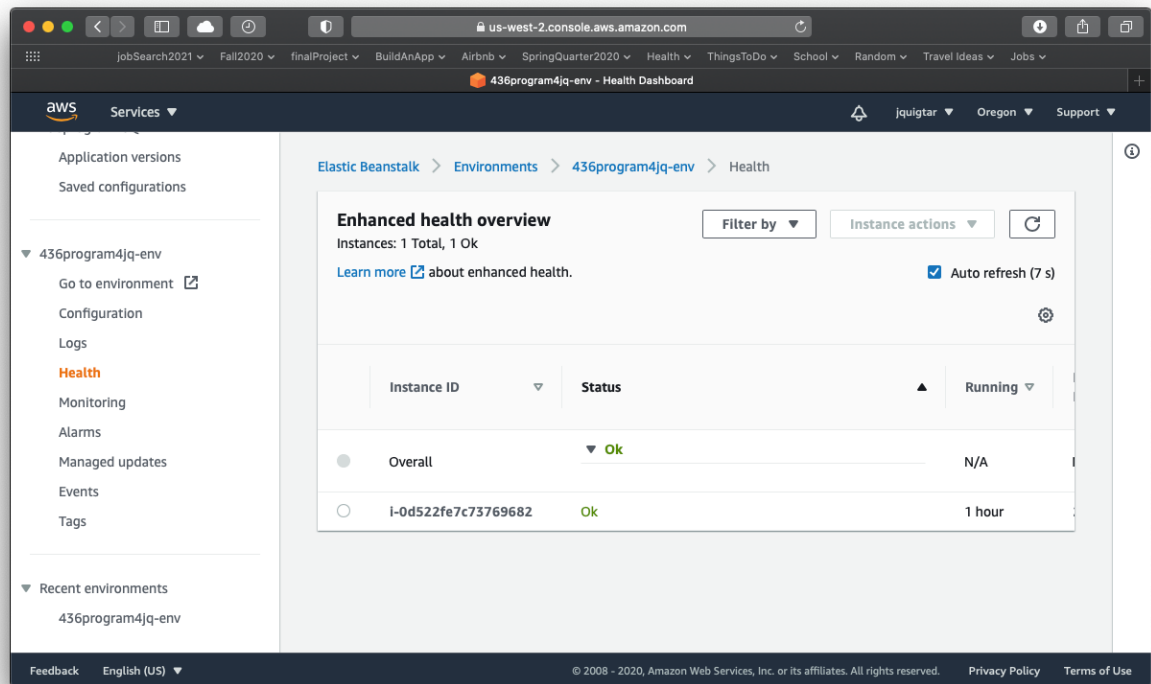
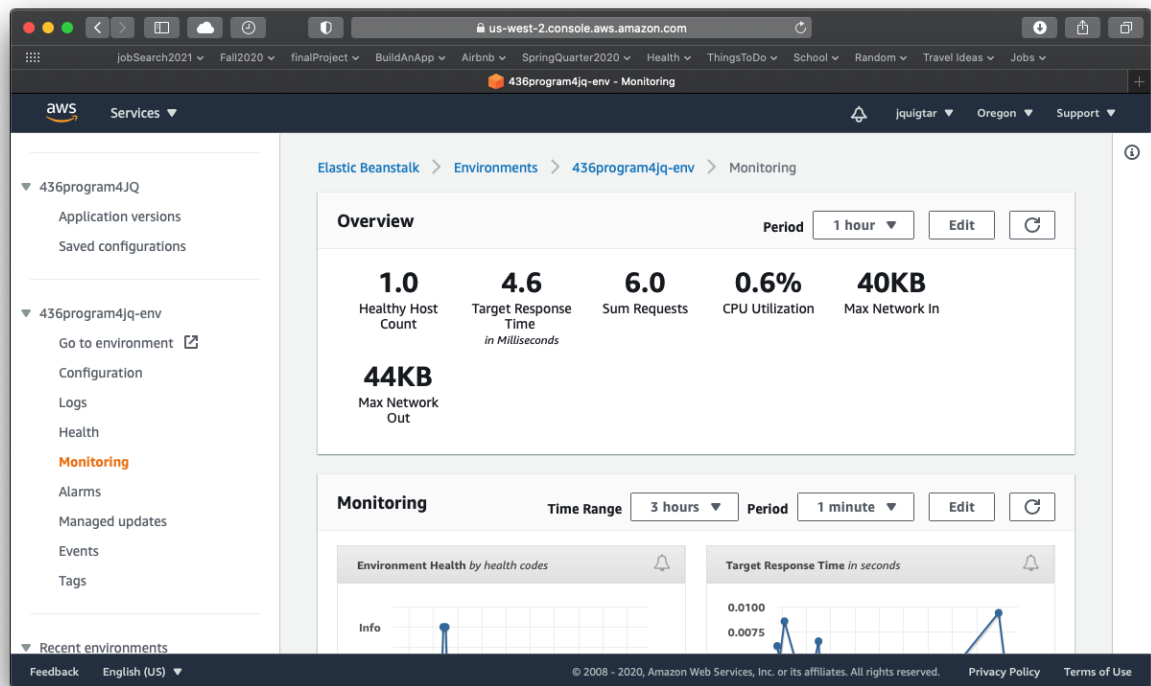
Sources: <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.CreateApp.html>
<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.CreateApp.html#GettingStarted.CreateApp.AWSresources>

How site will scale with load:

When uploading a web application to AWS elastic beanstalk it is set up with two cloudwatch alarms that monitor the load on the EC2 instances in the elastic beanstalk environment. If the load becomes too low or too high and the alarm is triggered, an auto scaling group will scale up or down in response. Also, in the AWS console you can configure these alarms to your own specific configurations such as amount of requests, or cpu utilization.

Source:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.CreateApp.html#GettingStarted.CreateApp.AWSresources>



Monitoring of site:

From the Amazon console I can view the health of my web app that includes the amount of hours the application has been up and running without fault, requests/second, as well as counts for amount of 2xx, 3xx, 4xx, 5xx responses given by my website, as well as latency statistics of my website. I can also view the “monitoring” section from my elastic beanstalk console which shows response time, cpu utilization, amount of requests to my website, and network performance.

SLA Estimate:

Amazon DynamoDB = 99.999%

2 x Amazon S3 = 99.9%

Amazon EC2 = 99.99%

Amazon elastic load balancing = 99.99%

SLA estimate = $99.999 * 99.9 * 99.9 * 99.99 * 99.99 = 99.9779\%$ uptime