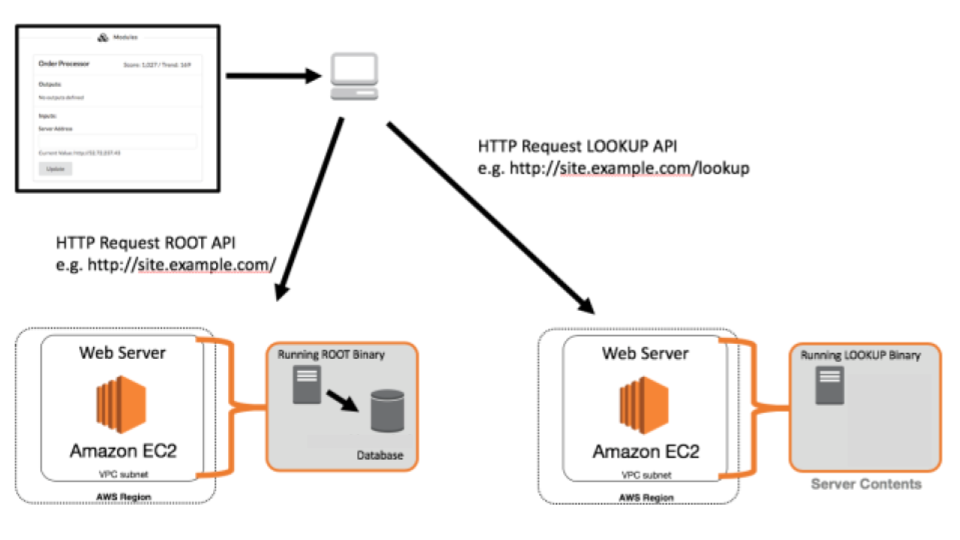
**Technical Details**

1. The root server application is deployed as a Go binary compiled from source. Do not alter the binary in any way as that will be grounds for disqualification.
2. The lookup server application is deployed as a Go binary compiled from source. Do not alter the binary in any way as that will be grounds for disqualification.
3. For this day of competition focus on EC2 as your compute resource. Do not use ECS or Fargate.
4. The root serve application can handle about 5 connections before starting to get slow. Be careful about overloading and watch for HTTP 5XX responses from the server when the queue fills.
5. The server applications are x86 statically linked, unstripped ELF executables found via an Drive URL provided during the test project.
6. The ROOT server in this version has many more requirements in order to run. In addition to the baseline requirements listed below there are additional service requirements outlined below.
   1. It must have permissions to listen on the TCP port defined (default port 80)
   2. It must have a configuration file supplied as "server.ini" (an example is provided below)
   3. It must have access to a running MySQL server.
   4. Example table definitions as well as default content for these tables can be found via the database.sql
   5. The server must be able to write to a log file. The file is named "log.txt" and located in the directory supplied in the server configuration file.
7. The LOOKUP server in this version has many more requirements in order to run. In addition to the baseline requirements listed below there are additional service requirements outlined below.
   1. It must have permissions to listen on the TCP port defined (default port 80)
   2. It must have a configuration file supplied as "main.ini" (an example is provided below)
   3. The server must be able to write to a log file. The file is named "log.txt" and located in the directory
8. The base OS that has been chosen is Amazon Linux Version 1 (https://aws.amazon.com/amazon-linux-ami/). This distribution was selected for its broad industry support, stability, availability of support and excellent integration with AWS.
9. Use of the AWS CLI (Command Line Interface) can be very helpful. Information on installing this tool locally can be found at http://docs.aws.amazon.com/cli/latest/userguide/installing.html. The AWS CLI is already installed on Amazon Linux EC2 instances.

**Architecture**



The above example illustrates one possible architectural design for the deployment of the application. This shows all required segments needed to server requests.

**Service Details**

**Overview**

Every time a request is sent to a server that you have deployed, the process must generate a response to send out (very slow). However, once you have answered a response, the data necessary to respond is stored in a database. If you received a request and have already answered it previously, you will not have to create a new answer, you will already have the answer that your server will respond with. If your server is using a centralized solution for each of these technologies, then it does not matter which instance responded to the request, all of the instances will have access to the answer. If however, you have a database server deployed on each instance, the answers will not be shared and each server will end up having to create a response for each request.