1.2 ELB

What does the “user data” script given above do? How?

```
 #!/bin/bash
  yum update -y
  yum install httpd -y
  curl http://169.254.169.254/latest/meta-data/public-hostname
>/var/www/html/index.html
  service httpd start
  chkconfig httpd on
```

The first line #!/bin/bash specifies that the file should be considered as a script and should always be run with bash. The second and third lines are for the sake of updating all packages with the yum as well as installing the httpd package. The HTTP Daemon or htddp for short is a software background program that runs in conjunction with the web server is responsible to answer incoming service requests. The requests are answered automatically and send data over the Internet using HTTP. The curl command is used to transfer data from or to a server using one of the supported protocols. Here the curl command is used to check the URL we have provided. The service httpd start command simply starts the Apache web server and turns on a service for the default run levels.
To conclude the script installs the necessary packages for an HTTP daemon and with the curl command we send a GET request to the provided URL and receive the document we asked for. After that, we start the httpd service and turn it on.

3. ECR

Please write a short essay on what steps you need to take to deploy your “web application” (currently only serving a single static web page) using Amazon ECS and AWS Fargate. Try to imagine that you’re working for a hot startup company and your application might suddenly have lots of traffic.

How are ECS and Fargate different from EC2 and ELB with AutoScaling? What would affect your decision in choosing between them?

In order to deploy a web application with ECS and Fargate, there are 4 steps in total.
The first step is to host the static content of our website on Amazon S3. S3 is a object storage service that is highly durable and available where we can transfer stored objects via HTTP. For that, you create a S3 bucket and enable the bucket to be used for static website hosting. In the second step, we will build a dynamic website. We want to create new microservices that are hosted using AWS Fargate. The reason for that is to integrate an application backend to our website. The benefit of using Fargate is that it enables us to run containers without having to worry about managing EC2 instances. It offers flexibility and automatically handles the provisioning and scaling of servers. We have to create a Docker container image that contains the code and configuration needed to run the websites’ backend as a microservice API. The Docker container image is pushed to the Amazon ECR

(Elastic Container Registry) and is available to be pulled from there when we create our service.
In the third step, we add a database with fast performance and ditch the use of static JSON files for the sake of future extensibility and scalability. We use tables created in Amazons Dynamo DB and manage it by a NoSQL database service. Now, we get to a more user-specific feature in which we enable user registration for the website. The tools needed for this are AWS Cognition and Amazon API Gateway. Registration and authentication of website users are handled by the use of User Pools. To ensure that only registered users that are authorized to do specific tasks, a REST API is deployed.

The general idea of using ECS or Fargate is to make management of EC2 instances easier. Management of EC2 instances in a cluster can be time-consuming and hard to get right. ECS makes this easier by automatically managing scaling, provisioning, and patching. AWS fargate enables you to focus on designing and building the application and lets you run containers without the hassle of managing them. Amazon ECS can be launched with Fargate type and requires the application to be packaged into containers and specify the CPU and memory requirements (maximum and minimum resources) as well as networking and IAM policies. One property of Fargate is its elasticity, where it seamlessly scales the resources up or down depending on the current usage. This means that for a hot startup, spikes in traffic can be dealt with automatically.

The difference between the different launch types of ECS is that Fargate launch types manage EC2 instances automatically, while the EC2 launch type gives you a finer granular control about the parameters of the system. The responsibility for provisioning, patching, and scaling of server cluster remains, but ECS finds the best server for a container to run on. You can decide the type of server that you want to use. This means that if you know what to do you can choose the preferred configuration you want to use and cut on cost. The Fargate route is easier and much better when launching a lot of EC2 instances, but is more expensive or result in you overpaying as opposed to the EC2 launch type option. So cost and scale play a massive role in choosing which option to use.

4. Lightsail

Familiarize yourself with AWS Lightsail. Write an essay on why you think Lightsail (as a product) exists. Give a short description on how you think it has been built (using aws products) and what pros/cons it has for different use cases.

Amazon Lightsail is a cloud service offered by Amazon and offers cloud computing and memory capabilities. The specified target for this service is new cloud users that have no to little experience in cloud services. What Lightsail basically does is to use the computing power from an EC2 instance and repackaging it to make it easier for inexperienced customers to use. AWS designed the Lightsail service in a way that is much simpler to understand and to use. The focus on this service is on its simplicity. The customer does not need to know anything about infrastructure or the backend powering the application, but only the implementation of the application. Lightsail launches virtual private servers, which are basically VMs with individual operating systems with restricted access to physical server

resources. The customer is forced to pay on a monthly subscription basis and can choose between different plans with different specifications. This is different from the go as you pay system of EC2s.
The target audience for this service are young startups and programmers that have no experience with handling and managing server infrastructure that hope to deploy a web service or application as easy and fast as possible.
There as massive cons when compared to EC2. The first being that servers are not as customizable and the selection is poor. The “more” complicated alternative has much better value when it comes to the cost compared to the specifications you get, but comes with the loss of the “simplicity” of the Lightsail service. I believe that using EC2 over Lightsail is much more economically advantageous as the resources that are needed to learn how to deploy EC2 instances and manage servers is openly available and the “go as you pay” system is much cheaper compared to a monthly subscription with poor feature selection. The reason for the Lightsail service to existing is to remove the hassle of handling infrastructure and server management of a new startup that has no experience in this field.