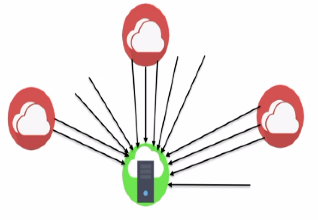
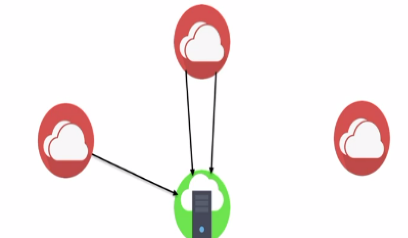
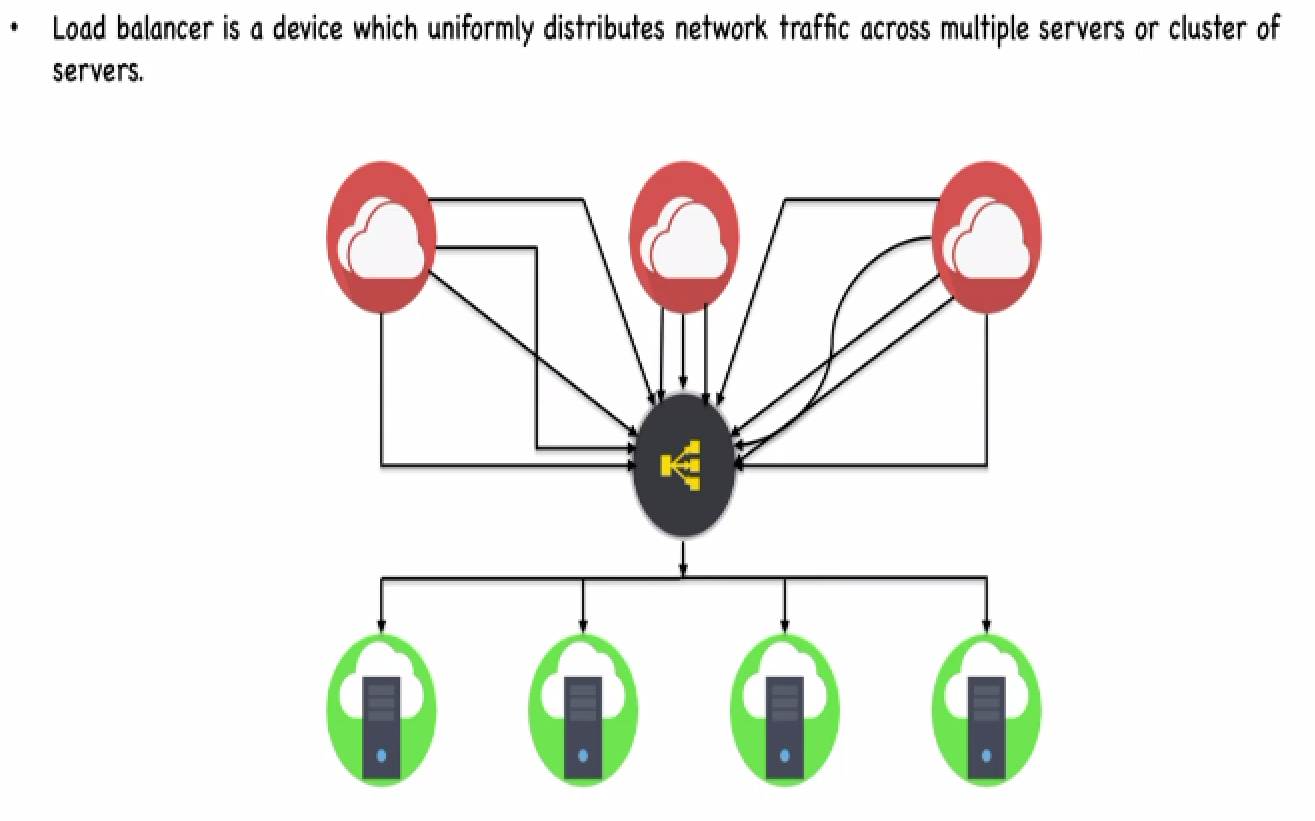
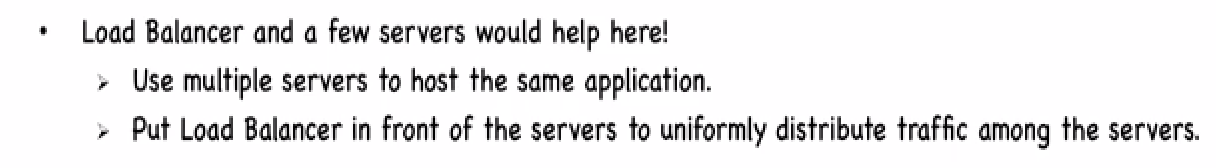
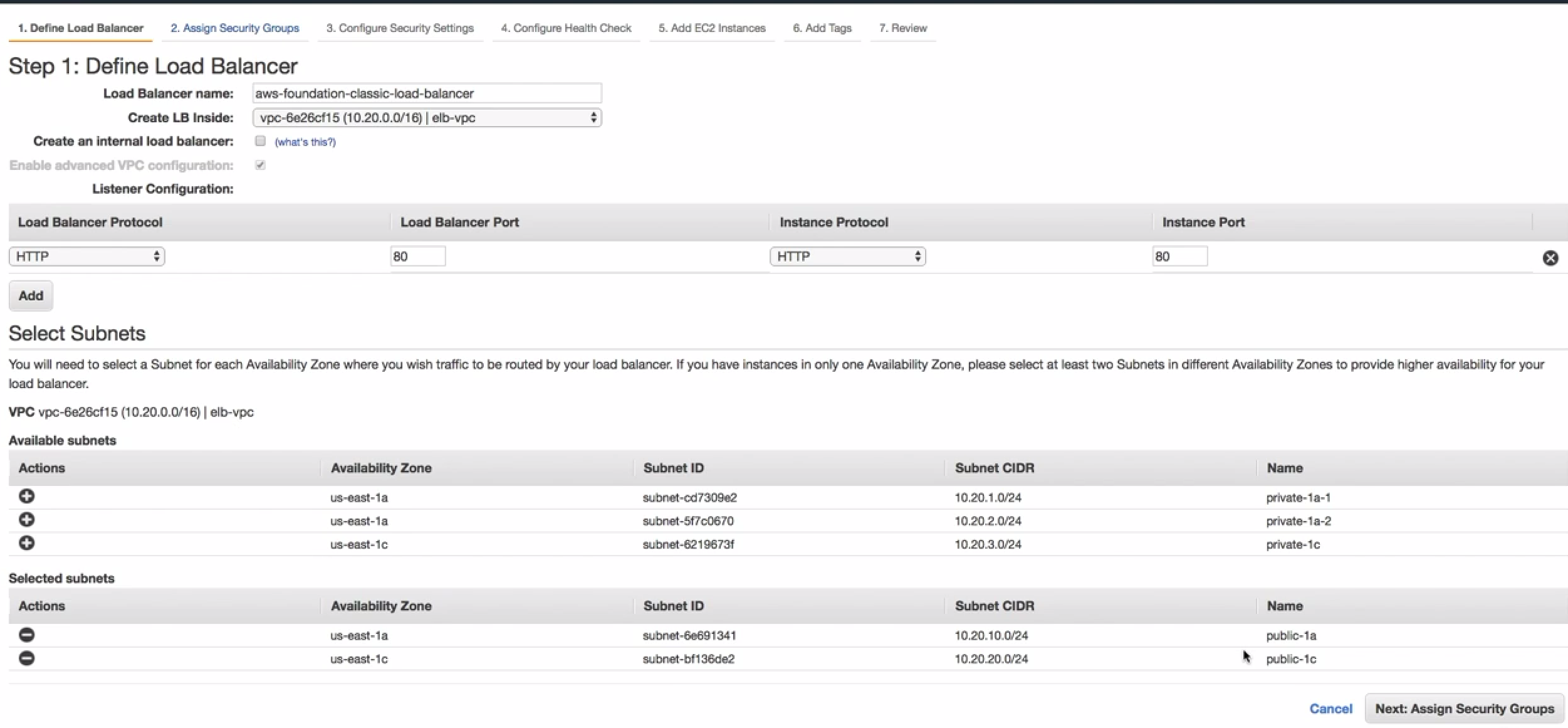
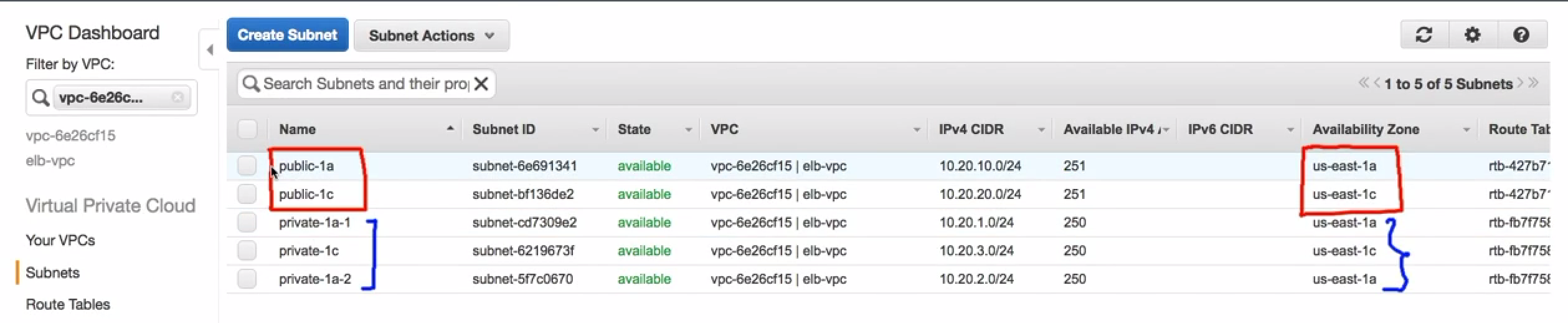
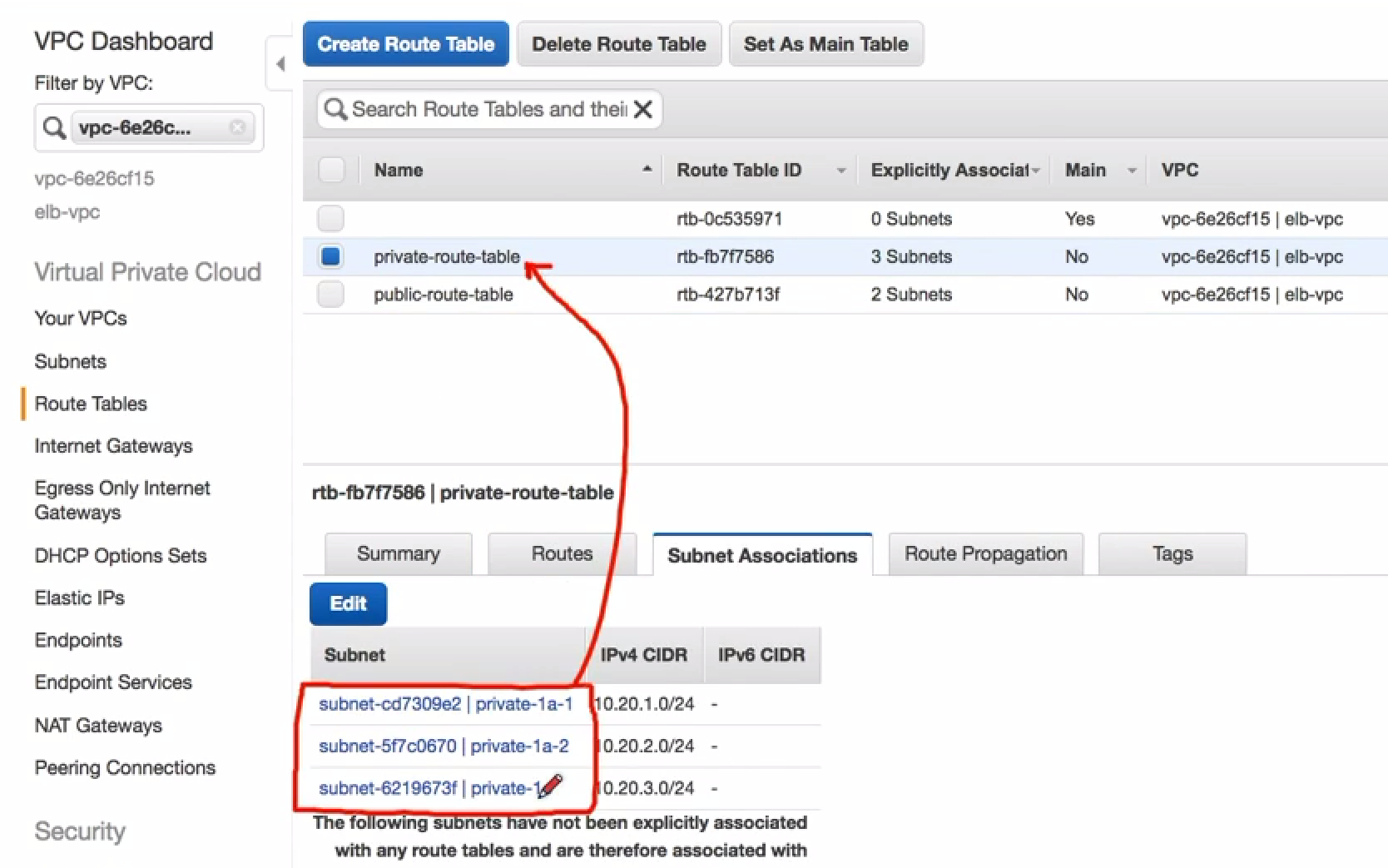
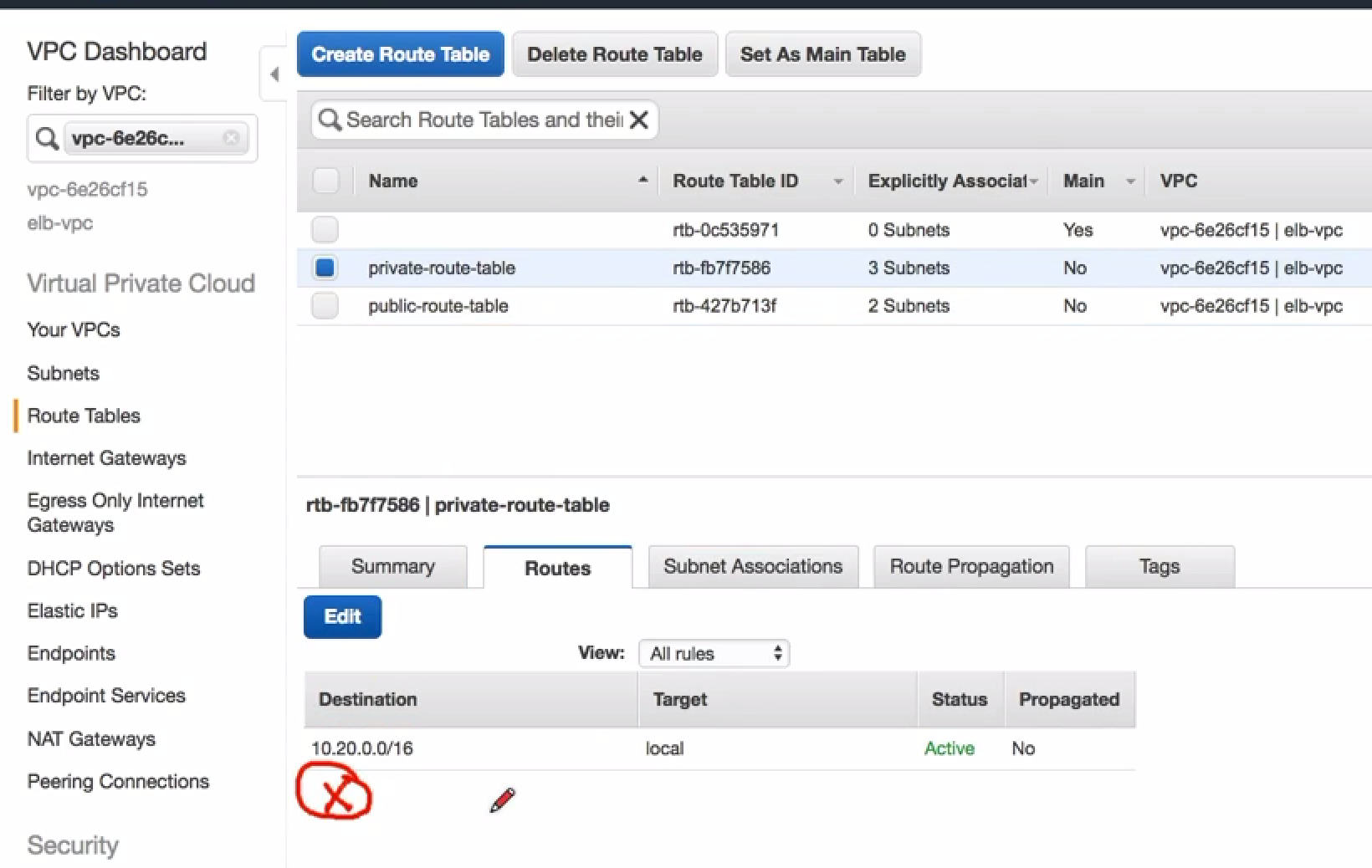
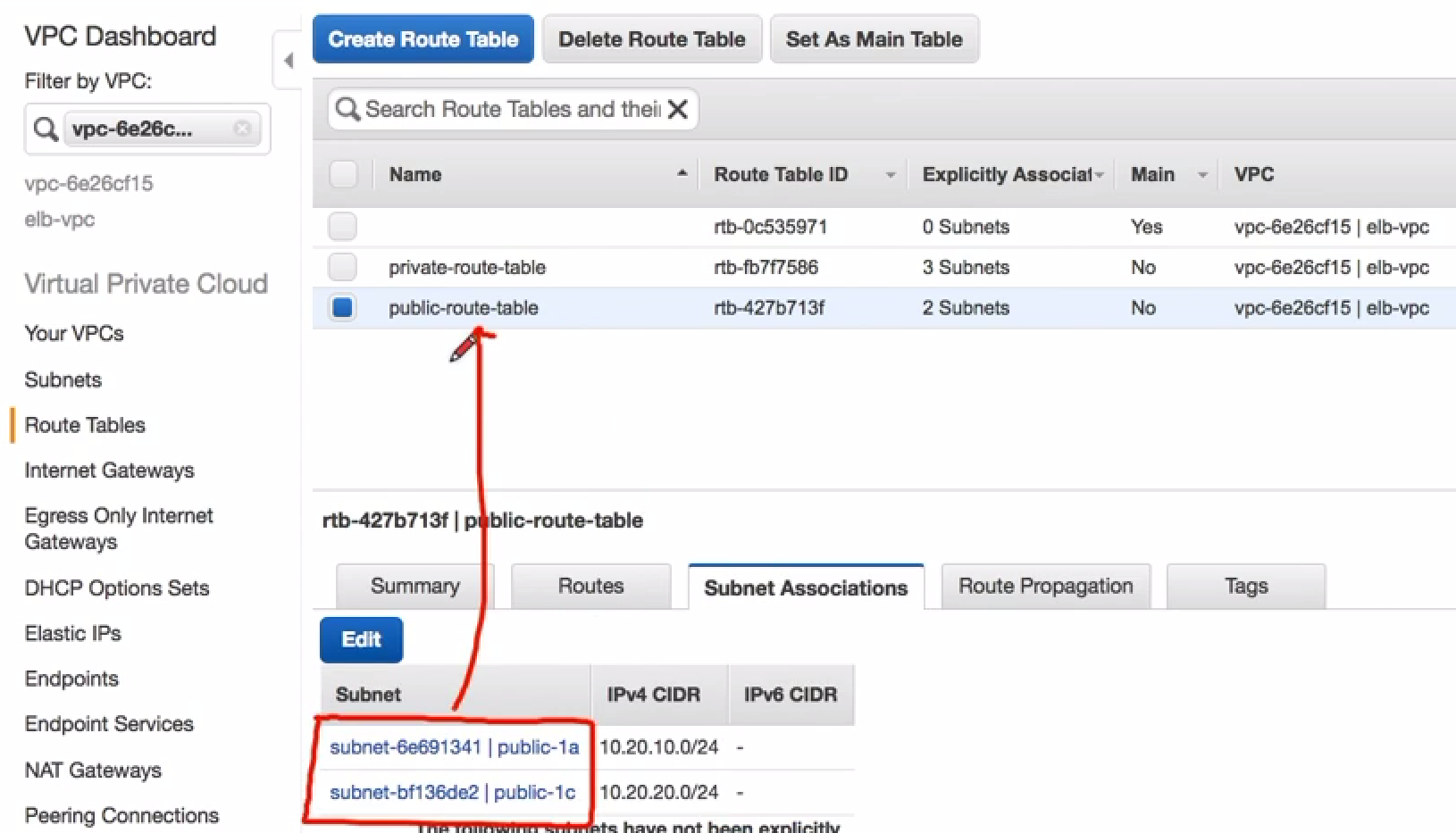
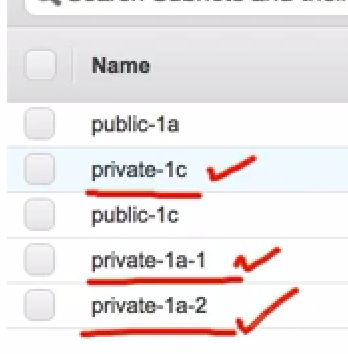
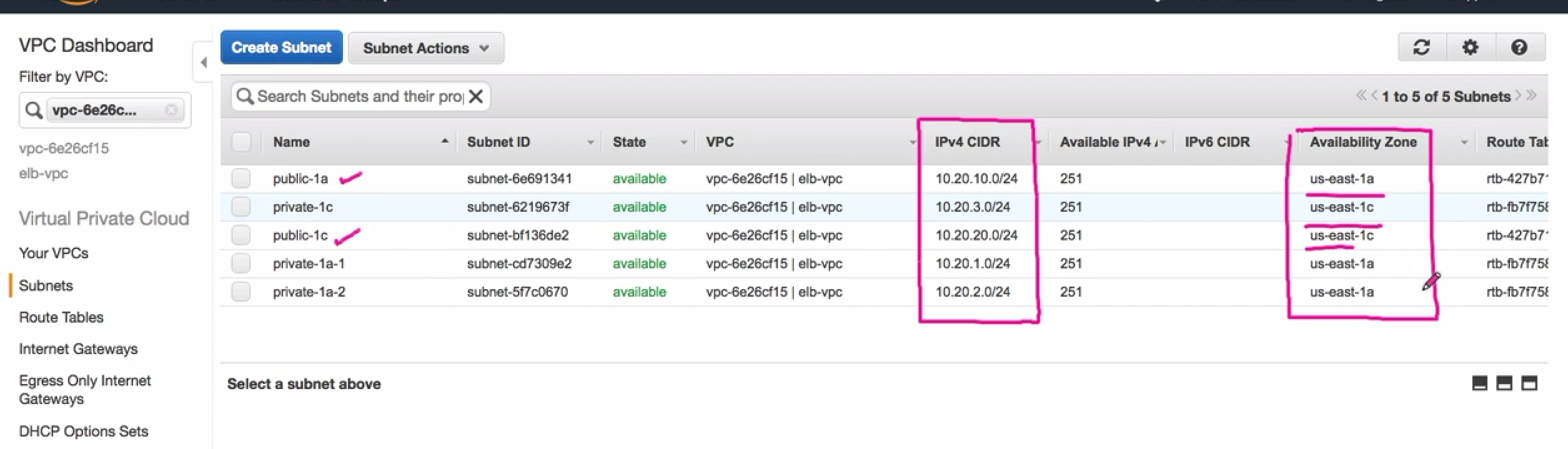
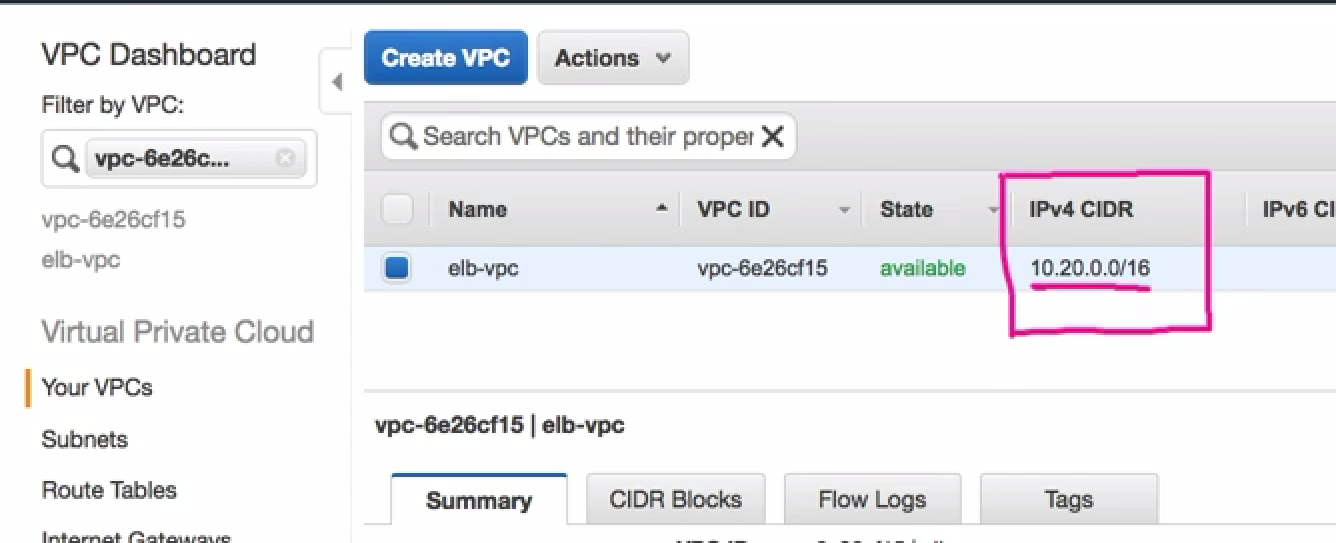
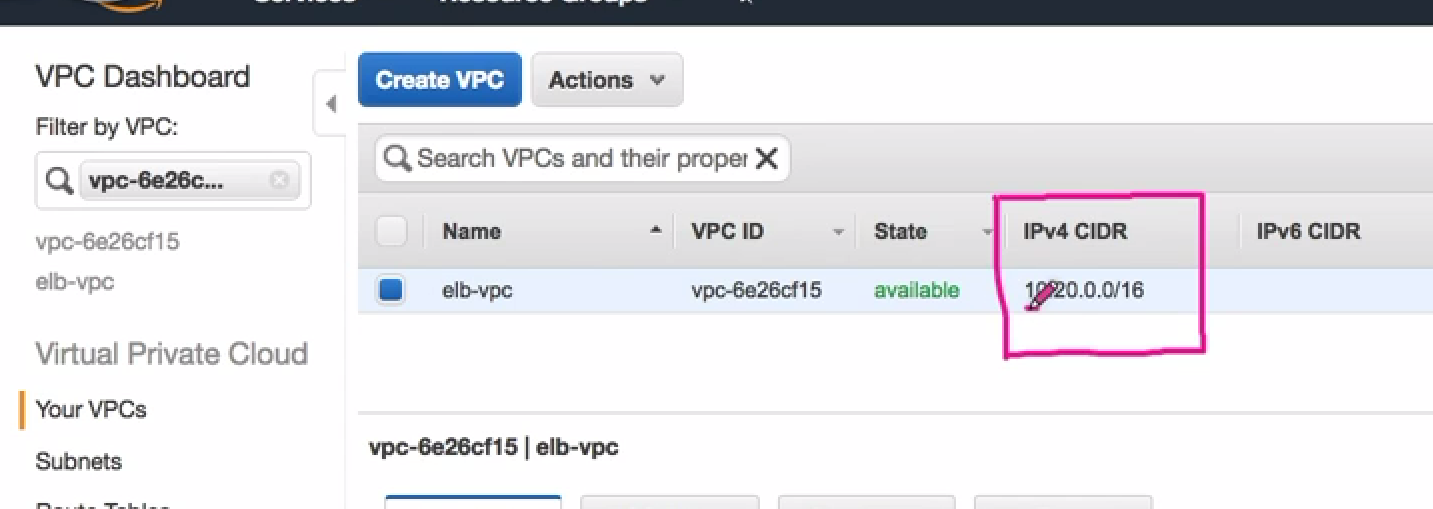
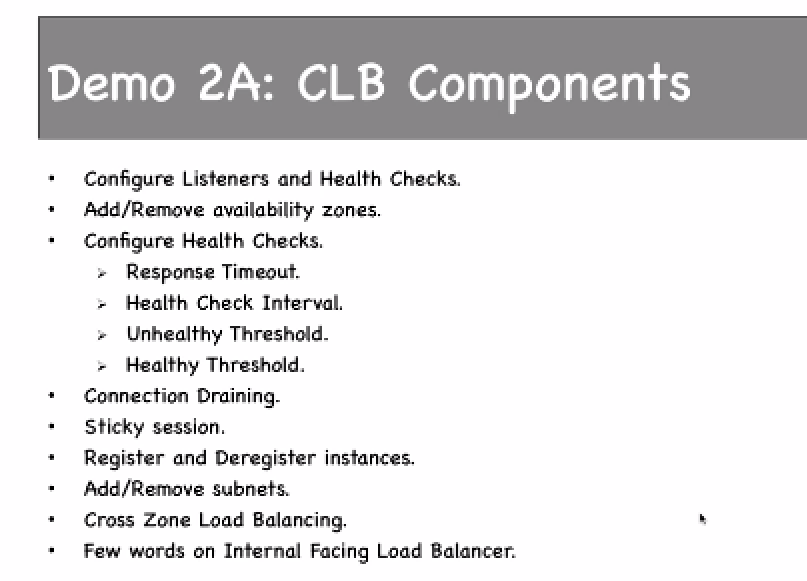
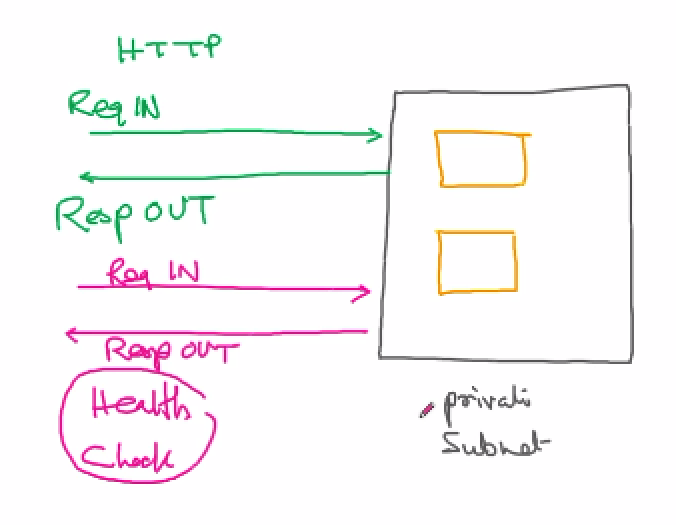
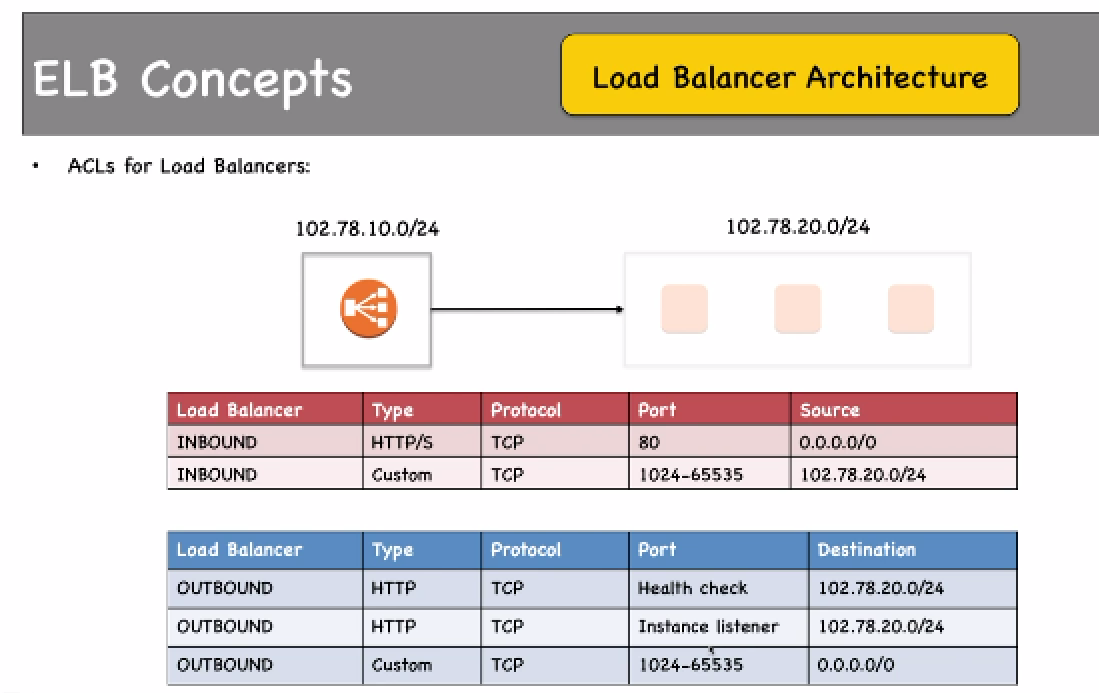
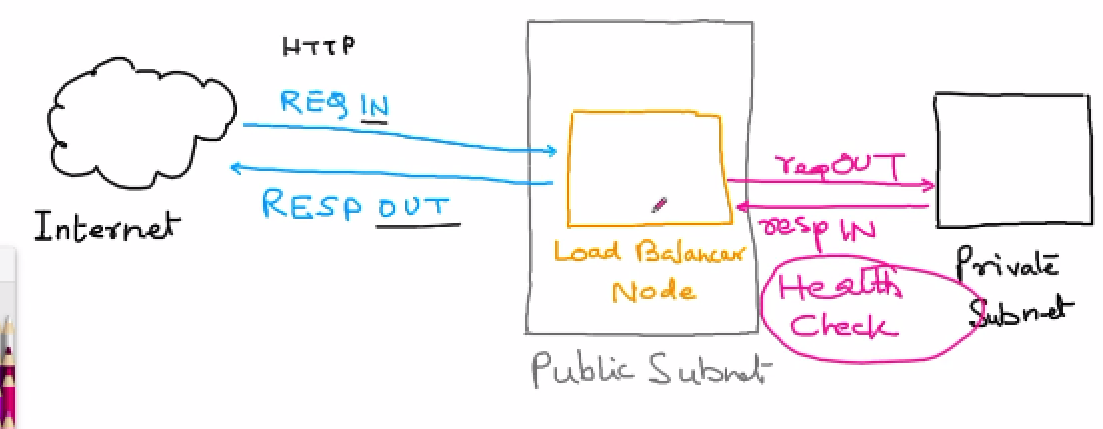
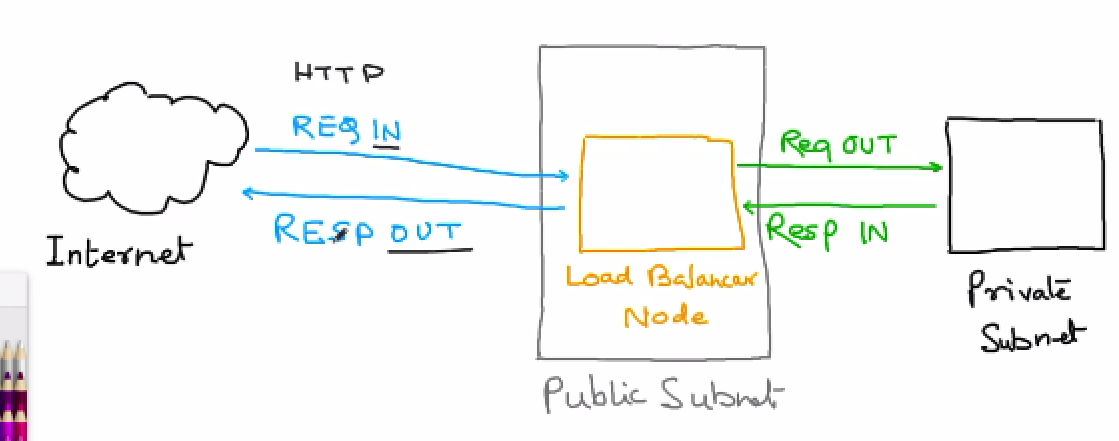
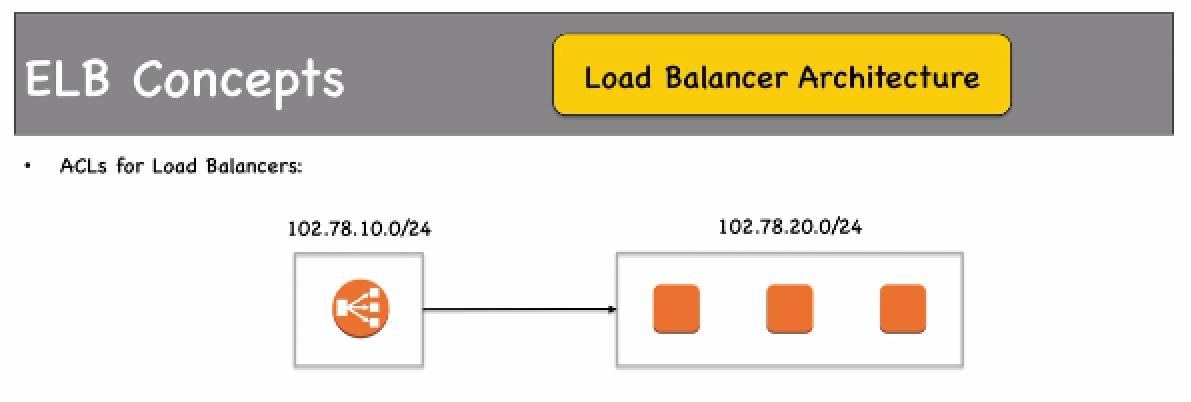
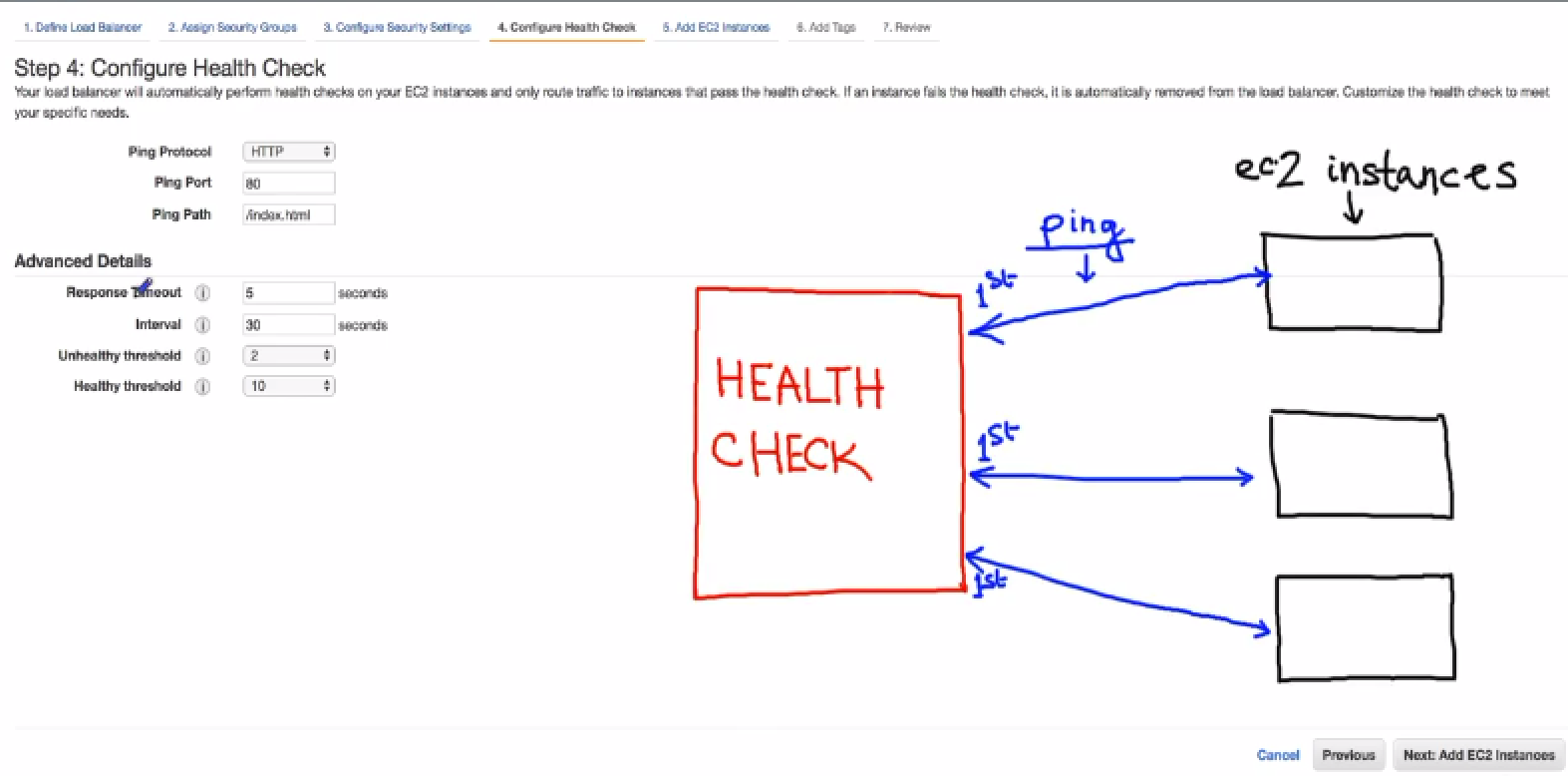
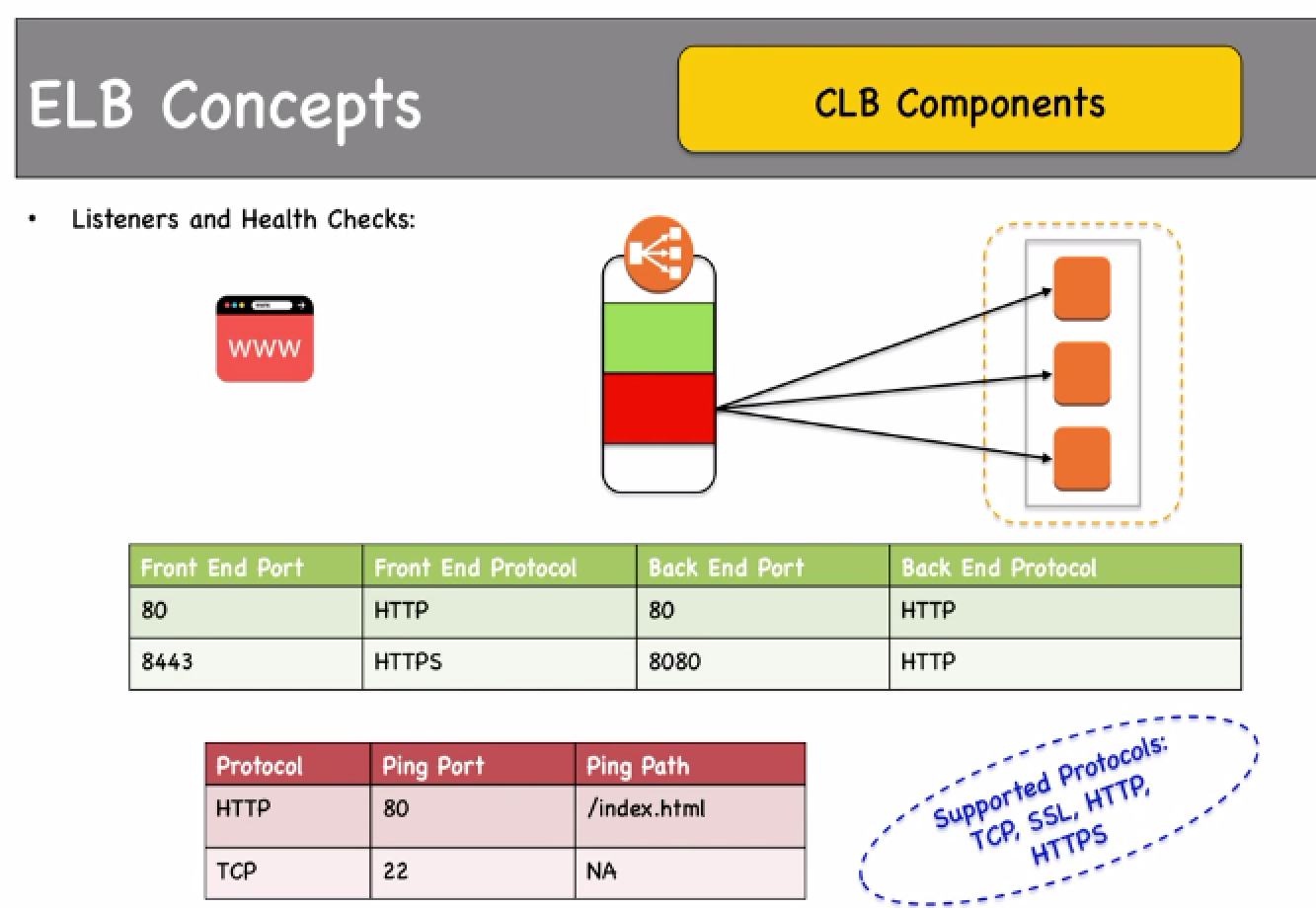
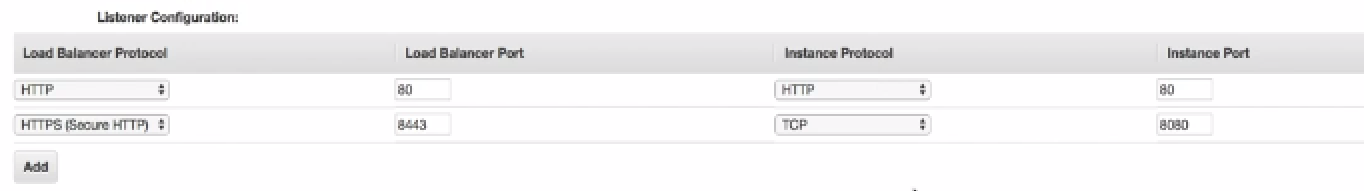
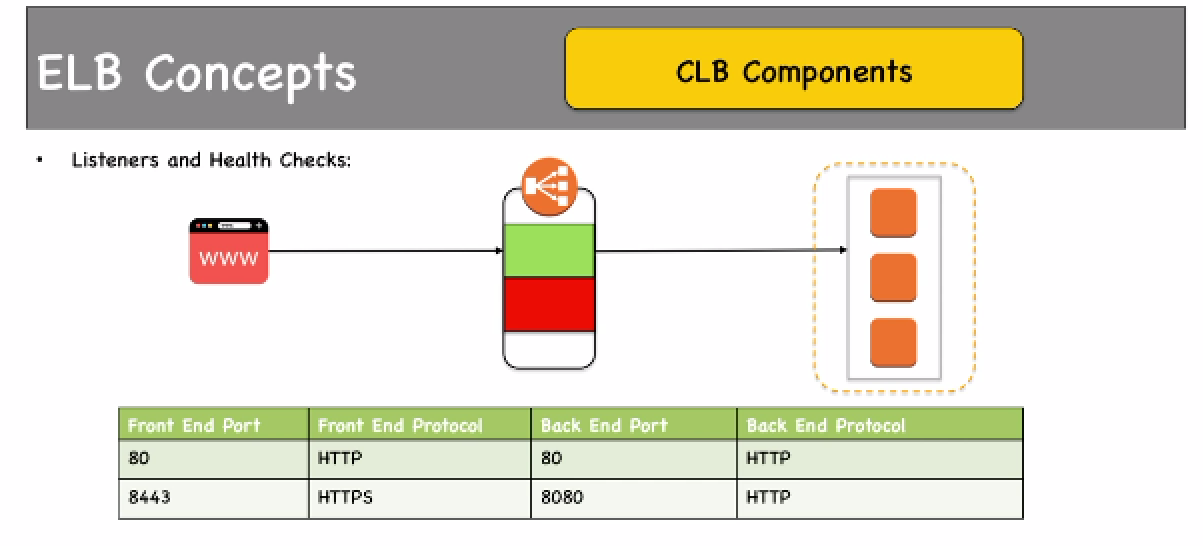
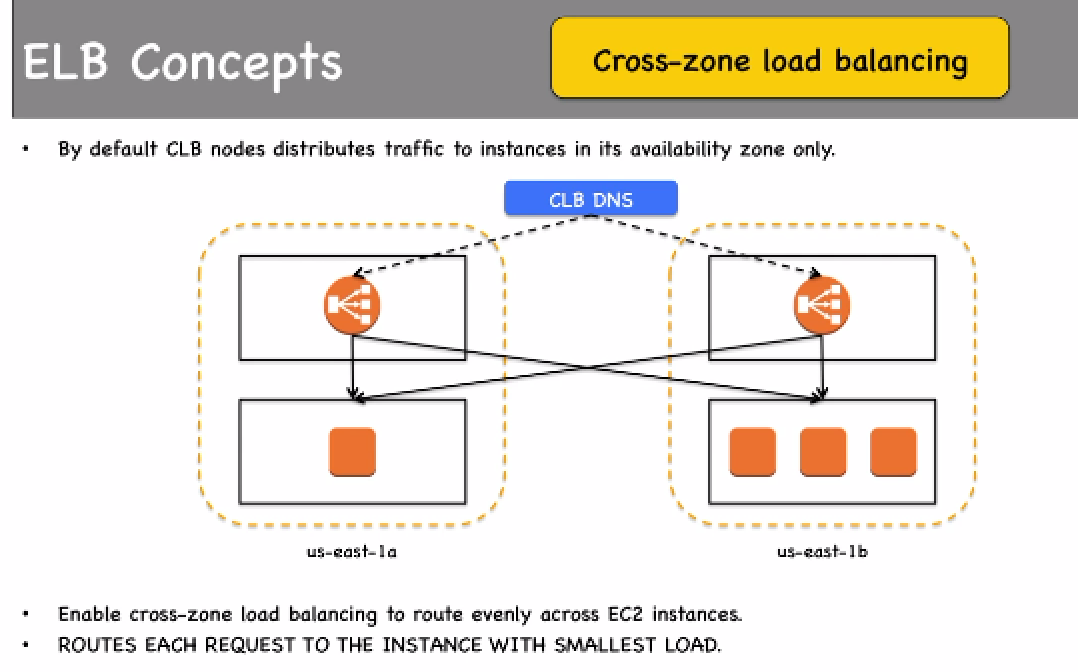
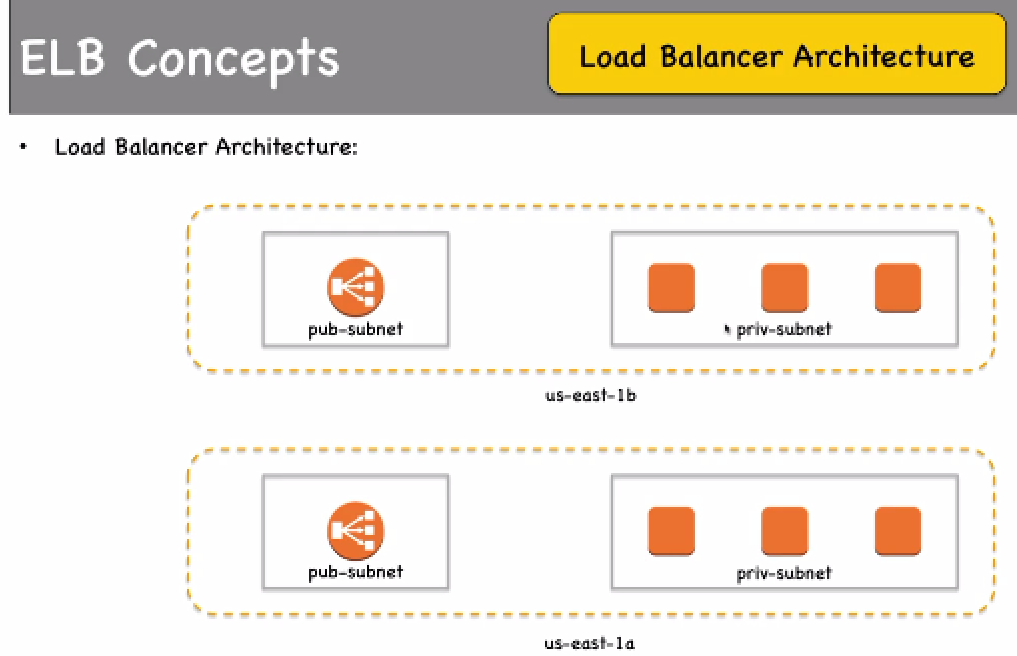
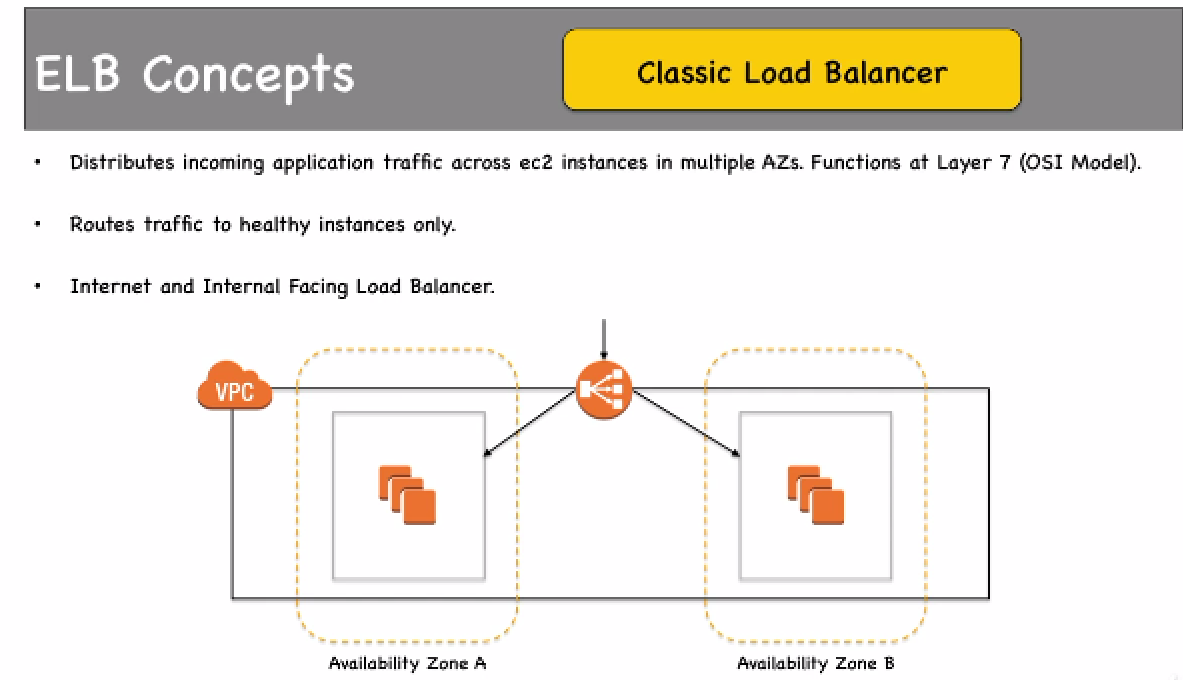
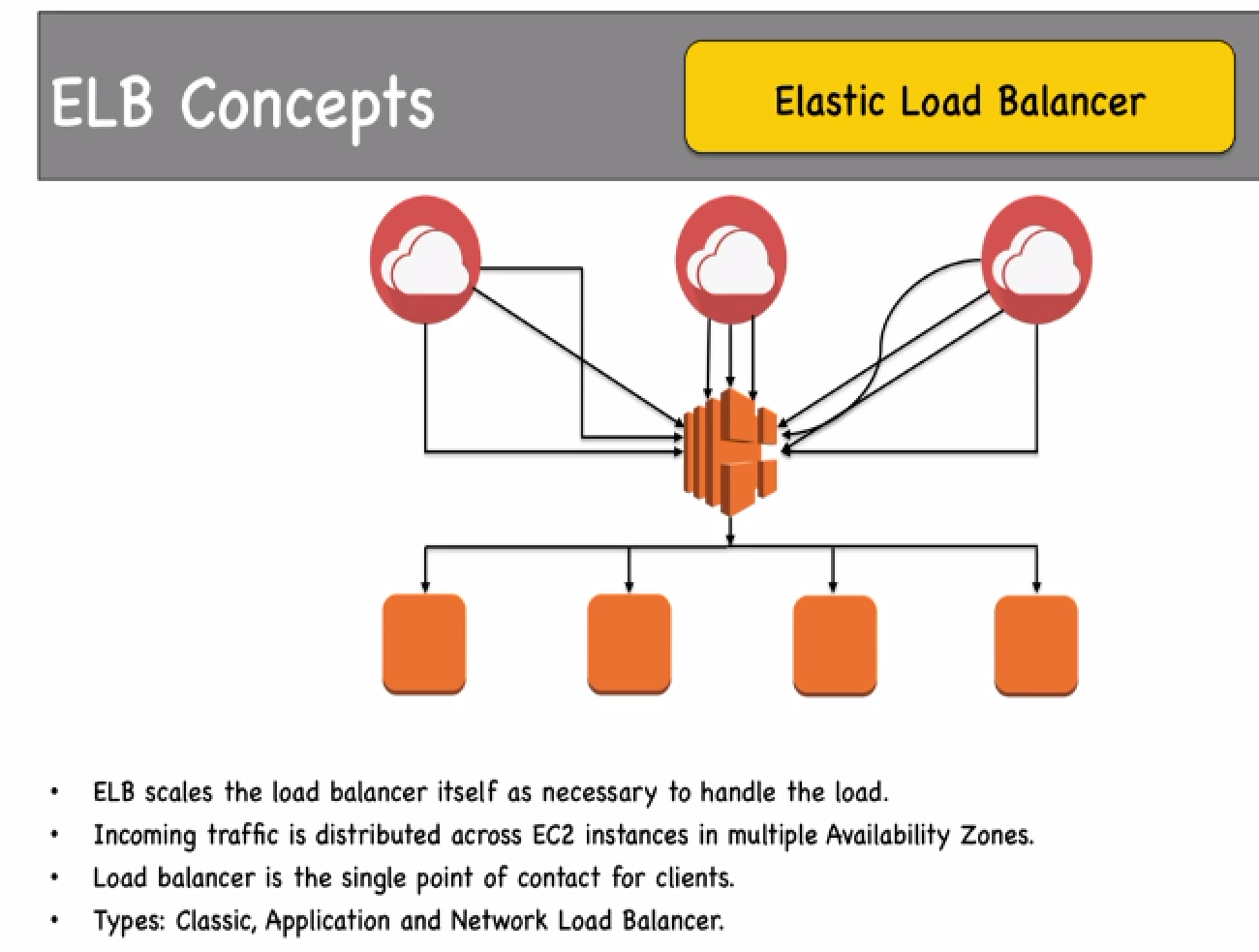


**PRE-ELB:**

Let say a server is hosting online retail webage(amazon.in).This is only 1 server hosting this website.

Users all around world. When Amazon announces big discounts... Server can’t handle that load and it crashes... 

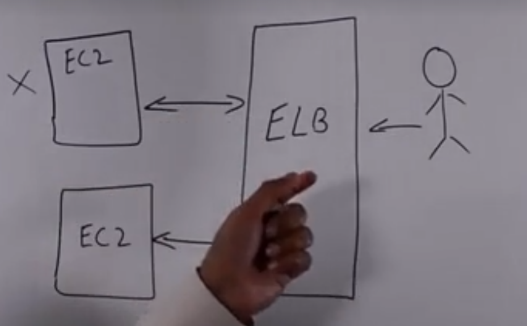


**Why ELB**

When load increases, We have options to:

Increase instance type (Upgrade cpu, RAM etc…).

If we upgrade instance, and instance goes completely offline, users cannot access webapp.



**But for HA** we can have ELB..

So what can be done is: We can have ELB infornt of our EC2 isntances, and other EC2 instance which can share the load of fusers.

Also, instead of accessing EC2 instance directly, users access ELB. ELB have separate DNS name.

ELB distributed traffic evenly across EC2.

ELB is a HA system.It makes sure it have multiple instances of ELB.This happens in background.

**HealthChecks:**

To make sure EC2 available or not by ELB.

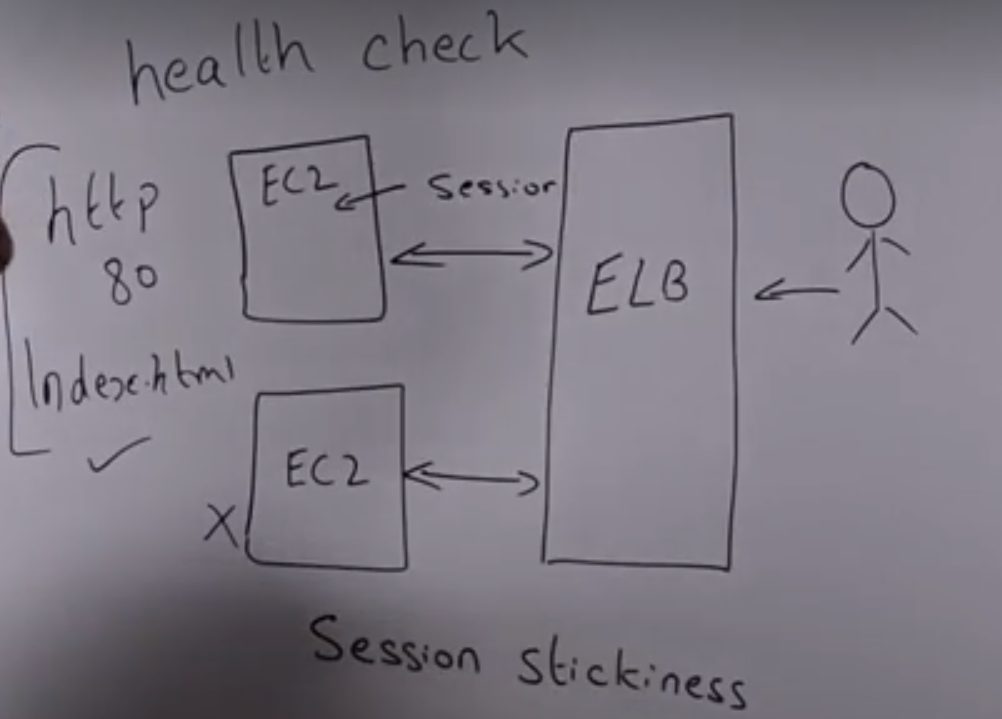
If EC2 crashes, it directs traffic to only healthy instance.

We can define healthcheck when creating ELB.We can change later on as well.

ELB checks if webapp is running on EC2 isntance on protocal http, port 80..And existence of html documents.

**Session stickness:**

When User type address goes to ELB. ELB have session established for this usr to EC2 isntance1.

With Session stickness, When same user comes to ELB gain, it will always redirect to EC2 isntance-1.

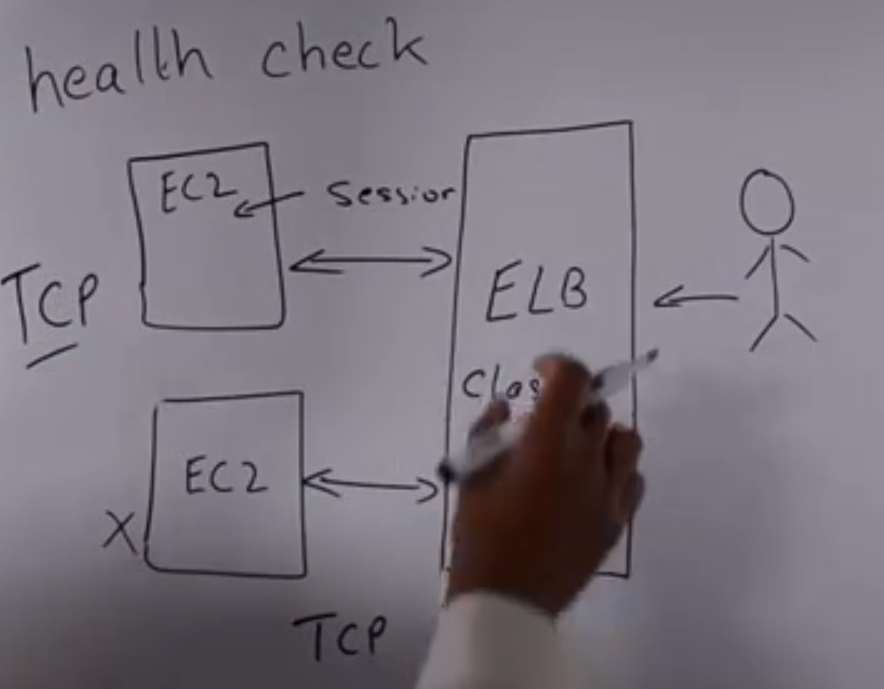
It basically locks user to that EC2 instance..**Requiremnt depends on your application completely..**

**ELB Types**

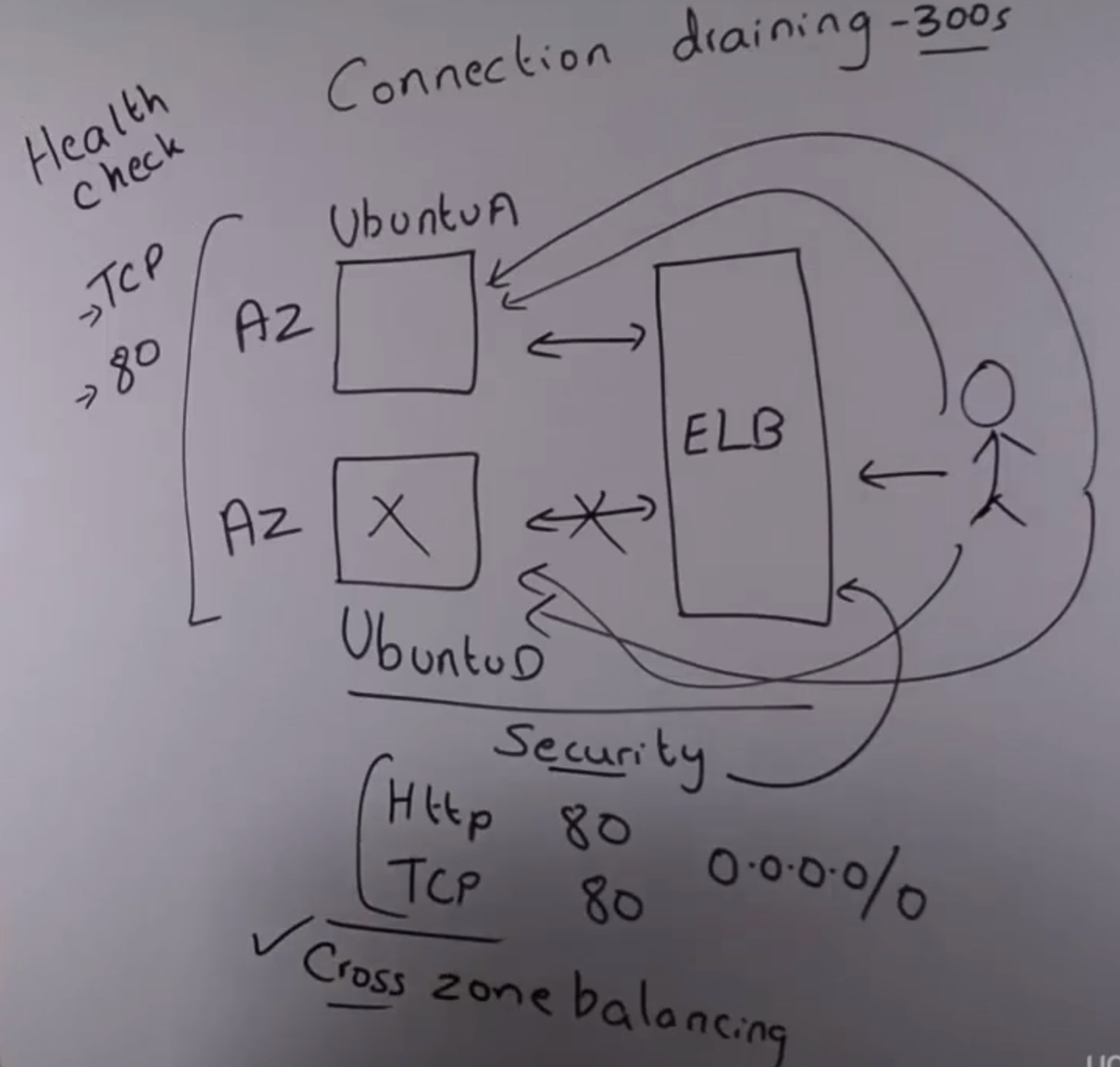
**Classic, APP(Modern day APP, Microservices)**

Classic (http, https, and also TCP).If we have Application running on EC2 instance is not WEBAPP.

It is basically an app. That uses TCP for communication.Then,we can ue Classic…ALB dosnt support TCP.



User will come to ELB. ELB will ccommunicate with each EC2 instance using TCP/IP protocol.ELB also does health checks son TCP protocol.



Ubuntu-A , D hostsed in different AZ with nginx running.

**Security Group:** We use security group attached to Ubuntu-A,D instances…Which here allows http, tcp ports 80 from any source. This ensures not only user can communicate directly with EC2 instances(Outside users can communicate with Ubuntu-A.D instances…Basically when user connects with ELB, ELB redirect user to connect with EC2 instances ..Ubuntu- A/D)..But also ELB with Ec2 isntance(ELB to connect with backend instances).

We also attach same security group to ELB as well.

**HEalthchecK:** Ensures EC2 health checks by ELB..We are going to do on TCP port 80.

**Connection Draining:**

Lets say one of Ec2 instance goes unhealthy, but there are already some requests which are in pipeline or serviced by that EC2 from user. What ELB does is that, it will be using Connection drain out(default 300sec).

During this 300 sec,ELB will give eEc2 server try(process inflight request)..After that it will stop ending requests to that EC2 isntance

**Cross-Zone LB(checkmark):**

ELB ensures connection is distributed evenly across EC2.

Sometimes what happens is that: When ELB sends req. to one of EC2 servers,the time to live for request from user browser. If server already serviced request to user, the user sends request again, request will go back to same server(A)..In that time, as server is overloaded , ELB needs to send request to server(D) now.

Time to live is still there.Ideally request goest server(A).When we enable cross-zone LB,ELB ensures even though there is time period for this request, it will send request to secondary server(D)