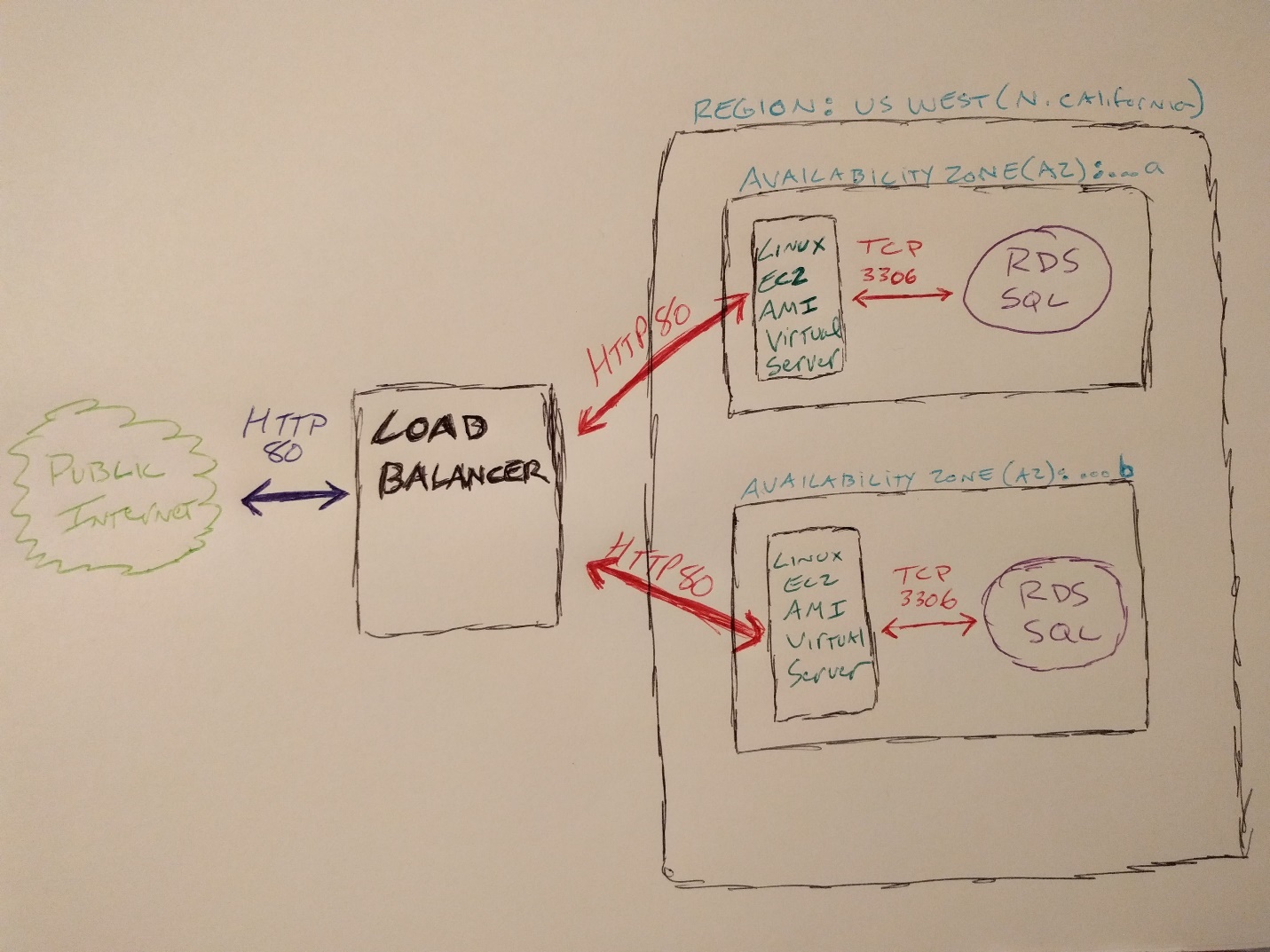
**Overview of Load Balancers**

We have a public internet and they will interface with the LB and a LB is basically a piece of software which distributes work in a round robin fashion you could think of it as your boss, requests come in and then the LB just round robin between whatever servers are available, will pass those requests to those servers and then LB will be connected to the servers, and only the LB on the servers can talk here. And the servers can talk with your databases with your SQL DB and each of these can be in different availability zones and remember each region can have many different availability zones.



Lets implement our LB. the first step in implementing our LB is to create a **Security Groups** for a different types of machines will be running.

One of the security groups we need to create is a LB Security Group and that LB Security Group will accept traffic from anywhere on TCP port 80 or HTTP traffic. Another Security Group we need to create is for our Web server and our Database and that security Group will call that the Web Tier Security Group, and for that Security Group we want it to only accept traffic if the traffic is coming from the LB on TCP port 80, so HTTP traffic from the LB will accept that traffic, we will accept traffic that is coming from us when it is SSH, so we can restrict that to our own IP address if we want or we are always of course providing our PIN file or private key. So that is another way of security check to make sure nobody’s SSH. So LB traffic except SSH traffic from our own IP from ourselves will accept and then internally any traffic which is occurring inside that web tier security groups, so if this has the web security group applied and that has the web tier security group applied will say we will allow the communication between machines that also have the web to your security group applied. So these are the rules we are going to set.

**Create security groups**

A security group acts as a virtual \*firewall\* that controls the traffic for one or more instances.

When you launch an instance, you associate one or more security groups with the instance.

You add rules to each security group that allow traffic to or from its associated instances.

You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

*## ELB (load balancer) security group*

*1. add this rule*

*- HTTP TCP 80 Anywhere*

*1. copy \*Group ID\**

*## Web tier security group*

*1. add these rules*

*- HTTP TCP 80 Custom IP ```<load-balancer-sg Group ID>```*

*- SSH TCP 22 My IP*

*1. copy \*\*Group ID\*\**

*1. add this rule*

*- MySql TCP 3306 Custom IP ```<web-servers-sg Group ID>```*

*# Load balancer*

*1. EC2 / Load balancers / Create load balancer*

*- application or classic*

*- name: web-elb*

*- http & https*

*- default VPC*

*- add two subnets*

*1. configure security groups*

*- choose "load-balancer" security group which we just setup*

*1. configure routing*

*- target group: web-servers-tg1*

*- ping path: /ping*

*- allows us to define a "healthy" web server*

*- load balancer will only forward to healthy web servers*

*1. register targets*

*1. create*

So we are going to go over to AWS and inside the AWS will go to ‘Services’ then go to ‘Security Group’ and we are going to create a Security Group, we name it to ‘loadbalancer-sg’ and description’ This allows traffic from anywhere TCP:80’ and for outbound rules we choose the Type to be ‘HTTP’ and Destination Type is ‘Anywhere’, then after we press the Create Security Group button. This will be our LB security group.

Now we need to create another security group called Web Tier Security Group and this is for machines in a web tier. This will allow traffic from SSH and that can be custom or anywhere we can do our IP address. So in Security Group page we name it to ‘WebTier-sg’ and description to ‘something’, for inbound rules type is SSH, so here we press create since I need to get some information.

I am going to our LB Security Group and this has a security Group ID, copy that, because I only want to look up traffic from Web Tier that comes from either SSH or internally between machines that also belong to the web tier or from my LB.

So to say I only want to accept traffic from the LB I get the LB security group ID, and I am going to go to my Web Tier Security group and edit the inbound rule and add another rule for type HTTP for custom source type and paste the Security Group for source field. Then save that. So now we have 2 rules on Web Tier; I have SSH and I have from the LB traffic come from the LB.

So new we have to go to the Web Tier Security Group and copy the Group ID and go back to inbound rule of the Web Tier Security Group and press Edit and choose Type of MYSQL/Aurora which it is going to be on port 3306, and for Custom Source Type paste the Group ID for the source.

So anything that has web tier security group applied to it on TCP 3306 can communicate with other machines that have web tier applied to it. So those are my security groups that I need I am going to apply those to the web tier and to my database.

So now I go to RDS and I have my SQL database running and go to instances then I go to instance actions and I will go to modify and then I have security group and MYSQL server is going to be in the Web tier and leave the option publicly accessible to Yes for now. So we press the continue and modify DB instance.

Then come back to Security Group page, have the Security groups because I just need LB web tier or default security group which was created when we launched, we could see the rules on that MYSQL port 3306 from my IP address is where that is being allowed, but we no longer need that, so we are going to delete that Security Group.

So next step we are going to create a LB.

We are going to get our LB running and to get our LB running we are going to AWS console, then we go to EC2, in there we click on LB . Then press on ‘Create Load Balancer’. There is a ‘classic LB’ and the ‘Application LB’ . Application LB is the preferred LB for HTTP/HTTPS.

So we will choose the ALB and press continue, give it a name ‘standard-lb’ and the schema is going to be internet-facing. Listeners, is the process that checks for connection requests using the protocol and port that you can configured, Load Balancer Protocol is HTTP and Port is 80. That is basically where we want ed to be listening, and availability zones we wanted to be working in us-west-1a and us-west-1b added both of those. Next I am going to configure Security settings and then just keep going after that saying if your traffic to LB needs to be secure use HTTPS .

And I want to add the LB security group, select that one, and then on Configure Routing page, just give this target group are going to be what machines are we are targeting and we will add them to target group. Give a name Stanadrad-tg. Then we have to give it a path in our code to check whether or not that server instance is healthy. So we will write code that looks like this and we have a path ‘/ping’, because I just put in this path field ‘/ping’

So LB will go to that route and ping, it needs to return just HTTP status of 200.

On the next page ‘Register Tragets’ we are going to register our target, we have this machine we will add that to registered, and so this machine is now registered to this LB, so registered instances.

Next is the review…

Then press the create a load balancer and hit close.

So now the healthy code is not running so healthy check will fail, so we have to put code in that machine to respond to that ping request.

**Implementing the LB**

To create Amazon Machine Image, so we go to ‘instances’ page there is ‘actions’ drop down list, select the ‘Image’ then select ‘create Image’ then will take us to create image form, were Instance ID is set already, we give image name to be ‘webserver’ and the image description to ‘this is the standard web server’, then leave the reset as is, then press the create image button. That will send you to another page ‘Create Image request received’ with the image ami-xxxxxx, we can close that page. So we have to wait till amazon create the image for us. When the image is ready we can see that, bt clicking on AMIs and we will see the image webserver is created with the id img-xxxxxx. We change the AZ on this image in different geographical region and be a different from the first instance EC2. So how we do that, we go to AMIs and webserver and select it, and doing the right click on it and go to ‘Lunach’ it will go to step 2 ‘Choose instance type’ we by pass on it ang to step 3 ‘configure instance’ in this page we have our Network set already and we want our subnet to be in a different AZ ex: us-west-1a, then pass to step 4 then go to step 5 add Tags and select a name like ‘web-server-002’. Then go to step 6 configure security group and select the ‘select an existing security group’ and on the button we select the ‘webtier-sg’, then review it and then hit the launch.

So that will spinning up another image and when we go to ‘instances’ we can see the 2 instances are launched.

Now we need to add the new instance to our ‘Target Group’. So go to ‘Target Groups’ , then we go to ‘Targets’ tab and then ‘Edit’ , we can see that we already have the ‘web-server-001’ added, so we want to add the ‘web-server-002’. So we select it on the button list and select the ‘Add to registered’ then click save.

So we can go to load balancer and copy the DNS name and that will be used to browser to our EC2 instances.

**Autoscaling and CloudFront**

In AWS we have many regions and within each region we have many different AZ, and you can create instances going to EC2 (Elastic Cloud Compute) and you can go to your instances and create instances of machine by choosing machine image and then go through and configure it and you could set Security Groups on your EC2.

The last thing we want to do is how to Auto scale this? Auto Scaling which as demand increases for your services you could set criteria and say when my processors hit 75% of their processing I want you to scale up some new machines or when the machines go below 50% of their processing takes some of those off line and or whatever metrics you want to set for when to scale up or when to scale down.

Also you should also be aware that on EC2 there is cloudFront.

# Create an AMI (Amazon Machine Image)

1. EC2 / Instances / right-click instance / create image

- image name: web-architecture-2019-10-31

- description: web server 2019 October 31

- no reboot: unchecked

- allowing your instance to reboot gives a better image

1. create image

## Launch a new instance of your AMI in a new availability zone (AZ)

1. what AZ is your current instance running in?

- EC2 / instances / look at the availability zone and make note of it

1. launch a new instance from your AMI

- EC2 / AMIs / right click / launch / next: configure

1. subnet: ```<choose a different AZ>``` / next: storage / next

1. tag

- value: web-server-0002

1. security group

- choose the "web-tier" security group we created

1. launch

- specify "key pair" we want the instance to use

1. launch instance

## Add new EC2 instance to load balancer's target group

1. add the new instance to the target group

1. enter load balancer DNS into a browser to see your load balancer in action

- refresh your browser to see the switching between web-servers-sg

# Create auto scaling

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited both to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

## Configure auto scaling

1. EC2 / autoscaling / launch configuration

1. create auto scaling group / create launch configuration

1. My AMIs / choose your AMI

- my image name was "web-architecture-2019-10-31"

- next / next

1. configure details

- name: \*\*auto-scale-config-2019-10-31\*\*

- next / next

1. configure security group

- select an existing security group / select the "web-servers-sg" security group

- next / next / create launch configuration

- choose an existing key pair / create launch configuration

## Create auto scaling group

1. Configure auto scaling group

- name: \*\*auto-scale-group-2019-10-31\*\*

- group size: this is the minimum number of instances we'll always be running

- network: default vpc

- subnet: choose the availability zones (AZs) into which you've launched instances

- advanced details

- load balancing: check "receive traffic from elastic load balancer"

- select your load balancer

- health check: ELB (this is what we set up)

1. configure scaling policies

- keep group at initial size

1. configure tags

- value: web-server-auto-scaled

1. create auto scaling group

1. Scaling policies

- this is where we'd add policies to say when we scale up / scale down

Change your web servers so that they both can access your MySQL database.

Show that each web server is accessing the database.

Getting the DB connected and running and be able to see that I am able access the same DB from 2 different instances and 2 different AZ.

We are have the code , compile it and move the binary to the ‘web-server-001’ EC2 instance. So now we go back to AWS console, so ‘web-server-001’ can only be access by LB, so we need to create a ALB. So selecting ALB, it will take us to the step1: configure LB page , we choose the Name to be ‘loadbalancer-standard’ and then on the same page at the bottom page we select add the 2 us-west-1a, and us-west-1b AZs to the list.

Then go to step 3: configure security groups, use the existing security group ‘loadbalancer-sg’, then in step 4: configure Routing, choose the name ‘webtier-tg’and the path ‘/ping’ for health check. Next step 5: Register Tragets, so add the web-server-001 to register instances list, then review, then create. So we have create our LB.

So now we go to the instances page, and on the web-server-001 right click and select image then select create image and give the name ‘standard-server-two’, and description ‘this is the second server image’ , then press the create image. We can see the image is ready on AMIs tab.

So now we can add this image to our instances, so on the instances page select the ‘launch new instance’ on step 3: configure instance details, select subnet to us-west-1a and leave the rest as default, new step 4: add storage , the step 5: add Tags, for the name ‘web-server-002’, next step 6: configure security group, from existing security group select ‘webtier-sg’ and then press the launch.

So now our 2 instances are running, so now we need to add them to Target group, so add theweb -server-002 to the registered list and save it.