ELASTIC LOAD BALANCING

* It is used to distribute the incoming traffic across ec2 instances which are inside of an ELB.
* It sends the health check requests to instances to check the health of instances. so that it sends incoming requests only to healthy instances.
* If an instance failed, elb will automatically remove the instances from the load balancer. health checks are basedon our configuration while creating loadbalancer.
* You can add ec2 instances which are present in different availability zones for fault tolerance. if an availability zone failed, it will send requests to instance which is in different availabilityzone.
* Your load balancer acts as a single point of contact for your clients. you can addand remove instances from elb without disrupting the overall flow ofrequests.

CREATE A LOAD BALANCER

* Go to ec2, in left navigation pane, select load balancer. Click on Create loadbalancer.
* Choose classic load balancer and click next. Type a name to elb, select a vpc. Toselect subnets, click on enable advanced vpc config, and selectsubnets.
* Select two subnets, because if you have an instance in one az(subnet) but you didn't selectthatsubnet,thereforeyoucan'taddinstancetoelbwhichis inthataz(subnet).
* Bydefault,theport80isaddedtolistener.ifyouwanttoaddanotherport,clickonadd and type your port.
* Before adding the port you have to allow this port in your instance securitygroup.
* You can also add https(443) to listener for encryption. you have to create anssl certificate and security policy and add this while creating elbto usehttps.
* Afteryouaddhttpsinlistener,innextpage,selectyoursslcertificatefromacmandselect your security policy. click,Next.
* Select a securitygroup.
* In health checkpage,

**Ping path =** destination for http (or) https requests.

**Response timeout(sec) =** amount of time to wait for receiving response from health check(2-60).

**Health check interval(sec) =** amount of time between health checks(5-300).

**Healthy threshold =** number of successful health checks must clear to declare an instance healthy.

**Unhealthy threshold =** number of failed health checks must happen to declare an instance unhealthy.

* Select Your ec2 instances. add tags and clickCreate.

IDLE TIME-OUT

* Foreachrequestclientmadethroughelb,itmaintainstwoconnections.oneisfor instance and one is forclient.
* For each connection, elb maintains an time-out , if no data is sent or received fora specific time period, it closes theconnection.
* If you use http (or) https listeners you have to enable **keep-alive** option in yourinstances.
* Keep-alive enables the load balancer to use the connection for a specific number of requests with a specific period of time between each request,which reduce the cpu utilization. Set the keep-alive time greater than idletime.
* Go to ELB,

**Select your load balancer.**

* In description tab,

**Choose edit IDLETIMEOUT.**

**Value = 1-4000 seconds.**

**Click save.**

CROSS-ZONE LOAD BALANCING

* Bydefault,elbdistributestrafficequallyacrossaz.
* For ex, you have 10 instances in one AZ and 2 instances in another AZ. It will distribute the same traffic for all instances in both AZ’s.
* Topreventthisweusecross-zoneloadbalancing.itwilldistributethe trafficevenly acrossaz.
* After enabling cross-zone load balancing, elb will share the traffic based on the number of instances present in that AZ.
* Gotoelb.

**Ondescriptiontab,**

**Selectchangecross-zonesetting.**

**Choose enable.**

CONNECTION DRAINING

* By default, when an instance is unhealthy(or)de-registered, elb will closes the connection immediately. If an request came for that instance, it will notcompleted.
* Forthisweuseconnectiondraining,itwillkeeptheinstanceforaspecificperiodoftime until requests completion (or) until the timeout(**1-3600**sec).
* To enable, Go toelb,

**Select your load balancer.**

* On instances tab.

**On connection draining.**

**Click edit.**

**Enable connection draining.**

**Time-out = 1-3600sec (default = 300).**

**Click Save.**

ACCESS LOGS

* Access logs captures detailed information about the requests sends to loadbalancer.
* Eachloadbalancercontains infosuchas,timewhentherequestwasreceived,client ip, server responses.
* You can use these logs for request patterns and to troubleshoot issues.
* These logs are stored in s3 bucket. There are noadditional costs for logs, only for storage in s3. Access logs are disabled bydefault.
* Firstyouhavetocreateabucketandcreateabucketpolicytogivewritepermissionto elb ins3.

**Go to s3.**

**Click on properties.**

* Go to permissions tab.

**Click policy generator.**

**Select type of policy = s3 bucketpolicy.**

**Principal = ELB Account id. Effect = allow.**

**Actions = put, delete (or) all actions.**

**ARN = arn:aws:s3:::bucket/my-app/AWSlogs/awsaccountid/\*. Click Add statement. Generate policy.**

* Copythepolicywhichisinjsonformatandpasteitinplaintextareains3permissions tab and clicksave.

**Toenableaccesslogs,**

**Gotoelb.**

* Ondescriptiontab.

**Clickeditonconfigureaccesslogs.**

**Select enable access logs.**

**Time interval = 60 min (default).**

**Location = s3 location with prefix(/bucket/MYapp/awslogs).**

* Ifyouwanttocreatethebucketsnowwiththeenteredprefix.Select**CREATETHIS LOCATION FOR ME**. Clicksave.
* Ifyouarecreatingbucketnow,noneedtogenerateabucketpolicy,awsautomatically create a policy along with newly createdbucket.

CUSTOM DOMAIN NAME

* ELB receives a default domain name when we created. You can connect yourwebsites by using this domain name.
* It is not easy to remember which includes region, acc ID etc. So,we create a custom domain name with route 53 and associate it with elb.
* when a client requests this custom domain, route 53 resolves it and points to elb domainname.
* Youhavetocreateahostedzone,whichcontains infoabouthowtoroutetrafficonthe internet for your domain and an alias which route queries for yourdomain.
* Go to route 53.

**Click create hostedzone.**

**Domain name = yourdomain.**

**Type = public hosted zone.**

* Select hosted zone you created,

**Click create recordset.**

**Name of your domain.**

**Type = Cname.**

**Alias Target = Select your elb domain name.**

**Select Routing policy.**

**Click Create.**

* Create another record set, just add **"WWW"**in name field. we created 2alias example.com and [www.example.com](http://www.example.com/) for our loadbalancer.

**APPLICATION LOAD BALANCER**

* An application load balancer functions at application layer of OSI model layer 7. Unlike classic load balancer, It has target groups with 1 (or) more instances in that group. It will send traffic to that group based on the rules we specified.
* You have to register an instance to a target group to send and receive traffic and you can also register a target(instance) to multiple target groups. Routing will be done independently for each group separately.
* You can add and remove targets without disrupting the overall flow of your application.
* To create an application elb, first we have to create target groups, add instances to target group and add that target group to application elb while creating it.
* To create target groups,

**Go to target groups.**

**Click Create target group.**

**Give a name to target group.**

**Select protocol, port, health checkpath and vpc.**

**Click Create.**

* After Creating target group,

**Go to targets tab.**

**Click edit.**

**Select the instances to add to group.**

**Select from which port it should send traffic.**

**Click save.**

* Now we have to create and add this target group to application load balancer.
* To create application elb,

**Go to load balancers tab.**

**Click create load balancer.**

**Choose application load balancer.**

**Give a name to load balancer.**

**Select scheme – internet (or) internal.**

**Select ip address type.**

**Select listeners.**

**Select availability zones.**

**Click next.**

* If you have added https listener, you have to add **SSLcertificate** and **securitypolicy** in this page. Otherwise it will take you to security group page.

**Select security group.**

**Click next.**

**Select target group (existing or new one).**

* If you are creating new target group, you have to add all the settings here as we did before while creating target group separately.

**Click review and finish.**

* Wait for few minutes until the instances status are **HEALTHY** and ALB status is **ACTIVE** and browse with the dnsname of your load balancer to check whether the elb is working (or) not. If it is working it will show your site configured in instances.
* You can use microservices architecture of your application with application elb. You can register an ec2 instance multiple times for each service with different port for each service.
* If you are having microservices, you can specify that path to route the traffic.
* You can use **pathbasedrouting** with application elb. You have to edit the listener and add another rule.
* For ex, you can route general requests to one target group and images requests to another target group.
* To use path based routing,

**Go to elb.**

**Select elb.**

**In listeners tab, click view and edit rules.**

**Click ‘+’ icon to add a new rule.**

**Choose path pattern**

**Path pattern = give a string(\*img\*).**

**Forward = target group.**

**Click save.**

* After creating path based routing, when an user type the url along with the string as we mentioned in path pattern, it will route the request to the target group which we specified in path based routing rule.
* There is another type of routing called **hostbasedrouting**. You can use your host to route the traffic to a target group. You have to create a rule for that host.
* select **host-header** in rule and select your target group to send traffic.

**Go to elb.**

**Select elb.**

**In listeners tab, click view and edit rules.**

**Click ‘+’ icon to add a new rule.**

**Choose host-header**

**Host-header = hostname.**

**Forward = target group.**

**Click save.**

* You can monitor application elb with **cloud watch metrics, cloud trail** and **access logs.**

**NETWORK LOAD BALANCER**

* Nlb is an 3rd type of load balancer provided by aws.
* It works on layer 4 on OSI model. Supports tcp protocol to route inbound connections to targets.
* Nlb uses flow hash algorithm to share the load among targets.

**Features:**

* It has ability to handle huge workloads and It can ability to scale millions of requests per second.
* Supports for routing requests to multiple applications in single ec2 instance by adding instance to target groups with different ports.
* NLB supports static ip address for each availability zone while creating. If your load balancer is internet facing, you can select elastic ip. As this ip address useful to store in dns records and firewall rules.
* While configuring routing in target group page, you can choose **instance** **or** **ip**. The **ip** option is useful when you want to route traffic to instances which are outside of your load balancer’s vpc or on-premises servers, which are connected via direct connect or vpn. you can choose ip and specify the server ipaddress in the targets. Once it is healthy, load balancer will route traffic to those servers. You can choose upto 1000 targets.

|  |  |
| --- | --- |
| Application | Network |
| Supports http/https | Supports tcp/udp |
| Layer 7 | Layer 4 |
| Incoming connection terminates at alb level and forwarded to targets. | Incoming connection forwarded to target directly without terminating. This is why there is no security group option for nlb. No filter is there at nlb. |
| Due to the above point, headers will be modified. (you can’t find the ipaddress of the client) | Due to the above point, headers won’t change. (you can find the ipaddress of the client) |
| Supports path-based and host-based routing | Won’t support path-based and host-based routing (only port based routing) |
| Won’t support static ip for AZ | Supports static ip for AZ |