

Graphical user interface, text, application

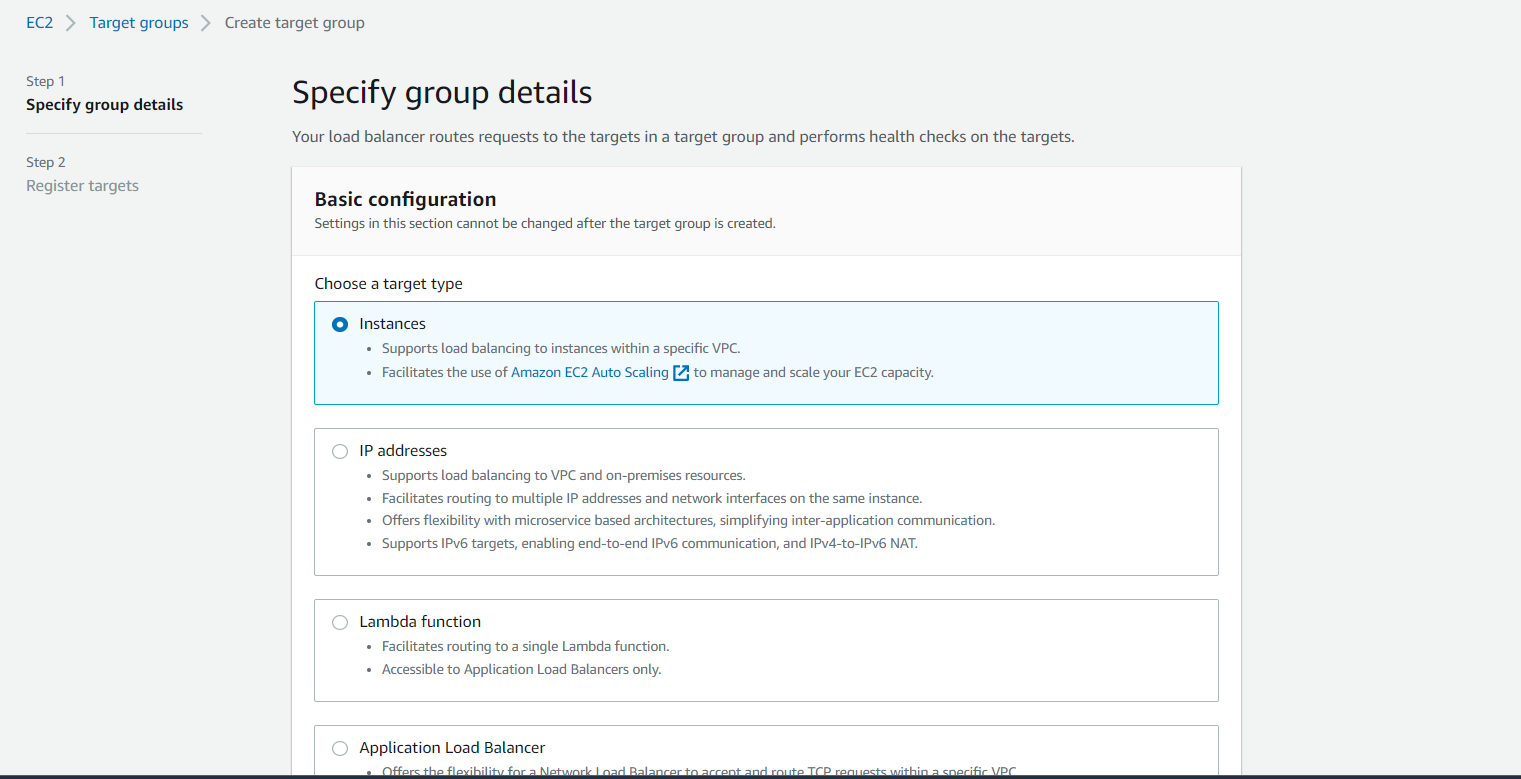
Description automatically generated

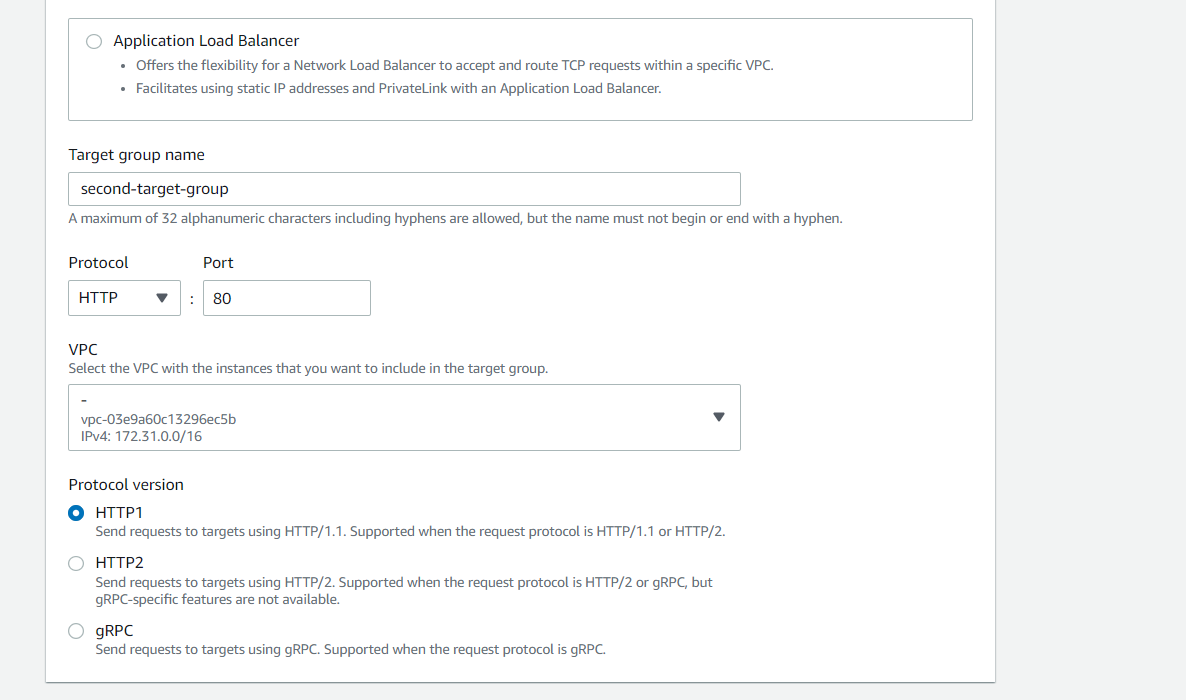
Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

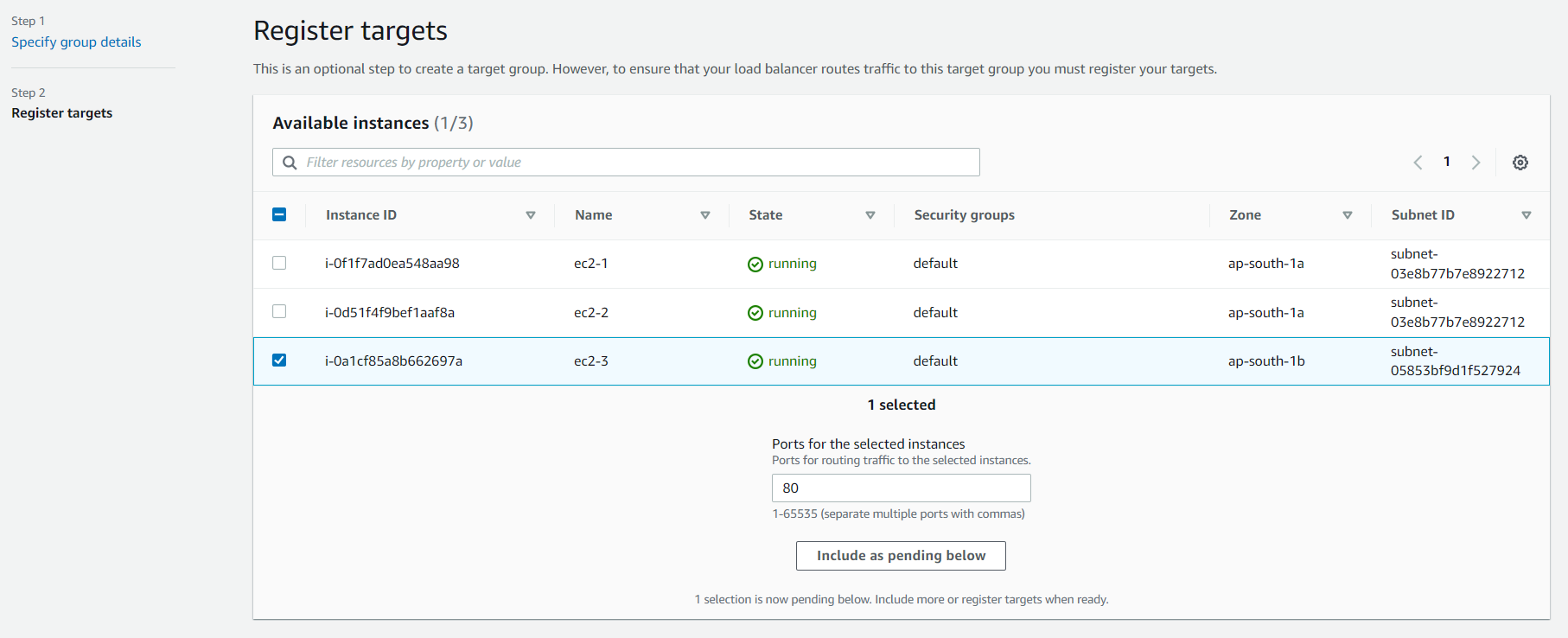
Description automatically generated





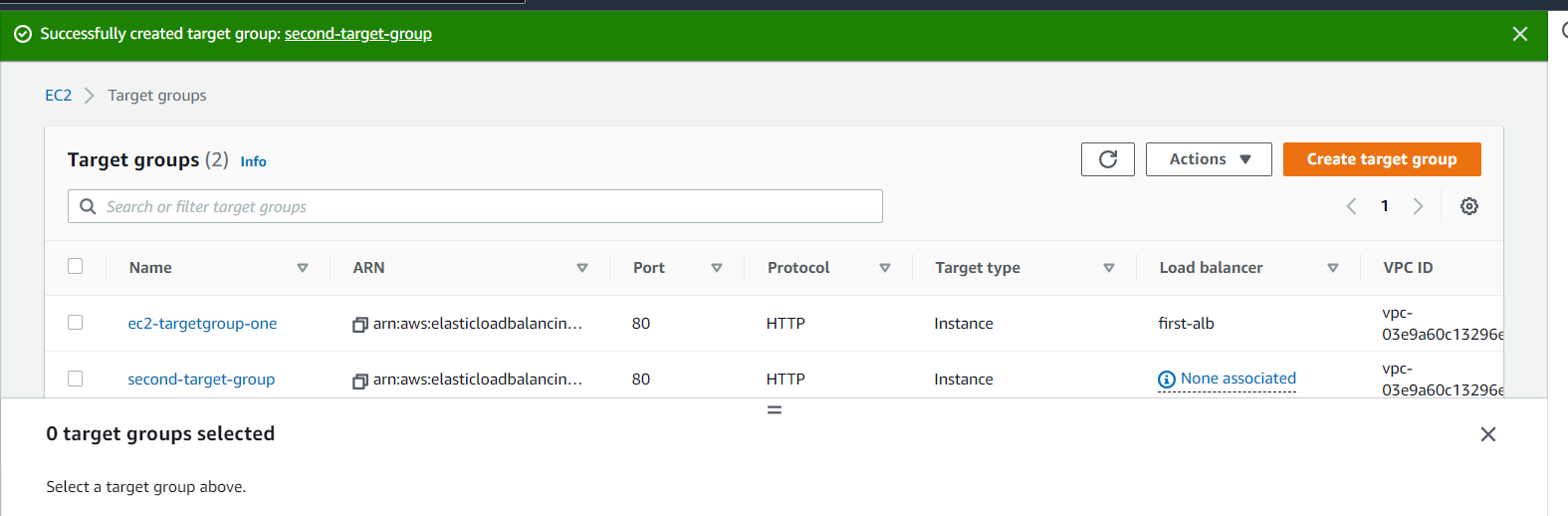
Graphical user interface, application

Description automatically generated



Graphical user interface, text, application

Description automatically generated



Graphical user interface, text, application, email

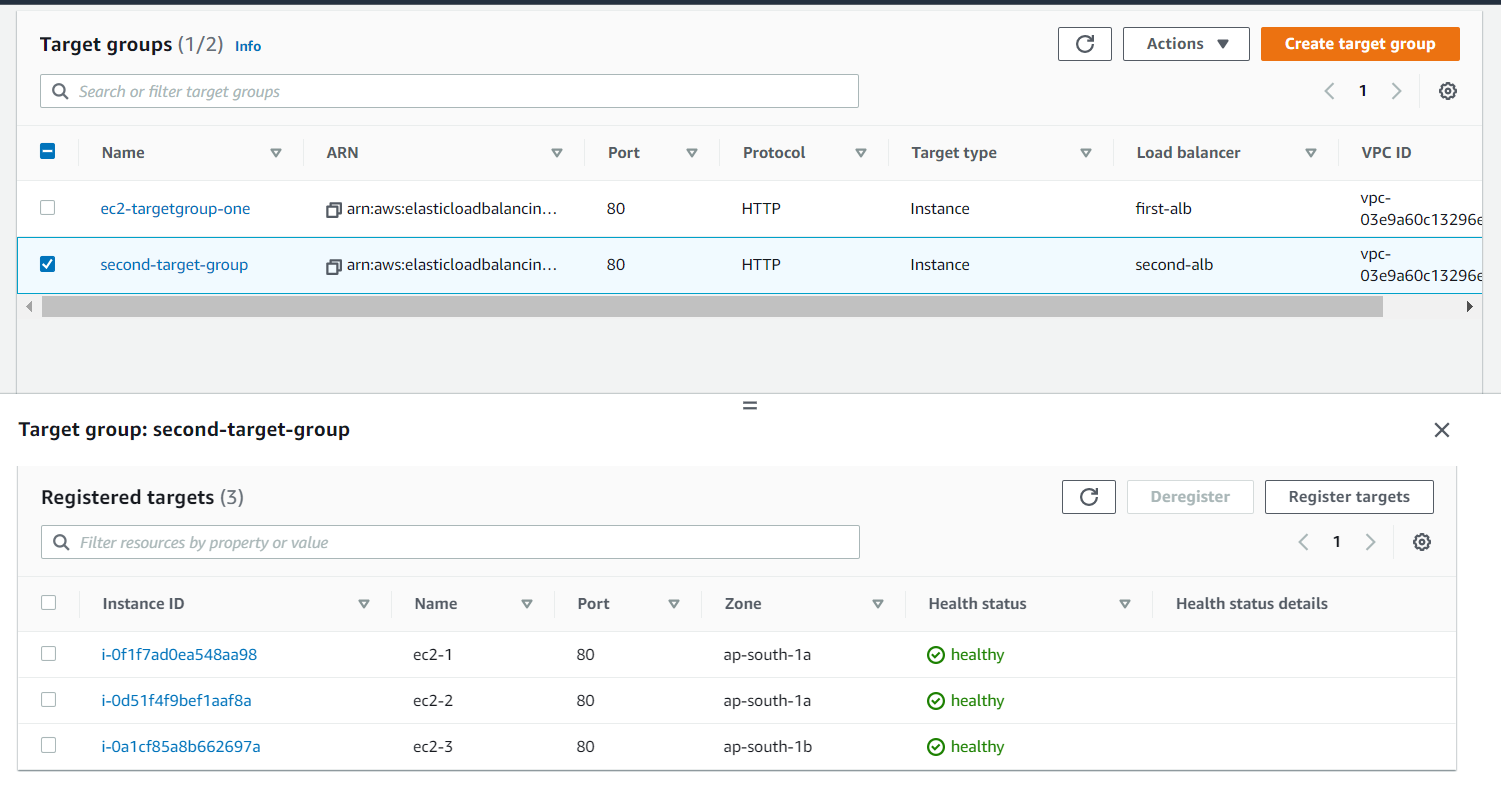
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



Graphical user interface, text, application, email

Description automatically generated

AUTO SCALING GROUP

Graphical user interface, application

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

EC2  
EC2 launch using template  
use EFS and EBS , EC2 instant store  
Security Group  
EC2 connect

Load balancer (ALB)  
  Target Group  
ASG  
Define Lauch Template  
min max desired capasity define  
SG,VPC,SUBNET,  
Can use ALB also it will auto attach create instance in target group in ELB.

What is subnet?

A subnet is a range of IP addresses in your VPC. You can attach AWS resources, such as EC2 instances and RDS DB instances, to subnets. You can create subnets to group instances together according to your security and operational needs.

How Amazon provides 99.95% RDS availability?

Multi-AZ configurations for MySQL, MariaDB, Oracle, and PostgreSQL are covered by the Amazon RDS Service Level Agreement ("SLA"). The RDS SLA affirms that AWS will use commercially reasonable efforts to make multi-AZ instances of Amazon RDS available with a Monthly Uptime Percentage of at least 99.95% during any monthly billing cycle.

In order to achieve that database has to be replicated in three different AZ. That is the reason why Amazon requires three subnets in three AZ.

What is the relationship between Region, AZ, Subnet?

The relationship is:

The region has more than one AZ

AZ belongs only to one region

AZ has more than one data centre

AZ can have more than one subnet. However, there is a soft limit of 200 subnets per AZ. You can ask Amazon for more than 200 if you need.

Subnet belongs only to one AZ

VPC belongs only to one region. You can have more than one VPC per region. However, there is a soft limit of 5 VPCs per region. You can ask Amazon for more than 5 if you need.

VPC can span more than one AZ.

You can have as many Subnets as you like in each AZ. By default, AWS creates one VPC containing one Subnet in each AZ. The number of AZ varies from region to region.

Q: How do I decide which load balancer to select for my application?

A: Elastic Load Balancing (ELB) supports four types of load balancers. You can select the appropriate load balancer based on your application needs. If you need to load balance HTTP requests, we recommend you use the Application Load Balancer (ALB). For network/transport protocols (layer4 – TCP, UDP) load balancing, and for extreme performance/low latency applications we recommend using Network Load Balancer. If your application is built within the Amazon Elastic Compute Cloud (Amazon EC2) Classic network, you should use Classic Load Balancer. If you need to deploy and run third-party virtual appliances, you can use Gateway Load Balancer.

Q: Can I privately access Elastic Load Balancing APIs from my Amazon Virtual Private Cloud (VPC) without using public IPs?

A: Yes, you can privately access Elastic Load Balancing APIs from your Amazon Virtual Private Cloud (VPC) by creating [VPC endpoints](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html). With VPC endpoints, the routing between the VPC and Elastic Load Balancing APIs is handled by the AWS network without the need for an Internet gateway, network address translation (NAT) gateway, or virtual private network (VPN) connection. The latest generation of VPC Endpoints used by Elastic Load Balancing are powered by AWS PrivateLink, an AWS technology enabling the private connectivity between AWS services using Elastic Network Interfaces (ENI) with private IPs in your VPCs. To learn more about [AWS PrivateLink](https://aws.amazon.com/privatelink/), visit the AWS PrivateLink [documentation](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Introduction.html#what-is-privatelink.html).

Q: Is there an SLA for load balancers?

A: Yes, Elastic Load Balancing guarantees a monthly availability of at least 99.99% for your load balancers (Classic, Application or Network). To learn more about the SLA and know if you are qualified for a credit, [visit here](https://aws.amazon.com/elasticloadbalancing/sla/).

Application Load Balancer

Q: Which operating systems does an Application Load Balancer support?

A: An Application Load Balancer supports targets with any operating system currently supported by the Amazon EC2 service.

Q: Which protocols does an Application Load Balancer support?

A: An Application Load Balancer supports load balancing of applications using HTTP and HTTPS (Secure HTTP) protocols.

Q: Is HTTP/2 Supported on an Application Load Balancer?

A: Yes. HTTP/2 support is enabled natively on an Application Load Balancer. Clients supporting HTTP/2 can connect to an Application Load Balancer over TLS.

Q: How can I use static IP or PrivateLink on my Application Load Balancer?

A: You can forward traffic from your Network Load Balancer, which provides support for PrivateLink and a static IP address per Availability Zone, to your Application Load Balancer. Create an Application Load Balancer-type target group, register your Application Load Balancer to it, and configure your Network Load Balancer to forward traffic to the Application Load Balancer-type target group.

Q: What TCP ports can I use to load balance?

A: You can perform load balancing for the following TCP ports: 1-65535

Q: Is WebSockets supported on an Application Load Balancer?

A: Yes. WebSockets and Secure WebSockets support is available natively and ready for use on an Application Load Balancer.

Q: Is Request tracing supported on an Application Load Balancer?

A: Yes. Request tracing is enabled by default on your Application Load Balancer.

Q: Does a Classic Load Balancer have the same features and benefits as an Application Load Balancer?

A: While there is some overlap, there is no feature parity between the two types of load balancers. Application Load Balancers are the foundation of our application layer load-balancing platform for the future.

Q: Can I configure my Amazon EC2 instances to accept traffic only from my Application Load Balancers?

A: Yes.

Q: Can I configure a security group for the front end of an Application Load Balancer?

A: Yes.

Q: Can I use the existing APIs that I use with my Classic Load Balancer with an Application Load Balancer?

A: No. Application Load Balancers require a new set of application programming interfaces (APIs).

Q: How do I manage both Application and Classic Load Balancers simultaneously?

A: The ELB Console will allow you to manage Application and Classic Load Balancers from the same interface. If you are using the command-line interface (CLI) or a software development kit (SDK), you will use a different ‘service’ for Application Load Balancers. For example, in the CLI you will describe your Classic Load Balancers using `aws elb describe-load-balancers` and your Application Load Balancers using `aws elbv2 describe-load-balancers`.

Q: Can I convert my Classic Load Balancer to an Application Load Balancer (and vice-versa)?

A: No, you cannot convert one load balancer type into another.

Q: Can I migrate to Application Load Balancer from Classic Load Balancer?

A: Yes. You can migrate to Application Load Balancer from Classic Load Balancer using one of the options listed in this [document](https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/migrate-to-application-load-balancer.html).

Q: Can I use an Application Load Balancer as a Layer-4 load balancer?

A: No. If you need Layer-4 features, you should use Network Load Balancer.

Q: Can I use a single Application Load Balancer for handling HTTP and HTTPS requests?

A: Yes, you can add listeners for HTTP port 80 and HTTPS port 443 to a single Application Load Balancer.

Q: Can I get a history of Application Load Balancing API calls made on my account for security analysis and operational troubleshooting purposes?

A: Yes. To receive a history of Application Load Balancing API calls made on your account, use [AWS CloudTrail](https://aws.amazon.com/cloudtrail/).

Q: Does an Application Load Balancer support HTTPS termination?

A: Yes, you can terminate HTTPS connection on the Application Load Balancer. You must install a Secure Sockets Layer (SSL) certificate on your load balancer. The load balancer uses this certificate to terminate the connection and then decrypt requests from clients before sending them to targets.

Q: What are the steps to get a SSL certificate?

A: You can either use [AWS Certificate Manager](https://aws.amazon.com/certificate-manager/) to provision an SSL/TLS certificate or you can obtain the certificate from other sources by creating the certificate request, getting the certificate request signed by a CA, and then uploading the certificate either using AWS Certification Manager or the [AWS Identity and Access Management](https://aws.amazon.com/iam/) (IAM) service.

Q: How does an Application Load Balancer integrate with AWS Certificate Manager (ACM)?

A: An Application Load Balancer is integrated with AWS Certificate Management (ACM). Integration with ACM simplifies binding a certificate to the load balancer, thereby streamlining the entire SSL offload process. Purchasing, uploading, and renewing SSL/TLS certificates is a complex, manual, and time-consuming process. With ACM integration with Application Load Balancer, this whole process has been shortened to simply requesting a trusted SSL/TLS certificate and selecting the ACM certificate to provision it with the load balancer.

Q: Is back-end server authentication supported with an Application Load Balancer?

A: No, only encryption is supported to the back-ends with an Application Load Balancer.

Q: How can I enable Server Name Indication (SNI) for my Application Load Balancer?

A: SNI is automatically enabled when you associate more than one TLS certificate with the same secure listener on a load balancer. Similarly, SNI mode for a secure listener is automatically disabled when you have only one certificate associated to a secure listener.

Q: Can I associate multiple certificates for the same domain to a secure listener?

A: Yes, you can associate multiple certificates for the same domain to a secure listener. For example, you can associate:

ECDSA and RSA certificates

Certificates with different key sizes (e.g. 2K and 4K) for SSL/TLS certificates

Single-Domain, Multi-Domain (SAN) and Wildcard certificates

Q: Is IPv6 supported with an Application Load Balancer?

A: Yes, IPv6 is supported with an Application Load Balancer.

Q: How do you set up rules on an Application Load Balancer?

A: You can configure rules for each of the listeners on the load balancer. The rules include conditions and corresponding actions if the conditions are satisfied. The supported conditions are Host header, path, HTTP headers, methods, query parameters, and source IP classless inter-domain routing (CIDR). The supported actions are redirect, fixed response, authenticate, and forward. Once you have set this up, the load balancer will use the rules to determine how a particular HTTP request should be routed. You can use multiple conditions and actions in a rule, and in each condition can specify a match on multiple values.

Q: Are there limits on the resources for an Application Load Balancer?

A: Your AWS account has these [limits](http://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-limits.html) for an Application Load Balancer.

Q: How can I protect my web applications behind a load balancer from web attacks?

A: You can integrate your Application Load Balancer with AWS Web Application Firewall (WAF), a web application firewall that helps protect web applications from attacks by allowing you to configure rules based on IP addresses, HTTP headers, and custom uniform resource identifier (URI) strings. Using these rules, AWS WAF can block, allow, or monitor (count) web requests for your web application. Please see AWS WAF developer guide for more information.

Q: Can I load balance to any arbitrary IP address?

A: You can use any IP address from the load balancer’s VPC CIDR for targets within load balancer’s VPC, and any IP address from RFC 1918 ranges (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) or RFC 6598 range (100.64.0.0/10) for targets located outside the load balancer’s VPC (for example, targets in Peered VPC, Amazon EC2 Classic, and on-premises locations reachable over AWS Direct Connect or VPN connection).

Q: How can I load balance applications distributed across a VPC and on-premises location?

A: There are various ways to achieve hybrid load balancing. If an application runs on targets distributed between a VPC and an on-premises location, you can add them to the same target group using their IP addresses. To migrate to AWS without impacting your application, gradually add VPC targets to the target group and remove on-premises targets from the target group.

If you have two different applications such that the targets for one application are in a VPC and the targets for other applications are in on-premises location, you can put the VPC targets in one target group and the on-premises targets in another target group and use content based routing to route traffic to each target group. You can also use separate load balancers for VPC and on-premises targets and use DNS weighting to achieve weighted load balancing between VPC and on-premises targets.

Q: How can I load balance to EC2-Classic instances?

A: You cannot load balance to EC2-Classic Instances when registering their Instance IDs as targets. However if you link these EC2-Classic instances to the load balancer's VPC using ClassicLink and use the private IPs of these EC2-Classic instances as targets, then you can load balance to the EC2-Classic instances. If you are using EC2 Classic instances today with a Classic Load Balancer, you can easily migrate to an Application Load Balancer.

Q: How do I enable cross-zone load balancing in Application Load Balancer?

A: Cross-zone load balancing is already enabled by default in Application Load Balancer.

Q: When should I authenticate users using the Application Load Balancer’s integration with Amazon Cognito vs. the Application Load Balancers’ native support for OpenID Connect (IODC) identity providers (IdPs)?

A: You should use authentication through Amazon Cognito if:

You want to provide flexibility to your users to authenticate via social network identities (Google, Facebook, and Amazon) or enterprise identities (SAML) or via your own user directories provided by Amazon Cognito’s User Pool.

You are managing multiple identity providers including OpenID Connect and want to create a single authentication rule in Application Load Balancer (ALB) that can use Amazon Cognito to federate your multiple identity providers.

You need to actively manage user profiles with one or more social or OpenID Connect identity providers from one central place. For example, you can put users in groups and add custom attributes to represent user status and control access for paid users.

Alternatively, if you have invested in developing custom IdP solutions and simply want to authenticate with a single OpenID Connect-compatible identity provider, you may prefer using Application Load Balancer’s native OIDC solution.

Q: What type of redirects does Application Load Balancer support?

A: The following three types of redirects are supported.

|  |  |
| --- | --- |
| **Types of redirects** | **Examples** |
| HTTP to HTTP | http://hostA to http://hostB |
| HTTP to HTTPS | http://hostA to https://hostB https://hostA:portA/pathA to https://hostB:portB/pathB |
| HTTPS to HTTPS | https://hostA to https://hostB |

Q: What content types does ALB support for the message body of fixed-response action?

A: The following content types are supported: text/plain, text/css, text/html, application/javascript, application/json.

Q: How does AWS Lambda invocation via Application Load Balancer work?

A: HTTP(S) requests received by a load balancer are processed by the content-based routing rules. If the request content matches the rule—with an action to forward it to a target group through a Lambda function as a target—then the corresponding Lambda function is invoked. The content of the request (including headers and body) is passed on to the Lambda function in JavaScript object notation (JSON) format. The response from the Lambda function should be in JSON format. The response from the Lambda function is transformed into an HTTP response and sent to the client. The load balancer invokes your Lambda function using the AWS Lambda Invoke API, and requires that you provide invoke permissions for your Lambda function to the Elastic Load Balancing service.

Q: Does Lambda invocation via Application Load Balancer support requests over both HTTP and HTTPS protocol?

A: Yes. Application Load Balancer supports Lambda invocation for requests over both HTTP and HTTPS protocol.

Q: In which AWS Regions can I use Lambda functions as targets with the Application Load Balancer?

A: You can use Lambda as a target with the Application Load Balancer in US East (N. Virginia), US East (Ohio), US West (Northern California), US West (Oregon), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada ( Central), EU (Frankfurt), EU (Ireland), EU (London), EU (Paris), South America (São Paulo), and GovCloud (US-West) AWS Regions.

Q: Is the Application Load Balancer available in AWS Local Zones?

A: Yes, Application Load Balancer is available in the Local Zone in Los Angeles. Within the Los Angeles Local Zone, Application Load Balancer will operate in a single subnet and scale automatically to meet varying levels of application load without manual intervention.

Application Load Balancer Pricing FAQs

Q: How does Application Load Balancer pricing work?

A: You are charged for each hour or partial hour that an Application Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used per hour.

Q: What is a Load Balancer Capacity Unit (LCU)?

A: An LCU is a new metric for determining how you pay for an Application Load Balancer. An LCU defines the maximum resource consumed in any one of the dimensions (new connections, active connections, bandwidth and rule evaluations) the Application Load Balancer processes your traffic.

Q: Will I be billed on Classic Load Balancers by LCU?

A: No, Classic Load Balancers will continue to be billed for bandwidth and hourly usage.

Q: How do I know the number of LCUs an Application Load Balancer is using?

A: We expose the usage of all four dimensions that constitute an LCU via Amazon CloudWatch.

Q: Will I be billed on all the dimensions in an LCU?

A: No. The number of LCUs per hour will be determined based on maximum resource consumed amongst the four dimensions that constitutes a LCU.

Q: Will I be billed on partial LCUs?

A: Yes.

Q: Is a free tier offered on an Application Load Balancer for new AWS accounts?

A: Yes. For new AWS accounts, a free tier for an Application Load Balancer offers 750 hours and 15 LCUs. This free tier offer is only available to new AWS customers, and is available for 12 months following your AWS sign-up date.

Q: Can I use a combination of Application Load Balancer and Classic Load Balancer as part of my free tier?

A: Yes. You can use both Classic and Application Load Balancers for 15 GB and 15 LCUs respectively. The 750 load balancer hours are shared between both Classic and Application Load Balancers.

Q: What are rule evaluations?

A: Rule evaluations are defined as the product of number of rules processed and the request rate averaged over an hour.

Q: How does the LCU billing work with different certificate types and key sizes?

A: Certificate key size affects only the number of new connections per second in the LCU computation for billing. The following table lists the value of this dimension for different key sizes for RSA and ECDSA certificates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RSA certificates** |  |  |  |  |
| Key Size | <=2K | <=4K | <=8K | >8K |
| New connections/sec | 25 | 5 | 1 | 0.25 |
| ECDSA Certificates |  |  |  |  |
| Key Size | <=256 | <=384 | <=521 | >521 |
| New connections/sec | 25 | 5 | 1 | 0.25 |

Q: Am I charged for regional AWS data transfer when enabling cross-zone load balancing in Application Load Balancer?

A: No. Since cross-zone load balancing is always on with Application Load Balancer, you are not charged for this type of regional data transfer.

Q: Is user authentication in Application Load Balancer charged separately?

A: No. There is no separate charge for enabling the authentication functionality in Application Load Balancer. When using Amazon Cognito with Application Load Balancer, Amazon Cognito pricing will apply.

Q: How do you charge for Application Load Balancer usage with AWS Lambda targets?

A: You are charged as usual for each hour or partial hour that an Application Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used per hour. For Lambda targets, each LCU offers 0.4 GB processed bytes per hour, 25 new connections per second, 3,000 active connections per minute, and 1,000 rule evaluations per second. For the processed bytes dimension, each LCU provides 0.4 GB per hour for Lambda targets versus 1 GB per hour for all other target types like Amazon EC2 instances, containers, and IP addresses. Please note that usual AWS Lambda charges apply to Lambda invocations by Application Load Balancer.

Q: How can I differentiate the bytes processed by Lambda targets versus bytes processed by other targets (Amazon EC2, containers, and on-premises servers)?

A: Applications Load Balancers emit two new CloudWatch metrics. LambdaTargetProcessedBytes metric indicates the bytes processed by Lambda targets, and the StandardProcessedBytes metric indicates bytes processed by all other target types.

Network Load Balancer

Q: Can I create a TCP or UDP (Layer 4) listener for my Network Load Balancer?

A: Yes. Network Load Balancers support both TCP, UDP, and TCP+UDP (Layer 4) listeners, as well as TLS listeners.

Q: What are the key features available with the Network Load Balancer?

A: Network Load Balancer provides both TCP and UDP (Layer 4) load balancing. It is architected to handle millions of requests per second and sudden volatile traffic patterns, and provides extremely low latencies. In addition, Network Load Balancer also supports TLS termination, preserves the source IP of the clients, and provides stable IP support and zonal isolation. It also supports long-running connections that are useful for WebSocket type applications.

Q: Can Network Load Balancer process both TCP and UDP protocol traffic on the same port?

A: Yes. To achieve this, you can use a TCP+UDP listener. For example, for a DNS service using both TCP and UDP, you can create a TCP+UDP listener on port 53, and the load balancer will process traffic for both UDP and TCP requests on that port. You must associate a TCP+UDP listener with a TCP+UDP target group.

Q: How does Network Load Balancer compare to what I get with the TCP listener on a Classic Load Balancer?

A: Network Load Balancer preserves the source IP of the client, which is not preserved in the Classic Load Balancer. Customers can use proxy protocol with Classic Load Balancer to get the source IP. Network Load Balancer automatically provides a static IP per Availability Zone (AZ) to the load balancer and also enables assigning an Elastic IP to the load balancer per AZ. This is not supported with Classic Load Balancer.

Q: Can I migrate to Network Load Balancer from Classic Load Balancer?

A: Yes. You can migrate to Network Load Balancer from Classic Load Balancer using one of the options listed in this document.

Q: Are there limits on the resources for my Network Load Balancer?

A: Yes, please refer to Network Load Balancer limits documentation for more information.

Q: Can I use the AWS Management Console to set up my Network Load Balancer?

A: Yes, you can use the AWS Management Console, AWS CLI, or the API to set up a Network Load Balancer.

Q: Can I use the existing API for Classic Load Balancers for my Network Load Balancers?

A: No. To create a Classic Load Balancer, use the 2012-06-01 API. To create a Network Load Balancer or an Application Load Balancer, use the 2015-12-01 API.

Q: Can I create my Network Load Balancer in a single Availability Zone?

A: Yes, you can create your Network Load Balancer in a single AZ by providing a single subnet when you create the load balancer.

Q: Does Network Load Balancer support DNS regional and zonal fail-over?

A: Yes, you can use Amazon Route 53 health checking and DNS failover features to enhance the availability of the applications running behind Network Load Balancers. Using Route 53 DNS failover, you can run applications in multiple AWS Availability zones and designate alternate load balancers for failover across regions.

In the event that you have your Network Load Balancer configured for multi-AZ, if there are no healthy Amazon EC2 instances registered with the load balancer for that AZ, or if the load balancer nodes in a given zone are unhealthy, then Route 53 will fail away to alternate load balancer nodes in other healthy AZs.

Q: Can I have a Network Load Balancer with a mix of ELB-provided IPs and Elastic IPs or assigned private IPs?

A: No. A Network Load Balancer’s addresses must be completely controlled by you, or completely controlled by ELB. This is to ensure that when using Elastic IPs with a Network Load Balancer, all addresses known to your clients do not change.

Q: Can I assign more than one EIP to my Network Load Balancer in each subnet?

A: No. For each associated subnet a Network Load Balancer is in, the Network Load Balancer can only support a single public/internet facing IP address.

Q: If I remove/delete a Network Load Balancer what will happen to the Elastic IP addresses that were associated with it?

A: The Elastic IP Addresses that were associated with your load balancer will return to your allocated pool and be available for future use.

Q: Does Network Load Balancer support internal load balancers?

A: Network Load Balancer can be set up as an internet-facing load balancer or an internal load balancer, similar to what is possible with Application Load Balancer and Classic Load Balancer.

Q: Can the internal Network Load balancer support more than one private IP in each subnet?

A: No. For each associated subnet that a load balancer is in, the Network Load Balancer can only support a single private IP.

Q: Can I set up Websockets with my Network Load Balancer?

A: Yes, configure TCP listeners that route the traffic to the targets that implement WebSockets protocol (https://tools.ietf.org/html/rfc6455 ). Because WebSockets is a layer 7 protocol and Network Load Balancer is operating at layer 4, no special handling exists in Network Load Balancer for WebSockets or other higher level protocols.

Q: Can I load balance to any arbitrary IP address?

A: Yes. You can use any IP address from the load balancer’s VPC CIDR for targets within load balancer’s VPC and any IP address from RFC 1918 ranges (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) or RFC 6598 range (100.64.0.0/10) for targets located outside the load balancer’s VPC (EC2-Classic and on-premises locations reachable over AWS Direct Connect). Load balancing to IP address target type is supported for TCP listeners only, and is currently not supported for UDP listeners.

Q: Can I use Network Load Balancer to setup AWS PrivateLink?

A: Yes, Network Load Balancers with TCP and TLS Listeners can be used to setup AWS PrivateLink. You cannot set up PrivateLink with UDP listeners on Network Load Balancers.

Q: What is a UDP flow?

A: While user datagram protocol (UDP) is connectionless, the load balancer maintains UDP flow state based on 5-tuple hash, ensuring that packets sent in the same context are consistently forwarded to the same target. The flow is considered active as long as traffic is flowing and until the idle timeout is reached. Once the timeout threshold is reached, the load balancer will forget the affinity, and the incoming UDP packet will be considered a new flow and load-balanced to a new target.

Q: What is the idle timeout supported by Network Load Balancer?

A: Network Load Balancer idle timeout for TCP connections is 350 seconds. The idle timeout for UDP flows is 120 seconds.

Q: What is the benefit of targeting containers behind a load balancer with IP addresses instead of instance IDs?

A: Each container on an instance can now have its own security group, and does not need to share security rules with other containers. You can attach security groups to an ENI, and each ENI on an instance can have a different security group. You can map a container to the IP address of a particular ENI to associate security group(s) per container. Load balancing using IP addresses also allows multiple containers running on an instance use the same port (say port 80). The ability to use the same port across containers allows containers on an instance to communicate with each other through well-known ports instead of random ports.

Q: How can I load balance applications distributed across a VPC and on-premises location?

A: There are various ways to achieve hybrid load balancing. If an application runs on targets distributed between a VPC and an on-premises location, you can add them to the same target group using their IP addresses. To migrate to AWS without impacting your application, gradually add VPC targets to the target group and remove on-premises targets from the target group. You can also use separate load balancers for VPC and on-premises targets and use DNS weighting to achieve weighted load balancing between VPC and on-premises targets.

Q: How can I load balance to EC2-Classic instances?

A: You cannot load balance to EC2-Classic Instances when registering their Instance IDs as targets. However if you link these EC2-Classic instances to the load balancer's VPC using ClassicLink and use the private IPs of these EC2-Classic instances as targets, then you can load balance to the EC2-Classic instances. If you are using EC2 Classic instances today with a Classic Load Balancer, you can easily migrate to a Network Load Balancer.

Q: How do I enable cross-zone load balancing in Network Load Balancer?

A: You can enable cross-zone loading balancing only after creating your Network Load Balancer. You achieve this by editing the load balancing attributes section and then selecting the cross-zone load balancing support checkbox.

Q: Am I charged for regional AWS data-transfer when I enable cross-zone load balancing in Network Load Balancer?

A: Yes, you will be charged for regional data transfer between Availability Zones with Network Load Balancer when cross-zone load balancing is enabled. Check the charges in the data transfer section of the [Amazon EC2 On-Demand Pricing page](https://aws.amazon.com/ec2/pricing/on-demand/).

Q: Is there any impact of cross-zone load balancing on Network Load Balancer limits?

A: Yes. Network Load Balancer currently supports 200 targets per Availability Zone. For example, if you are in two AZs, you can have up to 400 targets registered with Network Load Balancer. If cross-zone load balancing is on, then the maximum targets reduce from 200 per AZ to 200 per load balancer. So, in the example above: When cross-zone load balancing is on, even though your load balancer is in two AZs, you are limited to 200 targets that can be registered to the load balancer.

Q: Does Network Load Balancer support TLS termination?

A: Yes, you can terminate TLS connections on the Network Load Balancer. You must install an SSL certificate on your load balancer. The load balancer uses this certificate to terminate the connection and then decrypt requests from clients before sending them to targets.

Q: Is source IP is preserved when terminating TLS on Network Load Balancer?

A: Source IP continues to be preserved even if you terminate TLS on the Network Load Balancer.

Q: What are the steps to get a SSL certificate?

A: You can either use [AWS Certificate Manager](https://aws.amazon.com/certificate-manager/) to provision an SSL/TLS certificate, or you can obtain the certificate from other sources by creating the certificate request, getting the certificate request signed by a certificate authority (CA), and then uploading the certificate either using AWS Certification Manager (ACM) or the [AWS Identity and Access Management](https://aws.amazon.com/iam/) (IAM) service.

Q: How can I enable Server Name Indication (SNI) for my Network Load Balancer?

A: SNI is automatically enabled when you associate more than one TLS certificate with the same secure listener on a load balancer. Similarly, SNI mode for a secure listener is automatically disabled when you have only one certificate associated to a secure listener.

Q: How does the Network Load Balancer integrate with AWS Certificate Manager (ACM) or Identity Access Manager (IAM)?

A: Network Load Balancer is integrated with AWS Certificate Management (ACM). Integration with ACM makes it very simple to bind a certificate to the load balancer thereby making the entire SSL offload process very easy. Purchasing, uploading, and renewing SSL/TLS certificates is a time-consuming manual and complex process. With ACM integration with Network Load Balancer, this whole process has been shortened to simply requesting a trusted SSL/TLS certificate and selecting the ACM certificate to provision it with the load balancer. Once you create a Network Load balancer, you can now configure a TLS listener followed by an option to select a certificate from either ACM or Identity Access Manager (IAM). This experience is similar to what you have in Application Load Balancer or Classic Load Balancer.

Q: Is back-end server authentication supported with Network Load Balancer?

A: No, only encryption is supported to the back-ends with Network Load Balancer.

Q: What are the certificate types supported by Network Load Balancer?

A: Network Load Balancer only supports RSA certificates with 2K key size. We currently do not support RSA certificate key sizes greater than 2K or ECDSA certificates on the Network Load Balancer.

Q: In which AWS Regions is TLS Termination on Network Load Balancer supported?

A: You can use TLS Termination on Network Load Balancer in US East (N. Virginia), US East (Ohio), US West (Northern California), US West (Oregon), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), EU (Frankfurt), EU (Ireland), EU (London), EU (Paris), South America (São Paulo), and GovCloud (US-West) AWS Regions.

Network Load Balancer Pricing FAQs

Q: How does Network Load Balancer pricing work?

A: You are charged for each hour or partial hour that a Network Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used by Network Load Balancer per hour.

Q: What is a Load Balancer Capacity Unit (LCU)?

A: An LCU is a new metric for determining how you pay for a Network Load Balancer. An LCU defines the maximum resource consumed in any one of the dimensions (new connections/flows, active connections/flows, and bandwidth) the Network Load Balancer processes your traffic.

Q: What are the LCU metrics for TCP traffic on Network Load Balancer?

A: The LCU metrics for the TCP traffic are as follows:

800 new TCP connections per second.

100,000 active TCP connections (sampled per minute).

1 GB per hour for Amazon EC2 instances, containers, and IP addresses as targets.

Q: What are the LCU metrics for UDP traffic on Network Load Balancer?

A: The LCU metrics for the UDP traffic are as follows:

400 new flows per second.

50,000 active UDP flows (sampled per minute).

1 GB per hour for Amazon EC2 instances, containers, and IP addresses as targets.

Q: What are the LCU metrics for TLS traffic on Network Load Balancer?

A: The LCU metrics for the TLS traffic are as follows:

50 new TLS connections per second.

3,000 active TLS connections (sampled per minute).

1 GB per hour for Amazon EC2 instances, containers, and IP addresses as targets.

Q: Will I be billed on all the dimensions (Processed Bytes, New Flows and Active Flows)?

A: No, for each protocol you are charged only on one of the three dimensions (the highest for the hour).

Q: Is new connections/flows per sec same as requests/sec?

A: No. Multiple requests can be sent in a single connection.

Q: Will I be billed on Classic Load Balancers by LCU?

A: No. Classic Load Balancers will continue to be billed for bandwidth and hourly charge.

Q: How do I know the number of LCUs a Network Load Balancer is using?

A: We will expose the usage of all three dimensions that constitutes a LCU via Amazon CloudWatch.

Q: Will I be billed on all the dimensions in an LCU?

A: No. The number of LCUs per hour will be determined based on maximum resource consumed amongst the three dimensions that constitutes a LCU.

Q: Will I be billed on partial LCUs?

A: Yes.

Q: Is a free tier offered on a Network Load Balancer for new AWS accounts?

A: Yes. For new AWS accounts, a free tier for a Network Load Balancer offers 750 hours and 15 LCUs. This free tier offer is only available to new AWS customers, and is available for 12 months following your AWS sign-up date.

Q: Can I use a combination of Network Load Balancer, Application Load Balancer and Classic Load Balancer as part of my free tier?

A: Yes. You can use Application and Network each for 15 LCUs and Classic for 15 GB respectively. The 750 load balancer hours are shared between Application, Network, and Classic Load Balancers.

**Auto scaling group**

General

Q: What is Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling is a fully managed service designed to launch or terminate Amazon EC2 instances automatically to help ensure you have the correct number of Amazon EC2 instances available to handle the load for your application. Amazon EC2 Auto Scaling helps you maintain application availability through fleet management for EC2 instances, which detects and replaces unhealthy instances, and by scaling your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Amazon EC2 Auto Scaling to automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

Q. When should I use Amazon EC2 Auto Scaling vs. AWS Auto Scaling?

You should use AWS Auto Scaling to manage scaling for multiple resources across multiple services. AWS Auto Scaling lets you define dynamic scaling policies for multiple EC2 Auto Scaling groups or other resources using predefined scaling strategies. Using AWS Auto Scaling to configure scaling policies for all of the scalable resources in your application is faster than managing scaling policies for each resource via its individual service console. It’s also easier, as AWS Auto Scaling includes predefined scaling strategies that simplify the setup of scaling policies.

You should use EC2 Auto Scaling if you only need to scale Amazon EC2 Auto Scaling groups, or if you are only interested in maintaining the health of your EC2 fleet. You should also use EC2 Auto Scaling if you need to create or configure Amazon EC2 Auto Scaling groups, or if you need to set up scheduled or step scaling policies (as AWS Auto Scaling supports only target tracking scaling policies).

EC2 Auto Scaling groups must be created and configured outside of AWS Auto Scaling, such as through the EC2 console, Auto Scaling API or via CloudFormation. AWS Auto Scaling can help you configure dynamic scaling policies for your existing EC2 Auto Scaling groups.

Q: How is Predictive Scaling Policy different from Predictive Scaling of AWS Auto Scaling plan?

Predictive Scaling Policy brings the similar prediction algorithm offered through AWS Auto Scaling plan as a native scaling policy in EC2 Auto Scaling. You can use predictive scaling directly through AWS Command Line Interface (CLI), EC2 Auto Scaling Management Console, and AWS SDKs similar to how you use other scaling policies, such as Simple Scaling or Target Tracking etc.  You don’t have to create an AWS Auto Scaling plan just for using predictive scaling.

Q: What are the benefits of using Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling helps to maintain your Amazon EC2 instance availability. Whether you are running one Amazon EC2 instance or thousands, you can use Amazon EC2 Auto Scaling to detect impaired Amazon EC2 instances, and replace the instances without intervention. This ensures that your application has the compute capacity that you expect. You can use Amazon EC2 Auto Scaling to automatically scale your Amazon EC2 fleet by following the demand curve for your applications, reducing the need to manually provision Amazon EC2 capacity in advance. For example, you can set a condition to add new Amazon EC2 instances in increments to the ASG when the average utilization of your Amazon EC2 fleet is high; and similarly, you can set a condition to remove instances in increments when CPU utilization is low. You can also use Amazon CloudWatch to send alarms to trigger scaling activities and Elastic Load Balancing (ELB) to distribute traffic to your instances within the ASG. If you have predictable load changes, you can use Predictive Scaling policy to proactively increase capacity ahead of upcoming demand. Amazon EC2 Auto Scaling enables you to run your Amazon EC2 fleet at optimal utilization.

Q: What is fleet management and how is it different from dynamic scaling?

If your application runs on Amazon EC2 instances, then you have what’s referred to as a ‘fleet’. *Fleet management* refers to the functionality that automatically replaces unhealthy instances and maintains your fleet at the desired capacity. Amazon EC2 Auto Scaling fleet management ensures that your application is able to receive traffic and that the instances themselves are working properly. When Auto Scaling detects a failed [health check](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html), it can replace the instance automatically.

The *dynamic scaling* capabilities of Amazon EC2 Auto Scaling refers to the functionality that automatically increases or decreases capacity based on load or other metrics. For example, if your CPU spikes above 80% (and you have an alarm setup) Amazon EC2 Auto Scaling can add a new instance dynamically.

Q: What is target tracking?

Target tracking is a new type of scaling policy that you can use to set up dynamic scaling for your application in just a few simple steps. With target tracking, you select a load metric for your application, such as CPU utilization or request count, set the target value, and Amazon EC2 Auto Scaling adjusts the number of EC2 instances in your ASG as needed to maintain that target. It acts like a home thermostat, automatically adjusting the system to keep the environment at your desired temperature. For example, you can configure target tracking to keep CPU utilization for your fleet of web servers at 50%. From there, Amazon EC2 Auto Scaling launches or terminates EC2 instances as required to keep the average CPU utilization at 50%.

Q: What is an EC2 Auto Scaling group (ASG)?

An Amazon EC2 Auto Scaling group (ASG) contains a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of fleet management and dynamic scaling. For example, if a single application operates across multiple instances, you might want to increase the number of instances in that group to improve the performance of the application, or decrease the number of instances to reduce costs when demand is low. Amazon EC2 Auto Scaling will automaticallly adjust the number of instances in the group to maintain a fixed number of instances even if a instance becomes unhealthy, or based on criteria that you specify. You can find more information about ASG in the [Amazon EC2 Auto Scaling User Guide](http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html).

Q: What happens to my Amazon EC2 instances if I delete my ASG?

If you have an EC2 Auto Scaling group (ASG) with running instances and you choose to delete the ASG, the instances will be terminated and the ASG will be deleted.

Q: How do I know when EC2 Auto Scaling is launching or terminating the EC2 instances in an EC2 Auto Scaling group?

When you use Amazon EC2 Auto Scaling to scale your applications automatically, it is useful to know when EC2 Auto Scaling is launching or terminating the EC2 instances in your EC2 Auto Scaling group. [Amazon SNS](https://aws.amazon.com/sns/) coordinates and manages the delivery or sending of notifications to subscribing clients or endpoints. You can configure EC2 Auto Scaling to send an SNS notification whenever your EC2 Auto Scaling group scales. Amazon SNS can deliver notifications as HTTP or HTTPS POST, email (SMTP, either plain-text or in JSON format), or as a message posted to an Amazon SQS queue. For example, if you configure your EC2 Auto Scaling group to use the autoscaling: EC2\_INSTANCE\_TERMINATE notification type, and your EC2 Auto Scaling group terminates an instance, it sends an email notification. This email contains the details of the terminated instance, such as the instance ID and the reason that the instance was terminated.

For more information read [Getting SNS Notifications when your EC2 Auto Scaling Group Scales](http://docs.aws.amazon.com/autoscaling/latest/userguide/ASGettingNotifications.html).

Q: What is a launch configuration?

A launch configuration is a template that an EC2 Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance. When you create an EC2 Auto Scaling group, you must specify a launch configuration. You can specify your launch configuration with multiple EC2 Auto Scaling groups. However, you can only specify one launch configuration for an EC2 Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. Therefore, if you want to change the launch configuration for your EC2 Auto Scaling group, you must create a launch configuration and then update your EC2 Auto Scaling group with the new launch configuration. When you change the launch configuration for your EC2 Auto Scaling group, any new instances are launched using the new configuration parameters, but existing instances are not affected. You can see the [launch configurations](https://docs.aws.amazon.com/autoscaling/ec2/userguide/LaunchConfiguration.html) section of the EC2 Auto Scaling User Guide for more details.

Q: How many instances can an EC2 Auto Scaling group have?

You can have as many instances in your EC2 Auto Scaling group as your EC2 quota allows.

Q: What happens if a scaling activity causes me to reach my Amazon EC2 limit of instances?

Amazon EC2 Auto Scaling cannot scale past the Amazon EC2 limit of instances that you can run. If you need more Amazon EC2 instances, complete the [Amazon EC2 instance request form](https://aws.amazon.com/contact-us/ec2-request/).

Q: Can EC2 Auto Scaling groups span multiple AWS regions?

EC2 Auto Scaling groups are regional constructs. They can span Availability Zones, but not AWS regions.

Q: How can I implement changes across multiple instances in an EC2 Auto Scaling group?

You can use AWS CodeDeploy or CloudFormation to orchestrate code changes to multiple instances in your EC2 Auto Scaling group.

Q: If I have data installed in an EC2 Auto Scaling group, and a new instance is dynamically created later, is the data copied over to the new instances?

Data is not automatically copied from existing instances to new instances. You can use [lifecycle hooks](http://docs.aws.amazon.com/autoscaling/latest/userguide/lifecycle-hooks.html) to copy the data, or an [Amazon RDS](https://aws.amazon.com/rds/) database including replicas.

Q: When I create an EC2 Auto Scaling group from an existing instance, does it create a new AMI (Amazon Machine Image)?

When you create an Auto Scaling group from an existing instance, it does not create a new AMI. For more information see [Creating an Auto Scaling Group Using an EC2 Instance](http://docs.aws.amazon.com/autoscaling/latest/userguide/create-asg-from-instance.html).

Q: How does Amazon EC2 Auto Scaling balance capacity?

Balancing resources across [Availability Zones](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html) is a best practice for well-architected applications, as this greatly increases aggregate system availability. Amazon EC2 Auto Scaling automatically balances EC2 instances across zones when you [configure multiple zones](http://docs.aws.amazon.com/autoscaling/latest/userguide/as-add-availability-zone.html) in your EC2 Auto Scaling group settings. Amazon EC2 Auto Scaling always launches new instances such that they are balanced between zones as evenly as possible across the entire fleet. What’s more, Amazon EC2 Auto Scaling only launches into Availability Zones in which there is available capacity for the requested instance type.

Q: What are lifecycle hooks?

Lifecycle hooks let you take action before an instance goes into service or before it gets terminated. This can be especially useful if you are not baking your software environment into an Amazon Machine Image (AMI). For example, launch hooks can perform software configuration on an instance to ensure that it’s fully prepared to handle traffic before Amazon EC2 Auto Scaling proceeds to connect it to your load balancer. One way to do this is by connecting the launch hook to an AWS Lambda function that invokes RunCommand on the instance. Terminate hooks can be useful for collecting important data from an instance before it goes away. For example, you could use a terminate hook to preserve your fleet’s log files by copying them to an Amazon S3 bucket when instances go out of service.

Visit [lifecycle hooks](https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html) in our Amazon EC2 Auto Scaling User Guide for more information.

Q: What are the characteristics of an “unhealthy” instance?

An unhealthy instance is one where the hardware has become impaired for some reason (bad disk, etc.), or it is not passing a user-configured ELB health check. Amazon EC2 Auto Scaling performs [health checks](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html) on each individual EC2 instance at regular intervals, and if the instance is connected to an Elastic Load Balancing load balancer, it can also perform [ELB health checks](http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html).

Q: Can I customize a health check?

Yes, there is an API called *SetInstanceHealth*that allows you to change an instance's state to UNHEALTHY, which will then result in a termination and replacement.

Q: Can I suspend health checks (for example, to evaluate unhealthy instances)?

Yes, you can temporarily suspend Amazon EC2 Auto Scaling health checks by using the SuspendProcesses API. You can use the ResumeProcesses API to resume automatic health checks.

Q: Which health check type should I select?

If you are using Elastic Load Balancing (ELB) with your group, you should select an ELB health check. If you’re not using ELB with your group, you should select the EC2 health check.

Q: Can I use Amazon EC2 Auto Scaling for health checks and to replace unhealthy instances if I’m not using Elastic Load Balancing (ELB)?

You don't have to use ELB to use Auto Scaling. You can use the EC2 health check to identify and replace unhealthy instances.

Q: Do the Elastic Load Balancing (ELB) health checks work with Application Load Balancers and Network Load Balancers? Will an instance be marked as unhealthy if any target group associated with it becomes unhealthy?

Yes, Amazon EC2 Auto Scaling works with Application Load Balancers and Network Load Balancers including their health check feature.

Q: Is there any way to use Amazon EC2 Auto Scaling to only add a volume without adding an instance?

A volume is attached to a new instance when it is added. Amazon EC2 Auto Scaling doesn't automatically add a volume when the existing one is approaching capacity. You can use the EC2 API to add a volume to an existing instance.

Q: What does the term “stateful instances” refer to?

When we refer to a stateful instance, we mean an instance that has data on it, which exists only on that instance. In general, terminating a stateful instance means that the data (or state information) on the instance is lost. You may want to consider using lifecycle hooks to copy the data off of a stateful instance before it’s terminated, or enable instance protection to prevent Amazon EC2 Auto Scaling from terminating it.

Replacing Impaired Instances

Q: How does Amazon EC2 Auto Scaling replace an impaired instance?

When an impaired instance fails a health check, Amazon EC2 Auto Scaling automatically terminates it and replaces it with a new one. If you’re using an Elastic Load Balancing load balancer, Amazon EC2 Auto Scaling gracefully detaches the impaired instance from the load balancer before provisioning a new one and attaching it to the load balancer. This is all done automatically, so you don’t need to respond manually when an instance needs replacing.

Q: How do I control which instances Amazon EC2 Auto Scaling terminates when scaling in, and how do I protect data on an instance?

With each Amazon EC2 Auto Scaling group, you control when Amazon EC2 Auto Scaling adds instances (referred to as scaling out) or remove instances (referred to as scaling in) from your group. You can scale the size of your group manually by attaching and detaching instances, or you can automate the process through the use of a scaling policy. When you have Amazon EC2 Auto Scaling automatically scale in, you must decide which instances Amazon EC2 Auto Scaling should terminate first. You can configure this through the use of a termination policy. You can also use instance protection to prevent Amazon EC2 Auto Scaling from selecting specific instances for termination when scaling in. If you have data on an instance, and you need that data to be persistent even if your instance is scaled in, then you can use a service like S3, RDS, or DynamoDB, to make sure that it is stored off the instance.

Q: How long is the turn-around time for Amazon EC2 Auto Scaling to spin up a new instance at inService state after detecting an unhealthy server?

The turnaround time is within minutes. The majority of replacements happen within less than 5 minutes, and on average it is significantly less than 5 minutes. It depends on a variety of factors, including how long it takes to boot up the AMI of your instance.

Q: If Elastic Load Balancing (ELB) determines that an instance is unhealthy, and moved offline, will the previous requests sent to the failed instance be queued and rerouted to other instances within the group?

When ELB notices that the instance is unhealthy, it will stop routing requests to it. However, prior to discovering that the instance is unhealthy, some requests to that instance will fail.

Q: If you don’t use Elastic Load Balancing (ELB) how would users be directed to the other servers in a group if there was a failure?

You can integrate with Route53 (which Amazon EC2 Auto Scaling does not currently support out of the box, but many customers use). You can also use your own reverse proxy, or for internal microservices, can use service discovery solutions.

Security

Q: How do I control access to Amazon EC2 Auto Scaling resources?

Amazon EC2 Auto Scaling integrates with [AWS Identity and Access Management](https://aws.amazon.com/iam/) (IAM), a service that enables you to do the following:

Create users and groups under your organization's AWS account

Assign unique security credentials to each user under your AWS account

Control each user's permissions to perform tasks using AWS resources

Allow the users in another AWS account to share your AWS resources

Create roles for your AWS account and define the users or services that can assume them

Use existing identities for your enterprise to grant permissions to perform tasks using AWS resources

For example, you could create an IAM policy that grants the Managers group permission to use only the *DescribeAutoScalingGroups*, *DescribeLaunchConfigurations*, *DescribeScalingActivities*, and *DescribePolicies*API operations. Users in the Managers group could then use those operations with any Amazon EC2 Auto Scaling groups and launch configurations. With Amazon EC2 Auto Scaling resource-level permissions, you can restrict access to a particular EC2 Auto Scaling group or launch configuration.

For more information, see the [Controlling Access to Your Auto Scaling Resources](http://docs.aws.amazon.com/autoscaling/latest/userguide/control-access-using-iam.html) section of the Amazon EC2 Auto Scaling user guide.

Q: Can you define a default admin password on Windows instances with Amazon EC2 Auto Scaling?

You can use the Key Name parameter to CreateLaunchConfiguration to associate a key pair with your instance. You can then use the *GetPasswordData*API in EC2. This is also possible through the AWS Management Console.

Q: Are CloudWatch agents automatically installed on EC2 instances when you create an Amazon EC2 Auto Scaling group?

If your AMI contains a CloudWatch agent, it’s automatically installed on EC2 instances when you create an EC2 Auto Scaling group. With the stock Amazon Linux AMI, you need to install it (recommended, via yum).

Cost Optimization

Q: Can I create a single ASG to scale instances across different purchase options?

Yes. You can provision and automatically scale EC2 capacity across different EC2 instance types, Availability Zones, and On-Demand, RIs and Spot purchase options in a single Auto Scaling Group. You have the option to define the desired split between On-Demand and Spot capacity, select which instance types work for your application, and specify preference for how EC2 Auto Scaling should distribute the ASG capacity within each purchasing model.

Q: Can I use ASGs to launch and manage just Spot Instances or just On-Demand instances and RIs?

Yes. You can configure your ASG specifying all capacity to be only Spot instances or all capacity to be only On-Demand instances and RIs.

Q: Can I have a base capacity with On-Demand instances and RIs, and scale my ASG out on Spot instances?

Yes. When setting up an ASG to combine purchasing models, you can specify the base capacity of the group to be fulfilled by On-Demand instances. As the ASG scales in or scale out, EC2 Auto Scaling ensures the base capacity be fulfilled with On-Demand instances and anything beyond that be fulfilled with either only Spot instances or a specified percentage mix of On-Demand or Spot instances.

Q: Can I modify the configuration of an ASG to update the different properties pertaining to combining purchasing models and specifying multiple instance types?

Yes. Similar to other ASG parameters, customers can update an existing ASG to modify one or all parameters pertaining to combining purchasing models and specifying multiple instance types, including instance types, prioritization order for On-Demand instances, percentage split between On-Demand and Spot instances, and allocation strategy.

Q: Can I use RI discounts with On-Demand Instances in an ASG?

Yes. For example, if you have RIs for C4 instances and EC2 Auto Scaling launches a C4 you will receive your RI pricing for On-Demand Instances.

Q: Can I specify instances of different sizes (CPU cores, memory) in my Auto Scaling group?

Yes. You can specify any instance type available in a region. Additionally, you can specify an optional weight for each instance type, which defines the capacity units that each instance would contribute to your application’s performance.

Q: What if the instance types I like are not available in an Availability Zone?

If none of the specified instance types are available in an Availability Zone, Auto Scaling will retarget the launches in other Availability Zones associated with the Auto Scaling group. Auto Scaling will always prefer keeping your compute balanced across Availability Zones and retarget if all instance types are not available in an Availability Zone.

Pricing

Q: What are the costs for using Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling fleet management for EC2 instances carries no additional fees. The dynamic scaling capabilities of Amazon EC2 Auto Scaling are enabled by Amazon CloudWatch and carry no additional fees. Amazon EC2 and Amazon CloudWatch service fees apply and are billed separately.