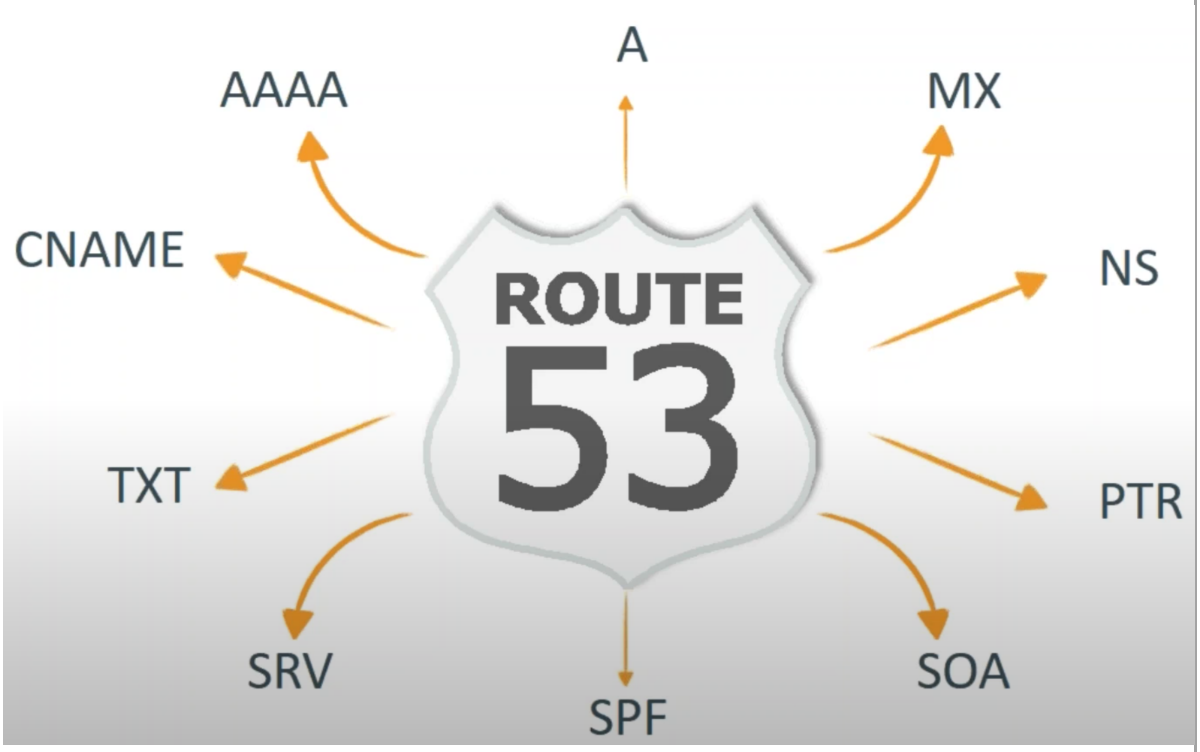
Route53

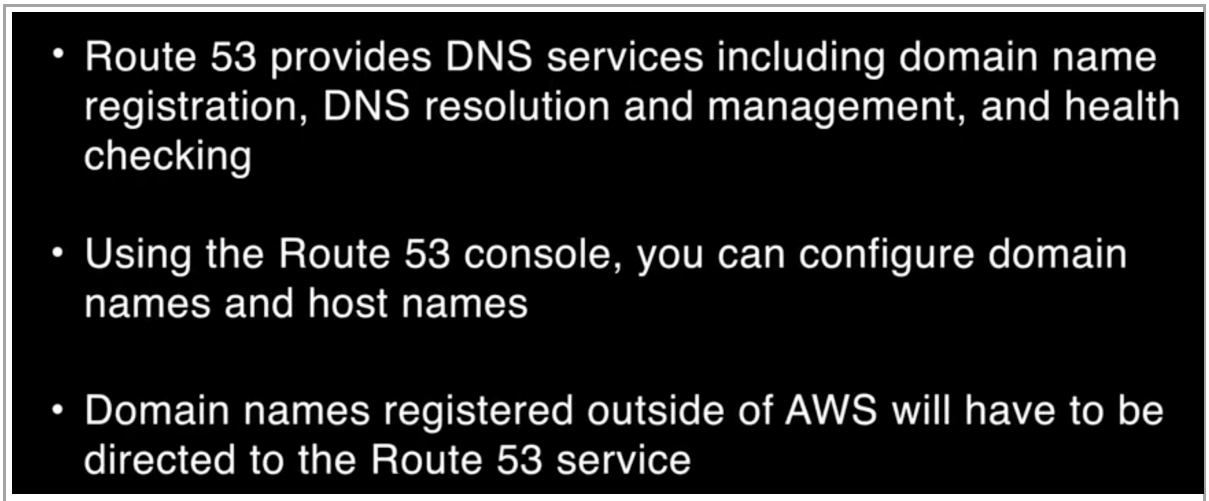
Reference Links
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-elb-load-balancer.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html

https://tutorialsdojo.com/aws-cheat-sheet-amazon-route-53/



Routing Policies
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html



Geolocation based Routing

- Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.
- For example, you might want all queries from Europe to be routed to an ELB load balancer in the Frankfurt region.
- When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users.
- You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights.
- Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint.

How do I direct traffic to specific resources or AWS regions based on the query's geographic location?

https://aws.amazon.com/premiumsupport/knowledge-center/geolocation-routing-policy

Routing Policy Options

When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

- **Simple routing policy**
  - – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
- **Failover routing policy**
  - – Use when you want to configure active-passive failover.
- **Geolocation routing policy**
  - – Use when you want to route traffic based on the location of your users.

- **Geoproximity routing policy**
  - – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- **Latency routing policy**
  - – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency.
- **Multivalue answer routing policy**
  - – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
- **Weighted routing policy**
  - – Use to route traffic to multiple resources in proportions that you specify.

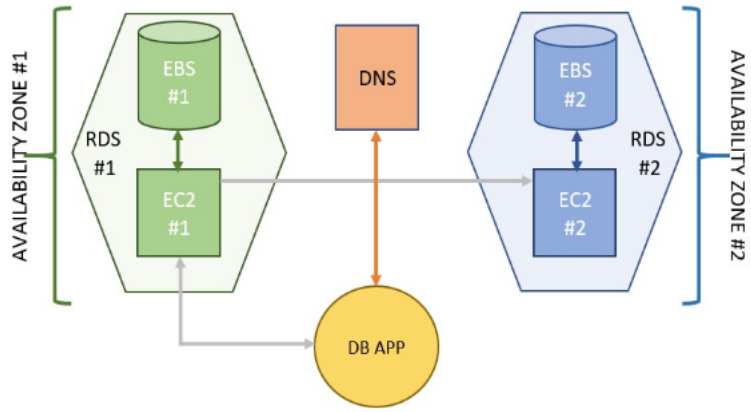
ROUTE 53 - FAILOVER setup

You can use Route 53 health checking to configure active-active and active-passive failover configurations.

You configure active-active failover using any routing policy (or combination of routing policies) other than failover, and

you configure active-passive failover using the failover routing policy.

- **Active-Active Failover**
  - Use this failover configuration when you want all of your resources to be available the majority of the time. When a resource becomes unavailable, Route 53 can detect that it's unhealthy and stop including it when responding to queries.
  - In active-active failover, all the records that have the same name, the same type (such as A or AAAA), and the same routing policy (such as weighted or latency) are active unless Route 53 considers them unhealthy.
  - Route 53 can respond to a DNS query using any healthy record.
  - DNS active-active failover allows access to your unhealthy instances to be redirected to active instances. Together with latency-based routing, customers accessing your web servers will be balanced throughout available healthy instances based on latency.
  -



- **Active-Passive Failover**
  - Use an active-passive failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable.
  - When responding to queries, Route 53 includes only the healthy primary resources. If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-types.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-configuring.html

Reference Links

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-route-53/

Latency Routing vs Geoproximity Routing vs Geolocation Routing:

https://tutorialsdojo.com/aws-cheat-sheet-latency-routing-vs-geoproximity-routing-vs-geolocation-routing/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services-for-udemy-students/

FAIL-OVER STRATEGIES

Creating Amazon Route 53 health checks and configuring DNS failover

Reference:

https://aws.amazon.com/premiumsupport/knowledge-center/fail-over-s3-r53/

http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html

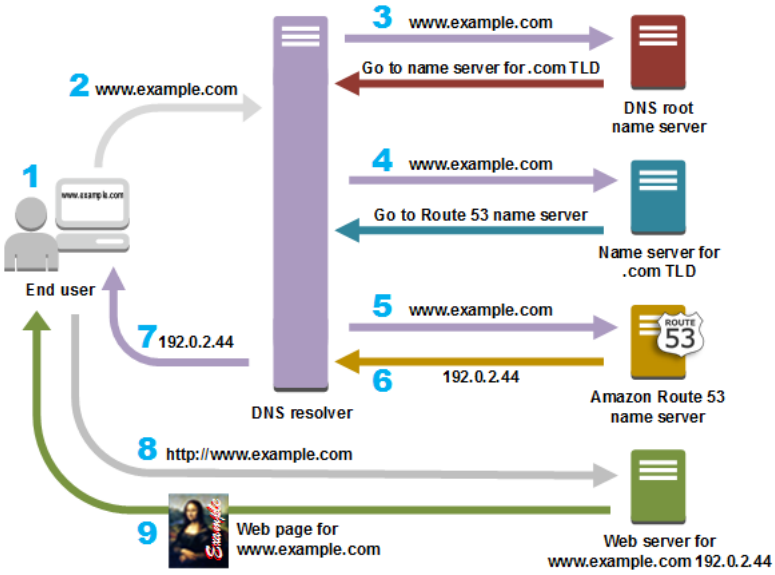
Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-route-53/

ROUTING TRAFFIC TO ELB

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-elb-load-balancer.html

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html



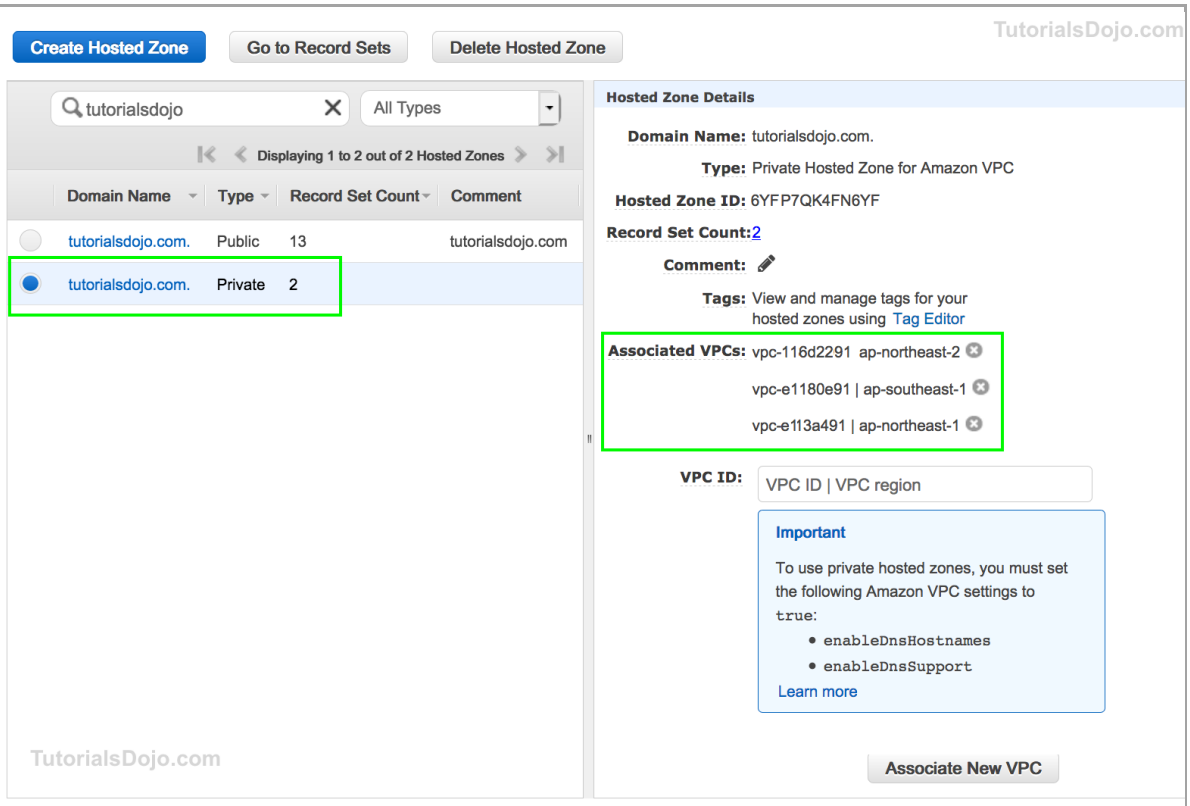
Route53 - Hosted Zones

In AWS, a hosted zone is a container for records, and records contain information about how you want to route traffic for a specific domain, such as tutorialsdojo.com, and its subdomains (portal.tutorialsdojo.com, database.tutorialsdojo.com).

A hosted zone and the corresponding domain have the same name. There are two types of hosted zones:

Public hosted zones contain records that specify how you want to route traffic on the internet.

Private hosted zones contain records that specify how you want to route traffic in an Amazon VPC



A *private hosted zone* is a container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs that you create with the Amazon VPC service.

Your VPC has attributes that determine whether your EC2 instance receives public DNS hostnames, and whether DNS resolution through the Amazon DNS server is supported.

enableDnsHostnames - Indicates whether the instances launched in the VPC get public DNS hostnames. If this attribute is true, instances in the VPC get public DNS hostnames, but only if the enableDnsSupport attribute is also set to true.

enableDnsSupport - Indicates whether the DNS resolution is supported for the VPC. If this attribute is false, the Amazon-provided DNS server in the VPC that resolves public DNS hostnames to IP addresses is not enabled. If this attribute is true, queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC IPv4 network range plus two (*.*.*.2) will succeed.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html#vpc-dns-support

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted-zones-private.html

Check out these Amazon VPC and Route 53 Cheat Sheets:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-vpc/

https://tutorialsdojo.com/aws-cheat-sheet-amazon-route-53/

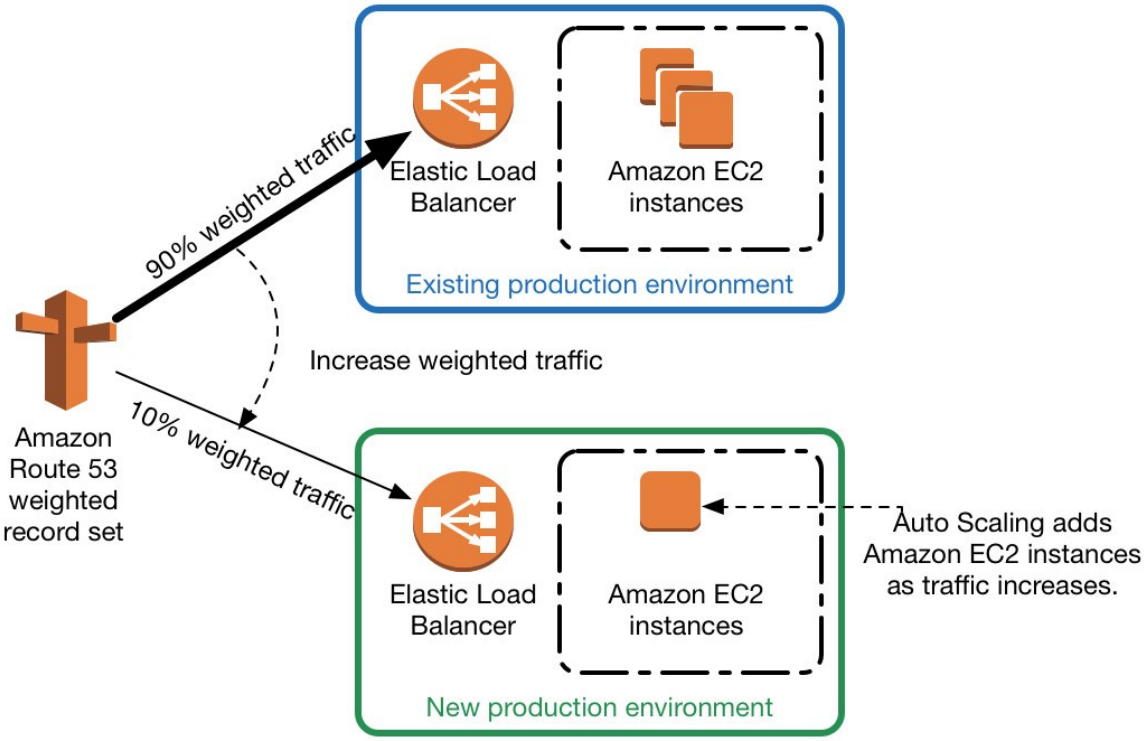
Route53 - Alias Record

To route domain traffic to an ELB load balancer, use Amazon Route 53 to create an alias record that points to your load balancer.
An alias record is a Route 53 extension to DNS.
It's similar to a CNAME record, but **you can create an alias record both for the root domain, such as example.com, and for subdomains**, such as www.example.com. (**You can create CNAME records only for subdomains**).

- For EC2 instances, always use a Type A Record without an Alias.
- For ELB, Cloudfront and S3, always use a Type A Record with an Alias and ,
- for RDS, always use the CNAME Record with no Alias

Route53 - Alias Record

If you host a website on multiple Amazon EC2 instances, you can distribute traffic to your website across the instances by using an Elastic Load Balancing (ELB) load balancer.
The ELB service automatically scales the load balancer as traffic to your website changes over time.
The load balancer can also monitor the health of its registered instances and route domain traffic only to healthy instances.



To route domain traffic to an ELB load balancer, use Amazon Route 53 to create an alias record that points to your load balancer.
An alias record is a Route 53 extension to DNS.

It's similar to a CNAME record, but you can create an alias record both for the root domain, such as example.com, and for subdomains, such as www.example.com. (You can create CNAME records only for subdomains.)

The following option is correct

- **Launch a new Elastic Load Balancer (ELB).**
- **Place all the EC2 instances behind the ELB.**
- **In Route53, create an alias A record that points to your ELB.**

You should use the CNAME of the ELB instead of its A record if using Non-Alias option.

Route53 - Alias Record

Amazon Route 53 *alias records* provide a Route 53–specific extension to DNS functionality. Alias records let you route traffic to selected AWS resources, such as CloudFront distributions and Amazon S3 buckets. They also let you route traffic from one record in a hosted zone to another record.

Unlike a CNAME record, you can create an alias record at the top node of a DNS namespace, also known as the *zone apex*. For example, if you register the DNS name tutorialsdojo.com, the zone apex is tutorialsdojo.com.

You can't create a CNAME record for tutorialsdojo.com, but you can create an alias record for tutorialsdojo.com that routes traffic to **www**.tutorialsdojo.com (take note of the **www** subdomain).

You can also type the domain name for the resource. For example:

- CloudFront distribution domain name: dtut0rial5d0j0.cloudfront.net

- Elastic Beanstalk environment CNAME: tutorialsdojo.elasticbeanstalk.com

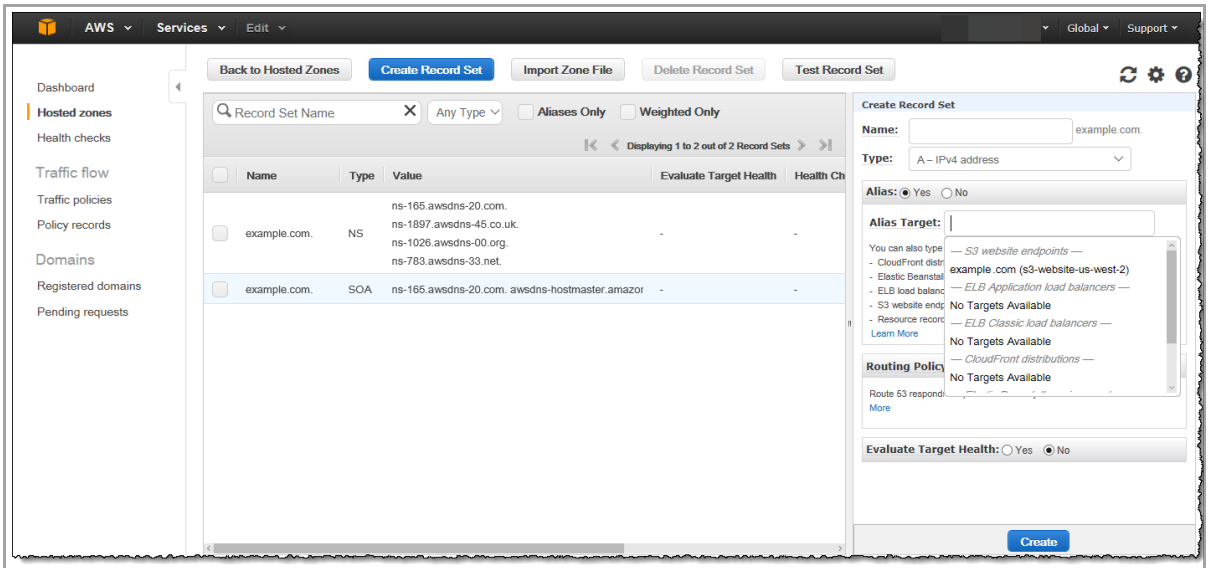
- ELB load balancer DNS name:tutorialsdojo-1.us-east-2.elb.amazonaws.com

- S3 website endpoint: s3-website.us-east-2.amazonaws.com

- Resource record set in this hosted zone: www.tutorialsdojo.com

- VPC endpoint: tutorialsdojo.us-east-2.vpce.amazonaws.com

- API Gateway custom regional API: d-tut5d0j0c0m.execute-api.us-west-2.amazonaws.com



Multi-value answer routing

- Multivalue answer routing lets you configure Amazon Route 53 to return multiple values, such as IP addresses for your web servers, in response to DNS queries.
- You can specify multiple values for almost any record, but multivalue answer routing also lets you check the health of each resource, so Route 53 returns only values for healthy resources.
- It's not a substitute for a load balancer, but the ability to return multiple health-checkable IP addresses is a way to use DNS to improve availability and load balancing.

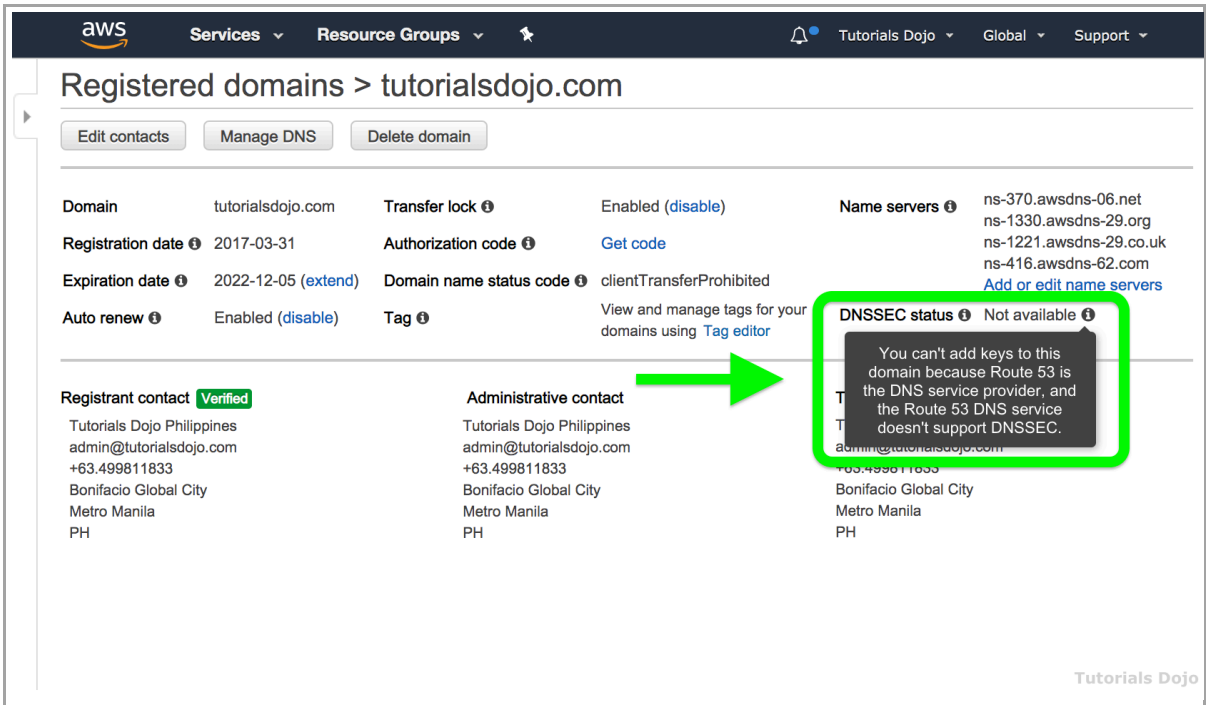
References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-multivalue

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html

Route53 - DNS Spoofing Control

Attackers sometimes hijack traffic to internet endpoints such as web servers by intercepting DNS queries and returning their own IP addresses to DNS resolvers in place of the actual IP addresses for those endpoints.
Users are then routed to the IP addresses provided by the attackers in the spoofed response, for example, to fake websites.
You can protect your domain from this type of attack, known as DNS spoofing or a man-in-the-middle attack, by configuring Domain Name System Security Extensions (DNSSEC), a protocol for securing DNS traffic.



Amazon Route 53

- supports DNSSEC for domain registration.
- does not support DNSSEC for DNS service, regardless of whether the domain is registered with Route 53.

If you want to configure DNSSEC for a domain that is registered with Route 53, you must either use another DNS service provider or set up your own DNS server.

- For the web domain registration, use Amazon Route 53 and then register a 2048-bit RSASHA256 encryption key from a third-party certificate service.
- Enable Domain Name System Security Extensions (DNSSEC) by using a 3rd party DNS provider that uses customer managed keys.
- Register the SSL certificates in ACM and attach them to the Application Load Balancer (your ELB site: global e-commerce marketplace).
- Configure the Server Name Identification extension in all user requests to the website.

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-configure-dnssec.html

https://aws.amazon.com/route53/faqs/

https://aws.amazon.com/premiumsupport/knowledge-center/r53-private-ubuntu/

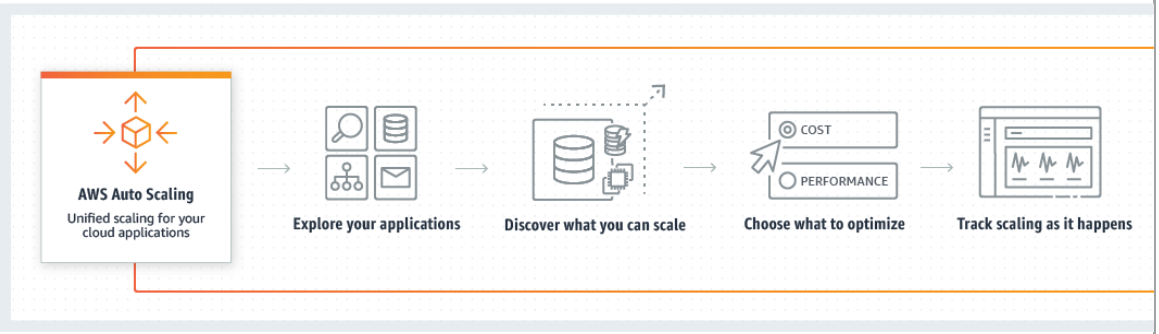
Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-route-53/

AWS - Autoscaling Setup

Launch Configs

- A launch configuration is a template that an 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



- You can specify your launch configuration with multiple Auto Scaling groups.
- However, you can only specify one launch configuration for an 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 an Auto Scaling group, you must create a launch configuration and then update your Auto Scaling group with the new launch configuration.

References:

http://docs.aws.amazon.com/autoscaling/latest/userguide/LaunchConfiguration.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html

Check out this AWS Auto Scaling Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-auto-scaling/

AWS - Autoscaling TYPES

Autoscaling Policies

Amazon EC2 Auto Scaling supports the following types of scaling policies:

- **Target tracking scaling**
- **Step scaling**
- **Simple scaling**

Target tracking scaling -

- Increase or decrease the current capacity of the group based on a target value **for a specific metric**.
- This is similar to the way that your thermostat maintains the temperature of your home – you select a temperature and the thermostat does the rest.

Step scaling -

- Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as *step adjustments*, that vary based on the size of the alarm breach.

Simple scaling -

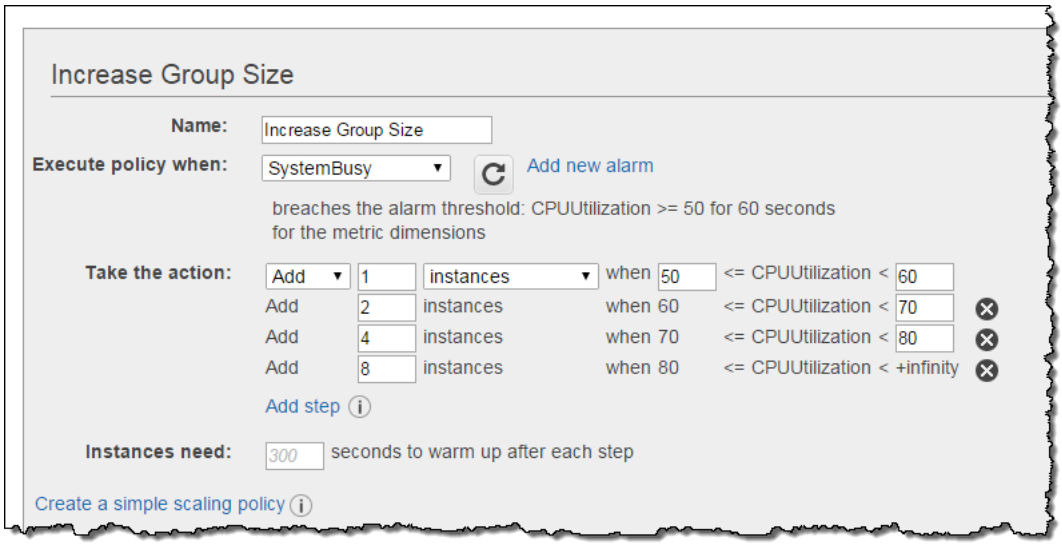
- Increase or decrease the current capacity of the group based on a single scaling adjustment.

If you are scaling based on a utilization metric that increases or decreases proportionally to the number of instances in an Auto Scaling group, then it is recommended that you use target tracking scaling policies.
Otherwise, it is better to use step scaling policies instead.

STEP Scaling

- With step scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process as well as define how your scalable target should be scaled when a threshold is in breach for a specified number of evaluation periods.
- Step scaling policies increase or decrease the current capacity of a scalable target based on a set of scaling adjustments, known as step adjustments.
- The adjustments vary based on the size of the alarm breach.
- After a scaling activity is started, the policy continues to respond to additional alarms, even while a scaling activity is in progress.
- Therefore, all alarms that are breached are evaluated by Application Auto Scaling as it receives the alarm messages.

- When you configure dynamic scaling, you must define how to scale in response to changing demand.
  - For example, you have a web application that currently runs on two instances and you want the CPU utilization of the Auto Scaling group to stay at around 50 percent when the load on the application changes.
  - This gives you extra capacity to handle traffic spikes without maintaining an excessive amount of idle resources.
  - You can configure your Auto Scaling group to scale automatically to meet this need. The policy type determines how the scaling action is performed.



References:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html

https://docs.aws.amazon.com/autoscaling/application/userguide/application-auto-scaling-step-scaling-policies.html

NOTE:

- **Scheduled Scaling** is incorrect because the scheduled scaling policy is based on a schedule that allows you to set your own scaling schedule for **predictable** load changes.
- This is not considered as one of the types of dynamic scaling.

AWS - ALB Routing Configs

PATH BASED ROUTING

- You can use path conditions to define rules that forward requests to different target groups based on the URL in the request (also known as *path-based routing*).
- This type of routing is the most appropriate solution for this scenario hence, **using path conditions to define rules that forward requests to different target groups based on the URL in the request** is the correct answer.

- Each path condition has one path pattern.
- If the URL in a request matches the path pattern in a listener rule exactly, the request is routed using that rule.

- A path pattern is case-sensitive, can be up to 128 characters in length, and can contain any of the following characters. You can include up to three wildcard characters.

  A–Z, a–z, 0–9

  _ - . $ / ~ " ' @ : +

  & (using &)

  * (matches 0 or more characters)

  ? (matches exactly 1 character)

  Example path patterns

  /img/*

  /js/*

References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html#path-conditions

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-elastic-load-balancing-elb/

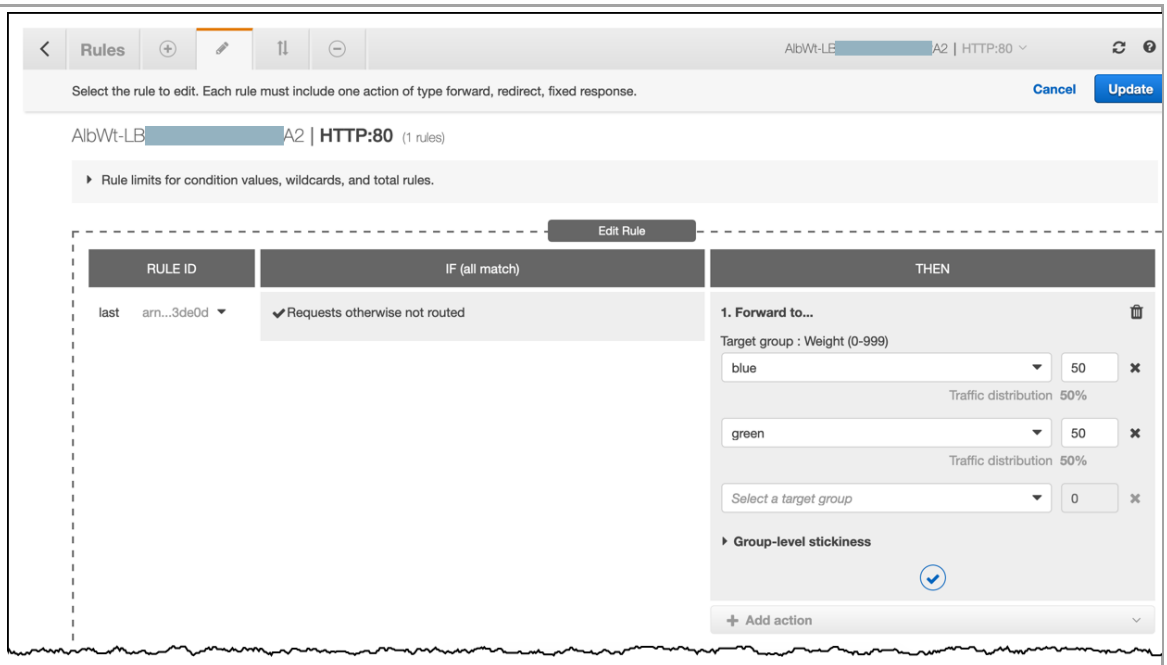
Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

https://tutorialsdojo.com/aws-cheat-sheet-application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/

AWS - Weighted Ruling ALB

ALB-Weighted rule

- Application Load Balancers support Weighted Target Groups routing.
- With this feature, you will be able to do weighted routing of the traffic forwarded by a rule to multiple target groups.
- This enables various use cases like blue-green, canary and hybrid deployments without the need for multiple load balancers.
- It even enables zero-downtime migration between on-premises and cloud or between different compute types like EC2 and Lambda.



ROUTE-53 weighted

- To divert 50% of the traffic to the new application in AWS and the other 50% to the application, you can also use Route 53 with Weighted routing policy.
- This will divert the traffic between the on-premises and AWS-hosted application accordingly.
- Weighted routing lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (portal.tutorialsdojo.com) and choose how much traffic is routed to each resource.
- This can be useful for a variety of purposes, including load balancing and testing new versions of software. You can set a specific percentage of how much traffic will be allocated to the resource by specifying the weights.
- For example,
  - if you want to send a tiny portion of your traffic to one resource and the rest to another resource, you might specify weights of 1 and 255.
  - The resource with a weight of 1 gets 1/256th of the traffic (1/1+255), and the other resource gets 255/256ths (255/1+255).
  - You can gradually change the balance by changing the weights. If you want to stop sending traffic to a resource, you can change the weight for that record to 0.

When you create a target group in your Application Load Balancer, you specify its target type.

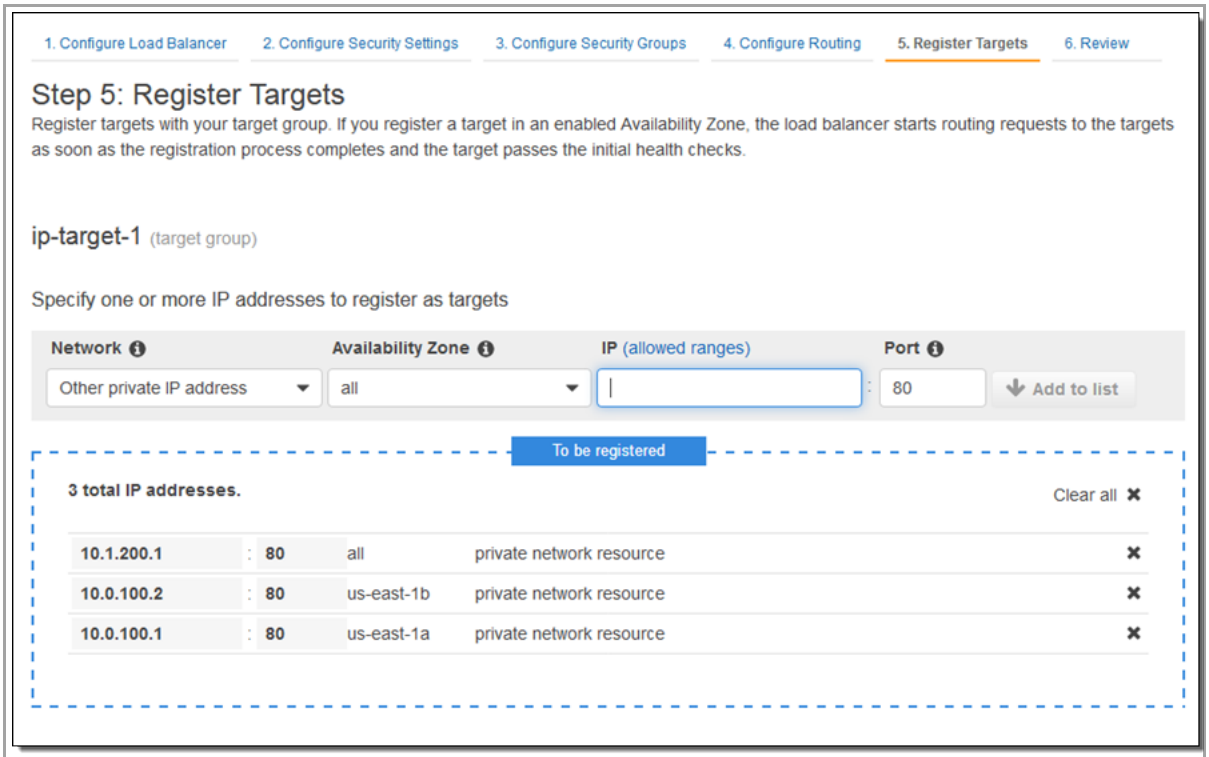
This determines the type of target you specify when registering with this target group.

You can select the following target types:

1. **instance** - The targets are specified by instance ID.

2. **ip** - The targets are IP addresses.

3. **Lambda** - The target is a Lambda function.



When the target type is **ip**, you can specify IP addresses from one of the following CIDR blocks:

- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)
- The subnets of the VPC for the target group

- These supported CIDR blocks enable you to register the following with a target group:
    - ClassicLink instances,
    - instances in a VPC that is peered to the load balancer VPC,
    - AWS resources that are addressable by IP address and port (for example, databases), and
    - on-premises resources linked to AWS through AWS Direct Connect or a VPN connection.
- Take note that you can not specify publicly routable IP addresses.
- If you specify targets using an instance ID, traffic is routed to instances using the primary private IP address specified in the primary network interface for the instance.
- If you specify targets using IP addresses, you can route traffic to an instance using any private IP address from one or more network interfaces.

- This enables multiple applications on an instance to use the same port. Each network interface can have its own security group.

References:

http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html

https://aws.amazon.com/blogs/aws/new-application-load-balancer-simplifies-deployment-with-weighted-target-groups/

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html

Check out this Amazon Route 53 Cheat Sheet:

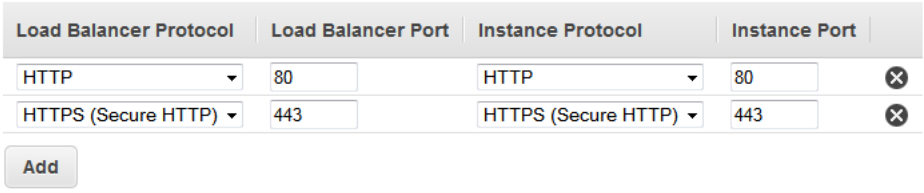
https://tutorialsdojo.com/aws-cheat-sheet-amazon-route-53/

AWS - ELB Health Checks

ELB Health checks

- A load balancer takes requests from clients and distributes them across the EC2 instances that are registered with the load balancer.
- You can create a load balancer that listens on both the HTTP (80) and HTTPS (443) ports.
- If you specify that the HTTPS listener sends requests to the instances on port 80, the load balancer terminates the requests and communication from the load balancer to the instances is not encrypted.
- If the HTTPS listener sends requests to the instances on port 443, communication from the load balancer to the instances is encrypted.

Monitoring ALB
https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-monitoring.html



- If your load balancer uses an encrypted connection to communicate with the instances, you can optionally enable authentication of the instances.
- This ensures that the load balancer communicates with an instance only if its public key matches the key that you specified to the load balancer for this purpose.

- The type of ELB that is mentioned in this scenario is an Application Elastic Load Balancer.

- This is used if you want a flexible feature set for your web applications with HTTP and HTTPS traffic.
- Conversely, it only allows 2 types of health check: HTTP and HTTPS.

Monitor your Application Load Balancers

You can use the following features to monitor your load balancers, analyze traffic patterns, and troubleshoot issues with your load balancers and targets.

CloudWatch metrics

- You can use Amazon CloudWatch to retrieve statistics about data points for your load balancers and targets as an ordered set of time-series data, known as *metrics*.
- You can use these metrics to verify that your system is performing as expected. For more information, see CloudWatch metrics for your Application Load Balancer.

Access logs

- You can use access logs to capture detailed information about the requests made to your load balancer and store them as log files in Amazon S3.
- You can use these access logs to analyze traffic patterns and to troubleshoot issues with your targets.
- For more information, see Access logs for your Application Load Balancer.

Request tracing

- You can use request tracing to track HTTP requests.
- The load balancer adds a header with a trace identifier to each request it receives.
- For more information, see Request tracing for your Application Load Balancer.

CloudTrail logs

- You can use AWS CloudTrail to capture detailed information about the calls made to the Elastic Load Balancing API and store them as log files in Amazon S3.
- You can use these CloudTrail logs to determine which calls were made, the source IP address where the call came from, who made the call, when the call was made, and so on.
- For more information, see Logging API calls for your Application Load Balancer using AWS CloudTrail.

References:

http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-elastic-load-balancing-elb/

EC2 Instance Health Check vs ELB Health Check vs Auto Scaling and Custom Health Check:

https://tutorialsdojo.com/aws-cheat-sheet-ec2-instance-health-check-vs-elb-health-check-vs-auto-scaling-and-custom-health-check-2/

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

https://tutorialsdojo.com/aws-cheat-sheet-application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services-for-udemy-students/

ELB - Securing Transfer

Elastic Load Balancing – Perfect Forward Secrecy and more new security features
https://aws.amazon.com/about-aws/whats-new/2014/02/19/elastic-load-balancing-perfect-forward-secrecy-and-more-new-security-features/

Elastic Load Balancing, there are various security features that you can use such as

- Predefined Security Policy,
- Server Order Preference,
- Perfect Forward Secrecy and
- many others

Perfect Forward Secrecy

- Perfect Forward Secrecy is a feature that provides additional safeguards against the eavesdropping of encrypted data, through the use of a unique random session key.
- This prevents the decoding of captured data, even if the secret long-term key is compromised.
- CloudFront and Elastic Load Balancing are the two AWS services that support Perfect Forward Secrecy.

- Hence, the correct answer is: **CloudFront and Elastic Load Balancing**

Server Order Preference

- **Server Order Preference** lets you configure the load balancer to enforce cipher ordering, providing more control over the level of security used by clients to connect with your load balancer

Predefined Security Policy

- The new **Predefined Security Policy** simplifies the configuration of your load balancer by providing a recommended cipher suite that adheres to AWS security best practices.
- The policy includes the latest security protocols (TLS 1.1 and 1.2), enables Server Order Preference, and offers high security ciphers such as those used for Elliptic Curve signatures and key exchanges.

Secure Content Delivery with Amazon CloudFront

https://d1.awsstatic.com/whitepapers/Security/Secure_content_delivery_with_CloudFront_whitepaper.pdf

ELB - Cheat Sheet
https://tutorialsdojo.com/aws-cheat-sheet-aws-elastic-load-balancing-elb/

- .

ELB - SNI with Multiple TLS

SNI - Server Name Indication

- SNI Custom SSL relies on the SNI extension of the Transport Layer Security protocol, which allows multiple domains to serve SSL traffic over the same IP address by including the hostname which the viewers are trying to connect to.
- You can host multiple TLS secured applications, each with its own TLS certificate, behind a single load balancer.
- In order to use SNI, all you need to do is bind multiple certificates to the same secure listener on your load balancer.
- ALB will automatically choose the optimal TLS certificate for each client.

- These features are provided at no additional charge.

References:

https://aws.amazon.com/blogs/aws/new-application-load-balancer-sni/

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cnames-https-dedicated-ip-or-sni.html#cnames-https-dedicated-ip

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html

Check out this Amazon CloudFront Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-cloudfront/

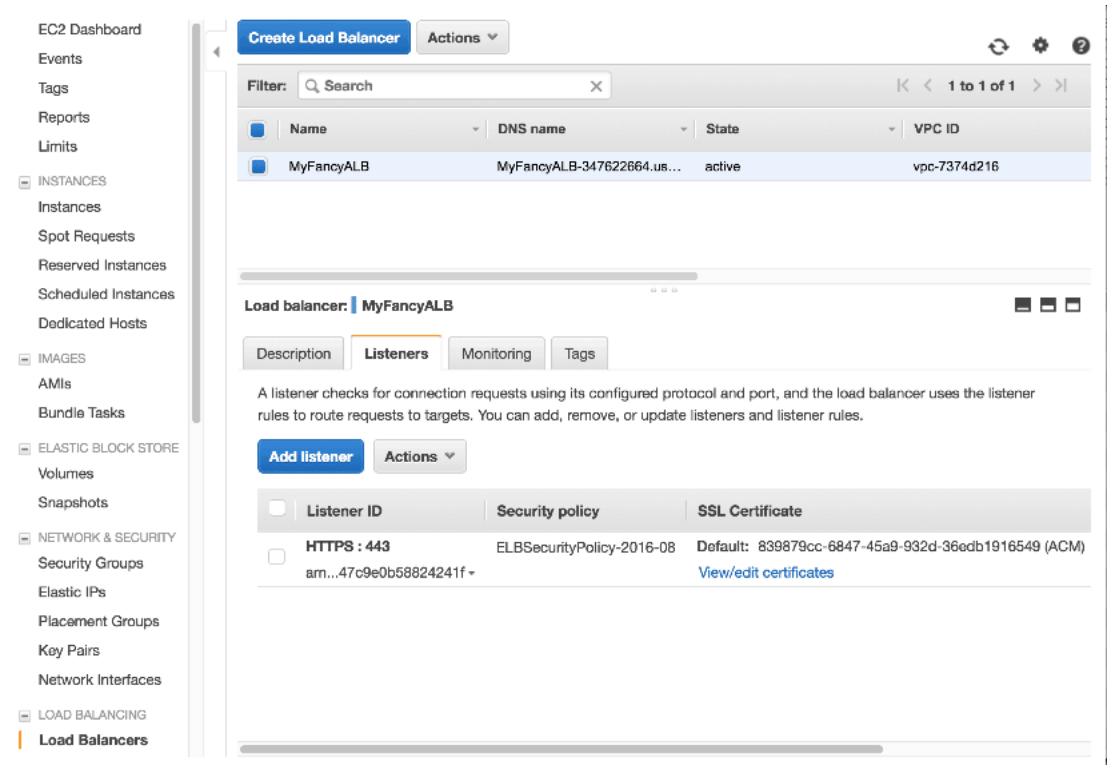
SNI Custom SSL vs Dedicated IP Custom SSL:

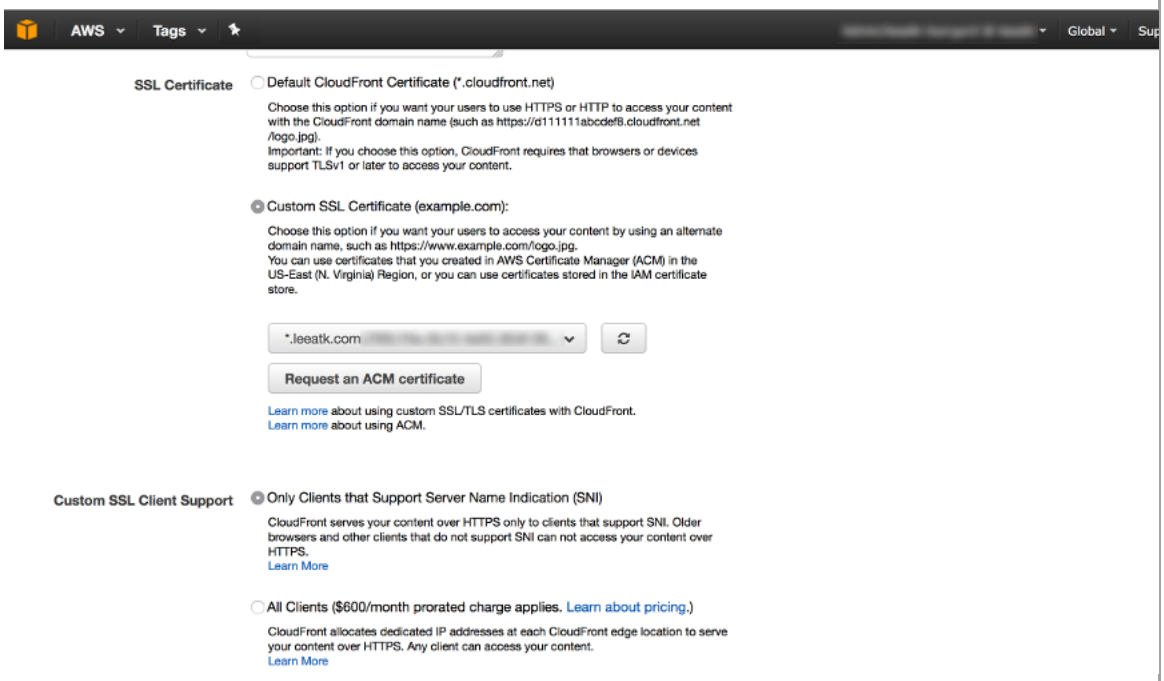
https://tutorialsdojo.com/aws-cheat-sheet-sni-custom-ssl-vs-dedicated-ip-custom-ssl/

Server Name Indication (SNI) Custom SSL	Dedicated IP Custom SSL
♦ Relies on the SNI extension of the TLS protocol, which allows multiple domains to serve SSL traffic over the same IP address. ♦ Offers the same level of security when using Dedicated IP Custom SSL. ♦ If you configure CloudFront to serve HTTPS requests using SNI, CloudFront associates your alternate domain name with an IP address for each edge location. The IP address to your domain name is determined during the SSL/TLS handshake negotiation, and isn't dedicated to your distribution. ♦ Some older browsers do not support SNI and will not be able to establish a connection with CloudFront to load the HTTPS version of your content. ♦ You can use SNI Custom SSL with no upfront or monthly fees for certificate management.	♦ Mainly useful for browsers that do not support SNI. ♦ For this feature, the Amazon content delivery network allocates dedicated IP addresses to serve your SSL content at each Edge location. ♦ You will need to upload a SSL certificate and associate it with your CloudFront distributions. ♦ You can associate more than two custom SSL certificate with your AWS Account by submitting a CloudFront Limit Increase Form. ♦ This method works for every HTTPS request, regardless of the browser or other viewer that the user is using. ♦ Because of the added cost associated with dedicating IP addresses per SSL certificate, AWS charges a fixed monthly fee of $600 for each custom SSL certificate you associate with your content delivery network distributions, pro-rated by the hour. ♦ You can switch to using a custom SSL/TLS certificate with SNI instead and eliminate the charge that is associated with dedicated IP addresses.

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services-for-udemy-students/





- You can upload all SSL certificates of the domains in the ALB using the console and bind multiple certificates to the same secure listener on your load balancer.
- ALB will automatically choose the optimal TLS certificate for each client using Server Name Indication (SNI).

- **Upload all SSL certificates of the domains in the ALB using the console and bind multiple certificates to the same secure listener on your load balancer.**

- **ALB will automatically choose the optimal TLS certificate for each client using Server Name Indication (SNI)**

OTHER ALTERNATE Option (INCORRECT)

- **Adding a Subject Alternative Name (SAN) for each additional domain to your certificate** is incorrect because although using SAN is correct, you will still have to reauthenticate and reprovision your certificate every time you add a new domain.
- One of the requirements in the scenario is that you should not have to reauthenticate and reprovision your certificate hence, this solution is incorrect.

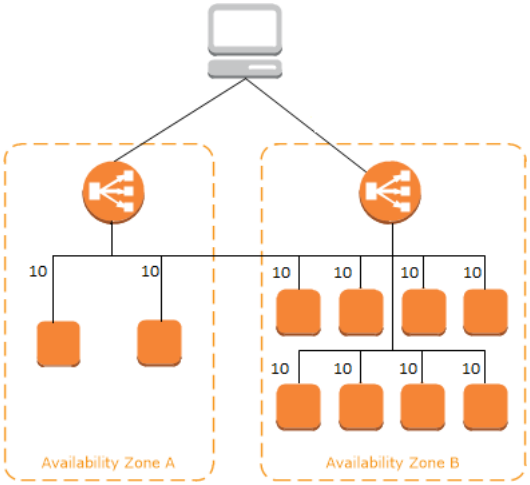
- Amazon CloudFront delivers your content from each edge location and offers the same security as the Dedicated IP Custom SSL feature.
- SNI Custom SSL works with most modern browsers, including Chrome version 6 and later (running on Windows XP and later or OS X 10.5.7 and later), Safari version 3 and later (running on Windows Vista and later or Mac OS X 10.5.6. and later), Firefox 2.0 and later, and Internet Explorer 7 and later (running on Windows Vista and later).
- Some users may not be able to access your content because some older browsers do not support SNI and will not be able to establish a connection with CloudFront to load the HTTPS version of your content.
- If you need to support non-SNI compliant browsers for HTTPS content, it is recommended to use the Dedicated IP Custom SSL feature.

ELB - Cross Zone LB

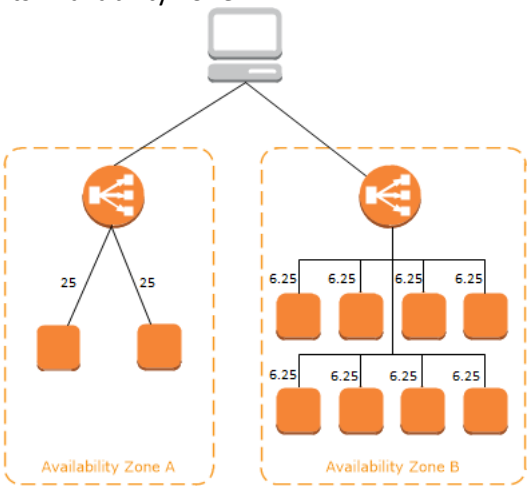
CROSS-ZONE Load Balancing

- Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances.
- When you create a Classic Load Balancer, the default for cross-zone load balancing depends on how you create the load balancer.
    - With the API or CLI, cross-zone load balancing is disabled by default.
    - With the AWS Management Console, the option to enable cross-zone load balancing is selected by default.
- After you create a Classic Load Balancer, you can enable or disable cross-zone load balancing at any time.
- The following diagrams demonstrate the effect of cross-zone load balancing.
- There are two enabled Availability Zones, with 2 targets in Availability Zone A and 8 targets in Availability Zone B. Clients send requests, and Amazon Route 53 responds to each request with the IP address of one of the load balancer nodes.
- This distributes traffic such that each load balancer node receives 50% of the traffic from the clients. Each load balancer node distributes its share of the traffic across the registered targets in its scope.

- If cross-zone load balancing is **enabled**, each of the 10 targets receives 10% of the traffic. This is because each load balancer node can route its 50% of the client traffic to all 10 targets.



- If cross-zone load balancing is **disabled**, each of the 2 targets in Availability Zone A receives 25% of the traffic and each of the 8 targets in Availability Zone B receives 6.25% of the traffic. This is because each load balancer node can route its 50% of the client traffic only to targets in its Availability Zone.



References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html#cross-zone-load-balancing

http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-disable-crosszone-lb.html

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-elastic-load-balancing-elb/