AWS Route 53

 | [Company address]

Study notes

Khan, Viquar

2017

**DNS Syntax Types Explained**

An “**A**” record, which stands for “address” is the most basic type of syntax used in DNS records, indicating the actual IP address of the domain.

The “**AAAA**” record is an IPV6 address record that maps a hostname to a 128-bit Ipv6 address.  Regular DNS addresses are mapped for 32-bit IPv4 addresses.

The “**CNAME**” record stands for “canonical name” and serves to make one domain an alias of another domain. CNAME is often used to associate new subdomains with an existing domain's DNS records.

The “**MX**” record stands for “mail exchange” and is basically a list of mail exchange servers that are to be used for the domain.

The “**PTR**” record stands for “pointer record” and maps an Ipv4 address to the CNAME on the host.

The “**NS**” record stands for “name server” and indicates which Name Server is authoritative for the domain.

An “**SOA**” record stands for “State of Authority” and is easily one of the most essential DSN records because it stores important information like when the domain was last updated and much more.

An “**SRV**” record stands for “service” and is used to define a TCP service on which the domain operates.

A “**TXT**” record lets the administrator insert any text they'd like into the DNS record, and it is often used for denoting facts about the domain.

## What AWS Route 53 brings to the DNS table

Amazon Route 53 is a highly available and scalable DNS service offered by AWS. Like any DNS service, Route 53 handles domain registration and routes users’ Internet requests to your application – whether it’s hosted on AWS or elsewhere.

But Route 53 also intelligently directs traffic based on sophisticated routing policies and, through automated health checks, away from servers that might be failing.

Like many AWS services, Route 53 is a pay-as-you-go service. You will be charged for the number of hosted zones you create and maintain and by the number of requests routed.

**Why you should consider migrating to Route 53**

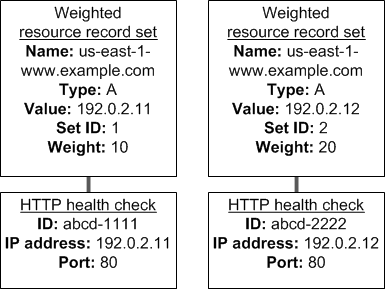
Once you’ve got your domain up and running, you can choose a routing policy that best fits your needs. To get the most out of the service however, you’ll need to properly understand the function of each policy type.

#### **Simple Routing Policy**

This is the most common and, as the name suggests, simplest routing type. If, say, your application server has a public IP address of 205.251.242.103, and you’d like to use the human readable address of amazon.com (assuming it’s available, of course), then you would use Route 53 to map amazon.com to your IP. From then on, any browser requests for amazon.com would be directed to 205.251.242.103.

#### **Weighted Routing Policy**

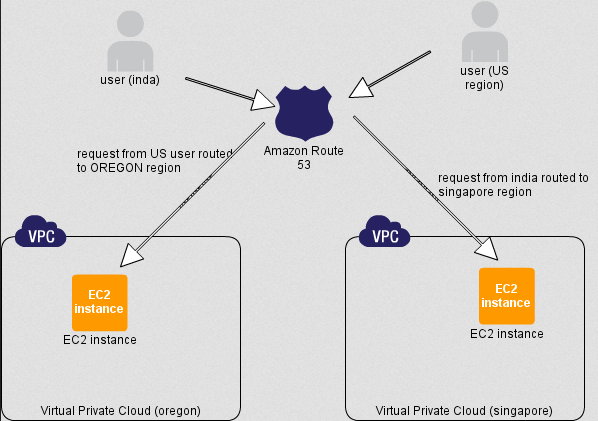
By assigning different numeric weights (or “priorities”) to multiple servers providing a web service, you can direct a higher or lower percentage of your incoming traffic to one particular server over another. This kind of routing can be useful for load balancing and testing new versions of a software package.



Example: If you want to move your 10% traffic to new version of your website and remaining 90 % on existing website.

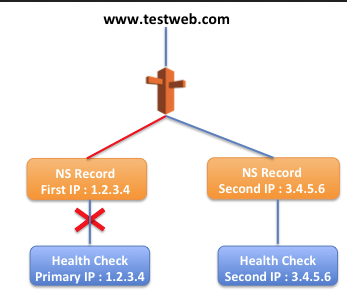
#### **Latency based Routing Policy**

A latency based policy directs traffic requests to the server that will be able to respond with the lowest possible latency (delay). You could, for instance, run your application in multiple AWS regions, and Route 53 will automatically route users to those that will deliver the quickest.



#### **Failover Routing Policy**

A failover policy will send all traffic to the server you set as primary for as long as that server is still healthy. If, however, health checks determine that it’s failing, traffic will be diverted to a designated backup resource.



#### **Geo location Routing Policy**

This policy lets you designate resource targets based on your users’ geographic location. So, for example, you might want all queries from India to be routed to a server located in the same physical region in order to limit latency.

### **Some Route 53 features**

**Domain Registration.**You can purchase domains from all top level domains (.com, .net, .org, etc.) directly from route 53, avoiding the need to migrate altogether. Route 53 also provides privacy protection for your WHOIS record at no extra charge.

**Private DNS.**Private DNS records let you create private hosted zones and route traffic using easily managed domain names within your VPCs. This can, for instance, allow you to quickly switch between IP-based resources without the need to update multiple embedded links.

**Health Check.** Route 53 monitors the health of your application and, when an outage is detected, redirects users to a healthy resource.

**AWS service integration.**Route 53 is tightly integrated with Elastic Load Balancing (ELB), CloudFront, and S3. You can easily route traffic to an ELB CNAME record, a static website hosted on S3, or generate custom domains for your CloudFront URLs.

**Alias Records.**Instead of an IP address, an alias resource record can, for instance, point directly to a CloudFront distribution, ELB load balancer, or Amazon S3 bucket. This way, even if the IP addresses of the underlying resources should change, traffic will still be sent to the correct endpoint.

### **Migration options**

1. **Leave your domain name with your own registrar**

If you would like to use Route 53’s routing features, but have no need to move your domain name from its current registrar, you simply need give your registrar the new name server addresses you’ll get from Route 53 upon creating a hosted zone. Once you have updated your name servers with your own providers, Route 53 will ensure the routing of all new domain requests through its own name servers.

Example :Godaddy you can go and add your ELB server ip address

1. **Migrate your domain name to Route 53**

When you decided to migrate, you will need get the DNS record data from your DNS provider. You will then import this data to a Route 53 hosted zone, and replace the registrar’s name server records with AWS name servers that you get by clicking on “**GethostedZone**”. Depending on your TTL settings, changes will take place within 24 to 48 hours.

* **Route 53** is AWS’ DNS service.
  + Route 53 supports all the standard DNS record types, but note that [**alias resource record sets**](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html) are not standard part of DNS, but a specific Route 53 feature. (It’s available from other DNS providers too, but each provider has a different name for it.)
  + Aliases are like an internal name (a bit like a CNAME) that is resolved internally on the server side. For example, traditionally you could have a CNAME to the DNS name of a CLB or ALB, but it’s often better to make an alias to the same load balancer. The effect is the same, but in the latter case, externally, all a client sees is the target the record points to.
  + It’s often wise to use alias record as an alternative to CNAMEs, since they can be updated instantly with an API call, without worrying about DNS propagation.
  + You can use them for CLBs/ALBs or any other resource where AWS supports it.
  + Somewhat confusingly, you can have CNAME and A aliases, depending on the type of the target.
  + Because aliases are extensions to regular DNS records, if exported, the output [zone file](https://en.wikipedia.org/wiki/Zone_file) will have additional non-standard “ALIAS” lines in it.
* [**Latency-based routing**](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-latency) allows users around the globe to be automatically directed to the nearest AWS region where you are running, so that latency is reduced.
* Understand that domain registration and DNS management (hosted zones) are two separate Route 53 services. When you buy/transfer a domain, Route 53 automaticaly assigns four name servers to it (e.g. ns-2.awsdns-00.com). Route 53 also offers to automatically create a hosted zone for DNS management, but you are not required do your DNS management in the same account or even in Route 53; you just need to create an NS record pointing to the servers assigned to your domain in Route 53.
  + One use case would be to put your domain registration (very mission critical) in a [bastion account](https://cloudonaut.io/your-single-aws-account-is-a-serious-risk/) while managing the hosted zones within another account which is accessible by your applications.

**Getting Started**

* DNS (Domain Name System) service which turns a website address into an IP address so that computers to connect with one another.
* Amazon Route 53 is a very scalable and available DNS which provides domain registration and health-checking web services. The service provides developers a cost-effective and reliable means to route users to their applications. Health-checking services can be combined with DNS to send traffic to healthy endpoints as well as monitor endpoints. Also, users can purchase domain names with Route 53. This service connects with others running AWS.
* Route 53 can monitor application performance and health. Also, Route 53 allows users to transfer existing domains into the management system.
* Users get started by signing up at the service page. Users with an existing domain name can use the management console or **CreateHostedZone API** to store the DNS records. **This will create a hosted zone and give users four name servers across four TLDs to ensure availability**. Users without an existing domain name can use the AWS management console or API to register one. A hosted zone is created and four name servers are given across four TLDs as well. Records in the hosted zone can be modified via the AWS management console.
* To ensure high availability and low latency, DNS servers are distributed globally. Route 53 uses a **global anycast**network to automatically answer queries from the location which provides optimal network conditions.
* When a hosted zone is created, it is assigned a server name by the system. This provides highly available service.
* Domains are the recognizable names you enter into an internet browser. They are different from a hosted zone which is a Route 53 term and concept. It is basically a file which collects records managed together; these records belong to a single domain name.
* Route 53 fees are based on usage. Users are charged for what they used and aren’t required to pay a minimum fee, make a commitment, and there are no overage fees. There is a calculator to estimate your bill each month.
* Using AWS IAM, you can control who has the ability to change DNS records and manage the assigned accounts’ permissions.
* **When signing up for Route 53, the activation process can take as much as 24 hours. If you haven’t received your confirmation email within this time frame, contact customer service.**
* Route 53 offers credits for customers who’s monthly uptime does not meet the percentage in the service commitment via the SLA.

Domain Name System (DNS)

* **Anycast is a services that ensures your users’ queries are answered from locations for optimal network conditions**.
* **Route 53 accounts are limited to 500 hosted zones as well as 10000 resource record sets for each hosted zone. Users may request a limit increase**.
* Users may import a zone by pasting the zone file directly into the console. Route 53 will then create records in the hosted zone.
* Users can create more than one hosted zone per domain name; this lets users verify settings before replicating those setting to the actual hosted zone to be used.
* **Website hosting is not available in Route 53; it is exclusively a DNS service. Amazon S3 does support static website hosting, and dynamic website hosting is supported by Amazon EC2.**
* Route 53 supports these record types: A, AAAA, CNAME, MX, NS, PTR, SOA, SPF, SRV, and TXT.
* Route 53 allows alias records to map record sets in a hosted zone to other Amazon Web Services that are configured as websites.
* Wildcard entries are supported; these make it easier to configure settings for a domain. A wildcard is a record which matches requests for any domain based on set configuration.
* Time to live (TTL) is the time in which it takes a resolver to cache a response. **No TTL defaults are set in Route 53. Users must specify a TLL for all records**.
* Users can use alias in order to map a sub-domain to other Amazon Web Services that are configured as websites.
* Any changes made to a record is transactional. This ensures consistent, independent, and reliable changes. Also, changes will complete entirely on an individual server or not at all to ensure queries are answered consistently. Change status can be tracked with the **ChangeResourceRecordSets API**.
* **Multiple IP addresses can be associated with an individual record in order to balance the geographical load of web servers.**
* Changes should be complete within 60 seconds given normal conditions.
* **CloudTrail** logs a history of changes and calls for Route 53.
* Users should not use CloudTrail to roll back changes made to hosted zones. The reconstruction may be incomplete if using this method. CloudTrail is for security analysis, change tracking, and compliance auditing.
* DNSSE is not supported by Route 53.
* **Forward (AAAA) as well as reverse (PTR) IPv6 records are supported by Route 53; however, the service is not currently available over IPv6.**
* Using an alias record, users can map a zone apex to an ELB. Due to scaling up or down or software updates, IP addresses with ELB can change any time. Therefore, Route 53 would respond to requests for an alias record with multiple IP addresses. These are free and are listed on the Route 53 usage report.

DNS Routing Policies

* **Weight Round Robin (WRR) allows users to assign weights to record sets that specify the frequency of different responses to be served**. This allows for testing of server responses when software changes occur.
* **Latency Based Routing (LBR) improves application performance for global audience**. LBR allows users to run application in multiple regions with multiple edge locations to ensure the lowest latency.
* LBR can be set up via the AWS management console or an API command. You create a set of records to including ELB names or IP addresses and note it as LBR-enabled and Route 53 handles everything else.
* The LBR feature is priced just as other features; users pay for the zones and queries used.
* Geo DNS allows u sers to balance requests to certain endpoints based on the location from which is came. Users can customize content based on location which ensures consistent routing. **Geo DNS works on three granularity levels: state, country, and continent.**It also keeps records when a user’s location doesn’t connect with any Geo DNS records that are set. Geo DNS can be combined with other services to provide the most consistent performance.

Geo DNS can be set up via the AWS management console or an API. Users create the record and set it with the desired values for the record. Then, mark it as Geo DNS-enabled and choose the region it will apply to.

* Setting a global record ensures a response to all queries from all locations. This helps in the case that Route 53 receive a query from an unrecognizable IP address or a location not include in your Geo DNS records.
* You can create Geo DNS record which overlap regions. Route 53 will respond with the most specific record for the location.
* Geo DNS is priced like other AWS services. Users pay for what they use.
* **Geo DNS determines routing based on request location whereas LBR focuses on latency measurements. LBR is best for minimizing latency.**

**DNS Traffic Flow**

* Route 53 Traffic Flow allows users to improve application performance and availability. By running many endpoints in different regions, Traffic Flows connects the best endpoint to users. Developers can set policies based on what constraints they most care about. Templates can be customized or policies built from scratch via the AWS management console.
* A traffic policy defines the rules that will route requests to endpoints. These policies can be created via the Route 53 console, API, CLI, or SDKs. Once created, traffic policies can be attached to DNS names with a policy record.
* A traffic policy can be reused to manage multiple DNS names. Additional policy records can be created using the existing policy; this requires an additional fee. Users can also create multiple policy records using a standard CNAME record.
* **You cannot not create an alias record that points to a DNS name which is being managed by a traffic policy**.
* There are only charges for policy records; no charges are billed for creating the policy.
* Users are charged per policy record. Billing is done each month and allows for partial month prorating.

**Private DNS**

* Route 53 offers Private DNS which allows users to have authoritative DNS in their VPCs and not expose their DNS records.
* Users create private zones; Route 53 will return queries from the VPC if it’s associated with the private zone.
* Setting up a Private DNS is done in the same way as creating a hosted zone; simply mark the zone as private and associate it with your VPC.
* **Internal DNS names can be resolved within a VPC even without an internet connection. But to configure the Private DNS zone, an internet connection is required since you’ll need access to an API endpoint outside of the VPC.**
* **Private DNS can only be used within a VPC**.
* **Multiple VPCs can be associated with a single zone**.
* Private DNS works across regions. VPCs in every region must have connectivity with other VPCs in other zones.
* Users can set up DNS failover for Private DNS. Failover checks record sets in your private DNS zone for health. These checks can be created against public IP addresses. If you only have private IP address endpoints, health checks cannot check public endpoints. Health checks designed around metrics can be created within CloudWatch.
* Private DNS can be used to block domains and specific DNS names

Health Checks & DNS Failover

* DNS Failover is a service consisting of failover and health checks. Health checks are a set as automated requests to ensure an application is functional, available, and reachable. DNS Failover returns answers of healthy and reachable resources.
* DNS Failover can be set up via the Route 53 console.
* DNS Failover supports ELB endpoints. This can be enable by creating an alias record that points to the ELB, with the evaluate target health to true. Health checks for your ELB are created and managed automatically by Route 53.
* DNS Failover can be utilized to maintain a user’s backup site. In the event your primary site is unreachable, this site will be the fail over site.
* With the exception of SOA and NS records, all record types can be associated with health checks.
* Even if the IP address is unknown, users can configure failover for ELBs or S3 buckets using the Route 53 Console. Route 53 creates and manages these checks automatically when creating an alias record which points to the ELB or S3 bucket and Evaluate Target Health is enabled. Other endpoints can be specified by DNS name.
* Endpoints outside of AWS can be checked using DNS failover as well.
* **If a failover happens, Route 53 will not consider load or traffic capacity at other endpoints. Users must make sure there is available capacity at the other endpoints or must be able to scale up**.
* **The threshold default is three health checks**. Even after an endpoint is considered failed, Route 53 will perform checks and will start routing traffic back after three successful checks. Threshold defaults can be set between 1 and 10.
* The default interval setting is **30 seconds**. The default can be set as quickly as 10 seconds. Increasing the interval creates more requests to an endpoint so this must be considered if limited capacity is an issue.
* Users can set the number and set of locations using the Route 53 console or an API.
* HTTP redirects are not checked.
* A TTL of 60 seconds or less is suggested for using DNS Failover. This decreases the amount of time for traffic to stop being routed at the failed point. You must use alias records with a TTL of 60 seconds to configure failover for ELB and S3 endpoints.
* If all endpoints fail, Route 53 will react like all checks are passing. At least one healthy endpoint is needed for your applications to failover.
* **LBR is not required for using DNS Failover.**
* **Health checks can be completed for sites with HTTPS, HTTP or TCP.**
* **Health checks do not authenticate an endpoint’s SSL certificate.**
* **HTTPS health checks do support SNI**.
* Health checks can be created to look for the presence of a string in a response by using Enable String Matching.
* Health check status and failure details can be viewed via the Route 53 console or an API. Results are also stored in CloudWatch organized by various metrics. A graph of metrics can be viewed in CloudWatch and alarms can be set to send notifications if health check status changes.
* Optional latency measures are available which provide data on the length of a request response. This additional metric shows how long it took the health checkers to connect and begin getting data.
* Notifications and automated actions can be set to trigger when values change beyond a certain threshold. This can be done in Route 53 or CloudWatch.
* Confirmation emails for an alarm’s SNS topic can be resent by using the alarm name in the Route 53 console. There is a box named send notification to. In the SNS console, there’s a list of topics you can create alarms on with the create subscription box.
* Since using DNS Failover with ELB is done with alias records there are no CloudWatch metrics created. Metrics can be found for these endpoints using ELB or by creating a health check against the CNAME.
* When alias records point to S3 buckets, the health checks examine the S3 in each region. The health check doesn’t check the bucket content but only if the S3 service is unavailable in the region of the bucket.
* CloudWatch metrics for checks are free.
* Health checks can be configured on any metric available in CloudWatch.
* If your server is receiving an unrequested health check, you can notify Route 53 of this

**Domain Name Registration**

* New domain names can be registered with the AWS management console or an API.
* Registration can take a few minutes or many hours depending on the TLD selected. Once registered, the domain name will be viewable in your account.
* Most domain names are registered for a year. Some TLDs have longer periods. Automatic renewal will be set at registration.
* The only information you need to register a domain name is the registrant contact information. If your administrative and/or technical contacts will be different, you’ll need that contact information as well.
* Contact information is required by the domain registration governing body, ICANN. Domains registered for individuals have privacy protections free of charge.
* When creating a domain name, a delegation set of four unique name servers is assigned. The delegation set is viewable via the Route 53 console. They are in the zone automatically created at domain registration. A new delegation set is created for each hosted zone created. Users can create reusable delegation sets to apply to multiple zones.
* Users will be charged for hosted zones created for your domain name and DNS queries against hosted zones. To avoid being charged for DNS service, delete the hosted zone. Some TLDS require valid name servers.
* Amazon Registrar is a company accredited by ICANN. AWS resells domain names registered with accredited registrar. The registrar your domain is registered with is listed for your WHOIS record.
* Gandi is a registrar which Amazon is a reseller of. If Gandi is the registrar of your domain name, they are required to verify your contact information. This verification must be done within 15 days to prevent suspension of the domain name.
* Route 53 does not transfer .com or .net registrations from Gandi currently.
* Whois is a database of domain names which keeps contact information as well as names servers. Anyone has access to the Whois database with the WHOIS command. Contact information is publicly available according to ICANN requirements.
* Transfer of domain names can be done in the domains section of your account. In order to transfer, the domain name must be unlocked, privacy protection must be disabled, and a valid authorization code must be obtained.
* To prevent disruption of web services during transfer of a domain name, a list of record data must be obtained. When that is ready, users can use the management console or a web service interface to create a zone for storing the records. Then, follow the transfer process. To complete the procedures, contact your registrar and follow the transfer process. Once propagated, queries will begin to be answered by Route 53 servers.
* Transfer status can be checked in the alerts section of the console.
* If transfer is unsuccessful, contact your registrar to determine the failure reason. Once resolved, try the transfer again.
* To transfer a domain name out of Route 53, you must complete a transfer request with the new registrar.
* **Customers are limited to 50 domains. To increase this limit, users can submit a request.**
* **DNSSEC is not supported by Route 53**. But the DNS registration can support configuration of DNSSEC keys from other DNS services.