**How Health Checks Work in Complex Amazon Route 53 Configurations**

Checking the health of resources in complex configurations works much the same way as in simple configurations. However, in complex configurations, you use a combination of alias records (such as weighted alias and failover alias) and non-alias records to build a decision tree that gives you greater control over how Route 53 responds to requests.

For example, you might use latency alias records to select a region close to a user and use weighted records for two or more resources within each region to protect against the failure of a single endpoint or an Availability Zone. The following diagram shows this configuration.


				DNS configuration that includes latency alias records and weighted alias records.
			

Here's how Amazon EC2 and Route 53 are configured. Let's start at the bottom of the tree because that's the order that you'll create records in:

* You have two EC2 instances in each of two regions, us-east-1 and ap-southeast-2. You want Route 53 to route traffic to your EC2 instances based on whether they're healthy, so you create a health check for each instance. You configure each health check to send health-checking requests to the corresponding instance at the Elastic IP address for the instance.

Route 53 is a global service, so you don't specify the region that you want to create health checks in.

* You want to route traffic to the two instances in each region based on the instance type, so you create a weighted record for each instance and give each record a weight. (You can change the weight later to route more or less traffic to an instance.) You also associate the applicable health check with each instance.

When you create the records, you use names such as us-east-1-www.example.com. and ap-southeast-2-www.example.com. You'll wait until you get to the top of the tree to give records the names that your users will use to access your website or web application, such as example.com.

* You want to route traffic to the region that provides the lowest latency for your users, so you choose the latency [routing policy](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html) for the records at the top of the tree.

You want to route traffic to the *records* in each region, not directly to the *resources* in each region (the weighted records already do that). As a result, you create latency [alias records](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html).

When you create the alias records, you give them the name that you want your users to use to access your website or web application, such as example.com. The alias records route traffic for example.com to the us-east-1-www.example.com and ap-southeast-2-www.example.com records.

For both latency alias records, you set the value of **Evaluate Target Health** to **Yes**. This causes Route 53 to determine whether there are any healthy resources in a region before trying to route traffic there. If not, Route 53 chooses a healthy resource in the other region.


				DNS configuration that includes latency alias records and weighted alias records.
			

The preceding diagram illustrates the following sequence of events:

1. Route 53 receives a query for example.com. Based on the latency for the user making the request, Route 53 selects the latency alias record for the us-east-1 region.
2. Route 53 selects a weighted record based on weight. **Evaluate Target Health** is **Yes** for the latency alias record, so Route 53 checks the health of the selected weighted record.
3. The health check failed, so Route 53 chooses another weighted record based on weight and checks its health. That record also is unhealthy.
4. Route 53 backs out of that branch of the tree, looks for the latency alias record with the next-best latency, and chooses the record for ap-southeast-2.
5. Route 53 again selects a record based on weight, and then checks the health of the selected resource. The resource is healthy, so Route 53 returns the applicable value in response to the query.

**Topics**

* [What Happens When You Associate a Health Check with an Alias Record?](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-complex-configs.html#dns-failover-complex-configs-hc-alias)
* [What Happens When You Omit Health Checks?](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-complex-configs.html#dns-failover-complex-configs-hc-omitting)
* [What Happens When You Set Evaluate Target Health to No?](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-complex-configs.html#dns-failover-complex-configs-eth-no)

**What Happens When You Associate a Health Check with an Alias Record?**

You can associate a health check with an alias record instead of or in addition to setting the value of **Evaluate Target Health** to **Yes**. However, it's generally more useful if Route 53 responds to queries based on the health of the underlying resources—the HTTP servers, database servers, and other resources that your alias records refer to. For example, suppose the following configuration:

* You assign a health check to a latency alias record for which the alias target is a group of weighted records.
* You set the value of **Evaluate Target Health** to **Yes** for the latency alias record.

In this configuration, both of the following must be true before Route 53 will return the applicable value for a weighted record:

* The health check associated with the latency alias record must pass.
* At least one weighted record must be considered healthy, either because it's associated with a health check that passes or because it's not associated with a health check. In the latter case, Route 53 always considers the weighted record healthy.

In the following illustration, the health check for the latency alias record on the top left failed. As a result, Route 53 stops responding to queries using any of the weighted records that the latency alias record refers to even if they're all healthy. Route 53 begins to consider these weighted records again only when the health check for the latency alias record is healthy again. (For exceptions, see [How Amazon Route 53 Chooses Records When Health Checking Is Configured](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/health-checks-how-route-53-chooses-records.html).)


					DNS configuration that includes an alias record with both Evaluate Target Health set to Yes and a health check 
						on the alias record.
				

**What Happens When You Omit Health Checks?**

In a complex configuration, it's important to associate health checks with all the non-alias records. In the following example, a health check is missing on one of the weighted records in the us-east-1 region.


					DNS configuration that includes one failed health check and one record that has no health check.
				

Here's what happens when you omit a health check on a non-alias record in this configuration:

1. Route 53 receives a query for example.com. Based on the latency for the user making the request, Route 53 selects the latency alias record for the us-east-1 region.
2. Route 53 looks up the alias target for the latency alias record, and checks the status of the corresponding health checks. The health check for one weighted record failed, so that record is omitted from consideration.
3. The other weighted record in the alias target for the us-east-1 region has no health check. The corresponding resource might or might not be healthy, but without a health check, Route 53 has no way to know. Route 53 assumes that the resource is healthy and returns the applicable value in response to the query.

**What Happens When You Set Evaluate Target Health to No?**

In general, you should set **Evaluate Target Health** to **Yes** for all the alias records in a tree. If you set **Evaluate Target Health** to **No**, Route 53 continues to route traffic to the records that an alias record refers to even if health checks for those records are failing.

In the following example, all the weighted records have associated health checks, but **Evaluate Target Health** is set to **No** for the latency alias record for the us-east-1 region:


					DNS configuration that includes an alias record with Evaluate Target Health set to No.
				

Here's what happens when you set **Evaluate Target Health** to **No** for an alias record in this configuration:

1. Route 53 receives a query for example.com. Based on the latency for the user making the request, Route 53 selects the latency alias record for the us-east-1 region.
2. Route 53 determines what the alias target is for the latency alias record, and checks the corresponding health checks. They're both failing.
3. Because the value of **Evaluate Target Health** is **No** for the latency alias record for the us-east-1 region, Route 53 must choose one record in this branch instead of backing out of the branch and looking for a healthy record in the ap-southeast-2 region.