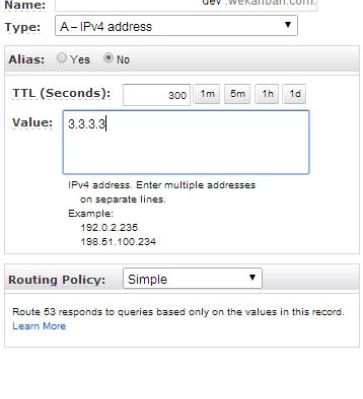
**Amazon Web Services Route 53 Part 2 – Routing Policies**

Today I will continue to explain Amazon Route 53 service and routing policies. Routing policies are used with record sets and enables us to configure failover scenarios like active-active and active-passive. Currently Amazon Route 53 supports four types of routing policies; simple,weighted,latency and failover. I will explain them step by step by giving examples. Let’s start…

**Simple routing policy** is used only for a specific record like A,MX,NS etc. In this way, our DNS query will resolve to only one ip/host like:

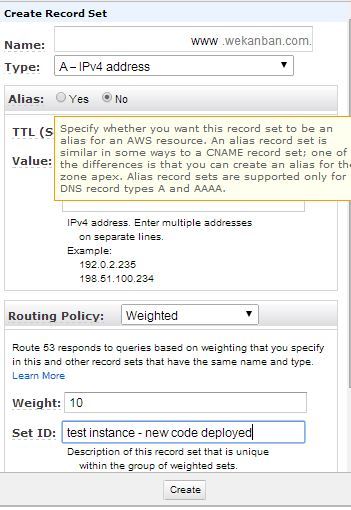
[](https://i1.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/simple.jpg)

As you see I created an A record as “dev.wekanban.com” and used simple record set that resolves as 3.3.3.3

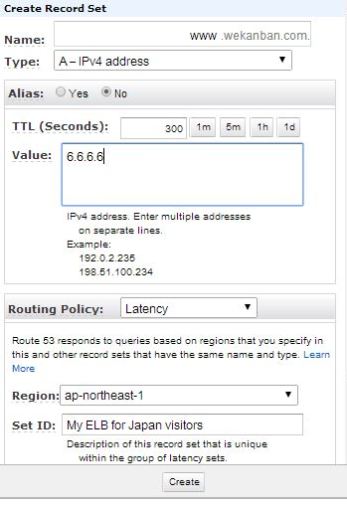
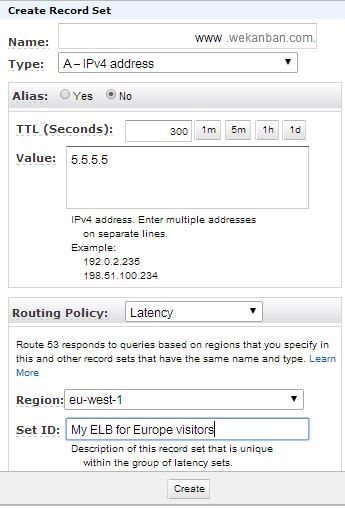
**Weighted routing policy** enables us to reply to a single DNS queries with different ip addresses. Imagine that we have two EC2 instances and have different instance types, m3.medium and m3.large. Since one of the instances have more resources, we can forward more traffic to it by using weighted routing policy. And also we may want to test our new deployed code on real traffic without effecting our whole clients/visitors. Again we can use weighted routing policy to redirect a small portion of traffic to our new deployed instances. There is a formula to calculate the probability of the record being selected as below:

**Weight of the record set/Sum of the weights of the record sets**

For example, if I have two instances and give 10 weight to one instance(test- new code deployed) and 90 to the other instance(production- with old code) ==> 10/10+90=0,1 which means %10 of the traffic will be redirected to test instance. As seen below I create two weighted records. In weighted policy “Set ID” is used to describe the record. Remember that, If you give a “0” as weight, routing to this record is disabled if you didn’t associate it with health check. I will explain health checks in another post.

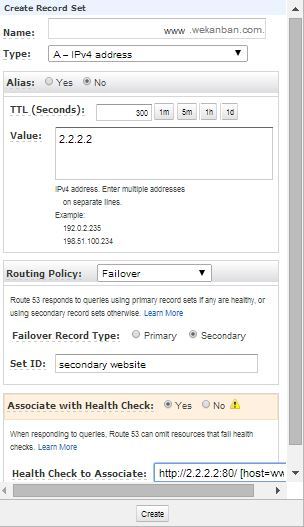
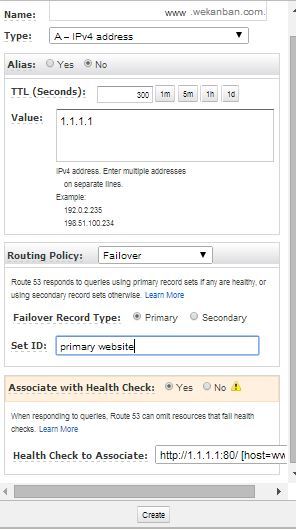
[](https://i0.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/weighted_10.jpg) [](https://i2.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/weighted_90.jpg)

Now imagine that your website analysis shows that most visitors source countries are Japan and UK. In that case you may want to improve your website speed by lowering your latency between your website and visitors. In this example we can use **latency routing policy** and create two records as below. In that way visitors in Japan DNS queries will resolve as 6.6.6.6 and for the UK 5.5.5.5

[](https://i1.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/latency_2.jpg) [](https://i2.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/latency_1.jpg)

In this example I didn’t use alias record but it is better and free to use alias for Elastic load balancers.

And finally we can create a **failover routing policy**. We can use this policy as a primary and a secondary site solution. For example you may have two instances and in case of one of them fails, DNS queries will resolve as secondary records. I have to say that before configuring failover policy, we have to create health checks for our records. Then we can associate with health checks and create our records.

[](https://i0.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/secondary.jpg) [](https://i0.wp.com/www.awsomeblog.com/wp-content/uploads/2014/05/primary.jpg)

If you have any question or comment please feel free to write and don’t forget to share please.