ASG-POLICIES

So now let's talk about

auto scaling group scaling policies.

And we have two different kinds,

we have the dynamic scaling policies first.

And so within the dynamics scaling policies,

we have three kinds, we have the target tracking scaling,

which is pretty easy.

It's the most simple and easy to set up.

And the idea is that you wanna say, for example,

I want to track the average CPU utilization

of my autoscaling groups across all my EC2 instances

to stay at around 40%.

This is when you want to have a default baseline

and wanna make sure that you're always available.

Simple and step scaling is more involved.

So you set up your own CloudWatch alarms

and when they're triggered so reasonable,

when the CPU goes over 70% for your ASG as a whole,

then add two units of capacity.

And then you would set up a second rule saying,

hey, in case the CPU utilization goes to less than 30%

as a whole within my ASG, then remove one unit.

But you would have to set up your CloudWatch alarms

as well as the steps which is,

how many units you wanna add at a time

and how many units you want to remove at a time.

And finally scheduled actions,

which is to anticipate scaling

based on the known users patterns.

For example, you're saying that,

hey, I know that there's going to be a big event

at 5:00 PM on Fridays

because when people are gonna be done with work

and they're going to use my application,

and therefore you want you to increase the min capacity

automatically of your ASG to 10 at 5:00 PM

every single Friday.

This is a scheduled action

where you know scaling in advance.

And there's a new kind of scaling,

which is called predictive scaling.

So with predictive scaling,

you continually have a forecast being made

by the autoscaling service in AWS.

And it will look at the load

and we'll schedule scaling ahead.

So what will happen is that

the historical load is going to be analyzed over time

and then forecast is going to be created.

And then based on that forecast,

they will be scaling actions being scheduled ahead of time,

which is a quite a cool way of doing scaling as well.

And I think this is the future

because this is machine learning powered,

and this really is a hands-off approach

to automatic scaling for ASG.

So some good metrics with scale on is a big question.

So it depends really on what your application is doing

and how it's working,

but usually here are a few.

So number one is CPU utilization

because every time your instance receive a request,

usually they will do some sort of computation

and so it will use some CPU.

And so if you look at the average CPU utilization

across all your instances and it goes higher,

that means that your instances are being more utilized

and so it would be a good metric to scale on.

Another metric to scale on,

it's more like application specific,

but it is a request counts per targets,

which is based on your testing.

You know that your EC2 instances

operate at an optimal request

of 1000 per request per target at a time

and so maybe this is the target you want to have

for your scaling.

So here's an example.

You have an auto scaling group with three EC2 instances,

and your lb is currently spreading the instance request

across all of them.

So right now the value of the request

counts per target metric is three

because each EC2 instance on average

has three requests outstanding.

Next, if your application is network bound,

so for example, there's a lot of uploads and downloads

and you know that network is going to be a bottleneck

for your EC2 instances,

then you may want to scale on the average network in or out

to make sure that if you reach some certain threshold,

then you're going to scale based on that.

Or any custom metrics that you push to CloudWatch,

you can set up your own metrics

that are going to be application specific

and based on that, you can set up your scaling policies.

Now, what else would you need to know about scaling policies

is what's called the scaling cooldown.

So the idea is that after there is a scaling activity,

so whenever you add, or you remove instances,

you are entering the cool-down period,

which is by default five minutes or 300 seconds.

And during that cooldown period,

the ASG will not launch or terminate additional instances.

And the reason behind this reasoning is that

you allow four metrics to stabilize, okay,

for your new instance to enter into effect

and to see what the new metric will become.

So the idea is that

when there is a scaling action that occurs,

the question is, is there a default cooldown in effects?

If yes, then ignore the action.

If no, then proceeded with the scaling action

which is to launch or terminate instances.

And so in advice to you, is to use a ready-to-use AMI

to reduce the configuration time for your EC2 instances

in order for them to request

or to be serving the requests faster.

So if you don't spend time configuring your EC2 instance,

then they can be in effect right away.

And then because there can be active way faster

then the cooldown period can be decreased,

and you can have a more dynamic scaling up and down

of your ASG.

And of course you need to make sure

to enable something like detailed monitoring for ASG

to get access to lower level two metrics every one minute,

and to make sure that you have these metrics

being updated fast enough.

So that's it for this lecture.

I hope you liked it,

and I will see you in the next lecture.