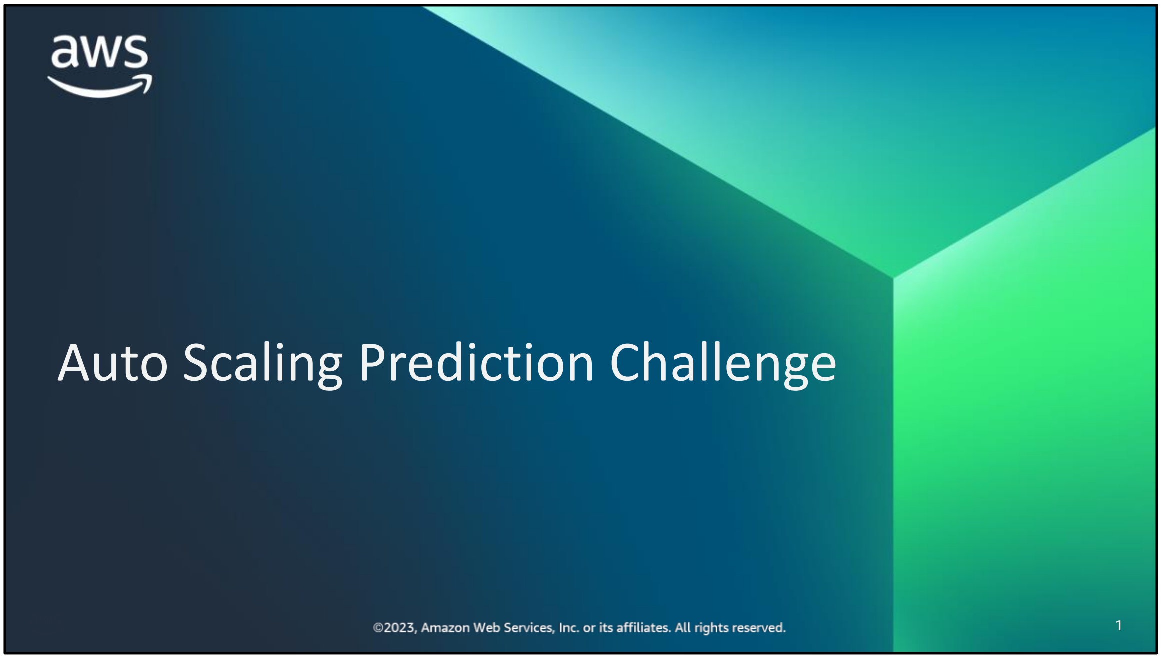
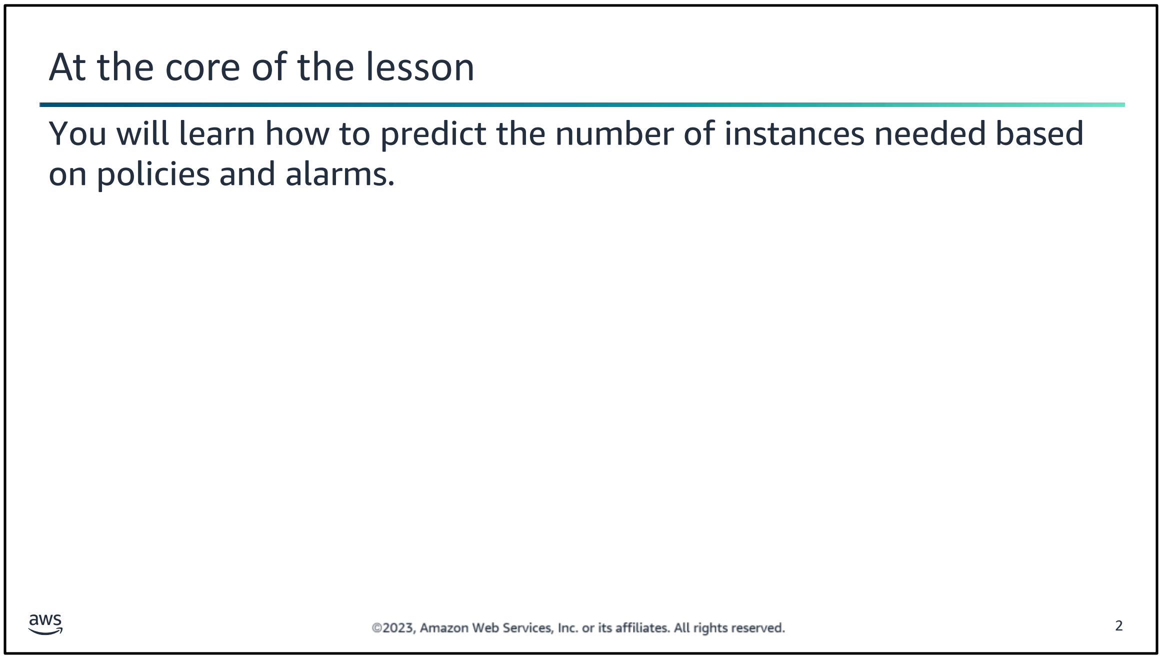
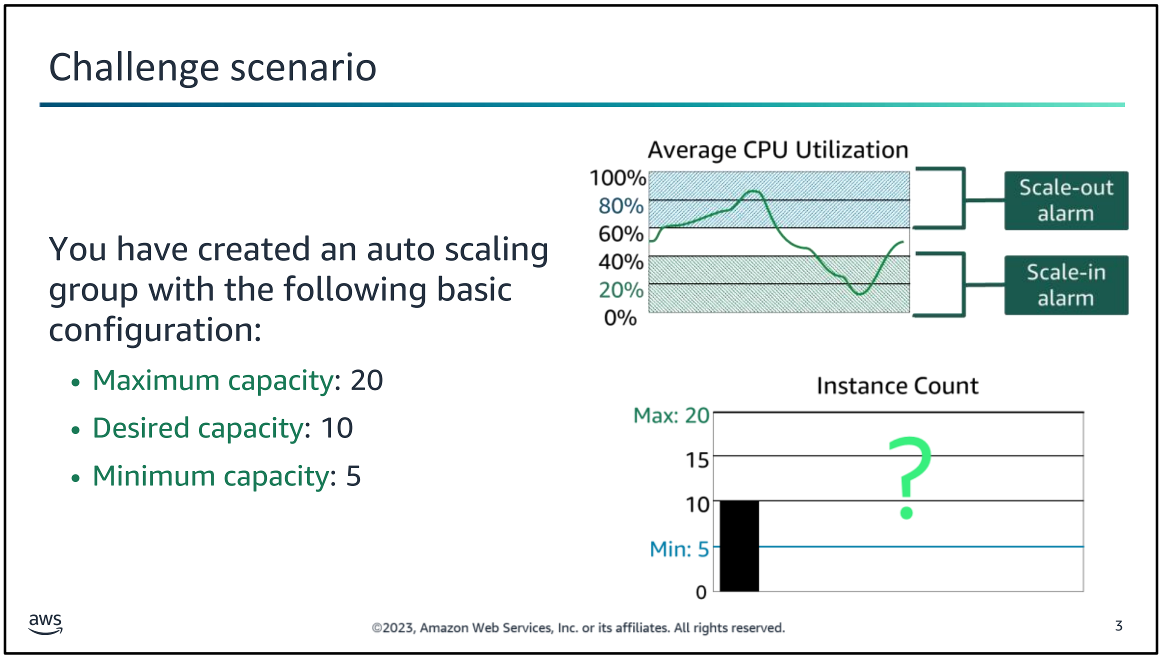


Auto Scaling Prediction Challenge



At the core of the lesson

You will learn how to predict the number of instances needed based on policies and alarms.



Challenge scenario

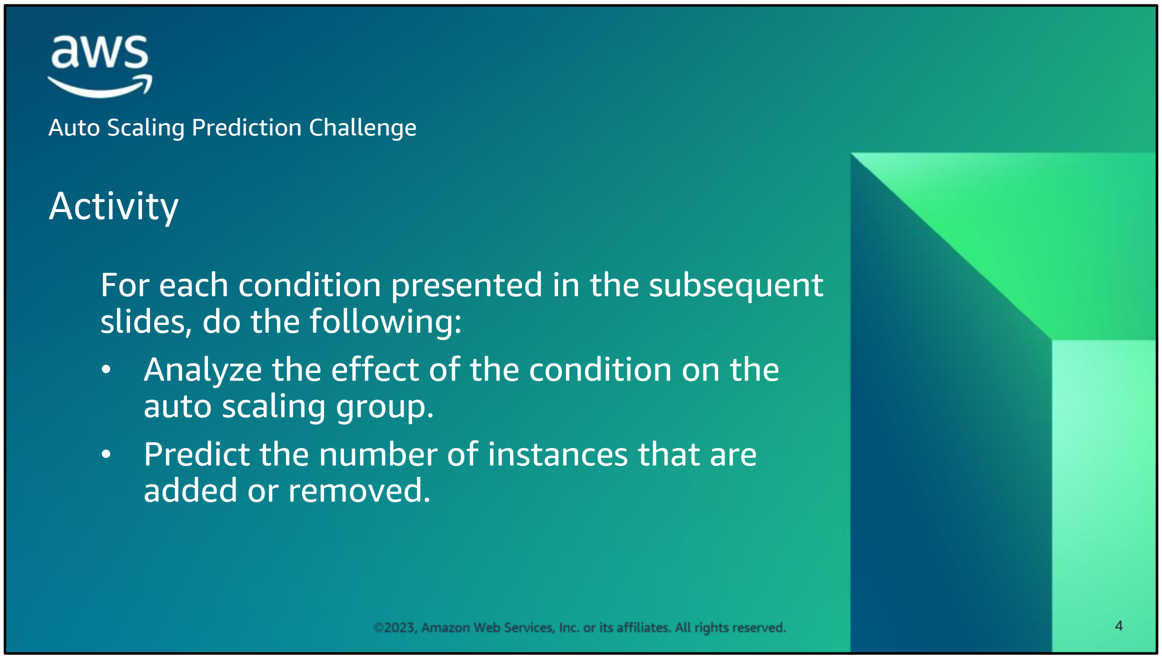
You have created an auto scaling group with the following basic configuration:

- Maximum capacity: 20
- Desired capacity: 10
- Minimum capacity: 5

aws

You have an auto scaling group with a maximum capacity of 20, desired capacity of 10, and minimum capacity of 5. You have decided to create an Amazon EC2 Auto Scaling policy for step scaling with scale-out and scale-in actions based on the average CPU utilization. However, you are unsure how the Amazon Elastic Compute Cloud (Amazon EC2) instances will scale out or in based on the current policies that you defined to control the behaviors.

To test the instance behaviors, you want to be able to predict the number of instances at any time based on the policies and respective alarms. How will the number of instances in the auto scaling group vary when the conditions in the next slides occur?



Auto Scaling Prediction Challenge

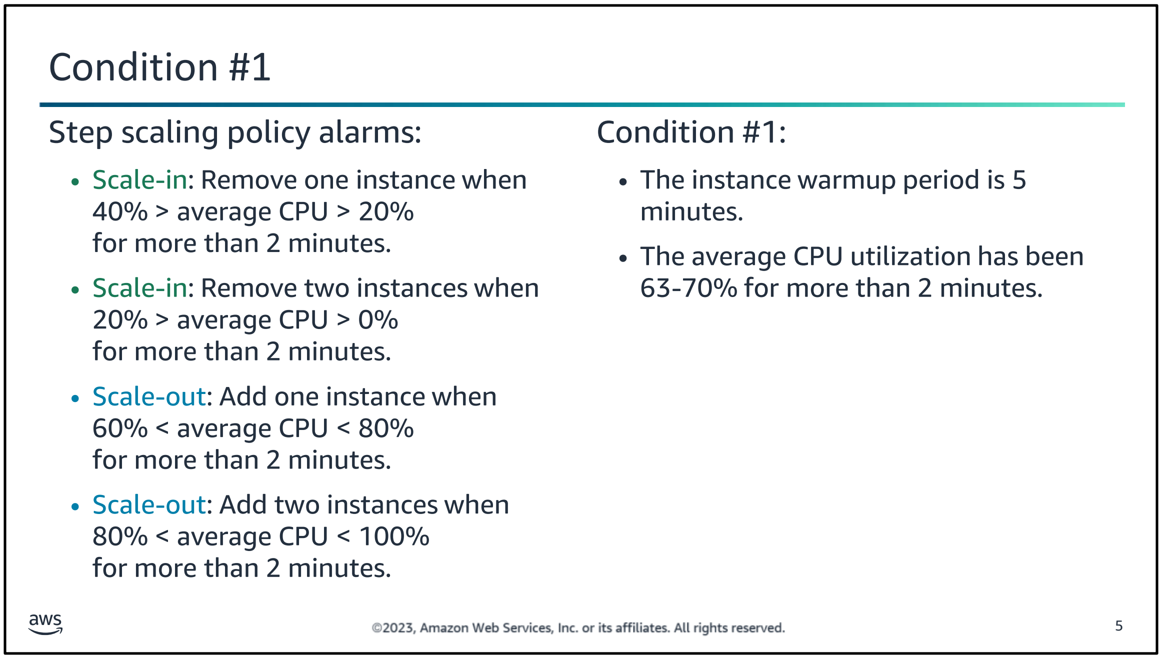
Activity

For each condition presented in the subsequent slides, do the following:

- Analyze the effect of the condition on the auto scaling group.
- Predict the number of instances that are added or removed.

Record your prediction for each condition by identifying the resulting change in the number of instances in the auto scaling group. The following are example predictions:

- Condition #1 results in no change.
- Condition #2 results in adding two instances.



Condition #1

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

Condition #1:

- The instance warmup period is 5 minutes.
- The average CPU utilization has been 63-70% for more than 2 minutes.

The left side of the slide shows the policy alarms that you defined to control the scale-in and scale-out behavior that you want to implement.

You configure Amazon CloudWatch alarms so that when the average CPU load in the auto scaling group is 20–40 percent for more than 2 minutes, the size of the group will be reduced by one instance.

If the average CPU load is less than 20 percent for more than 2 minutes, then the group will be reduced by two instances.

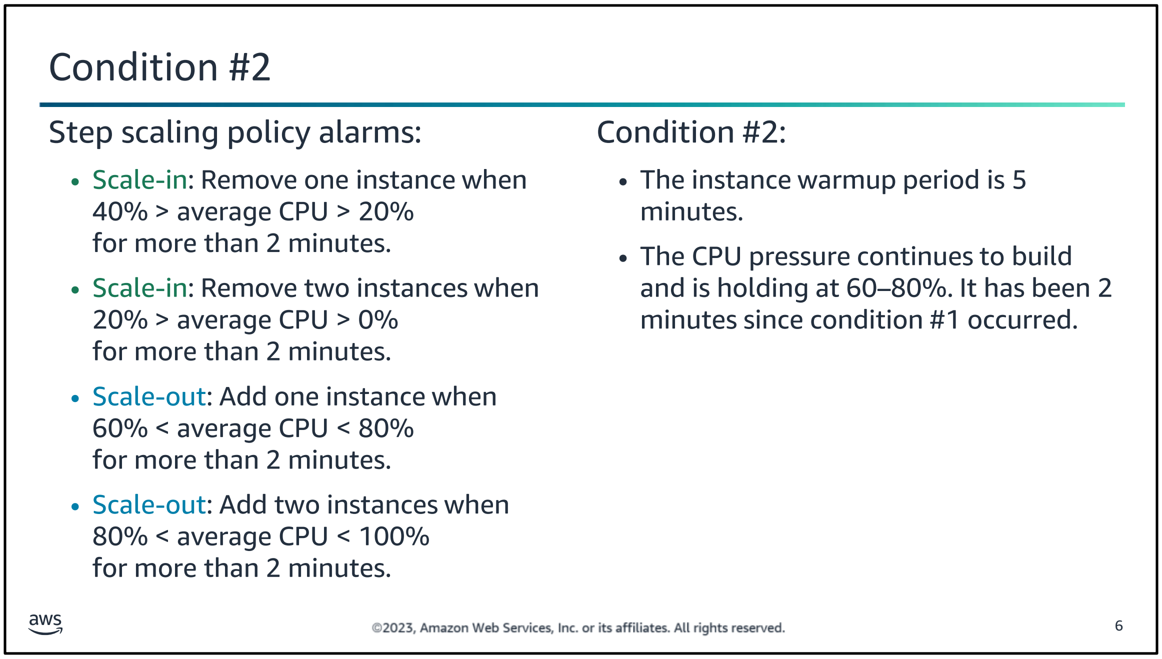
On the other hand, if CPU utilization is 60–80 percent for more than 2 minutes, an instance will be added. If it exceeds 80 percent for more than 2 minutes, two instances will be added.

Now consider what will happen under the following circumstances:

- Condition #1: The average CPU utilization has been 63–70 percent for more than 2 minutes. The instance warmup period is set to 5 minutes.

Predict the number of instances that are added or removed.

Auto scaling result: The policy adds an instance to the auto scaling group, but the new instance won't be counted as part of the instance count until its warmup period (that is, 5 minutes) is over.



Condition #2

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

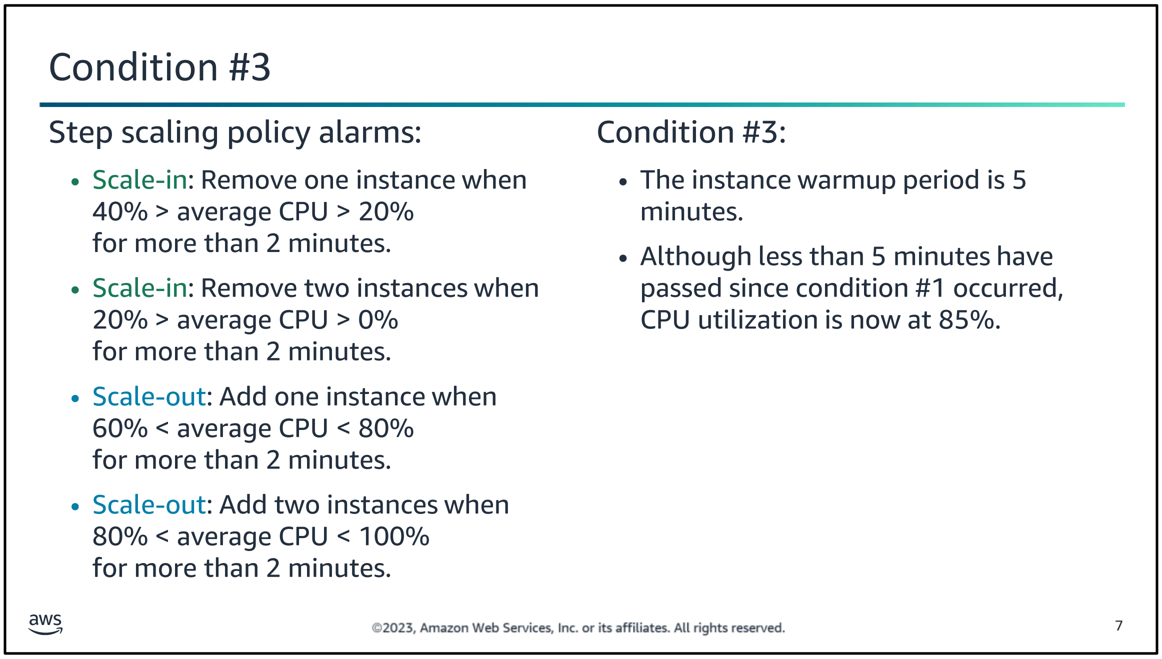
Condition #2:

- The instance warmup period is 5 minutes.
- The CPU pressure continues to build and is holding at 60–80%. It has been 2 minutes since condition #1 occurred.

Condition #2: The CPU pressure continues to build and is holding in the 60–80 percent range. It has been 2 minutes since condition #1 occurred.

Predict the number of instances that are added or removed. Assume the instance warmup period is set to 5 minutes.

Auto scaling result: An alarm invokes another scale-out by one instance. However, no additional instance is added because the instance warmup period has not elapsed.



Condition #3

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

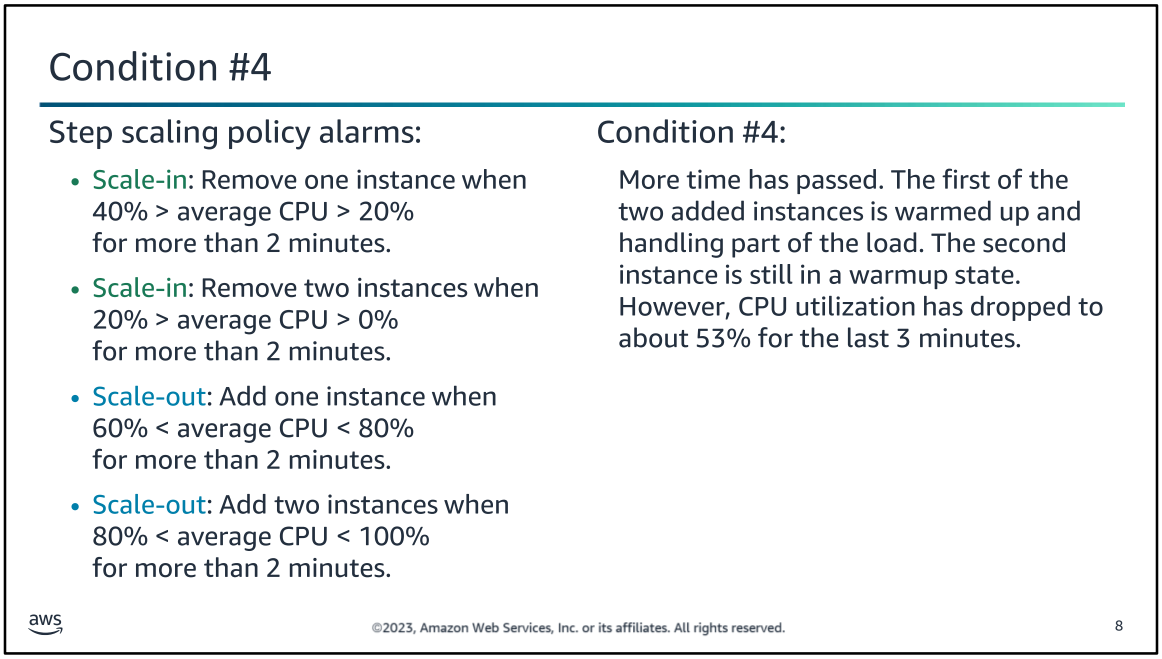
Condition #3:

- The instance warmup period is 5 minutes.
- Although less than 5 minutes have passed since condition #1 occurred, CPU utilization is now at 85%.

Condition #3: It is still less than 5 minutes since condition #1 occurred, but now CPU utilization is at 85 percent.

Predict the number of instances that are added or removed. Assume the instance warmup period is set to 5 minutes.

Auto scaling result: An alarm invokes the auto scaling policy to add two instances. Because one instance is already warming up, only one additional instance is launched.



Condition #4

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

Condition #4:

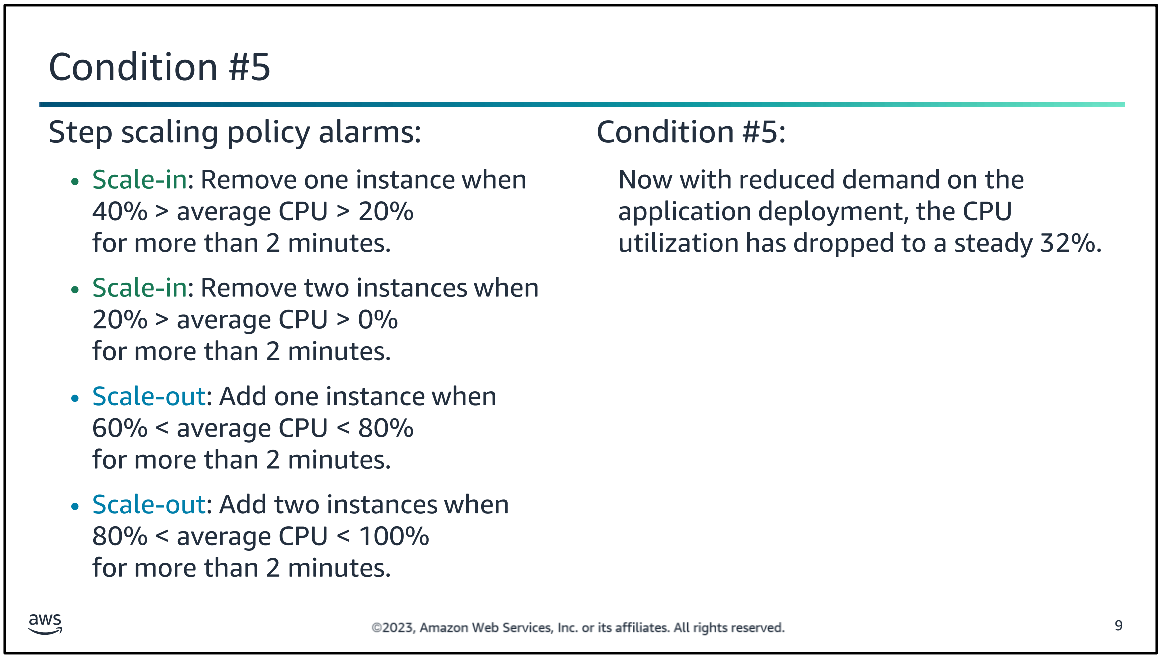
More time has passed. The first of the two added instances is warmed up and handling part of the load. The second instance is still in a warmup state. However, CPU utilization has dropped to about 53% for the last 3 minutes.

Now consider what will happen under the following circumstances:

Condition #4: More time has passed, and the first of the two added instances is now fully warmed up and handling part of the load while the second instance is still in a warmup state. CPU utilization pressure is subsiding. The average utilization has dropped below 60 percent, but it is still above 40 percent.

Predict the number of instances that are added or removed.

Auto scaling result: No alarms are invoked, and no actions are taken.



Condition #5

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

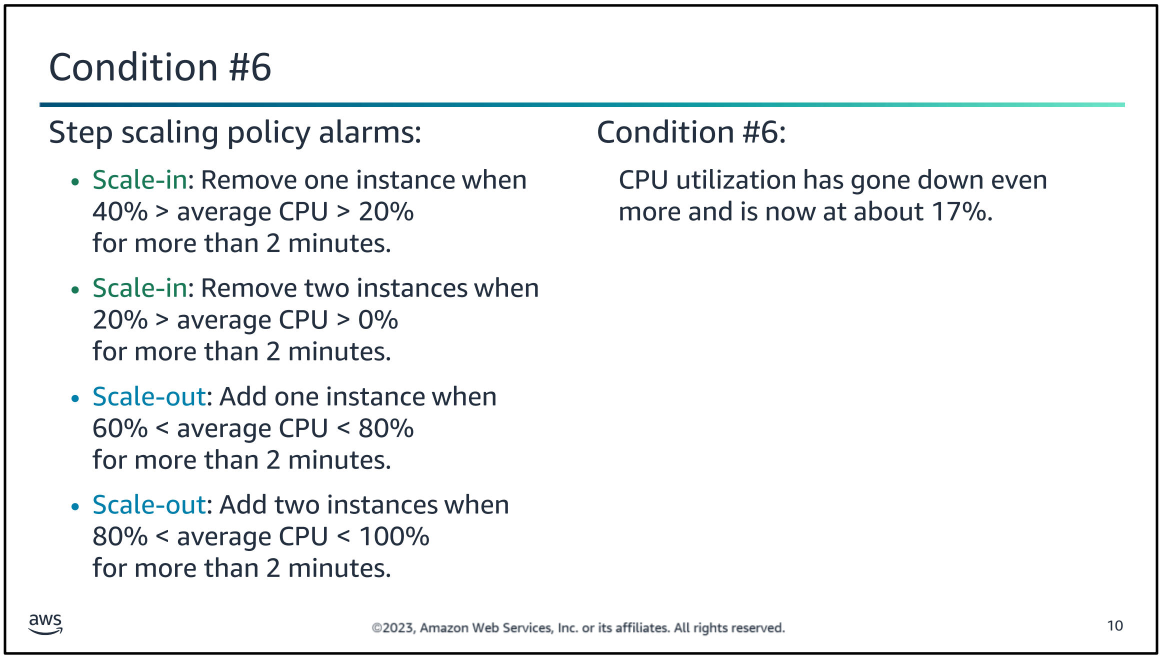
Condition #5:

Now with reduced demand on the application deployment, the CPU utilization has dropped to a steady 32%.

Condition #5: Now with 12 instances in the group and perhaps some reduced demand on the application deployment that this group supports, the CPU utilization has dropped below the 40 percent threshold to a steady 32 percent.

Predict the number of instances that are added or removed.

Auto scaling result: After 2 minutes below the 40 percent threshold, one of the scale-in alarms is invoked, which initiates the policy to remove an instance.



Condition #6

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

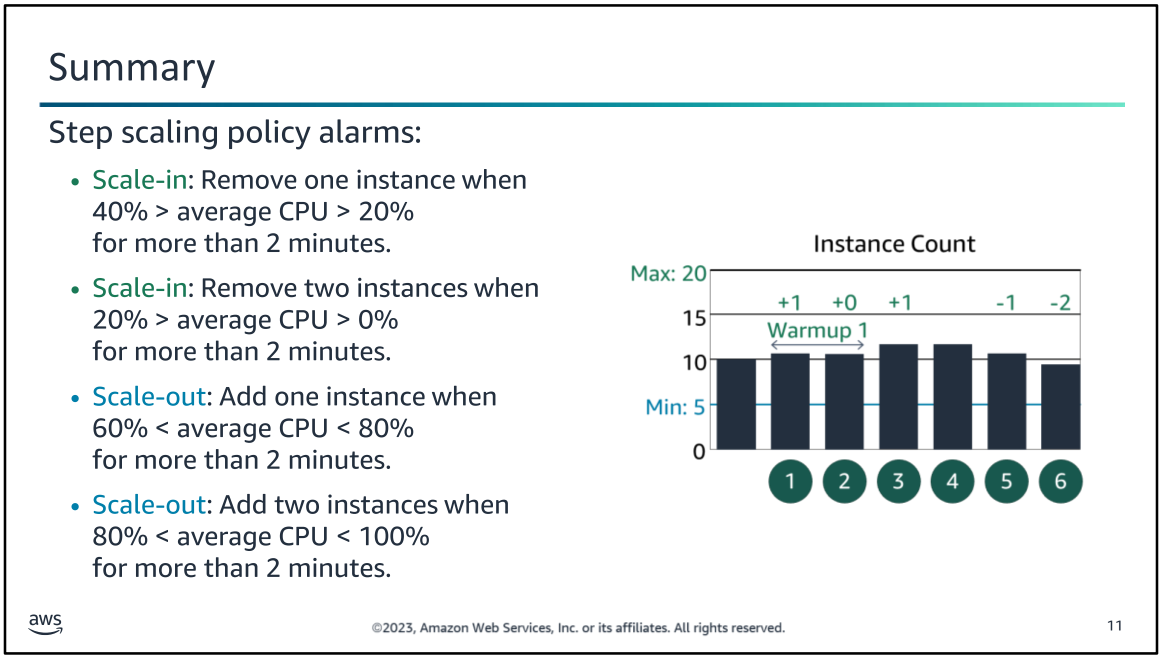
Condition #6:

CPU utilization has gone down even more and is now at about 17%.

Condition #6: CPU utilization then goes down even more and is now at about 17 percent.

Predict the number of instances that are added or removed.

Auto scaling result: Two instances are removed. Now, the group has only nine instances.



Summary

Step scaling policy alarms:

- Scale-in: Remove one instance when 40% > average CPU > 20% for more than 2 minutes.
- Scale-in: Remove two instances when 20% > average CPU > 0% for more than 2 minutes.
- Scale-out: Add one instance when 60% < average CPU < 80% for more than 2 minutes.
- Scale-out: Add two instances when 80% < average CPU < 100% for more than 2 minutes.

Here is the answer key for the automatic scaling scenario.

1. A condition has existed for at least 2 minutes. This condition invoked the alarm, and the alarm initiated the policy that adds an instance to the auto scaling group.

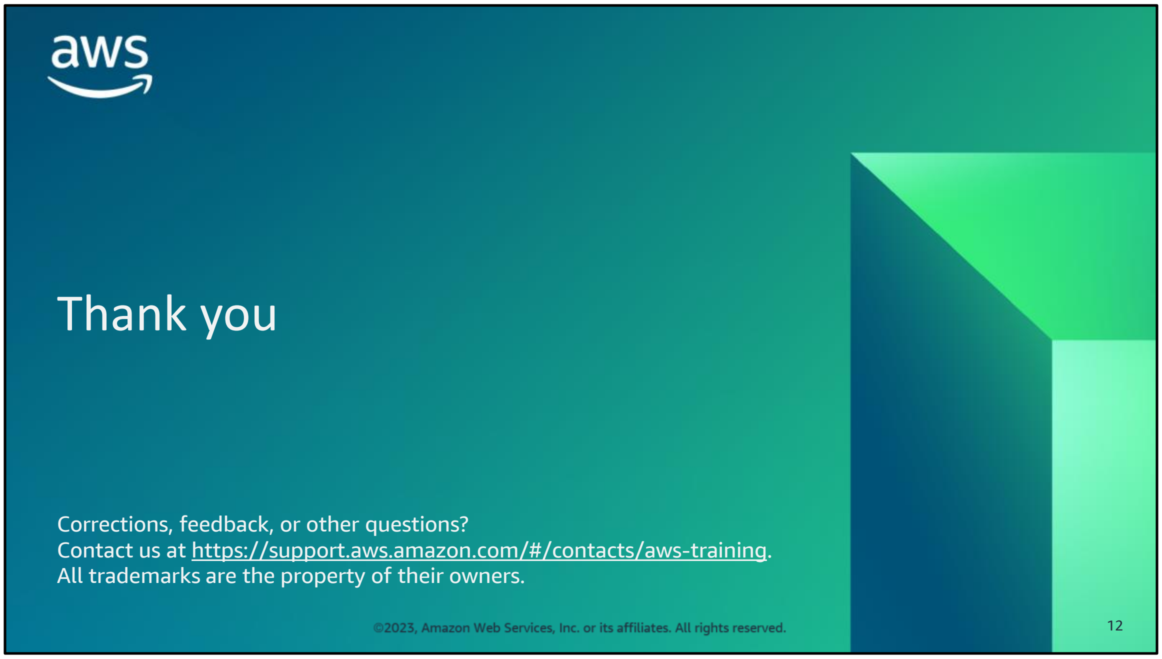
2. The CPU pressure continues to build, but it has still not crossed the 80 percent threshold that would invoke the action to add two instances. However, 2 minutes after the initial alarm is invoked, another alarm is invoked to scale out by one instance because CPU utilization remains at 60–80 percent. Because the instance warmup period had not elapsed when the second alarm was invoked, the second alarm does not result in adding another instance.

3. The CPU utilization crosses the 80 percent threshold, which invokes the alarm for the scale-out policy to add two instances. The third alarm (the first 80 percent alarm) occurred during the warmup period for the instance that was added when the first alarm was invoked. Because one instance is already warming up, only one additional instance is launched.

4. More time has passed, and the first of the two added instances is now fully warmed up and handling part of the load while the second instance is still in a warmup state. CPU utilization pressure is subsiding. The average utilization has dropped below 60 percent, but it is still above 40 percent. At this time, no alarms are invoked.

5. Now with 12 instances in the group, and perhaps some reduced demand on the application deployment that this group supports, the CPU utilization has dropped below the 40 percent threshold. After 2 minutes below the 40 percent threshold, one of the scale-in alarms is invoked, which initiates the policy to remove an instance.

6. CPU utilization then goes down even more, which invokes the alarm that will remove two instances because CPU utilization is now below 20 percent. Now, the group has only nine instances. Money is being saved because unneeded instances are no longer running.



Thank you

Corrections, feedback, or other questions?
Contact us at https://support.aws.amazon.com/#/contacts/aws-training.
All trademarks are the property of their owners.