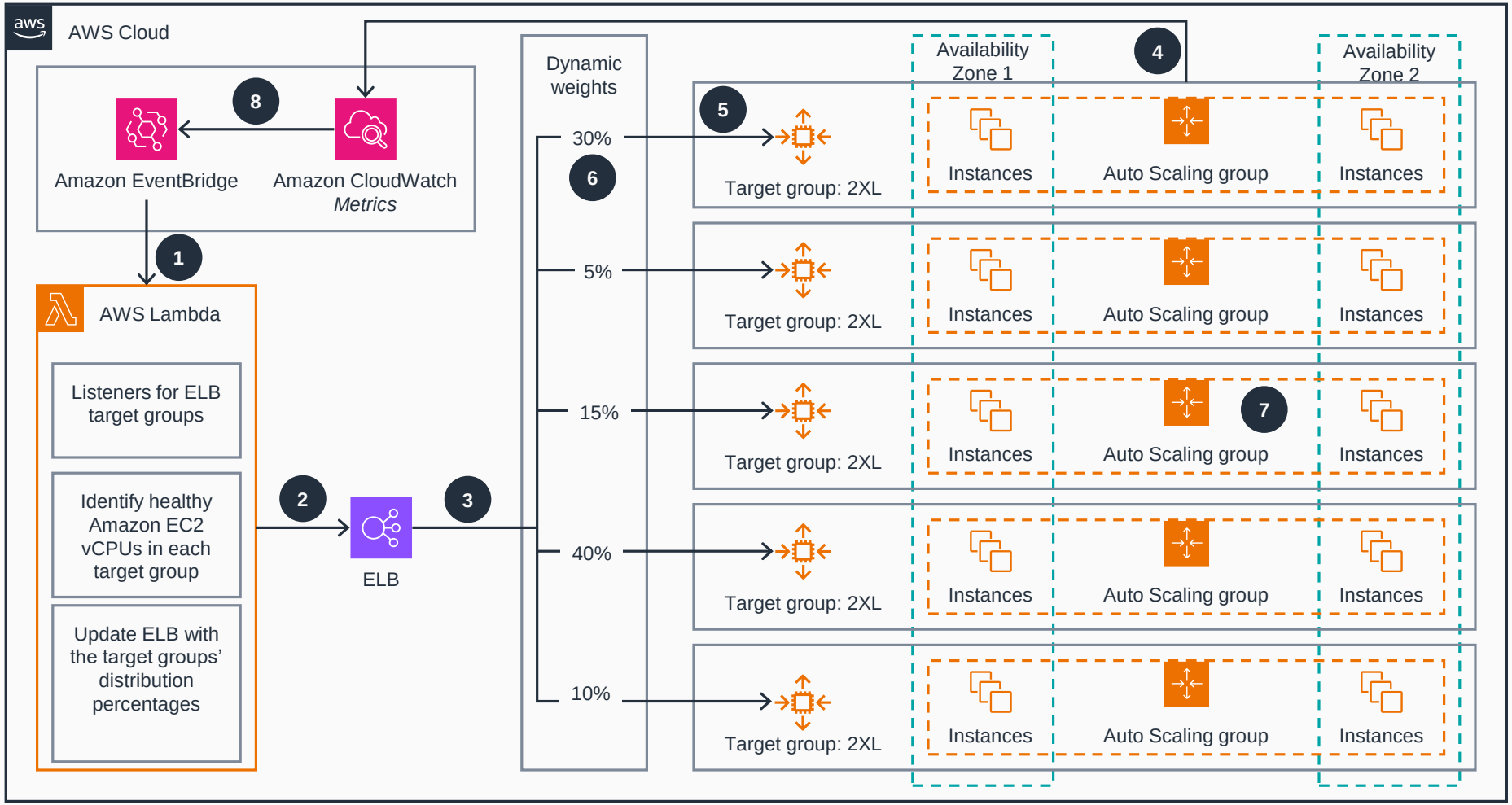
Guidance for Optimizing Heterogeneous Auto Scaling Group Resource Utilization on AWS

This architecture diagram shows how to use Elastic Load Balancing to efficiently distribute traffic across a heterogeneous Auto Scaling group by considering the capacity of individual Amazon EC2 instance target groups to optimize resource utilization.



1. Configure an **AWS Lambda** function with a load balancer Amazon Resource Name (ARN) through **Elastic Load Balancing** (ELB) and with listener ARNs. You can have multiple listeners for each load balancer.
2. The **Lambda** function updates the target group weights for each listener dynamically and periodically (default: 15 minutes).
3. **ELB** dynamically routes traffic based on the weighted percentage of the target groups.
4. Define multiple homogeneous **Amazon Elastic Compute Cloud (Amazon EC2)** Auto Scaling groups. Configure similarly capable instances in a single Auto Scaling group.
5. Define up to five target groups. Map multiple Auto Scaling groups to corresponding target groups. For example, you could group instances that are based on virtual CPUs (vCPUs).
6. The diagram shows example percentages of **ELB** listeners' forwarding weights to each target group.
7. Use attribute-based instance type selection to select similarly capable instances.
8. Configure a **Lambda** function to update the **ELB** target group weights at a specified interval (for instance, every 5 minutes) or based on **Amazon CloudWatch** metrics or **Amazon EventBridge** events.

Reviewed for technical accuracy May 21, 2024

AWS Reference Architecture