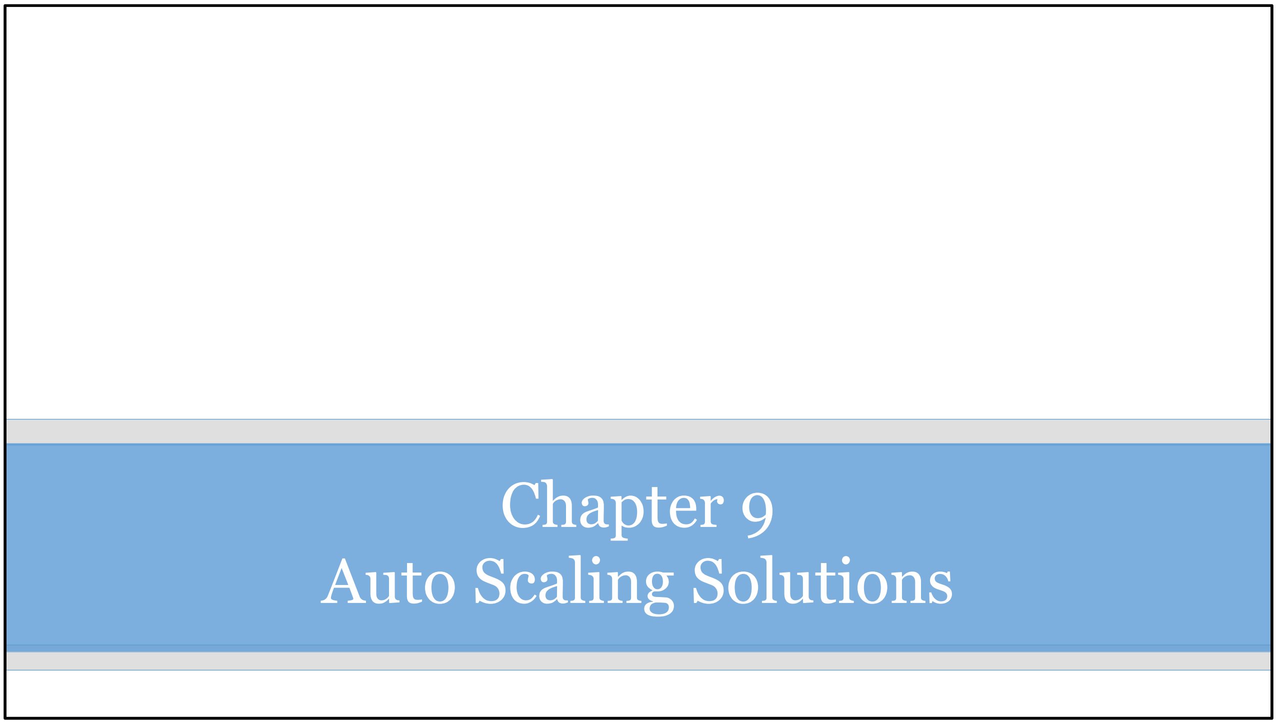
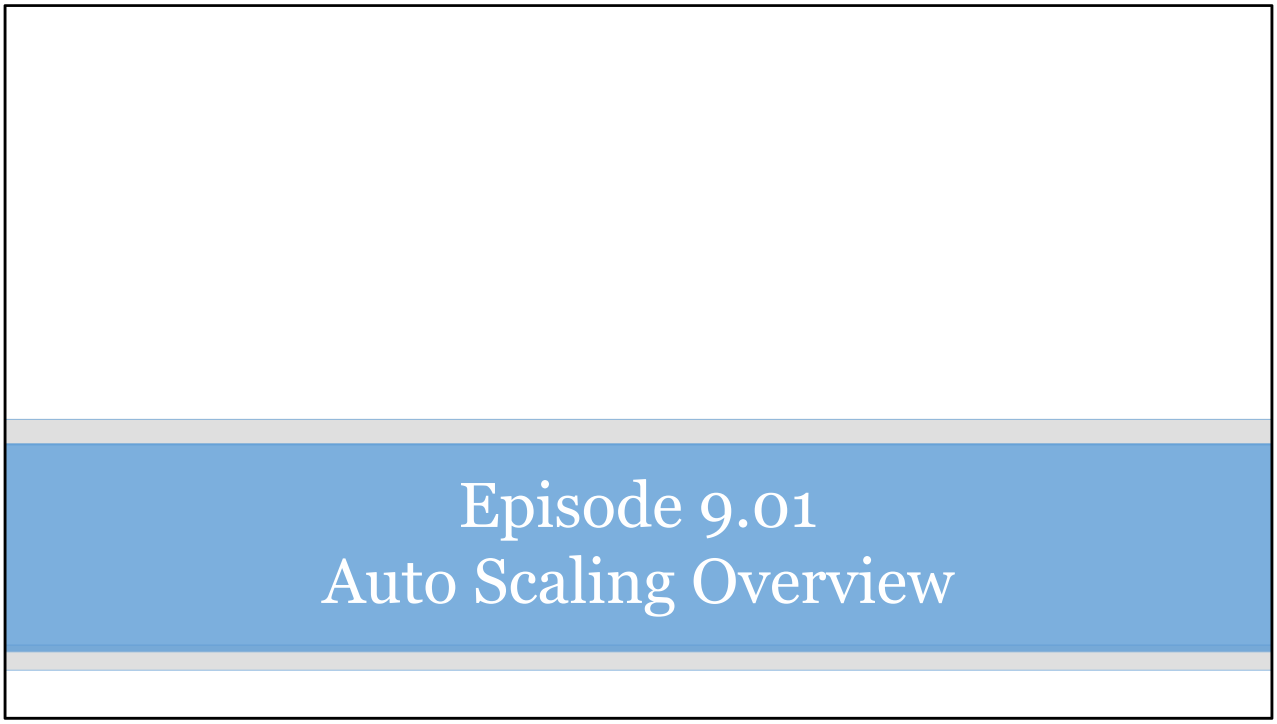
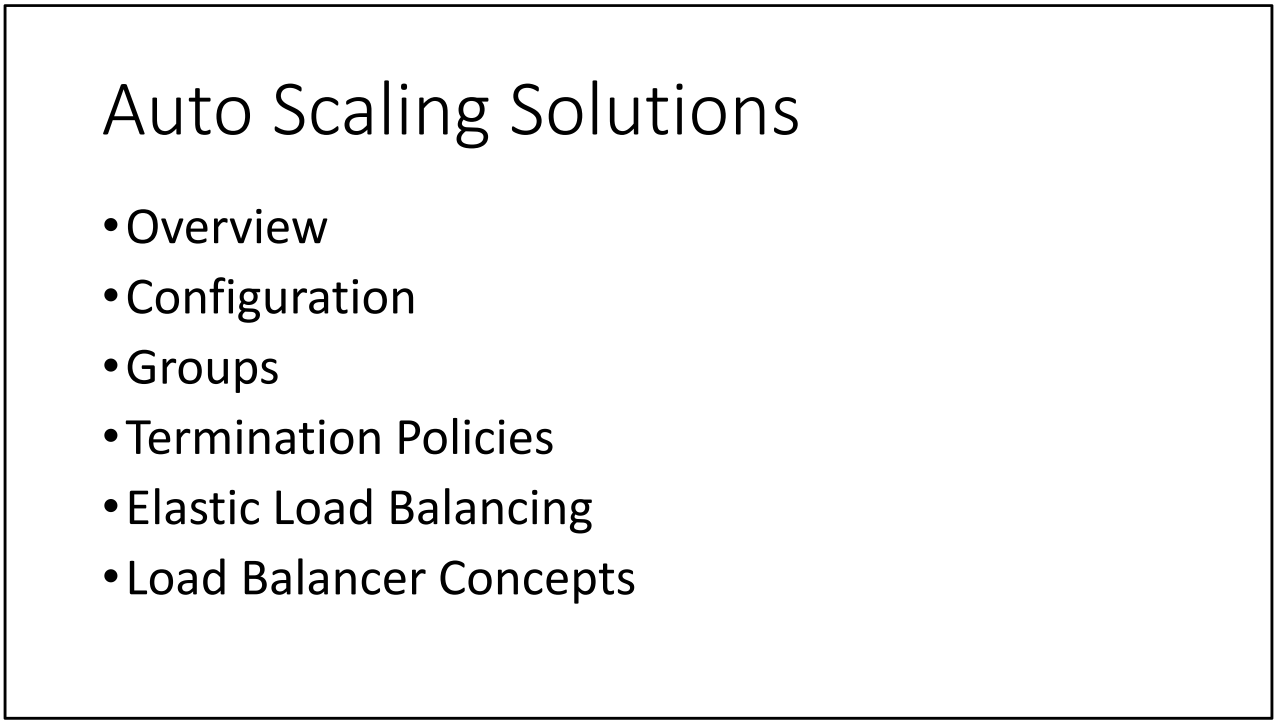


Chapter 9
Auto Scaling Solutions

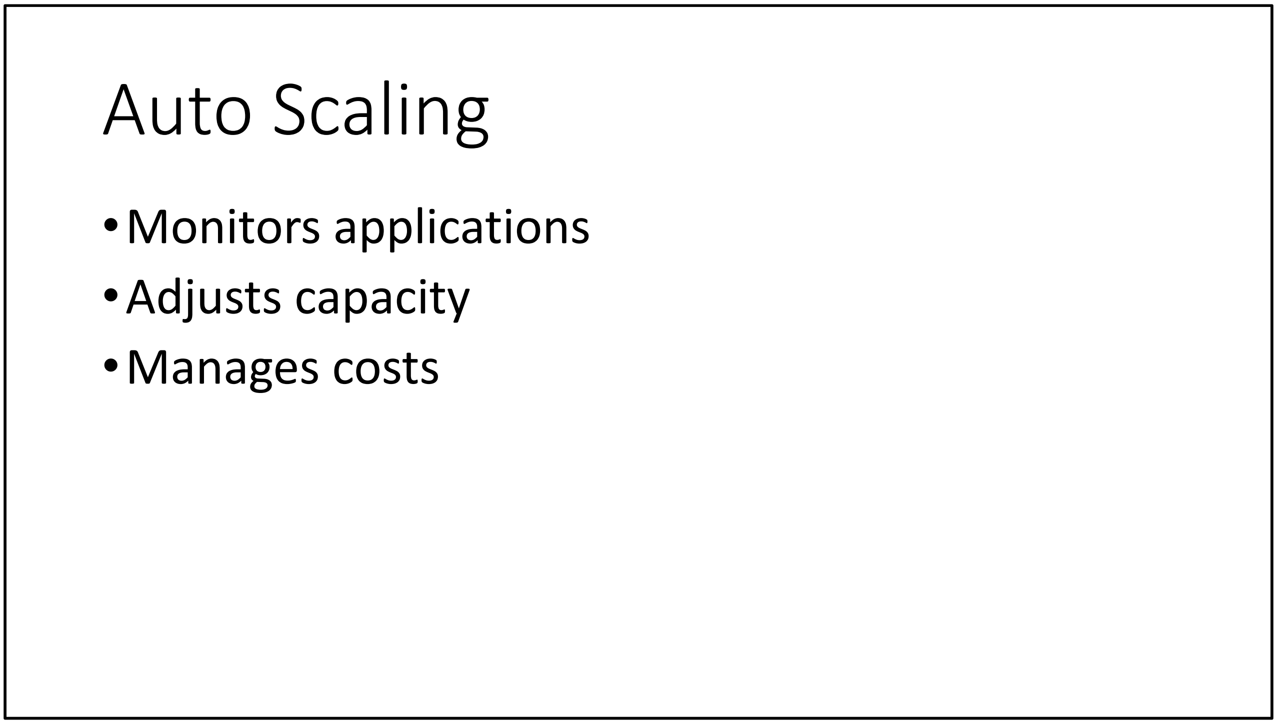


Episode 9.01
Auto Scaling Overview



Auto Scaling Solutions

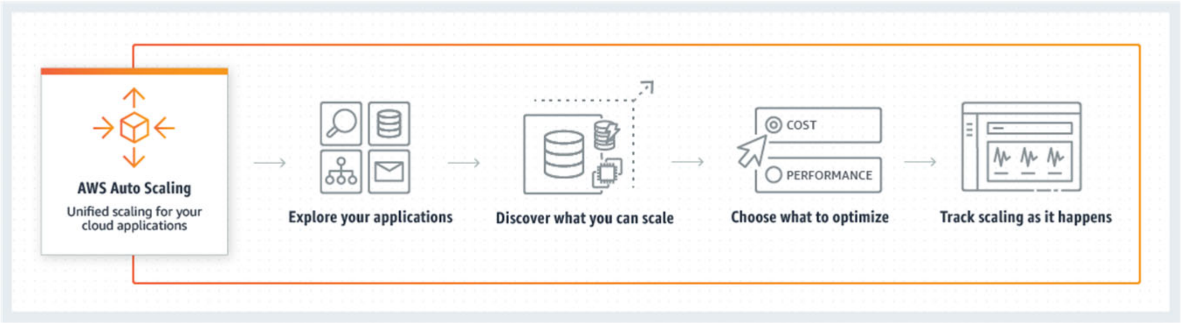
- Overview
- Configuration
- Groups
- Termination Policies
- Elastic Load Balancing
- Load Balancer Concepts

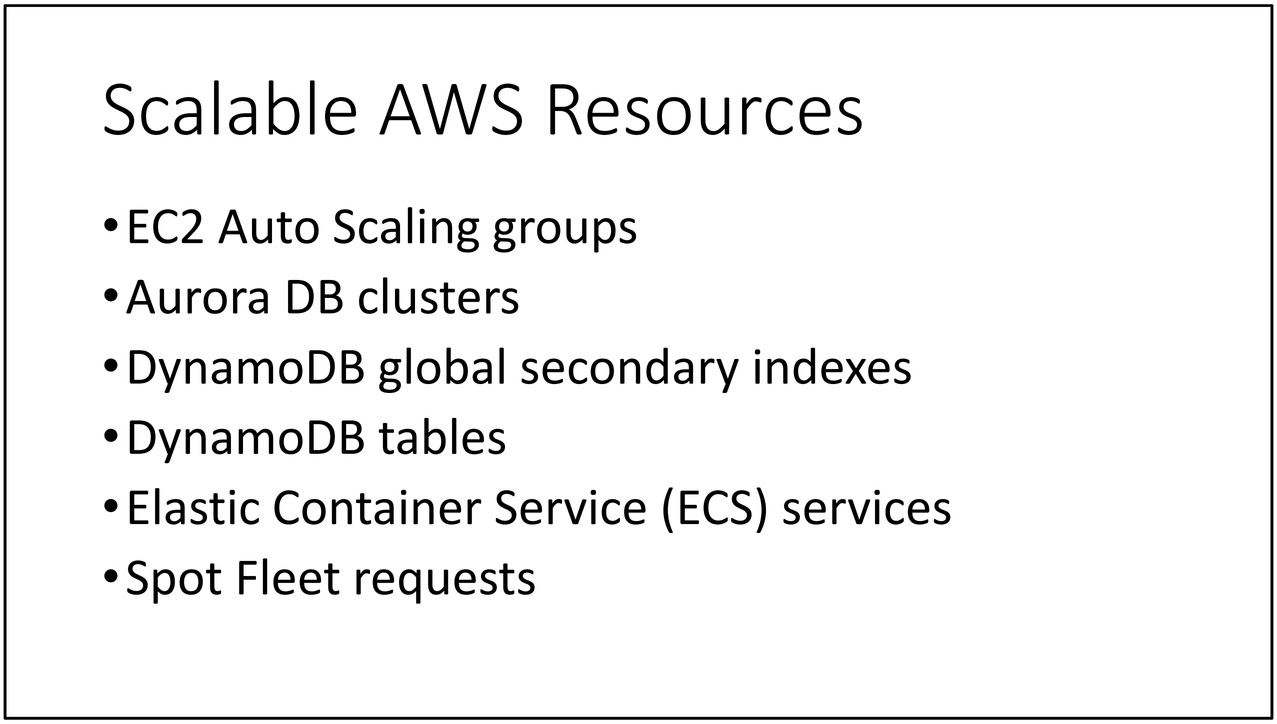


Auto Scaling

- Monitors applications
- Adjusts capacity
- Manages costs

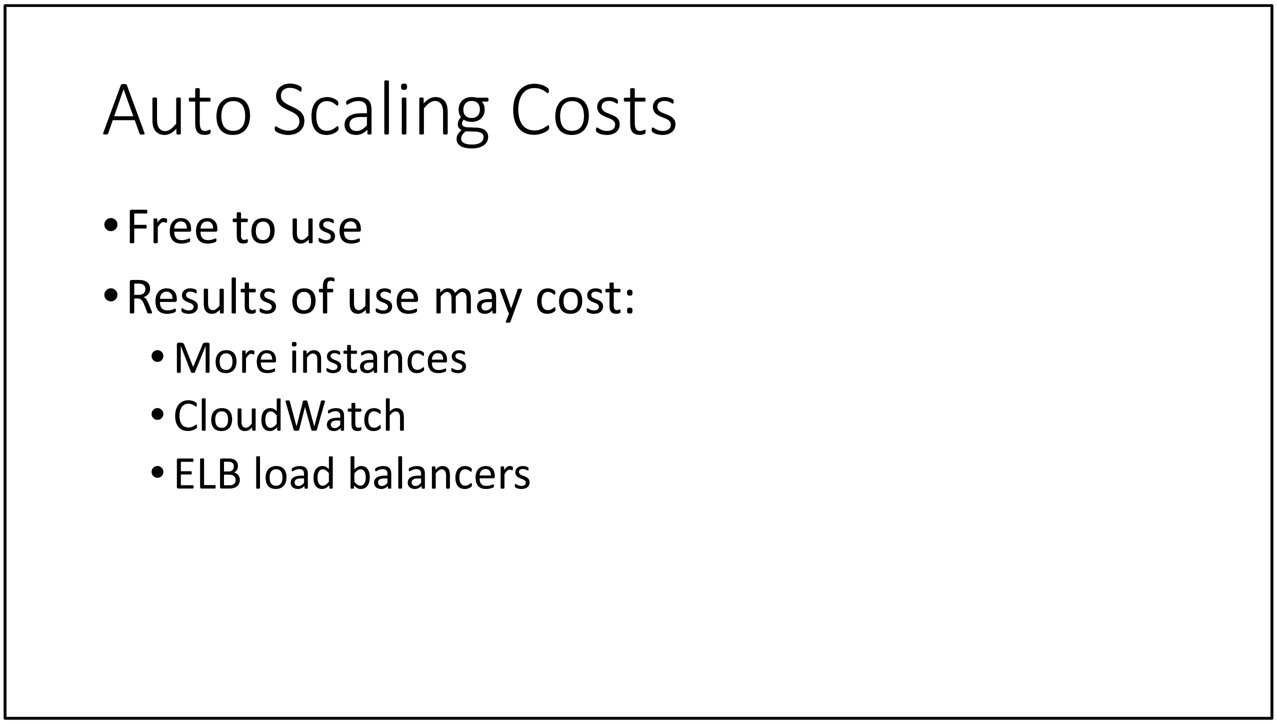
Auto Scaling Functionality


AWS Auto Scaling
Unified scaling for your cloud applications
Explore your applications
Discover what you can scale
COST
PERFORMANCE
Choose what to optimize
Track scaling as it happens



Scalable AWS Resources

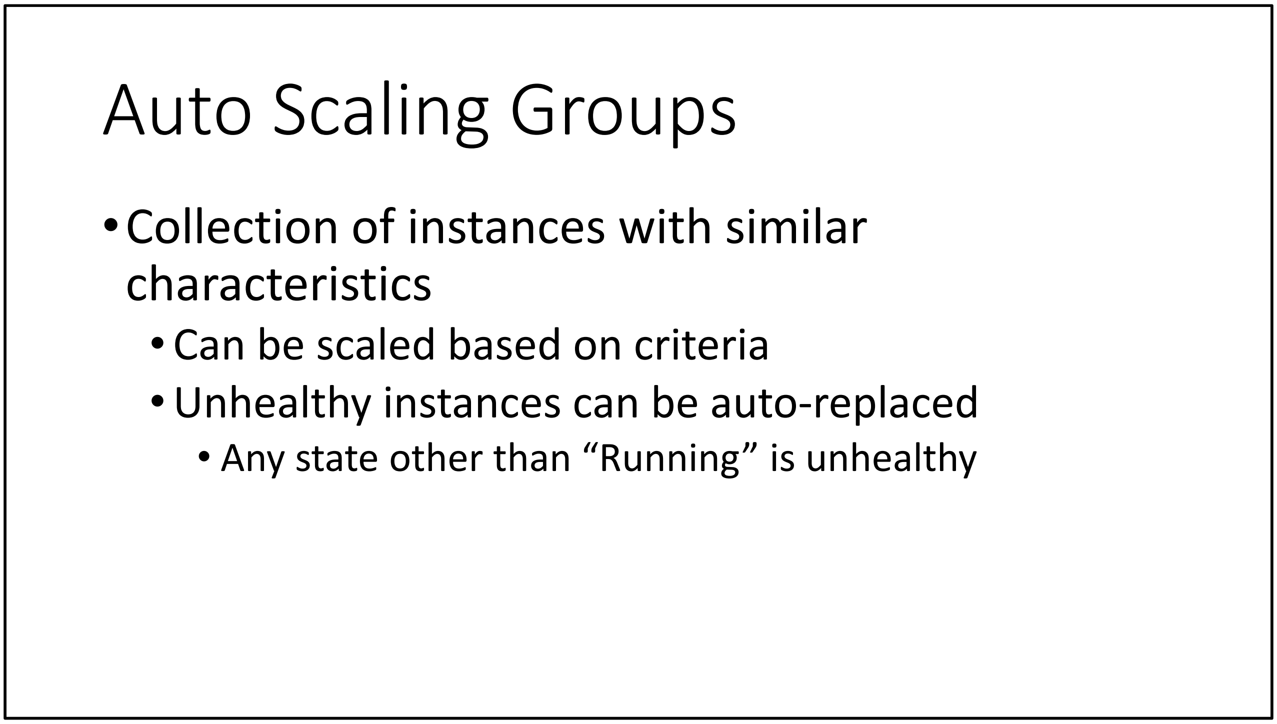
- EC2 Auto Scaling groups
- Aurora DB clusters
- DynamoDB global secondary indexes
- DynamoDB tables
- Elastic Container Service (ECS) services
- Spot Fleet requests



Auto Scaling Costs

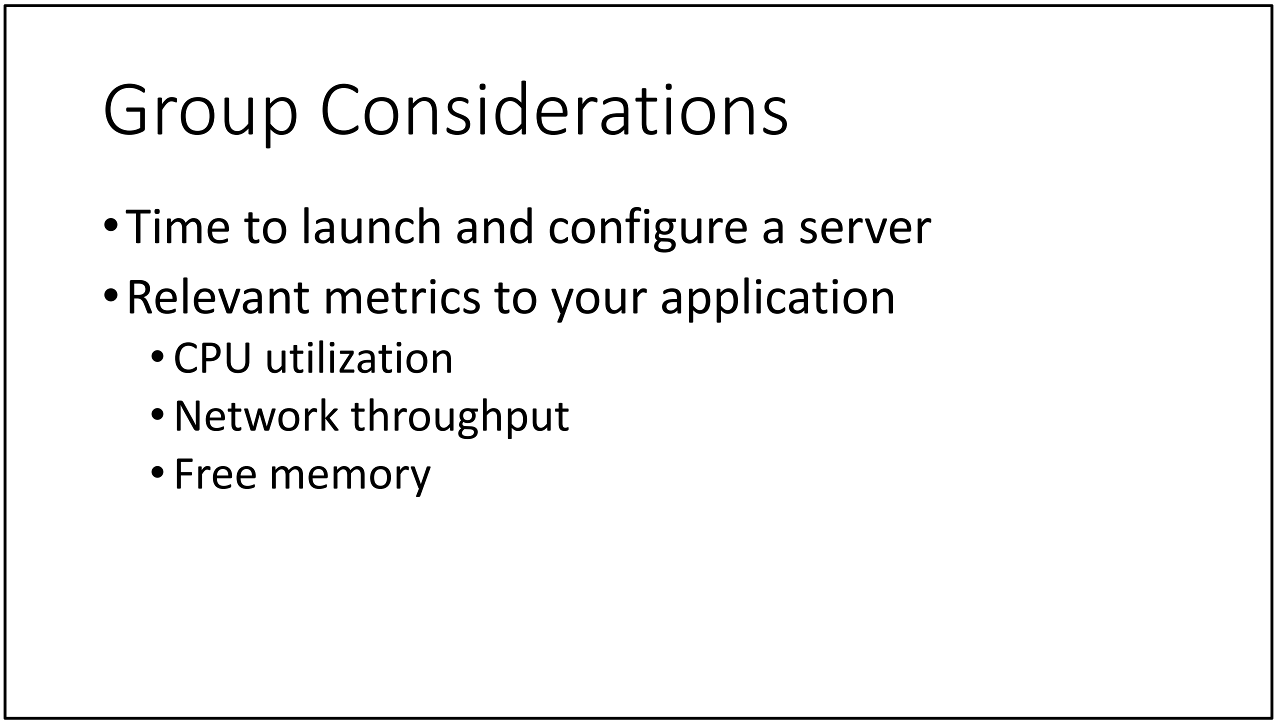
- Free to use
- Results of use may cost:
  - More instances
  - CloudWatch
  - ELB load balancers

Episode 9.02
Auto Scaling Groups



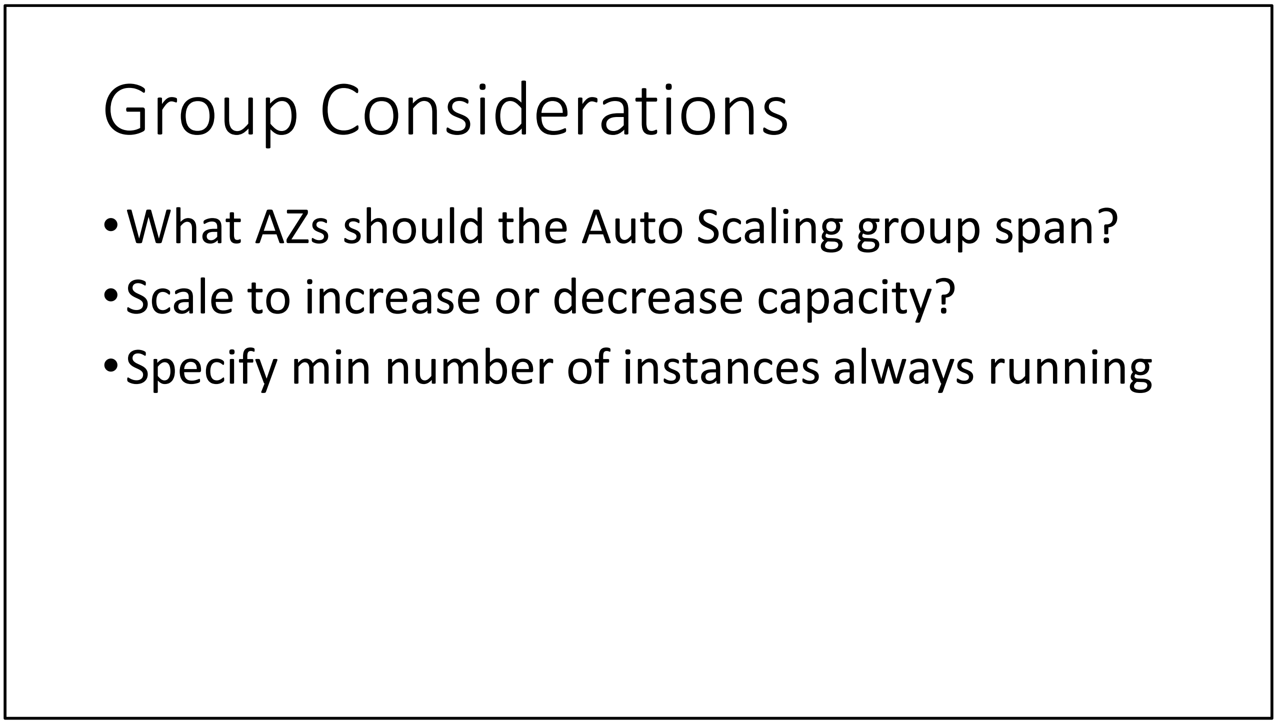
Auto Scaling Groups

- Collection of instances with similar characteristics
  - Can be scaled based on criteria
  - Unhealthy instances can be auto-replaced
    - Any state other than “Running” is unhealthy



Group Considerations

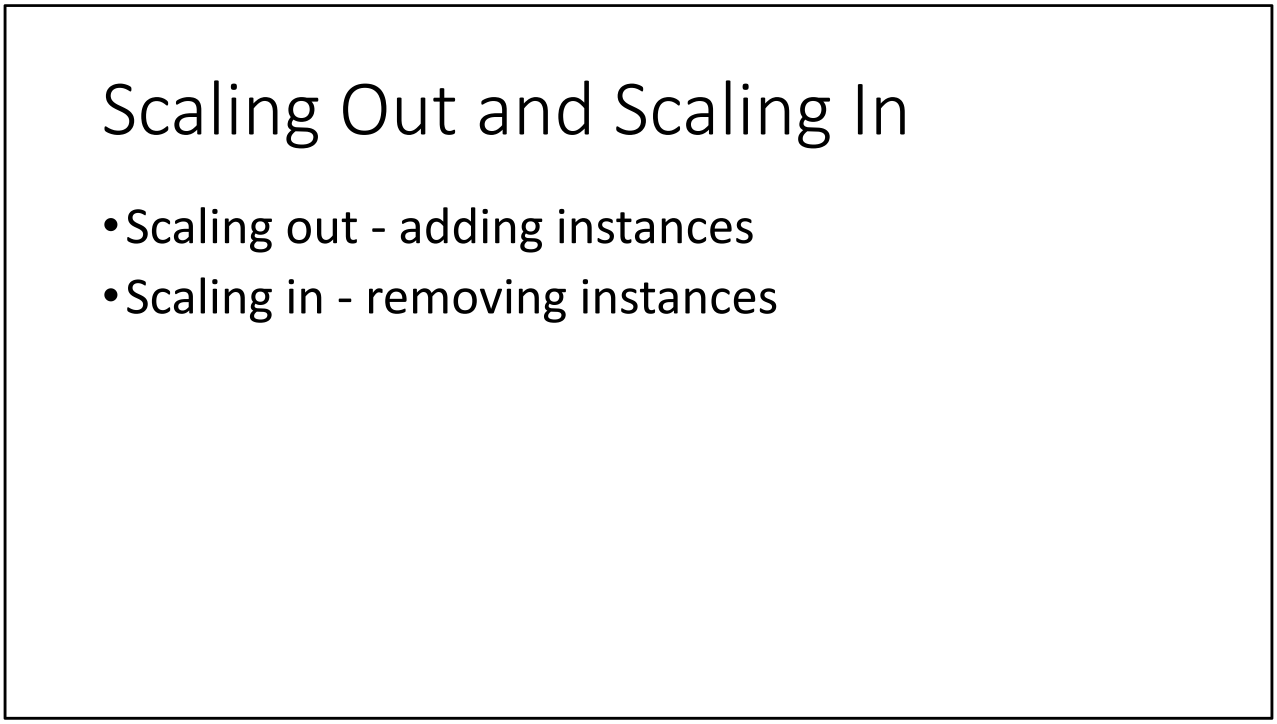
- Time to launch and configure a server
- Relevant metrics to your application
  - CPU utilization
  - Network throughput
  - Free memory



Group Considerations

- What AZs should the Auto Scaling group span?
- Scale to increase or decrease capacity?
- Specify min number of instances always running

Episode 9.03
Termination Policies

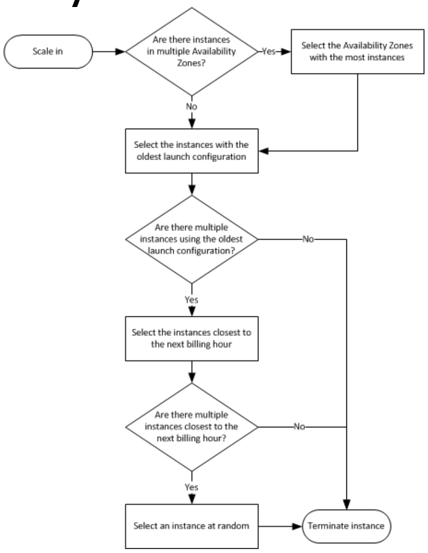


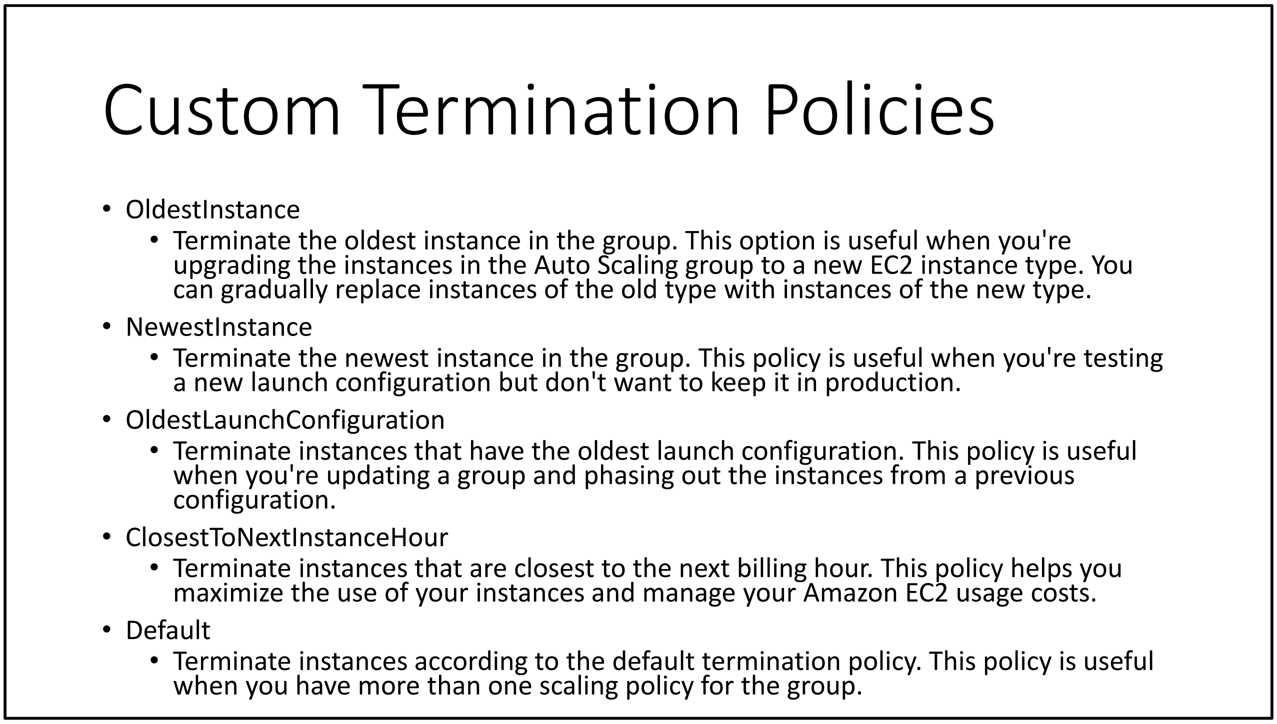
Scaling Out and Scaling In

- Scaling out - adding instances
- Scaling in - removing instances

Default Termination Policy

- If there are instances in multiple Availability Zones, select the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, select the Availability Zone with the instances that use the oldest launch configuration.
- Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.
- If there are multiple instances that use the oldest launch configuration, determine which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances and manage your Amazon EC2 usage costs.) If there is one such instance, terminate it.
- If there is more than one unprotected instance closest to the next billing hour, select one of these instances at random.

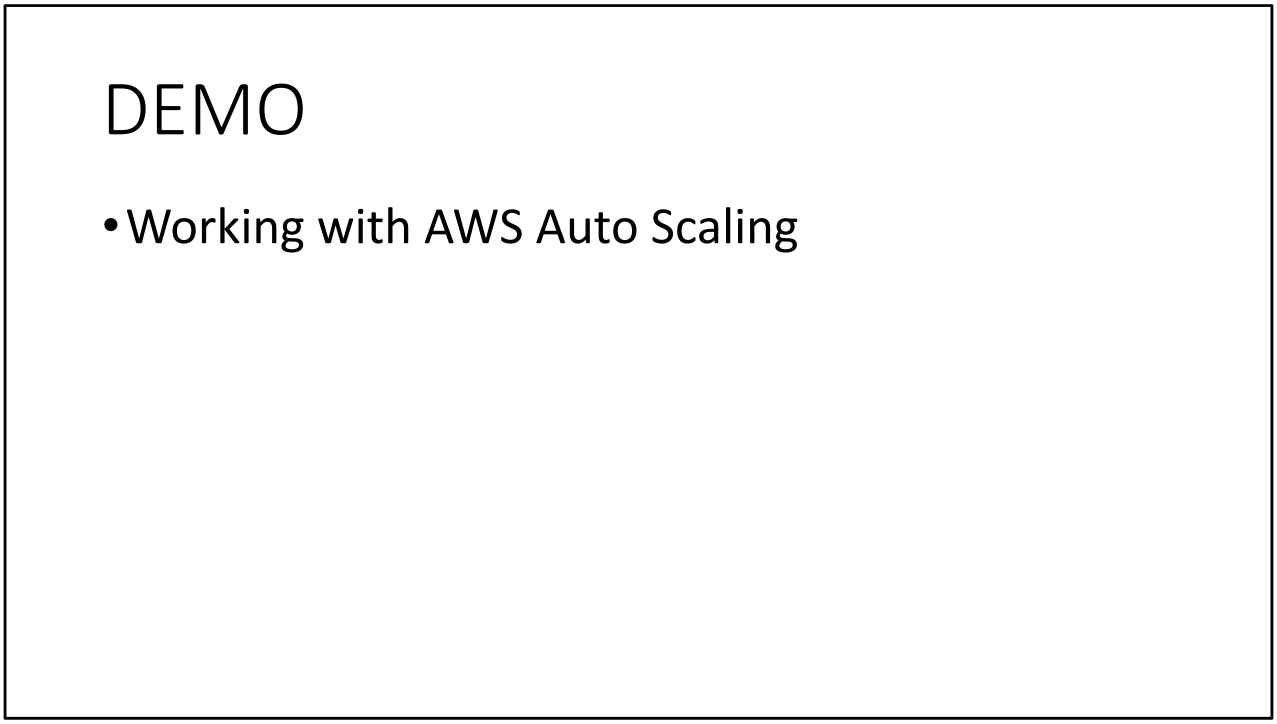

Scale in
Are there instances in multiple Availability Zones?
Yes
Select the Availability Zones with the most instances
No
Select the instances with the oldest launch configuration
Are there multiple instances using the oldest launch configuration?
No
Yes
Select the instances closest to the next billing hour
Are there multiple instances closest to the next billing hour?
No
Yes
Select an instance at random
Terminate instance



Custom Termination Policies

- OldestInstance
  - Terminate the oldest instance in the group. This option is useful when you're upgrading the instances in the Auto Scaling group to a new EC2 instance type. You can gradually replace instances of the old type with instances of the new type.
- NewestInstance
  - Terminate the newest instance in the group. This policy is useful when you're testing a new launch configuration but don't want to keep it in production.
- OldestLaunchConfiguration
  - Terminate instances that have the oldest launch configuration. This policy is useful when you're updating a group and phasing out the instances from a previous configuration.
- ClosestToNextInstanceHour
  - Terminate instances that are closest to the next billing hour. This policy helps you maximize the use of your instances and manage your Amazon EC2 usage costs.
- Default
  - Terminate instances according to the default termination policy. This policy is useful when you have more than one scaling policy for the group.

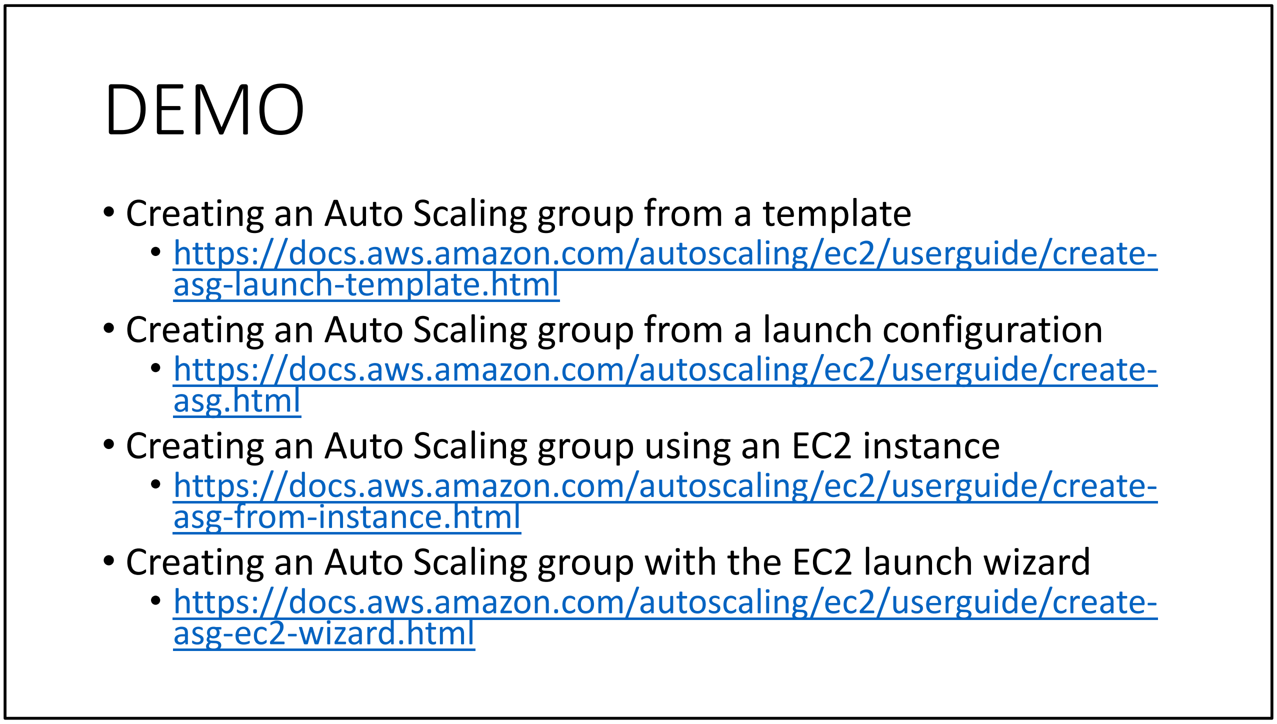
Episode 9.04
Auto Scaling Configuration Lab



DEMO

- Working with AWS Auto Scaling

Episode 9.05
Launch Methods

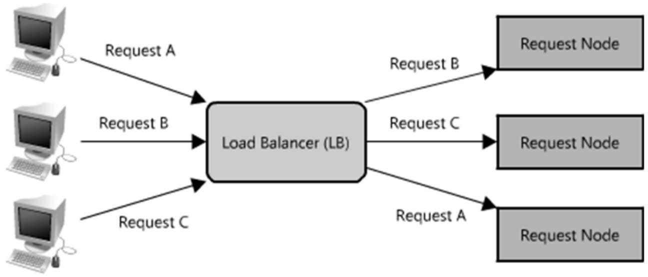


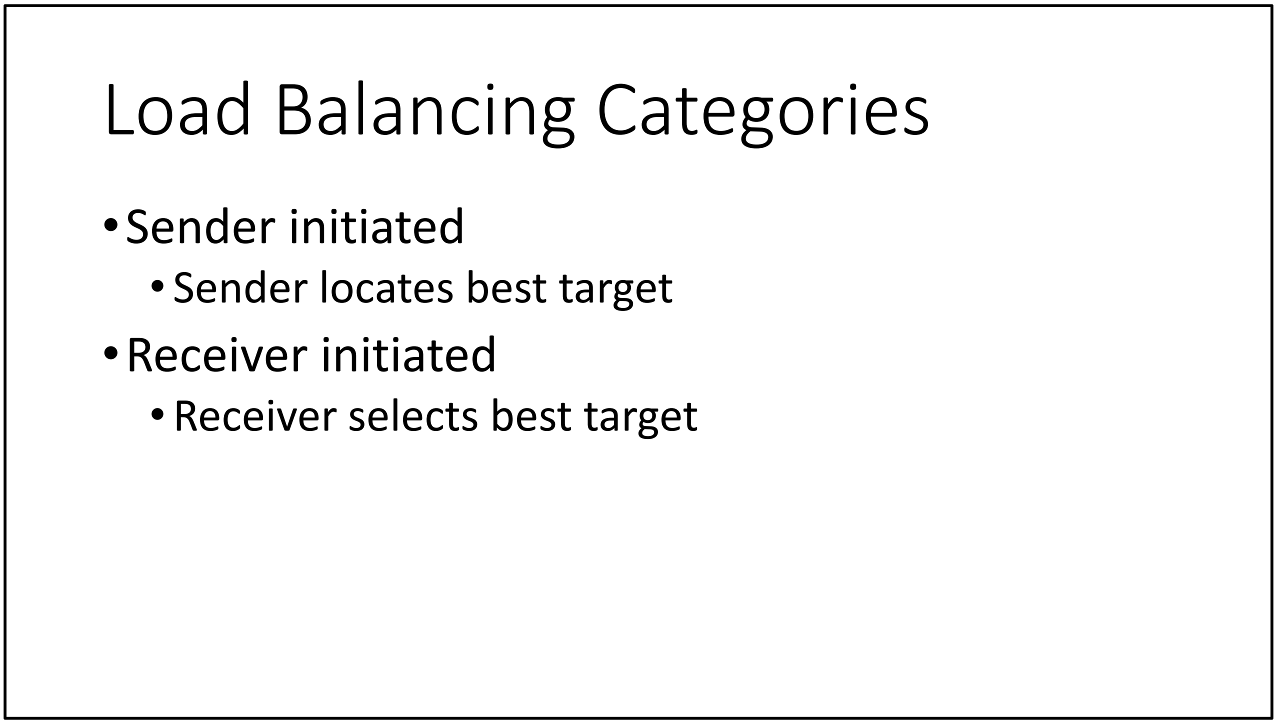
DEMO

- Creating an Auto Scaling group from a template
  - https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg-launch-template.html
- Creating an Auto Scaling group from a launch configuration
  - https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg.html
- Creating an Auto Scaling group using an EC2 instance
  - https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg-from-instance.html
- Creating an Auto Scaling group with the EC2 launch wizard
  - https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg-ec2-wizard.html

Episode 9.06
Load Balancer Concepts

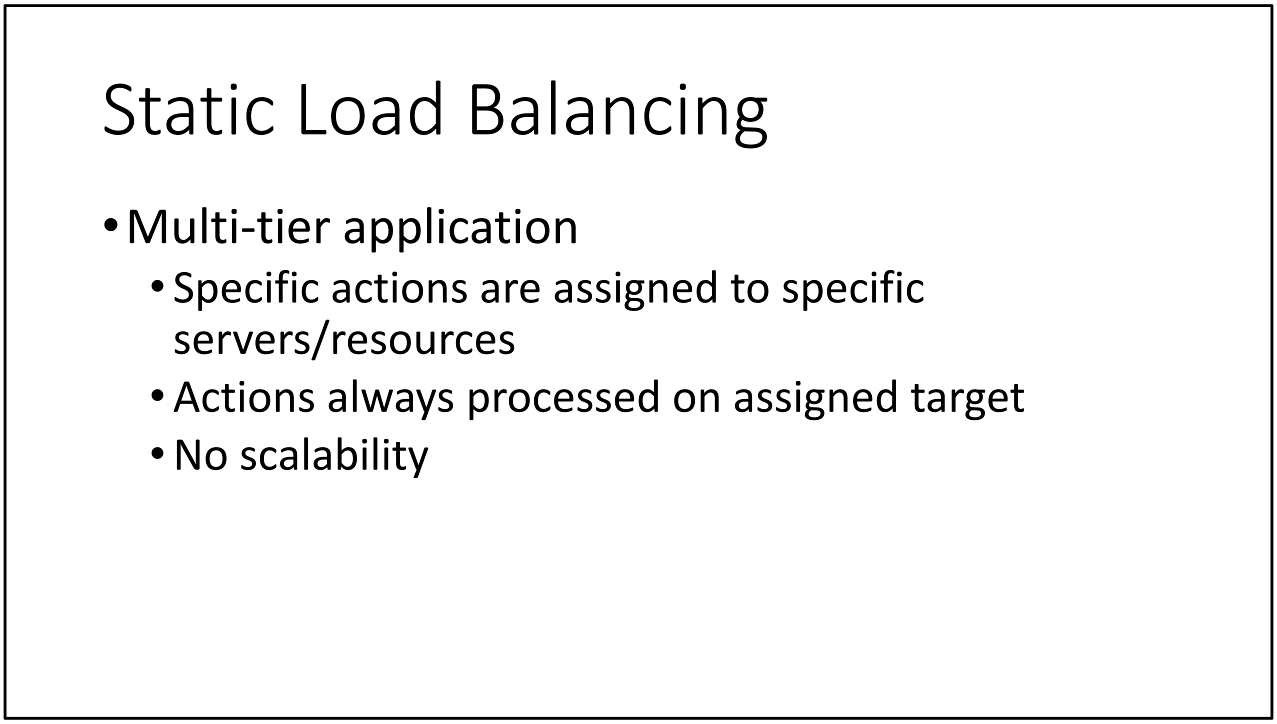
Load Balancing Defined





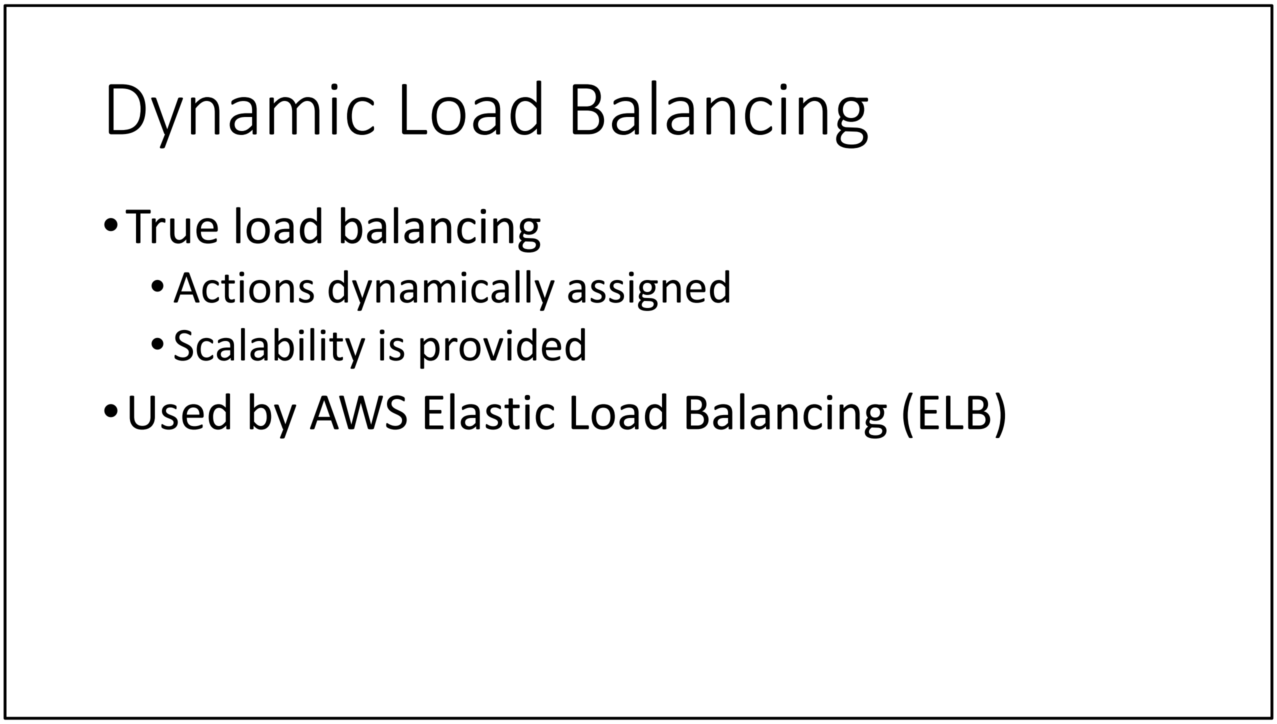
Load Balancing Categories

- Sender initiated
  - Sender locates best target
- Receiver initiated
  - Receiver selects best target



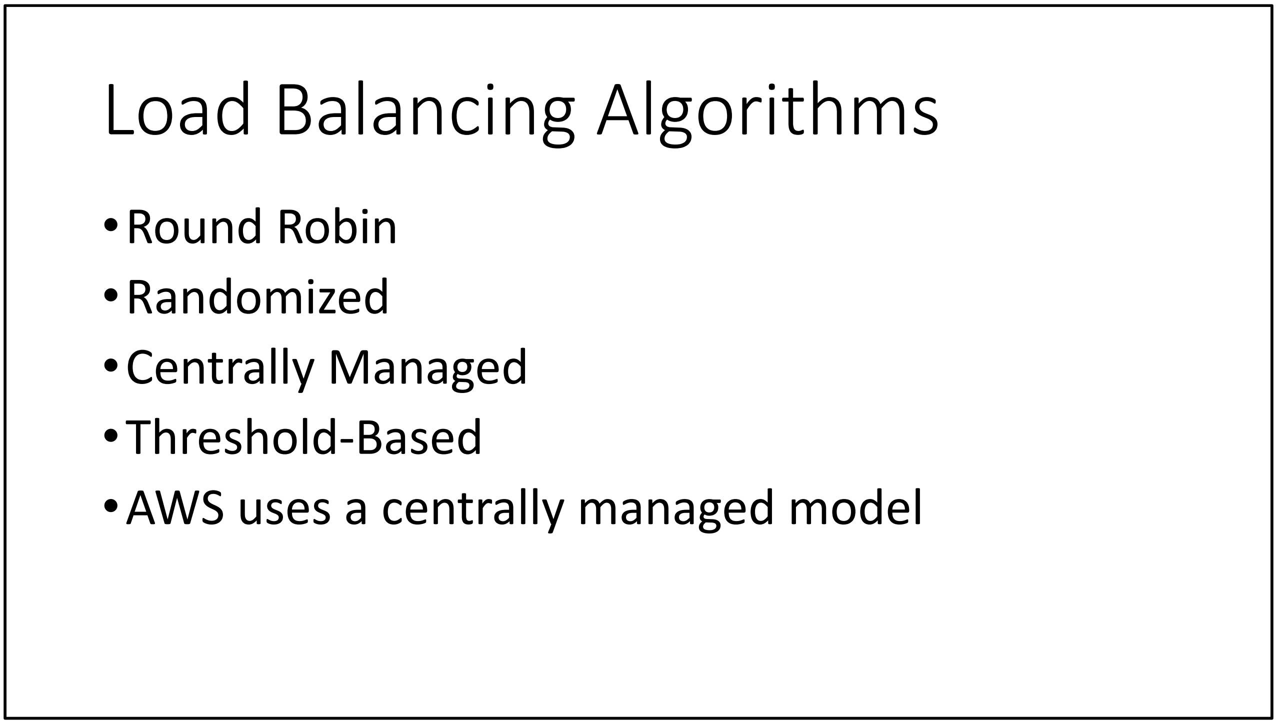
Static Load Balancing

- Multi-tier application
  - Specific actions are assigned to specific servers/resources
  - Actions always processed on assigned target
  - No scalability



Dynamic Load Balancing

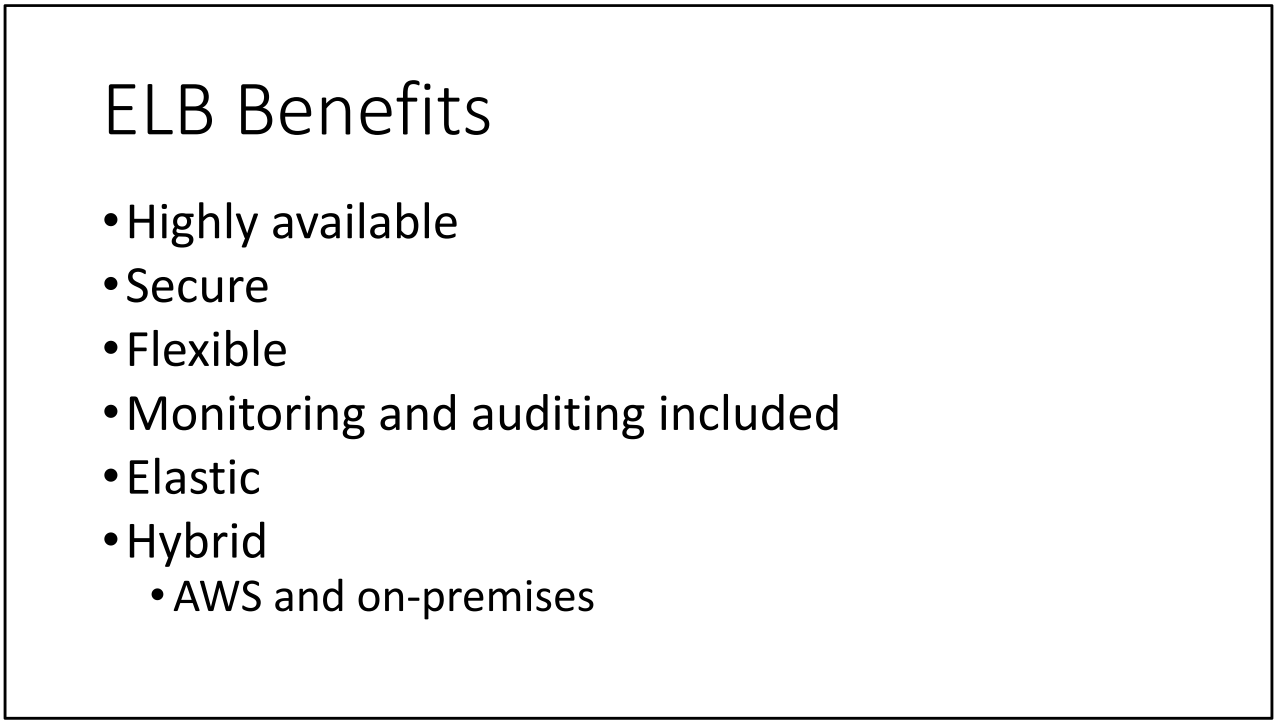
- True load balancing
  - Actions dynamically assigned
  - Scalability is provided
- Used by AWS Elastic Load Balancing (ELB)



Load Balancing Algorithms

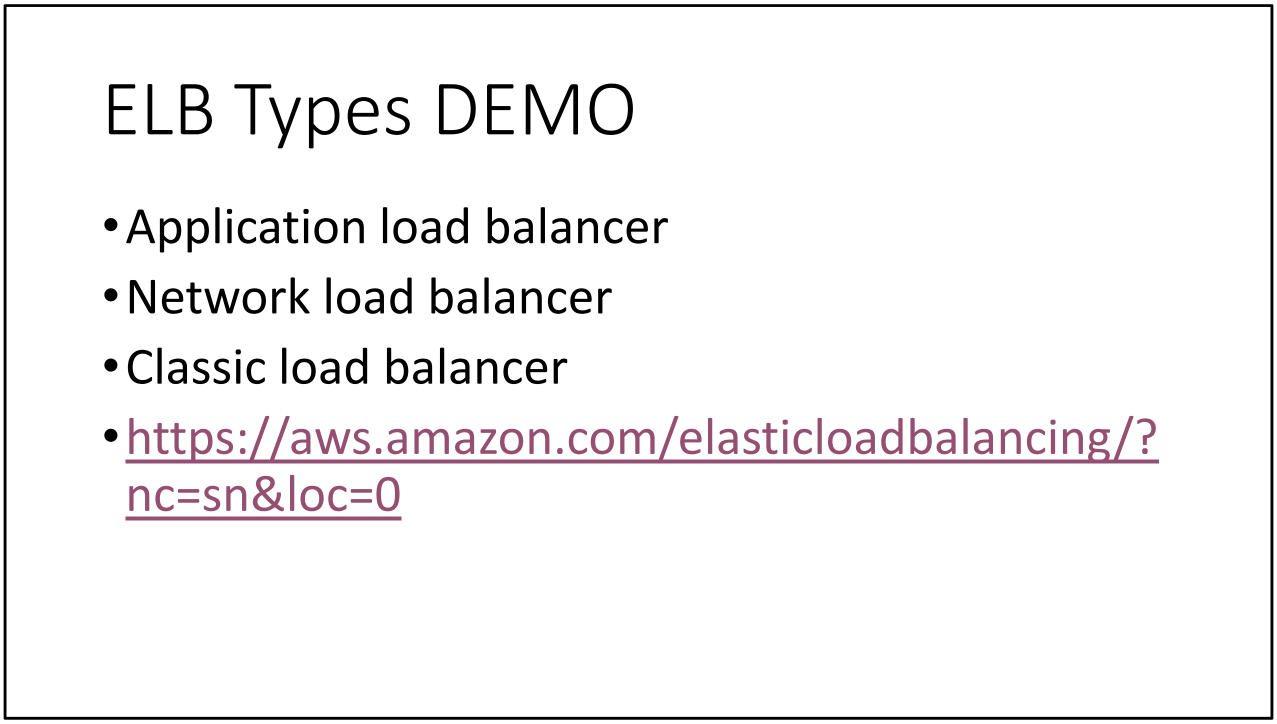
- Round Robin
- Randomized
- Centrally Managed
- Threshold-Based
- AWS uses a centrally managed model

Episode 9.07
Elastic Load Balancing (ELB)



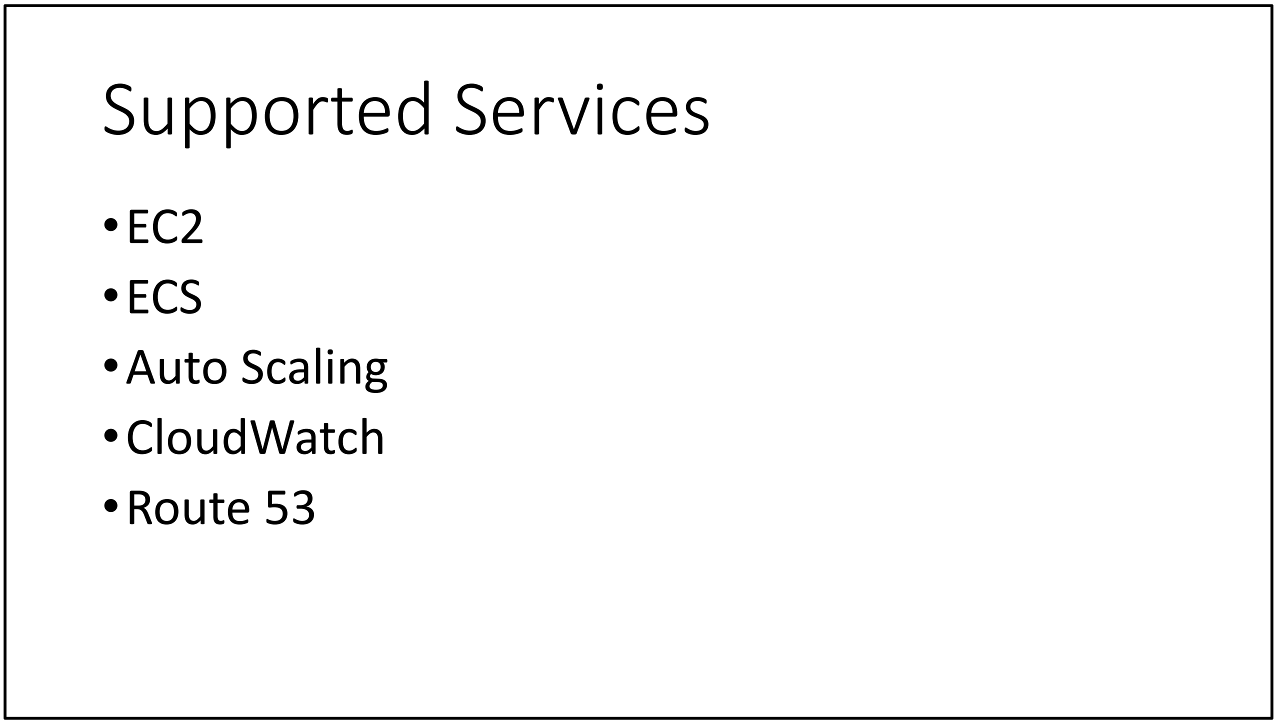
ELB Benefits

- Highly available
- Secure
- Flexible
- Monitoring and auditing included
- Elastic
- Hybrid
  - AWS and on-premises



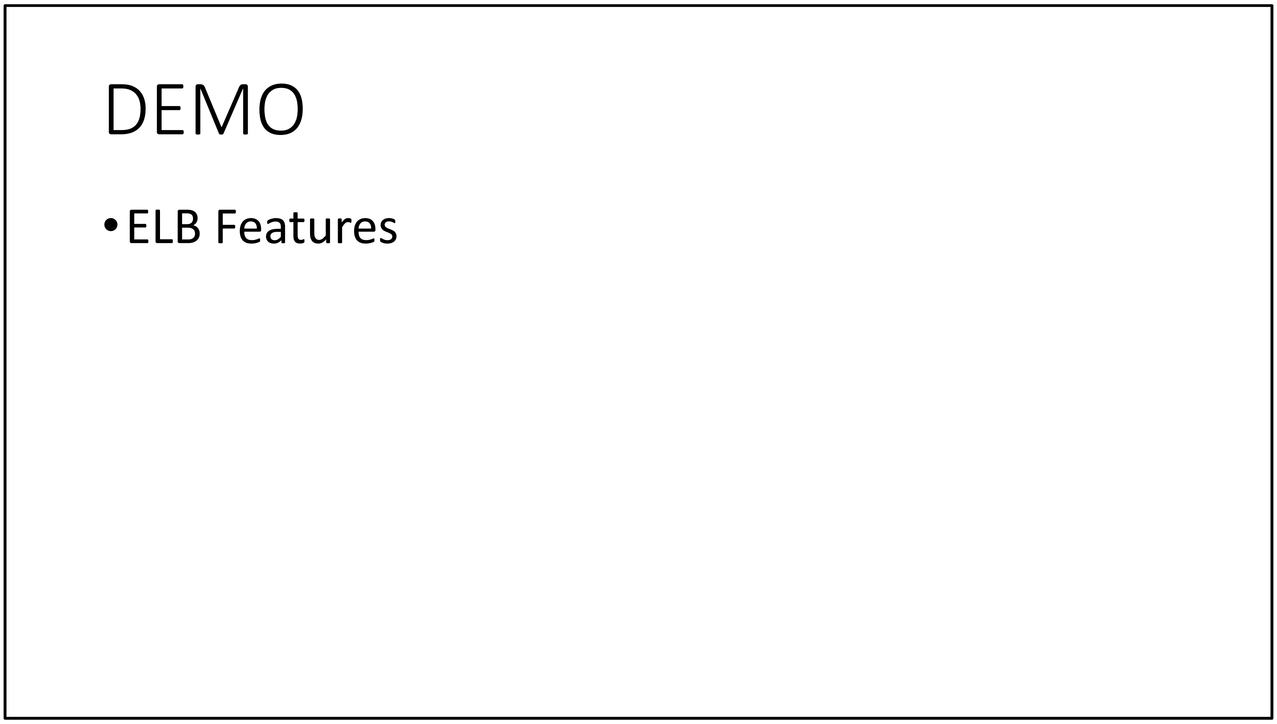
ELB Types DEMO

- Application load balancer
- Network load balancer
- Classic load balancer
- https://aws.amazon.com/elasticloadbalancing/?nc=sn&loc=0



Supported Services

- EC2
- ECS
- Auto Scaling
- CloudWatch
- Route 53



DEMO

- ELB Features

https://aws.amazon.com/elasticloadbalancing/features