Auto Scaling

Maintain Right Compute Capacity

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

amazon web services

Auto Scaling

- Maintain right amount of capacity for your application
- Improved fault tolerance – detect unhealthy instances, terminate and replace them
- Distribute instances across availability zones
- Dynamic scaling – increase or decrease capacity depending on current traffic demands
- Better cost management – keep only the instances that are needed and terminate them when not needed
- No additional fees

Scenarios

- Handling Variable Demand
- Multi-tier Application
- Availability Zone Instance Distribution
- Pending Request Based Scaling

- CloudWatch Alarm

Concept

Auto Scaling Group – A Collection of EC2 instances

Criteria	Description
Minimum	Minimum number of instances that auto scaling group should keep running at all times
Desired	Slider that can move between minimum and maximum. Number of instances that are required for current traffic demands. Auto scaling group adjusts running instances to meet desired count
Maximum	Maximum number of instances that auto scaling group can scale out to

Concept

Component	Purpose
Auto Scaling Group	A collection of EC2 instances that is treated as a logical unit
Launch Configuration	Template for EC2 instances that part of Auto Scaling Group – Includes AMI ID, Key Pair, Instance Type, Security Groups, User Data, Block Devices and so forth
Scaling Plans	When and How to scale – Scale based on specific CloudWatch metric value or at a scheduled window

Rebalancing

- Auto Scaling tries to maintain equal number of instances across all enabled availability zones
- Auto Scaling can rebalance instances when:
  - Availability Zone added or removed
  - Unhealthy Availability Zone is healthy
  - Explicit call to terminate instances
  - Spot market price changes – availability zone becomes affordable/meets your bid price

Rebalancing

- New instances are launched before terminating old ones
- May momentarily exceed maximum by greater of 10% or 1 instance - group is near maximum capacity and needs rebalancing

Auto Scaling Lifecycle

Figure: Auto Scaling Lifecycle

Scale Out Triggers - Lifecycle Event

- Manually increase group size (minimum, desired)
- Demand based scaling out
- Schedule based scaling out

Scale Out Actions

- Auto scaling group uses *Launch Configuration* to launch required number of instances
- Instances are initially in *pending* state
- Optional – attach custom actions using lifecycle hooks
- Instances enter *InService* state once fully configured and after it passes EC2 health checks
- Newly added instance is counted against the desired capacity

Instances In Service

Instance remain *InService* until

- A Scale-in event occurs
- Put the instance to *Standby* mode
- *Detach* the instance from Auto Scaling group
- Instance fails health check

Scale In Triggers

Create a scale in trigger for every scale out trigger you have

Scale in can occur under:

- Manually decrease group size
- Demand based scaling in
- Schedule based scaling in

Scale In Actions

- Instance selected for terminated are *Detached* from auto scaling group
- Instances enter *Terminating* state
- Optional – attach custom actions using lifecycle hooks
- Instances are terminated and enter *Terminated* state

Attach an Instance

- Attach an existing *running* instance to auto scaling group
- Needs to meet following criteria:
  - AMI should still be available
  - Instance is not part of another auto scaling group
  - Same availability zone as auto scaling group
  - If load balancer is attached to a scaling group, then instance and load balancer must be in the same VPC or EC2-Classic
- Desired capacity is automatically increased – if it exceed maximum, operation fails
- Automatically registered with Load Balancers (if any)

Detach an Instance

- Detach an instance from auto scaling group
- No longer part of auto scaling group
- Detached Instance can be attached to a different auto scaling group
- Desired Capacity is decremented – if not decremented, Auto scaling launches a replacement instance
- Instance is deregistered from Load Balancers (if any)

Standby

- Move any *InService* to *Standby* state
- Standby instances are still part of Auto Scaling Group
- Allows you to remove instance from service for troubleshooting or upgrade and put it back into service
- Decrements desired capacity – to prevent launch of new replacement instances
- Auto scaling does not perform health checks on standby instances
- Instances are deregistered from load balancers (if any)

Exiting Standby

- Put *Standby* to *InService*
- Desired Capacity is incremented
- Instance is registered with Load balancers (if any)
- Health check resumes on the instance

Demo 1

Launch Auto Scaling Group with 2 instances

- Demonstrate EC2 Health Checks
- Demonstrate ELB Health Checks
- Demonstrate Automatic Recovery

Demo 2

Launch Availability Zone rebalancing

- Launch Auto Scaling Group in a single AZ
- Attach two instances
- Add second AZ to the ASG
- Observe rebalancing

Demo 3

Add scaling policy to increase desired capacity

Add scaling policy to decrease desired capacity

Launch Configuration

- Template for EC2 instances – similar to what you specify in the EC2 management console
- Used by Auto Scaling Group to launch instances
- Contains important details about EC2 instances
    - Amazon Machine Image (AMI ID)
    - Instance Type, Size
    - Security Groups
    - Key Pair
    - Block Device mapping
    - User Data for customization

Launch Configuration

- Create Launch Configuration from Console
- Create using an existing EC2 instance – Auto Scaling automatically creates a Launch configuration
    - When you create a new auto scaling group from existing instances
- One launch configuration per Auto Scaling Group
- Many Auto Scaling Groups can use the same launch configuration
- Launch configuration cannot be modified – need to create a new one

Auto Scaling Group

- A collection of EC2 instances that is treated as a logical unit. Consists of:
- A Launch Configuration
- Capacity information
- Availability Zones (EC2-Classic) or Subnets (VPC)
- Health Checks
- Metrics – for reporting and scaling out and in
- If you delete Auto Scaling Group, all scaling group instances are terminated

Elastic Load Balancers

- Attach one or more Elastic Load Balancers (Classic Load Balancer)
- Attach one or more Target Groups (Application Load Balancer)
- Instances that are part of Auto Scaling Group are automatically registered with ELB
- Instances are automatically deregistered from ELB when removed or terminated

Elastic Load Balancers

- When Load Balancer is added, all instances are registered as part of the Load Balancer
- When at least one instance passes the health check, Load Balancer enter *InService* state
- Only after Load Balancer enters *InService* state, Auto Scaling starts monitoring and can terminate, replace instances
    - Protects against misconfigured health checks

Health Checks

- Auto Scaling Group periodically verifies the instance status check results
- Instances that fails instance status checks are considered unhealthy. They are terminated and replaced
- If instances are part of ELB, you can also enable Auto Scaling Group to use ELB health checks results
- Auto scaling group configured to use ELB health checks can detect failed ELB health checks and replace the unhealthy instances

Scaling Options

- Constant – Maintain desired number of instances
- Manual – Adjust manually desired number of instances
- Scheduled – Time and Date based adjustment
- Dynamic – Scale based on demand
    - Simple Scaling – single scaling adjustment based on alarm breach
    - Step Scaling – Adjust based on magnitude of alarm breach
- Instances may take several minutes to be *InService* state (launch delay, software installation, and so forth)

Multiple Scaling Policies

- Multiple scaling policies can be attached to an auto scaling group
- Generally, a scale-out policy should have a corresponding scale-in policy
- You could have different scaling policies based on different metrics
- If two policies are triggered at the same time, policy that has greatest impact to scaling group is used – larger number of instances in service

Dynamic Scaling

- Adjust capacity based on demand
- CloudWatch Alarm triggers a scale-out or scale-in action
- Adjustment Types
    - Change In Capacity – Increase or decrease by specified count
    - Exact Capacity – New desired capacity
    - Percent Change In Capacity – Increase or decrease by specified percentage
- Simple or Step Scaling Policies

Simple Scaling Policy

- Single scaling adjustment based on alarm breach
- When scaling action is in progress – a cooldown period is enforced
- New alarm breaches during cooldown period are ignored – gives a chance for new instances to handle the traffic
- After cooldown period expires – any new alarm breach is handled according to policy
- Default cooldown period is 300 seconds
- Unhealthy instances are handled right away

Step Scaling Policy

- Step Scaling - Adjust based on magnitude of alarm breach
- No cooldown period – continuous evaluation of alarm breaches
- Actions are defined based as series of steps
    - 50-60% CPU utilization – no action
    - 60-70% CPU utilization – add 1 instance
    - > 70% CPU utilization – add 3 instances

Step Scaling Policy

- Instance Warmup Period specifies number of seconds needed for instance to start handling traffic
  - Instance is not considered towards aggregated metrics of auto scaling until warmup period expires
  - Instance is not considered as part of current capacity during warmup
  - Multiple alarm breaches at same step are treated as a single scaling action
  - Alarm breaches at different steps are evaluated – and any additional instances are launched

Example: Step Scaling

- CloudWatch Alarm raised when Average CPU utilization above 50% for auto scaling group in 1 minute time period
- Policy:
  - 50-60% CPU utilization – no action
  - 60-70% CPU utilization – add 1 instance
  - > 70% CPU utilization – add 3 instances
  - Warmup Period – 300 Seconds

Example: Step Scaling

Alarm received by Auto Scaling Group

- If average value is 55% - no action is taken
- If average value is 65% - 1 instance is launched. Any additional alarms that fall in the same step is ignored for 300 seconds
- If during warmup period, a new alarm is received with average value of 75% - 2 additional instances are launched (for a total of 3 instances)

Instance Termination Policy

- Scale-in event requires instance termination
- Exclude instances that have scale-in protection enabled
- Default Termination Policy - Flow
    - Select instance from Availability Zone that has most instances
    - Select instance with oldest launch configuration
    - Select instance close to next billing hour
    - Select instance at random

Instance Termination Policy

- Custom Termination Policies
  - OldestInstance
  - NewestInstance
  - OldestLaunchConfiguration
  - ClosestToNextInstanceHour
  - Default

Instance Protection

- Protect instances from termination by enabling Instance Protection
- Specify at auto scaling group level or individual instance level
- Unhealthy instances are terminated and replaced irrespective of instance protection setting
- Does not apply to spot instances

Suspending Auto Scaling Process

- You can suspend any auto scaling processes
- Useful for troubleshooting configuration issues
- List of auto scaling processes

Monitoring

Auto Scaling - CloudWatch Metrics

Need to enable it manually

Aggregates metrics at auto scale group level