Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Course Introduction

Welcome to the AWS Certified DevOps Engineer - Professional Level course.

This course is designed to provide an experience DevOps Engineer with the resources to prepare for and pass the AWS Certified DevOps Engineer Professional level certification exam.
Topics in this course include:

1. Auto Scaling Deployment Concepts
How to configure Auto Scaling and deploy applications using Auto Scaling.

2. Deployment Concepts With EC2
Using EC2 in deployments including EC2 Logging and Backup Strategies.

3. CloudWatch For DevOps
Utilizing CloudWatch for DevOps including Monitoring and Logging.

4. CloudFormation For DevOps
Utilizing CloudFormation to deploy infrastructure on AWS.

5. Elastic Beanstalk For Devops
Using Elastic Beanstalk to quickly deploy applications on AWS.

6. Application Deployments With OpsWorks
Using OpsWorks to create and manage resources for our applications, as well as managing the applications themselves.

7. DynamoDB Concepts
Reviewing the basics of DynamoDB as well as understanding the role of DynamoDB and its use cases in Devops.

8. Data Pipeline With DynamoDB
We will use AWS Data Pipeline to retrieve data from a tab-delimited file in Amazon S3 to populate a DynamoDB table, use a Hive script to define the necessary data transformation steps, and automatically create an Amazon EMR cluster to perform the work.

9. S3 Concepts For DevOps
Utilizing S3 Buckets as a key repository for DevOps via Cross-Account Access.

10. A/B Testing
A/B Testing concepts for DevOps.

11. Blue/Green Deployments
Using Automation For Devops to Facilitate Blue/Green Deployments.

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
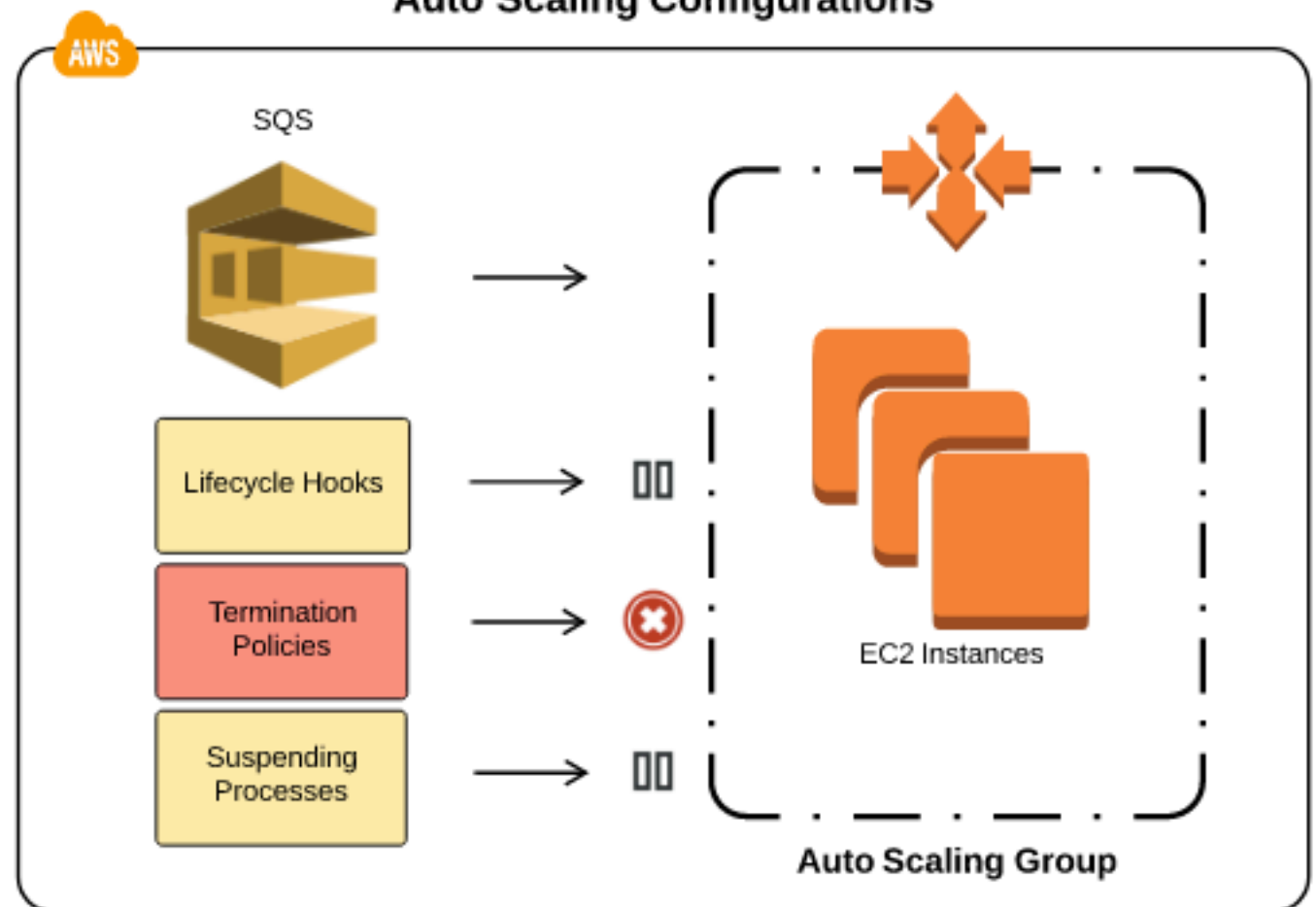- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

Scenario:

A DevOps Engineer will spend a great deal of time creating, monitoring, and tuning Auto Scaling Groups. They can be used in a variety of different ways and understanding various use cases and proper configuration is a key skill set. In this section the student will revisit the basics of Auto Scaling, Termination Policies, Suspending Auto Scaling processes, and take a deep dive into Lifecycle Hooks. Hooks are a powerful tool in controlling instance and application deployment.

- Deployment Concepts
- Termination Policies
- Suspending Processes
- Lifecycle Hooks
- ASG API, CLI, SDK
- SQS With Auto Scaling

Auto Scaling Configurations

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
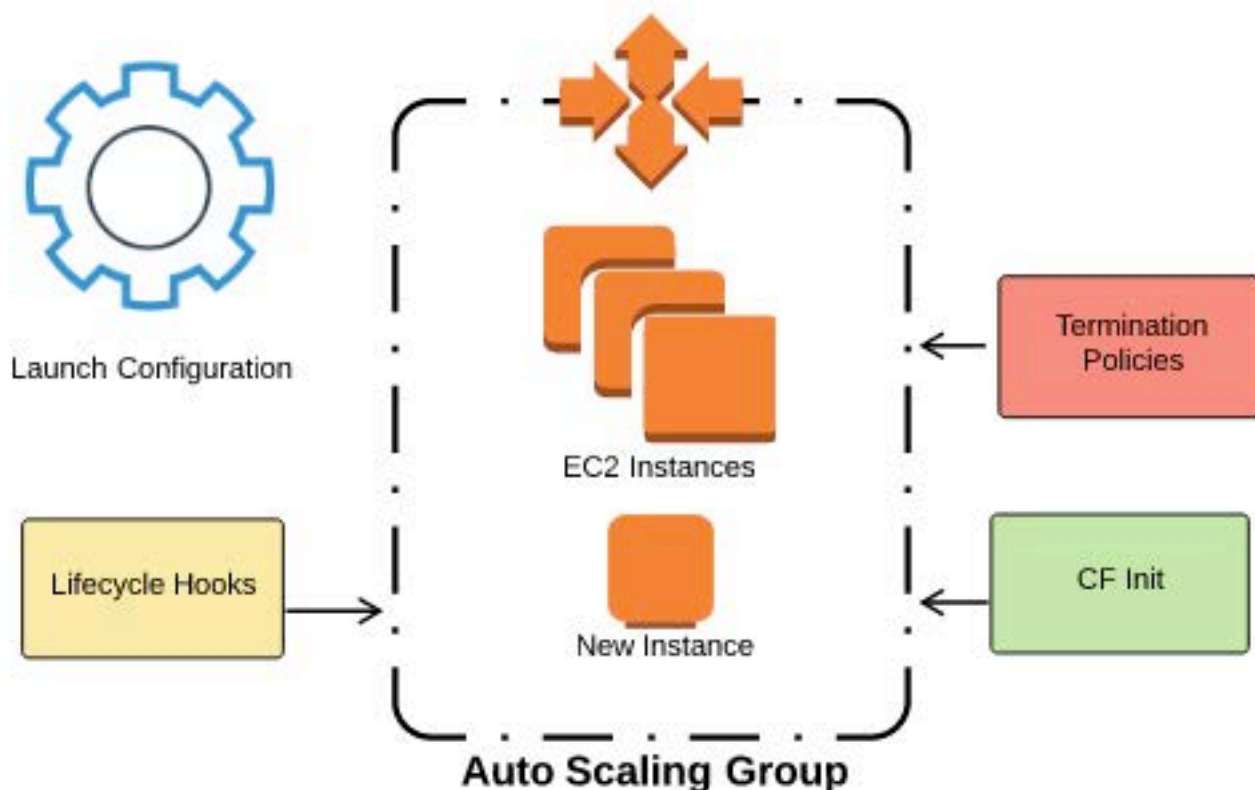- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

Scenario Solver X

AWS

Deployment Concepts With Auto Scaling

Launch Configuration

EC2 Instances

Termination Policies

Lifecycle Hooks

New Instance

CF Init

Auto Scaling Group

Deployment Concepts

- Auto Scaling can play a major role in deployments
- Avoid downtime during deployments
- How long to deploy code and configure an instance?
- How do you test new launch configurations?
- How do you phase out older launch configurations?
- Use Lifecycle Hooks for custom actions
- CloudFormation init scripts
- Cloud init scripts
- Scale Out/Scale In

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
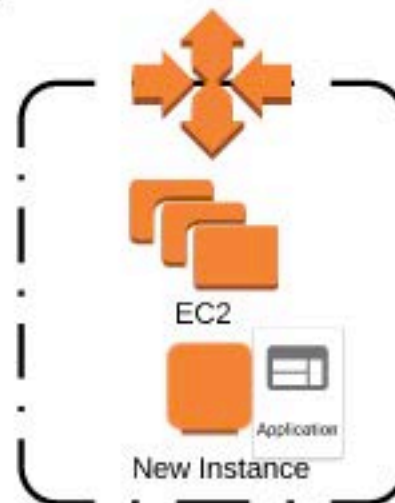- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

X

AWS

Scenario: Bootstrapping instances in ASG takes 10 minutes. Instances are reported as in-service before bootstrapping completes and you are getting application alarms.

Solution

EC2

Application

New Instance

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

X

Lifecycle Hooks → pending: wait

Solution: Create an ASG Lifecycle Hook to hold the instance in a pending:wait state until bootstrapping is complete. Move to pending:complete when bootstrapping is finished.

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

Default Termination Policy   X

Auto Scaling Termination Policies

AWS

Launch Configuration

Termination Policies →
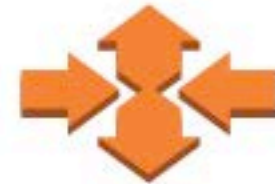
EC2 Instances

Which instance to terminate?

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.
- **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.

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver

Deployment Pipelines

API Gateway

Lambda

Secrets Manager

Auto Scaling Deployment Concepts
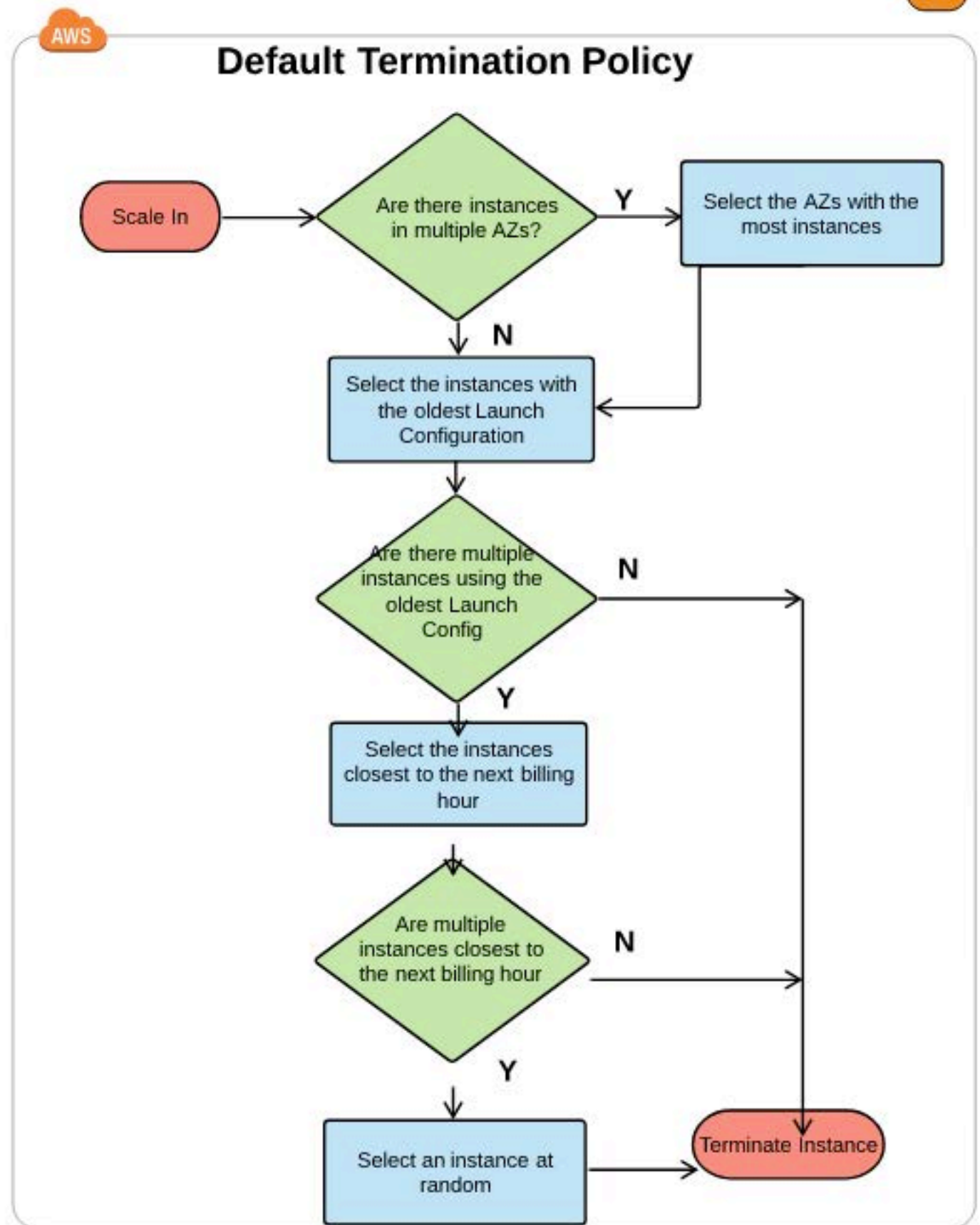
AWS

Default Termination Policy

Scale In → Are there instances in multiple AZs?

Y → Select the AZs with the most instances

N → Select the instances with the oldest Launch Configuration

Are there multiple instances using the oldest Launch Config

N → (to Terminate Instance)

Y → Select the instances closest to the next billing hour

Are multiple instances closest to the next billing hour

N → (to Terminate Instance)

Y → Select an instance at random → Terminate Instance

Linux Academy | **Cloud Assessments**

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

X

AWS
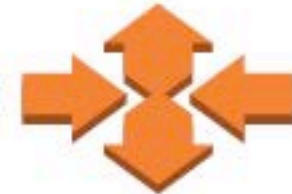
Suspending Auto Scaling Processes

Launch Configuration

Suspend Auto Scaling Processes →

EC2 Instances

Suspend Processes

Auto Scaling Group

Suspending Auto Scaling Processes

Auto Scaling Processes

- Launch - If you suspend Launch, this disrupts other processes.
- Terminate - If you suspend Terminate, this disrupts other processes.
- Health Check
- ReplaceUnhealthy
- AZRebalance
- AlarmNotification
- ScheduledActions
- AddToLoadBalancer

Auto Scaling Deployment Concepts

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Lifecycle Hooks Transitions   X

AWS Auto Scaling Lifecycle Hooks

AWS

Launch Configuration

Lifecycle Hook →

EC2 Instances

New Instance

Auto Scaling Group

Creating AutoScaling Lifecycle Hooks

Steps:
1) Create Launch Configuration
2) Create Auto Scaling Group
3) Create Lifecycle Hook
4) Launch an EC2 instance
5) Monitor the Lifecycle Status of the instance.
6) Verify successful installation of software on the instance.

Choose a Topic

Introduction

Auto Scaling Deployment Concepts

Deployment Concepts With EC2

CloudWatch For DevOps

CloudFormation For DevOps

Elastic Beanstalk For DevOps

Application Deployments With OpsWorks

DynamoDB Concepts

S3 Concepts For DevOps

Blue/Green Deployments

Scenario Solver
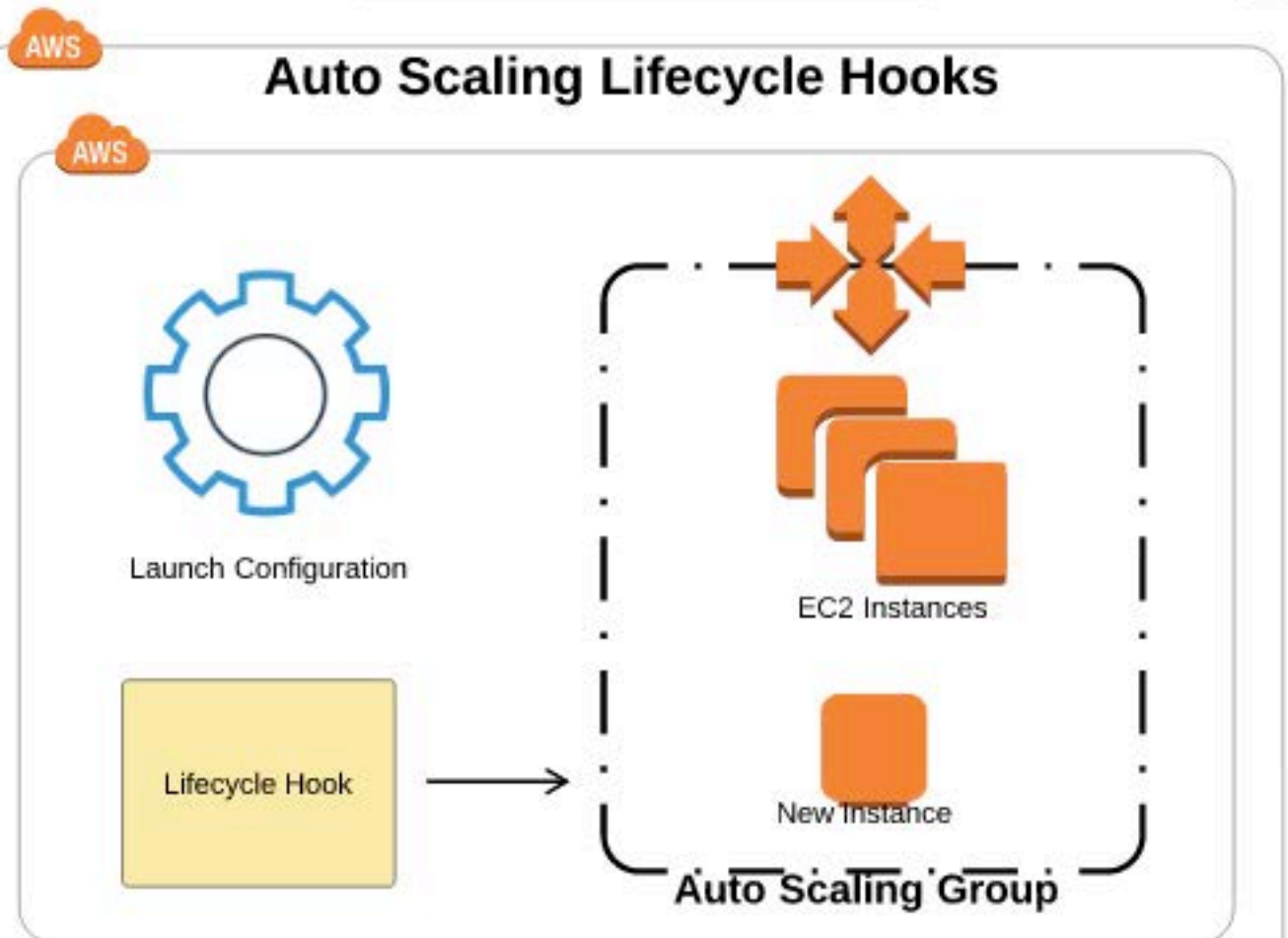
Deployment Pipelines

API Gateway

Lambda

Secrets Manager

Auto Scaling Deployment Concepts

X

AWS

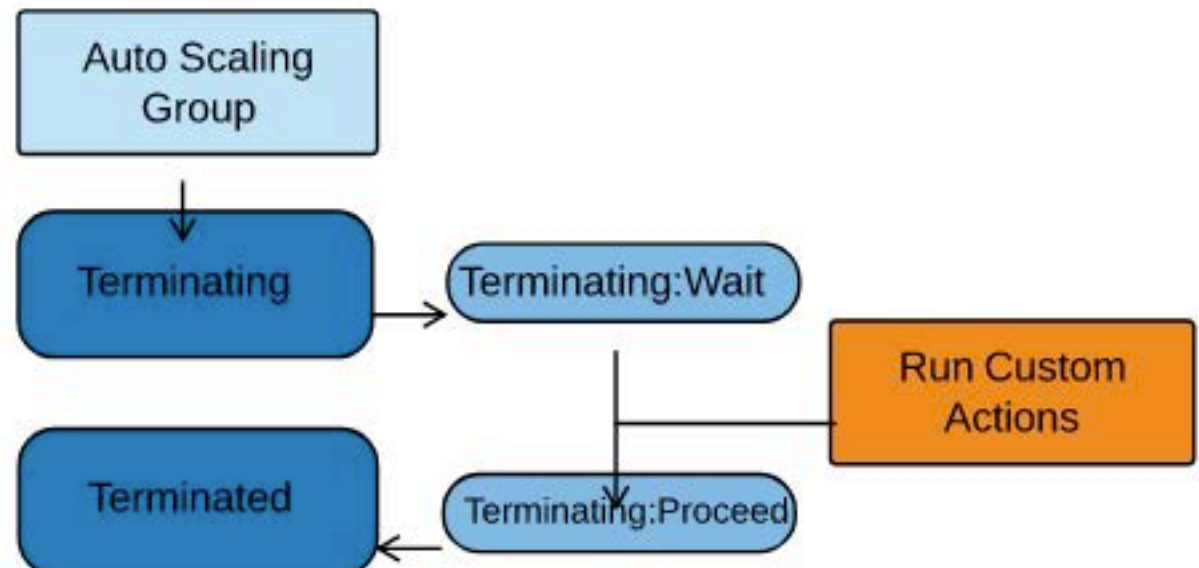## Transitions Between Instance States
Scale Out
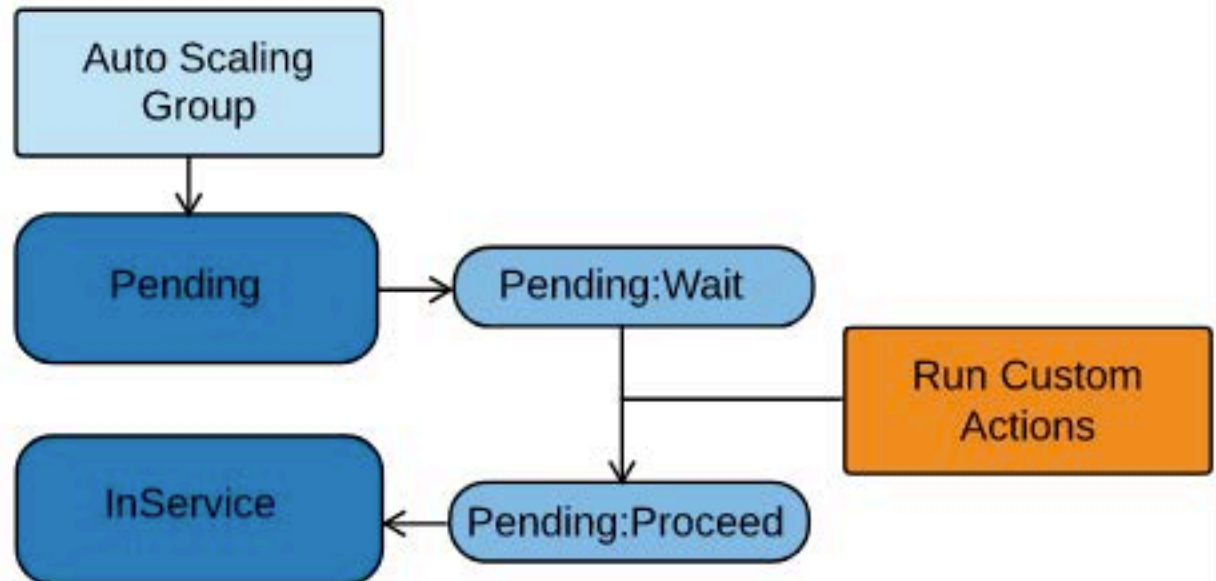
Auto Scaling Group

→

Pending → Pending:Wait

Run Custom Actions

InService ← Pending:Proceed

Scale In

Auto Scaling Group

→

Terminating → Terminating:Wait

Run Custom Actions

Terminated ← Terminating:Proceed

Linux Academy | Cloud Assessments

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

X

AWS

Auto Scaling API, CLI, and SDK Calls

Launch Configuration

AWS CLI

AWS API

EC2 Instances

Auto Scaling API, CLI, and SDK Calls

- Common Auto Scaling API Calls.
- When using the CLI or different SDKs the method names might be different than the API call names.
- Find the API Auto Scaling Guide.
- Look at common API calls.
- Compare common API calls to the CLI or Restful name.
- Script calls for automation.
- CLI Name: put-life-cycle-hook vs API Name: PutLifecycleHook

Choose a Topic

- Introduction
- Auto Scaling Deployment Concepts
- Deployment Concepts With EC2
- CloudWatch For DevOps
- CloudFormation For DevOps
- Elastic Beanstalk For DevOps
- Application Deployments With OpsWorks
- DynamoDB Concepts
- S3 Concepts For DevOps
- Blue/Green Deployments
- Scenario Solver
- Deployment Pipelines
- API Gateway
- Lambda
- Secrets Manager

Auto Scaling Deployment Concepts

X

Auto Scaling API, CLI, and SDK Calls

AWS

Launch Configuration

AWS CLI →

AWS API →

EC2 Instances

Auto Scaling API, CLI, and SDK Calls

- Common Auto Scaling API Calls.
- When using the CLI or different SDKs the method names might be different than the API call names.
- Find the API Auto Scaling Guide.
- Look at common API calls.
- Compare common API calls to the CLI or Restful name.
- Script calls for automation.
- CLI Name: put-life-cycle-hook vs API Name: PutLifecycleHook

Auto Scaling Deployment Concepts

X

Using Auto Scaling to Process SQS Messages

SQS

Web Server

EC2 Instances
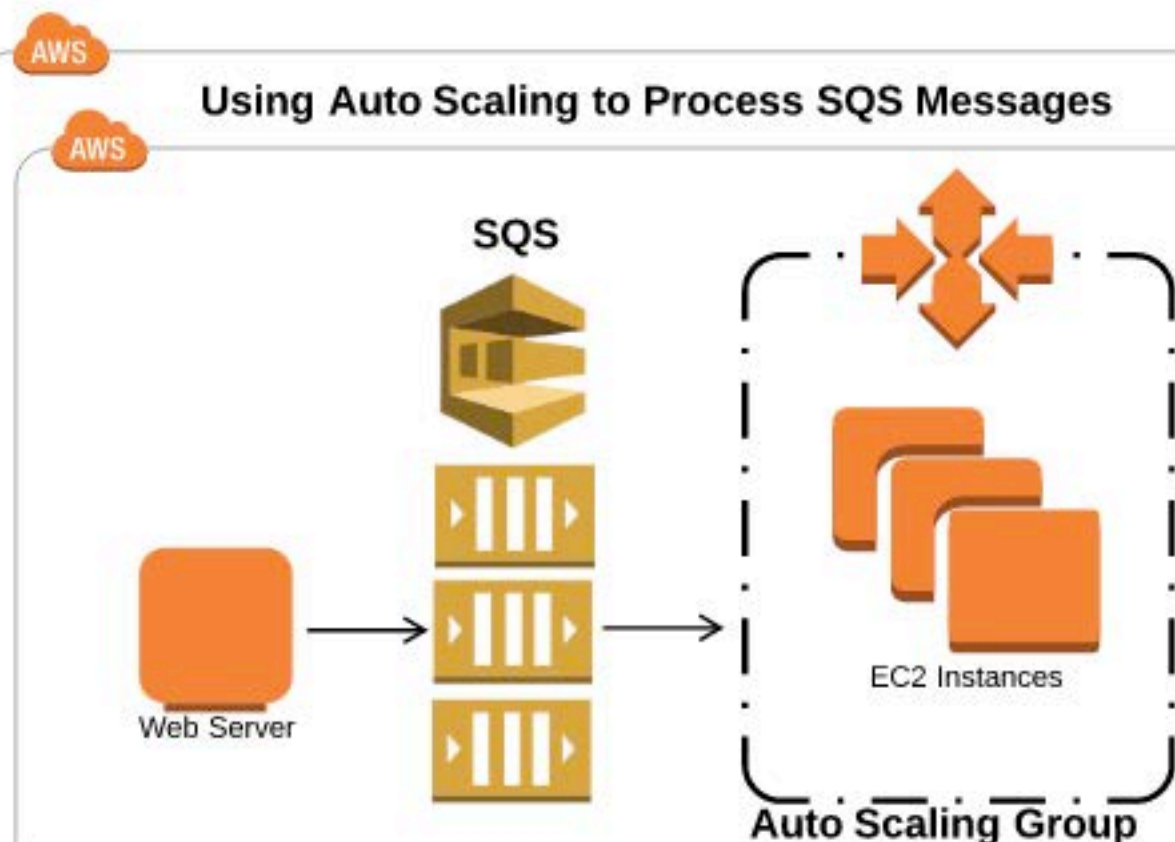
Auto Scaling Group

Using Auto Scaling to Process SQS Messages

Scenario:

- Web App receives orders from customers.
- The app places the orders in an Amazon SQS queue until they are picked up for processing.
- Orders are processed.
- Processed orders sent back to the customer.
- Scale up/down based on request queue size.

Steps To Configure And Test:

1) Create the scaling policies
2) Create CloudWatch Alarms
3) Verify Your Scaling Policies and CloudWatch Alarms
4) Test Your Scale Out and Scale In Policies