Creating Auto Scaling Groups with AWS CLI

What is Amazon Auto Scaling Group?

"AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost" . Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

It's easy to get started with AWS Auto Scaling using the AWS Management Console, Command Line Interface (CLI), or SDK. AWS Auto Scaling is available at no additional charge. You pay only for the AWS resources needed to run your applications and Amazon CloudWatch monitoring fees.

Step 1

- Initially, log in to AWS CLI and configure Access key, Secret Access Key and Default region (Availability Zone).

```
[ec2-user@ip-172-31-16-234 ~]$ aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key                <not set>             None    None
secret_key                <not set>             None    None
    region                ap-south-2             imds
```

```
[ec2-user@ip-172-31-16-234 ~]$ aws configure
AWS Access Key ID [None]: AKIA53DW6FV2EGFOG6BX
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-2
Default output format [None]:
```

- Check if the configured settings were applied or not

```
[ec2-user@ip-172-31-16-234 ~]$ aws configure
AWS Access Key ID [****************G6BX]:
AWS Secret Access Key [****************UhxD]:
Default region name [ap-south-2]:
Default output format [None]:
```

Step 2

- Create Launch Template using AWS CLI. To create a basic launch template, use the create-launch-template command as follows:

```
[ec2-user@ip-172-31-16-234 ~]$ aws ec2 create-launch-template --launch-template-name
my-template-for-auto-scaling --version-description version1     --launch-template-data
'{"ImageId":"ami-0ec8bff925443089f","InstanceType":"t3.micro"}'
```

```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-05049af3c25b52232",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "CreateTime": "2023-11-27T09:32:58+00:00",
        "CreatedBy": "arn:aws:iam::951584697716:user/siva",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

- This example creates a launch template with the name my-template-for-auto-scaling. If the instances created by this launch template are launched in a default VPC, they receive a public IP address by default.

- Check Launch Templates page in AWS Console, a new launch template will be created there.

Step 3

- This example creates an Auto Scaling group that launches a single instance using a launch template to optionally specify the ID of an existing network interface (ENI ID) to use. It specifies an Availability Zone that matches the specified network interface.

```
[ec2-user@ip-172-31-16-234 ~]$ aws autoscaling create-auto-scaling-group --auto-scaling-group-name
my-asg-launch-template --launch-template "LaunchTemplateName=my-template-for-auto-scaling,Version=1"
--min-size 1 --max-size 1 --availability-zones ap-south-2b
```

- Check the Auto Scaling Groups list in AWS Console, ASG named "my-asg-launch-template" will be created.

- **Commands Explanation**

- –auto-scaling-group-name (string)

The name of the Auto Scaling group. This name must be unique per Region per account.

- LaunchTemplateName -> (string)

The name of the launch template. To get the template name, use the Amazon EC2 DescribeLaunchTemplates API operation. New launch templates can be created using the Amazon EC2 CreateLaunchTemplate API. You must specify either a template ID or a template name.

- Version -> (string)

The version number, $Latest , or $Default . To get the version number, use the Amazon EC2 DescribeLaunchTemplateVersions API operation. New launch template versions can be created using the Amazon EC2 CreateLaunchTemplateVersion API.

- –min-size (integer)

The minimum size of the group.

- –max-size (integer)

The maximum size of the group.

- –availability-zones (list)

One or more Availability Zones for the group. This parameter is optional if you specify one or more subnets for VPCZoneIdentifier.