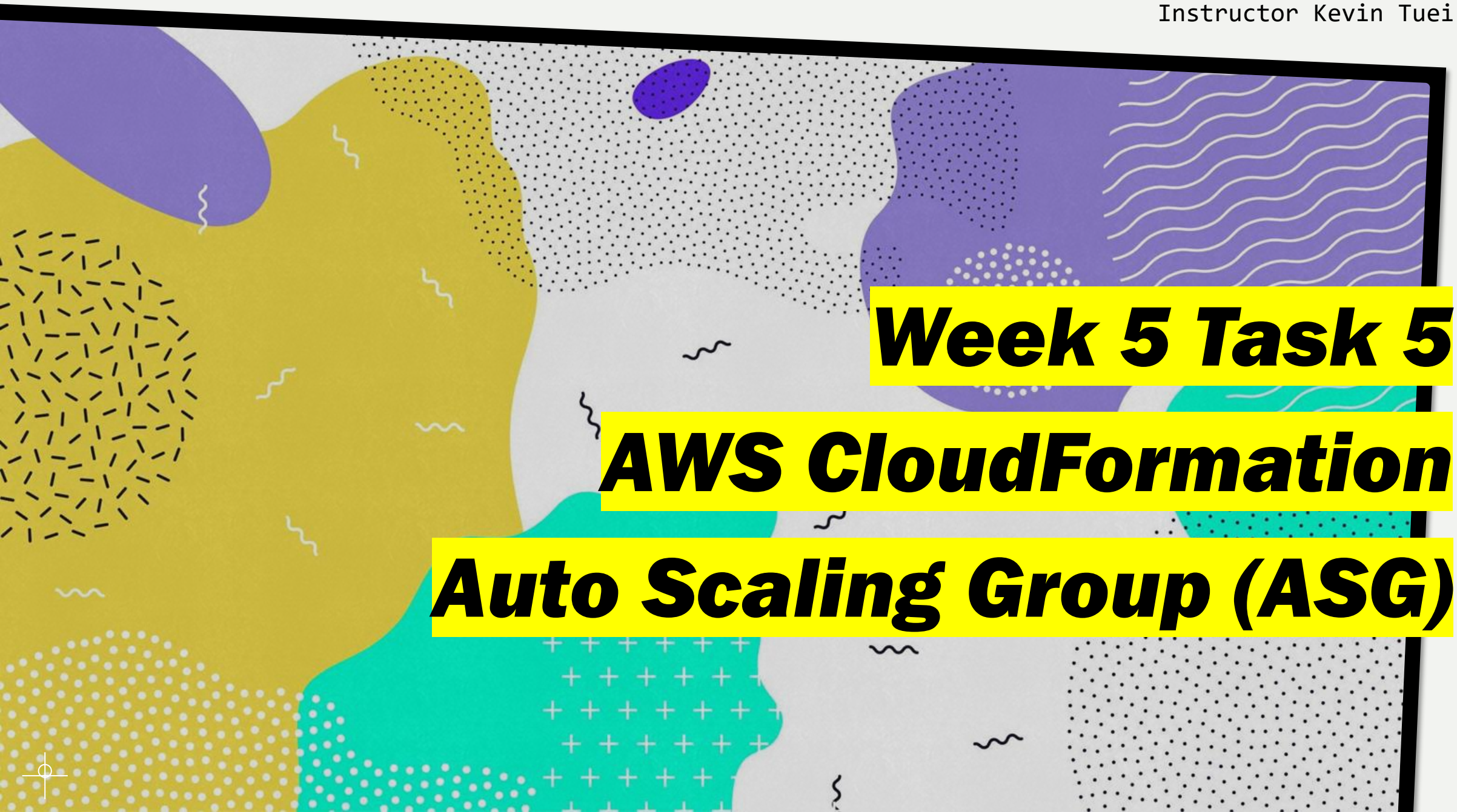
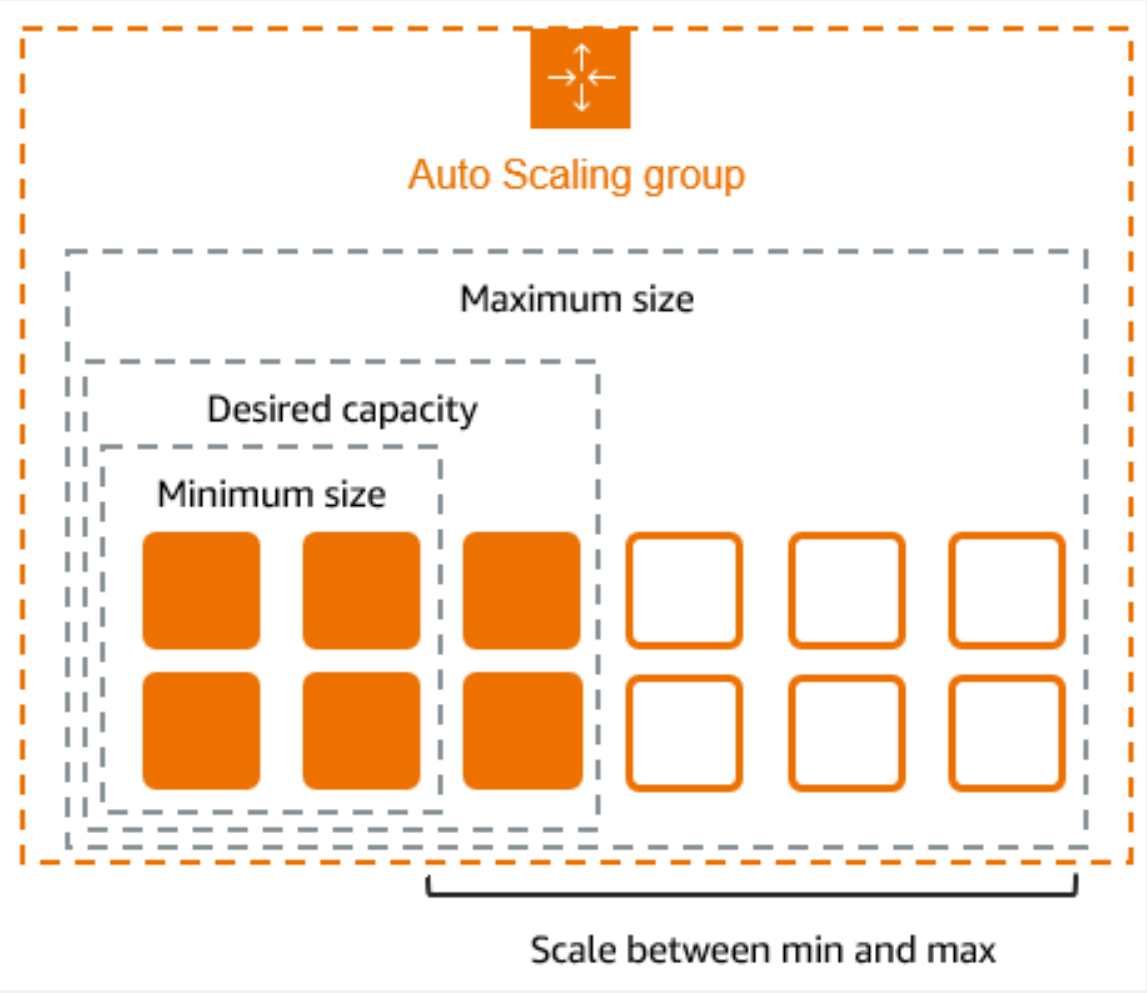
Instructor Kevin Tuei



What is an Auto-Scaling Group?

- An Auto-Scaling Group (ASG) is a service provided by AWS that automatically adjusts the number of EC2 instances in your application based on predefined conditions.
- It ensures that the desired number of instances is always running, handles scaling up and down according to demand, and replaces instances if they become unhealthy.



What are the benefits of having an ASG?

- **Automatic Scaling:** Adjusts the number of instances based on the current demand, ensuring application availability and performance.
- **Fault Tolerance:** Automatically replaces instances that become unhealthy or fail.
- **Cost Efficiency:** Scales in during low demand to save on costs, and scales out during high demand to maintain performance.
- **Integration with Load Balancers:** Distributes incoming traffic across instances to balance the load and improve performance.

What is required when defining an ASG?

- **Launch Configuration or Launch Template:** Defines the configuration for instances that will be launched by the ASG.
- **Desired Capacity, Minimum Size, Maximum Size:** Specifies the number of instances to maintain, the minimum and maximum bounds for scaling.
- **VPC Subnets:** The subnets where the instances will be launched.
- **Scaling Policies:** Define how and when the ASG should scale in or out based on metrics or schedules.

Template Snippet for ASG

```
AutoScalingGroup:
  Type: 'AWS::AutoScaling::AutoScalingGroup'
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref LaunchTemplate
      Version: '1'
    MinSize: 1
    MaxSize: 5
    DesiredCapacity: 2
    VPCZoneIdentifier:
      - !Ref PublicSubnet1
      - !Ref PublicSubnet2
    Tags:
      - Key: Name
        Value: Skillsync-ASG
        PropagateAtLaunch: true
    TargetGroupARNs:
      - !Ref ALBTargetGroup
    HealthCheckType: EC2
    HealthCheckGracePeriod: 300
```

What are the Best Practices for ASGs?

- **Define Proper Scaling Policies:** Set up policies based on metrics like CPU utilization or custom metrics to ensure that scaling actions align with application needs.
- **Use Multiple Availability Zones:** Deploy instances across multiple Availability Zones to enhance fault tolerance.
- **Monitor ASG and Adjust Scaling Policies:** Continuously monitor ASG performance and adjust scaling policies as needed based on application usage patterns.

What are the Best Practices for ASGs?

- **Implement Health Checks:** Configure health checks to ensure that only healthy instances are included in the ASG.
- **Tag Resources:** Use tags to identify and manage ASG resources easily.

Template Snippet for ScaleOutPolicy

```
ScaleOutPolicy:
  Type: 'AWS::AutoScaling::ScalingPolicy'
  Properties:
    AutoScalingGroupName: !Ref
AutoScalingGroup
    PolicyType: TargetTrackingScaling
    TargetTrackingConfiguration:
      TargetValue: 50.0 # Adjust
accordingly
      PredefinedMetricSpecification:
        PredefinedMetricType:
ASGAverageCPUUtilization
```

Template Snippet for ScaleInPolicy

```
ScaleInPolicy:
  Type: 'AWS::AutoScaling::ScalingPolicy'
  Properties:
    AutoScalingGroupName: !Ref AutoScalingGroup
    PolicyType: TargetTrackingScaling
    TargetTrackingConfiguration:
      TargetValue: 20.0 # Adjust accordingly
      PredefinedMetricSpecification:
        PredefinedMetricType:
ASGAverageCPUUtilization
```


Hands-On Demonstration