Managing EC2 Instances and Auto Scaling

Presented By Mohd Shahid

Introduction

- Managing EC2 ensures high availability, fault tolerance, and cost optimization.
- Auto Scaling adjusts instance count automatically to match demand.

Why Manage EC2 Instances?

- ✅ Cost Optimization
- ✅ Performance Management
- ✅ Security Maintenance
- ✅ Resource Utilization
- ✅ Backup & Recovery

Why Auto Scaling?

- 🔄 High Availability
- 💰 Cost Efficiency
- ⚡ Improved Performance
- 🛡️ Fault Tolerance

Types of Auto Scaling

- Dynamic Scaling
  - Target Tracking
  - Step Scaling
  - Simple Scaling
- Scheduled Scaling
- Predictive Scaling (ML-powered)

Auto Scaling Approaches

- **Horizontal Scaling**
  - Adds/removes EC2 instances
  - Ideal for microservices, web apps
- **Vertical Scaling**
  - Changes instance type (e.g. t2.micro → t3.medium)
  - Ideal for high CPU/memory needs

Horizontal vs Vertical Scaling

Feature	Horizontal Scaling	Vertical Scaling
Definition	Adds more instances to distribute the load.	Increases the capacity of an existing instance.
Scalability	Infinite scaling potential by adding instances.	Limited by the maximum instance capacity.
Performance	Better for handling high traffic loads.	Suitable for applications needing higher compute power per instance.
Cost	Can be cost-efficient as smaller instances can be used.	May become expensive as larger instances are required.
Complexity	Requires load balancers and distributed systems.	Easier to implement but has hardware limitations.
Use Cases	Web applications, microservices, and cloud-native apps.	Databases and applications needing high memory or CPU.

Auto Scaling Use Cases

- Web Applications
- Batch Processing
- Data Analytics
- Microservices Architectures

Key Auto Scaling Components

- Launch Template/Configuration
- Auto Scaling Group (ASG)
- Scaling Policies:
  - Dynamic
  - Scheduled
  - Predictive

Auto Scaling Parameters

- Desired Capacity
- Min/Max Size
- Scaling Policies

EC2 Management (GUI)

1. Open EC2 Dashboard
2. Start/Stop/Reboot an instance
3. Resize instance
4. Monitor metrics (CPU, Disk I/O)

EC2 Management (CLI)

- Start: aws ec2 start-instances
- Stop: aws ec2 stop-instances
- Resize:
  - Stop → Modify Type → Start
- Monitor: aws ec2 describe-instance-status

Auto Scaling Setup (GUI)

1. Create Launch Config/Template
2. Create Auto Scaling Group
3. Configure Scaling Policies
4. Monitor with CloudWatch
5. Delete when done

Auto Scaling Setup (CLI)

- Create Launch Config
- Create ASG
- Apply Target Tracking Policy
- Monitor with CloudWatch
- Delete ASG: --force-delete

Benefits of Auto Scaling

- ✅ Enhanced Availability
- ✅ Dynamic Scalability
- ✅ Cost Efficiency
- ✅ Reliability

Real-World Examples

- E-commerce Sales Events
- Media Streaming
- Big Data Processing
- Microservices Scaling

Best Practices

- Set realistic thresholds
- Use multiple AZs
- Enable monitoring
- Optimize scaling policies
- Integrate with CI/CD

Hands-On Demo Summary

- Launch EC2 & install Apache
- Create launch template
- Configure Auto Scaling Group
- Simulate load with stress command
- Observe auto-scaling
- Clean up resources