AWS Certified Machine Learning Engineer - Associate Exam Guide

# Section 3.2 — Create and Script Infrastructure (In-Depth)

This section focuses on creating, scripting, and managing ML infrastructure efficiently. It covers infrastructure as code, scaling strategies, containerization, and AWS services for automation.

## 1. Provisioning Resources

• On-Demand instances → flexible, pay-as-you-go.

• Reserved Instances → long-term predictable workloads with cost savings.

• Spot Instances → cost savings for fault-tolerant jobs (training, batch).

• Provisioned throughput → guarantees capacity for critical workloads.

## 2. Scaling Strategies

• Auto Scaling → automatically adjusts instance count based on demand.

• Vertical scaling → change instance size (CPU/GPU/memory optimized).

• Horizontal scaling → increase number of instances for throughput.

• Scaling policies → latency-based, CPU utilization, invocations per instance.

## 3. Infrastructure as Code (IaC)

• AWS CloudFormation → declarative templates to define infrastructure.

• AWS CDK (Cloud Development Kit) → infrastructure provisioning in Python/TypeScript/Java.

• Benefits: reproducibility, versioning, automation of infrastructure deployments.

## 4. Containerization

• Amazon ECR → container image storage.

• Amazon ECS → container orchestration service (good for batch inference).

• Amazon EKS → managed Kubernetes for scalable ML workloads.

• SageMaker BYOC (Bring Your Own Container) → custom ML environments with dependencies.

## 5. SageMaker Deployment Options

• SageMaker Hosting → managed endpoints for real-time/batch inference.

• SageMaker Serverless Inference → automatic scaling, pay-per-request.

• SageMaker Multi-Model Endpoints → cost-effective hosting for many models.

• SageMaker Async Inference → process long-running jobs asynchronously.

## 6. Best Practices

• Automate provisioning with IaC tools (CloudFormation, CDK).

• Use Spot Instances for training jobs to save cost.

• Configure endpoints inside VPCs for security.

• Select scaling metrics carefully (latency, CPU, request rate).

• Use monitoring (CloudWatch, CloudTrail) for infrastructure reliability.

## ✅ Exam Tips

• CloudFormation = declarative IaC, CDK = programmatic IaC.

• Spot Instances = best for cost savings on training.

• Multi-Model Endpoints = cost optimization strategy.

• EKS/ECS for containerized inference workloads, BYOC for custom environments.

• Always deploy inside VPC for security in production.