AWS ML Associate — Section 4.3: Optimize ML Workloads

This section focuses on optimizing ML workloads for cost efficiency and performance. It includes best practices for compute, storage, and debugging.

# Optimization Techniques

• Use spot instances for cost-effective training; always checkpoint models to handle interruptions

• Right-size instance types — choose GPU for deep learning, CPU for lightweight models, memory-optimized for large datasets

• SageMaker Debugger — monitor training jobs, detect bottlenecks, and profile system usage

• Distributed training — use when datasets/models exceed single-instance limits

# Best Practices

• Enable automatic scaling for inference endpoints to handle variable traffic

• Use managed storage options (S3 lifecycle policies) to control data costs

• Profile workloads regularly and eliminate unused resources

# Exam Tips

• Spot instances reduce cost significantly but require checkpointing

• Debugger is key for identifying training bottlenecks and improving efficiency

• Always choose the right instance type for the workload to balance cost vs. performance