AI-300 · CHEAT SHEET · CERTIFYDECK

Microsoft Certified: Machine Learning Operations Engineer Associate

Dominios, pesos oficiales y habilidades clave del temario de AI-300, para repasar antes del examen.

Nivel: Intermediate Duración: 120 min Nota de corte: 700/1000 Fuente: guía de estudio oficial de Microsoft Learn

1. Design and implement an MLOps infrastructure

15–20%

Create and manage resources in a Machine Learning workspace

- Create and manage a workspace
- Create and manage datastores
- Create and manage compute targets
- Configure identity and access management for workspaces

Create and manage assets in a Machine Learning workspace

- Create and manage data assets
- Create and manage environments
- Create and manage components
- Share assets across workspaces by using registries

Implement IaC for Machine Learning

- Configure GitHub integration with Machine Learning to enable secure access
- Deploy Machine Learning workspaces and resources by using Bicep and Azure CLI
- Automate resource provisioning by using GitHub Actions workflows
- Restrict network access to Machine Learning workspaces
- Manage source control for machine learning projects by using Git

2. Implement machine learning model lifecycle and operations

25–30%

Orchestrate model training

- Configure experiment tracking with MLflow
- Use automated machine learning to explore optimal models
- Use notebooks for experimentation and exploration
- Automate hyperparameter tuning
- Run model training scripts

Implement model registration and versioning

- Package a feature retrieval specification with the model artifact
- Register an MLflow model
- Evaluate a model by using responsible AI principles
- Manage model lifecycle, including archiving models

Deploy machine learning models for production environments

- Deploy models as real-time or batch endpoints with managed inference options
- Test and troubleshoot model endpoints
- Implement progressive rollout and safe rollback strategies

Monitor and maintain machine learning models in production

- Detect and analyze data drift
- Monitor performance metrics of models deployed to production
- Configure retraining or alert triggers when thresholds are exceeded

3. Design and implement a GenAIOps infrastructure

20–25%

Implement Foundry environments and platform configuration

- Create and configure Foundry resources and project environments
- Configure identity and access management with managed identities and role-based access control
- Implement network security and private networking configurations
- Deploy infrastructure using Bicep templates and Azure CLI

Deploy and manage foundation models for production workloads

- Deploy foundation models by using serverless API endpoints and managed compute options
- Select appropriate models for specific use cases
- Implement model versioning and production deployment strategies
- Configure provisioned throughput units for high-volume workloads

Implement prompt versioning and management with source control

- Design and develop prompts
- Create prompt variants and compare performance across different prompts
- Implement version control for prompts by using Git repositories

4. Implement generative AI quality assurance and observability

10–15%

Configure evaluation and validation for generative AI applications and agents

- Create test datasets and data mapping for comprehensive model evaluation
- Implement AI quality metrics including groundedness, relevance, coherence, and fluency
- Configure risk and safety evaluations for harmful content detection
- Set up automated evaluation workflows by using built-in and custom evaluation metrics

Implement observability for generative AI applications and agents

- Examine continuous monitoring in Foundry
- Monitor latency, throughput, and response times
- Track and optimize token consumption and resource usage
- Configure detailed logging, tracing, and debugging capabilities

5. Optimize generative AI systems and model performance

10–15%

Optimize retrieval-augmented generation performance and accuracy

- Tune similarity thresholds, chunk sizes, and retrieval strategies
- Select and fine-tune embedding models for domain-specific use cases
- Implement hybrid search combining semantic and keyword-based retrieval
- Evaluate and improve RAG performance by using relevance metrics and A/B testing

Implement advanced fine-tuning and model customization

- Design and implement advanced fine-tuning methods
- Create and manage synthetic data for fine-tuning
- Monitor and optimize fine-tuned model performance
- Manage a fine-tuned model from development through production deployment