

AI-300T00: OPERATIONALIZE MACHINE LEARNING AND GENERATIVE AI SOLUTIONS



DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
4 Days	Intermediate	Azure	Instructor-led / VILT	N/A

INTRODUCTION

This course prepares learners to design, implement, and operate Machine Learning Operations (MLOps) and Generative AI Operations (GenAIOps) solutions on Azure. It covers building secure and scalable AI infrastructure, managing the full lifecycle of traditional machine learning models with Azure Machine Learning, and deploying, evaluating, monitoring, and optimizing generative AI applications and agents using Microsoft Foundry. Learners will gain hands-on knowledge of automation, continuous integration and delivery, infrastructure as code, and observability by using tools such as GitHub Actions, Azure CLI, and Bicep. The course emphasizes collaboration with data science and DevOps teams to deliver reliable, production-ready AI systems aligned with modern MLOps and GenAIOps best practices.

AUDIENCE PROFILE

This course is intended for data scientists, machine learning engineers, and DevOps professionals who want to design and operate production-grade AI solutions on Azure. It is suited for learners with experience in Python, a foundational understanding of machine learning concepts, and basic familiarity with DevOps practices such as source control, CI/CD, and command-line tools, who are preparing to implement MLOps and GenAIOps workflows using Azure-native services.

PREREQUISITES

- Programming experience with Python or R
- Experience developing and training machine learning models
- Familiarity with basic Azure Machine Learning concepts

COURSE OBJECTIVES

After completing this course, students will be able to:

- Operationalize machine learning models (MLOps)
- Operationalize generative AI applications (GenAIOps)

COURSE CONTENT

Module 1: Experiment with Azure Machine Learning

Learn how to find the best machine learning model with automated machine learning (AutoML), MLflow-tracked notebooks, and the Responsible AI dashboard.

Lessons:

- Introduction
- Preprocess data and configure featurization
- Run an automated machine learning experiment
- Evaluate and compare models

- Configure MLflow for model tracking in notebooks
- Train and track models in notebooks
- Evaluate models with the Responsible AI dashboard
- Exercise - Find the best classification model with Azure Machine Learning
- Module assessment
- Summary

After completing this module, students will be able to:

- Prepare your data to use AutoML for classification.

- Configure and run an AutoML experiment.
- Evaluate and compare AutoML models.
- Configure MLflow for model tracking in notebooks.
- Use MLflow for model tracking in notebooks.
- Evaluate a trained model using the Responsible AI dashboard.

Module 2: Perform hyperparameter tuning with Azure Machine Learning

Learn how to perform hyperparameter tuning with a sweep job in Azure Machine Learning.

Lessons:

- Introduction
- Define a search space
- Configure a sampling method
- Configure early termination
- Use a sweep job for hyperparameter tuning
- Exercise - Run a sweep job
- Module assessment
- Summary

After completing this module, students will be able to:

- Define a hyperparameter search space.
- Configure hyperparameter sampling.
- Select an early-termination policy.
- Run a sweep job.

Module 3: Run pipelines in Azure Machine Learning

Learn how to create and use components to build pipeline in Azure Machine Learning. Run and schedule Azure Machine Learning pipelines to automate machine learning workflows.

Lessons:

- Introduction
- Create components
- Create a pipeline
- Run a pipeline job
- Exercise - Run a pipeline job
- Module assessment
- Summary

After completing this module, students will be able to:

- Create components.
- Build an Azure Machine Learning pipeline.
- Run an Azure Machine Learning pipeline.

Module 4: Trigger Azure Machine Learning jobs with GitHub Actions

Learn how to automate your machine learning workflows by using GitHub Actions.

Lessons:

- Introduction
- Understand the business problem
- Explore the solution architecture
- Use GitHub Actions for model training
- Exercise
- Module assessment
- Summary

After completing this module, students will be able to:

- Create and assign a service principal the permissions needed to run an Azure Machine Learning job.

- Store Azure credentials securely using secrets in GitHub Secrets.
- Create a GitHub Action using YAML that uses the stored Azure credentials to run an Azure Machine Learning job.

Module 5: Trigger GitHub Actions with feature-based development

Learn how to protect your main branch and how to trigger tasks in the machine learning workflow based on changes to the code.

Lessons:

- Introduction
- Understand the business problem
- Explore the solution architecture
- Trigger a workflow
- Exercise
- Module assessment
- Summary

After completing this module, you will be able to:

- Work with feature-based development.
- Protect the main branch.
- Trigger a GitHub Actions workflow by merging a pull request.

Module 6: Work with environments in GitHub Actions

Learn how to train, test, and deploy a machine learning model by using environments as part of your machine learning operations (MLOps) strategy.

Lessons:

- Introduction
- Understand the business problem
- Explore the solution architecture
- Set up environments
- Exercise
- Module assessment
- Summary

After completing this module, you will be able to:

- Set up environments in GitHub.
- Use environments in GitHub Actions.
- Add approval gates to assign required reviewers before moving the model to the next environment.

Module 7: Deploy a model with GitHub Actions

Learn how to automate and test model deployment with GitHub Actions and the Azure Machine Learning CLI (v2).

Lessons

- Introduction
- Understand the business problem

- Explore the solution architecture
- Model deployment
- Exercise
- Module assessment
- Summary

After completing this module, you will be able to:

- Deploy a model to a managed endpoint.
- Trigger model deployment with GitHub Actions.
- Test the deployed model.

Module 8: Plan and prepare a GenAIOps solution

Learn how to develop chat applications with language models using a code-first development approach. By developing generative AI apps code-first, you can create robust and reproducible flows that are integral for generative AI Operations, or GenAIOps.

Lessons

- Introduction
- Explore use cases for GenAIOps
- Select the right generative AI model
- Understand the development lifecycle of a language model application
- Explore available tools and frameworks to implement GenAIOps
- Exercise - Compare language models from the model catalog
- Module assessment
- Summary

After completing this module, you will be able to:

- Identify use cases for Generative AI applications.
- Select a model for your Generative AI application.
- Describe what GenAIOps is, and how it defines the app lifecycle.

Module 9: Manage prompts for agents in Microsoft Foundry with GitHub

Learn how to manage AI prompts as versioned assets using GitHub. Apply software engineering best practices to create, test, and promote prompt versions used in Microsoft Foundry as part of a GenAIOps workflow.

Lessons:

- Introduction
- Apply version control to prompts
- Understand Microsoft Foundry agents and prompt versioning
- Organize prompts in GitHub repositories
- Develop safe prompt deployment workflows
- Exercise - Develop prompt and agent versions

- Knowledge check
 - Summary
- After completing this module, you will be able to:
- Apply version control principles to manage prompts as code assets.
 - Understand how prompts integrate with Microsoft Foundry agents and versioning strategies.
 - Design a GitHub repository structure for prompt versioning and collaboration.
 - Develop a workflow for testing and deploying prompts safely.

Module 10: Evaluate and optimize AI agents through structured experiments

Learn how to optimize AI agents through structured evaluation that transforms guesswork into evidence-based engineering decisions. You'll explore how to design evaluation experiments with clear metrics for quality, cost, and performance; organize experiments using Git-based workflows; create evaluation rubrics for consistent scoring; and compare results to make informed optimization decisions.

Lessons

- Introduction
- Design evaluation experiments
- Apply Git-based workflows to optimization experiments
- Apply evaluation rubrics for consistent scoring
- Exercise - Evaluate and compare AI agent versions
- Knowledge check
- Summary

After completing this module, you will be able to:

- Design evaluation experiments with clear metrics for quality, cost, and performance
- Apply Git-based workflows to organize and compare agent variants systematically
- Create evaluation rubrics that ensure consistent scoring across human evaluators
- Compare experiment results to make evidence-based optimization decisions.

Module 11: Automate AI evaluations with Microsoft Foundry and GitHub Actions

Learn how to implement automated evaluations for AI agent responses using Microsoft Foundry evaluators, create evaluation datasets from production data and synthetic generation, run batch evaluations with Python scripts, and integrate evaluation workflows into GitHub Actions for continuous quality assurance.

Lessons

- Introduction
- Understand why automated evaluations matter
- Align evaluators with human criteria
- Create evaluation datasets
- Implement batch evaluations with Python
- Integrate evaluations into GitHub Actions
- Exercise - Set up automated evaluations
- Knowledge check
- Summary

After completing this module, you will be able to:

- Explain why automated evaluations complement human evaluations in AI quality assurance.
- Select evaluators that align with human evaluation criteria for validation.
- Create evaluation datasets with appropriate composition for comprehensive testing.
- Implement batch evaluations using Python scripts with Microsoft Foundry.
- Integrate automated evaluation workflows into GitHub Actions for continuous testing.

Module 12: Monitor your generative AI application

Learn how to monitor the performance of your generative AI application using Microsoft Foundry. This module teaches you to track key metrics like latency and token usage to make informed, cost-effective deployment decisions.

Lessons

- Introduction
- Why do you need to monitor?
- Understand key metrics to monitor
- Explore how to monitor with Azure

- Integrate monitoring into your app
- Interpret monitoring results
- Exercise - Enable monitoring for a generative AI application
- Knowledge check
- Summary

After completing this module, you will be able to:

- Understand why monitoring is essential when moving Gen AI apps toward production readiness.
- Identify and interpret key performance metrics: latency, throughput, token usage, and error rates.
- Use Azure Monitor together with Microsoft Foundry to observe and analyze app behavior.
- Apply insights to optimize performance, cost, and user experience in Gen AI solutions.

Module 13: Analyze and debug your generative AI app with tracing

Learn how to implement tracing in your generative AI applications using Microsoft Foundry and OpenTelemetry. This module teaches you to capture detailed execution flows, debug complex workflows, and understand application behavior for better reliability and optimization.

Lessons

- Introduction
- Why do you need to use tracing?
- Identify what to trace in generative AI applications
- Implement tracing in generative AI applications
- Debug complex workflows with advanced tracing patterns
- Make informed decisions with trace data analysis
- Exercise - Enable tracing for a generative AI application
- Knowledge check
- Summary

After completing this module, you will be able to:

- Deploy a model to a managed endpoint.
- Trigger model deployment with GitHub Actions.
- Test the deployed model

ASSOCIATED CERTIFICATIONS & EXAM

This course will prepare delegates to write the Microsoft AI-300: Operationalizing Machine Learning and Generative AI Solutions Exam.