APPENDIX A

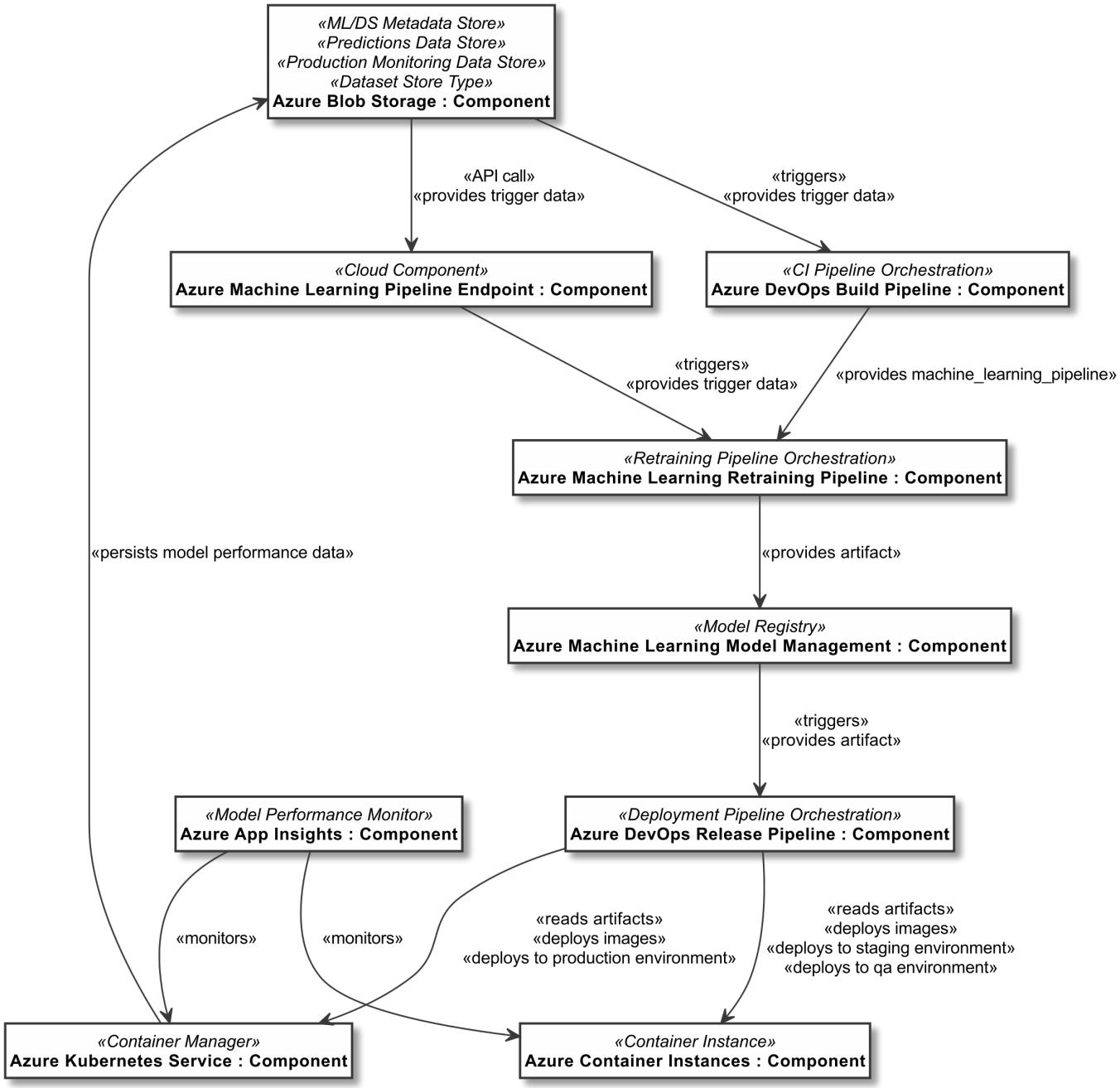

Fig. 14: Component diagram.

Figure 14 depicts the components for the Microsoft Azure DevOps and Azure Machine Learning system described in Section III-B. The Azure Blob Storage component, which stores various kinds of data, can trigger either the Azure Machine Learning Pipeline Endpoint via an API call (which subsequently triggers the Azure Machine Learning Retraining Pipeline) or the Azure DevOps Build Pipeline (which is a CI pipeline orchestration component type).

The build pipeline provides an ML pipeline to the Azure Machine Learning Retraining Pipeline, which is an orchestration component type, which subsequently provides an ML model to the Azure Machine Learning Model Management model registry component. This registry then triggers the Azure DevOps Release Pipeline. This deployment pipeline orchestration component type deploys model images to Azure Container Instances (in the case of deployments to staging or quality assurance environments) or Azure Kubernetes Services (in the case of deployments to production environments).

Azure App Insights is a model performance monitor component that monitors the Azure Kubernetes Service. The Azure Kubernetes Service saves model performance data to Azure Blob Storage.

APPENDIX B

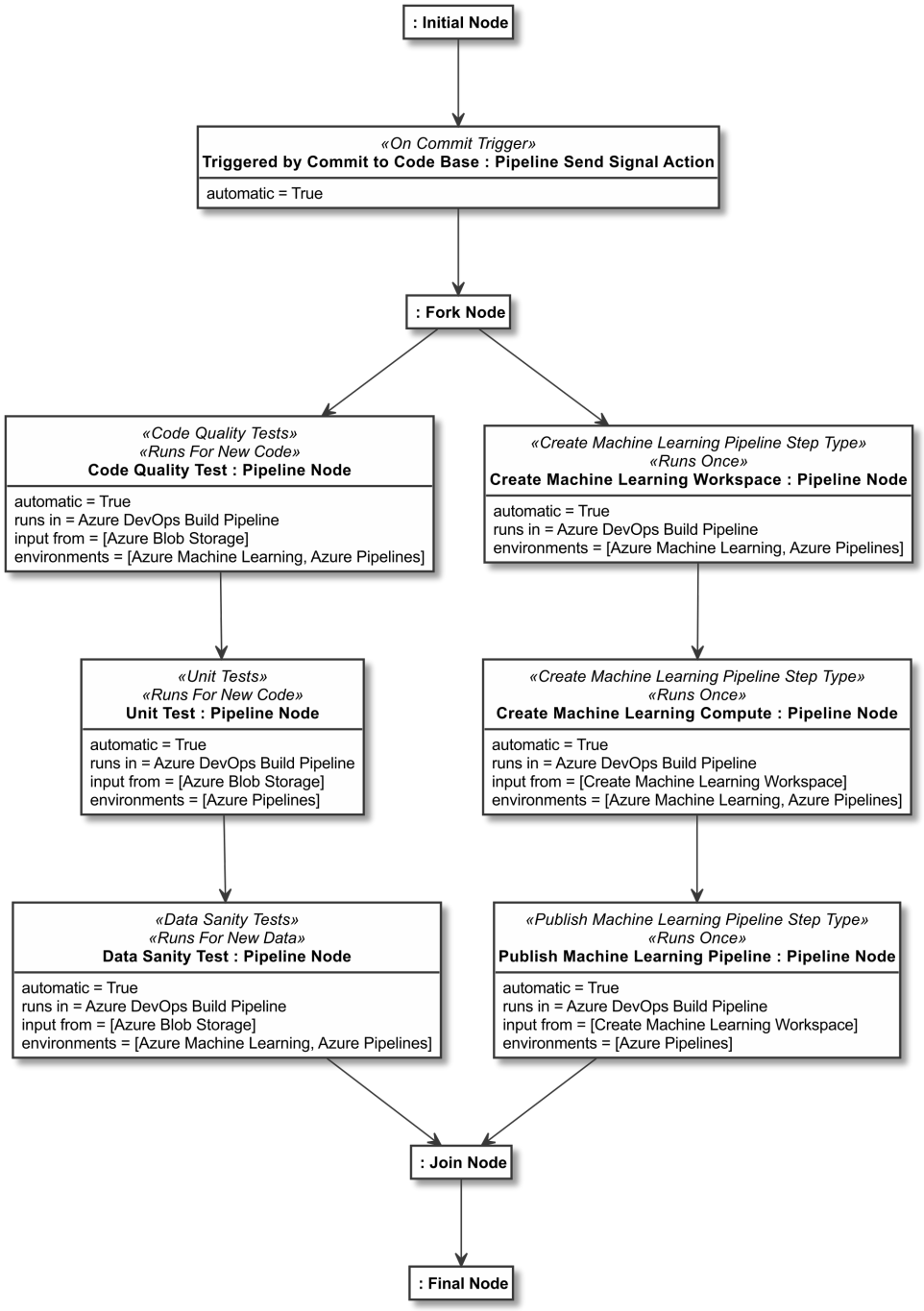

Fig. 15: Build pipeline diagram.

Figure 15 depicts the build pipeline for the Microsoft Azure DevOps and Azure Machine Learning system described in Section III-B. The build pipeline operates as a crucial component of the development process, triggered automatically with each code check-in. This pipeline's primary function is to publish an updated Azure Machine Learning pipeline. This process encompasses building the code and executing a suite of tests to ensure the integrity and quality of the codebase. The build pipeline itself consists of several distinct steps, each contributing to the overall goal. All pipeline steps run automatically within the Azure DevOps Build Pipeline. The test steps take their input from Azure Blob Storage, and all pipeline steps run in either Azure Pipelines or Azure Machine Learning environments.

One of the key aspects within the build pipeline focuses on code quality. These tasks are instrumental in ensuring that

the code adheres to the established standards and guidelines of the development team. The unit tests play a vital role in verifying the code's functionality, assessing its code coverage, and confirming its stability. Additionally, the build pipeline encompasses data tests. These tests are specifically designed to validate that the data samples utilised within the system align with the anticipated schema and distribution. It's worth noting that the data test can be customised to suit different use cases.

Aside from the recurring build pipeline tasks, there are several one-time setup tasks associated with establishing the infrastructure for Azure Machine Learning and the Python SDK. These tasks are foundational to the overall operation of the machine learning system. Firstly, creating the workspace that hosts all Azure Machine Learning-related resources is a critical initial step. The workspace serves as the central hub for organising and managing various ML components and resources. Another essential task involves setting up the computing resources responsible for executing training jobs. These resources are pivotal for the efficient execution of machine learning tasks, ensuring that the necessary computational power is readily available. Subsequently, the creation of the machine learning pipeline, complete with the updated training script, is a one-time task that streamlines the workflow. This pipeline encapsulates the entire process of training machine learning models and orchestrating the associated tasks. Finally, the machine learning pipeline is published as a REST endpoint, providing a valuable interface for orchestrating the training workflow. This REST endpoint simplifies the process of initiating and managing training runs, promoting a more streamlined and automated approach to machine learning.