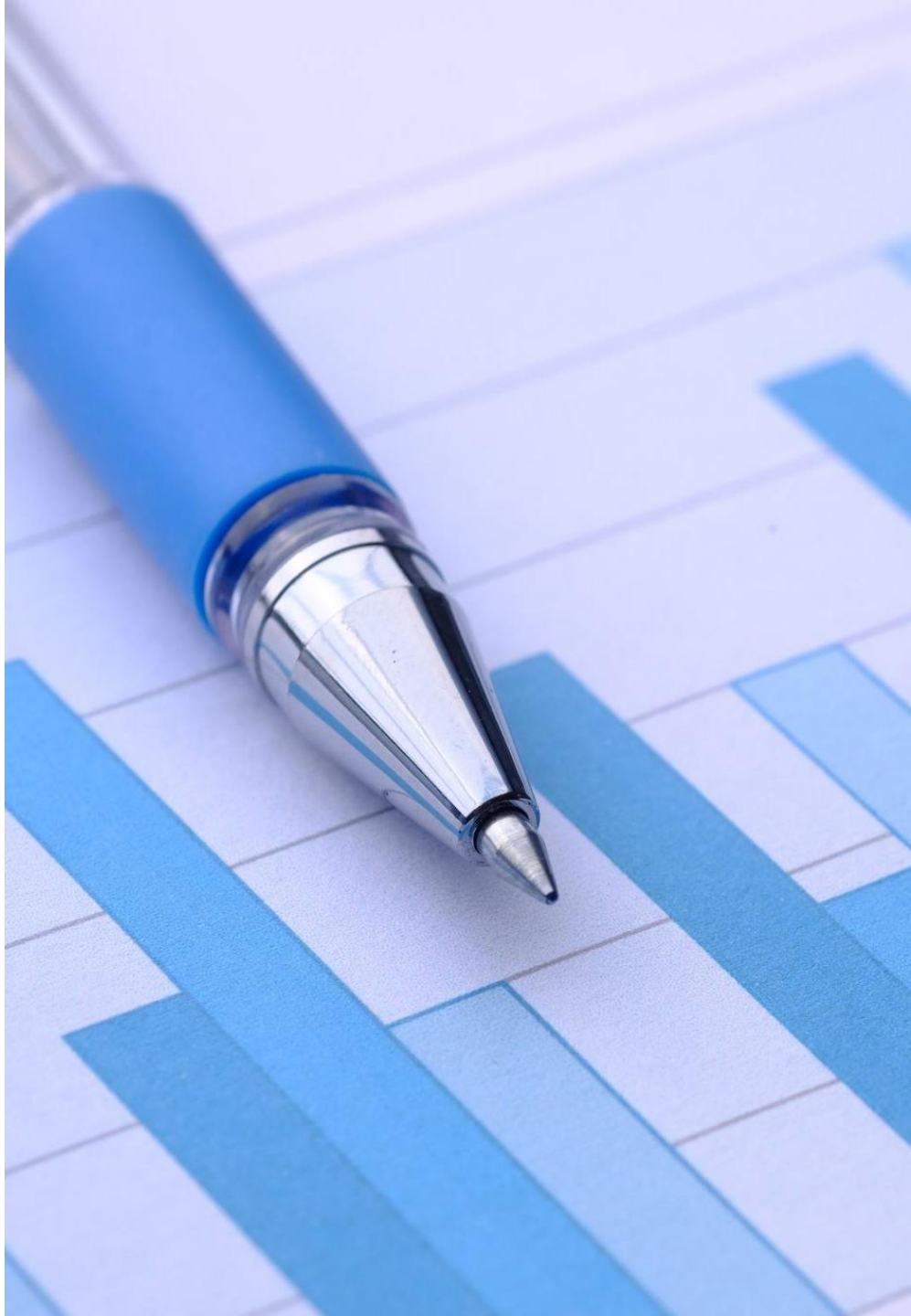
KUBEFLOW IMPLEMENTATION ON AZURE

AGENDA

- Introduction to Kubeflow
- Benefits of Kubeflow on Azure
- Prerequisites for Implementation
- Step-by-step Implementation Guide
- Best Practices & Tips
- Troubleshooting
- Conclusion



INTRODUCTION TO KUBEFLOW

- **Kubeflow** is an open-source platform designed to enable easy deployment, orchestration, and scaling of machine learning (ML) workflows on Kubernetes. It provides a set of purpose-built components and tools to help ML practitioners develop, train, and serve models while leveraging the scalability and flexibility of Kubernetes.

KEY COMPONENTS

01

KUBEFLOW PIPELINES: OFFERS A PLATFORM TO COMPOSE, DEPLOY, AND MANAGE END-TO-END ML WORKFLOWS.

02

KATIB: PROVIDES HYPERPARAMETER TUNING TO OPTIMIZE MODEL TRAINING.

03

KFSERVING: ENABLES THE SERVING AND MONITORING OF ML MODELS IN A SCALABLE WAY.

04

JUPYTER NOTEBOOK: INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) FOR CREATING AND SHARING LIVE CODE, EQUATIONS, AND VISUALIZATIONS.

05

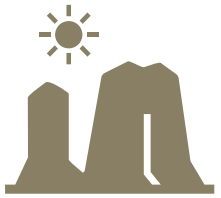
TFJOB & PYTORCHJOB: CUSTOM KUBERNETES CONTROLLERS FOR RUNNING TENSORFLOW AND PYTORCH JOBS, RESPECTIVELY.

BENEFITS OF KUBEFLOW ON AZURE

- Seamless Scalability
- Integrated Developer Experience
- Azure's AI and Data Services
- Cost Efficiency

PRE-REQUISITES FOR IMPLEMENTATION

Azure Subscription and Account



Azure Kubernetes Service (AKS) Cluster

Helm, kubectl, and kfctl tools installed

Basic knowledge of Kubernetes

STEP 1: SETTING UP AKS

Create

Create an AKS Cluster:

- Use the Azure Portal or Azure Command-Line Interface (CLI) to initiate the creation of an Azure Kubernetes Service (AKS) cluster, specifying the desired configurations.

Configure

Configure AKS Networking:

- Set up the networking for the AKS cluster to ensure secure and efficient communication between services, pods, and external resources. This includes defining network policies, setting up virtual networks, and configuring ingress and egress rules.

STEP 2: INSTALLING REQUIRED TOOLS

Helm:

- A package manager for Kubernetes, Helm simplifies the deployment of applications and services onto Kubernetes clusters by using pre-defined packages called charts.

kubectl:

- The official command-line tool for interacting with and managing Kubernetes clusters. It allows users to deploy applications, inspect resources, and manage cluster operations.

kfctl:

- Kubeflow's command-line interface (CLI) tool, **kfctl** streamlines the setup, deployment, and management of Kubeflow components on a cluster.

STEP 3: DEPLOYING KUBEFLOW ON AKS

Download

Download Kubeflow's Configuration:

- Obtain the necessary configuration files, typically in YAML format, from Kubeflow's official repositories. These files define how Kubeflow components will be deployed on the AKS cluster.

Customize

Customize Configurations:

- If needed, modify the downloaded configurations to suit specific requirements, such as resource limits, storage options, or custom integrations.

Use

Use kfctl to Deploy:

- With the kfctl command-line tool, initiate the deployment of Kubeflow onto the AKS cluster using the provided or customized configuration files.

STEP 4: ACCESSING KUBEFLOW DASHBOARD

- **Port Forwarding or Load Balancer Setup:**
  - Configure network access to the Kubeflow services. This can be done using port forwarding for local access or by setting up a load balancer for broader access.
- **Access the Kubeflow Central Dashboard:**
  - Once the network access is set up, use a web browser to navigate to the Kubeflow dashboard's URL. This dashboard provides a central interface to manage and monitor Kubeflow's components and workloads.

STEP 5: RUNNING WORKFLOWS & PIPELINES

Introduction to Kubeflow Components:

Familiarize oneself with the core components of Kubeflow, including tools for training, hyperparameter tuning, serving, and more, to understand how they fit into ML workflows.

Deploying a Sample ML Pipeline:

Utilize Kubeflow Pipelines to deploy a predefined or custom ML pipeline, which automates the stages of training, evaluation, and deployment of machine learning models.

BEST PRACTICES & TIPS

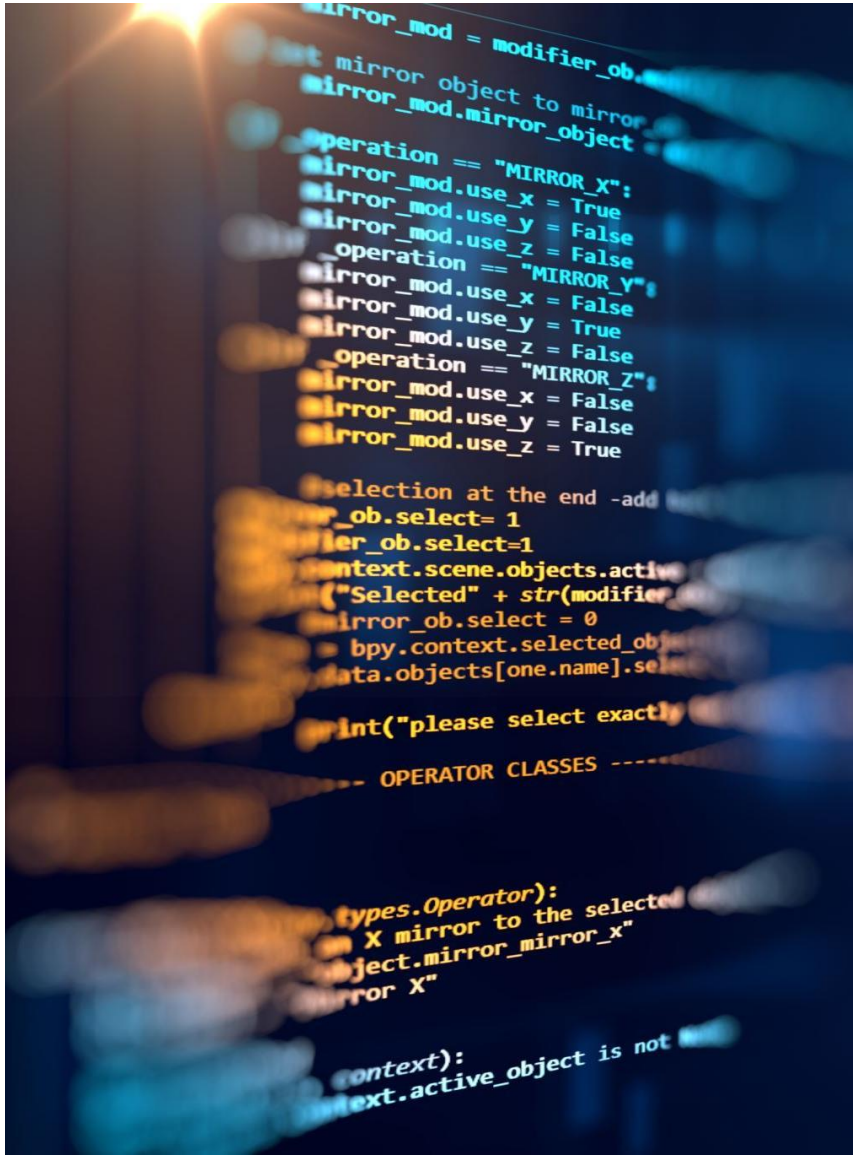
- Continuous monitoring of resources
- Securing your Kubeflow deployment
- Regularly updating components
- Handling data with care and diligence

TROUBLESHOOTING COMMON ISSUES

- Networking and connectivity challenges
- Resource quotas and limitations
- Authentication and RBAC issues
- Image: Error icon with common solutions

INTEGRATION WITH AZURE SERVICES

- Connecting to Azure Blob Storage
- Using Azure Machine Learning with Kubeflow
- Leveraging Azure Monitor and Azure DevOps



CONCLUSION & KEY TAKEAWAYS

Importance of Kubeflow in MLOPs:

- Kubeflow plays a pivotal role in Machine Learning Operations (MLOPs) by streamlining the end-to-end machine learning lifecycle, from data preprocessing to model deployment.

Ease of Deployment on Azure:

- Deploying Kubeflow on Azure simplifies the setup process, thanks to Azure's integrated tools and services, making it straightforward for users to get started.

Leveraging Azure's Ecosystem:

- By utilizing Kubeflow on Azure, users can tap into Azure's extensive array of cloud services, enhancing the capabilities and scalability of their ML workflows.