Databricks Sandbox POC – Synopsis

This document outlines the phases and tasks required to design and implement a secure, automated Sandbox environment on **Azure Free Trial** using **GitHub Actions** and **Bicep**.
Each phase highlights objectives and deliverables without prescriptive step-by-step guidance.

Phase 0: Account & Tools Setup (Prerequisites)

Category: Setup
Synopsis:

- Establish an Azure Free Trial subscription with necessary verification.
- Prepare local and cloud tooling, including Azure CLI.
- Ensure connectivity between GitHub Actions and Azure subscription via service principal or federated identity.
- Validate baseline access to Resource Groups and subscription context.

Outcome: Ready-to-use Azure subscription, CLI access, and authentication channel for automation.

Phase 1: Plan Architecture

Category: Design
Synopsis:

- Define a hub-and-spoke network topology to isolate application and data workloads.
- Identify resource boundaries: AKS and Key Vault in application spoke; Storage, SQL, and Databricks in data spoke.
- Establish traffic flow rules and security boundaries.
- Determine identity model leveraging managed identities and service principals.

- Define a secret management strategy centralized on Azure Key Vault.

Outcome: Architecture blueprint defining connectivity, isolation, and identity/security design.

Phase 2: Infrastructure as Code (Bicep Templates)

Category: Infrastructure
Synopsis:

- Create modular Bicep templates representing network, security, compute, data, and monitoring layers.
- Organize repository with clear folder structure for infra, apps, workflows, and documentation.
- Implement reusable modules for VNets, AKS, Key Vault, Databricks, ADLS, and monitoring.
- Ensure templates are parameterized for flexible deployment and reusability.
- Validate IaC definitions against Azure standards.

Outcome: Deployable set of Bicep templates covering end-to-end Sandbox infrastructure.

Phase 3: Sample Workloads & Configurations

Category: Applications & Data
Synopsis:

- Deploy lightweight applications on AKS (e.g., simple APIs, scheduled jobs, UI component).
- Integrate applications with Key Vault via CSI driver for secure secret access.
- Establish Databricks workload to demonstrate ingestion, transformation, and persistence of sample datasets.
- Validate functional connectivity between application layer, data layer, and secret store.

Outcome: Running workloads showcasing Sandbox functionality and data flow.

Phase 4: GitHub & CI/CD Pipelines

Category: Automation
Synopsis:

- Design GitHub Actions workflows to validate, deploy, and destroy Sandbox resources.
- Implement separate pipelines for pull request validation, main branch deployments, and manual teardown.
- Automate infrastructure provisioning, application deployment, and optional Databricks job triggers.
- Ensure workflows enforce code-driven governance and auditability.

Outcome: Automated CI/CD pipeline enabling consistent, repeatable environment lifecycle management.

Phase 5: Monitoring, Alerting & Self-Healing

Category: Operations
Synopsis:

- Enable observability for AKS clusters and Databricks jobs.
- Configure alerts for key operational scenarios (e.g., pod restarts, ETL failures).
- Demonstrate self-healing mechanisms such as pod auto-restart and job retries.
- Establish monitoring as an integrated part of the Sandbox lifecycle.

Outcome: Observable and resilient Sandbox environment.