### **Week 1: Rapid Containerization with Docker and Introduction to Kubernetes**

* Docker Fundamentals
  + - **Introduction to Containerization**
      * Understand the differences between virtualization and containerization.
      * Learn how containers solve deployment consistency issues.
    - **Installing Docker**
      * Install Docker on your local machine.
      * Familiarize yourself with Docker CLI commands.
    - **Working with Docker Images and Containers**
      * Pull and run Docker images from Docker Hub.
      * Understand the lifecycle of a container.
    - **Creating Custom Docker Images**
      * Write a simple Dockerfile.
      * Build and tag your custom Docker images.
* Multi-Container Applications and Transition to Kubernetes
  + - **Docker Compose**
      * Install Docker Compose.
      * Define and run multi-container applications using docker-compose.yml.
      * Practice with sample applications (e.g., web app with a database).
    - **Introduction to Kubernetes**
      * Understand the limitations of Docker in handling large-scale applications.
      * Learn about Kubernetes architecture: Master node, Worker nodes, Pods, Services, etc.
    - **Setting Up Kubernetes Environment**
      * Install Minikube or access a cloud-based Kubernetes cluster.
      * Install kubectl for cluster management.
* Deep Dive into Kubernetes
  + Deploying Applications on Kubernetes
    - Create and manage Pods manually.
    - Use Deployments for scaling and updating applications.
    - Expose applications using Services (ClusterIP, NodePort, LoadBalancer).
  + Kubernetes Configuration and Secrets Management
    - Use ConfigMaps to manage application configurations.
    - Secure sensitive information using Secrets.
    - Practice injecting configurations and secrets into Pods.
  + Storage and Stateful Applications
    - Understand PersistentVolumes and PersistentVolumeClaims.
    - Deploy StatefulSets for stateful applications.
    - Configure dynamic provisioning of storage.
  + Advanced Kubernetes Concepts
    - Explore Ingress controllers for external access.
    - Implement service discovery and load balancing.
    - Learn about DaemonSets and Jobs.
  + Hands-On Project
    - Deploy a complex application (e.g., a microservices-based app).
    - Implement autoscaling with Horizontal Pod Autoscaler.
    - Monitor the cluster using Kubernetes dashboard and CLI tools.

### **Week 2: Automating Infrastructure with Terraform**

* Introduction to Infrastructure as Code (IaC)
  + Understand the principles and benefits of IaC.
  + Install Terraform and configure the environment.
  + Learn about Terraform providers and resources.
* Terraform Basics
  + Write your first Terraform configuration file.
  + Understand Terraform's workflow: init, plan, apply, and destroy.
  + Manage Terraform state files and remote backends.
* Variables, Outputs, and Provisioners
  + Use variables and outputs for dynamic configurations.
  + Employ provisioners to execute scripts on remote machines.
  + Handle dependencies between resources.
* Terraform Modules and Reusability
  + Create reusable modules for common configurations.
  + Learn best practices for module structure.
  + Use community modules from the Terraform Registry.
* Infrastructure Provisioning on Cloud Providers
  + Configure Terraform to work with AWS, Azure, or GCP.
  + Provision a Kubernetes cluster using managed services like EKS, AKS, or GKE.
  + Manage networking components (VPCs, subnets, security groups).
* Collaboration and Team Workflows
  + Implement Terraform workspaces for environment isolation.
  + Use version control with Terraform configurations.
  + Understand Terraform Cloud and Enterprise features.
* Hands-On Project
  + Automate the provisioning of infrastructure for your Kubernetes cluster.
  + Integrate Terraform configurations into a CI/CD pipeline.

### **Week 3: Streamlining Deployments with Jenkins**.

* Jenkins Setup and Fundamentals
  + Install Jenkins and configure initial settings.
  + Learn about Jenkins architecture and plugins.
  + Create your first freestyle project.
* Pipeline as Code with Jenkinsfile
  + Write declarative and scripted Jenkins pipelines.
  + Understand stages, steps, and post actions.
  + Use Jenkinsfile for pipeline versioning.
* Integrating Jenkins with Version Control Systems
  + Connect Jenkins with GitHub, GitLab, or Bitbucket.
  + Trigger builds based on code commits and pull requests.
  + Manage credentials and secure access.
* Automating Docker Builds and Tests
  + Write pipelines to build Docker images.
  + Incorporate unit and integration tests.
  + Push images to Docker Hub or private registries.
* Deploying Applications to Kubernetes via Jenkins
  + Configure Jenkins agents to interact with Kubernetes.
  + Implement blue-green and canary deployments.
  + Roll back deployments in case of failures.
* Advanced Jenkins Features
  + Set up Jenkins distributed builds with master-agent architecture.
  + Use Jenkins credentials and secret management.
  + Monitor Jenkins and optimize performance.
* Hands-On Project
  + Develop a robust CI/CD pipeline for your application.
  + Include stages for building, testing, security scanning, and deployment.
  + Implement notifications and reporting.

### **Week 4: In-Depth Exploration of Kubeflow**.

* Understanding Kubeflow Architecture
  + Explore Kubeflow components: Notebooks, Pipelines, KFServing, Katib, etc.
  + Learn how Kubeflow leverages Kubernetes for ML workloads.
  + Set up Kubeflow on your existing Kubernetes cluster.
* Kubeflow Notebooks and Data Management
  + Launch Jupyter Notebooks within Kubeflow.
  + Access data from various sources (S3, GCS, Azure Blob).
* Building and Managing Kubeflow Pipelines
  + Create complex ML pipelines using the Kubeflow Pipelines SDK.
  + Understand pipeline components, parameters, and artifacts.
  + Visualize pipeline runs and track experiments.
* Distributed Training and Hyperparameter Tuning
  + Use TensorFlow, PyTorch, and MPI operators for distributed training.
  + Implement hyperparameter tuning with Katib.
  + Analyze results to optimize model performance.
* Model Serving with KFServing
  + Deploy models using serverless inference with KFServing.
  + Explore advanced serving features like batching and scaling.
  + Implement custom inference logic.
* Security and Multi-Tenancy in Kubeflow
  + Understanding the authentication with Identity-Aware Proxy (IAP) or OIDC.
  + Set up role-based access control (RBAC) for users and teams.
  + Isolate resources using namespaces and network policies.
* Hands-On Project
  + Build an end-to-end ML workflow:
    - Data preprocessing
    - Model training and validation
    - Hyperparameter tuning
    - Model deployment and monitoring
  + Document the workflow and present findings.

### **Week 5: Mastering Experiment Tracking with MLflow**

* MLflow Tracking Deep Dive
  + Set up an MLflow tracking server with a backend store.
  + Log experiments, parameters, metrics, and artifacts.
  + Query and visualize experiment results.
* MLflow Projects and Packaging
  + Define MLflow Projects using MLproject files.
  + Package code and dependencies for reproducible runs.
  + Run projects remotely and in Docker environments.
* Advanced MLflow Models
  + Serve models using MLflow's built-in model serving.
  + Integrate custom flavors and model types.
  + Deploy models to cloud services (AWS SageMaker, Azure ML).
* MLflow Model Registry
  + Manage model lifecycle with the Model Registry.
  + Automate transitions between stages (Development, Staging, Production).
  + Implement access control and governance.
* Integrating MLflow with Kubeflow Pipelines
  + Automatically log Kubeflow pipeline runs to MLflow.
  + Use MLflow tracking in distributed training jobs.
  + Visualize MLflow metrics within Kubeflow UI.
* Scaling MLflow for Enterprise Use
  + Deploy MLflow in a multi-user, multi-project environment.
  + Implement authentication and authorization.
  + Optimize performance with database backends and artifact stores.
* Hands-On Project
  + Develop a comprehensive ML project:
    - Use MLflow for experiment tracking and model management.
    - Integrate MLflow with CI/CD pipelines.
    - Deploy the model using Kubeflow and monitor performance.

### **Week 6: Advanced Monitoring and Observability**

**Objective:** Implement sophisticated monitoring and observability strategies to maintain system health and performance.

* Monitoring Principles and Strategies
  + Understand the four golden signals: latency, traffic, errors, and saturation.
  + Learn about white-box vs. black-box monitoring.
  + Define service-level objectives (SLOs) and indicators (SLIs).
* Deep Dive into Prometheus
  + Install and configure Prometheus in a Kubernetes cluster.
  + Write custom PromQL queries for detailed metrics.
  + Set up Alertmanager for notifications.
* Advanced Grafana Dashboards
  + Connect Grafana to multiple data sources.
  + Create dynamic and interactive dashboards.
  + Implement dashboard variables and templating.
* Distributed Tracing with Jaeger or OpenTelemetry
  + Instrument applications for tracing.
  + Visualize request flows and identify bottlenecks.
  + Integrate tracing with logs and metrics.
* Comprehensive Monitoring with Datadog
  + Set up Datadog agents and integrations.
  + Use Datadog APM for application performance.
  + Monitor Kubernetes clusters and serverless functions.
* Log Management with Splunk
  + Configure Splunk for log ingestion from multiple sources.
  + Create complex search queries and dashboards.
  + Set up real-time alerts based on log patterns.
* Hands-On Project
  + Implement a full observability stack:
    - Instrument applications for metrics, logs, and traces.
    - Set up alerts for critical thresholds.
    - Simulate failures and analyze system behavior.

**Finale: Mastering the Cloud-Native MLOps Platform**

By the end of Week 6, you will have gained deep expertise in building and managing a robust machine learning platform:

* **Accelerated Containerization:** Quickly containerize applications using Docker in just two days.
* **Advanced Orchestration:** Utilize Kubernetes for deploying and scaling complex applications.
* **Automated Infrastructure:** Employ Terraform for scalable and repeatable infrastructure management.
* **Sophisticated CI/CD Pipelines:** Implement advanced Jenkins pipelines for continuous integration and delivery.
* **In-Depth Machine Learning Workflows:** Master Kubeflow and MLflow for comprehensive ML lifecycle management.
* **Robust Monitoring and Observability:** Ensure system health with Prometheus, Grafana, Datadog, Splunk, and tracing tools.