House Price Prediction

An end-to-end MLOps pipeline for real estate price prediction

Project Foundation & Dataset

Dataset

1,000+

House Price Records

Comprehensive property dataset with multi-dimensional features for accurate price prediction.

Features

- Square Footage
- Bedrooms
- Bathrooms
- Location
- Year Built
- Condition

Price Statistics

Mean: **$553,234**

Median: **$495,000**

Range: **$249K–$1.25M**

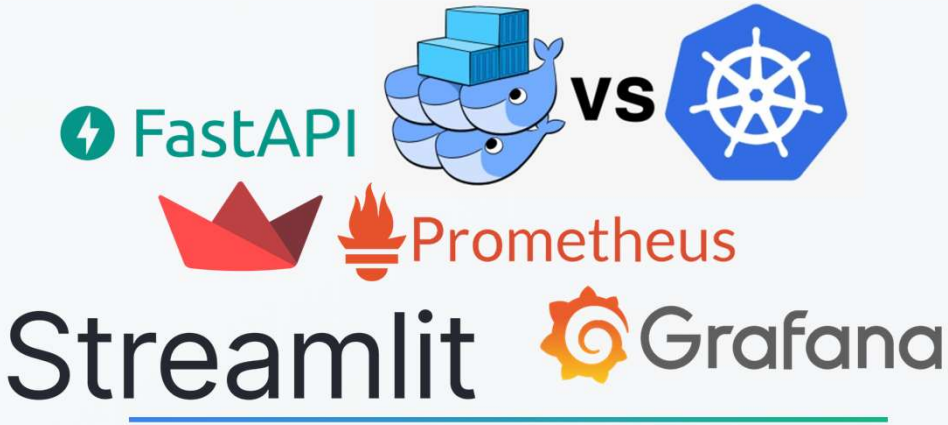
Technology Stack

CI/CD Automation

GitHub Actions automates data processing and container publishing on every Git push.

Model Packaging

FastAPI + Uvicorn converts trained models into production-ready web APIs.



Container Registry

DockerHub stores versioned container images for consistent deployment.

Model Deployment

Kubernetes (kind) manages and scales the containerized application.

Monitoring Stack

Prometheus + Grafana provides real-time performance insights and alerting.

User Interface

Streamlit provides the user-friendly web application for predictions.

End-to-End Workflow Overview

Data Processing

Raw CSV data is cleaned, transformed, and prepared for training with feature engineering and validation.

Model Training

GradientBoosting model trained with comprehensive MLflow experiment tracking and versioning.

Build & Publish (CI)

GitHub Actions orchestrates testing, packages model with FastAPI into versioned Docker images, and pushes to Docker Hub.

Deploy (CD)

Kubernetes pulls images from Docker Hub, deploys pods, and configures load balancer for high availability.

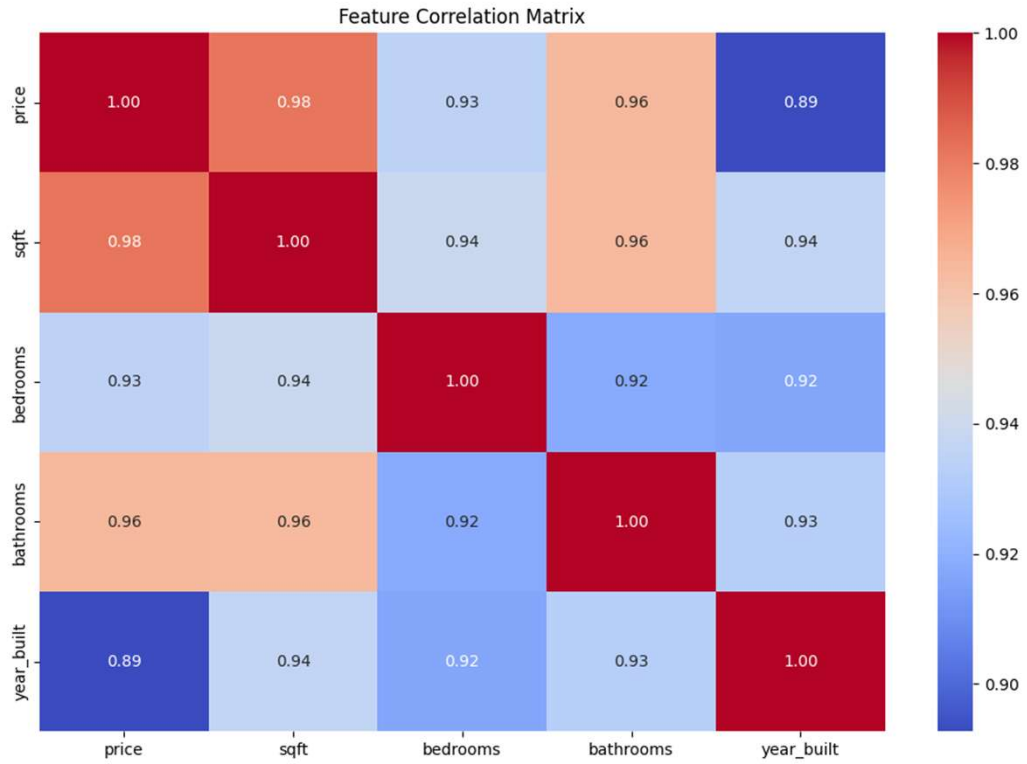
Serve, Scale & Monitoring

- Load balancer distributes traffic across multiple pods
- Ensures consistent performance under any load
- Real-time monitoring and alerting

GitHub Actions · Docker · Kubernetes · FastAPI · Prometheus + Grafana

Feature Relationships & Analysis

Feature Correlation Matrix



Key Insights

Strongest Correlations

Price shows strong positive correlation (>0.98) with square footage, indicating size is a primary pricing factor.

Property Size Impact

Square footage demonstrates the strongest linear relationship with price among all features.

Amenity Value

Bathrooms show significant correlation with price(0.96), highlighting the importance of luxury amenities.

Model Performance Comparison

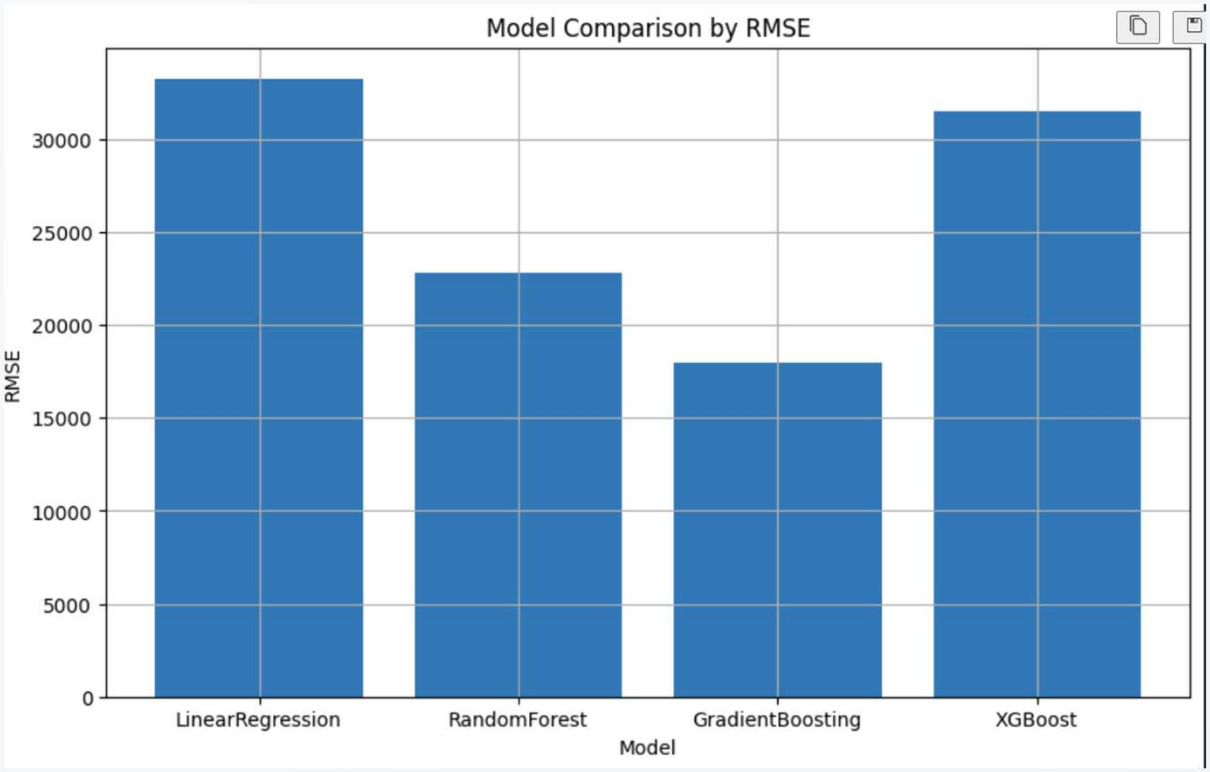
Comparison of different ML algorithms for house price prediction. GradientBoosting model shows optimal performance for this regression task.

GradientBoosting

R^2 Score:	0.99
RMSE (USD):	$17,968
Error %	3.25%

Performance Characteristics

Model	R^2 Score	RMSE (USD)	Error %
GradientBoosting	0.9931	$17,968	3.25%
Random Forest	0.9888	$22,818	4.12%
XGBoost	0.9787	$31,513	5.70%
Linear Regression	0.9762	$33,267	6.01%



MLFlow for Experiment Tracking

Overview

Comprehensive

Experiment Management

MLflow provides comprehensive tracking and and versioning for all model experiments, ensuring ensuring reproducibility and easy comparison. comparison.

- Reproducibility
- Easy Comparison
- Version Control

Key Features

Hyperparameter Tracking
Record and compare model parameters across experiments

Metrics Logging
Track performance metrics to identify optimal models

Artifact Storage
Save model artifacts and configurations

Configuration Export
Export to model_config.yaml for deployment

Model Export
Export best_model.pkl for production use

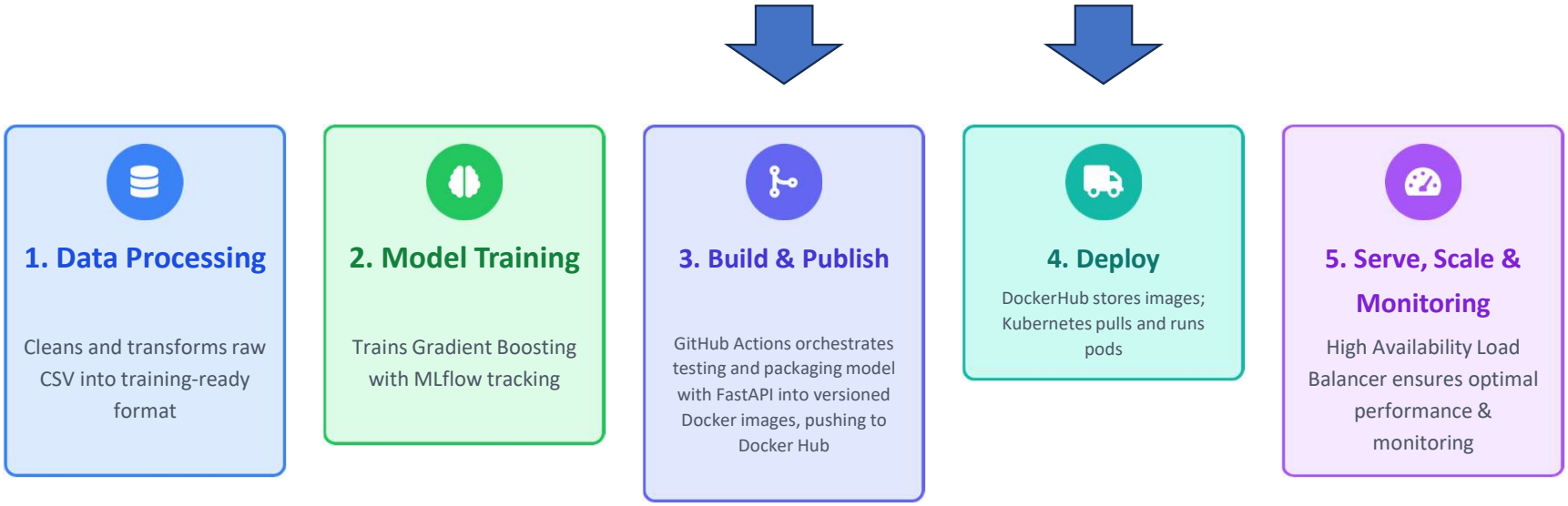
Workflow

01 **Configure Parameters**
Set estimators to 250 for optimal performance

02 **Track Experiments**
Log all metrics and model versions

03 **Save Artifacts**
Export configuration and trained model

Workflow Overview



Production API & User Interface

FastAPI Backend

Production-ready API for house price prediction:

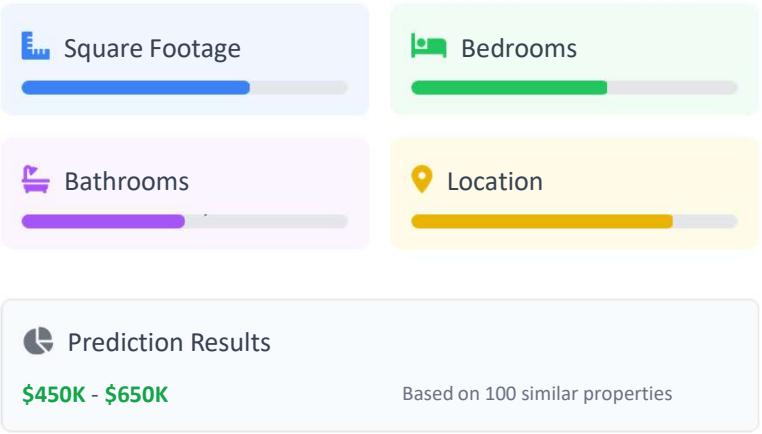
/predict
Single prediction endpoint

/batch-predict
Bulk processing capability

/metrics
Performance monitoring

Streamlit Frontend

User-friendly interface for predictions:



FastAPI + Uvicorn

Streamlit

Streamlit Application

User Interface

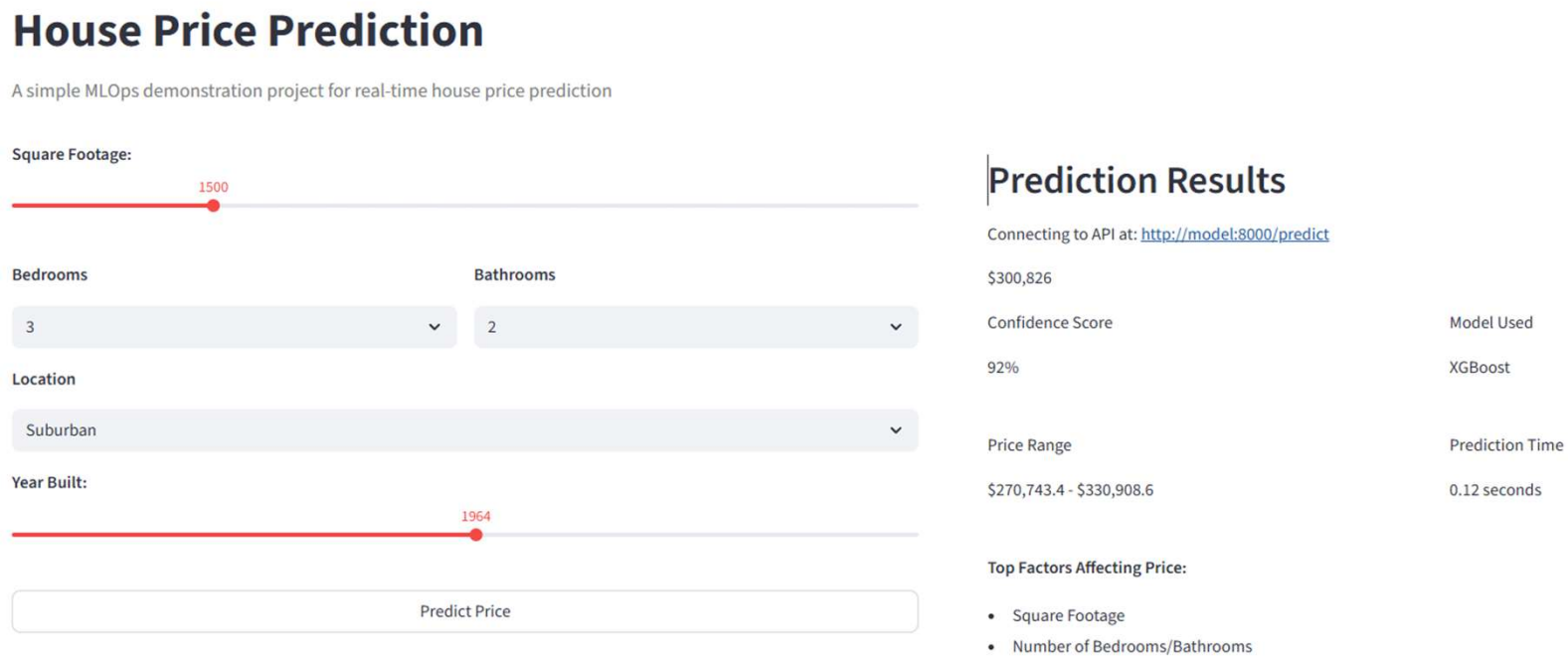
Streamlit provides a web app for house price predictions with:

- Simple input forms for property details
- Real-time price estimation
- Visualization of influencing factors

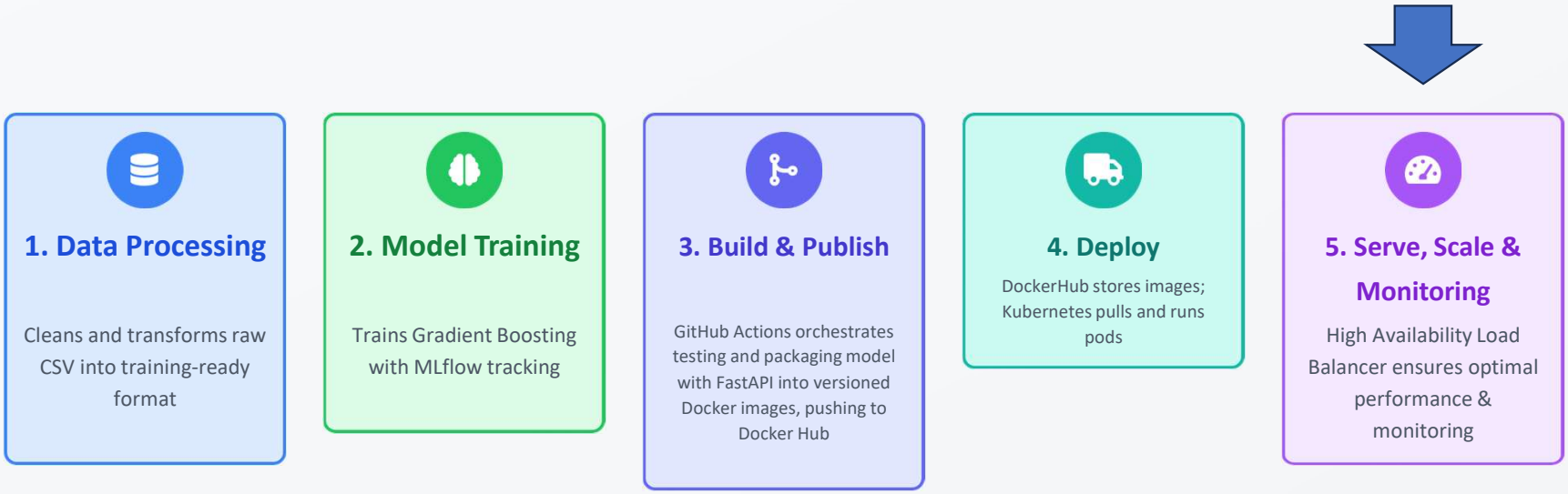
Key Input Fields

- Square Footage
- Bedrooms & Bathrooms
- Location

Prediction Results



Workflow Overview



Docker Containerization

The Magic Packaging Box

Docker is like a sturdy, special box that securely packs your application with everything it needs to run.

This "box" ensures your application runs consistently across any environment, solving the "But it worked on my machine!" problem.

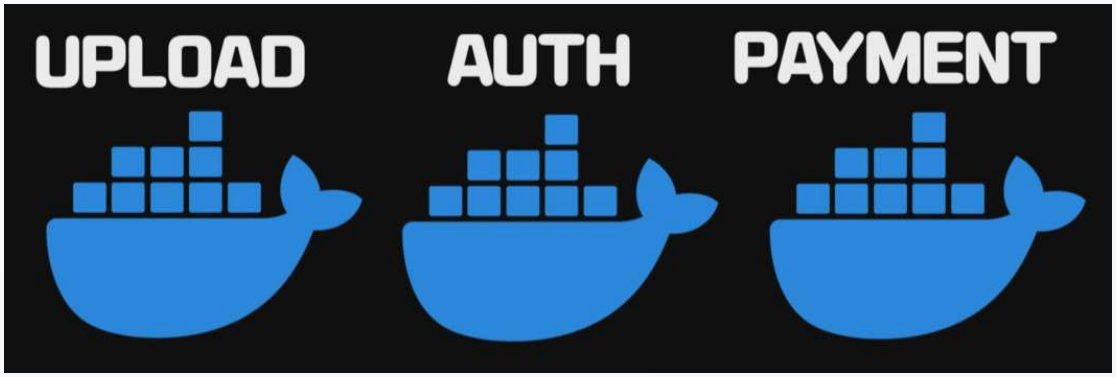
Universal Compatibility

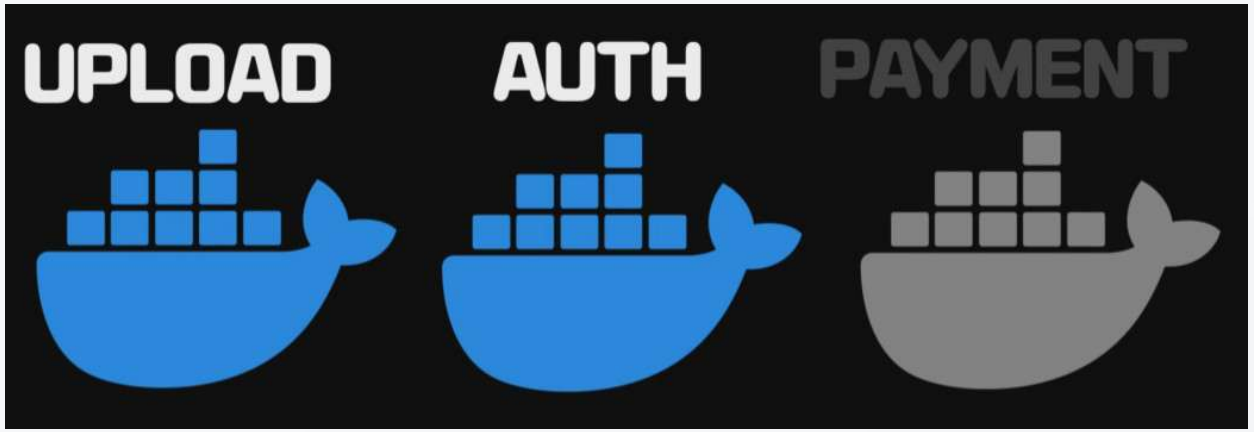
Works identically on any computer—no more environment-specific issues.

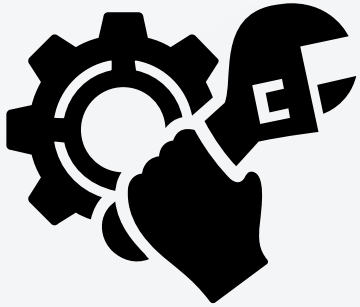
Complete Package

Packs your application, dependencies, and configuration into one perfect container.

Why Kubernetes?







Kubernetes Orchestration

Kubernetes orchestrates Docker containers like a factory manager overseeing production at scale.

Auto-Scaling

Automatically adds containers during peak traffic and removes them during quiet periods.

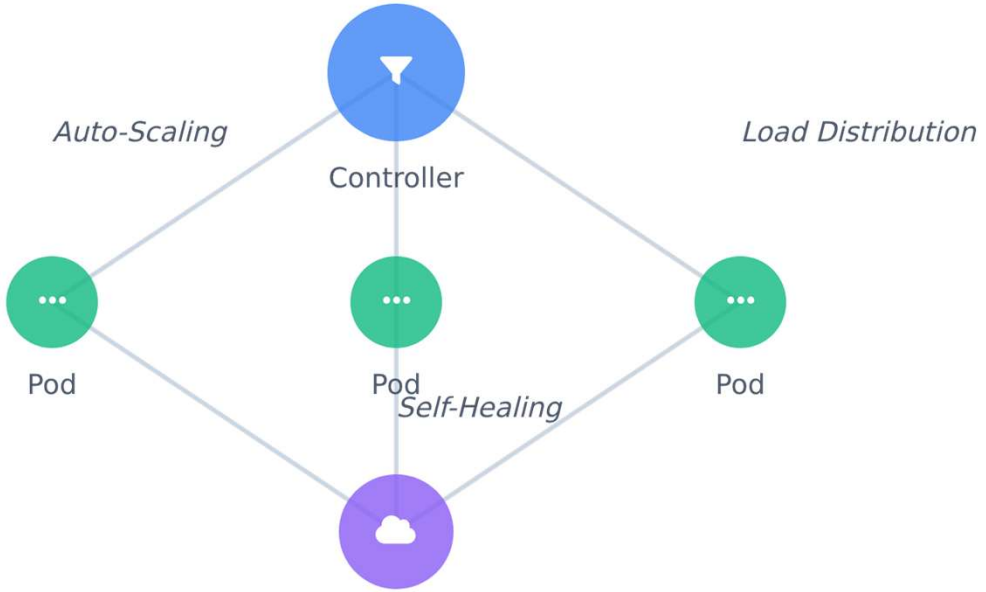
Load Distribution/Orchestration

Distributes containers evenly across many servers during high traffic.

Self-Healing

Detects failures instantly and moves containers to healthy machines.

Orchestration Visualization



Kubernetes Core Resources

Pod

- The smallest execution unit Kubernetes manages
- A collection of one or more Docker containers
- Share networking and storage resources
- Example: FastAPI model or Streamlit UI

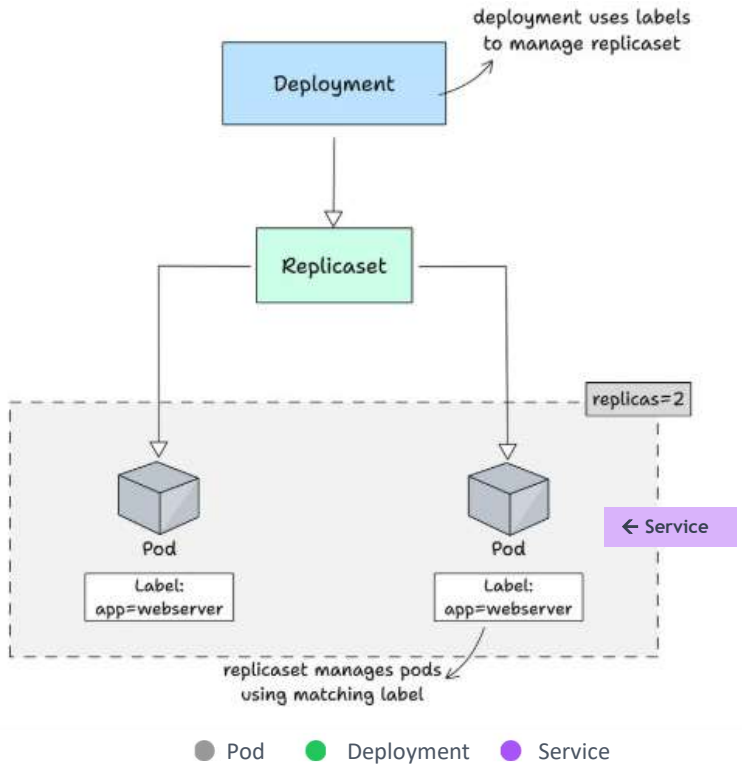
Deployment

- Management rule that automatically controls and replicates Pods
- Ensures FastAPI Model Pods are kept at the specified number of replicas
- Automatically recovers them if one fails

Service

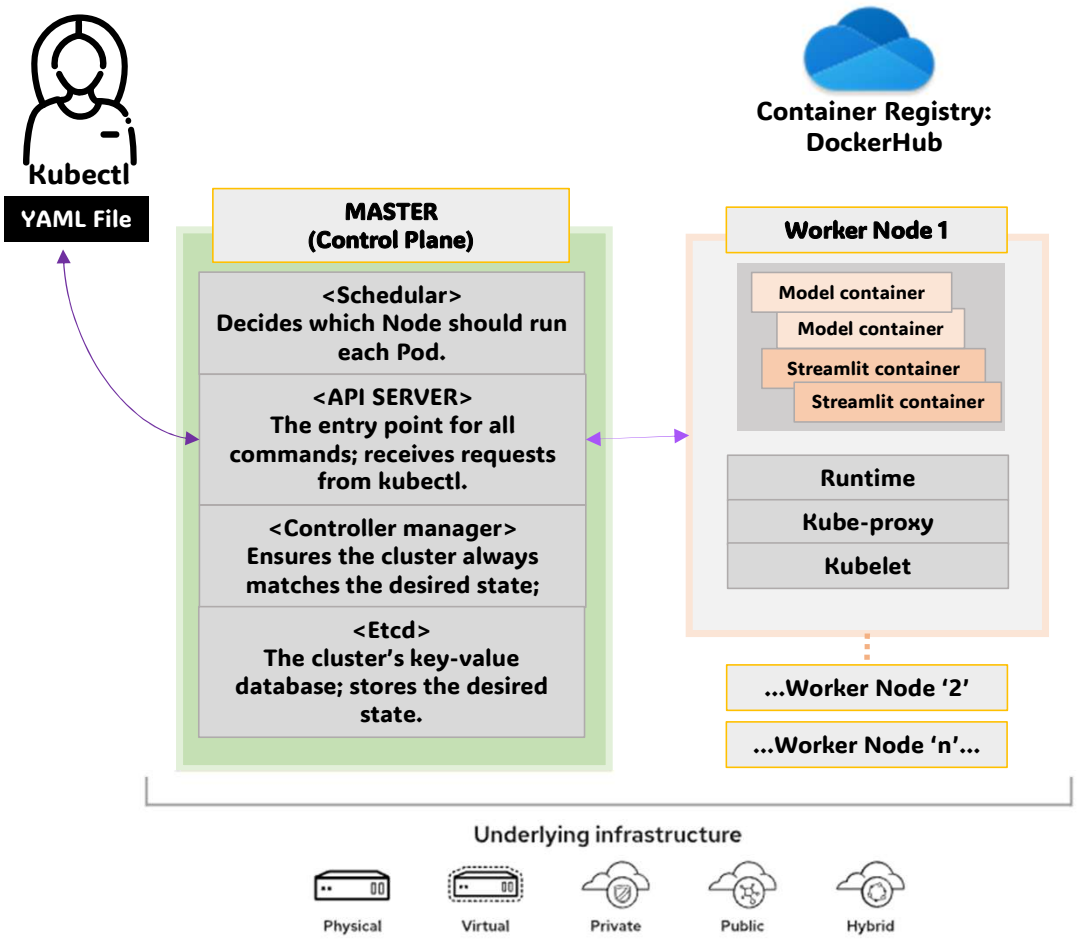
- Network administrator that provides a fixed access point
- Handles traffic distribution (Load Balancing)
- Ensures Streamlit UI can talk to the Model Pods reliably

Kubernetes Architecture



Production Deployment Status

Cluster Overview




```
honor@LAPTOP-NG18QTQ5 MINGW64 ~/spicedAcademy/mlops/bootcamp/mlops_house-price-predictor (main)
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
kube-ops-view-7d756f7b94-jxpq6      1/1     Running   6 (8h ago)    12d
model-6656fc88c7-dd485              1/1     Running   6 (8h ago)    11d
model-6656fc88c7-h9l62              1/1     Running   6 (8h ago)    11d
streamlit-cb664c6d7-w4ldn           1/1     Running   6 (8h ago)    11d
streamlit-cb664c6d7-x9n8c           1/1     Running   6 (8h ago)    11d
```

FastAPI Model Server

Two model replicas are running, successfully providing the predictions predictions service.

- ✓ Fulfills requirements for load balancing and high availability availability

Streamlit UI

Two UI replicas are running, enabling the application to distribute user distribute user traffic effectively across instances.

- ✓ Ensures consistent user experience experience

Resources and Reliability in Action

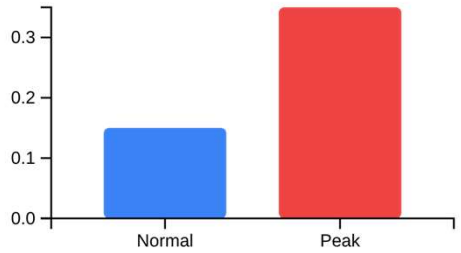
- Load Distribution(Service): Spreads incoming traffic evenly across 2 Pods.
- Self-Healing(by Controller Manager): If FastAPI Pod #1 crashes, Kubernetes automatically creates a new Pod.
- Autoscaling(yaml): During a traffic spike, Pods scale from 2 to 4 automatically.

Grafana Monitoring Dashboard

Real-time Performance Insights

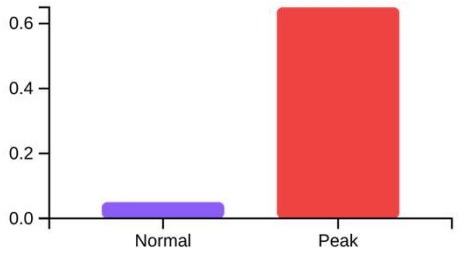
Prometheus + Grafana provides real-time insights into our House Price Prediction service, enabling us to identify performance bottlenecks.

Request Rate



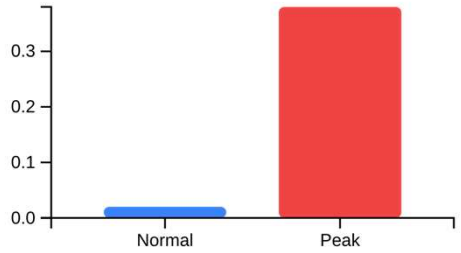
/metrics:	No change (0.1–0.2)
/predict:	Surge (0.1 to ~0.4)

Latency (95th Percentile)

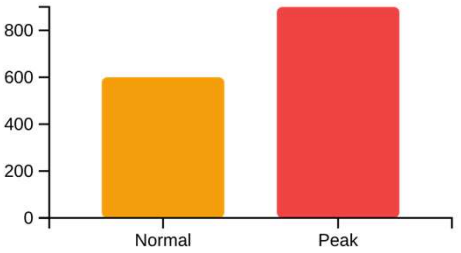


/metrics:	No change (~0)
/predict:	Surge (0.4 to 0.9)

Request Size



Response Size



Grafana Dashboard

Peak Load Analysis

- A sudden influx of requests, increasing system load
- Response time increased to nearly 1 second
- Response data size increased by approximately 50%

During peak activity, the /predict endpoint experiences: latency failure during traffic spikes