Contents

Fox ML Infrastructure – Client Onboarding Guide

Document Hierarchy: This document is provided for guidance only. In case of any conflict, ‘COMMERCIAL_LICENSE controls for free use.

> **Commercial Licensees Only:** This document applies only to customers with a valid commercial license. free users are not eligible for the policies, services, or terms described here.
>
> **No Rights Granted:** This document does not grant any license rights. All rights and restrictions are defined solely in `COMMERCIAL_LICENSE.md`.

This guide helps commercial licensees get started with Fox ML Infrastructure.
It covers setup, configuration, integration, and how to request custom features.

---

1. Prerequisites

1.1 Required Dependencies

System Requirements: - **OS**: Linux (Arch Linux, Ubuntu 22.04+, or similar distributions) - Tested on: Arch Linux (kernel 6.17+) - Build tools: GCC 11+ (provided via conda-forge) - **Python**: 3.10 (as specified in `environment.yml`) - **Git**: For repository access - **Storage**: Sufficient disk space for datasets and models

Python Dependencies: - Core dependencies are listed in `requirements.txt` (included in repository) - Additional dependencies may be required based on model types used

Hardware Requirements:

[WARNING] **Important**: FoxML Core requires significant computational resources. This is **not suitable for laptop/desktop deployment** for production workloads.

Minimum (Development/Testing Only): - CPU: Multi-core processor (8+ cores) - RAM: 16 GB minimum (32 GB recommended for development) - Storage: 50 GB+ free space - GPU: Optional (CUDA 11.8+ if using GPU)

Production (Recommended): - CPU: Multi-core processor (16+ cores, 32+ cores for large workloads) - RAM: 64 GB minimum (128 GB+ recommended) - Storage: 500 GB+ SSD (1 TB+ for large datasets) - GPU: Recommended (11 GB+ VRAM for optimal performance)

Verified Stable Range: Up to **100 GB RAM** (tested and verified stable in continuous operation) **Targeted Capacity**: **1 TB+ RAM** (enterprise deployment, requires production hardware verification)

See System Requirements for complete specifications.

1.2 Access Setup

Repository Access: - Commercial licensees receive access to private GitHub repositories - Access is granted via GitHub organization or repository-level permissions - See `LEGAL/ENTERPRISE_DELIVERY.md` for repository structure details

Support Access: - Support email: jenn.lewis5789@gmail.com - Support tier determines response times (see `LEGAL/SUPPORT_POLICY.md`)

---

2. Initial Setup

2.1 Repository Cloning

Enterprise Base Repository:

```
git clone https://github.com/Fox-ML-infrastructure/fox-v1-infra-enterprise.git
cd fox-v1-infra-enterprise
git checkout v1.0.0  # or latest version tag
```

Client-Specific Repository (if applicable):

```
git clone https://github.com/Fox-ML-infrastructure/client-<your-org>-fox-infra.git
cd client-<your-org>-fox-infra
```

2.2 Environment Setup

Python Virtual Environment:

```
python3 -m venv venv
source venv/bin/activate  # On Linux/macOS
pip install -r requirements.txt
```

Configuration: - Copy example configuration files to your working directory - Review configuration options in `config/` directory - See `docs/01_tutorials/configuration/CONFIG_BASICS.md` for configuration details

---

3. Directory Structure

3.1 Expected Structure

Core Directories:

```
fox-v1-infra/
+-- CONFIG/              # Configuration files
+-- DATA_PROCESSING/     # Data processing pipelines
+-- TRAINING/            # Model training framework
+-- docs/                # Documentation
+-- config/              # Runtime configuration
+-- tests/               # Test suite
```

Client-Specific Structure:

```
client-<your-org>-fox-infra/
+-- config/              # Client-specific configurations
+-- custom/              # Custom modules and features
+-- deployments/        # Deployment recipes
+-- strategies/         # Client-specific strategies (if applicable)
```

3.2 Data Directory

Recommended Data Structure:

```
data/
+-- raw/                # Raw market data
+-- processed/          # Processed features
+-- models/             # Trained models
+-- results/            # Backtest results and outputs
```

---

4. Market Data Integration

4.1 Data Sources

Supported Data Sources: - yfinance (default, for equity data) - Custom data sources (via data adapters) - Client-provided data files

4.2 Configuring Data Sources

Configuration Example:

```yaml
data:
  source: yfinance
  auto_adjust: false
  cache: true
  tickers: [SPY, QQQ, IWM]
```

Custom Data Integration: - Implement data adapter following the existing adapter pattern - See `DATA_PROCESSING/` for examples - Contact support for integration assistance

---

5. Model Configuration

5.1 Available Models

Model Types: - LightGBM, XGBoost (gradient boosting) - MLP, LSTM, Transformer (deep learning) - Ensemble models - Multi-task models - Probabilistic models (NGBoost, Quantile regression)

See `docs/02_reference/models/` for complete model documentation.

5.2 Configuring Models

Configuration Example:

```yaml
model:
  type: lightgbm
  variant: conservative
  params:
    n_estimators: 100
    learning_rate: 0.05
```

Configuration Files: - Model configs are in `CONFIG/` directory - Variants available: `conservative`, `balanced`, `aggressive` - See `docs/01_tutorials/configuration/ADVANCED_CONFIG.md` for advanced configuration

6. Running Your First Pipeline

6.1 Basic Workflow

Step 1: Data Processing

```
python -m DATA_PROCESSING.pipelines.process_data \
    --input data/raw \
    --output data/processed \
    --config config/data_config.yaml
```

Step 2: Feature Engineering

```
python -m DATA_PROCESSING.features.build_features \
    --input data/processed \
    --output data/features \
    --config config/features_config.yaml
```

Step 3: Model Training

```
python -m TRAINING.train \
    --data data/features \
    --model lightgbm \
    --output models/ \
    --config config/model_config.yaml
```

6.2 Documentation

Tutorials: - `docs/01_tutorials/pipelines/FIRST_PIPELINE_RUN.md` – First pipeline walkthrough - `docs/01_tutorials/training/BASIC_TRAINING.md` – Training basics - `docs/00_executive/GETTING_STARTED.md` – Quick start guide

---

7. Custom Features & Integration

7.1 Requesting Custom Features

Process: 1. **Contact support** – Email jenn.lewis5789@gmail.com with feature request 2. **Scoping** – Feature request is evaluated and scoped 3. **Statement of Work (SOW)** – Custom features require a separate SOW for enterprise licensees 4. **Development** – Feature is developed in client-specific repository 5. **Delivery** – Feature is delivered via client repository update

Contact jenn.lewis5789@gmail.com for custom development pricing.

7.2 Integration with Existing Systems

Integration Support: - **Architecture review** – Enterprise/Premium support includes pre-purchase architectural discussions - **Integration guidance** – Support can provide guidance on integrating with client systems - **Custom adapters** – Custom data adapters or integrations can be developed via SOW

---

8. Receiving Updates

8.1 Version Updates

Update Process: 1. **Check release notes** – Review `CHANGELOG_ENTERPRISE.md` for changes 2. **Review migration notes** – Check for breaking changes or migration requirements 3. **Update version tag** – Update your repository to the new version tag 4. **Test in development** – Test updates in a development environment first 5. **Update production** – Update production after successful testing

Example:

```
# In enterprise base repository
git fetch
git checkout v1.1.0  # Update to new version

# In client repository (if applicable)
git pull origin main  # Pull any client-specific updates
```

See `LEGAL/RELEASE_POLICY.md` for versioning strategy and update recommendations.

8.2 Update Frequency

Recommended Schedule: - **Patch releases** – Update within 1-2 weeks (especially security patches) - **Minor releases** – Update within 1-2 months (test in development first) - **Major releases** – Update when ready (plan migration, test thoroughly)

---

9. Support & Resources

9.1 Support Channels

Email Support: - jenn.lewis5789@gmail.com - Response times depend on support tier (see `LEGAL/SUPPORT_POLICY.md`)

Documentation: - `docs/` – Complete documentation hierarchy - `docs/INDEX.md` – Documentation navigation - `docs/00_executive/GETTING_STARTED.md` – Quick start guide

9.2 Support Tiers

Standard Support (Included): - Email support - 72-hour response time - Documentation access

Business Support (Add-on): - 24-hour response time - Priority bug-fix handling - Pricing: Contact for quote

Enterprise Support (Add-on): - Same-business-day response - Scheduled support calls - Priority engineering resources - Pricing: $60,000-$120,000/month (see `LEGAL/SUBSCRIPTIONS.md` for complete pricing tiers)

Premium Support (Add-on): - White-glove service - Highest priority engineering - Flexible support scheduling - Pricing: Custom quote

See `LEGAL/SUPPORT_POLICY.md` for complete support tier details and `LEGAL/SUBSCRIPTIONS.md` for pricing.

---

10. Best Practices

10.1 Configuration Management

- **Version control** – Keep configurations in version control
- **Environment-specific** – Use separate configs for development, staging, production
- **Secrets management** – Never commit secrets; use environment variables or secure vaults

10.2 Testing

- **Test in development** – Always test updates in a development environment first
- **Backup before updates** – Backup configurations and models before major updates
- **Version pinning** – Pin dependency versions for reproducible builds

10.3 Monitoring

- **Logging** – Review logs regularly for errors or warnings
- **Performance monitoring** – Monitor model performance and pipeline execution times
- **Resource usage** – Monitor CPU, memory, and GPU usage

---

11. Troubleshooting

11.1 Common Issues

Installation Issues: - Check Python version (3.9+ required) - Verify all dependencies are installed - Check system requirements

Configuration Issues: - Review configuration file syntax (YAML) - Check file paths and permissions - Verify data source connectivity

Performance Issues: - Check system resources (CPU, RAM, GPU) - Review model configuration parameters - Consider data preprocessing optimizations

11.2 Getting Help

Support Process: 1. **Check documentation** – Review relevant documentation first 2. **Gather information** – Collect logs, configs, and error messages 3. **Contact support** – Email support with detailed information 4. **Provide context** – Include version, environment, and steps to reproduce

---

12. Next Steps

12.1 Recommended Reading

1. **Getting Started** – `docs/00_executive/GETTING_STARTED.md`
2. **Architecture Overview** – `docs/ARCHITECTURE.md`

3. **Configuration Basics** – `docs/01_tutorials/configuration/CONFIG_BASICS.md`
4. **First Pipeline Run** – `docs/01_tutorials/pipelines/FIRST_PIPELINE_RUN.md`

12.2 Advanced Topics

- **Feature Engineering** – `docs/01_tutorials/pipelines/FEATURE_ENGINEERING_TUTORIAL.md`
- **Model Training** – `docs/01_tutorials/training/BASIC_TRAINING.md`
- **Model Integration** – Models can be integrated with external systems and applications

---

Contact

For onboarding assistance or questions:

Jennifer Lewis
Fox ML Infrastructure LLC
Email: **jenn.lewis5789@gmail.com**
Subject: *Onboarding Inquiry – Fox ML Infrastructure*

Related Documents

- `LEGAL/ENTERPRISE_DELIVERY.md` – Repository structure and delivery model
- `LEGAL/SUPPORT_POLICY.md` – Support tiers and response times
- `LEGAL/RELEASE_POLICY.md` – Versioning and update policies
- `LEGAL/SECURITY.md` – Security practices and data handling
- `docs/INDEX.md` – Complete documentation navigation