
✅ Chameleon KVM: End-to-End MLOps Pipeline Workflow

This guide walks through setting up a scalable MLOps system for chest X-ray diagnosis using Docker containers, persistent storage, and a feedback loop, deployed on a manually configured Chameleon KVM with assigned public IP.

---

1. 🔐 Access KVM Instance

Start by SSH-ing into your KVM instance:

```
ssh -i ~/.ssh/id_rsa_chameleon_project44 cc@<KVM_IP>
```

---

2. 📁 Clone the Project Repository

```
git clone https://github.com/Shrey12202/MedAI-Scalable-Diagnosis-with-Machine-Learning.git
```

---

3. 🐳 Launch Core Infrastructure Containers

Start services: **MLflow**, **Grafana**, **Prometheus**, **MinIO**, and **Postgres**.

```
cd MedAI-Scalable-Diagnosis-with-Machine-Learning/CheXspert\ MLOps/kvm\ container
```

```
docker compose -f docker-compose-mlflow-minio-grafana-prometheus-postgres up -d
```

Ports Used:

Service	Port
MLflow	8000
Grafana	3000
Prometheus	8001
MinIO	9000
Postgres	9001

---

4. 🚀 Launch Inference & Feedback Services

Start **FastAPI**, **Label Studio**, and the **feedback transfer** component:

```
cd MedAI-Scalable-Diagnosis-with-Machine-Learning/CheXspert\ MLOps/serving/docker


docker compose -f eval-loop-chi/docker/docker-compose-production.yaml up -d

docker compose -f eval-loop-chi-main/docker/docker-compose-labelstudio.yaml up -d


docker build -t user-feedback-transfer .

docker run --rm user-feedback-transfer
```

Service	**Port**
FastAPI	5000
Label Studio	8080

---

5. 💾 Mount Persistent Storage (CHI@TACC)

On a node with access to node-persist, run:

```
curl https://rclone.org/install.sh | sudo bash

sudo sed -i '/^#user_allow_other/s/^#//' /etc/fuse.conf

mkdir -p ~/.config/rclone

nano ~/.config/rclone/rclone.conf
```

Paste into rclone.conf:

```
[chi_tacc]

type = swift

user_id = sdv2034

application_credential_id = 292824fe1f334db69bc98130063de7f1

application_credential_secret = <your-secret>
```

```
auth = https://chi.tacc.chameleoncloud.org:5000/v3
region = CHI@TACC
```

Then mount the object store:

```
sudo mkdir -p /mnt/object
sudo chown -R cc /mnt/object
sudo chgrp -R cc /mnt/object

rclone mount chi_tacc:object-persist-project44-1 /mnt/object \
  --read-only --allow-other --vfs-cache-mode=full \
  --dir-cache-time=72h --swift-fetch-until-empty-page --daemon
```

---

6. 🧠 Model Training

Prepare Docker and run the training pipeline:

```
sudo apt update && sudo apt install docker.io
sudo systemctl start docker && sudo systemctl enable docker

cd MedAI-Scalable-Diagnosis-with-Machine-Learning/CheXspert\ MLOps/train

# Optional: Add user to Docker group
sudo usermod -aG docker $USER && newgrp docker

docker build -t chexpert-train .

docker run --shm-size=16g \
  -v $(pwd):/workspace \
  -v /mnt/object:/mnt/object \
  -v /mnt/persistent:/mnt/persistent \
  chexpert-train
```

---

7. 📊 Monitoring, Inference & Feedback

- **MLflow UI (metrics, models):** http://129.114.27.181:8000
- **MinIO (artifacts):** http://129.114.27.181:9000
- **FastAPI (inference):** http://129.114.27.181:5000
- **Label Studio (review/correct):** http://129.114.27.181:8080/projects/
- **Grafana (data drift & metrics):** http://129.114.27.181:3000

---

8. 🔁 Feedback Transfer & Retraining Loop

After running inference:

```
cd MedAI-Scalable-Diagnosis-with-Machine-Learning/CheXspert\ MLOps/serving/workspace

docker build -t user-feedback-transfer .

docker run --rm user-feedback-transfer
```

This moves flagged predictions into Label Studio for manual correction. Once corrected, retraining can begin again from step 6, using:

- Original data from /mnt/object
- Simulated feedback data stored in MinIO
