Complete Setup Instructions

📁 Required Files

Create these files in your project directory:

```
your-project/
├── main.py                 # FastAPI app with Prometheus integration
├── requirements.txt        # Python dependencies
├── Dockerfile              # Container build instructions
├── keda-http-scaler.yaml   # Kubernetes and KEDA configuration
├── build.sh                # Build script (make executable)
├── deploy.sh               # Deploy script (make executable)
├── test-integration.sh     # Testing script (make executable)
└── pod_predictor.pkl       # Your trained model file
```

🔧 Configuration Steps

1. Update Configuration

Edit `keda-http-scaler.yaml` and update these values:

yaml

```yaml
# In ConfigMap section
data:
  PROMETHEUS_URL: "http://prometheus-server.monitoring.svc.cluster.local:80"  # Your Prometheus URL
  TARGET_DEPLOYMENT: "your-actual-workload-name"  # Your workload to scale
  TARGET_NAMESPACE: "default"  # Namespace of your workload

# In ScaledObject section
spec:
  scaleTargetRef:
    name: your-actual-workload-deployment  # Same as TARGET_DEPLOYMENT
```

2. Update Build Script

Edit `build.sh` and set your registry:

bash

```bash
REGISTRY="your-registry.com"  # Replace with your Docker registry
```

3. Add Your Model File

Place your trained model file as `pod_predictor.pkl` in the project directory.

🚀 Deployment Process

Step 1: Make Scripts Executable

bash

```
chmod +x build.sh deploy.sh test-integration.sh
```

Step 2: Build and Push Image

bash

```
./build.sh
```

Step 3: Deploy to Kubernetes

bash

```
./deploy.sh
```

Step 4: Test the Integration

bash

```
./test-integration.sh
```

🔍 Verification

Check if everything is working:

1. **Pod Predictor Service**

   bash

   ```
   kubectl get pods -l app=pod-predictor
   kubectl logs -l app=pod-predictor
   ```

2. **KEDA Scaling**

   bash

   ```
   kubectl get scaledobject ml-predictor-scaler
   kubectl describe scaledobject ml-predictor-scaler
   ```

3. **Test Prediction**

   bash

   ```
   kubectl port-forward svc/pod-predictor-service 8080:80
   curl http://localhost:8080/predict-from-prometheus
   ```

🔧 Troubleshooting

Common Issues:

1. **Model not found**
   - Ensure `pod_predictor.pkl` is in your project directory
   - Check ConfigMap: `kubectl get configmap pod-predictor-model`
2. **Prometheus connection failed**
   - Verify Prometheus URL in ConfigMap
   - Test connectivity: `kubectl exec deployment/pod-predictor -- curl http://prometheus-server.monitoring.svc.cluster.local:80/api/v1/query?query=up`
3. **KEDA not scaling**
   - Check ScaledObject status: `kubectl describe scaledobject ml-predictor-scaler`
   - Check KEDA operator logs: `kubectl logs -n keda -l app=keda-operator`
4. **Container registry issues**
   - Ensure you're logged in: `docker login your-registry.com`
   - Update image name in `keda-http-scaler.yaml`

🎯 Final Flow

Once deployed, this is what happens:

1. **Pod Predictor** pulls metrics from Prometheus every 30 seconds
2. **Model** processes metrics and predicts optimal pod count
3. **KEDA** queries prediction service every 30 seconds
4. **Kubernetes** scales your workload based on predictions

Your ML model is now automatically scaling your Kubernetes workloads! 🎉