

# IBM Cloud Deployment Guide for Peoria Platform

---

## Quick Start Deployment

---

### 1. Prerequisites Checklist

- ☐ IBM Cloud account with billing enabled
- ☐ IBM Cloud CLI installed and configured
- ☐ Docker installed locally
- ☐ kubectl CLI tool installed
- ☐ Watson X services provisioned

## 2. One-Click Setup Script

```
#!/bin/bash
# deploy-to-ibm-cloud.sh

set -e

# Variables
PROJECT_NAME="peoria-agricultural-platform"
NAMESPACE="peoria-namespace"
CLUSTER_NAME="peoria-cluster"
REGION="us-south"

echo "🚀 Starting IBM Cloud deployment for Peoria Platform..."

# 1. Login to IBM Cloud
echo "📋 Logging into IBM Cloud..."
ibmcloud login --sso

# 2. Target the region
echo "🌐 Targeting region: $REGION"
ibmcloud target -r $REGION

# 3. Create namespace in container registry
echo "📦 Creating container registry namespace..."
ibmcloud cr namespace-add $NAMESPACE || true

# 4. Build and push Docker image
echo "🔧 Building Docker image..."
docker build -t $PROJECT_NAME .
docker tag $PROJECT_NAME us.icr.io/$NAMESPACE/$PROJECT_NAME:latest

echo "📦 Pushing to IBM Cloud Container Registry..."
ibmcloud cr login
docker push us.icr.io/$NAMESPACE/$PROJECT_NAME:latest

# 5. Create Kubernetes cluster (if not exists)
echo "🏠 Setting up Kubernetes cluster..."
ibmcloud ks cluster create vpc-gen2 \
  --name $CLUSTER_NAME \
  --zone ${REGION}-1 \
  --version 1.28 \
  --flavor bx2.4x16 \
  --workers 3 \
  --vpc-id $(ibmcloud is vpcs --output json | jq -r '[0].id') \
  --subnet-id $(ibmcloud is subnets --output json | jq -r '[0].id') || true

# Wait for cluster to be ready
echo "⌚ Waiting for cluster to be ready..."
ibmcloud ks cluster wait --cluster $CLUSTER_NAME

# 6. Configure kubectl
echo "🔑 Configuring kubectl..."
ibmcloud ks cluster config --cluster $CLUSTER_NAME

# 7. Create secrets
echo "🔒 Creating Kubernetes secrets..."
kubectl create namespace peoria-platform || true

kubectl create secret generic peoria-secrets \
  --namespace=peoria-platform \
  --from-literal=database-url="$DATABASE_URL" \
  --from-literal=nextauth-secret="$NEXTAUTH_SECRET" \
  --dry-run=client -o yaml | kubectl apply -f -
```

```

kubectl create secret generic watson-secrets \
  --namespace=peoria-platform \
  --from-literal=ai-api-key="$WATSON_AI_API_KEY" \
  --from-literal=ai-url="$WATSON_AI_URL" \
  --from-literal=ai-project-id="$WATSON_AI_PROJECT_ID" \
  --from-literal=orchestrate-api-key="$WATSON_ORCHESTRATE_API_KEY" \
  --from-literal=orchestrate-url="$WATSON_ORCHESTRATE_URL" \
  --from-literal=orchestrate-environment-id="$WATSON_ORCHESTRATE_ENVIRONMENT_ID" \
  --from-literal=assistant-api-key="$WATSON_ASSISTANT_API_KEY" \
  --from-literal=assistant-url="$WATSON_ASSISTANT_URL" \
  --from-literal=assistant-id="$WATSON_ASSISTANT_ID" \
  --dry-run=client -o yaml | kubectl apply -f -

# 8. Deploy application
echo "🚢 Deploying application..."
kubectl apply -f k8s/ --namespace=peoria-platform

# 9. Wait for deployment
echo "⌚ Waiting for deployment to be ready..."
kubectl rollout status deployment/peoria-agricultural-platform --namespace=peoria-platform

# 10. Get application URL
echo "🎉 Deployment complete!"
echo "Application URL: https://peoria.onticworks.io"
echo "Kubernetes Dashboard: $(ibmcloud ks cluster get --cluster $CLUSTER_NAME --output json | jq -r '.masterURL')"

echo "✅ Peoria Platform successfully deployed to IBM Cloud!"

```

### 3. Environment Variables Setup

Create a `.env.ibm-cloud` file with your IBM Cloud specific variables:

```
# IBM Cloud Database for PostgreSQL
DATABASE_URL="postgresql://ibm_cloud_user:password@host:port/peoria_db?sslmode=require"

# Watson X.ai Configuration
WATSON_AI_API_KEY="your_watson_x_ai_api_key"
WATSON_AI_URL="https://us-south.ml.cloud.ibm.com"
WATSON_AI_PROJECT_ID="your_watson_x_project_id"

# Watson Orchestrate Configuration
WATSON_ORCHESTRATE_API_KEY="your_orchestrate_api_key"
WATSON_ORCHESTRATE_URL="https://us-south.wa.cloud.ibm.com"
WATSON_ORCHESTRATE_ENVIRONMENT_ID="your_orchestrate_environment_id"

# Watson Assistant Configuration
WATSON_ASSISTANT_API_KEY="your_assistant_api_key"
WATSON_ASSISTANT_URL="https://api.us-south.assistant.watson.cloud.ibm.com"
WATSON_ASSISTANT_ID="your_assistant_id"

# IBM Cloud Configuration
IBM_CLOUD_API_KEY="your_ibm_cloud_api_key"
IBM_CLOUD_REGION="us-south"

# Application Configuration
NEXTAUTH_URL="https://peoria.onticworks.io"
NEXTAUTH_SECRET="your_secure_nextauth_secret"
NODE_ENV="production"
```

## Watson X Services Configuration

### 1. Watson X.ai Setup

```
# Create Watson Machine Learning service
ibmcloud resource service-instance-create watson-ml-peoria \
  pm-20 lite us-south

# Get service credentials
ibmcloud resource service-key-create watson-ml-key Manager \
  --instance-name watson-ml-peoria

# Deploy AI models for agricultural analysis
curl -X POST \
  "$WATSON_AI_URL/ml/v4/models" \
  -H "Authorization: Bearer $(ibmcloud iam oauth-tokens --output json | jq -r '.iam_token')" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "environmental-analysis-model",
    "type": "scikit-learn_1.0",
    "software_spec": {
      "name": "runtime-22.1-py3.9"
    },
    "project_id": "'$WATSON_AI_PROJECT_ID'"
  }'
```

## 2. Watson Orchestrate Setup

```
# Create Watson Orchestrate service
ibmcloud resource service-instance-create watson-orchestrate-peoria \
  conversational-ai standard us-south

# Configure agricultural workflows
curl -X POST \
  "$WATSON_ORCHESTRATE_URL/v2/workflows" \
  -H "Authorization: Bearer $(ibmcloud iam oauth-tokens --output json | jq -r '.iam_token')" \
  -H "Content-Type: application/json" \
  -d @agricultural-workflow-template.json
```

## 3. Watson Assistant Setup

```
# Create Watson Assistant service
ibmcloud resource service-instance-create watson-assistant-peoria \
  conversation standard us-south

# Import agricultural knowledge base
curl -X POST \
  "$WATSON_ASSISTANT_URL/v2/assistants/$WATSON_ASSISTANT_ID/sessions" \
  -H "Authorization: Bearer $(ibmcloud iam oauth-tokens --output json | jq -r '.iam_token')" \
  -H "Content-Type: application/json"
```

# Performance and Scaling Configuration

---

## 1. Auto-scaling Configuration

```
# k8s/hpa.yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: peoria-platform-hpa
  namespace: peoria-platform
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: peoria-agricultural-platform
  minReplicas: 3
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 70
  - type: Resource
    resource:
      name: memory
      target:
        type: Utilization
        averageUtilization: 80
  behavior:
    scaleUp:
      stabilizationWindowSeconds: 300
      policies:
      - type: Percent
        value: 100
        periodSeconds: 15
    scaleDown:
      stabilizationWindowSeconds: 300
      policies:
      - type: Percent
        value: 10
        periodSeconds: 60
```

## 2. IBM Cloud Load Balancer

```
# k8s/load-balancer.yaml
apiVersion: v1
kind: Service
metadata:
  name: peoria-load-balancer
  namespace: peoria-platform
  annotations:
    service.kubernetes.io/ibm-load-balancer-cloud-provider-name: "ibm"
    service.kubernetes.io/ibm-load-balancer-cloud-provider-zone: "us-south-1"
spec:
  type: LoadBalancer
  selector:
    app: peoria-platform
  ports:
  - port: 443
    targetPort: 3000
    protocol: TCP
  loadBalancerSourceRanges:
  - 0.0.0.0/0
```

## Monitoring and Logging

### 1. IBM Cloud Monitoring

```
# Create monitoring service
ibmcloud resource service-instance-create peoria-monitoring \
  sysdig-monitor graduated-tier us-south

# Configure monitoring agent
curl -sL https://raw.githubusercontent.com/draios/sysdig-cloud-scripts/master/
agent_deploy/IBMCloud-Kubernetes-Service/install-agent-k8s.sh | bash -s -- -a $SYS-
DIG_ACCESS_KEY -c $SYSDIG_COLLECTOR -ac 'sysdig_capture_enabled: false'
```

### 2. IBM Cloud Logging

```
# Create logging service
ibmcloud resource service-instance-create peoria-logging \
  logdna standard us-south

# Configure logging agent
kubectl create secret generic logdna-agent-key --from-literal=logdna-agent-key=$LOG-
DNA_INGESTION_KEY
kubectl apply -f https://raw.githubusercontent.com/logdna/logdna-agent/master/k8s/
agent-resources.yaml
```



## Security Configuration

---

### 1. IBM Cloud Certificate Manager

```
# Create certificate manager instance
ibmcloud resource service-instance-create peoria-cert-manager \
  cloudcerts free us-south

# Order SSL certificate
ibmcloud cm certificate-order --name peoria-ssl --domains peoria.onticworks.io
```

### 2. IBM Cloud Internet Services (Cloudflare)

```
# Create Internet Services instance
ibmcloud resource service-instance-create peoria-cis \
  internet-svcs standard global

# Configure DNS and security settings
ibmcloud cis domain-add peoria-cis onticworks.io
ibmcloud cis dns-record-create peoria-cis onticworks.io --name peoria --type A --con-
tent $LOAD_BALANCER_IP
```

## Cost Optimization

---

### 1. Resource Quotas

```
# k8s/resource-quota.yaml
apiVersion: v1
kind: ResourceQuota
metadata:
  name: peoria-resource-quota
  namespace: peoria-platform
spec:
  hard:
    requests.cpu: "4"
    requests.memory: 8Gi
    limits.cpu: "8"
    limits.memory: 16Gi
    pods: "10"
    services: "5"
```

## 2. Watson X Usage Monitoring

```
// lib/watson-usage-monitor.js
class WatsonUsageMonitor {
  async trackUsage(service, operation, tokens) {
    await fetch('/api/usage/watson', {
      method: 'POST',
      body: JSON.stringify({
        service,
        operation,
        tokens,
        timestamp: new Date().toISOString()
      })
    });
  }
}
```

## Backup and Disaster Recovery

### 1. Database Backup

```
# Schedule daily backups
ibmcloud resource service-instance-update peoria-postgresql \
  --parameters '{"backup_encryption_key_crn": "crn:v1:bluemix:public:kms:us-south:a/
account-id:key-protect-instance:key:key-id"}'
```

### 2. Application State Backup

```
# k8s/backup-job.yaml
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: peoria-backup
  namespace: peoria-platform
spec:
  schedule: "0 2 * * *" # Daily at 2 AM
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: backup
              image: us.icr.io/peoria-namespace/backup-tool:latest
              env:
                - name: DATABASE_URL
                  valueFrom:
                    secretKeyRef:
                      name: peoria-secrets
                      key: database-url
              restartPolicy: OnFailure
```

# Troubleshooting

---

## Common Issues

### 1. Watson X Service Timeout

```
```bash
# Check service status
kubectl logs -l app=peoria-platform -namespace=peoria-platform | grep "Watson"

# Restart pods
kubectl rollout restart deployment/peoria-agricultural-platform -namespace=peoria-platform
```
```

### 1. Database Connection Issues

```
bash
# Test database connectivity
kubectl run -it --rm --image=postgres:13 --restart=Never test-db -- psql $DATABASE_URL
```

### 2. Container Registry Issues

```
```bash
# Re-authenticate with container registry
ibmcloud cr login

# Check image availability
ibmcloud cr images --restrict peoria-namespace
```
```

## Support Resources

- **IBM Cloud Support:** <https://cloud.ibm.com/unifiedsupport> (<https://cloud.ibm.com/unifiedsupport>)
- **Watson X Documentation:** <https://www.ibm.com/watson> (<https://www.ibm.com/watson>)
- **Kubernetes Troubleshooting:** <https://kubernetes.io/docs/tasks/debug-application-cluster/> (<https://kubernetes.io/docs/tasks/debug-application-cluster/>)

## Cost Estimation

### Monthly Cost Breakdown (USD)

| Service            | Configuration             | Estimated Cost      |
|--------------------|---------------------------|---------------------|
| IKS Cluster        | 3 x bx2.4x16 worker nodes | \$360               |
| Watson X.ai        | Standard plan             | \$150               |
| Watson Orchestrate | Standard plan             | \$100               |
| Watson Assistant   | Standard plan             | \$80                |
| PostgreSQL         | Standard plan             | \$60                |
| Container Registry | 10GB storage              | \$5                 |
| Load Balancer      | Standard                  | \$15                |
| <b>Total</b>       |                           | <b>~\$770/month</b> |

### Cost Optimization Tips

1. Use reserved capacity for predictable workloads
2. Implement auto-scaling to reduce idle resource costs
3. Monitor Watson X usage and optimize API calls
4. Use IBM Cloud cost tracking and budgets
5. Consider dev/test environment sleeping during off-hours

**Ready to deploy?** Run the deployment script:

```
chmod +x deploy-to-ibm-cloud.sh
./deploy-to-ibm-cloud.sh
```

The platform will be available at: <https://peoria.onticworks.io>