

# Bio-Quantum AI Trading Platform - Production Deployment Guide



## PRODUCTION DEPLOYMENT OVERVIEW

This guide provides step-by-step instructions for deploying the Bio-Quantum AI Trading Platform in a production environment with enterprise-grade security, monitoring, and reliability.

---



## PRE-DEPLOYMENT REQUIREMENTS

### System Requirements

- **Operating System:** Ubuntu 22.04 LTS or CentOS 8+
- **CPU:** Minimum 8 cores, Recommended 16+ cores
- **Memory:** Minimum 16GB RAM, Recommended 32GB+
- **Storage:** Minimum 100GB SSD, Recommended 500GB+ NVMe SSD
- **Network:** Stable internet connection with low latency to exchanges

### Software Dependencies

- **Python:** 3.11+
- **PostgreSQL:** 14+
- **Redis:** 6.2+
- **Nginx:** 1.20+ (for reverse proxy)
- **Docker:** 20.10+ (optional, for containerized deployment)
- **Git:** 2.30+

## Security Requirements

- **SSL/TLS Certificates:** Valid certificates for production domain
- **Firewall:** UFW or iptables configured
- **SSH:** Key-based authentication only
- **User Management:** Non-root service user created

## Exchange API Requirements

- **Binance API:** Production API keys with trading permissions
  - **Coinbase Pro API:** (Optional) Production API keys
  - **API Permissions:** Spot trading, account information, order management
- 

## STEP 1: SYSTEM PREPARATION

### 1.1 Update System

Bash

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y curl wget git unzip software-properties-common
```

### 1.2 Install Python 3.11

Bash

```
sudo add-apt-repository ppa:deadsnakes/ppa -y
sudo apt update
sudo apt install -y python3.11 python3.11-venv python3.11-dev
```

### 1.3 Install PostgreSQL

Bash

```
sudo apt install -y postgresql postgresql-contrib
sudo systemctl enable postgresql
sudo systemctl start postgresql
```

## 1.4 Install Redis

Bash

```
sudo apt install -y redis-server
sudo systemctl enable redis-server
sudo systemctl start redis-server
```

## 1.5 Install Nginx

Bash

```
sudo apt install -y nginx
sudo systemctl enable nginx
sudo systemctl start nginx
```

## 1.6 Create Service User

Bash

```
sudo useradd -r -s /bin/false -d /opt/bio-quantum-ai bio-quantum
sudo mkdir -p /opt/bio-quantum-ai
sudo chown bio-quantum:bio-quantum /opt/bio-quantum-ai
```



# STEP 2: APPLICATION DEPLOYMENT

## 2.1 Clone Repository

Bash

```
cd /opt/bio-quantum-ai
sudo -u bio-quantum git clone https://github.com/your-org/bio-quantum-ai.git
```

```
app
cd app
```

## 2.2 Set Up Virtual Environment

Bash

```
sudo -u bio-quantum python3.11 -m venv /opt/bio-quantum-ai/venv
sudo -u bio-quantum /opt/bio-quantum-ai/venv/bin/pip install --upgrade pip
sudo -u bio-quantum /opt/bio-quantum-ai/venv/bin/pip install -r
requirements.txt
```

## 2.3 Copy Configuration Files

Bash

```
sudo -u bio-quantum cp bio-quantum-deployment/config/production.yml /opt/bio-
quantum-ai/config/
sudo -u bio-quantum chmod 600 /opt/bio-quantum-ai/config/production.yml
```

## 2.4 Set Environment Variables

Bash

```
sudo -u bio-quantum tee /opt/bio-quantum-ai/.env > /dev/null <<EOF
# Database Configuration
DB_PASSWORD=your_secure_database_password_here
REDIS_PASSWORD=your_secure_redis_password_here

# Exchange API Keys
BINANCE_API_KEY=your_binance_api_key_here
BINANCE_API_SECRET=your_binance_api_secret_here

# Security
JWT_SECRET=$(openssl rand -base64 64)

# Email Configuration
SMTP_USERNAME=your_smtp_username_here
SMTP_PASSWORD=your_smtp_password_here

# Slack Integration
SLACK_WEBHOOK_URL=your_slack_webhook_url_here
```

```
# Optional: Coinbase Pro
COINBASE_API_KEY=your_coinbase_api_key_here
COINBASE_API_SECRET=your_coinbase_api_secret_here
COINBASE_PASSPHRASE=your_coinbase_passphrase_here
EOF

sudo chown bio-quantum:bio-quantum /opt/bio-quantum-ai/.env
sudo chmod 600 /opt/bio-quantum-ai/.env
```

## STEP 3: DATABASE SETUP

### 3.1 Configure PostgreSQL

Bash

```
sudo -u postgres psql <<EOF
CREATE DATABASE bio_quantum_prod;
CREATE USER bio_quantum WITH PASSWORD 'your_secure_database_password_here';
GRANT ALL PRIVILEGES ON DATABASE bio_quantum_prod TO bio_quantum;
ALTER USER bio_quantum CREATEDB;
\q
EOF
```

### 3.2 Configure PostgreSQL Security

Bash

```
sudo tee -a /etc/postgresql/14/main/postgresql.conf > /dev/null <<EOF
# Bio-Quantum AI Configuration
ssl = on
ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem'
ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key'
max_connections = 100
shared_buffers = 256MB
effective_cache_size = 1GB
work_mem = 4MB
maintenance_work_mem = 64MB
EOF
```

```
sudo systemctl restart postgresql
```

### 3.3 Run Database Migrations

Bash

```
cd /opt/bio-quantum-ai/app
sudo -u bio-quantum /opt/bio-quantum-ai/venv/bin/python manage.py migrate
```



## STEP 4: SECURITY CONFIGURATION

### 4.1 Generate SSL Certificates

Bash

```
# For production, use Let's Encrypt or your CA
sudo mkdir -p /opt/bio-quantum-ai/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:4096 \
    -keyout /opt/bio-quantum-ai/ssl/bio-quantum.key \
    -out /opt/bio-quantum-ai/ssl/bio-quantum.crt \
    -subj "/C=US/ST=State/L=City/O=Bio-Quantum-AI/CN=yourdomain.com"

sudo chown bio-quantum:bio-quantum /opt/bio-quantum-ai/ssl/*
sudo chmod 600 /opt/bio-quantum-ai/ssl/bio-quantum.key
sudo chmod 644 /opt/bio-quantum-ai/ssl/bio-quantum.crt
```

### 4.2 Configure Firewall

Bash

```
sudo ufw enable
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow 22/tcp      # SSH
sudo ufw allow 80/tcp      # HTTP
sudo ufw allow 443/tcp     # HTTPS
sudo ufw allow from 10.0.0.0/8 to any port 9090 # Monitoring
```

## 4.3 Secure Redis

Bash

```
sudo tee /etc/redis/redis-bio-quantum.conf > /dev/null <<EOF
port 6380
bind 127.0.0.1
protected-mode yes
requirepass your_secure_redis_password_here
maxmemory 2gb
maxmemory-policy allkeys-lru
save 900 1
save 300 10
save 60 10000
EOF

sudo systemctl restart redis-server
```



## STEP 5: SERVICE CONFIGURATION

### 5.1 Create Systemd Services

Run the deployment script to create all systemd services:

Bash

```
cd /opt/bio-quantum-ai/app/bio-quantum-deployment/scripts
chmod +x deploy-production.sh
./deploy-production.sh
```

### 5.2 Configure Nginx Reverse Proxy

Bash

```
sudo tee /etc/nginx/sites-available/bio-quantum-ai > /dev/null <<EOF
upstream bio_quantum_backend {
    server 127.0.0.1:8000;
    server 127.0.0.1:8001 backup;
}
}
```

```

server {
    listen 80;
    server_name yourdomain.com www.yourdomain.com;
    return 301 https://\${server_name}\${request_uri};
}

server {
    listen 443 ssl http2;
    server_name yourdomain.com www.yourdomain.com;

    ssl_certificate /opt/bio-quantum-ai/ssl/bio-quantum.crt;
    ssl_certificate_key /opt/bio-quantum-ai/ssl/bio-quantum.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers
ECDHE+AESGCM:ECDHE+CHACHA20:DHE+AESGCM:DHE+CHACHA20:!aNULL:!MD5:!DSS;
    ssl_prefer_server_ciphers off;

    location / {
        proxy_pass http://bio_quantum_backend;
        proxy_set_header Host \${host};
        proxy_set_header X-Real-IP \${remote_addr};
        proxy_set_header X-Forwarded-For \${proxy_add_x_forwarded_for};
        proxy_set_header X-Forwarded-Proto \${scheme};
        proxy_connect_timeout 30s;
        proxy_send_timeout 30s;
        proxy_read_timeout 30s;
    }

    location /metrics {
        proxy_pass http://127.0.0.1:9090;
        allow 10.0.0.0/8;
        deny all;
    }

    location /health {
        proxy_pass http://bio_quantum_backend/health;
        access_log off;
    }
}
EOF

```

```

sudo ln -s /etc/nginx/sites-available/bio-quantum-ai /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx

```





## STEP 6: MONITORING SETUP

### 6.1 Install Prometheus

Bash

```
cd /tmp
wget
https://github.com/prometheus/prometheus/releases/download/v2.40.0/prometheus-2.40.0.linux-amd64.tar.gz
tar xzf prometheus-2.40.0.linux-amd64.tar.gz
sudo mv prometheus-2.40.0.linux-amd64/prometheus /usr/local/bin/
sudo mv prometheus-2.40.0.linux-amd64/promtool /usr/local/bin/

sudo mkdir -p /etc/prometheus /var/lib/prometheus
sudo chown prometheus:prometheus /etc/prometheus /var/lib/prometheus
```

### 6.2 Configure Prometheus

Bash

```
sudo tee /etc/prometheus/prometheus.yml > /dev/null <<EOF
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'bio-quantum-ai'
    static_configs:
      - targets: ['localhost:9090']
    scrape_interval: 5s
    metrics_path: /metrics

  - job_name: 'node-exporter'
    static_configs:
      - targets: ['localhost:9100']
EOF
```

### 6.3 Install Grafana

Bash

```
sudo apt-get install -y software-properties-common
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable
main"
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
sudo apt-get update
sudo apt-get install -y grafana

sudo systemctl enable grafana-server
sudo systemctl start grafana-server
```

---

## STEP 7: DEPLOYMENT EXECUTION

### 7.1 Start All Services

Bash

```
sudo systemctl enable bio-quantum-api bio-quantum-execution bio-quantum-
knowledge bio-quantum-manager
sudo systemctl start bio-quantum-api bio-quantum-execution bio-quantum-
knowledge bio-quantum-manager
```

### 7.2 Verify Service Status

Bash

```
sudo systemctl status bio-quantum-*
```

### 7.3 Check Logs

Bash

```
sudo journalctl -u bio-quantum-api -f
sudo journalctl -u bio-quantum-execution -f
sudo journalctl -u bio-quantum-knowledge -f
sudo journalctl -u bio-quantum-manager -f
```

---

## STEP 8: HEALTH CHECKS

### 8.1 API Health Check

Bash

```
curl -k https://yourdomain.com/health
```

### 8.2 Database Connectivity

Bash

```
sudo -u bio-quantum psql -d bio_quantum_prod -c "SELECT 1;"
```

### 8.3 Redis Connectivity

Bash

```
redis-cli -p 6380 -a "your_secure_redis_password_here" ping
```

### 8.4 Exchange Connectivity

Bash

```
curl -X GET "https://api.binance.com/api/v3/ping"
```

---

## STEP 9: MONITORING AND ALERTING

### 9.1 Access Monitoring Dashboards

- **Grafana:** <https://yourdomain.com:3000> (admin/admin)
- **Prometheus:** <https://yourdomain.com:9090>

## 9.2 Configure Alerts

1. Set up email notifications in Grafana
2. Configure Slack webhooks for critical alerts
3. Set up SMS alerts for emergency situations

## 9.3 Key Metrics to Monitor

- **System Health:** CPU, Memory, Disk usage
  - **Application Performance:** Response times, error rates
  - **Trading Metrics:** Execution success rate, P&L, positions
  - **Security Events:** Failed logins, API errors
- 



## STEP 10: BACKUP AND RECOVERY

### 10.1 Configure Automated Backups

Bash

```
sudo crontab -e
# Add the following lines:
0 2 * * * /opt/bio-quantum-ai/scripts/backup-database.sh
0 */6 * * * /opt/bio-quantum-ai/scripts/backup-application.sh
```

### 10.2 Test Backup Restoration

Bash

```
# Test database backup restoration
sudo -u bio-quantum /opt/bio-quantum-ai/scripts/restore-database.sh
/path/to/backup.sql

# Test application backup restoration
```

```
sudo -u bio-quantum /opt/bio-quantum-ai/scripts/restore-application.sh  
/path/to/backup.tar.gz
```

---

## SECURITY BEST PRACTICES

### 10.1 Regular Security Updates

- Update system packages monthly
- Update Python dependencies quarterly
- Rotate API keys every 90 days
- Review access logs weekly

### 10.2 Access Control

- Use SSH key authentication only
- Implement VPN for administrative access
- Regular security audits
- Monitor for suspicious activities

### 10.3 Data Protection

- Encrypt all sensitive data at rest
  - Use TLS 1.3 for all communications
  - Regular penetration testing
  - Compliance with financial regulations
- 

## TROUBLESHOOTING

# Common Issues and Solutions

## Service Won't Start

Bash

```
# Check service status
sudo systemctl status bio-quantum-api

# Check logs
sudo journalctl -u bio-quantum-api -n 50

# Check configuration
sudo -u bio-quantum /opt/bio-quantum-ai/venv/bin/python -m trading_api_integration.main --check-config
```

## Database Connection Issues

Bash

```
# Check PostgreSQL status
sudo systemctl status postgresql

# Test connection
sudo -u bio-quantum psql -d bio_quantum_prod -c "SELECT version();"

# Check configuration
sudo cat /etc/postgresql/14/main/postgresql.conf | grep -E "listen_addresses|port"
```

## Exchange API Issues

Bash

```
# Test API connectivity
curl -X GET "https://api.binance.com/api/v3/time"

# Check API key permissions
curl -X GET "https://api.binance.com/api/v3/account" -H "X-MBX-APIKEY: your_api_key"

# Verify API key configuration
sudo -u bio-quantum grep -E "BINANCE_API" /opt/bio-quantum-ai/.env
```

## Performance Issues

Bash

```
# Check system resources
htop
df -h
free -m

# Check application metrics
curl -k https://yourdomain.com/metrics

# Analyze slow queries
sudo -u postgres psql -d bio_quantum_prod -c "SELECT query, mean_time FROM pg_stat_statements ORDER BY mean_time DESC LIMIT 10;"
```

## SUPPORT AND MAINTENANCE

### Emergency Contacts

- **System Administrator:** [admin@bio-quantum.ai](mailto:admin@bio-quantum.ai)
- **DevOps Team:** [devops@bio-quantum.ai](mailto:devops@bio-quantum.ai)
- **Security Team:** [security@bio-quantum.ai](mailto:security@bio-quantum.ai)

### Maintenance Schedule

- **Daily:** Health checks, log review
- **Weekly:** Performance analysis, security review
- **Monthly:** System updates, backup verification
- **Quarterly:** Security audit, dependency updates

### Documentation Updates

- Keep deployment documentation current

- Document all configuration changes
  - Maintain runbooks for common procedures
  - Regular disaster recovery testing
- 

## **POST-DEPLOYMENT CHECKLIST**

- ☐ All services running and healthy
  - ☐ Database connectivity verified
  - ☐ Exchange API connectivity confirmed
  - ☐ SSL certificates installed and valid
  - ☐ Monitoring dashboards accessible
  - ☐ Backup procedures tested
  - ☐ Security hardening completed
  - ☐ Performance baselines established
  - ☐ Alert notifications configured
  - ☐ Documentation updated
  - ☐ Team training completed
  - ☐ Disaster recovery plan tested
- 

 **Congratulations! Your Bio-Quantum AI Trading Platform is now deployed and ready for production use!**

For additional support and advanced configuration options, please refer to the detailed component documentation in the `/docs` directory.