version: '3.8'

services:

sportai-app:

build: .

container\_name: sportai-suite

ports:

- "8501:8501"

environment:

- APP\_ENV=production

- STREAMLIT\_SERVER\_PORT=8501

- STREAMLIT\_SERVER\_ADDRESS=0.0.0.0

- STREAMLIT\_SERVER\_HEADLESS=true

- STREAMLIT\_BROWSER\_GATHER\_USAGE\_STATS=false

- SECRET\_KEY=${SECRET\_KEY:-your-secret-key-here}

- DATABASE\_URL=${DATABASE\_URL:-sqlite:///database/sportai.db}

- SESSION\_TIMEOUT=${SESSION\_TIMEOUT:-3600}

- MAX\_LOGIN\_ATTEMPTS=${MAX\_LOGIN\_ATTEMPTS:-5}

volumes:

- ./database:/app/database

- ./logs:/app/logs

- ./audit\_logs:/app/audit\_logs

- ./configurations:/app/configurations

- ./uploads:/app/uploads

restart: unless-stopped

healthcheck:

test: ["CMD", "curl", "-f", "http://localhost:8501/\_stcore/health"]

interval: 30s

timeout: 10s

retries: 3

start\_period: 40s

networks:

- sportai-network

depends\_on:

- redis

- postgres

redis:

image: redis:7-alpine

container\_name: sportai-redis

ports:

- "6379:6379"

volumes:

- redis-data:/data

command: redis-server --appendonly yes --requirepass ${REDIS\_PASSWORD:-sportai2025}

restart: unless-stopped

networks:

- sportai-network

healthcheck:

test: ["CMD", "redis-cli", "ping"]

interval: 30s

timeout: 3s

retries: 3

postgres:

image: postgres:15-alpine

container\_name: sportai-postgres

ports:

- "5432:5432"

environment:

- POSTGRES\_DB=${POSTGRES\_DB:-sportai}

- POSTGRES\_USER=${POSTGRES\_USER:-sportai\_user}

- POSTGRES\_PASSWORD=${POSTGRES\_PASSWORD:-sportai\_pass}

- PGDATA=/var/lib/postgresql/data/pgdata

volumes:

- postgres-data:/var/lib/postgresql/data

- ./scripts/init\_db.sql:/docker-entrypoint-initdb.d/init.sql

restart: unless-stopped

networks:

- sportai-network

healthcheck:

test: ["CMD-SHELL", "pg\_isready -U ${POSTGRES\_USER:-sportai\_user}"]

interval: 30s

timeout: 3s

retries: 3

nginx:

image: nginx:alpine

container\_name: sportai-nginx

ports:

- "80:80"

- "443:443"

volumes:

- ./nginx/nginx.conf:/etc/nginx/nginx.conf:ro

- ./nginx/ssl:/etc/nginx/ssl:ro

- ./static:/usr/share/nginx/html/static:ro

depends\_on:

- sportai-app

restart: unless-stopped

networks:

- sportai-network

backup:

image: alpine:latest

container\_name: sportai-backup

volumes:

- ./database:/database:ro

- ./backups:/backups

- ./scripts:/scripts:ro

environment:

- BACKUP\_SCHEDULE=${BACKUP\_SCHEDULE:-0 2 \* \* \*}

command: >

sh -c "

apk add --no-cache dcron bash postgresql-client &&

echo '${BACKUP\_SCHEDULE:-0 2 \* \* \*} /scripts/backup.sh' | crontab - &&

crond -f -l 2

"

restart: unless-stopped

networks:

- sportai-network

prometheus:

image: prom/prometheus:latest

container\_name: sportai-prometheus

ports:

- "9090:9090"

volumes:

- ./monitoring/prometheus.yml:/etc/prometheus/prometheus.yml:ro

- prometheus-data:/prometheus

command:

- '--config.file=/etc/prometheus/prometheus.yml'

- '--storage.tsdb.path=/prometheus'

restart: unless-stopped

networks:

- sportai-network

grafana:

image: grafana/grafana:latest

container\_name: sportai-grafana

ports:

- "3000:3000"

environment:

- GF\_SECURITY\_ADMIN\_PASSWORD=${GRAFANA\_PASSWORD:-admin}

- GF\_INSTALL\_PLUGINS=redis-datasource

volumes:

- grafana-data:/var/lib/grafana

- ./monitoring/grafana/dashboards:/etc/grafana/provisioning/dashboards:ro

- ./monitoring/grafana/datasources:/etc/grafana/provisioning/datasources:ro

depends\_on:

- prometheus

restart: unless-stopped

networks:

- sportai-network

networks:

sportai-network:

driver: bridge

volumes:

redis-data:

postgres-data:

prometheus-data:

grafana-data: