Deployment & Operations

AI-Powered Personal Productivity System

5.1 Deployment Pipeline

CI/CD Configuration

yaml

```yaml
# .github/workflows/deploy.yml structure

stages:
  - lint
  - test
  - build
  - deploy

lint:
  parallel:
    - eslint
    - typescript
    - prettier
  fail_fast: true

test:
  parallel:
    - unit_tests
    - integration_tests
    - accessibility_tests
  coverage:
    threshold: 85%

build:
  environment:
    - NODE_ENV: production
    - ANALYZE: true
  artifacts:
    - dist/
    - stats.json

deploy:
  staging:
    trigger: push to develop
    url: staging.app.vercel.app
    tests: smoke_tests

  production:
    trigger: push to main
    url: app.vercel.app
    tests: e2e_tests
    rollback: automatic on failure
```

Environment Configuration

```bash
# Required environment variables

# Vercel
VERCEL_ENV=production
VERCEL_URL=https://app.vercel.app
VERCEL_REGION=iad1

# Supabase
NEXT_PUBLIC_SUPABASE_URL=https://[project].supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=[anon_key]
SUPABASE_SERVICE_KEY=[service_key]

# Security
ENCRYPTION_KEY=[generated_key]
JWT_SECRET=[jwt_secret]
CORS_ORIGINS=https://app.vercel.app

# AI/LLM
LLM_MODEL_PATH=/models/mistral-7b
LLM_WORKER_MEMORY=2048

# Monitoring
SENTRY_DSN=[optional_sentry_dsn]
ANALYTICS_ID=[optional_analytics]
```

Vercel Configuration

```json
```

```
// vercel.json
{
  "buildCommand": "npm run build",
  "outputDirectory": "dist",
  "devCommand": "npm run dev",
  "installCommand": "npm install",
  "framework": "nextjs",

  "functions": {
    "api/tasks/*.js": {
      "maxDuration": 10,
      "memory": 1024
    },
    "api/sync/*.js": {
      "maxDuration": 30,
      "memory": 512
    },
    "api/ai/*.js": {
      "maxDuration": 60,
      "memory": 2048
    }
  },

  "crons": [
    {
      "path": "/api/cron/daily-cleanup",
      "schedule": "0 2 * * *"
    },
    {
      "path": "/api/cron/sync-check",
      "schedule": "*/15 * * * *"
    },
    {
      "path": "/api/cron/backup",
      "schedule": "0 3 * * *"
    }
  ],

  "headers": [
    {
      "source": "/(.*)",
      "headers": [
        {
```

```
        "key": "X-Content-Type-Options",
        "value": "nosniff"
      },
      {
        "key": "X-Frame-Options",
        "value": "DENY"
      },
      {
        "key": "X-XSS-Protection",
        "value": "1; mode=block"
      },
      {
        "key": "Referrer-Policy",
        "value": "strict-origin-when-cross-origin"
      },
      {
        "key": "Content-Security-Policy",
        "value": "default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline';"
      }
    ]
  }
 ],

 "redirects": [
  {
    "source": "/docs",
    "destination": "/documentation",
    "permanent": true
  }
 ],

 "rewrites": [
  {
    "source": "/api/:path*",
    "destination": "/api/:path*"
  }
 ]
}
```

GitHub Actions Workflow

yaml

```yaml
# .github/workflows/main.yml

name: Deploy to Production

on:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  lint:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
          cache: 'npm'
      - run: npm ci
      - run: npm run lint
      - run: npm run type-check

  test:
    runs-on: ubuntu-latest
    needs: lint
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
          cache: 'npm'
      - run: npm ci
      - run: npm run test:unit
      - run: npm run test:integration
      - uses: actions/upload-artifact@v3
        with:
          name: coverage
          path: coverage/

  security:
    runs-on: ubuntu-latest
    needs: lint
```

```yaml
  steps:
    - uses: actions/checkout@v3
    - run: npm audit
    - uses: snyk/actions/node@master
      env:
        SNYK_TOKEN: ${{ secrets.SNYK_TOKEN }}

  build:
    runs-on: ubuntu-latest
    needs: [test, security]
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
          cache: 'npm'
      - run: npm ci
      - run: npm run build
      - uses: actions/upload-artifact@v3
        with:
          name: dist
          path: dist/

  deploy:
    runs-on: ubuntu-latest
    needs: build
    if: github.ref == 'refs/heads/main'
    steps:
      - uses: actions/checkout@v3
      - uses: actions/download-artifact@v3
        with:
          name: dist
          path: dist/
      - uses: amondnet/vercel-action@v20
        with:
          vercel-token: ${{ secrets.VERCEL_TOKEN }}
          vercel-args: '--prod'
          vercel-org-id: ${{ secrets.VERCEL_ORG_ID }}
          vercel-project-id: ${{ secrets.VERCEL_PROJECT_ID }}
```

5.2 Monitoring & Observability

Key Metrics Dashboard

typescript

```typescript
interface MetricsDashboard {
  // User Metrics
  activeUsers: {
    daily: number;
    weekly: number;
    monthly: number;
  };

  // Performance Metrics
  performance: {
    p50_response_time: number;
    p95_response_time: number;
    p99_response_time: number;
    error_rate: number;
  };

  // Business Metrics
  usage: {
    tasks_created: number;
    tasks_completed: number;
    completion_rate: number;
    avg_session_duration: number;
  };

  // System Health
  health: {
    uptime: number;
    cpu_usage: number;
    memory_usage: number;
    database_connections: number;
  };

  // AI Metrics
  ai: {
    inference_time: number;
    parse_accuracy: number;
    cache_hit_rate: number;
    fallback_rate: number;
  };
}
```

Alert Configurations

Alert	Condition	Severity	Action
High Error Rate	>1% errors	Critical	Page on-call
Slow Response	p95 >2s	High	Slack notification
Low Completion Rate	<50% daily	Medium	Email daily
Storage Warning	>80% used	Medium	Email team
Sync Failures	>10/hour	High	Investigate logs
AI Timeout	>5% timeout	Low	Monitor trend

Logging Strategy

javascript

```
// Structured logging configuration
const logger = {
  levels: {
    error: 0,
    warn: 1,
    info: 2,
    http: 3,
    verbose: 4,
    debug: 5,
    silly: 6
  },

  format: {
    timestamp: true,
    level: true,
    message: true,
    metadata: true,
    context: {
      userId: true,
      sessionId: true,
      requestId: true,
      environment: true
    }
  },

  transports: {
    console: process.env.NODE_ENV !== 'production',
    file: {
      error: '/logs/error.log',
      combined: '/logs/combined.log'
    },
    remote: {
      service: 'datadog',
      apiKey: process.env.DATADOG_API_KEY
    }
  }
};

// Usage example
logger.info('Task created', {
  taskId: task.id,
  userId: user.id,
  duration: performance.now() - startTime,
```

```
  metadata: {
    priority: task.priority,
    hasDeadline: !!task.dueDate
  }
});
```

Health Checks

```typescript
// Health check endpoints
app.get('/health', (req, res) => {
  res.json({
    status: 'healthy',
    timestamp: new Date().toISOString(),
    uptime: process.uptime(),
    memory: process.memoryUsage(),
    version: process.env.APP_VERSION
  });
});

app.get('/health/detailed', async (req, res) => {
  const checks = {
    database: await checkDatabase(),
    redis: await checkRedis(),
    storage: await checkStorage(),
    api: await checkExternalAPIs()
  };

  const allHealthy = Object.values(checks).every(c => c.status === 'healthy');

  res.status(allHealthy ? 200 : 503).json({
    status: allHealthy ? 'healthy' : 'degraded',
    checks,
    timestamp: new Date().toISOString()
  });
});
```

5.3 Maintenance Procedures

Backup Strategy

```yaml
```

```yaml
backup_schedule:
  database:
    frequency: daily
    retention: 30 days
    time: 02:00 UTC
    location: encrypted_s3

  user_data:
    frequency: hourly
    retention: 7 days
    incremental: true
    encryption: client_side

  system_config:
    frequency: on_change
    retention: 90 days
    versioned: true

restore_procedures:
  priority_order:
    1: authentication_system
    2: user_preferences
    3: active_tasks
    4: completed_tasks
    5: analytics_data

  validation:
    - checksum_verification
    - data_integrity_check
    - user_acceptance_test
```

Database Maintenance

```sql
```

```sql
-- Daily maintenance tasks
-- Run at 02:00 UTC

-- Update statistics
ANALYZE tasks;
ANALYZE users;
ANALYZE sync_log;

-- Vacuum tables
VACUUM (ANALYZE) tasks;
VACUUM (ANALYZE) sync_log;

-- Refresh materialized views
REFRESH MATERIALIZED VIEW CONCURRENTLY task_statistics;

-- Archive old data
INSERT INTO tasks_archive
SELECT * FROM tasks
WHERE deleted_at < NOW() - INTERVAL '90 days';

DELETE FROM tasks
WHERE deleted_at < NOW() - INTERVAL '90 days';

-- Clean up old sync logs
DELETE FROM sync_log
WHERE synced_at < NOW() - INTERVAL '30 days';
```

Update Procedures

bash

```
# Zero-downtime deployment process

1. Pre-deployment
   - Run test suite
   - Build artifacts
   - Verify dependencies

2. Blue-Green Deploy
   - Deploy to green environment
   - Run smoke tests
   - Switch traffic (10% → 50% → 100%)
   - Monitor metrics

3. Rollback Criteria
   - Error rate >2%
   - Response time >3s
   - Memory usage >300MB
   - Failed health checks

4. Post-deployment
   - Clear CDN cache
   - Update documentation
   - Notify users of changes
```

Incident Response

```yaml
```

```yaml
incident_levels:
  P1_Critical:
    description: Complete service outage
    response_time: 15 minutes
    escalation: Immediate page to on-call

  P2_High:
    description: Major feature broken
    response_time: 1 hour
    escalation: Notify team lead

  P3_Medium:
    description: Minor feature issue
    response_time: 4 hours
    escalation: Team notification

  P4_Low:
    description: Cosmetic issue
    response_time: Next business day
    escalation: Log in backlog

response_procedures:
  1_detect:
    - Automated monitoring alert
    - User report
    - Manual discovery

  2_triage:
    - Assess severity
    - Identify affected users
    - Determine scope

  3_respond:
    - Implement immediate mitigation
    - Communicate status
    - Begin investigation

  4_resolve:
    - Deploy fix
    - Verify resolution
    - Monitor stability

  5_postmortem:
```

- Document timeline
- Identify root cause
- Define prevention measures

5.4 Security Operations

Security Scanning

```bash
# Regular security scans

# Dependency scanning
npm audit
snyk test
safety check

# Code scanning
eslint --ext .ts,.tsx src/
semgrep --config=auto src/

# Container scanning (if using Docker)
docker scan app:latest
trivy image app:latest

# Infrastructure scanning
checkov -d infrastructure/
tfsec infrastructure/
```

Certificate Management

```yaml
```

```yaml
ssl_certificates:
  provider: Let's Encrypt
  auto_renewal: true
  renewal_threshold: 30 days

  monitoring:
    - Check expiry daily
    - Alert 14 days before expiry
    - Auto-renew 7 days before expiry

  domains:
    - app.vercel.app
    - api.app.vercel.app
    - "*.app.vercel.app"
```

Access Control

yaml

```yaml
rbac_matrix:
  admin:
    - Full system access
    - Production deployment
    - Database management
    - User management

  developer:
    - Code repository access
    - Staging deployment
    - Read-only production logs
    - Development database access

  support:
    - User data read access
    - Support ticket management
    - Read-only analytics

  viewer:
    - Read-only dashboard access
    - Public documentation
```

Document Version: 1.0.0

Last Updated: January 2024

Next Review: February 2024