AuriVault Cost Optimization & Technical Roadmap

Phase 1: Immediate Cost Reduction (Next 30 Days)

Replace Expensive AI APIs

Current: OpenAI GPT-4 (~$0.03-0.06/1K tokens) **Replace with:**

- **Ollama** for local LLM hosting (Free, self-hostable)
- **OpenRouter** as fallback (cheaper API aggregator)
- **Groq** for fast inference (free tier + cheap pricing)

```typescript
// New AI service architecture
interface AIProvider {
  name: string;
  cost: 'free' | 'low' | 'medium' | 'high';
  selfHostable: boolean;
  fallback?: AIProvider;
}

const aiProviders: AIProvider[] = [
  { name: 'ollama', cost: 'free', selfHostable: true },
  { name: 'groq', cost: 'low', selfHostable: false },
  { name: 'openrouter', cost: 'medium', selfHostable: false },
  { name: 'openai', cost: 'high', selfHostable: false }
];
```

Replace Vector Database

Current: Pinecone ($70+/month) **Replace with:**

- **Qdrant** (open-source, self-hostable, free)
- **Weaviate** (open-source alternative)
- **ChromaDB** (simple, embedded option)

Self-Hosted Infrastructure Stack

```yaml
```

```yaml
# docker-compose.yml for complete self-hosted setup
version: '3.8'
services:
  aurivault:
    build: .
    ports:
      - "3000:3000"
    depends_on:
      - postgres
      - qdrant
      - ollama

  postgres:
    image: postgres:15
    environment:
      POSTGRES_DB: aurivault
      POSTGRES_PASSWORD: ${DB_PASSWORD}
    volumes:
      - postgres_data:/var/lib/postgresql/data

  qdrant:
    image: qdrant/qdrant:latest
    ports:
      - "6333:6333"
    volumes:
      - qdrant_data:/qdrant/storage

  ollama:
    image: ollama/ollama:latest
    ports:
      - "11434:11434"
    volumes:
      - ollama_data:/root/.ollama
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [gpu]

volumes:
  postgres_data:
  qdrant_data:
  ollama_data:
```

Phase 2: Developer-First Features (Next 60 Days)

API-First Architecture

```typescript
// Public API endpoints developers need
POST /api/v1/documents/upload
GET  /api/v1/documents
POST /api/v1/query
GET  /api/v1/embeddings
POST /api/v1/chat/stream
GET  /api/v1/analytics
```

Developer Experience Features

1. **CLI Tool**

```bash
npm install -g @aurivault/cli
aurivault init
aurivault upload ./docs/**/*.md
aurivault query "API rate limits"
```

2. **SDK Libraries**

```typescript
import { AuriVault } from '@aurivault/sdk';

const vault = new AuriVault({
  endpoint: 'https://your-instance.com',
  apiKey: 'your-key'
});

const result = await vault.query('deployment strategies');
```

3. **IDE Integrations**
   - VS Code extension
   - JetBrains plugin
   - Vim/Neovim integration

Developer-Specific Document Types

- **Git repositories** (automatic syncing)

- **API documentation** (OpenAPI/Swagger)
- **Code comments** extraction
- **Changelog parsing**
- **Documentation sites** (GitBook, Notion, Confluence)

Phase 3: Monetization Strategy

Free Tier (Community Edition)

- Up to 1,000 documents
- Basic AI models (Ollama)
- Community support
- Self-hosting required
- Core API access

Pro Tier ($29/month per team)

- Unlimited documents
- Advanced AI models
- Cloud hosting option
- Email support
- Advanced analytics
- Integrations (Slack, Discord, etc.)

Enterprise Tier ($199/month + custom)

- SSO/SAML
- Advanced security features
- Dedicated support
- Custom AI model fine-tuning
- On-premise deployment support
- SLA guarantees

Usage-Based Add-ons

- **AI Query Credits:** $0.01 per query above limits
- **Storage:** $5/GB beyond included storage
- **API Calls:** $0.001 per API call above limits

Phase 4: Go-to-Market Strategy

Target Developer Personas

1. **DevOps Engineers** (20-100 person companies)
   - Pain: Scattered infrastructure docs, runbooks, incident reports
   - Value prop: "Chat with your entire ops knowledge base"
2. **API Companies** (Developer tools, SaaS)
   - Pain: Customer support scaling, developer onboarding
   - Value prop: "Self-service developer support powered by your docs"
3. **Open Source Projects**
   - Pain: Contributor onboarding, maintaining wikis
   - Value prop: "Make your project documentation searchable and interactive"

Distribution Channels

1. **GitHub Marketplace** presence
2. **Developer conferences** (KubeCon, DockerCon, etc.)
3. **Technical blog content**
4. **Open source contributions** to related projects
5. **Hacker News** and **r/selfhosted** community engagement

Content Marketing Topics

- "Building a RAG system that actually works"
- "Self-hosting your AI knowledge base"
- "Why developers need specialized documentation tools"
- "Comparing vector databases for production use"

Technical Implementation Priority

Week 1-2: Core Infrastructure

- [ ] Integrate Qdrant vector database
- [ ] Add Ollama support for local LLMs
- [ ] Create Docker Compose setup
- [ ] Basic API authentication

Week 3-4: Developer Features

- [ ] Public API endpoints
- [ ] Basic CLI tool

- [ ] Git repository integration
- [ ] Simple SDK (TypeScript/Python)

Week 5-8: Polish & Testing

- [ ] Comprehensive documentation
- [ ] Example deployments (AWS, GCP, DigitalOcean)
- [ ] Performance optimization
- [ ] Security audit

Week 9-12: Go-to-Market

- [ ] Landing page optimization
- [ ] Pricing page
- [ ] Documentation site
- [ ] First customer interviews

Revenue Projections

Conservative Estimates (12 months)

- **Month 1-3:** 0 customers (building & validating)
- **Month 4-6:** 10 teams × $29 = $290/month
- **Month 7-9:** 50 teams × $29 = $1,450/month
- **Month 10-12:** 100 teams × $29 + 5 enterprise × $199 = $3,895/month

Key Metrics to Track

- **Free-to-paid conversion rate** (target: 15%)
- **Monthly churn rate** (target: <5%)
- **Customer acquisition cost** (target: <$50)
- **Average revenue per user** (target: $35)

Success Indicators

- **Technical:** 1-click deployment working reliably
- **Product:** Developers can get value in <15 minutes
- **Business:** First $1K MRR within 6 months
- **Community:** 100+ GitHub stars, active Discord