

version: '3.8'

services: ##### # Core Service  
Containers #####

# Strategy Selection Engine opportunity-scanner: build:  
./core/strategy\_selection/opportunity\_scanner container\_name: amgf-opportunity-scanner  
restart: unless-stopped depends\_on: - mongodb - redis environment: -  
MONGODB\_URI=mongodb://mongodb:27017/amgf - REDIS\_URI=redis://redis:6379  
volumes: - ./core/strategy\_selection/opportunity\_scanner/app - ./templates/templates  
networks: - amgf-network

roi-predictor: build: ./core/strategy\_selection/roi\_predictor container\_name: amgf-roi-  
predictor restart: unless-stopped depends\_on: - mongodb - redis environment: -  
MONGODB\_URI=mongodb://mongodb:27017/amgf - REDIS\_URI=redis://redis:6379  
volumes: - ./core/strategy\_selection/roi\_predictor/app - ./templates/templates networks: -  
amgf-network

resource-allocator: build: ./core/strategy\_selection/resource\_allocator container\_name:  
amgf-resource-allocator restart: unless-stopped depends\_on: - mongodb - redis -  
opportunity-scanner - roi-predictor environment: -  
MONGODB\_URI=mongodb://mongodb:27017/amgf - REDIS\_URI=redis://redis:6379  
volumes: - ./core/strategy\_selection/resource\_allocator/app networks: - amgf-network

# Execution System digital-service-executor: build:  
./core/execution/digital\_service\_executor container\_name: amgf-digital-service-executor  
restart: unless-stopped depends\_on: - mongodb - redis - resource-allocator environment: -  
MONGODB\_URI=mongodb://mongodb:27017/amgf - REDIS\_URI=redis://redis:6379  
volumes: - ./core/execution/digital\_service\_executor/app -  
./templates/services/templates/services networks: - amgf-network

content-arbitrage-system: build: ./core/execution/content\_arbitrage\_system container\_name:  
amgf-content-arbitrage-system restart: unless-stopped depends\_on: - mongodb - redis -  
resource-allocator environment: - MONGODB\_URI=mongodb://mongodb:27017/amgf -  
REDIS\_URI=redis://redis:6379 volumes: - ./core/execution/content\_arbitrage\_system/app -  
./templates/content/templates/content networks: - amgf-network

micro-arbitrage-engine: build: ./core/execution/micro\_arbitrage\_engine container\_name:  
amgf-micro-arbitrage-engine restart: unless-stopped depends\_on: - mongodb - redis -  
resource-allocator environment: - MONGODB\_URI=mongodb://mongodb:27017/amgf -  
REDIS\_URI=redis://redis:6379 volumes: - ./core/execution/micro\_arbitrage\_engine/app  
networks: - amgf-network

# Monitoring & Optimization performance-tracker: build:  
./core/monitoring/performance\_tracker container\_name: amgf-performance-tracker restart:  
unless-stopped depends\_on: - mongodb - redis - digital-service-executor - content-  
arbitrage-system - micro-arbitrage-engine environment: -  
MONGODB\_URI=mongodb://mongodb:27017/amgf - REDIS\_URI=redis://redis:6379  
volumes: - ./core/monitoring/performance\_tracker/app networks: - amgf-network

real-time-optimizer: build: ./core/monitoring/real\_time\_optimizer container\_name: amgf-real-  
time-optimizer restart: unless-stopped depends\_on: - mongodb - redis - performance-  
tracker environment: - MONGODB\_URI=mongodb://mongodb:27017/amgf -  
REDIS\_URI=redis://redis:6379 volumes: - ./core/monitoring/real\_time\_optimizer/app  
networks: - amgf-network

reinvestment-manager: build: ./core/monitoring/reinvestment\_manager container\_name:  
amgf-reinvestment-manager restart: unless-stopped depends\_on: - mongodb - redis -  
performance-tracker - real-time-optimizer environment: -  
MONGODB\_URI=mongodb://mongodb:27017/amgf - REDIS\_URI=redis://redis:6379  
volumes: - ./core/monitoring/reinvestment\_manager/app networks: - amgf-network

##### # Utility Containers  
#####

```
# Data Storage mongodb: image: mongo:latest container_name: amgf-mongodb restart:
unless-stopped environment: -
MONGO_INITDB_ROOT_USERNAME=MONGO_USERNAME : -admin
- MONGO_INITDB_ROOT_PASSWORD={MONGO_PASSWORD:password} volumes: -
mongodb-data:/data/db ports: - "27017:27017" networks: - amgf-network
```

```
redis: image: redis:latest container_name: amgf-redis restart: unless-stopped volumes: -
redis-data:/data ports: - "6379:6379" networks: - amgf-network
```

```
# Monitoring prometheus: image: prom/prometheus:latest container_name: amgf-
prometheus restart: unless-stopped volumes: -
./utility/monitoring/prometheus/etc/prometheus - prometheus-data:/prometheus ports: -
"9090:9090" networks: - amgf-network
```

```
grafana: image: grafana/grafana:latest container_name: amgf-grafana restart: unless-stopped
depends_on: - prometheus volumes: - ./utility/monitoring/grafana/etc/grafana/provisioning -
grafana-data:/var/lib/grafana ports: - "3000:3000" networks: - amgf-network
```

```
# Logging elasticsearch: image: docker.elastic.co/elasticsearch/elasticsearch:7.10.0
container_name: amgf-elasticsearch restart: unless-stopped environment: -
discovery.type=single-node - "ES_JAVA_OPTS=-Xms512m -Xmx512m" volumes: -
elasticsearch-data:/usr/share/elasticsearch/data ports: - "9200:9200" networks: - amgf-
network
```

```
logstash: image: docker.elastic.co/logstash/logstash:7.10.0 container_name: amgf-logstash
restart: unless-stopped depends_on: - elasticsearch volumes: -
./utility/monitoring/logstash/usr/share/logstash/pipeline ports: - "5000:5000" networks: -
amgf-network
```

```
kibana: image: docker.elastic.co/kibana/kibana:7.10.0 container_name: amgf-kibana restart:
unless-stopped depends_on: - elasticsearch environment: -
ELASTICSEARCH_HOSTS=http://elasticsearch:9200 ports: - "5601:5601" networks: -
amgf-network
```

```
##### # Integration Containers
#####
```

```
# API Gateway api-gateway: build: ./integration/api_gateway container_name: amgf-api-
gateway restart: unless-stopped depends_on: - mongodb - redis environment: -
MONGODB_URI=mongodb://mongodb:27017/amgf - REDIS_URI=redis://redis:6379
ports: - "8080:8080" volumes: - ./integration/api_gateway/app networks: - amgf-network
```

```
# Platform Integration fiverr-integration: build: ./integration/platform_integration/fiverr
container_name: amgf-fiverr-integration restart: unless-stopped depends_on: - mongodb -
redis - api-gateway environment: - MONGODB_URI=mongodb://mongodb:27017/amgf -
REDIS_URI=redis://redis:6379 - FIVERR_API_KEY=FIVERR_API_KEY
- FIVERR_API_SECRET={FIVERR_API_SECRET} volumes: -
./integration/platform_integration/fiverr/app networks: - amgf-network
```

```
upwork-integration: build: ./integration/platform_integration/upwork container_name: amgf-
upwork-integration restart: unless-stopped depends_on: - mongodb - redis - api-gateway
environment: - MONGODB_URI=mongodb://mongodb:27017/amgf -
REDIS_URI=redis://redis:6379 - UPWORK_API_KEY=UPWORK_API_KEY
- UPWORK_API_SECRET={UPWORK_API_SECRET} volumes: -
./integration/platform_integration/upwork/app networks: - amgf-network
```

```
content-platform-integration: build: ./integration/platform_integration/content_platforms
container_name: amgf-content-platform-integration restart: unless-stopped depends_on: -
mongodb - redis - api-gateway environment: -
MONGODB_URI=mongodb://mongodb:27017/amgf - REDIS_URI=redis://redis:6379 -
MEDIUM_API_KEY=MEDIUM_API_KEY - YOUTUBE_API_KEY={YOUTUBE_API_KEY}
volumes: - ./integration/platform_integration/content_platforms/app networks: - amgf-
network
```

```
marketplace-integration: build: ./integration/platform_integration/marketplaces
container_name: amgf-marketplace-integration restart: unless-stopped depends_on: -
```

```

mongodb - redis - api-gateway environment: -
MONGODB_URI=mongodb://mongodb:27017/amgf - REDIS_URI=redis://redis:6379 -
ETSY_API_KEY=ETSY_API_KEY - ETSY_API_SECRET={ETSY_API_SECRET} volumes: -
./integration/platform_integration/marketplaces:/app networks: - amgf-network

##### # Specialized Function Containers
#####

# Content Generation content-generation: build: ./specialized/content_generation
container_name: amgf-content-generation restart: unless-stopped depends_on: - mongodb -
redis environment: - MONGODB_URI=mongodb://mongodb:27017/amgf -
REDIS_URI=redis://redis:6379 - OPENAI_API_KEY=${OPENAI_API_KEY} volumes: -
./specialized/content_generation:/app - ./templates/content:/templates/content networks: -
amgf-network

# Analytics market-analysis: build: ./specialized/analytics/market_analysis container_name:
amgf-market-analysis restart: unless-stopped depends_on: - mongodb - redis environment:
- MONGODB_URI=mongodb://mongodb:27017/amgf - REDIS_URI=redis://redis:6379
volumes: - ./specialized/analytics/market_analysis:/app networks: - amgf-network

# Decision Support decision-support: build: ./specialized/decision_support container_name:
amgf-decision-support restart: unless-stopped depends_on: - mongodb - redis environment:
- MONGODB_URI=mongodb://mongodb:27017/amgf - REDIS_URI=redis://redis:6379
volumes: - ./specialized/decision_support:/app networks: - amgf-network

# Dashboard dashboard: build: ./specialized/dashboard container_name: amgf-dashboard
restart: unless-stopped depends_on: - mongodb - redis - api-gateway environment: -
MONGODB_URI=mongodb://mongodb:27017/amgf - REDIS_URI=redis://redis:6379
ports: - "8081:80" volumes: - ./specialized/dashboard:/app networks: - amgf-network

```

## Networks and Volumes

networks: amgf-network: driver: bridge

volumes: mongodb-data: redis-data: elasticsearch-data: prometheus-data: grafana-data: