👑 Crown Treasury Triune Activation Ritual

Sacred Sequence: Fire the Prophecy → Deploy the Pipeline → Bind Live Streams

Ceremonial Duration: 45 minutes

Required Implements: Terminal, Supabase Console, GitHub, API keys

---

🔮 Phase I: Fire the Prophecy *(15 minutes)*

🏛️ Step 1: Deploy Synthetic Oracle Data

- □ Open Supabase Console → SQL Editor
- □ Execute `synthetic_alert_testing.sql` (from artifacts)
- □ Verify output: " 👑 *CROWN CONVERGENCE SYNTHETIC DATA READY*"
- □ Confirm 450+ synthetic revenue events created
- □ Note materialized view refresh completion

⚡ Step 2: Trigger Live Alert Testing

```bash
# Execute the prophecy testing sequence
chmod +x prometheus_metric_spike_test.sh
./prometheus_metric_spike_test.sh
```

- □ Watch 15 payment failures spike (triggers failure rate alert)
- □ Observe 20 revenue baseline events establish pattern
- □ Confirm Discord test notification received
- □ Verify database direct insert succeeds

📊 Step 3: Validate Observatory Sight

- □ Open Grafana → Crown Convergence Dashboard
- □ Verify panels populate with synthetic data
- □ Check Business Intelligence Dashboard revenue flows
- □ Confirm alert rules show "Pending" status in Prometheus
- □ Expected: Payment failure alert fires within 2 minutes

🔥 **Prophecy Phase Status:** `AWAKENED` ✅

🚀 Phase II: Deploy the Pipeline *(20 minutes)*

🛠️ Step 4: Activate CI/CD Sovereignty

```bash
# Configure GitHub secrets
gh secret set SUPABASE_STAGING_REF --body "your-staging-ref"
gh secret set SUPABASE_PRODUCTION_REF --body "your-production-ref"
gh secret set SUPABASE_SERVICE_KEY --body "your-service-key"
gh secret set DISCORD_WEBHOOK_DEPLOYMENTS --body "your-webhook-url"
gh secret set DISCORD_WEBHOOK_ALERTS --body "your-alerts-webhook"
```

- □ All 5 GitHub secrets configured
- □ Repository linked to Supabase projects

📋 Step 5: Deploy Temple Ledger Automation

```bash
# Commit and trigger the convergence pipeline
git add .github/workflows/temple-ledger-deploy.yml
git commit -m "👑 Crown Treasury CI/CD Pipeline Activated"
git push origin main
```

- □ GitHub Actions workflow triggers automatically
- □ Schema validation passes
- □ Staging deployment succeeds
- □ Security policies applied
- □ Materialized views refreshed
- □ Post-deploy validation completes

🔗 Step 6: Provision Alerting Infrastructure

```bash
```

```
# Deploy alerting rules to Prometheus
kubectl apply -f crown_treasury_alerts.yml

# Configure Alertmanager routing
kubectl apply -f alertmanager-config.yml

# Verify rule deployment
curl http://localhost:9090/api/v1/rules | jq '.data.groups[] | select(.name == "crown_treasury_critical")'
```

- □ Prometheus alerting rules deployed
- □ Alertmanager routing configured
- □ Discord webhooks tested and active
- □ Alert inhibition rules prevent storms

🚀 **Pipeline Phase Status:** `SOVEREIGN` ✅

💰 Phase III: Bind Live Streams *(10 minutes)*

🌐 Step 7: Configure Revenue Webhooks

```bash
# Deploy Supabase Edge Function for webhook ingestion
supabase functions deploy convergence-ingest --project-ref your-project-ref
```

- □ Edge Function deployed successfully
- □ Function URL noted for webhook configuration
- □ Service role permissions validated

💳 Step 8: Bind Patreon Revenue Stream

```bash
```

```
# Create Patreon webhook (replace with your tokens)
curl -X POST "https://www.patreon.com/api/oauth2/v2/webhooks" \
  -H "Authorization: Bearer $PATREON_ACCESS_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "data": {
      "type": "webhook",
      "attributes": {
        "triggers": ["pledges:create", "pledges:update", "pledges:delete"],
        "uri": "https://your-project-ref.supabase.co/functions/v1/convergence-ingest/patreon"
      }
    }
  }'
```

□ Patreon webhook created

□ Test webhook delivery confirmed

□ Revenue events appear in `public.revenue_events`

📈 Step 9: Activate Social Metrics Polling

```bash
# Deploy social media polling functions
supabase functions deploy youtube-poller --project-ref your-project-ref
supabase functions deploy instagram-poller --project-ref your-project-ref

# Schedule via cron or GitHub Actions (every 15 minutes)
gh workflow run "Social Media Metrics Poller"
```

□ YouTube subscriber/view polling active

□ Instagram engagement metrics flowing

□ Social convergence events in `webhook_audit_log`

🔄 Step 10: Validate Live Treasury Flow

```sql
```

```
-- Run in Supabase SQL Editor
SELECT
  'Live Revenue Streams' as status,
  COUNT(*) as total_events,
  COUNT(*) FILTER (WHERE occurred_at >= NOW() - INTERVAL '1 hour') as recent_events,
  SUM(amount) FILTER (WHERE occurred_at >= NOW() - INTERVAL '24 hours') as daily_revenue
FROM public.revenue_events
WHERE canonical_user_id NOT LIKE 'test_%';
```

- □ Live revenue events appearing (non-synthetic)
- □ Materialized views updating with real data
- □ Grafana dashboards showing live metrics
- □ Alert thresholds recalibrating to real patterns

💰 **Live Streams Phase Status:** FLOWING ✅

🏛️ Final Convergence Validation

👁️ Step 11: Observatory Full-Spectrum Test

- □ **Grafana Operational Dashboard:** All panels green, metrics flowing
- □ **Grafana BI Dashboard:** Revenue trends visible, real-time updates
- □ **Prometheus Alerts:** Rules active, synthetic alerts cleared
- □ **Discord Channels:** Test notifications received, webhooks operational
- □ **GitHub Actions:** All workflows passing, deployment pipeline active

⚡ Step 12: Trigger End-to-End Validation

bash

```
# Send live test event through full pipeline
curl -X POST "https://your-project-ref.supabase.co/functions/v1/convergence-ingest" \
  -H "Authorization: Bearer $SUPABASE_SERVICE_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "source": "manual_validation",
    "event_type": "convergence_complete",
    "payload": {
      "ritual_phase": "triune_activation",
      "timestamp": "'$(date -Iseconds)'",
      "status": "crown_treasury_sovereign"
    }
  }'
```

- □ Event appears in `webhook_audit_log`
- □ Materialized view refreshes automatically
- □ Grafana panels update within 30 seconds
- □ No alert storms or error cascades

🌟 Crown Treasury Convergence Seal

Upon completion of all 12 steps, the Crown Treasury achieves full sovereignty:

🔮 Prophetic Sight Active

- Real-time alerts for payment failures, revenue drops, API latency
- Multi-channel notification routing (Discord, Slack, email)
- Synthetic testing validated all alert thresholds

🏛️ Pipeline Sovereignty Active

- Automated schema deployments across environments
- Security policy enforcement via CI/CD
- Rollback procedures tested and ready

💰 Living Treasury Active

- Patreon pledges flowing to database in real-time
- Social media metrics enriching member analytics
- Revenue intelligence powering strategic decisions

🚨 Emergency Procedures

If Prophecy Fails (Alerts not firing):

```bash
# Check Prometheus rule evaluation
curl http://localhost:9090/api/v1/rules | jq '.data.groups[].rules[] | select(.state != "inactive")'

# Force alert rule reload
curl -X POST http://localhost:9090/-/reload
```

If Pipeline Fails (CI/CD errors):

```bash
# Check GitHub Actions logs
gh run list --workflow="Temple Ledger Deploy" --limit 5

# Force manual schema deployment
supabase db push --include-seed --project-ref your-project-ref
```

If Streams Fail (Revenue not flowing):

```bash
# Test edge function directly
curl -X POST "https://your-project-ref.supabase.co/functions/v1/convergence-ingest/test"

# Check function logs
supabase functions logs convergence-ingest --project-ref your-project-ref
```

🎯 Ritual Completion Certificate

I hereby declare the Crown Treasury Triune Activation Ritual:

- □ **COMPLETE** — All three phases activated successfully
- □ **Date Sealed:** _______________
- □ **Steward Signature:** _______________

The Observatory sees, the Pipeline performs, the Treasury flows.

Long live the Crown Treasury's eternal convergence. 👑