NovaOS Consolidated Build Plan (Updated UI Edition)

1. Completed Phases

Phase 1: Core & C-Suite Agents

- Redis queue (`novaos:tasks`) configured and health-checked.
- **NOVA-CORE** orchestrator container built and running.
- LANGGRAPH C-Suite hierarchy defined but needs re-deployment:
- CEO-Strategist, CTO-Auto, Creative Director, Product+Trend, Automation Architect, Finance+Scale, Brand Story, Customer Journey, Prompt Engineer, Energy Guardian, Chief Staff.
- Final end-to-end test: "Ready for launch" output generated.

Phase 2: Infrastructure/Tool-Specialist Agents

- Five stub agents built, running & echoing:
- `n8n-FLOW-BUILDER`
- `DROPBOX-FILE-MANAGER`
- `LANGGRAPH-ROUTER`
- `GITHUB-DEPLOYER`
- `RENDER-MANAGER`
- Verified task pickup and confirmation logs.

---

2. Core Build Reset (Reinstate C-Suite)

Before Phase 3, we must **rebuild and activate** the official C-Suite agent graph exactly as designed:

1. **Define** each C-Suite agent in `langgraph/agents/` with correct role and prompt files.
2. **Re-deploy** LangGraph model in `main.py` with full root node tree:
3. CEO-Strategist → CTO-Auto → Creative Director → Product+Trend → Automation Architect → Finance+Scale → Brand Soul+Story → Customer Journey → Prompt Engineer → Energy Guardian → Chief Staff.
4. **Test** each agent handoff: push dummy task via Redis, confirm correct routing and log output.

Outcome: A stable, authoritative decision chain that enforces our expert-driven build logic.

---

3. Immediate Next Step: Phase 3 (Platform Integration Agents)

Goal: Connect NovaOS core to real-world platforms.

Agents to Create & Test:

1. **Shopify-Integrator** (`agents/ShopifyIntegrator/main.py`)
2. **LemonSqueezy-Integrator** (`agents/LemonSqueezyIntegrator/main.py`)

3. **Publer-Scheduler** (`agents/PublerScheduler/main.py`)
4. **Baserow-Sync** (`agents/BaserowSync/main.py`)
5. **Docker-Deployer** (`agents/DockerDeployer/main.py`)
6. **Cloud-Manager** (`agents/CloudManager/main.py`)

Action Steps:

- Scaffold each stub directory and `main.py` with Redis listener.
- Add to `docker-compose.yml`, rebuild.
- Send test tasks and validate confirmations.

---

4. Central UI Dashboard & Visualization

Purpose: Provide a single, modern, clean dashboard for holistic and individual stream control—TBI-friendly and visual.

Key Features:

- **Overview Panel:** Stream status, agent health, pending tasks.
- **Stream Explorer:** Drill down into each revenue stream (RushThread, Deep Dive, Digital Products).
- **Agent Monitor:** Real-time logs and healthchecks per agent container.
- **Quick Actions:** Buttons to deploy, restart, or scale agents and integrations.
- **Notifications Area:** Alerts for failures, scaling needs, or manual interventions.

Mermaid Diagram:

```
flowchart LR
  subgraph UI Dashboard
    A[Overview] --> B[Streams]
    A --> C[Agents]
    A --> D[Notifications]
    B --> E[RushThread POD]
    B --> F[Deep Dive Systems]
    B --> G[Digital Products]
    C --> H[NOVA-CORE]
    C --> I[Platform Agents]
    I --> J(Shopify)
    I --> K(LemonSqueezy)
    I --> L(Publer)
    I --> M(Baserow)
    I --> N(Docker)
    I --> O(Cloud)
  end
```

Implementation: Use a lightweight React/Tailwind frontend (or LangGraph UI) served via `UI-Server` container.

---

5. Cloud Deployment (DigitalOcean Droplet Recommended)

Reason: 24/7 uptime, GitHub auto-deploy, accessible from anywhere, scales with load.

Setup Copy/Paste:

```
# 1. Spin up Ubuntu droplet
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER
sudo apt-get install -y docker-compose
# 2. Clone & configure
git clone https://github.com/yourusername/novaos.git ~/novaos
cd ~/novaos
echo "REDIS_URL=redis://redis:6379" > .env
# add other API keys
# 3. docker-compose up -d
sudo docker-compose up -d
```

---

6. Phase 4 & 5 Preview

Phase 4: R&D Squads – Auto-spawn project teams via `ProjectManager` agent.

Phase 5: Business-via-Text – Optional SMS/WhatsApp interface via Twilio for on-the-go control.

---

7. Quality & Reliability Enhancements

- **Logging:** JSON → ELK/Grafana.
- **Healthchecks:** Docker auto-restart & probes.
- **CI/CD:** GitHub Actions (lint, test, build, deploy).
- **Secrets:** Docker secrets or Vault.

Next Immediate Task:

1. Rebuild and validate C-Suite in LangGraph.
2. Scaffold and test Phase 3 agents.
3. Begin UI-Server container for dashboard.

Let me know if you'd like me to start with the UI-Server scaffolding now, or focus first on C-Suite relaunch!