**Tech Stack Mapping – Cloud-Based Workflow Automation Platform**

## **🧱 Project Tech Stack**

### **1. 🌐 Frontend (User Interface)**

| **Layer** | **Tech** | **Purpose** |
| --- | --- | --- |
| UI (Web) | React.js + TailwindCSS | Clean, responsive dashboard & workflow builder |
| Dev Mode UI | Monaco Editor / JSON Editor | Custom payloads, debug views |
| Auth Views | React + Context/Auth0 SDK | Login/Signup, role-based routing |
| Charts/Logs | Recharts / Chart.js | Workflow analytics & execution logs |

Optional: Java Swing-based desktop companion app

### **2. 🔙 Backend (Server-Side)**

| **Component** | **Tech** | **Purpose** |
| --- | --- | --- |
| API Layer | Java (Spring Boot) | Core backend logic, REST APIs |
| Auth | Spring Security + JWT | Secure login, role-based access |
| Task Scheduling | Quartz Scheduler / Spring Events | Trigger → Action execution queues |
| Plugin System | Java Interfaces + Reflection | Runtime plugin loading for integrations |
| Webhook Listener | Spring WebSocket / Controllers | Accept events from external apps |
| API Gateway | Spring Cloud Gateway / Zuul | Route services, rate limit, load balance |

### **3. 💾 Database & Storage**

| **Purpose** | **Tech** | **Description** |
| --- | --- | --- |
| Users & Workflows | PostgreSQL | Relational data for users, workflows, plugins |
| Logs | MongoDB / Elasticsearch | Flexible storage for action logs, history |
| Secrets Vault | HashiCorp Vault / Jasypt | Encrypt & manage API tokens, secrets |
| Plugin Storage | Amazon S3 / Local file system | Store uploaded plugin .jar files |

### **4. ☁️ Cloud, DevOps, and Infra**

| **Component** | **Tech** | **Role** |
| --- | --- | --- |
| Hosting | AWS / Render / Railway | Deploy backend/frontend/cloud infra |
| CI/CD | GitHub Actions / Jenkins | Auto testing & deployment |
| Monitoring | Prometheus + Grafana / ELK | Metrics, error tracking |
| Containerization | Docker + Docker Compose | Environment consistency |
| Reverse Proxy | NGINX | Serve frontend + route API calls securely |
| Domain/SSL | Cloudflare + HTTPS | Secure and fast domain resolution |

### **5. 🔐 Cybersecurity Tools & Practices**

| **Feature** | **Tool/Technique** |
| --- | --- |
| Pen Testing | OWASP ZAP / Burp Suite |
| Auth Encryption | JWT + AES/RSA (Jasypt) |
| Network Security | SSL/TLS, IP whitelisting, rate limiting |
| Activity Monitoring | Spring Actuator + Alerts via Email/Slack |
| Input Validation | Hibernate Validator / Apache Commons Validator |

**Tech Stack Mapping – Zapier-Like Cloud-Based Automation Platform**

### **🖥️ Backend (Java)**

* **Java 21 (LTS)** – Core backend language
* **Spring Boot** – REST APIs, plugin execution, workflow engine
* **Maven** – Dependency & build management
* **PostgreSQL** – Primary database
* **Quartz Scheduler** – Workflow scheduling and timed triggers
* **JJWT / Spring Security** – Auth with JWT support
* **Jasypt / Vault** – Secrets encryption and storage

### **🌐 Frontend (Web UI)**

* **React.js** – SPA architecture, modular UI
* **Tailwind CSS** – Utility-first styling
* **Monaco Editor** – Advanced developer text editing (Dev Mode)
* **D3 / Chart.js** – Analytics & visualization
* **Axios** – API requests

### **☁️ Cloud & Deployment**

* **Render / Railway / AWS EC2** – Cloud deployment platforms
* **Docker** – Containerized deployment
* **Nginx** – Reverse proxy (optional)
* **GitHub Actions** – CI/CD automation

### **🧪 Testing & Security**

* **JUnit / Mockito** – Unit testing
* **OWASP ZAP / Burp Suite** – Cybersecurity testing tools
* **Spring Security** – Role-based access control

### **🔧 Development Environment**

* **IntelliJ IDEA** – Primary backend IDE for Java, Maven, and Spring
* **VS Code** – For frontend development (React, Tailwind, JSON)
* **GitHub** – Version control and repo hosting