**Blu AI cloud server:**

IP address : 206.189.19.160

N8N\_ip: https://n8n.bluai.fit

**Run docker:**

* docker compose -f n8n.yml up -d
* docker compose -f n8n.yml down
* docker compose -f n8n.yml logs -f

**1. Best Practices for Data Safety**

1. **Use a Dedicated Volume Folder** (we already did: /root/n8n/data)
2. **Back Up This Folder** regularly, for example:

tar czvf n8n-backup-$(date +%F).tar.gz /root/n8n/data

**Keep Updates Safe**

* To update n8n safely:
* docker compose -f n8n.yml pull
* docker compose -f n8n.yml up -d

Server setup:

1. Domain & DNS:

Bought and created a subdomain:

* Domain : bluai.fit (namecheap)
* Sub-domain: n8n.bluai.fit (cloudfare)

2. Server & Docker Setup:

* Created a DigitalOcean droplet (Ubuntu 22.04 LTS).
* Installed Docker + Docker Compose.
* Created /root/n8n/n8n.yml for n8n Docker container.
* Added a volume /root/n8n/data:/home/node/.n8n to persist workflows & credentials.

3.Run n8n in Docker:

* docker compose -f n8n.yml up -d

4. Secure with HTTPS

* Installed Nginx as a reverse proxy.
* Created a Nginx config to forward n8n.bluai.fit → localhost:5678.
* Installed Certbot and enabled free Let’s Encrypt SSL:
* sudo certbot --nginx -d n8n.bluai.fit

5. Harden Security

* Removed direct 5678 port exposure from n8n.yml.Instead we expose to localhost so Nginx can detect it
* Enabled UFW firewall:
* sudo ufw allow OpenSSH
* sudo ufw allow 'Nginx Full'
* sudo ufw deny 5678
* sudo ufw enable

**✅ Final Result**

* **Persistent n8n instance** running in Docker
* **Accessible securely via HTTPS**
* **No direct access to internal port 5678**
* **Data safe in /root/n8n/data**
* **Firewall protecting all other port**