Run n8n in AWS Cloud!

Link to Automation Avenue platform

Prerequisites:
active AWS account and optional but advised - AWS Budget configured.

Create an EC2 instance in AWS:
Go to EC2 service:
- Name - can be anything, let's say 'n8n'

Amazon Machine Image:
- choose Ubuntu Server 22.04 LTS (but 24.04 LTS and any Debian instance should work with that guide without any changes)
- Instance type - choose either t2.micro or t3.micro - whichever is currently displayed as 'Free Tier eligible'

Key pair:
- Its your SSH key, just click 'Create a new key pair', choose a name for your key pair and click 'Create'

Network Settings:
- Auto-assign public ip - make sure it's 'Enabled'
- Allow SSH traffic from - choose 'My IP'
- Allow HTTPS traffic from the Internet - make sure its 'ticked'

Configure Storage:
- change 8GB size to 30GB type GP3 (it should be within Free Tier threshold)
- Click 'Advanced' and make sure the encryption is enabled.

Click 'Launch instance' - your virtual server should be created within next few seconds.

Go to your instance, click 'Connect' , then 'SSH Client' and read the instruction.

You need to first locate the private SSH key that was automatically downloaded, and run:

```
chmod 400 "<key_name.pem>"
```

Then use the 'Example' to build your ssh command, it should be sth like:

```
ssh -i /home/${USER}/Downloads/<key_name.pem>
ubuntu@<publilc_ip_address_of_your_server>
```

Create alias so you don’t have to use that long command:

alias n8n=’ssh -i /home/${USER}/Downloads/<key_name.pem> ubuntu@<publilc_ip_address_of_your_server>’

From now on to log on to your server you can simply run command ‘n8n’ instead of that long command.

Once you are logged on - you can run ‘*df -h*’ command to check the disk size.

Next run command:
apt-get update && apt-get upgrade -y

This will update repositories and upgrade exiting programs that run on this server.

Reboot your server with command:
sudo reboot

Wait for 30 seconds and run ‘n8n’ command again to log back on.

Follow this guide from official n8n website:
https://docs.n8n.io/hosting/installation/server-setups/docker-compose/

Install Docker and Docker Compose (copy-paste below) :

```
# Remove incompatible or out of date Docker implementations if they exist
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker
containerd runc; do sudo apt-get remove $pkg; done
# Install prereq packages
sudo apt-get update
sudo apt-get install ca-certificates curl
# Download the repo signing key
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/
docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
# Configure the repository
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo
"${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | sudo tee /etc/apt/
sources.list.d/docker.list > /dev/null
# Update and install Docker and Docker Compose
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
```

Check if docker and docker compose were installed succesfully:

docker --version
docker compose version

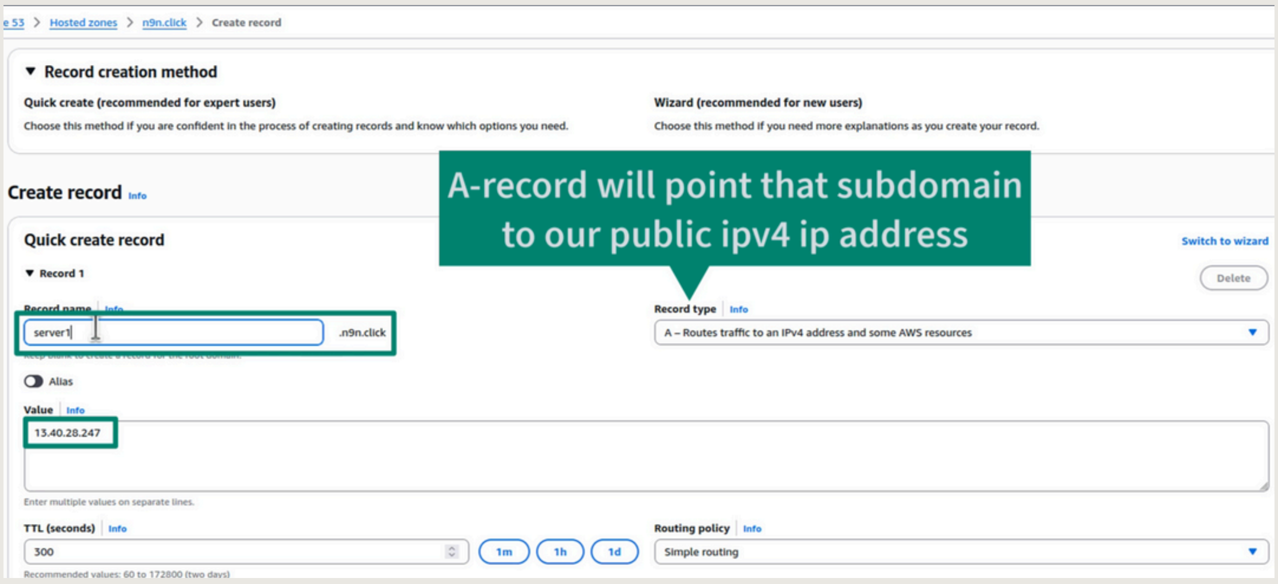
Add non-root user access:

sudo usermod -aG docker ${USER}
# Register the `docker` group memebership with current session without changing your primary group
exec sg docker newgrp

Purchase a domain in AWS Route53 service:

- Go to Route53 service
- Select 'Register a domain'
- Search for a domain
- Once you've found your domain - click 'Select' and complete purchase of the domain

Now go to Route53 - Hosted Zones - find your domain and add an A-record.
Choose a subdomain for your domain (for example - if my domain is n9n.click, I can add 'server1' in front as subdomain)
Add public ip address and click 'Create records':



A-record will point that subdomain to our public ipv4 ip address

e 53 > Hosted zones > n9n.click > Create record

▼ Record creation method

Quick create (recommended for expert users)
Choose this method if you are confident in the process of creating records and know which options you need.

Wizard (recommended for new users)
Choose this method if you need more explanations as you create your record.

Create record Info

Quick create record

▼ Record 1

Record name: server1 .n9n.click

Record type: A – Routes traffic to an IPv4 address and some AWS resources

Alias

Value: 13.40.28.247
Enter multiple values on separate lines.

TTL (seconds): 300
Recommended values: 60 to 172800 (two days)

Routing policy: Simple routing

Create a folder and change directory to be inside that folder:

```
mkdir n8n-compose
cd n8n-compose
```

Run *‘nano .env’* command and paste below:

```
# DOMAIN_NAME and SUBDOMAIN together determine where n8n will be reachable
from
# The top level domain to serve from
DOMAIN_NAME=example.com

# The subdomain to serve from
SUBDOMAIN=n8n

# The above example serve n8n at: https://n8n.example.com

# Optional timezone to set which gets used by Cron and other scheduling nodes
# New York is the default value if not set
GENERIC_TIMEZONE=Europe/Berlin

# The email address to use for the TLS/SSL certificate creation
SSL_EMAIL=user@example.com
```

You need to adjust the values above to match your domain and subdomain, for example my domain is ‘n9n.click’ and my subdomain is ‘server1'

Click Ctrl+O , Enter, Ctrl+X to save and exit nano.

Run *‘cat .env’* to check if the file is configured correctly.
Create another local-files folder inside n8n-compose folder you should be currently in:

```
mkdir local-files
```

Create docker compose file:
Run ‘*nano compose.yaml’* command and paste below:

```yaml
services:
  traefik:
    image: "traefik"
    restart: always
    command:
      - "--api.insecure=true"
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--entrypoints.web.address=:80"
      - "--entrypoints.web.http.redirections.entryPoint.to=websecure"
      - "--entrypoints.web.http.redirections.entrypoint.scheme=https"
      - "--entrypoints.websecure.address=:443"
      - "--certificatesresolvers.mytlschallenge.acme.tlschallenge=true"
      - "--certificatesresolvers.mytlschallenge.acme.email=${SSL_EMAIL}"
      - "--certificatesresolvers.mytlschallenge.acme.storage=/letsencrypt/acme.json"
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - traefik_data:/letsencrypt
      - /var/run/docker.sock:/var/run/docker.sock:ro

  n8n:
    image: docker.n8n.io/n8nio/n8n
    restart: always
    ports:
      - "127.0.0.1:5678:5678"
    labels:
      - traefik.enable=true
      - traefik.http.routers.n8n.rule=Host(`${SUBDOMAIN}.${DOMAIN_NAME}`)
      - traefik.http.routers.n8n.tls=true
      - traefik.http.routers.n8n.entrypoints=web,websecure
      - traefik.http.routers.n8n.tls.certresolver=mytlschallenge
      - traefik.http.middlewares.n8n.headers.SSLRedirect=true
      - traefik.http.middlewares.n8n.headers.STSSeconds=315360000
      - traefik.http.middlewares.n8n.headers.browserXSSFilter=true
      - traefik.http.middlewares.n8n.headers.contentTypeNosniff=true
      - traefik.http.middlewares.n8n.headers.forceSTSHeader=true
      - traefik.http.middlewares.n8n.headers.SSLHost=${DOMAIN_NAME}
      - traefik.http.middlewares.n8n.headers.STSIncludeSubdomains=true
      - traefik.http.middlewares.n8n.headers.STSPreload=true
      - traefik.http.routers.n8n.middlewares=n8n@docker
    environment:
      - N8N_HOST=${SUBDOMAIN}.${DOMAIN_NAME}
      - N8N_PORT=5678
      - N8N_PROTOCOL=https
      - NODE_ENV=production
      - WEBHOOK_URL=https://${SUBDOMAIN}.${DOMAIN_NAME}/
      - GENERIC_TIMEZONE=${GENERIC_TIMEZONE}
    volumes:
      - n8n_data:/home/node/.n8n
      - ./local-files:/files

volumes:
  n8n_data:
  traefik_data:
```

You can now start your docker service by running:
sudo docker compose up -d

This might take several minutes before it starts when you run this command for the first time as all images have to be downloaded first.
It will be much quicker on any subsequent run though.
To stop the service , you can run :

sudo docker compose stop