## Guia completo (AWS + Terraform + Ansible + DuckDNS + Nginx + HTTPS + n8n)

Este documento é um guia passo a passo para qualquer pessoa subir uma VM na AWS e deixar **Backend + n8n** funcionando com **domínio (DuckDNS)** e **HTTPS (Certbot)**.

Se você travar em um passo e o troubleshooting não resolver, pergunte para uma IA (ChatGPT/afins) colando:
- o comando que você rodou
- o erro completo
- seu sistema operacional (Mac/Windows/Linux)

### Resultado final

- Backend: `https://SEU_DOMINIO_BACKEND/health` → `200 OK`
- n8n UI: `https://SEU_DOMINIO_N8N/` → pede usuário/senha
- Webhooks do n8n: `https://SEU_DOMINIO_N8N/webhook/...` → público (não pede senha; pode retornar `404` se o path não existir ainda)

---

## 0) Pré-requisitos (na sua máquina)

- Conta AWS ativa
- AWS CLI instalado e configurado (`aws configure`)
- Terraform instalado
- Ansible instalado
- Chave SSH criada:

```bash
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub
```

---

## 1) Subir a VM (Terraform)

Entre em:

```bash
cd onboarding/aws/terraform
cp terraform.tfvars.example terraform.tfvars
```

Edite `terraform.tfvars` e preencha:
- `aws_region`
- `ssh_public_key` (cole o conteúdo de `~/.ssh/id_rsa.pub`)
- `ansible_user` (Amazon Linux 2023: `ec2-user`)

Depois:

```bash
terraform init
terraform apply
```

Anote o **IP público** da VM.

---

## 2) Preparar o Ansible (inventário + variáveis)

Entre em:

```bash
cd onboarding/ansible
cp inventory.ini.example inventory.ini
cp group_vars/all.yml.example group_vars/all.yml
```

Edite `inventory.ini` e substitua `SEU_IP_OU_HOST_AQUI` pelo IP público da VM.

Edite `group_vars/all.yml` e preencha:
- Firebase: `firebase_project_id`, `firebase_client_email`, `firebase_private_key`
- Google OAuth (se usar): `google_client_id`, `google_client_secret`, `google_redirect_uri`
- CORS: `cors_origin` (URL do frontend no Firebase)
- n8n: `n8n_password`, `n8n_host`, `n8n_protocol`, `n8n_webhook_url`

---

## 3) Provisionar (Ansible)

Prévia (não aplica):

```bash
ansible-playbook playbook.app.yml --check
```

Aplicar:

```bash
ansible-playbook playbook.app.yml
```

Validar na VM (via SSH):

```bash
curl -i http://localhost:3000/health | head -n 10
curl -i http://localhost:5678/healthz | head -n 10
```

---

## 4) DuckDNS (domínio grátis)

No site `https://www.duckdns.org`:
- crie um domínio para o backend: `SEU_DOMINIO_BACKEND.duckdns.org`
- crie um domínio para o n8n: `SEU_DOMINIO_N8N.duckdns.org`
- aponte ambos para o IP público da VM

Validar DNS na sua máquina:

```bash
dig +short SEU_DOMINIO_BACKEND.duckdns.org
dig +short SEU_DOMINIO_N8N.duckdns.org
```

### Atualização automática do IP (recomendado)

Na VM (Amazon Linux 2023), use `systemd timer` (substitua `SEU_TOKEN_AQUI` e `SEU_DOMINIO`):

```bash
sudo tee /etc/systemd/system/duckdns-update.service >/dev/null <<'EOF'
[Unit]
Description=DuckDNS updater

[Service]
Type=oneshot
ExecStart=/usr/bin/curl -fsS https://www.duckdns.org/update?domains=SEU_DOMINIO&
token=SEU_TOKEN_AQUI&ip=
EOF

sudo tee /etc/systemd/system/duckdns-update.timer >/dev/null <<'EOF'
[Unit]
Description=Run DuckDNS updater every 5 minutes

[Timer]
OnBootSec=1min
OnUnitActiveSec=5min
Unit=duckdns-update.service

[Install]
WantedBy=timers.target
EOF

sudo systemctl daemon-reload
sudo systemctl enable --now duckdns-update.timer
sudo systemctl status duckdns-update.timer --no-pager -l
```

---

## 5) Nginx (reverse proxy) + senha na UI do n8n

Por que?
- você expõe só **80/443** para a internet
- mantém backend/n8n em portas internas
- consegue colocar **HTTPS** com certificado grátis

Na VM:

```bash
sudo dnf install -y nginx httpd-tools
sudo systemctl enable --now nginx
sudo htpasswd -c /etc/nginx/.htpasswd admin
```

Config do Nginx (ajuste os domínios):

```bash
sudo tee /etc/nginx/conf.d/agenda-calendar.conf >/dev/null <<'EOF'
server {
  listen 80;
  server_name SEU_DOMINIO_BACKEND;

  location / {
    proxy_pass http://127.0.0.1:3000;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
```

```
}

server {
  listen 80;
  server_name SEU_DOMINIO_N8N;

  location ~ ^/(webhook|webhook-test)/ {
    proxy_pass http://127.0.0.1:5678;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }

  location / {
    auth_basic "Restricted";
    auth_basic_user_file /etc/nginx/.htpasswd;

    proxy_pass http://127.0.0.1:5678;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
}
EOF

sudo nginx -t
sudo systemctl reload nginx
```

Testes (HTTP):

```bash
curl -i http://SEU_DOMINIO_BACKEND/health | head -n 10
curl -i http://SEU_DOMINIO_N8N/webhook-test/qualquercoisa | head -n 10
```

---

## 6) HTTPS com Certbot (Let’s Encrypt)

Na VM:

```bash
sudo dnf install -y certbot python3-certbot-nginx
sudo certbot --nginx -d SEU_DOMINIO_BACKEND -d SEU_DOMINIO_N8N
```

Validar:

```bash
curl -i https://SEU_DOMINIO_BACKEND/health | head -n 10
curl -I https://SEU_DOMINIO_N8N/ | head -n 5
```

---

## 7) Segurança (AWS Security Group)

Depois do Nginx+HTTPS:
- mantenha inbound: **22, 80, 443**

- remova inbound: **3000, 5678**

---

## Troubleshooting (dúvidas comuns)

### “Ansible não conecta na VM”
- confira IP público no `inventory.ini`
- confira usuário (`ec2-user` em Amazon Linux)
- confira a chave `ansible_ssh_private_key_file`
- tente `ssh -i ~/.ssh/id_rsa ec2-user@SEU_IP`

### “Certbot falhou / não conseguiu validar domínio”
- `dig +short SEU_DOMINIO` precisa apontar para o IP público
- portas **80/443** precisam estar abertas no Security Group
- Nginx precisa estar rodando: `sudo systemctl status nginx`

### “n8n pede senha e minha Kiwify não consegue chamar”
- webhooks devem estar em `/webhook/...` (sem senha)
- a UI fica com senha, mas webhooks ficam públicos

### “curl no webhook dá 404”
Normal se você testou um path que ainda não existe no n8n.