Usage instructions:

1. Launch the product via 1-click. **Please wait until** the instance passes **all** status checks and is running. You can connect using your Amazon private key and '**ubuntu**' login via your SSH client.

   To update software, use: **sudo apt-get update -y**

2. **Wait for first boot to complete**

Authentik starts automatically on first boot. Wait about 2 to 5 minutes after the instance is running.

3. Open Authentik in your browser

First time setup:

- On a fresh instance, the Welcome page may automatically take you to the setup flow:

  http://YOUR_INSTANCE_PUBLIC_IP:9000/if/flow/initial-setup/

- Follow the prompts to create your initial admin credentials and password.

After setup:

- You will return to the main login page at:

  http://YOUR_INSTANCE_PUBLIC_IP:9000/

Fallback If Setup URL Says Not Found

If the setup flow link shows Not Found, you can still set the admin password and log in.

1. SSH to the instance
2. Run:

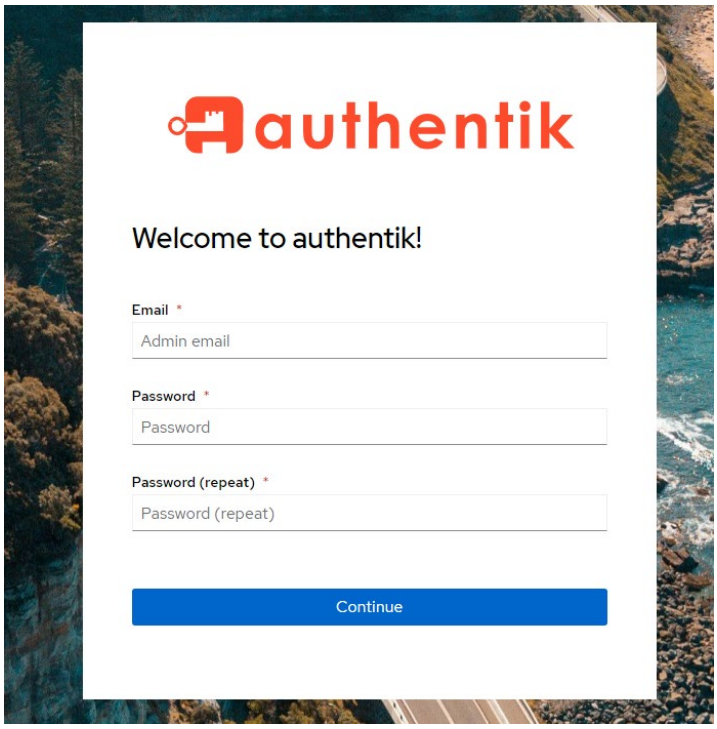
cd /home/ubuntu/.docker/compose/authentik

sudo docker compose exec server ak changepassword akadmin

3. Then log in at:

http://YOUR_INSTANCE_PUBLIC_IP:9000/if/flow/default-authentication-flow/?next=%2F

Username: akadmin
Password: the one you just set



Note: Complete the registration. If you have difficulty, refresh your web page.

Optional: Add a Custom Domain and Trusted SSL (HTTPS on Ports 80 and 443) Ubuntu 22.04

Prerequisites

1. You own a domain name.
2. Your domain's DNS record points to this instance's public IP.

DNS setup

Create an **A record**:

auth.example.com → YOUR_INSTANCE_PUBLIC_IP

Wait for DNS to propagate.

Step 1: Open required ports in the Security Group

Required:

- TCP **80** from 0.0.0.0/0

- TCP **443** from 0.0.0.0/0

Optional:

- TCP **22** (SSH), restrict to your IP

After HTTPS works, you may remove public access to TCP **9000**.

Step 2: Install Nginx and Certbot

SSH into the instance and run:

```
sudo apt update
sudo apt install -y nginx certbot python3-certbot-nginx
sudo systemctl enable --now nginx
```

Step 3: Create an Nginx reverse proxy for Authentik

Create the site file:

```
sudo nano /etc/nginx/sites-available/authentik
```

Paste this and replace auth.example.com with your domain:

```
server {
    listen 80;
    server_name auth.example.com;

    location / {
        proxy_pass http://127.0.0.1:9000;
        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;
    }
}
```

Enable the site:

```
sudo ln -s /etc/nginx/sites-available/authentik /etc/nginx/sites-enabled/authentik
sudo nginx -t
sudo systemctl reload nginx
```

Test HTTP (should load Authentik without :9000):

http://auth.example.com

Step 4: Enable HTTPS with a trusted SSL certificate

Run Certbot:

sudo certbot --nginx

Follow the prompts:

- Select your domain
- Choose the option to **redirect HTTP to HTTPS** when asked

Step 5: Access Authentik via HTTPS

Open:

https://auth.example.com

No port number required.

Automatic SSL Renewal

Certbot installs renewal automatically. You can verify:

systemctl list-timers | grep certbot

For more configurations, see documentation:

https://docs.goauthentik.io/docs/installation/configuration#about-authentik-configurations

AWS Data

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized_keys & /home/ubuntu/.ssh/authorized_keys
- Monitor the health:
    - Navigate to your Amazon EC2 console and verify that you're in the correct region.
    - Choose Instance and select your launched instance.
    - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

Extra Information: **(Optional)**

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html