Open in Onyx

Deploy Onyx on Digital Ocean Droplets

1 **Create a Droplet**

Create a Droplet with the appropriate resources. For this guide, we will use a recommended configuration with Docker pre-installed.

> ⊕ Read our **Resourcing guide** for more details.

Give your Droplet a descriptive name like `onyx-prod`

Select a region close to your users

Select the `Docker` image from the *Marketplace* tab

Choose a Droplet size with at least 4 vCPUs and 16GB RAM

Configure storage following the Resourcing Guide

Add your SSH key for secure access

2 **Create the Droplet**

Click Create Droplet and then view your Droplet details.

> 💡 Save the **Public IP address** of the Droplet!

3 **Point domain to the instance**

> ⊕ If you don't have a domain, buy one from a DNS provider like **GoDaddy** or just skip HTTPS for now.

To point our domain to the new instance, we need to add an `A` and `CNAME` record to our DNS provider.

The `A` record should be the subdomain that you would like to use for the Onyx instance like `prod`.

The `CNAME` record should be the same name with the `www.` in front resulting in `www.prod` pointing to the full domain like `prod.onyx.app` .

A records use an IP address to connect your domain to a website. They're also used to create subdomains such as www or store, that point to an IP address.

Type *	Name *	Value *	TTL
A ⌄	@	<YOUR_INSTANCE_IP>	Custom ⌄
			Seconds
			600

Save   Close

CNAME records are a type of subdomain, or alias, that points to another domain name.

Type *	Name *	Value *	TTL
CNAME ⌄	www	<YOUR_DOMAIN>	1 Hour ⌄

Save   Close

4   **Install Onyx requirements**

Since we selected the Docker Marketplace image, Docker and Docker Compose are already installed.

We just need to install `git` :

```
sudo apt update
sudo apt install -y git
```

5   **Install and Configure Onyx**

To install Onyx, we'll need to clone the repo and set the necessary environment variables.

```
git clone --depth 1 https://github.com/onyx-dot-app/onyx.git
```

```
cd onyx/deployment/docker_compose
cp env.prod.template .env
cp env.nginx.template .env.nginx
```

Fill out the `.env` and `.env.nginx` files.

.env

```
WEB_DOMAIN=<YOUR_DOMAIN>  # Something like "onyx.app"

# If your email is something like "chris@onyx.app", then this should be "onyx.app"
# This prevents people outside your company from creating an account
VALID_EMAIL_DOMAINS=<YOUR_COMPANIES_EMAIL_DOMAIN>

# See our auth guides for options here
```

••• See all 8 lines

.env.nginx

```
DOMAIN=<YOUR_DOMAIN>  # Something like "onyx.app"
```

6 **Launch Onyx**

Running the `init-letsencrypt.sh` script will get us a SSL certificate from letsencrypt and launch the Onyx stack.

```
./init-letsencrypt.sh
```

> ⚠️ You will hit an error if you fail the letsencrypt workflow more than 5 times. You will need to wait 72 hours or request a new domain.

If you are skipping the HTTPS setup, start Onyx manually:

```
docker compose -f docker-compose.dev.yml -p onyx-stack up -d --build --for
```

> ⚠ Give Onyx a few minutes to start up.
>
> You can monitor the progress with `docker logs onyx-stack-api_server-1 -f`.

You can access Onyx from the instance Public IPv4 or from the domain you set up earlier!

```
docker compose -f docker-compose.dev.yml -p onyx-stack up -d --build --for
```