Blockscout

Manual Deployment

General deployment instructions for a hardware or cloud services environment

> For automated deployment on AWS, see Ansible deployment.

> For local deployment only using Docker, see Docker Integration.

Manual Deployment

Check your environment is prepared with General Requirements and Database Storage Requirements.

BlockScout requires a **full archive node** in order to import every state change for every address on the target network. For client specific settings related to a node running Erigon/Geth/Nethermind, please see Client Settings.

> For testing purposes, instead of an archive node, a test Ethereum client can be used. For instance, ganache-cli

Deployment Steps

1) `git clone https://github.com/blockscout/blockscout`

2) `cd blockscout`

3) Provide DB URL:
`export DATABASE_URL=postgresql://user:password@localhost:5432/blockscout`

- **Linux:** Update the database username and password configuration
- **Mac:** Use logged-in user name and empty password
- **Optional:** Change credentials in `apps/explorer/config/test.exs` for test env

> *Example usage:* Changing the default Postgres port from localhost:5432 if Boxen is installed.

4) Install Mix dependencies and compile them `mix do deps.get, local.rebar --force, deps.compile`

5) Generate a new secret_key_base for the DB by setting a corresponding ENV var:
`export SECRET_KEY_BASE=VTIB3uHDNbvrY0+60ZWgUoUBKDn9ppLR8MI4CpRz4/qLyEFs54ktJfaNT6Z221No`

ⓘ In order to generate a new `secret_key_base` run `mix phx.gen.secret`

6) If you have deployed previously, remove static assets from the previous build `mix phx.digest.clean`.

7) Set environment variables as needed.

CLI Example:

```
export ETHEREUM_JSONRPC_VARIANT=nethermind
export ETHEREUM_JSONRPC_HTTP_URL=http://localhost:8545
export DATABASE_URL=postgresql://...
export COIN=DAI
export MIX_ENV=prod
export ...
```

ⓘ It is important to set the variable **`MIX_ENV=prod`** during deployment. The current default is `MIX_ENV=dev` which is a slower and less secure setting.

ⓘ The `ETHEREUM_JSONRPC_VARIANT` will vary depending on your client (nethermind, geth etc). More information on client settings.

8) Install and start the smart contract verification microservice. You can use docker, build from source, or use cargo directly (example below). If you experience issues, see the extensive smart contract verifier readme.

1. Using docker:
   - `docker run -p 8050:8050 ghcr.io/blockscout/smart-contract-verifier:latest`
2. Or install rust and build from sources:
   - `cargo install --locked --git https://github.com/blockscout/blockscout-rs smart-contract-verifier-server`
   - Then run the binary as `smart-contract-verifier-server`
3. Set ENV variables in CLI to enable the rust microservice for Blockscout (these can also be set at runtime).

```
export MICROSERVICE_SC_VERIFIER_ENABLED=true
export MICROSERVICE_SC_VERIFIER_URL=http://0.0.0.0:8050/
```

9) Compile the application: `mix compile`

10) If not already running, start Postgres: `pg_ctl -D /usr/local/var/postgres start`

> To check postgres status: `pg_isready`

11) Create and migrate database `mix do ecto.create, ecto.migrate`

> If you are in dev environment and have run the application previously with a different blockchain, drop the previous database `mix do ecto.drop, ecto.create, ecto.migrate`
> Be careful since it will delete all data from the DB. Don't execute it on production if you don't want to lose all the data!

12) Install Node.js dependencies

> *Optional: If preferred, use `npm ci` rather than `npm install` to strictly follow all package versions in `package-lock.json`.*

- `cd apps/block_scout_web/assets; npm install && node_modules/webpack/bin/webpack.js --mode production; cd -`
- `cd apps/explorer && npm install; cd -`

13) Build static assets for deployment `mix phx.digest`

14) Enable HTTPS in development. The Phoenix server only runs with HTTPS.

- `cd apps/block_scout_web; mix phx.gen.cert blockscout blockscout.local; cd -`
- Add blockscout and blockscout.local to your `/etc/hosts`

```
127.0.0.1       localhost blockscout blockscout.local

255.255.255.255 broadcasthost

::1             localhost blockscout blockscout.local
```

> If using Chrome, Enable `chrome://flags/#allow-insecure-localhost`

15) Return to the root directory and start the Phoenix Server. `mix phx.server`