# Variables

ENV variables must be configured per chain

Configuration

There are three groups of variables required to build BlockScout. The first is required to create infrastructure, the second to build BlockScout instances and the third is required both for infra and BS itself.

For your convenience we have divided variable templates into three files accordingly -infrastructure.yml.example ,blockscout.yml.example andall.yml.example . Also we have divided those files to place them in thegroup_vars and inhost_vars folders, so you will not have to repeat some of the variables for each host/group.

To deploy BlockScout, you will setup the following set of files for each instance:

```

Copy / | - group_vars | | - group.yml (combination of [blockscout+infrastructure+all].yml.example) | | - all.yml (optional, one for all instances) | - host_vars | | - host.yml (combination of [blockscout+infrastructure+all].yml.example) | - hosts (one for all instances)

```

The subsections below describe the variables you may want to adjust.

- Infrastructure Variables
- BlockScout Variables
- Common (All) Variables
-

Infrastructure Variables

- terraform_location
- is an address of the Terraform binary on the builder;
- dynamodb_table
- represents the name of table that will be used for Terraform state lock management;
- lfec2_ssh_key_content
- variable is not empty, Terraform will try to create EC2 SSH key with theec2_ssh_key_name
- name. Otherwise, the existing key withec2_ssh_key_name
- name will be used;
- instance_type
- defines a size of the Blockscout instance that will be launched during the deployment process;
- vpc_cidr
- ,public_subnet_cidr
- ,db_subnet_cidr
- represents the network configuration for the deployment. Usually you want to leave it as is. However, if you want to modify it, please, expect thatdb_subnet_cidr
- represents not a single network, but a group of networks started with defined CIDR block increased by 8 bits.
-

Example: Number of networks: 2db_subnet_cidr : "10.0.1.0/16" Real networks: 10.0.1.0/24 and 10.0.2.0/24 * An internal DNS zone withdns_zone_name * name will be created to take care of BlockScout internal communications; * Theroot_block_size * is the amount of storage on your EC2 instance. This value can be adjusted by how frequently logs are rotated. Logs are located in/opt/app/logs * of your EC2 instance; * Each of thedb_* * variables configures the database for each chain. Each chain will have the separate RDS instance; * instance_type * represent the size of the EC2 instance to be deployed in production; * use_placement_group * determines whether or not to launch BlockScout in a placement group. *

BlockScout Variables

- blockscout_repo
-
    - a direct link to the BlockScout repo;
- branch
-
    - maps branch atblockscout_repo
- to each chain;
- Specify themerge_commit
- variable if you want to merge any of the specifiedchains
- with the commit in the other branch. Usually used to update production branches with the releases from master

branch;
- skip_fetch
-
    - if this variable is set totrue
- , BlockScout repo will not be cloned and the process will start from building the dependencies. Use this variable to prevent playbooks from overriding manual changes in cloned repo;
- ps_*
- variables represents a connection details to the test Postgres database. Thisis not installed automatically
- , so make sureps_*
- credentials are valid before starting the deployment;
-

Common (All) Variables

- ansible_host
-
    - is an address where BlockScout will be built. If this variable is set to localhost, also setansible_connection
- tolocal
- for better performance.
- chain
- variable set the name of the network (Kovan, Core, xDAI, etc.). Will be used as part of the infrastructure resource names.
- env_vars
- represents the set of environment variables used by BlockScout.They are available here.
-
    - One can defineBULD_*
-
    - set of the variables, where asterisk stands for any environment variables. All variables defined withBUILD_*
-
    - will override default variables while building the dev server.
- *
- aws_access_key
- andaws_secret_key
- is a credentials pair that provides access to AWS for the deployer; You can use theaws_profile
- instead. In that case, AWS CLI profile will be used. Also, if none of the access key and profile provided, thedefault
- AWS profile will be used. Theaws_region
- should be left atus-east-1
- as some of the other regions fail for different reasons;
- backend
- variable defines whether deployer should keep state files remote or locally. Setbackend
- variable totrue
- if you want to save state file to the remote S3 bucket;
- upload_config_to_s3
-
    - set totrue
- if you want to upload configall.yml
- file to the S3 bucket automatically after the deployment. Will not work ifbackend
- is set to false;
- upload_debug_info_to_s3
-
    - set totrue
- if you want to upload full log output to the S3 bucket automatically after the deployment. Will not work ifbackend
- is set to false.
-

Locally logs are stored atlog.txt which is not cleaned automatically. Do not forget to clean manually or use theclean.yml playbook * bucket * represents a globally unique name of the bucket where your configs and state will be stored. It will be created automatically during the deployment; *

a chain name SHOULD NOT be more than 5 characters. Otherwise, it will throw an error because the aws load balancer name should not be greater than 32 characters.

Last updated5 months ago