VM setup - Create host directories and files

Create directory with secrets

For security reasons, sensitive info (such as tokens and passwords) are not hardcoded into the docker image, nor passed as environment variables at runtime. They are passed to the docker container from the host VM as files inside a directory. Each file's name is the name of the variable and the file content is the value. These are read from inside the running container when necessary.

More info on how to create `secrets` directory and files can be found here.

Create directory for build artifacts

The build artifacts should be kept on a directory outside the docker container, so it is easier to replace the container without losing the builds. For portability across VMs a persistent disk can be used (as described here).

Note: The directories created inside that directory will be owned by user `www-data`.

Create SSL certificates (Optional for dev)

The host VM can attach a directory containing the SSL certificate and key to be used by the nginx server for serving the hosted previews. More info on how to attach the directory when starting the container can be found here.

In order for the container to be able to find the certificate and key, they should be named `<DOMAIN_NAME>.crt` and `<DOMAIN_NAME>.key` respectively. For example, for a domain name `ngbuild.io`, nginx will look for files `ngbuilds.io.crt` and `ngbuilds.io.key`. More info on how to specify the domain name see here.

If no directory is attached, nginx will use an internal self-signed certificate. This is convenient during development, but is not suitable for production.

Note: Since nginx needs to be able to serve requests for both the main domain as well as any subdomain (e.g. `ngbuilds.io/` and `foo-bar.ngbuilds.io/`), the provided certificate needs to be a wildcard certificate covering both the domain and subdomains.

Create directory for logs (Optional)

Optionally, a logs directory can pe passed to the docker container for storing non-system-related logs. If not provided, the logs are kept locally on the container and will be lost whenever the container is replaced (e.g. when updating to use a newer version of the docker image). Log files are rotated and retained for 6 months.

The following log files are kept in this directory:

- `clean-up.log`: Output of the `aio-clean-up` command, run as a cronjob for cleaning up the build artifacts of closed PRs.
- `init.log`: Output of the `aio-init` command, run (by default) when starting the container.
- `nginx/{access,error}.log`: The access and error logs produced by the nginx server while serving "production" files.
- `nginx-test/{access,error}.log`: The access and error logs produced by the nginx server while serving "test" files. This is only used when running tests locally from inside the container, e.g. with the `aio-verify-setup` command. (See here for more info.)
- `preview-server-{prod,test,verify-setup}-*.log`: The logs produced by the Node.js preview-server while serving either:
  - `-prod`: "Production" files (g.g during normal operation).
  - `-test`: "Test" files (e.g. when a test instance is started with the `aio-preview-server-test` command).
  - `-verify-setup`: "Test" files, but while running `aio-verify-setup`.

  (See here for more info the commands mentioned above.)
- `verify-setup.log`: The output of the `aio-verify-setup` command (e.g. Jasmine output), except for preview-server output which is logged to `preview-server-verify-setup-*.log` (see above).