VM setup - Start docker container

The `docker run` command

Once everything has been setup and configured, a docker container can be started with the following command:

```
sudo docker run \
  --detach \
  --dns 127.0.0.1 \
  --name <instance-name> \
  --publish 80:80 \
  --publish 443:443 \
  --restart unless-stopped \
  --volume <host-secrets-dir>:/aio-secrets:ro \
  --volume <host-builds-dir>:/var/www/aio-builds \
 [--volume <host-cert-dir>:/etc/ssl/localcerts:ro] \
 [--volume <host-logs-dir>:/var/log/aio] \
 [--volume <host-dockerbuild-dir>:/dockerbuild] \
  <name>[:<tag>]
```

Below is the same command with inline comments explaining each option. The API docs for `docker run` can be found here.

```
sudo docker run \

  # Start as a daemon.
  --detach \

  # Use the local DNS server.
  # (This is necessary for mapping internal URLs, e.g. for the Node.js preview-
server.)
  --dns 127.0.0.1 \

  # USe `<instance-name>` as an alias for the container.
  # Useful for running `docker` commands, e.g.: `docker stop <instance-name>`
  --name <instance-name> \

  # Map ports of the host VM (left) to ports of the docker container (right)
  --publish 80:80 \
  --publish 443:443 \

  # Automatically restart the container (unless it was explicitly stopped by the
user).
  # (This ensures that the container will be automatically started on boot.)
  --restart unless-stopped \

  # The directory the contains the secrets (e.g. GitHub token, JWT secret, etc).
  # (See [here](vm-setup--set-up-secrets.md) for more info.)
  --volume <host-secrets-dir>:/aio-secrets:ro \

  # The build artifacts and hosted previews will stored to and served from this
```

```
directory.
  # (If you are using a persistent disk - as described [here](vm-setup--attach-
persistent-disk.md) -
  #  this will be a directory inside the disk.)
  --volume <host-builds-dir>:/var/www/aio-builds \

  # The directory the contains the SSL certificates.
  # (See [here](vm-setup--create-host-dirs-and-files.md) for more info.)
  # If not provided, the container will use self-signed certificates.
 [--volume <host-cert-dir>:/etc/ssl/localcerts:ro] \

  # The directory where the logs are being kept.
  # (See [here](vm-setup--create-host-dirs-and-files.md) for more info.)
  # If not provided, the logs will be kept inside the container, which means they will
be lost
  # whenever a new container is created.
 [--volume <host-logs-dir>:/var/log/aio] \

  # This directory allows you to share the source scripts between the host and the
container when
  # debugging. (See [here](misc--debug-docker-container.md) for how to set this up.)
 [--volume <host-dockerbuild-dir>:/dockerbuild] \

  # The name of the docker image to use (and an optional tag; defaults to `latest`).
  # (See [here](vm-setup--create-docker-image.md) for instructions on how to create
the image.)
  <name>[:<tag>]
```

Example

The following command would start a docker container based on the previously created `foobar-builds` docker image, alias it as 'foobar-builds-1' and map predefined directories on the host VM to be used by the container for accessing secrets and SSL certificates and keeping the build artifacts and logs; and will map the source scripts from the host to the container.

```
sudo docker run \
  --detach \
  --dns 127.0.0.1 \
  --name foobar-builds-1 \
  --publish 80:80 \
  --publish 443:443 \
  --restart unless-stopped \
  --volume /foobar-secrets:/aio-secrets:ro \
  --volume /mnt/disks/foobar-builds:/var/www/aio-builds \
  --volume /etc/ssl/localcerts:/etc/ssl/localcerts:ro \
  --volume /foobar-logs:/var/log/aio \
  --volume ~/angular/aio/aio-builds-setup/dockerbuild:/dockerbuild \
  foobar-builds
```