Setting Up a Multi-Container BigBlueButton Server Using Docker and Docker-Compose

# Overview

In this document you will find the prerequisitesinstallation steps, instructions, and minimum requirements to get a BigBlueButton server running the HTML5 client in Docker, with components split up into individual containers.

# Prerequisites

## Ubuntu 16.04 Server, locally or via AWS EC2

* 4GB RAM
* Quad Core CPU
* TCP Ports 80, 443, 3478
* UDP Port 3478

## Docker-CE + post-install steps

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

<https://docs.docker.com/install/linux/linux-postinstall/>

## Docker Compose

sudo apt-get install docker-compose

## BigBlueButton Repository

git clone -b multi\_container\_docker --single-branch <https://github.com/Seneca-CDOT/bigbluebutton.git>

# Setup

## Run the Makefile to build the docker images

cd ~/bigbluebutton/labs/docker

make release

## Export necessary environment variables

**IMPORTANT:** replace the example SERVER\_DOMAIN value with your own FQDN. If you decide to shut down or reboot your server at any point, you’ll need to recreate these environment variables again.

export SERVER\_DOMAIN=docker.bigbluebutton.org

export EXTERNAL\_IP=$(dig +short $SERVER\_DOMAIN | grep '^[0-9]\*\.[0-9]\*\.[0-9]\*\.[0-9]\*$' | head -n 1)

export SHARED\_SECRET=`openssl rand -hex 16`

export COTURN\_REST\_SECRET=`openssl rand -hex 16`

export SECRET\_KEY\_BASE=`openssl rand -hex 64`

export SCREENSHARE\_EXTENSION\_KEY=akgoaoikmbmhcopjgakkcepdgdgkjfbc

export SCREENSHARE\_EXTENSION\_LINK=https://chrome.google.com/webstore/detail/bigbluebutton-screenshare/akgoaoikmbmhcopjgakkcepdgdgkjfbc

## Create a volume for the SSL certs and generate them using Let’s Encrypt

docker volume create docker\_ssl-conf

docker run --rm -p 80:80 -v docker\_ssl-conf:/etc/letsencrypt -it certbot/certbot certonly --non-interactive --register-unsafely-without-email --agree-tos --expand --domain $SERVER\_DOMAIN --standalone

**IMPORTANT:** If running on AWS, you won't be able to use the default Public DNS for your SERVER\_DOMAIN as Let's Encrypt doesn't allow generating SSL certs from any \*.amazonaws.com domain. Alternatively, you can create a PTR record that goes from a non-AWS FQDN to the AWS FQDN.

## Create a volume for the static files (optional)

docker volume create docker\_static

cd ~/bigbluebutton/bigbluebutton-config/web/

docker run -d --rm --name nginx -v docker\_static:/var/www/bigbluebutton-default nginx tail -f /dev/null

docker cp . nginx:/var/www/bigbluebutton-default

docker exec -it nginx chown -R www-data:www-data /var/www/bigbluebutton-default

docker stop nginx

# Run

## Launch all containers using docker-compose

cd ~/bigbluebutton/labs/docker/

docker-compose up

## Access your server, create meeting rooms, begin sessions

Navigate to https://<your\_fqdn\_here>/b in your browser to use the Greenlight front-end

## Shut down containers and exit gracefully (stops BigBlueButton)

CTRL-C