Installing and Scaling Web Conferencing Platform

BigBlueButton

Mohamed Ali Ahmed

Installing and Scaling BigBlueButton

Outline

- Introduction to BigBlueButton
- BBB installation choices
- BBB minimum server requirements
- BBB front-end
- Turn Server
- Introduction to Scalelite
- Architecture of Scalelite
- NFS shared volume
- PostgreSQL setup
- Redis Cache setup
- Deploying Scalelite Containers

Installing BigBlueButton

What is BBB?

- BigBlueButton is an open source web conferencing system for online learning.
- Open source - you have full access to BigBlueButton's source code under an open source license. With the source code, installation steps, and community support, you can easily deploy your own BigBlueButton server (or 10 servers if you want). For each server you can customize it, modify it and integrate it into your products and services. Cool.
- Web conferencing system - you get the core features you would expect from a commercial web conferencing system (but under an open source license). These features include real-time sharing of audio, video, presentation, and screen – along with collaboration tools such as whiteboard, shared notes, polling, and breakout rooms. BigBlueButton can record your sessions for later playback.
- Online learning - BigBlueButton extends these core features to enable a teacher to engage students for learning.

Installing BigBlueButton

Overview

- BigBlueButton is an HTML5-based web application. Unlike many commercial web conferencing systems that require you to install software, BigBlueButton runs within your web browser. You click a link (such as in Greenlight), your browser runs BigBlueButton and prompts you to join the audio bridge. There is no plugin to download, no software to install. BigBlueButton provides high-quality audio, video, and screen sharing using the browser's built-in support for web real-time communication (WebRTC) libraries.
- WebRTC is a standard supported by all major browsers, including Chrome, FireFox, Safari, and Safari Mobile. For best results on desktop and laptops, we recommend Chrome or Firefox.

Installing BigBlueButton

Installation choices

- When installing BigBlueButton you have two choices: bbb-install.sh and step-by-step.
- Regardless of which choice you make, to have a successful installation you need to:
  - obtain a dedicated server,
  - ensure the server meets BigBlueButton's minimum set of requirements,
  - assign a hostname (recommended to set up SSL), and
  - configure the server's firewall (if needed).

Installing BigBlueButton

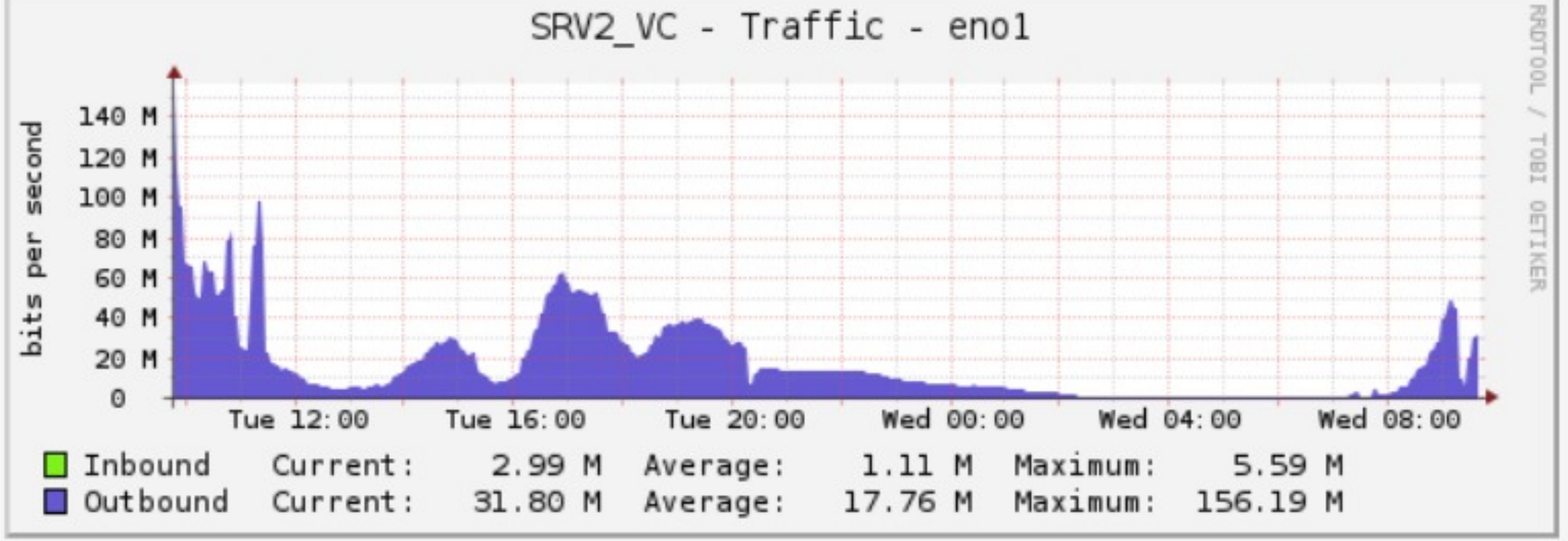
Minimum Server Requirements

The minimum requirements for a BigBlueButton server are:

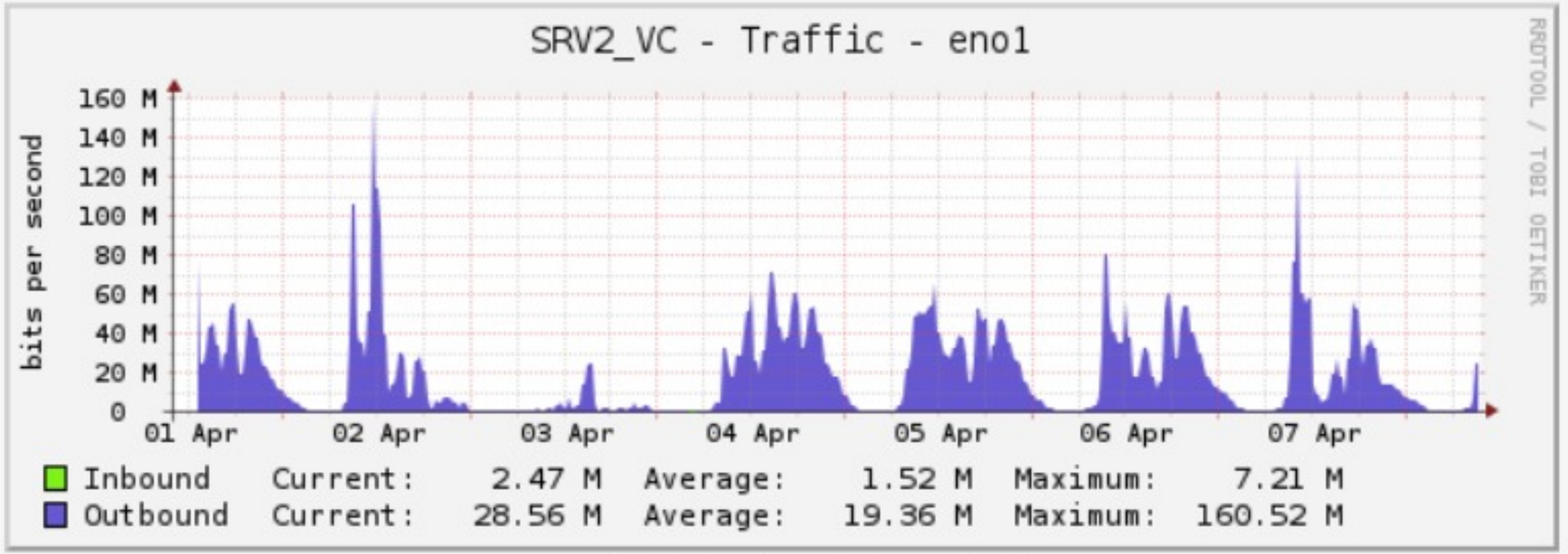
- Ubuntu 16.04 64-bit OS running Linux kernel 4.x
- 8 GB of memory with swap enabled (16 GB of memory is better)
- 4 CPU cores (8 is better)
- TCP ports 80 and 443 are accessible
- UDP ports 16384 - 32768 are accessible
- Port 80 is not in use by another application

For a server intended for production, it is additionally recommended:

- 500G of free disk space (or more) for recordings
- 250 Mbits/sec bandwidth (symmetrical) or more(In our case, the VC used more upload than download, See next slide)
- Dedicated (bare metal) hardware
- A hostname (such as bbb.example.com) for setup of a SSL certificate
- IPV4 and IPV6 address



Daily (5 Minute Average)



Weekly (30 Minute Average)

Installing BigBlueButton

bbb-install

Before running bbb-install.sh, it is strongly recommended that you:

- Ensure that your server meets the minimal server requirements
- Configure a fully qualified domain name (FQDN), such as bbb.example.com, that resolves to the external IP address of the server.
- Configure the firewall if your server is behind firewall

Installing BigBlueButton

bbb-install

Install with SSL/TLS

- wget -qO- https://ubuntu.bigbluebutton.org/bbb-install.sh | bash -s -- -v xenial-220 -s bbb.example.com -e info@example.co

Install Greenlight(a simple front-end for BigBlueButton)

- wget -qO- https://ubuntu.bigbluebutton.org/bbb-install.sh | bash -s -- -v xenial-220 -s bbb.example.com -e info@example.com -g

Installing BigBlueButton

Turn Server

- BigBlueButton normally requires a wide range of UDP ports to be available for WebRTC communication. In some network restricted sites or development environments, such as those behind NAT or a firewall that restricts outgoing UDP connections, users may be unable to make outgoing UDP connections to your BigBlueButton server.
- The TURN protocol is designed to allow UDP-based communication flows like WebRTC to bypass NAT or firewalls by having the client connect to the TURN server, and then have the TURN server connect to the destination on their behalf.
- You need a separate server (not the BigBlueButton server) to setup as a TURN server. Specifically you need:
    - An Ubuntu 18.04 server with a public IP address
    - On the TURN server, you need to have the following ports (in additon port 22) availalbe for BigBlueButton to connect (port 3478 and 443) and for the coturn to connect to your BigBlueButton server (49152 - 65535).

Installing BigBlueButton

Turn Server

To configure the TURN server you need:

- A fully qualified domain name (FQDN) with a DNS entry that resolves to the external public IP address of the TURN server
- An e-mail address for Let's Encrypt
- A secret key (it can be an 8 to 16 character random string that you create).

With the above information, you can setup a TURN server for BigBlueButton using bbb-install.sh as follows:

- wget -qO- https://ubuntu.bigbluebutton.org/bbb-install.sh | bash -s -- -c turn.example.com:1234abcd -e info@example.com

Installing BigBlueButton

BBB Front-end

Learning management systems

- BigBlueButton has built-in integrations with all the major learning management systems (LMS), including Canvas, Jenzabar, Moodle, Sakai, and Schoology.

Greenlight

- Greenlight is a simple front-end for BigBlueButton written in Ruby on Rails.
- It lets users create accounts, have permanent rooms, and manage their recordings.
- It also lets you, as the administrator, manage the user accounts (such as approve or deny users).

Scaling BigBlueButton

Scalelite

- Scalelite is an open source load balancer that manages a pool of BigBlueButton servers. It makes the pool of servers appear as a (very scalable) BigBlueButton. A front-end, such as Moodle or Greenlight, sends standard BigBlueButton API requests to the Scalelite server which, in turn, distributes those request to the least loaded BigBlueButton server in the pool.
- A single BigBlueButton server that meets the minimum configuration supports around 200 concurrent users.
- For many schools and organizations, the ability to have 4 simultaneous classes of 50 users, or 8 simultaneous meetings of 25 users, is enough capacity.

Scaling BigBlueButton

Scalelite

- However, what if a school wants to support 1,500 users across 50 simultaneous classes? A single BigBlueButton server cannot handle such a load.
- With Scalelite, a school can create a pool of 4 BigBlueButton servers and handle 16 simultaneous classes of 50 users. Want to scale higher, add more BigBlueButton servers to the pool.

Scaling BigBlueButton

How Scalelite works?

- To load balance the pool, Scalelite periodically polls each BigBlueButton to check if it is reachable online, ready to receive API requests, and to determine its current load (number of connected users). With this information, when Scalelite receives an incoming API call to create a new meeting, it places the new meeting on the least loaded server in the pool. In this way, Scalelite can balance the load of meeting requests evenly across the pool.
- Many BigBlueButton servers will create many recordings. Scalelite can serve a large set of recordings by consolidating them together, indexing them in a database, and, when receiving an incoming getRecordings, use the database index to return quickly the list of available recordings.

Scaling BigBlueButton

Before you begin

The Scalelite installation process requires advanced technical knowledge. You should, at a minimum, be very familar with:

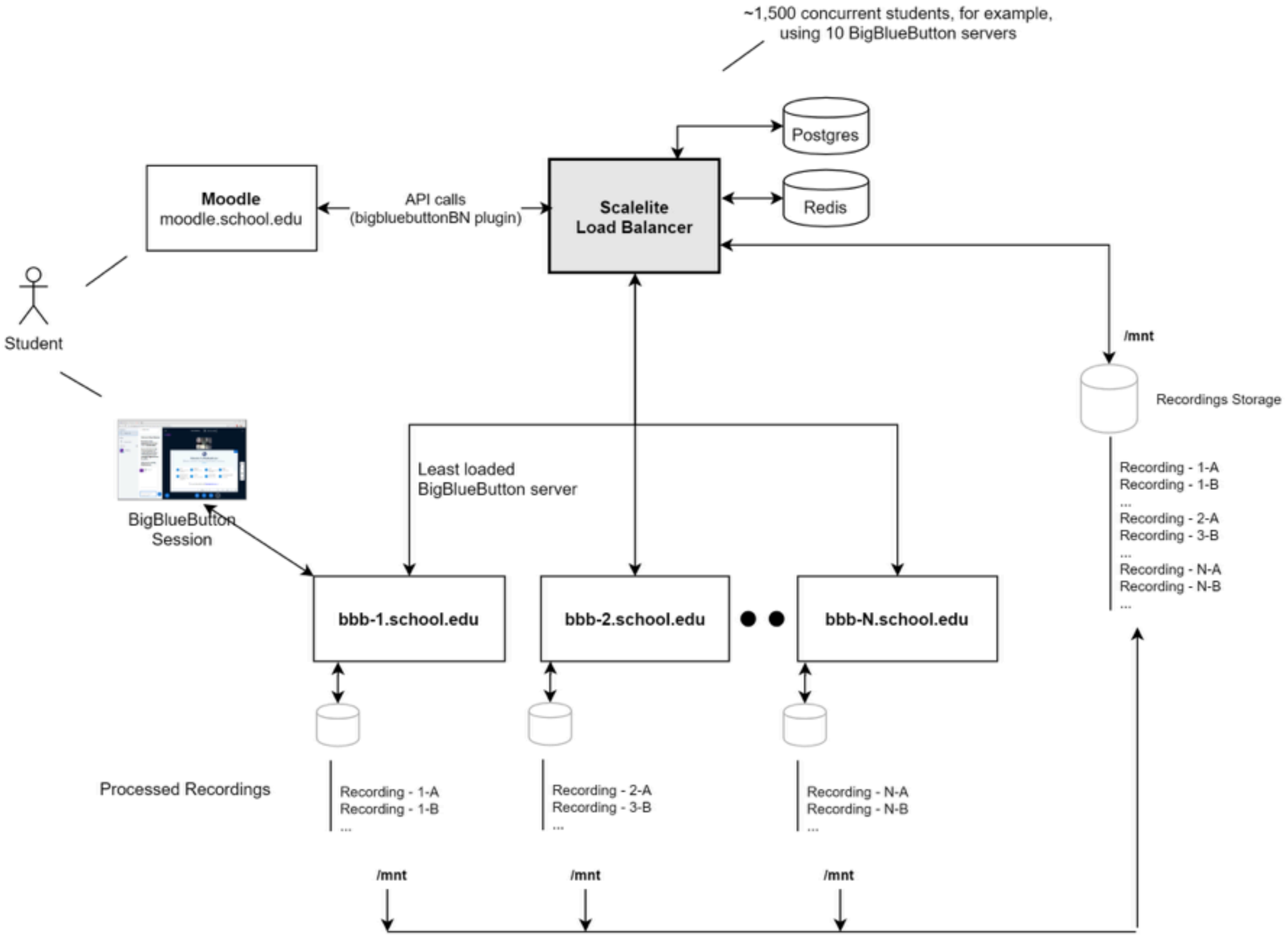
- Setup and administration of a BigBlueButton server
- Setup and administration of a Linux server and using common tools, such as systemd, to manage processes on the server
- How the BigBlueButton API works with a front-end
- How docker containers work
- How UDP and TCP/IP work together
- How to administrate a Linux Firewall
- How to setup a TURN server

Scaling BigBlueButton

Architecture of Scalelite

There are several components required to get Scalelite up and running:

- Multiple BigBlueButton Servers
- Scalelite LoadBalancer Server
- NFS Shared Volume
- PostgreSQL Database
- Redis Cache


~1,500 concurrent students, for example,
using 10 BigBlueButton servers
Postgres
Redis
Moodle
moodle.school.edu
API calls
(bigbluebuttonBN plugin)
Scalelite
Load Balancer
Student
/mnt
Recordings Storage
Recording - 1-A
Recording - 1-B
...
Recording - 2-A
Recording - 3-B
...
Recording - N-A
Recording - N-B
...
Least loaded
BigBlueButton server
BigBlueButton
Session
bbb-1.school.edu
bbb-2.school.edu
bbb-N.school.edu
Processed Recordings
Recording - 1-A
Recording - 1-B
...
Recording - 2-A
Recording - 3-B
...
Recording - N-A
Recording - N-B
...
/mnt
/mnt
/mnt

Scaling BigBlueButton

Minimum Server Requirements

For the Scalelite Server, the minimum recommended server requirements are:

- 4 CPU Cores
- 8 GB Memory

For the external Postgres Database, the minimum recommended server requirements are:

- 2 CPU Cores
- 2 GB Memory
- 20 GB Disk Space (should be good for tens of thousands of recordings)

For the external Redis Cache, the minimum recommended server requirements are:

- 2 CPU Cores
- 0.5GB Memory
- Persistence must be enabled

Scaling BigBlueButton

Setup a pool of BigBlueButton Server

- To setup a pool of BigBlueButton servers (minimum recommended number is 3), it is better to use bbb-install.sh as it can automate the steps to install, configure (with SSL + Let's Encrypt), and update the server when new versions of BigBlueButton are released.
- To help users who are behind restrictive firewalls to send/receive media (audio, video, and screen share) to your BigBlueButton server, you should setup a TURN server and configure each BigBlueButton server to use it.

Scaling BigBlueButton

Setup a shared volume for recordings

- A shared volume should be mounted via NFS on the following systems:
  - BigBlueButton servers
  - Host system for scalelite-nginx Docker container
  - Host system for scalelite-recording-importer Docker container
- NFS (Network File System) allows you to 'share' a directory located on one networked computer with other computers/devices on that network. The computer where directory located is called the server(Scalelite server) and computers or devices connecting to that server are called clients(BBB Servers). Clients usually 'mount' the shared directory to make it a part of their own directory structure.

Scaling BigBlueButton

Setup a shared volume for recordings

- A shared volume should be mounted via NFS on the following systems:
    - BigBlueButton servers
    - Host system for scalelite-nginx Docker container
    - Host system for scalelite-recording-importer Docker container
- The mount point should be different from any of the paths used by stock BigBlueButton. A good choice is /mnt/scalelite-recordings.
- NFS (Network File System) allows you to 'share' a directory located on one networked computer with other computers/devices on that network. The computer where directory located is called the server(Scalelite server) and computers or devices connecting to that server are called clients(BBB Servers). Clients usually 'mount' the shared directory to make it a part of their own directory structure.

Scaling BigBlueButton

Setup a shared volume for recordings

- Install NFS server on Scalelite server:

  apt update

  sudo apt install nfs-kernel-server

- You can configure the directories to be exported by adding them to the /etc/exports file. For example:

  /mnt/scalelite-recordings *(rw,sync,no_root_squash)

  You can replace * with one of the hostname formats. Make the hostname declaration as specific as possible so unwanted systems cannot access the NFS mount.

  To start the NFS server, you can run the following command at a terminal prompt:

  sudo systemctl start nfs-kernel-server.service

- NFS (Network File System) allows you to 'share' a directory located on one networked computer with other computers/ devices on that network. The computer where directory located is called the server(Scalelite server) and computers or devices connecting to that server are called clients(BBB Servers). Clients usually 'mount' the shared directory to make it a part of their own directory structure.

Scaling BigBlueButton

Setup a shared volume for recordings

- NFS Client Configuration on BBB servers:
- Use the mount command to mount a shared NFS directory from Scalelite machine, by typing a command line similar to the following at a terminal prompt:

  mkdir /mnt/scalelite-recordings

  sudo mount bbb.example.com: /mnt/scalelite-recordings /mnt/scalelite-recordings
- An alternate way to mount an NFS share from another machine is to add a line to the /etc/fstab file. The line must state the hostname of the NFS server(Scalelite), the directory on the server being exported, and the directory on the local machine where the NFS share is to be mounted.
- The general syntax for the line in /etc/fstab file is as follows::

  bbb.example.com:/mnt/scalelite-recordings /mnt/scalelite-recordings nfs defaults 0 0
- If you have trouble mounting an NFS share, make sure the nfs-common package is installed on your client. To install nfs-common enter the following command at the terminal prompt:

  sudo apt install nfs-common

Scaling BigBlueButton

Setup up a PostgreSQL Database

- PostgreSQL, or Postgres, is a relational database management system that provides an implementation of the SQL querying language. It is a popular choice for many small and large projects and has the advantage of being standards-compliant and having many advanced features like reliable transactions and concurrency without read locks.
- You can install Postgres on a dedicated server or install it on the same server as Scalelite.
- Ubuntu's default repositories contain Postgres packages, so we can install these easily using the apt packaging system.:

  sudo apt-get update

  sudo apt-get install postgresql postgresql-contrib
- Using PostgreSQL Roles and Databases
- By default, Postgres uses a concept called "roles" to handle in authentication and authorization. These are, in some ways, similar to regular Unix-style accounts, but Postgres does not distinguish between users and groups and instead prefers the more flexible term "role".
- Upon installation Postgres is set up to use ident authentication, which means that it associates Postgres roles with a matching Unix/Linux system account. If a role exists within Postgres, a Unix/Linux username with the same name will be able to sign in as that role. So, you'll need to create a new user called scalelite on your server as well as creating DB user named scalelite.

Scaling BigBlueButton

Setup up a PostgreSQL Database

- If you install Postgres on the same machine as Scalelite then you can define the url of your DB in /etc/default/scalelite like this:

  DATABASE_URL=postgresql://IPofScalelite:5432

- But if you have installed on a dedicated server, here is how you would define it:

  postgresql://username:password@connection_url

- You have to enable remote login in postgresql.conf:

  listen_addresses = '*'

- pg_hba.conf also needs to be edited to allow Scalelite docker IP addresses and users to access postgres like this:

  host scalelite scalelite 172.18.0.2/32 trust

  host scalelite scalelite 172.18.0.3/32 trust

Scaling BigBlueButton

Setup a Redis Cache

- Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker.
- You can install Redis on a dedicated server or install it on the same server as Scalelite.
- Ubuntu's default repositories contain Redis packages, so we can install these easily using the apt packaging system.:

  sudo apt-get update

  sudo apt-get install redis-server

  Next is to enable Redis to start on system boot

  sudo systemctl enable redis-server.service
- You can configure Redis as you need by editing redis.conf file which is well documented
- You have to enable Redis remote login like this:

  bind 0.0.0.0 or

  bind IPofScalelite
- You can easily install Redis using docker allowing access only from Scalelite docker network and enabling persistence on your Redis server:

  First create Redis's env file /etc/default/redis with content of: REDIS_DATA_DIR=/mnt/redis

  Then run the docker container:

  /usr/bin/docker run --name redis --env-file /etc/default/redis --network scalelite --hostname redis -v ${REDIS_DATA_DIR}:/data redis redis-server --appendonly yes
- This will save alot of time you would have spent why Redis is not persisting the servers you configure on Scalelite. Set the redis_url in Scalelite env file /etc/default/scalelite:

  REDIS_URL=redis://redis

Scaling BigBlueButton

Deploying Scalelite Docker Containers

- Common configuration for Docker host system

  URL_HOST=scale.somaliren.org.so

  SECRET_KEY_BASE=

  LOADBALANCER_SECRET=

  DATABASE_URL=postgresql://IPofScalelite:5432

  REDIS_URL=redis://redis

  SCALELITE_TAG=v1

  SCALELITE_RECORDING_DIR=/mnt/scalelite-recordings/var/bigbluebutton

  NGINX_SSL=true

  SCALELITE_NGINX_EXTRA_OPTS=--mount type=bind,source=/etc/letsencrypt,target=/etc/nginx/ssl,readonly

- To run the web front-end containers, SSL is needed:

  First install Nginx on your server and then obtain SSL from let'sencrypt

  Remove Nginx from your server and make sure that no service is using port 80 and 443 before installing docker containers.

- The rest of the document on GitHub from blindsidenetwork is a straightforward to install the required docker containers.

Links

BBB and Scalelite Docs

- BBB script: https://github.com/bigbluebutton/bbb-install
- BBB step-by-step: https://docs.bigbluebutton.org/2.2/install.html
- Scalelite Docs: https://github.com/blindsidenetworks/scalelite
- To get help from the community: https://bigbluebutton.org/support/community/