

• BigBlueButton v. 2.2 inkl. greenlight v. 2

von **Carsten Rieger**|Aktualisiert



o Installationsanleitung für BigBlueButton inkl. Greenlight-Verwaltung/Administration

Die minimalen Servervoraussetzungen für einen BigBlueButton Server sind:

- Ubuntu 16.04.x 64 Bit mit einem 4.x-er Kernel

Support bis 2024

- 8+ GB RAM
- 4 + CPU cores
- TCP 80, 443 und 7443 frei und offen
- UDP 16384 – 32768 frei und offen
- Port 80 darf nicht von anderen Webanwendungen in Nutzung sein
- 300+ GB Festplattenplatz
- 250+ Mbit/s Bandbreite (symmetrisch)
- Dedidizierter Server (physikalisch / bare metal empfohlen!)
- ein hostname und DNS -Record (A, AAAA), um unter anderem SSL nutzen zu können
- IPv4 und IPv6

Serverempfehlungen:

netcup: <https://www.netcup.de/vserver/#root-server-details>

hetzner: <https://www.hetzner.de/dedicated-rootserver>

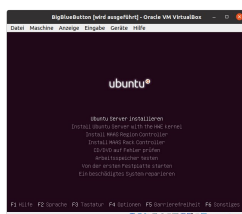
Installationsquellen (Stand 13. November 2020):

[Ubuntu 16.04.7 64Bit LTS ISO-Download](https://ubuntu.com/server/download)

<https://docs.bigbluebutton.org/2.2/install.html>

<https://docs.bigbluebutton.org/greenlight/gl-overview.html>

Installieren Sie Ubuntu 16.04.x LTS Server (das Installationsmedium finden Sie [hier](#)) und konfigurieren die A- und AAAA-Records für den BigBlueButtonServer.



```

$ gcc 10.c -o 10
$ ./10
Ich bin noch da!
Gib das Passwort ein: 123456789
Du bist der Herr der Welt!
  
```

[illegible]

```

$ gcc -std=c99 -Wall test01.c -o test01
$ ./test01
105

```

The screenshot shows a terminal window with a dark background. The title bar reads "Kalkulator - [Datei] [Bearbeiten] [Ansicht] [Einfügen] [Formatieren] [Hilfe]". The terminal content shows the compilation and execution of a C program. The program defines a variable 'char' and prints its value using 'printf'. The output shows the ASCII value 105 for the character 'i'.

[illegible]

The screenshot shows the 'New' window in Oracle VM VirtualBox. The 'Name' field contains 'Rechner', the 'Machine Folder' is 'Rechner', and the 'Operating System' is 'Linux'. The 'Type' is 'Linux' and the 'Version' is 'Ubuntu (64-bit)'. The 'Hard Disk' field is empty. The 'Create Hard Disk Now' button is highlighted in red.

```

// 1. Struktur und Prototypen definieren

// Person
struct Person {
    char* name;
    int age;
};

// PersonList
struct PersonList {
    Person* persons;
    int count;
};

// 2. Funktionen definieren

// PersonList neu anlegen
PersonList* newPersonList() {
    PersonList* list = (PersonList*) malloc(sizeof(PersonList));
    list->persons = (Person*) malloc(sizeof(Person) * 10);
    list->count = 0;
    return list;
}

// Person zu PersonList hinzufügen
void addPerson(PersonList* list, char* name, int age) {
    list->persons[list->count] = (Person) {name, age};
    list->count++;
}

// 3. Hauptfunktion
int main() {
    PersonList* list = newPersonList();
    addPerson(list, "Gerd", 30);
    addPerson(list, "Anna", 25);
    addPerson(list, "Hans", 40);
    // ...
}

```

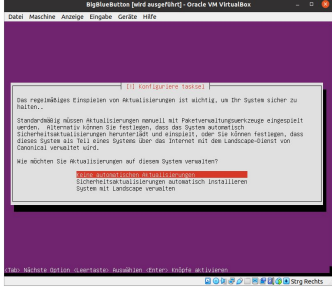
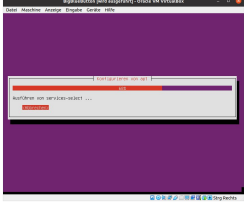
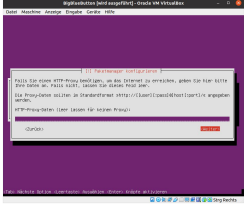
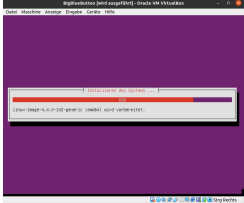
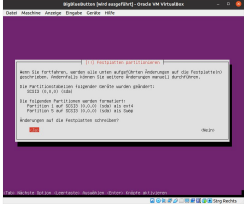
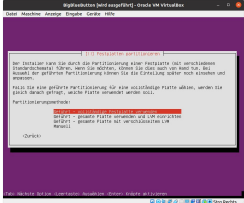
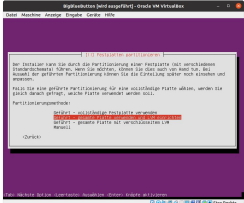
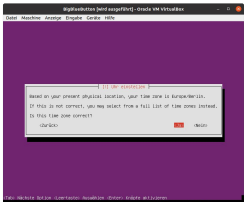
Output:

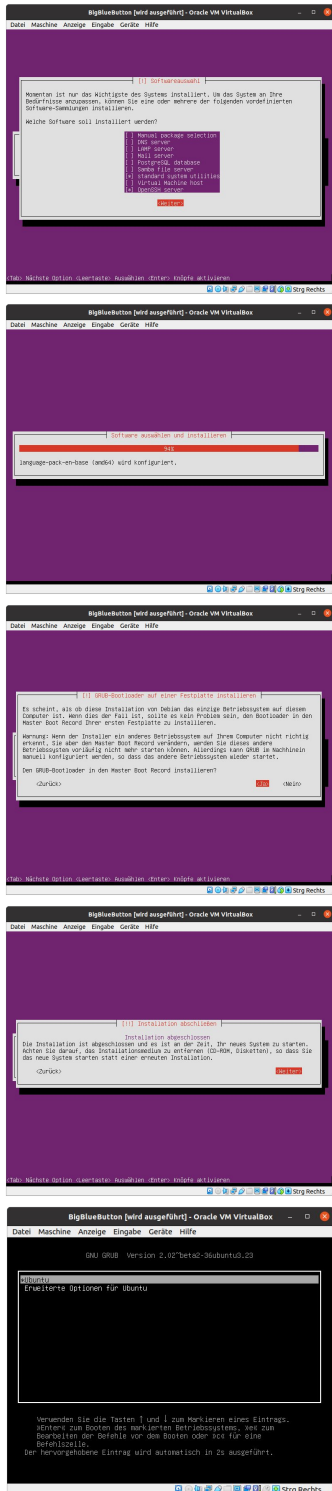
```

// 3. Hauptfunktion
int main() {
    PersonList* list = newPersonList();
    addPerson(list, "Gerd", 30);
    addPerson(list, "Anna", 25);
    addPerson(list, "Hans", 40);
    // ...
}

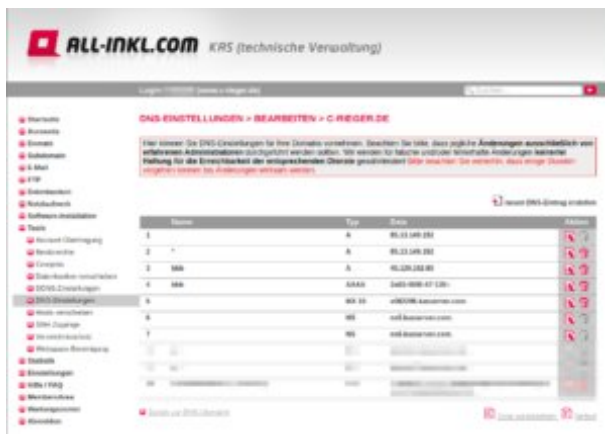
```

[illegible][illegible]





Für den nachfolgenden Screenshot und den weiteren Verlauf dieser Anleitung verwende ich als Synonym die Werte „bbb.c-rieger.de“ und „ihre.domäne.de“. Bitte ersetzen Sie diese Domäne an allen Stellen mit ihrer Domäne.



Nach der Ubuntu 16.04.x-Serverinstallation beginnen wir mit der Überprüfung aller Voraussetzungen für eine BigBlueButton Serverinstallation. Überprüfen wir also zuerst die Spracheinstellungen am Server, erwartet wird „en_US.UTF-8“:

```
sudo -s
```

```
cat /etc/default/locale
```

Das Ergebnis sollte wie folgt aussehen:

```
LANG="en_US.UTF-8"
```

Andernfalls korrigieren Sie diese Einstellungen mittels der nachfolgenden Befehle

```
apt install language-pack-en
```

```
update-locale LANG=en_US.UTF-8
```

und überprüfen dann, ob noch weitere Einträge in der Datei „/etc/default/locale“ vorhanden sind. Wenn ja, dann entfernen Sie diese bitte, melden sich vom System ab, dann wieder am System an und überprüfen die Ausgabe mittels des nachfolgenden Befehls erneut.

```
systemctl show-environment
```

```
root@bbb: # systemctl show-environment
LANG=en_US.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
root@bbb: #
```

Nun installieren wir, sofern noch nicht vorhanden, die Firewall (ufw)

```
apt install -y ufw
```

und richten diese wie folgt ein:

```
ufw allow 22/tcp
```

```
ufw allow 80/tcp
```

```
ufw allow 443/tcp
```

```
ufw allow 7443/tcp
```

```
ufw allow 16384:32768/udp
```

```
ufw logging medium
```

```
ufw default deny incoming
```

```
ufw enable
```

```
ufw status verbose
```

Nun setzen wir den hostnamen

```
nano /etc/hosts
```

nach der folgenden Syntax:

EXTERNAL_IP_ADDRESS EXTERNAL_HOST_NAME

127.0.1.1 bbb.ihredomain.de

12.34.56.78 bbb.ihredomain.de



Mit Hinblick auf die Verwendung von Let's Encrypt und der Verwendung von https konfigurieren wir die Softwarequellen:

```
apt update
```

```
apt install -y software-properties-common
```

```
add-apt-repository universe
```

```
add-apt-repository ppa:certbot/certbot
```

```
apt install -y certbot
```

Dann erstellen wir einen 4096-Bit diffie-hellman Schlüssel um die Sicherheit zu erhöhen.

```
mkdir -p /etc/nginx/ssl
```

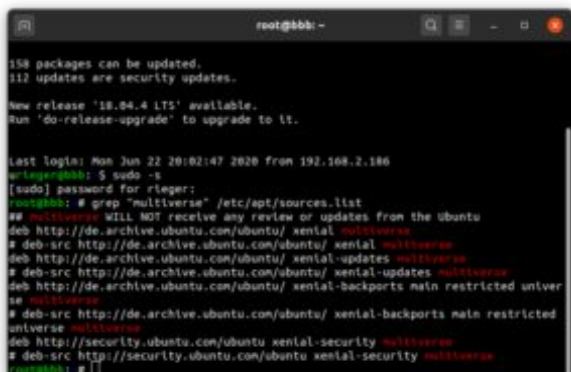
```
openssl dhparam -out /etc/nginx/ssl/dhp-4096.pem 4096
```

Die SSL-Zertifikate selbst erstellen wir erst später im Anschluß an die BigBlueButton-Installation.

Bevor die Installation beginnt muss noch geprüft werden, ob xenial multiverse in der /etc/apt/sources.list verfügbar ist:

```
grep "multiverse" /etc/apt/sources.list | grep -v "^#"

```



Sollte der grep kein Ergebnis anzeigen, so ergänzen Sie die fehlende Softwarequelle durch:

```
echo "deb http://archive.ubuntu.com/ubuntu/ xenial multiverse" | sudo tee -a /etc/apt/sources.list
```

Nun fügen wir die BBB Softwarequellen hinzu und starten die Installation:

```
add-apt-repository ppa:bigbluebutton/support -y
```

```
add-apt-repository ppa:rmescandon/yq -y
```

```
add-apt-repository ppa:libreoffice/ppa
```

```
apt update && apt dist-upgrade
```

Der HTML5-Client benötigt die MongoDB und nodejs:

```
wget -qO - https://www.mongodb.org/static/pgp/server-3.4.asc | sudo apt-key add -
```

```
echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4 multiverse" |  
sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list
```

```
apt update && apt install -y mongodb-org curl
```

```
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
```

```
apt install -y nodejs
```

Fügen Sie den BBB-Schlüssel und den BBB-Download zum Server hinzu:

```
wget https://ubuntu.bigbluebutton.org/repo/bigbluebutton.asc -O- | sudo apt-key add -
```

```
echo "deb https://ubuntu.bigbluebutton.org/xenial-22/ bigbluebutton-xenial main" | sudo tee  
/etc/apt/sources.list.d/bigbluebutton.list
```

```
apt update
```

Es ist soweit, die Installation beginnt:

```
apt install -y bigbluebutton bbb-html5
```

```
apt-get install -f && apt dist-upgrade
```

Die Installation ist nun bereits abgeschlossen und die Konfiguration beginnt. Wir ersetzen den Flash- mit dem HTML5-Client. Dazu sichern wir die Datei

```
cp /usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties /usr/share/bbb-web/WEB-  
INF/classes/bigbluebutton.properties.bak
```

und bearbeiten

```
nano /usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties
```

```
[...]
```

```
attendeesJoinViaHTML5Client=true
```

```
...
```

```
moderatorsJoinViaHTML5Client=true
```

```
[...]
```

```
autoStartRecording=false
```

```
[...]
```

Weitere, globale Einstellungen für den HTML5-Client werden in der Datei

```
/usr/share/meteor/bundle/programs/server/assets/app/config/settings.yml
```

vorgenommen. Weitere Informationen finden Sie [hier](#).

Bevor wir nun die SSL-Zertifikate erstellen und einbinden können müssen wir BBB konfigurieren.

```
bbb-conf --setip ihre.domäne.de
```

Ersetzen Sie **ihre.domäne.de** und requestieren dann die SSL-Zertifikate:

```
certbot --webroot -w /var/www/bigbluebutton-default/ -d ihre.domäne.de certonly
```

Ersetzen Sie erneut „**ihre.domäne.de**“ mit ihrer richtigen Domäne, bspw. **bbb.c-rieger.de**

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at /etc/letsencrypt/live/**ihre.domäne.de**/fullchain.pem. Your cert will expire on 2020-YY-ZZ. To obtain a new version of the certificate in the future, simply run Let's Encrypt again.
 - If you like Let's Encrypt, please consider supporting our work by: Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
Donating to EFF: <https://eff.org/donate-le>
- Sie finden die neuen Zertifikatsdateien (cert.pem, chain.pem, fullchain.pem, privkey.pem) unter

```
ls /etc/letsencrypt/live/ihre.domäne.de
```

Passen Sie nun den Webserver nginx an:

```
cp /etc/nginx/sites-available/bigbluebutton /etc/nginx/sites-available/bigbluebutton.bak
```

```
nano /etc/nginx/sites-available/bigbluebutton
```

Kopieren und ersetzen Sie die folgenden Zeilen und **ihre.domäne.de** mit ihrer Domäne:

```
server {  
listen 80;  
listen [::]:80;  
server_name ihre.domäne.de;  
listen 443 ssl;  
listen [::]:443 ssl;  
ssl_certificate /etc/letsencrypt/live/ihre.domäne.de/fullchain.pem;  
ssl_certificate_key /etc/letsencrypt/live/ihre.domäne.de/privkey.pem;  
ssl_session_cache shared:SSL:10m;  
ssl_session_timeout 10m;  
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
ssl_ciphers  
"ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH  
+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:!aNULL:!MD5:!DSS:!AES256";  
ssl_prefer_server_ciphers on;  
ssl_dhparam /etc/nginx/ssl/dhp-4096.pem;  
access_log /var/log/nginx/bigbluebutton.access.log;  
[...]
```

```

root@bbb:~# cat /etc/nginx/nginx.conf
#user www;
worker_processes 1;

error_log /var/log/nginx/error.log;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
    #                '$status $body_bytes_sent "$http_referer" '
    #                '"$http_user_agent" "$http_x_forwarded_for"';
    #access_log /var/log/nginx/access.log main;

    send_timeout 6h;
    client_body_timeout 6h;
    proxy_read_timeout 6h;
    proxy_send_timeout 6h;

    keepalive_timeout 65;
}

```

Testen sie die neue Webserverkonfiguration

nginx -t

und starten den Webserver neu, sofern die Syntaxprüfung okay war.

systemctl restart nginx

Denken Sie unbedingt daran, die Domäne gemäß Ihrer Domäne umzubenennen. Um die Zertifikate automatisch erneuern zu lassen richten wir noch zwei Cronjobs ein:

crontab -e

30 2 * * 1 /usr/bin/certbot renew >> /var/log/le-renew.log

35 2 * * 1 /bin/systemctl reload nginx

Auch die FreeSWITCH Konfiguration muss für SSL noch angepasst werden:

cp /etc/bigbluebutton/nginx/sip.nginx /etc/bigbluebutton/nginx/sip.nginx.bak

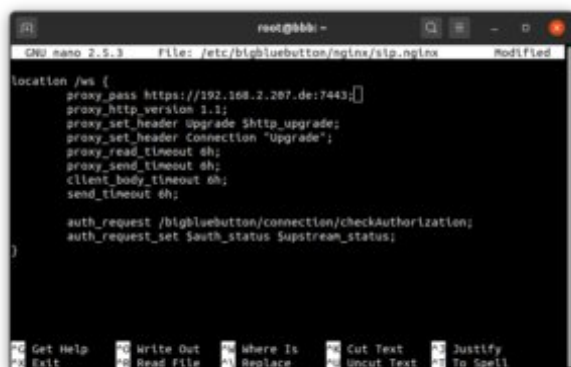
nano /etc/bigbluebutton/nginx/sip.nginx

```

location /ws {
    proxy_pass https://<ihre-ip.de>:7443;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_read_timeout 6h;
    proxy_send_timeout 6h;
    client_body_timeout 6h;
    send_timeout 6h;

    auth_request /bigbluebutton/connection/checkAuthorization;
    auth_request_set $auth_status $upstream_status;
}

```



```

root@bbb:~# nano /etc/bigbluebutton/nginx/sip.nginx
GNU nano 2.5.3 File: /etc/bigbluebutton/nginx/sip.nginx
location /ws {
    proxy_pass https://192.168.2.267.de:7443;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_read_timeout 6h;
    proxy_send_timeout 6h;
    client_body_timeout 6h;
    send_timeout 6h;

    auth_request /bigbluebutton/connection/checkAuthorization;
    auth_request_set $auth_status $upstream_status;
}

```

Ebenso muss BBB noch für die Verwendung von https-Sessions angepasst werden:

- 1. bigbluebutton.properties

```
cp /usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties /usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties.bak
```

```
nano /usr/share/bbb-web/WEB-INF/classes/bigbluebutton.properties
```

```
bigbluebutton.web.serverURL=https://ihre.domäne.de
```

- 2. screenshare.properties

```
cp /usr/share/red5/webapps/screenshare/WEB-INF/screenshare.properties /usr/share/red5/webapps/screenshare/WEB-INF/screenshare.properties.bak
```

```
nano /usr/share/red5/webapps/screenshare/WEB-INF/screenshare.properties
```

```
streamBaseUrl=rtmp://ihre.domäne.de/screenshare
```

```
jnlpUrl=https://ihre.domäne.de/screenshare
```

```
jnlpFile=https://ihre.domäne.de/screenshare/screenshare.jnlp
```

- 3. config.xml

```
sed -e 's|http://|https://|g' -i /var/www/bigbluebutton/client/conf/config.xml
```

- 4. kurento ws

```
cp /usr/share/meteor/bundle/programs/server/assets/app/config/settings.yml /usr/share/meteor/bundle/programs/server/assets/app/config/settings.yml.bak
```

```
nano /usr/share/meteor/bundle/programs/server/assets/app/config/settings.yml
```

kurento:

```
wsUrl: wss://ihre.domäne.de/bbb-webrtc-sfu
```

und

note:

```
enabled: true
```

```
url: https://ihre.domäne.de/pad
```

- 5. bigbluebutton.yml

```
cp /usr/local/bigbluebutton/core/scripts/bigbluebutton.yml /usr/local/bigbluebutton/core/scripts/bigbluebutton.yml.bak
```

```
nano /usr/local/bigbluebutton/core/scripts/bigbluebutton.yml
```

```
playback_protocol: https
```

Schließen Sie die Konfiguration ab indem Sie openjdk-**8** installieren und anschließend Version **8** auswählen:

```
apt install -y openjdk-8-jdk
```

```
update-alternatives --config java
```

Editieren Sie zuletzt noch die Datei

```
nano /opt/freeswitch/etc/freeswitch/vars.xml
```

und ersetzen ihre Domäne an zwei Stellen:

```
<X-PRE-PROCESS cmd="set" data="external_rtp_ip=stun:ihre.domain.de:5349"/>
```

und

```
<X-PRE-PROCESS cmd="set" data="external_sip_ip=stun:ihre.domain.de:5349"/>
```

Starten Sie BBB neu:

```
bbb-conf --restart
```

Um BigBlueButton administrieren zu können und eine Benutzerverwaltung nutzen zu können bedienen wir uns greenlight. Dazu gehen wir wie folgt vor:

- 6. Installation von docker und docker-compose

```
apt update
```

```
apt install -y apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
apt update && apt install -y docker-ce docker-ce-cli containerd.io
```

Testen Sie nun, dass docker lauffähig ist:

```
docker -v
```

Legen Sie den **~/greenlight** Ordner an und wechseln Sie in diesen hinein:

```
mkdir ~/greenlight && cd ~/greenlight
```

Um greenlight zu starten muss der Docker geladen und die Konfiguration erstellt werden:

```
docker run --rm bigbluebutton/greenlight:v2 cat ./sample.env > .env
```

Um greenlight einrichten und konfigurieren zu können erstellen wir das notwendige Secret (**SECRET_KEY_BASE**):

```
docker run --rm bigbluebutton/greenlight:v2 bundle exec rake secret
```

Lassen Sie sich dann die weiteren Secret-Informationen anzeigen

(**BIGBLUEBUTTON_ENDPOINT** und **BIGBLUEBUTTON_SECRET**) und tragen alle drei in die Datei **~/greenlight/.env** ein.

```
bbb-conf --secret
```

```
cd ~/greenlight
```

```
nano .env
```

Setzen Sie die vier Werte für:

SECRET_KEY_BASE, **BIGBLUEBUTTON_ENDPOINT**, **BIGBLUEBUTTON_SECRET** und **SAFE_HOSTS** (ohne https, also bspw: **ihre.domain.de**).



```
1. Create a secret key for the Rails
2. You can generate a secret key through the greenlight docker image
3. With the command:
4. docker run --rm bigbluebutton/greenlight:v2 cat ./sample.env > .env
5. docker run --rm bigbluebutton/greenlight:v2 bundle exec rake secret
6. BIGBLUEBUTTON_ENDPOINT=https://your-domain.de
7. BIGBLUEBUTTON_SECRET=your-secret-key
8. The endpoint and secret for your BigBlueButton server.
9. Set them in your .env file using the following command:
10. The line below shows the command to run in your BigBlueButton server:
11. bbb-conf --secret
12. Configuration summary:
13. BIGBLUEBUTTON_ENDPOINT=https://your-domain.de
14. BIGBLUEBUTTON_SECRET=your-secret-key
15. The hostname that the application is accessible from.
```

Überprüfen Sie dann die Konfiguration mittels:

```
cd ~/greenlight
```

```
docker run --rm --env-file .env bigbluebutton/greenlight:v2 bundle exec rake conf:check
```

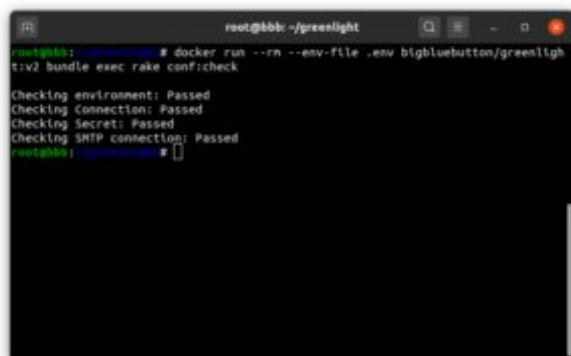
Ohne SMTP müssen alle drei Tests erfolgreich (**passed**) dargestellt werden. Haben Sie SMTP konfiguriert

```
nano ~/greenlight/.env
```

```
[...]  
ALLOW_MAIL_NOTIFICATIONS=true  
[...]  
SMTP_SERVER=smtp.domäne.de  
SMTP_PORT=587  
SMTP_DOMAIN=ihredomäne.de  
SMTP_USERNAME=smtpbenutzer@ihredomäne.de  
SMTP_PASSWORD=IhrPasswort  
SMTP_AUTH=plain  
SMTP_STARTTLS_AUTO=true  
SMTP_SENDER=email@ihredomäne.de  
[...]
```

```
ALLOW_MAIL_NOTIFICATIONS=true  
#  
# The notifications are sent using sendmail, unless the SMTP_SERVER variable is set.  
# In that case, make sure the rest of the variables are properly set.  
#  
# If your mail server has a self-signed certificate, you'll also need to include the line below.  
# Please note that enable this presents its own security risks and should not be done unless necessary.  
# SMTP_OPENSSL_VERIFY_MODE=none  
#  
SMTP_SERVER=smtp.localhost localhost.com  
SMTP_PORT=587  
SMTP_DOMAIN=localhost localhost.com  
SMTP_USERNAME=smtp localhost.com  
SMTP_PASSWORD=smtp localhost.com  
SMTP_AUTH=plain  
SMTP_STARTTLS_AUTO=true  
# Specify the email address that all mail is sent from  
SMTP_SENDER=smtp@localhost localhost.com
```

so müssen alle vier Tests als erfolgreich (**passed**) dargestellt werden.



```
root@bbb: ~/greenlight  
root@bbb:~# docker run --rm --env-file .env bigbluebutton/greenlight:v2 bundle exec rake conf:check  
Checking environment: Passed  
Checking Connection: Passed  
Checking Secret: Passed  
Checking SMTP connection: Passed  
root@bbb:~#
```

Greenlight wird nun nach **<ihredomäne.de>** installiert, wodurch später keine Konflikte mit BBB auftreten können.

```
docker run --rm bigbluebutton/greenlight:v2 cat ./greenlight.nginx | sudo tee  
/etc/bigbluebutton/nginx/greenlight.nginx
```

Prüfen Sie, ob die nachfolgende Datei Inhalt aufweist:

```
cat /etc/bigbluebutton/nginx/greenlight.nginx
```

```
# Routes requests to Greenlight based on the '/b' prefix.
# Use this file to route '/b' paths on your BigBlueButton server
# to the Greenlight application. If you are using a different
# subpath, you should change it here.

location /b {
    proxy_pass          http://127.0.0.1:5000;
    proxy_set_header    Host          $host;
    proxy_set_header    X-Forwarded-For  $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_http_version  1.1;
}

location /b/cable {
    proxy_pass          http://127.0.0.1:5000;
    proxy_set_header    Host          $host;
    proxy_set_header    X-Forwarded-For  $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_set_header    Upgrade        $http_upgrade;
    proxy_set_header    Connection     'Upgrade';
    proxy_http_version  1.1;
    proxy_read_timeout  6h;
    proxy_send_timeout  6h;
    client_body_timeout 6h;
    send_timeout        6h;
}

# Allow larger body size for uploading presentations
location ~ /preupload/presentation$ {
    client_max_body_size 30m;

    proxy_pass          http://127.0.0.1:5000;
    proxy_set_header    Host          $host;
    proxy_set_header    X-Forwarded-For  $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_http_version  1.1;
}

# Only needed if using presentations and deployed at a relative root (ex "/b")
# If deploying at "/", delete the section below

location /rails/active_storage {
    return 301 /b$request_uri;
}
```

Die Administrationsoberfläche wäre technisch schon unter <https://<ihredomäne.de>/b> erreichbar, doch wir wollen BBB zukünftig direkt in greenlight unter <https://<ihredomäne.de>> starten. Aus diesem Grund fügen wir am Ende der vHost-Datei (/etc/nginx/sites-available/bigbluebutton) folgende Zeilen vor der letzten Klammer `},` ein:

```
cp /etc/nginx/sites-available/bigbluebutton /etc/nginx/sites-available/bigbluebutton.bak
nano /etc/nginx/sites-available/bigbluebutton

location = / {
    return 307 /b;
}
```

Überprüfen Sie die Webserverkonfiguration mittels

```
nginx -t
```

und starten, einen fehlerfreien Test mit `nginx -t` vorausgesetzt, letztmalig den Webserver `nginx` neu:

```
systemctl restart nginx
```

Um Greenlight zu starten wird `docker-compose` benötigt. Prüfen Sie, ob es bereits vorhanden ist:

```
cd ~/greenlight
```

```
docker-compose -v
```

Ist es nicht vorhanden, so installieren Sie es wie folgt nach:

```
curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

Kopieren Sie die **docker-compose.yml** Datei aus dem Greenlight image nach **~/greenlight:**

```
docker run --rm bigbluebutton/greenlight:v2 cat ./docker-compose.yml > docker-compose.yml
```

Erzeugen Sie ein zufälliges Passwort für PostgreSQL und ersetzen es in den `.env`- und `docker-compose.yml`-Dateien:

```
export pass=$(openssl rand -hex 8); sed -i
's/POSTGRES_PASSWORD=password/POSTGRES_PASSWORD=$pass/g' docker-compose.yml;sed -i
's/DB_PASSWORD=password/DB_PASSWORD=$pass/g' .env
```

Nun starten Sie die Administrationsanwendung greenlight

```
cd ~/greenlight
```

```
docker-compose up -d
```

und rufen dann BigBlueButton und greenlight erstmalig auf:

<https://<ihredomäne.de>>

Für mögliche Analysen finden Sie die Logdateien unter:

```
~/greenlight/log
```

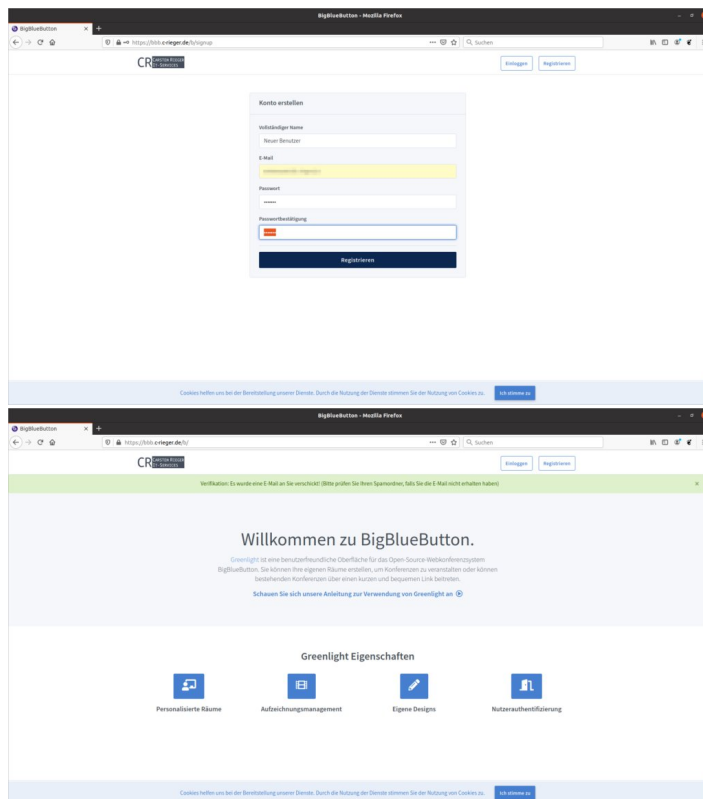
Richten Sie sich noch einen BBB-Administratoraccount ein

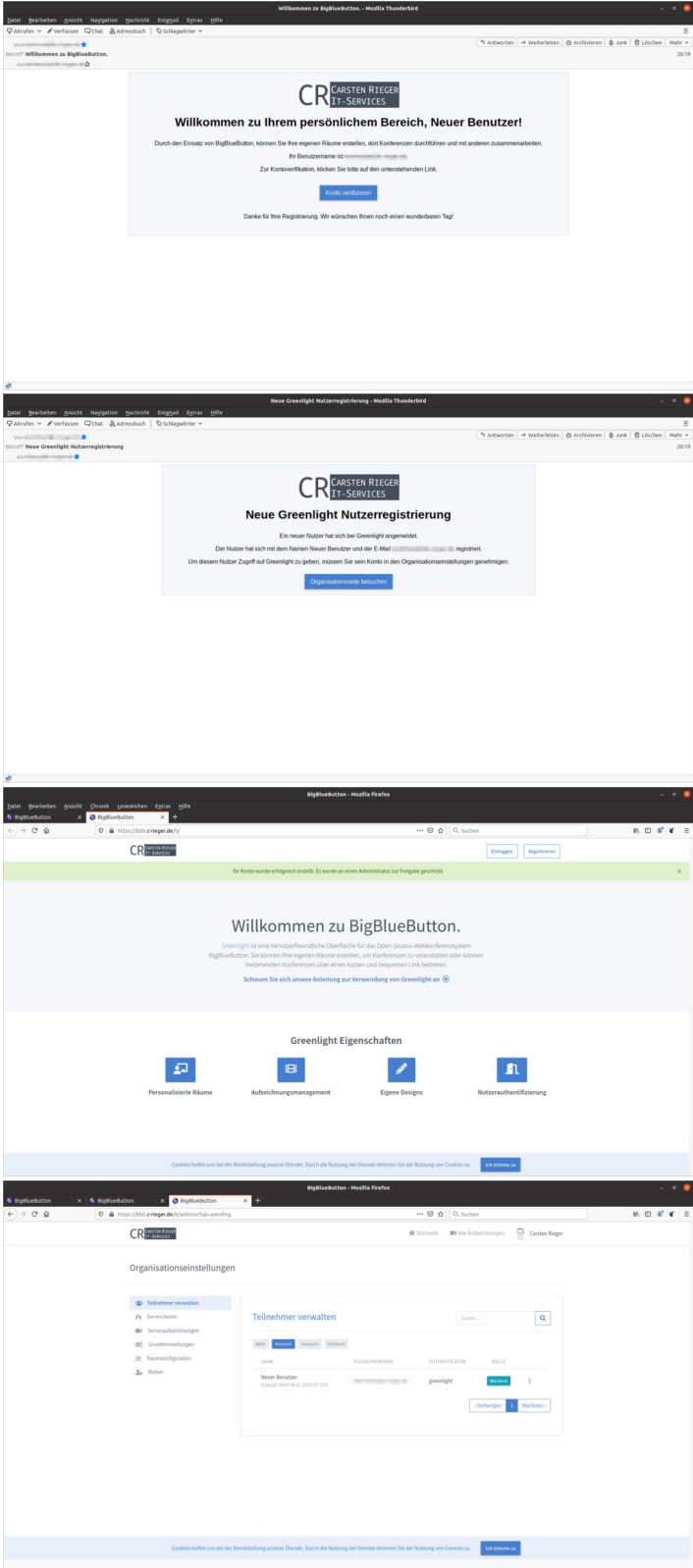
```
docker exec greenlight-v2 bundle exec rake user:create["name","email","password","admin"]
```

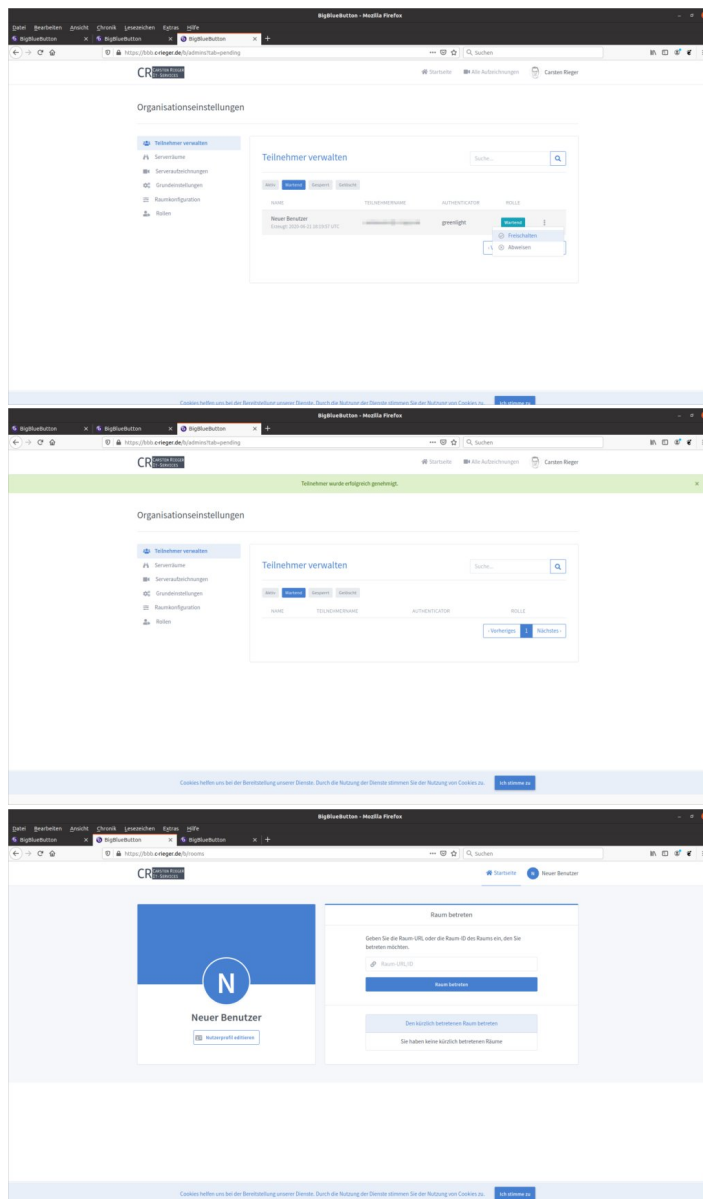
um sich am Webfrontend anmelden zu können. Ersetzen Sie dabei die Werte „**name**“, „**email**“ und „**password**“ mit den realen Werten Ihres Administrato-Accounts. Zudem können Sie auch einen „normalen“ Benutzer, ohne Administratorenberechtigungen, erzeugen:

```
docker exec greenlight-v2 bundle exec rake user:create["name","email","password","user"]
```

Auch hier gilt es die Werte „**name**“, „**email**“ und „**password**“ mit den „echten“ Werten des BBB-Benutzers zu ersetzen.







Um Problemen mit Verbindungen über verschiedene Netzwerke (bspw. verursacht durch NAT o.Ä.) zu verhindern, sollten Sie Ihren eigenen **Coturn-Server** nutzen. Dazu erstellen wir eine eigene Version der Datei

```
mv /usr/share/bbb-web/WEB-INF/classes/spring/turn-stun-servers.xml /usr/share/bbb-web/WEB-INF/classes/spring/turn-stun-servers.xml.bak
```

```
nano /usr/share/bbb-web/WEB-INF/classes/spring/turn-stun-servers.xml
```

und passen diese Zeilen in der Vorlage an Ihr System an:

```
...
<constructor-arg index="0" value="stun:coturn.ihre.domain:5349"/>
...
<constructor-arg index="0" value="abc...gh...987"/>
<constructor-arg index="1" value="turns:coturn.ihre.domain:5349?transport=tcp"/>
...
<constructor-arg index="0" value="abc...gh...987"/>
<constructor-arg index="1" value="turn:coturn.ihre.domain:5349?transport=tcp"/>
```

...

Kopieren Sie den gesamten, angepassten Inhalt der [Vorlage](#) in die noch leere Datei und starten dann BigBlueButton neu:

bbb-conf --restart

FERTIG – Somit nutzen Sie ihren eigenen coturn-Server und verlassen Ihre Serverumgebung nicht. Die asugezeichnete Integration in Ihre Nextcloud beschreibt [der folgende Artikel](#) auf dieser Webseite.