

# Guide d'Installation

## BigBlueButton

Infrastructure RBER - Visioconférence

| Paramètre             | Valeur             |
|-----------------------|--------------------|
| Domaine               | visio.rber.bj      |
| IP Publique (HAProxy) | 102.222.216.6      |
| IP Interne VM BBB     | 10.29.113.101      |
| Version BBB           | 3.0 (Ubuntu 24.04) |
| Date                  | Décembre 2025      |

RBER - Réseau Béninois pour l'Éducation et la Recherche

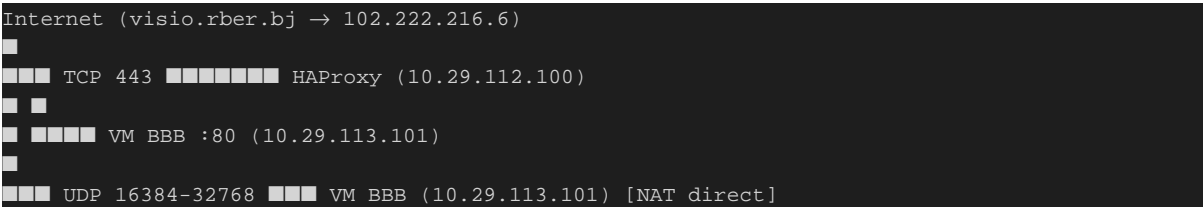
# Table des matières

1. Architecture et prérequis
2. Préparation de la VM
3. Installation de BigBlueButton
4. Configuration post-installation
5. Configuration HAProxy
6. Configuration du firewall
7. Configuration TURN/STUN
8. Tests et validation
9. Intégration Moodle
10. Maintenance et dépannage

# 1. Architecture et prérequis

## 1.1 Schéma d'architecture

L'architecture utilise HAProxy comme point d'entrée unique pour le trafic HTTPS. Le trafic UDP WebRTC (audio/vidéo) est routé directement vers la VM BBB via NAT.



## 1.2 Prérequis VM BigBlueButton

| Ressource | Minimum          | Recommandé       |
|-----------|------------------|------------------|
| CPU       | 8 vCPU           | 16 vCPU          |
| RAM       | 16 Go            | 32 Go            |
| Disque    | 50 Go SSD        | 100 Go SSD       |
| OS        | Ubuntu 22.04 LTS | Ubuntu 24.04 LTS |
| Réseau    | 100 Mbps         | 1 Gbps           |

## 1.3 Ports réseau requis

| Port        | Protocole | Usage              | Accès              |
|-------------|-----------|--------------------|--------------------|
| 80          | TCP       | HTTP (interne)     | HAProxy uniquement |
| 443         | TCP       | HTTPS API + Client | Via HAProxy        |
| 5349        | TCP       | TURN TLS           | Via HAProxy        |
| 16384-32768 | UDP       | WebRTC média       | Direct Internet    |

## 2. Préparation de la VM

### 2.1 Mise à jour du système

```
# Connexion SSH à la VM BBB
ssh admin@10.29.113.101

# Mise à jour complète
sudo apt update && sudo apt upgrade -y

# Installer les outils de base
sudo apt install -y curl wget gnupg2 software-properties-common
```

### 2.2 Configuration du hostname

Le hostname doit correspondre au FQDN utilisé pour accéder au service.

```
# Définir le hostname
sudo hostnamectl set-hostname bbb.internal.rber.bj

# Ajouter l'entrée dans /etc/hosts
echo "10.29.113.101 bbb.internal.rber.bj" | sudo tee -a /etc/hosts

# Vérifier
hostname -f
```

### 2.3 Désactivation IPv6

IPv6 peut causer des problèmes avec WebRTC. Il est recommandé de le désactiver.

```
# Désactiver IPv6 temporairement
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1

# Désactiver IPv6 de façon permanente
cat << 'EOF' | sudo tee -a /etc/sysctl.conf
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
EOF

# Appliquer
sudo sysctl -p
```

### 2.4 Configuration des locales

```
# Générer les locales
sudo locale-gen en_US.UTF-8
sudo update-locale LANG=en_US.UTF-8

# Vérifier
locale
```

## 3. Installation de BigBlueButton

### 3.1 Téléchargement du script d'installation

BigBlueButton fournit un script d'installation automatisé qui gère toutes les dépendances.

```
# Télécharger le script
wget -qO-
https://raw.githubusercontent.com/bigbluebutton/bbb-install/v3.0.x-release/bbb-install.sh >
bbb-install.sh

# Rendre exécutable
chmod +x bbb-install.sh
```

### 3.2 Exécution de l'installation

L'installation se fait SANS certificat SSL car HAProxy gère le HTTPS.

```
# Installation BBB 3.0 avec Greenlight (sans SSL)
sudo ./bbb-install.sh -v noble-300 -s visio.rber.bj -g

# Paramètres:
# -v noble-300 : Version BBB 3.0 sur Ubuntu 24.04 (noble)
# -s visio.rber.bj : Hostname du service
# -g : Installer Greenlight (interface web)

# L'installation prend 15-30 minutes selon la connexion
```

### 3.3 Vérification de l'installation

```
# Vérifier l'état de BBB
sudo bbb-conf --check

# Afficher la configuration
sudo bbb-conf --status

# Récupérer le secret API (IMPORTANT - à noter)
sudo bbb-conf --secret

# Exemple de sortie:
# URL: http://visio.rber.bj/bigbluebutton/
# Secret: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Noter le Secret affiché. Il sera nécessaire pour l'intégration avec Moodle.

## 4. Configuration post-installation

### 4.1 Configuration de bbb-web

Éditer le fichier de configuration principal de BBB.

```
# Éditer la configuration
sudo nano /etc/bigbluebutton/bbb-web.properties

# Modifier/ajouter les lignes suivantes:
bigbluebutton.web.serverURL=https://visio.rber.bj
defaultMaxUsers=150
defaultMeetingDuration=360
disableRecordingDefault=false
muteOnStart=false
```

### 4.2 Configuration pour fonctionner derrière un proxy

```
# Éditer la configuration Nginx de BBB
sudo nano /etc/bigbluebutton/nginx/bbb-html5.nginx

# S'assurer que ces headers sont présents dans le bloc location:
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
```

### 4.3 Configuration du client HTML5

```
# Créer/éditer le fichier de surcharge
sudo nano /etc/bigbluebutton/bbb-html5.yml

# Contenu:
public:
  app:
    basename: '/html5client'
    bbbServerVersion: '3.0'
    kurento:
      wsUrl: wss://visio.rber.bj/bbb-webrtc-sfu
```

### 4.4 Redémarrage des services

```
# Redémarrer BBB avec la nouvelle configuration
sudo bbb-conf --restart

# Vérifier que tous les services sont actifs
sudo bbb-conf --check

# Vérifier les services individuellement
sudo systemctl status bbb-html5-frontend@1
sudo systemctl status bbb-html5-frontend@2
sudo systemctl status bbb-webrtc-sfu
sudo systemctl status kurento-media-server
```

## 5. Configuration HAProxy

### 5.1 Configuration du frontend HTTPS

Ajouter cette configuration dans `/etc/haproxy/haproxy.cfg` sur le serveur HAProxy (10.29.112.100).

```
#-----  
# Frontend HTTPS - Ajouter l'ACL pour BBB  
#-----  
frontend ft_https  
bind *:443 ssl crt /etc/haproxy/certs/rber.bj.pem alpn h2,http/1.1  
mode http  
  
# ACL existantes...  
  
# ACL pour BigBlueButton  
acl is_visio hdr(host) -i visio.rber.bj  
  
# Routing vers BBB  
use_backend bk_bbb if is_visio
```

### 5.2 Configuration du backend BBB

```
#-----  
# Backend BigBlueButton  
#-----  
backend bk_bbb  
mode http  
balance roundrobin  
  
# Options HTTP  
option forwardfor  
option httpchk GET /bigbluebutton/api  
http-check expect status 200  
  
# Headers pour proxy inverse  
http-request set-header X-Forwarded-Proto https  
http-request set-header X-Forwarded-Port 443  
http-request set-header X-Real-IP %[src]  
  
# Timeouts longs pour WebSocket (sessions vidéo)  
timeout server 3600s  
timeout tunnel 3600s  
timeout client-fin 30s  
timeout server-fin 30s  
  
# Serveur BBB  
server bbb1 10.29.113.101:80 check inter 10s fall 3 rise 2
```

### 5.3 Configuration pour TURN sur TLS (optionnel)

```
#-----  
# Frontend TURN TLS (port 5349)  
#-----  
frontend ft_turn_tls  
bind *:5349 ssl crt /etc/haproxy/certs/rber.bj.pem  
mode tcp  
default_backend bk_turn_tls  
  
backend bk_turn_tls  
mode tcp
```

```
server bbb1 10.29.113.101:5349 check
```

## 5.4 Validation et rechargement

```
# Vérifier la syntaxe
sudo haproxy -c -f /etc/haproxy/haproxy.cfg

# Recharger HAProxy (sans interruption)
sudo systemctl reload haproxy

# Vérifier le statut
sudo systemctl status haproxy

# Tester l'accès
curl -I https://visio.rber.bj/bigbluebutton/api
```



## 6. Configuration du firewall

### 6.1 Firewall sur la VM BBB (UFW)

```
# Sur la VM BBB (10.29.113.101)

# Autoriser SSH
sudo ufw allow 22/tcp

# Autoriser HTTP depuis HAProxy uniquement
sudo ufw allow from 10.29.112.100 to any port 80 proto tcp

# Autoriser TURN TLS depuis HAProxy
sudo ufw allow from 10.29.112.100 to any port 5349 proto tcp

# Autoriser les ports WebRTC UDP depuis Internet
sudo ufw allow 16384:32768/udp

# Activer le firewall
sudo ufw enable

# Vérifier les règles
sudo ufw status verbose
```

### 6.2 Configuration NAT sur le routeur/firewall périmétrique

Ces règles doivent être configurées sur le firewall qui gère l'IP publique 102.222.216.6.

```
# Règles NAT à ajouter (syntaxe conceptuelle)

# Règle existante - TCP 443 vers HAProxy
NAT TCP 102.222.216.6:443 -> 10.29.112.100:443

# Nouvelle règle - UDP WebRTC vers BBB
NAT UDP 102.222.216.6:16384-32768 -> 10.29.113.101:16384-32768

# Nouvelle règle - TURN TLS (si utilisé directement)
NAT TCP 102.222.216.6:5349 -> 10.29.112.100:5349
```

### 6.3 Exemple iptables (si applicable)

```
# Si le routeur utilise iptables

# NAT pour WebRTC UDP
iptables -t nat -A PREROUTING -d 102.222.216.6 -p udp \
--dport 16384:32768 -j DNAT --to-destination 10.29.113.101

# Masquerade pour le retour
iptables -t nat -A POSTROUTING -s 10.29.113.0/24 -j MASQUERADE

# Sauvegarder
iptables-save > /etc/iptables/rules.v4
```

## 7. Configuration TURN/STUN

TURN permet aux clients derrière des NAT stricts de participer aux visioconférences. C'est essentiel pour une compatibilité maximale.

### 7.1 Configuration de coturn

```
# Sur la VM BBB, éditer la configuration TURN
sudo nano /etc/turnserver.conf

# Contenu complet:
listening-port=3478
tls-listening-port=5349

listening-ip=0.0.0.0
relay-ip=10.29.113.101

# IMPORTANT: IP publique / IP privée
external-ip=102.222.216.6/10.29.113.101

min-port=16384
max-port=32768

realm=visio.rber.bj
server-name=visio.rber.bj

# Authentification
lt-cred-mech
use-auth-secret
static-auth-secret=GENERER_UN_SECRET_ALEATOIRE_ICI

# Certificats (si TURN TLS direct, sinon commenter)
# cert=/etc/letsencrypt/live/visio.rber.bj/fullchain.pem
# pkey=/etc/letsencrypt/live/visio.rber.bj/privkey.pem

# Logging
log-file=/var/log/turn.log
verbose

# Sécurité
no-multicast-peers
denied-peer-ip=0.0.0.0-0.255.255.255
denied-peer-ip=10.0.0.0-10.255.255.255
denied-peer-ip=172.16.0.0-172.31.255.255
denied-peer-ip=192.168.0.0-192.168.255.255

# Exception pour le réseau local RBER
allowed-peer-ip=10.29.113.0-10.29.113.255
```

### 7.2 Générer le secret TURN

```
# Générer un secret aléatoire
openssl rand -hex 32

# Exemple de sortie:
# alb2c3d4e5f6g7h8i9j0k1l2m3n4o5p6q7r8s9t0ulv2w3x4y5z6

# Utiliser ce secret dans:
# 1. /etc/turnserver.conf (static-auth-secret)
# 2. /etc/bigbluebutton/bbb-web.properties (voir ci-dessous)
```

## 7.3 Configurer BBB pour utiliser TURN

```
# Éditer la configuration BBB
sudo nano /etc/bigbluebutton/bbb-web.properties

# Ajouter/modifier:
stunServers=stun:visio.rber.bj:3478
turnServers=turn:visio.rber.bj:3478?transport=udp,turns:visio.rber.bj:5349?transport=tcp
turnSecret=VOTRE_SECRET_TURN_ICI
```

## 7.4 Redémarrer les services

```
# Redémarrer coturn
sudo systemctl restart coturn
sudo systemctl enable coturn

# Vérifier le statut
sudo systemctl status coturn

# Redémarrer BBB
sudo bbb-conf --restart
```

## 8. Tests et validation

### 8.1 Test de l'API BBB

```
# Test local (depuis la VM BBB)
curl http://localhost/bigbluebutton/api

# Test via HAProxy (depuis n'importe où)
curl https://visio.rber.bj/bigbluebutton/api

# Réponse attendue:
# <response>
# <returncode>SUCCESS</returncode>
# <version>3.0</version>
# </response>
```

### 8.2 Test de création de salle

```
# Récupérer le secret
SECRET=$(sudo bbb-conf --secret | grep Secret: | awk '{print $2}')

# Générer le checksum
CALL="create"
PARAMS="name=TestRoom&meetingID=test123"
CHECKSUM=$(echo -n "${CALL}${PARAMS}${SECRET}" | shasum | awk '{print $1}')

# Créer la salle
curl "https://visio.rber.bj/bigbluebutton/api/create?${PARAMS}&checksum=${CHECKSUM}"

# Vérifier les salles actives
CHECKSUM2=$(echo -n "getMeetings${SECRET}" | shasum | awk '{print $1}')
curl "https://visio.rber.bj/bigbluebutton/api/getMeetings?checksum=${CHECKSUM2}"
```

### 8.3 Test WebRTC et TURN

Tester depuis un navigateur web en accédant à Greenlight.

```
# Accéder à l'interface Greenlight
https://visio.rber.bj/

# Actions de test:
# 1. Créer un compte ou se connecter
# 2. Créer une nouvelle salle
# 3. Démarrer la salle
# 4. Activer le microphone
# 5. Activer la caméra
# 6. Partager l'écran

# Vérifier dans les logs BBB
sudo tail -f /var/log/bigbluebutton/bbb-web.log
sudo tail -f /var/log/kurento-media-server/kurento-media-server.log
```

### 8.4 Test de connectivité TURN

```
# Installer l'outil de test
sudo apt install coturn-utils

# Tester STUN
turnutils_stunclient visio.rber.bj

# Tester TURN UDP
```

```
turnutils_uclient -p 3478 -u testuser -W VOTRE_SECRET visio.rber.bj  
  
# Vérifier les logs TURN  
sudo tail -f /var/log/turn.log
```

## 8.5 Checklist de validation

| Test                  | Commande/Action                              | Résultat attendu  |
|-----------------------|--|-------------------|
| API accessible        | curl https://visio.rber.bj/bigbluebutton/api | SUCCESS           |
| Greenlight accessible | Navigateur → https://visio.rber.bj/          | Page de connexion |
| Création de salle     | Via Greenlight ou API                        | Salle créée       |
| Audio fonctionne      | Rejoindre une salle, activer micro           | Son transmis      |
| Vidéo fonctionne      | Activer la caméra                            | Image visible     |
| Partage écran         | Bouton partage écran                         | Écran partagé     |
| TURN actif            | Logs /var/log/turn.log                       | Connexions TURN   |

## 9. Intégration Moodle

### 9.1 Installation du plugin BBB dans Moodle

Le plugin BigBlueButton est inclus par défaut dans Moodle 4.0+. Si ce n'est pas le cas, l'installer manuellement.

```
# Vérifier si le plugin est installé
# Administration du site → Plugins → Vue d'ensemble des plugins
# Chercher "BigBlueButton"

# Si absent, télécharger depuis:
# https://moodle.org/plugins/mod_bigbluebuttonbn

# Installer via l'interface admin ou:
cd /var/www/moodle/mod
sudo -u www-data unzip bigbluebuttonbn.zip
sudo -u www-data php /var/www/moodle/admin/cli/upgrade.php
```

### 9.2 Configuration du plugin

Dans l'interface d'administration Moodle :

- Administration du site → Plugins → Modules d'activité → BigBlueButton
- Paramètres généraux :

| Paramètre                    | Valeur  |
|------------------------------|---|
| URL du serveur BigBlueButton | <a href="https://visio.rber.bj/bigbluebutton/">https://visio.rber.bj/bigbluebutton/</a> |
| Secret partagé               | [Secret obtenu avec bbb-conf --secret]  |
| Activer les enregistrements  | Oui (si souhaité)   |
| Participants max par défaut  | 150   |

### 9.3 Créer une activité BBB dans un cours

Une fois le plugin configuré, les enseignants peuvent ajouter des salles BBB à leurs cours.

```
# Procédure pour l'enseignant:
# 1. Aller dans le cours
# 2. Activer le mode édition
# 3. Ajouter une activité → BigBlueButton
# 4. Configurer:
# - Nom de la salle
# - Description
# - Options de modération
# - Options d'enregistrement
# 5. Enregistrer

# Les étudiants verront l'activité et pourront rejoindre
# quand l'enseignant aura ouvert la salle
```

### 9.4 Test de l'intégration

```
# Vérifier la connexion depuis Moodle
# Administration → Plugins → BigBlueButton → Tester la connexion

# Résultat attendu:
# "Connexion au serveur BigBlueButton réussie"

# En cas d'erreur:
# - Vérifier l'URL (avec le / final)
# - Vérifier le secret
# - Vérifier que HAProxy route correctement
# - Consulter les logs: /var/log/bigbluebutton/bbb-web.log
```

## 10. Maintenance et dépannage

### 10.1 Commandes de maintenance BBB

```
# Vérifier l'état complet
sudo bbb-conf --check

# Redémarrer tous les services
sudo bbb-conf --restart

# Nettoyer les anciens enregistrements (plus de 30 jours)
sudo bbb-record --delete-older-than 30

# Voir les réunions en cours
sudo bbb-conf --meetings

# Arrêter toutes les réunions
sudo bbb-conf --clean
```

### 10.2 Fichiers de logs importants

| Composant    | Fichier de log   |
|--------------|--|
| BBB Web      | /var/log/bigbluebutton/bbb-web.log                     |
| HTML5 Client | /var/log/bigbluebutton/html5/html5client.log           |
| Kurento      | /var/log/kurento-media-server/kurento-media-server.log |
| FreeSWITCH   | /var/log/freeswitch/freeswitch.log                     |
| TURN         | /var/log/turn.log                                      |
| Nginx        | /var/log/nginx/error.log                               |

### 10.3 Problèmes courants et solutions

#### *Problème : Audio ne fonctionne pas*

```
# Vérifier FreeSWITCH
sudo systemctl status freeswitch

# Redémarrer si nécessaire
sudo systemctl restart freeswitch

# Vérifier les logs
sudo tail -100 /var/log/freeswitch/freeswitch.log | grep ERR
```

#### *Problème : Vidéo/partage d'écran ne fonctionne pas*

```
# Vérifier Kurento
sudo systemctl status kurento-media-server

# Vérifier les connexions WebRTC
sudo netstat -tuln | grep -E "16384|32768"

# Vérifier les logs Kurento
sudo tail -100 /var/log/kurento-media-server/kurento-media-server.log
```



### Problème : Erreur 502 via HAProxy

```
# Vérifier que Nginx répond sur BBB
curl -I http://10.29.113.101/bigbluebutton/api

# Vérifier les logs HAProxy
sudo tail -100 /var/log/haproxy.log

# Vérifier le backend
echo "show stat" | sudo socat stdio /var/run/haproxy/admin.sock
```

## 10.4 Mise à jour de BBB

```
# Sauvegarder la configuration
sudo cp /etc/bigbluebutton/bbb-web.properties /root/bbb-web.properties.bak
sudo cp /etc/turnserver.conf /root/turnserver.conf.bak

# Mettre à jour BBB
sudo apt update
sudo apt upgrade bigbluebutton

# Ou réexécuter le script d'installation
sudo ./bbb-install.sh -v noble-300 -s visio.rber.bj -g

# Restaurer les configurations personnalisées si écrasées
sudo bbb-conf --restart
```

## Résumé de la configuration

| Élément                 | Valeur  |
|-------------------------|---|
| URL publique            | <a href="https://visio.rber.bj">https://visio.rber.bj</a> |
| IP publique             | 102.222.216.6   |
| IP HAProxy              | 10.29.112.100   |
| IP VM BBB               | 10.29.113.101   |
| Ports TCP (via HAProxy) | 443, 5349   |
| Ports UDP (direct)      | 16384-32768   |
| Secret API              | [À récupérer avec bbb-conf --secret]                      |
| Secret TURN             | [À générer avec openssl rand -hex 32]                     |

### Contacts support

Email : [support@rber.bj](mailto:support@rber.bj)

Documentation BBB : <https://docs.bigbluebutton.org>

Document généré le 17 décembre 2025 - RBER Bénin