

Provisioning of a Strato server

- At strato.de
 - Ordered Linux V40
 - 8 CPU vCore
 - 32 GB RAM
 - 800 GB SSD
 - 500 MBit/s
 - The main limiting factor is the Virtual Server limit on the number of threads ("numproc"), which can be seen when executing ``sudo cat /proc/user_beancounters``
 - The V30 server has a numproc limit of 700, which is too low to even start BBB.
 - According to Strato support, a V40 server usually has a numproc limit of 1100, but that may differ depending on the virtualisation framework. That's why it should be checked straight away before proceeding.
- At <https://config.stratoserver.net/>
 - Installed "Ubuntu 16.04 LTS 64bit" (*without* Plesk)
- At domain provider
 - Created a new subdomain with an A-record for that server's IP address
- On server
 - Login to server via ssh:
`ssh root@<IP address>`
 - Create a new sudo user:
`adduser <user name>`
`usermod -aG sudo <user name>`
 - Optional: Add ssh public key for future logins
`su - <user name>`
`mkdir .ssh`
`chmod 700 .ssh`
`<copy/paste public key into .ssh/authorized_keys>`
`chmod 644 authorized_keys`

Pre-installation of BBB

- Ensure [ports are open and host is added](#)
 - Ubuntu does not have a firewall by default, so no action was required here.
 - The netcat listening commands requires the ``p`` parameter on the Ubuntu version:
`netcat -lp 7443`
`netcat -ulp 17000`
- Execute the [pre-installation checks](#). The following changes were required

- Change of locale to “en_US.UTF-8” via `update-locale` and `systemctl set-environment` commands
- The hostname was provided above, but the creation of the SSL certificate via Let’s Encrypt should be postponed until after the installation of BBB.
- Install nginx (as `nginx` command was not found)
 - sudo apt-get update
 - sudo apt-get install nginx

Install BBB

- Follow the [manual installation steps](#)
 - The NodeJS version 8.x is very old, but you can ignore the warning. 8.4 is the version specified in the [BBB webhooks code](#)
 - On one of two servers, I needed to run `sudo apt-get install software-properties-common`, as add-apt-repository was not a known command!?
 - Otherwise installation was successful and I set html5 as the default as suggested
 - The first run failed as freeswitch could not be started. [Disabling the CPUSchedulingPolicy](#) did the trick.
- Restart the server once after installation with `sudo reboot`
- Ensure to also execute the [final steps](#), in particular the assignment of the host name
 - Without it, connections to BBB via the browser won’t work and the `sudo bbb-conf --check` command will return “Could not connect to the configured hostname/IP address”.
 - After doing this, the warning “IP does not match” comparing the IP with the host name can be ignored.
- Create the directory for the SSL certificate
 - sudo mkdir /etc/nginx/ssl/
- [Create an SSL certificate](#)
 - This is actually a hard requirement
 - Ignore the section [Configure nginx to use HTTPS](#) if you use Let’s Encrypt
 - Call `systemctl reload nginx` after making the suggested changes to the nginx config

Customisations

- There are multiple [BBB customisations](#) that could be useful.
- Increasing the “Max open file limit” for the redis service?
 - At a later stage, the redis service may hit the “Max open file limit”.
 - It’s set to 4096 (`sudo cat /proc/\$(pgrep redis)/limits`), but the recommended value is as high as 65536.
 - This *may* be high enough for BBB and one could wait until this limit is hit before increasing it. `sudo ls -l /proc/\$(pgrep redis)/fd/ | wc -l` shows the current usage, which has a baseline of about 50.

- It can be increased by adding the following line to the [Service] section of `/etc/systemd/system/redis.service` resolved the issue: ``LimitNOFILE=65536``
- This new setting can be applied with:
``sudo systemctl daemon-reload && sudo systemctl restart redis``
- See [this post](#) for more details.

Hitting the the numproc limit

- This happened on the V30 server during BBB startup, but may also happen on the V40 server of BBB spins up more threads on high load. I am not sure if that would be the case.
- All or most BBB spin up multiple parallel threads (up to 182 for the red5 media server!) to handle high loads.
- It should be possible to reduce this, e.g. by setting the max-threads property for the services using Tomcat Catalina, but that did not work and there have been no suggestions yet from in the BBB setup group to [this thread](#).
- The symptoms of the limit being hit are out of memory errors in the logs. The exact error messages vary:
 - “java.lang.OutOfMemoryError: unable to create new native thread” (in `/var/log/syslog` and less frequently in `/var/log/bigbluebutton/bbb-web.log`)
 - “too many open files” or “cannot fork” or “fork: Cannot allocate memory” (in the terminal)
- The numproc limit and any violations of it can be watched with ``watch -n1 sudo cat /proc/user_beancounters``