

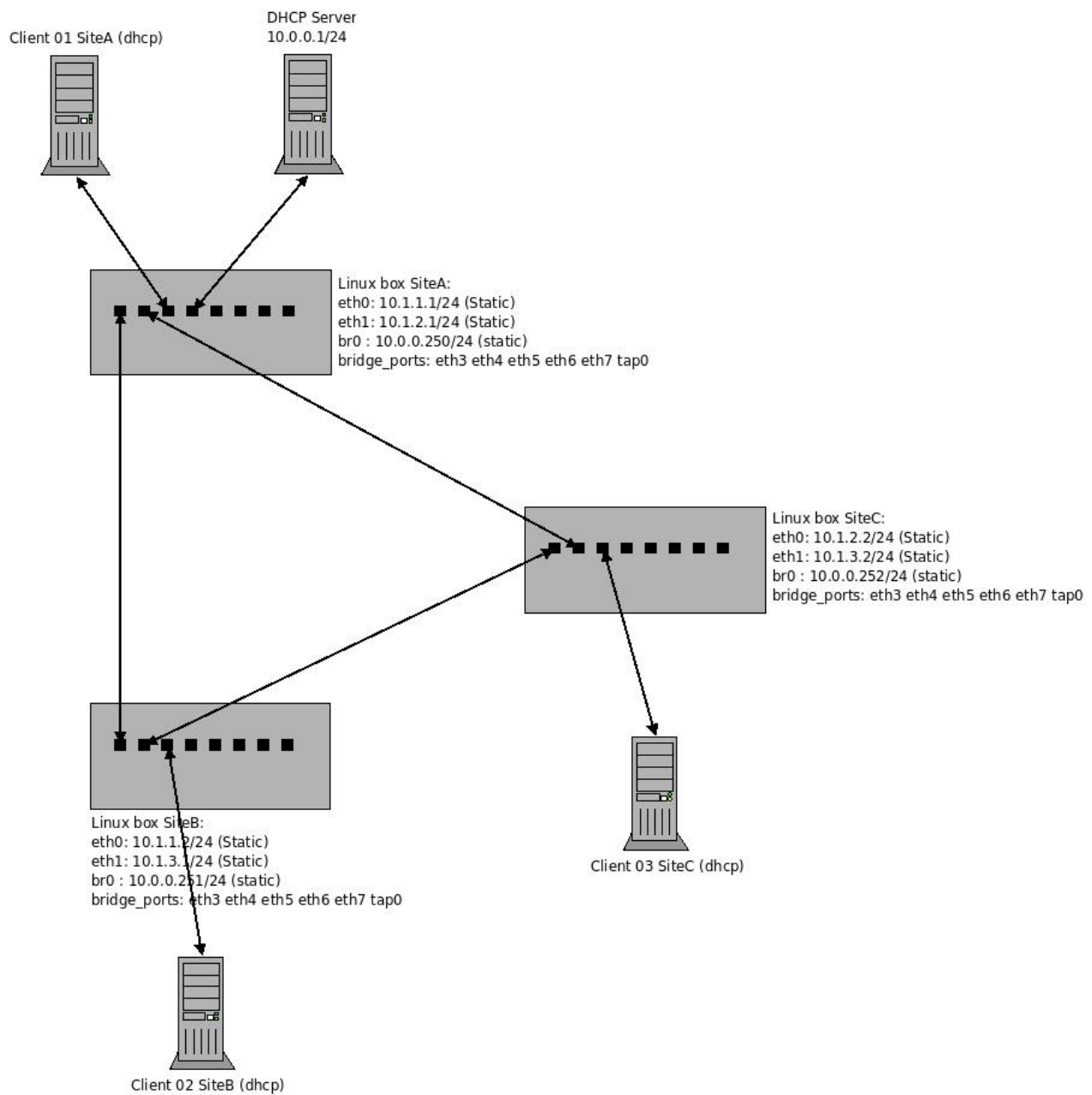
## Using CloudVPN for LAN to LAN secure connection in Ubuntu Jaunty

I want to connect 3 sites with secure line, using ethernet level and mesh capability.

### Requirement:

- 3 Linux box with at least 4 NIC
- CloudVPN from git (git clone <http://e-x-a.org/repos/cloudvpn.git>)
- Ubuntu Jaunty server latest kernel
- bridge-utils, autoconf, libnftls-dev, git-core

### Site configuration:



*Illustration 1: Site configuration*

## CloudVPN configuration:

### 1. Site A:

- a `#apt-get install bridge-utils autoconf libgnutls-dev git-core`
- b `#git clone http://e-x-a.org/repos/cloudvpn.git`
- c `#cd cloudvpn`
- d `#autoconf`
- e `#!/autogen.sh`
- f `#!/configure && make && make install`
- g `#mkdir /etc/cloudvpn`
- h `#cp ./extras/init-debian/cloudvpn /etc/init.d/cloudvpn & chmod +x /etc/init.d/cloudvpn`
- i Create the VPN keys following instruction on the CloudVPN web site (<http://www.e-x-a.org/?view=cloudvpn-howto>)
- j Create config file as `siteA.cloud`, save to `/etc/cloudvpn`
- k The content of config file:

```
#!/usr/local/bin/cloud -@include

# load the keys you generated in step 1
ca /etc/cloudvpn/ca.crt
key /etc/cloudvpn/mynode.key
cert /etc/cloudvpn/mynode.crt
dh /etc/cloudvpn/dh1024.pem

# now, if this is a server, assign it to listen on CloudVPN port 15135
listen 10.1.1.1 15135
listen 10.1.2.1 15135

# you can also listen on IPv6, and also repeat 'listen'
# options as many times you need it.
# listen :: 15136
# listen 10.0.0.1 36324

# if this is not a server, you have to connect to something.
# connect mydarknet.myserver.com 15135

# you can connect to multiple servers - it's beneficial for splitting load.
connect 10.1.1.2 15135
connect 10.1.2.2 15135
#connect serv1.dark.net 15135
#connect serv2.dark.net 15135
#connect ipv6.dark.net 15136

# to see what is being connected and routed, let's periodically export
# status to a text file. You can watch it when Cloud is running.

status-file /var/log/status.txt

# regenerate stats every minute. Please note that all time values in CloudVPN
# are measured in microseconds. Hence 7 zeroes.
status-interval 60000000

# now, make a gate that the gate-clients can use to connect.
# we will later connect to this socket with a Ethernet client,
# that will use cloud to transport packets.
# (don't mistake this with "clients" from client/server model. This here is
# much more like mesh/client.)
#
# If address contains with a slash, it means that we use unix(7) local sockets
gate /var/run/gate.sock

# You can also allow computers on local LAN to use your computer as a gate.
# 15137 is a standard gate port.
#gate 192.168.0.1 15137
```

l Now, we configure the interface, open file `/etc/network/interfaces` with your favorite editor

m Content of the *interfaces* as follow:

```
auto eth0
iface eth0 inet static
    address 10.1.1.1
    netmask 255.255.255.0

auto eth1
iface eth1 inet static
    address 10.1.2.1
    netmask 255.255.255.0

auto br0
face br0 inet static
    address 10.0.0.250
    netmask 255.255.255.0
    pre-up /etc/init.d/cloudvpn start
    pre-up /usr/local/bin/ether -gate /var/run/gate.sock -bridge yes &
    pre-up ifconfig eth2 down
    pre-up ifconfig eth3 down
    pre-up ifconfig eth4 down
    pre-up ifconfig eth5 down
    pre-up ifconfig eth6 down
    pre-up ifconfig eth7 down
    pre-up brctl addbr br0
    pre-up brctl addif br0 eth2
    pre-up brctl addif br0 eth3
    pre-up brctl addif br0 eth4
    pre-up brctl addif br0 eth5
    pre-up brctl addif br0 eth6
    pre-up brctl addif br0 eth7
    pre-up brctl addif br0 tap0
    pre-up ifconfig eth2 0.0.0.0 promisc
    pre-up ifconfig eth3 0.0.0.0 promisc
    pre-up ifconfig eth4 0.0.0.0 promisc
    pre-up ifconfig eth5 0.0.0.0 promisc
    pre-up ifconfig eth6 0.0.0.0 promisc
    pre-up ifconfig eth7 0.0.0.0 promisc
    pre-up ifconfig tap0 0.0.0.0
    pre-down kill -9 `pidof ether`
    pre-down /etc/init.d/cloudvpn stop
    post-down ifconfig eth2 down
    post-down ifconfig eth3 down
    post-down ifconfig eth4 down
    post-down ifconfig eth5 down
    post-down ifconfig eth6 down
    post-down ifconfig eth7 down
    post-down ifconfig br0 down
    post-down brctl delif br0 eth2
    post-down brctl delif br0 eth3
    post-down brctl delif br0 eth4
    post-down brctl delif br0 eth5
    post-down brctl delif br0 eth6
    post-down brctl delif br0 eth7
    post-down brctl delbr br0
```

n Reboot the machine

2. Site B:

- a Do the same thing as Site A point a to m
- b Before create the VPN key, copy the *ca.key* from the Site A to site B box, use it as key to create certificate
- c Adjust the config file to SiteB.cloud and the IP address of the servers at *listen* and pointed connection *connect*
- d Adjust the *interfaces* file to the proper IP following the illustration

e Do this procedure for Site C

**Testing:**

1. Check the log file `#tail -f /var/log/status.txt`
2. Look at the connection, it should have connection and route
3. Use static IP of the client to test the connection, if you could ping the server in the SiteA, it means your connection works properly
4. Then, use DHCP to get the IP from DHCP server

Viola...

Connection works with secure line...

If you have any troubles, go to IRC at server [irc.freenode.org](http://irc.freenode.org) channel `#cloudvpn`

Fadjar Tandabawana  
[fadjar340@gmail.com](mailto:fadjar340@gmail.com)