



Kamailio Start to Finish

Document v1.2 - Last edited 12/16/2008 by Chris Sherwood

I was recently tasked with figuring out a way to load balance multiple trixbox Pro servers, and after some research, I came across Kamailio. Kamailio is an open source SIP server that can process thousands of call setups per seconds. It can be used as a SIP load balancer, registrar, location server, proxy server, redirect server, gateway, or advanced VoIP application server.

So great...let's use Kamailio to set up my load balancer. How do we do that? Well...unfortunately, like most open-source projects, it suffers from a severe lack of useful documentation. Let's face it...programmers/developers are awesome...but they're not great at documentation. Personally, I like documentation to tell me "Hey stupid...use THIS operating system, and download using THIS link, and then click here and type this..." etc. You get the picture. Of course, nothing like that exists for Kamailio...so here's my little contribution to the community. THIS documentation will tell you how to get Kamailio up and running start to finish if you follow these instructions exactly. If you know what you are doing, you can expand upon them and do your own thing, but don't complain to me when you've driven off the map and are now lost.

Onto the installation...

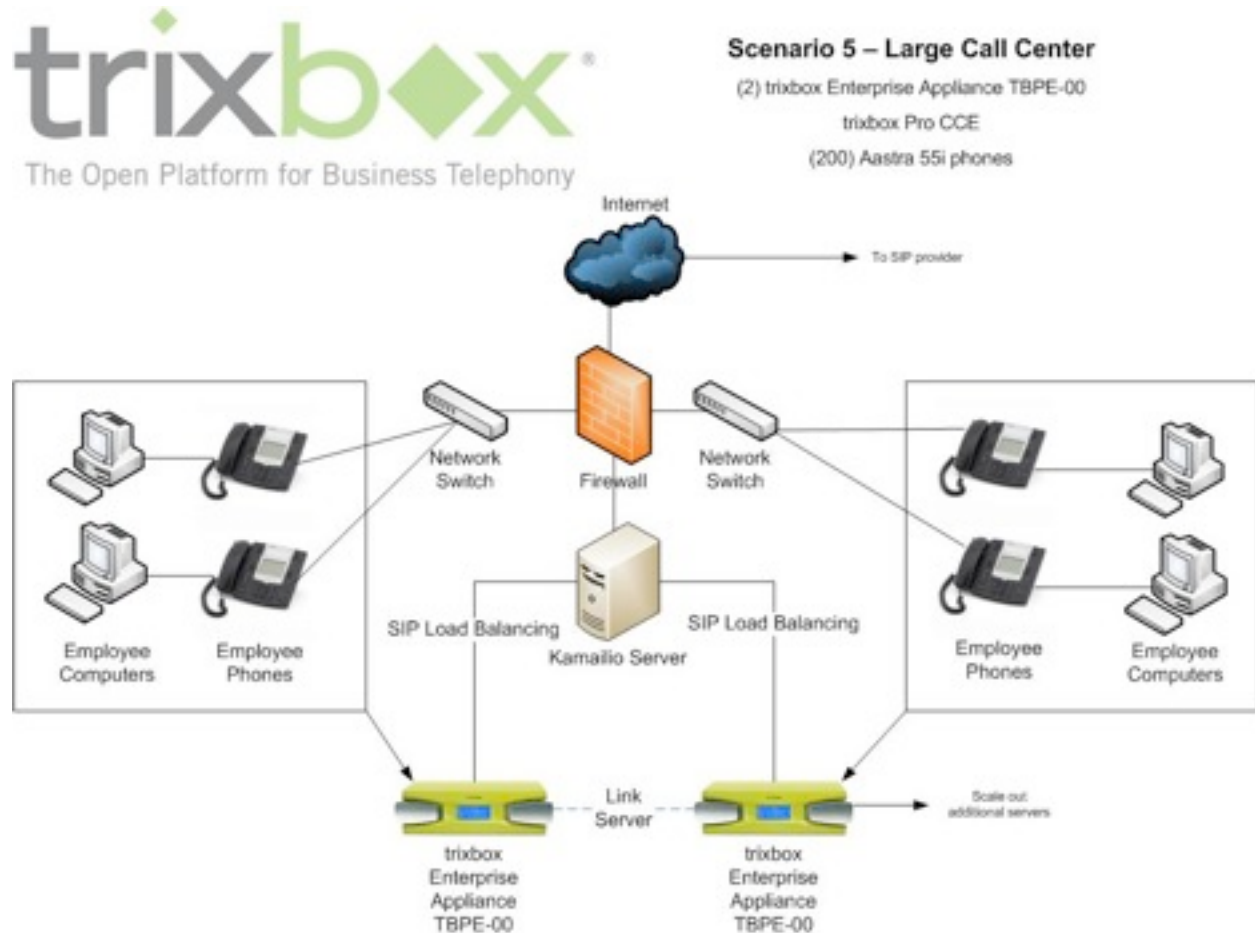
-
- 1.0 - Overview
 - 2.0 - Install Fedora 9
 - 3.0 - Maintenance
 - 3.1 - Install Webmin
 - 4.0 - DNS configuration
 - 5.0 - MySQL configuration
 - 6.0 - Kamailio Installation
 - 6.1 - Kamailio configuration
 - 6.2 - Create Kamailio MySQL database entries
 - 6.3 - Start Kamailiolivepage.apple.com
 - 6.4 - Add some users to Kamailio

- 7.0 - Phone configuration
- 8.0 - Conclusion

1.0 - Overview

I chose Fedora 9 as my operating system for no other reason than I like it, and I'm familiar with Red Hat flavors. This installation will include Kamailio 1.4 with MySQL authentication for users and trunks (otherwise, it is just a dumb SIP switch).

As I said earlier, I originally looked at Kamailio for a SIP load balancing solution similar to the diagram below:



The Kamailio server in the middle will load balance SIP connections between the multiple tribox Pro PBX systems for our large call center. This provides a highly scalable call center environment leveraging the efficiency of the Kamailio server and the power of tribox Pro Call Center Edition.

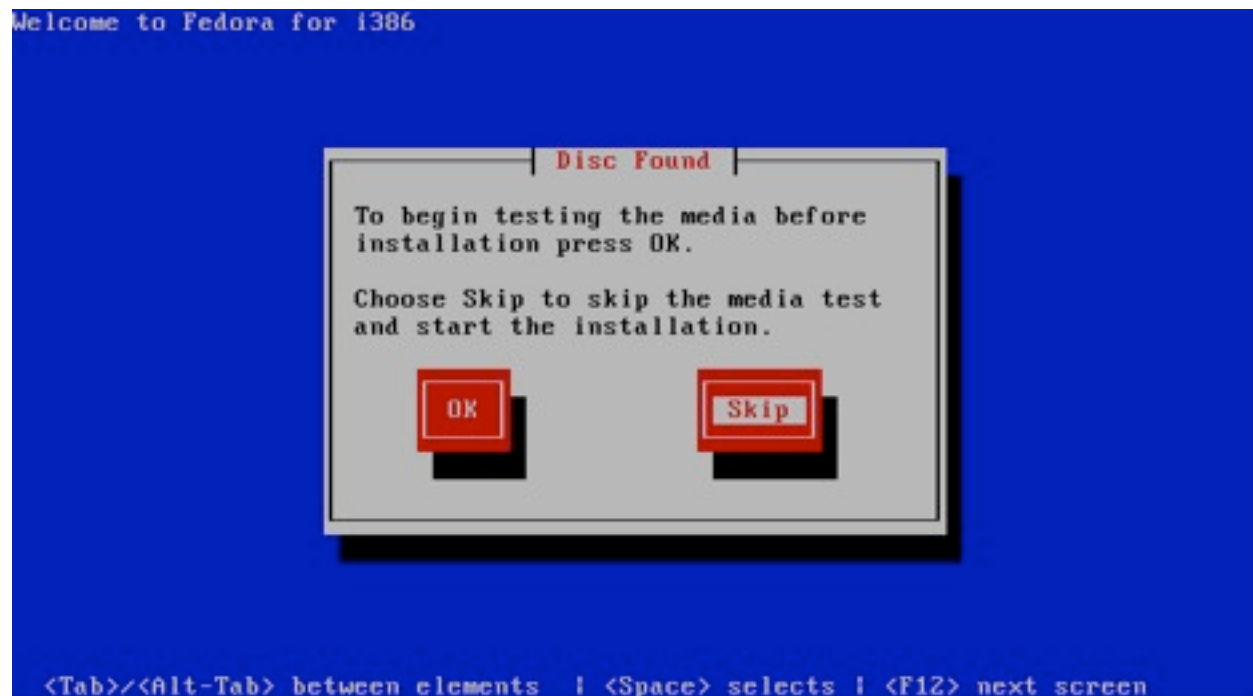
2.0 - Install Fedora 9

The first step is to download and install Fedora 9. I use bittorrent as it seems to be the fastest way to get everything. Download from here: <http://fedoraproject.org/en/get-fedora>

Once you have downloaded the DVD ISO and have burned it to DVD, boot to the DVD and the installation begins.

*** NOTE: This WILL destroy your hard drive...if there is any data on the server you are using that you don't want to erase, stop now and back your stuff up.

First, choose to test the DVD media or skip it. I always skip it, but I live on the edge.



You will eventually end up at the main Fedora 9 installation screen. Click Next.



Copyright © 2003-2008 Red Hat, Inc. and others. All rights reserved.

← Back

Next →

Next select your language (Next) and keyboard (I chose US English for both) and click Next. You are now warned that your hard drive is about to be erased. Click Yes.

For PBX systems, I always set a static IP address, so when the networking window pops up, I first click 'Edit.' I enter in my IP addressing information and disable IPv6 and then click OK.

Edit Interface

Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
Hardware address: 00:0c:29:30:64:ff

Enable IPv4 support

- Dynamic IP configuration (DHCP)
- Manual configuration

IP Address: / Prefix (Netmask):

Enable IPv6 support

- Automatic neighbor discovery
- Dynamic IP configuration (DHCPv6)
- Manual configuration

IP Address: / Prefix:

Now I can finish the network configuration by entering in my hostname, Gateway and DNS information. Once finished, click next.

fedora

Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix	Edit
<input checked="" type="checkbox"/>	eth0	192.168.200.30/24	Disabled	

Hostname

Set the hostname:

automatically via DHCP

manually (e.g., host.domain.com)

Miscellaneous Settings

Gateway:

Primary DNS:

Secondary DNS:

Select your time zone and click Next.

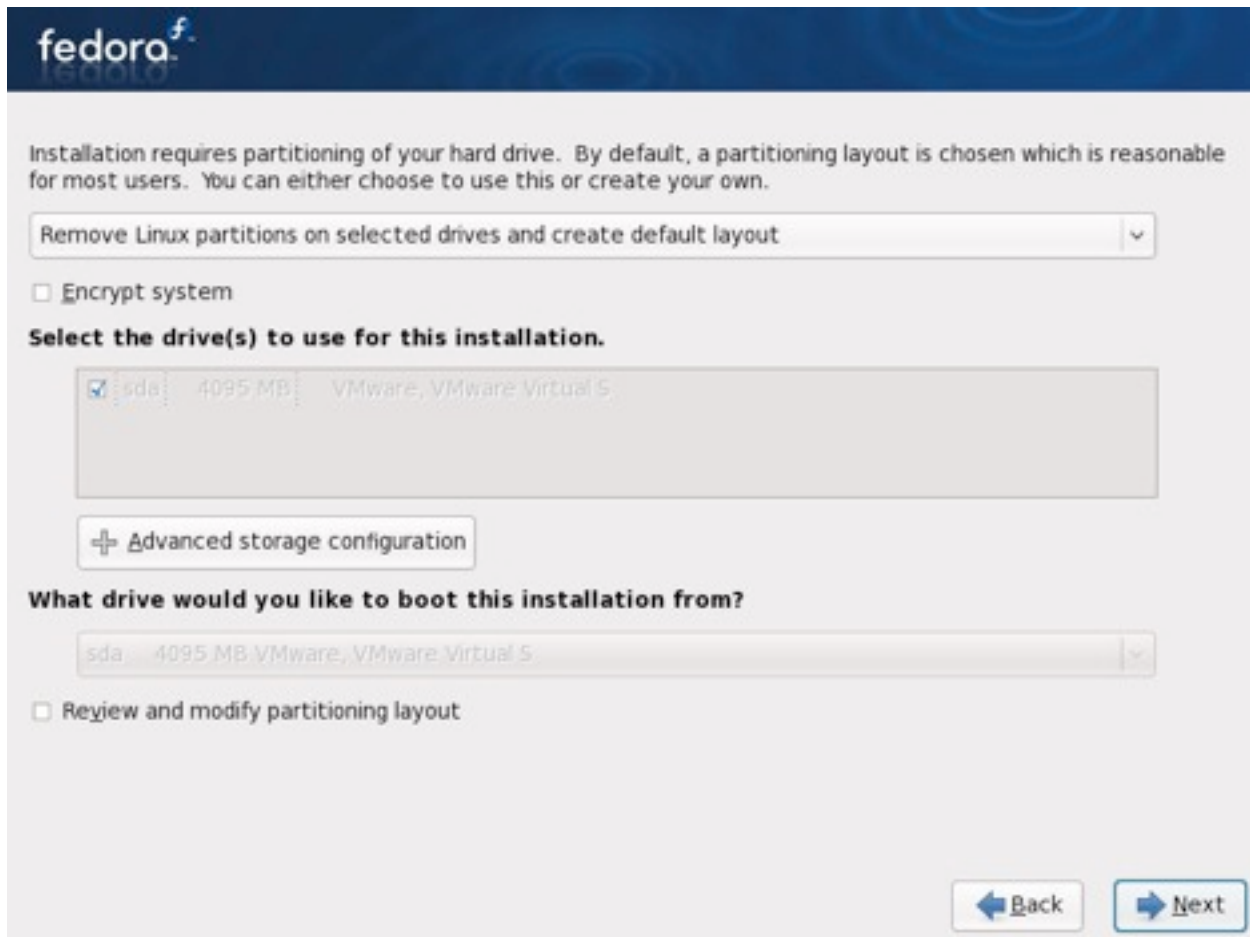
Enter in your root password and click Next.

 The root account is used for administering the system. Enter a password for the root user.

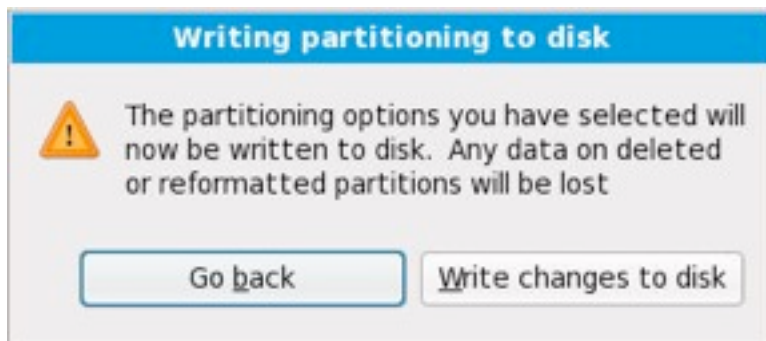
Root Password:

Confirm:

For the partition configuration, take defaults and click Next.



When the warning dialog pops up, click 'Write changes to disk.'



Time to choose packages...pay close attention to these steps! It matters...not just to me...to the world.

Uncheck 'Office and Productivity,' and select 'Customize Now.' Click Next.


The default installation of Fedora includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?


Office and Productivity

- Software Development
- Web server

Please select any additional repositories that you want to use for software installation.

- Additional Fedora Software
- Fedora

 Add additional software repositories

 Modify repository

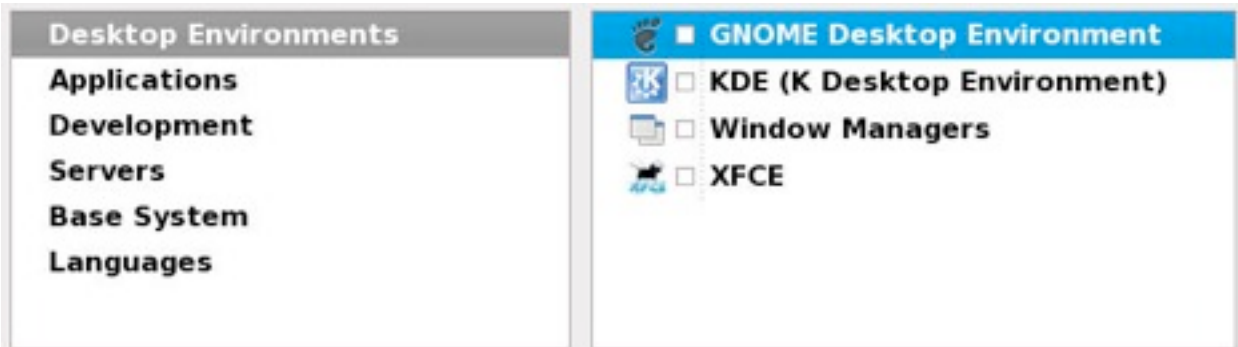
You can further customize the software selection now, or after install via the software management application.

- Customize later
- Customize now

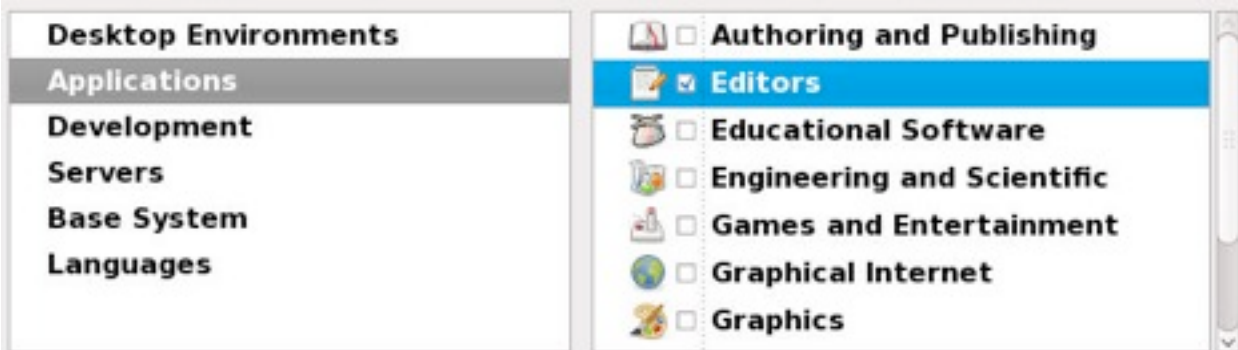
 Back

 Next

Under 'Desktop Environments' uncheck everything. We're real Linux people...we don't need no stinkin' GUI.



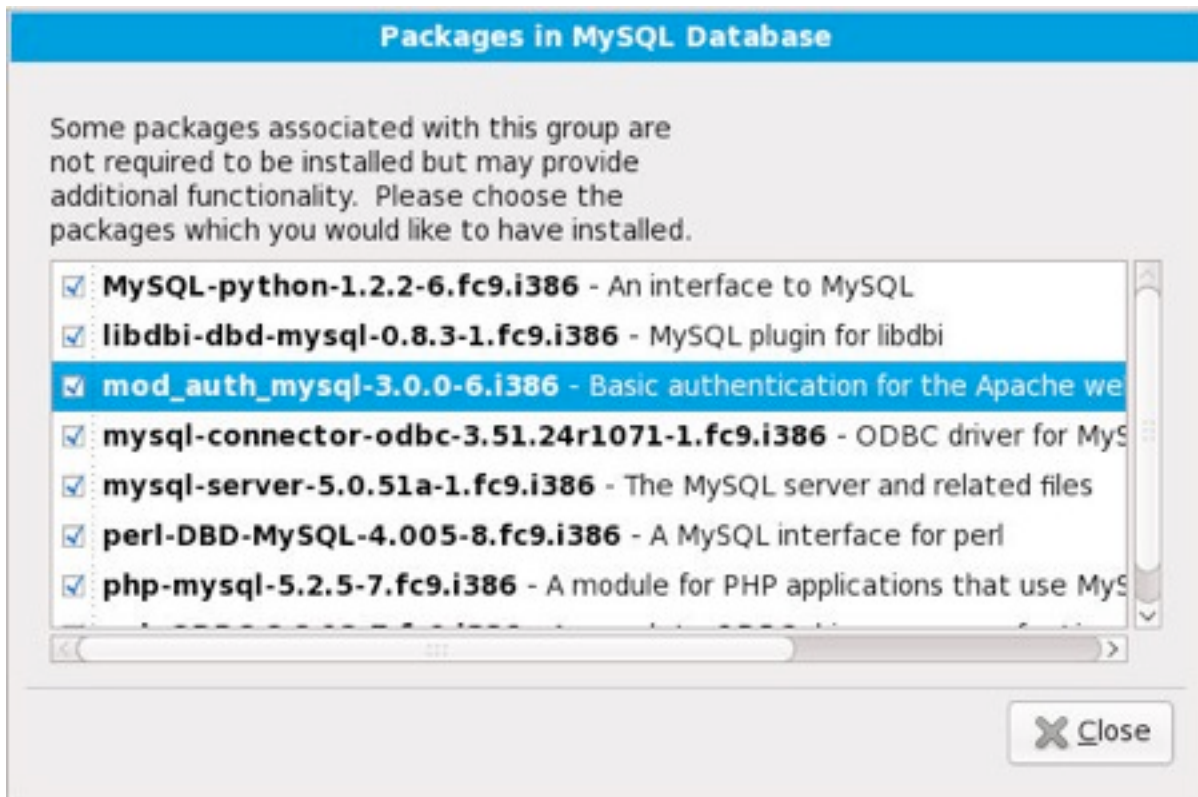
Under 'Applications,' uncheck everything except for Editors and Text-based Internet.



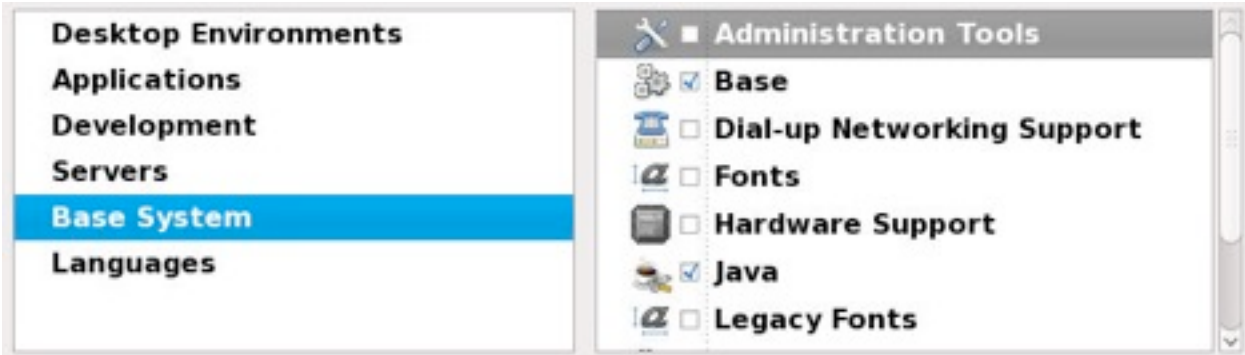
Under 'Development,' check Development Libraries and Development Tools.



Under Servers, check DNS Name Server, Mail Server, MySQL Database, and Web Server. To further configure Servers, click on MySQL Database and click 'Optional Packages.' In Optional Packages, make sure everything is checked. Click 'Close.'



Under 'Base System' uncheck Administration Tools, Dial-up Networking Support, Fonts, Hardware Support, Legacy Fonts, and X Windows System. (Basically, only leave Base and Java checked).



Click 'Next.' Installation begins...go get a cup of coffee...this takes a while.



Once finished, click 'Reboot.'

3.0 - Maintenance

You should now be able to SSH to the box with your root password. You can also use the console. Log into the server and let's do some maintenance.

First, we're going to disable the firewall and we're also going to disable SELinux. These programs mess with ports. In our example, I'm assuming that your Kamailio server is behind a firewall, and therefore already protected. Configuring iptables is beyond the scope of this document...so we're just going to disable it.

Run the following commands in the Linux CLI:

```
chkconfig iptables off
chkconfig ip6tables off
service iptables stop
service ip6tables stop
```

Disable SELinux:

```
nano /etc/selinux/config
```

Find the line that says `SELINUX=enforcing` and change it to `SELINUX=disabled`.
CTRL+X to exit and then Y when asked to save.

REBOOT!

Make sure networking is enabled and on (needs to be done from the console if you're not able to connect remotely):

```
chkconfig --levels 2345 network on  
service network start
```

Import GPG keys

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
```

Install an additional program

```
yum -y install mysql-devel
```

Update all packages

```
yum -y update
```

You may only get 1 package to update the first time you run that command. If that's the case, once that package updates, run the update again:

```
yum -y update
```

Go grab another cup of coffee...this takes some time.

3.1 - Install Webmin

Webmin is a great HTTP GUI interface for configuring various Linux services and settings. It makes configuring DNS especially easy.

```
cd /usr/src  
wget http://prdownloads.sourceforge.net/webadmin/webmin-1.430-1.noarch.rpm  
rpm -ivh webmin-1.430-1.noarch.rpm
```

Update the 'wget' command with whatever version of Webmin is current at the time you are installing.

You should now be able to connect to `https://1.2.3.4:10000` (where 1.2.3.4 is your server's IP address). Note the connection to `https`, and not `http`.

Log in as root.

4.0 - DNS configuration

We are going to use our Kamailio server for DNS, so let's enable and configure our Bind 9 server (should already be installed).

Start DNS and make sure it starts when the system is booted:

```
chkconfig --levels 235 named on  
service named start
```

Now, open Webmin and go to Servers --> Bind DNS Server.

Click on Forwarding and Transfers to add a forwarder.

I'm going to set a forwarder (for unresolved requests) address of 4.2.2.2. Click 'Save.'

The screenshot shows the 'Global forwarding and zone transfer options' configuration window. It features a table for 'Servers to forward queries to' with columns for 'IP address' and 'Port (optional)'. The first row contains '4.2.2.2' in the IP address field. Below the table are several radio button options: 'Lookup directly if forwarders cannot?' (Default selected), 'Maximum zone transfer time' (Default selected), 'Zone transfer format' (Default selected), and 'Maximum concurrent zone transfers' (Default selected).

Now click on Addresses and Topology. By default, BIND is configured to answer on the localhost address (127.0.0.1) only, so we're going to add our eth0 IP address as well. Add the address after 127.0.0.1 (with a space in between) and then click 'Save.'

The screenshot shows the 'Global address and topology options' configuration window. The 'Addresses' field is highlighted, showing '127.0.0.1 192.168.200.30'. Other fields include 'Port' (53), 'Source IP address for queries' (Default), and 'Source port for queries' (Default).

Now that those two items are configured, click the 'Apply Changes' button at the bottom of the main Bind DNS config screen. Your BIND DNS server is now configured to answer requests on your eth0 IP address, and forward unresolved requests to a public DNS server.

5.0 - MySQL configuration

First, we need to start MySQL and configure it so that it starts automatically upon system boot:

```
chkconfig --levels 2345 mysqld on
service mysqld start
```

Now we need to give MySQL a root password:

```
mysql -u root -p
```

(password is blank by default)

```
mysql> use mysql;
mysql> update user set password=PASSWORD("new_password_here") where
User='root';
mysql> flush privileges;
mysql> quit
```

6.0 - Kamailio Installation

It's finally time to install Kamailio. We are going to install the non-TLS version (we don't need the extra security) so here we go:

```
cd /usr/src
wget http://www.kamailio.org/pub/kamailio/1.4.1/src/kamailio-1.4.1-notls_src.tar.gz
tar zxvf kamailio-1.4.1-notls_src.tar.gz
cd kamailio-1.4.1-notls_src
```

Now we are ready to build and install. First, we need to 'make.'

```
make prefix=/ include_modules="db_mysql" all
```

Let's take a look at that make string. The 'prefix=/' means that we're using the root directory (/) as the installation root. This means that our config files will be in /etc/kamailio. If you do not use the 'prefix=/' option, your files will install to /usr/local/etc/kamailio. Keep in mind that whatever prefix= you use, you need to use the same prefix for the 'make install' command coming up next.

I have also included the 'include_modules="db_mysql"' option so that we are building the makefile with MySQL authentication enabled.

The 'all' means we are including all default modules with out installation.

Time to install:

```
make prefix=/ include_modules="db_mysql" all install
```

***NOTE: If you get any db_mysql errors, make sure you have installed the mysql_devel package as mentioned above.

6.1 - Kamailio configuration

First we need to set up some environmental variables that Kamailio needs. These are located in the file /etc/kamailio/kamctlrc. Let's edit that file.

```
nano /etc/kamailio/kamctlrc
```

Uncomment SIP_DOMAIN=domain.com and set it to your server's domain (in my case, it is kamailio.local).

Uncomment DBENGINE=MYSQL.

CTRL+X to exit and 'Y' when asked to save.

6.2 - Create Kamailio MySQL database entries

```
/sbin/kamdbctl create
```

Enter in your MySQL root password when prompted, and answer 'y' to both of the questions you are asked.

Now we need to configure Kamailio to use MySQL authentication. We do this by uncommenting some lines in the /etc/kamailio/kamailio.cfg.

```
nano /etc/kamailio/kamailio.cfg
```

Uncomment the following lines:

```
loadmodule "db_mysql.so"

loadmodule "auth.so"
loadmodule "auth_db.so"

modparam("usrloc", "db_mode", 2)
modparam("usrloc", "db_url", "mysql://openser:openserrw@localhost/openser")

modparam("auth_db", "calculate_ha1", yes)
modparam("auth_db", "password_column", "password")
modparam("auth_db", "db_url", "mysql://openser:openserrw@localhost/openser")
modparam("auth_db", "load_credentials", "")

if (!www_authorize("mydomain.com", "subscriber"))
{
www_challenge("mydomain.com", "0");
exit;
}

if (!check_to())
{
sl_send_reply("403","Forbidden auth ID");
exit;
}
```

Make sure you set the '**mydomain.com**' to your own domain (2 different places).

CTRL+X to exit and 'Y' when asked to save.

6.3 - Start Kamailio

Phew! About time...if you have done everything correctly, the following command will start your Kamailio server.

```
kamctl start
```

If everything starts fine, you'll see 'INFO: started (pid: XXXX)' where XXXX is the process ID. If you get an error, go back through the instructions and find where you messed up.

We can now monitor the status of our Kamailio server by running:

```
kamctl moni
```

6.4 - Add some users to Kamailio

Now it's time to create some users and fire up some phones. Here are the users I want to add:

Extension Password

1000 12345

1001 12345

1002 12345

To do this, we need to run:

```
kamctl add (username/extension number) (password)
```

So, for extension 1000, we would do:

```
kamctl add 1000 12345
```

When prompted for the MySQL password for user 'openser@localhost,' use the default password 'openserrw'.

Repeat that process for all of your extensions/users.

7.0 - Phone configuration

For our testing purposes, I'm going to use the free X-Lite softphone from Counterpath. Download and install the phone (it is best to do this on two systems so that you can actually make a call from one phone to another). Open 'SIP Account Settings' and then double-click on the 'Acct #1' line (or click Add if none exists).

Use these settings:

Display Name: (whatever)

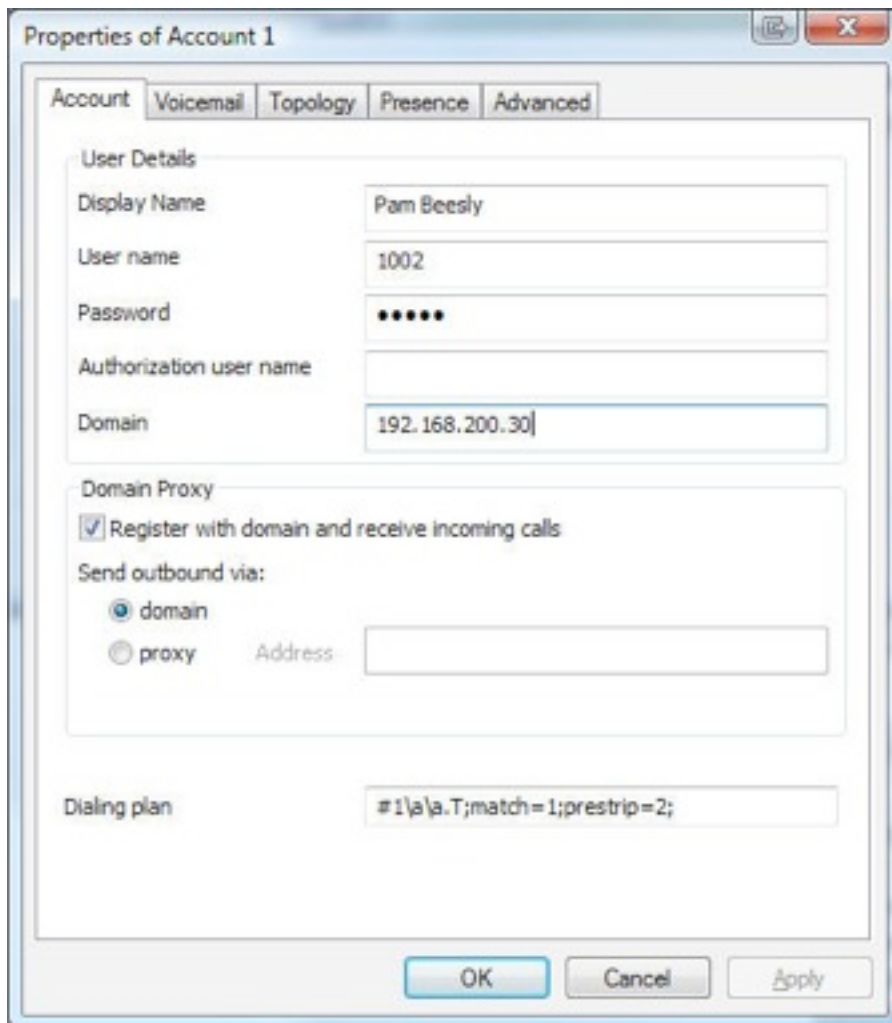
User Name: 1002 (your desired extension added to MySQL above)

Password: 12345 (the password for the corresponding extension)

Domain: 192.168.200.30 (the IP address or FQDN of your server)

Check 'Register with domain and receive incoming calls'

Send outbound via: domain



Click 'Apply' and OK.

Repeat this for another extension, and you can now call from one extension to another...congratulations...your Kamailio server is up and running.

8.0 - Conclusion

Ok...so now your Kamailio server is up and running? What next? That's up to you. I'm not going to cover further configuration of Kamailio in this document because every configuration is different.

A note from the author:

Hey everyone!

If you have found this document useful, and plan on purchasing some VoIP equipment, I would greatly appreciate if you use the banner link for Voipsupply.com below. I have purchased equipment from Voipsupply.com many times, and I have always found their pricing to be great and their shipping fast, so I definitely recommend them. By using the link below, you won't be raising YOUR purchase price at all...you would simply be

donating a small percentage of your purchase to me, and I would definitely appreciate it!

Thanks,
-Chris Sherwood

My Voipsupply.com banner link:

**LARGEST SELECTION OF
DIGIUM PRODUCTS...**

