CUSTOM VoIP SYSTEM USING ASTERISK FRAMEWORK

DEVELOPER DOCUMENTATION

OVERVIEW

The project is being built as a substitute of the PBX system and IP PBX system we use at societies, hospitals, hotels or offices. We are trying to build a system that can provide the benefits of the earlier PBX systems in the user mobile phone. People connected in the same Wi-Fi network will be able to make voice and video calls, send messages, make emergency calls, send mails to the other users who are connected to the network and in a much more organised way than that of a IP PBX or PBX system. We would briefly discuss the benefits here:

- Much easier to install & configure than a proprietary phone system or an IP phone system.
- Easier to manage because of web/GUI based configuration interface.
- Eliminate phone wiring.
- Scalable.
- Better phone usability.
- Many features at one single application.

REQUIREMENTS:

- A desktop, which is configured as per the average number of calls expected.
- A Wi-Fi setup.
- User need to install an app called Zoiper available in google play store or windows store.
- FreePBX distro.

FUNCTIONALITY:

- Users can make voice and video calls within the network to other users.
- Users are able to send messages.
- Users can send voice messages.

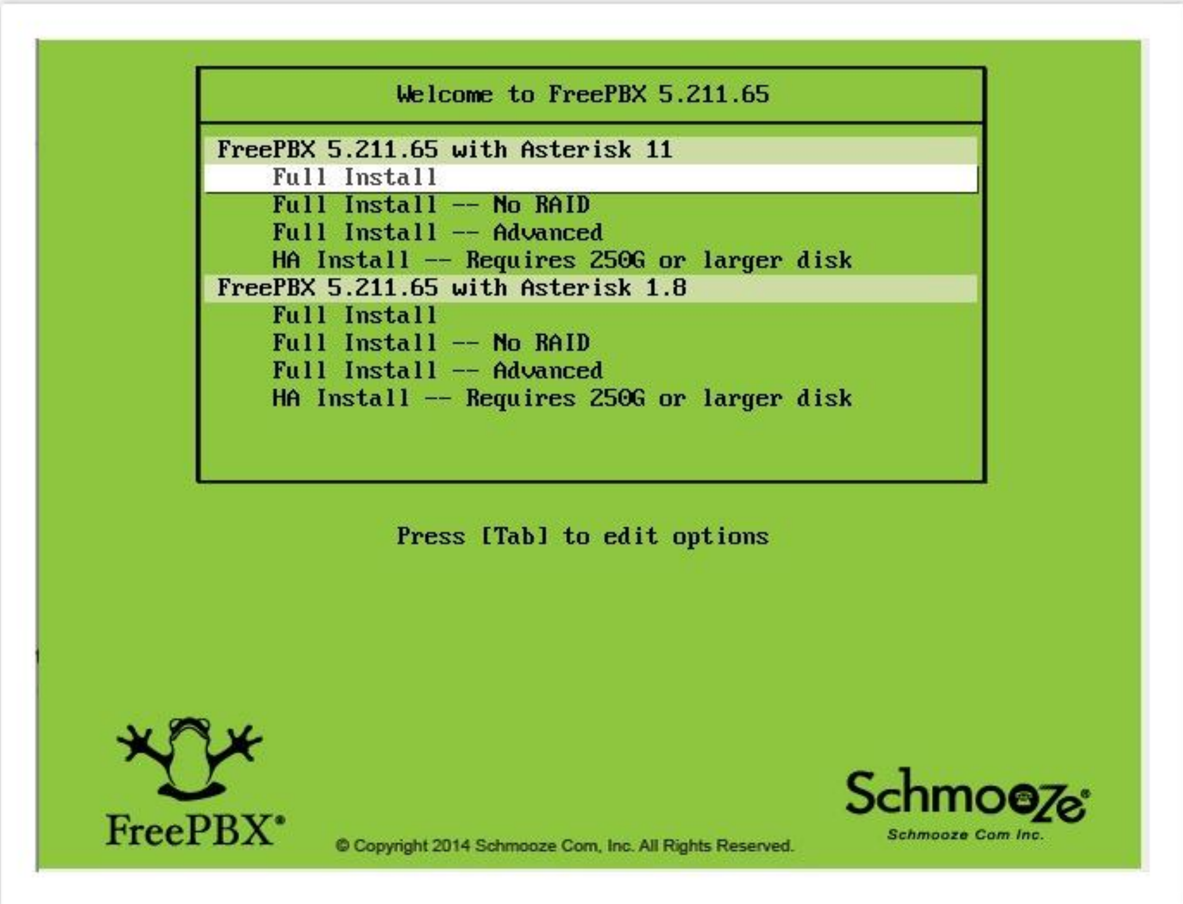
PLATFORM OF DEVELOPMENT:

We are using Asterisk which is open source framework for building communication applications, but instead of using its direct command line version, we are using a derived product called FreePBX which is a web-based open source graphical user interface (GUI) that manages Asterisk, a voice over IP and telephone server. FreePBX is licensed under the GNU General Public License version 3. It is a component of the FreePBX Distro, which is an independently maintained Linux system derived from the source code of the CentOS distribution, having Asterisk pre-installed. FreePBX provides a nice user interface where one can configure the system quite easily and some feature which the developer wants and is not present can be implemented by them by using the core asterisk framework and by writing some scripts.

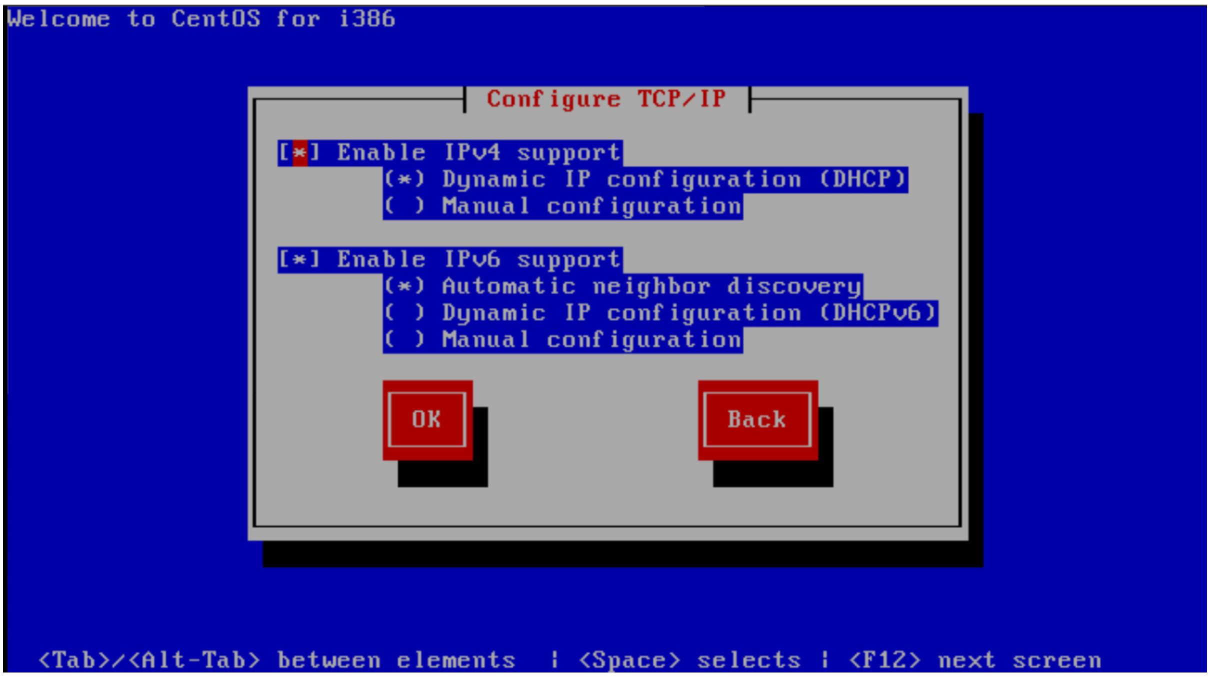
FREEPBX SETUP STEPS:

1. First we need the FreePBX distro, which is the easiest way to install FreePBX. It can be downloaded from http://www.freepbx.org/downloads.

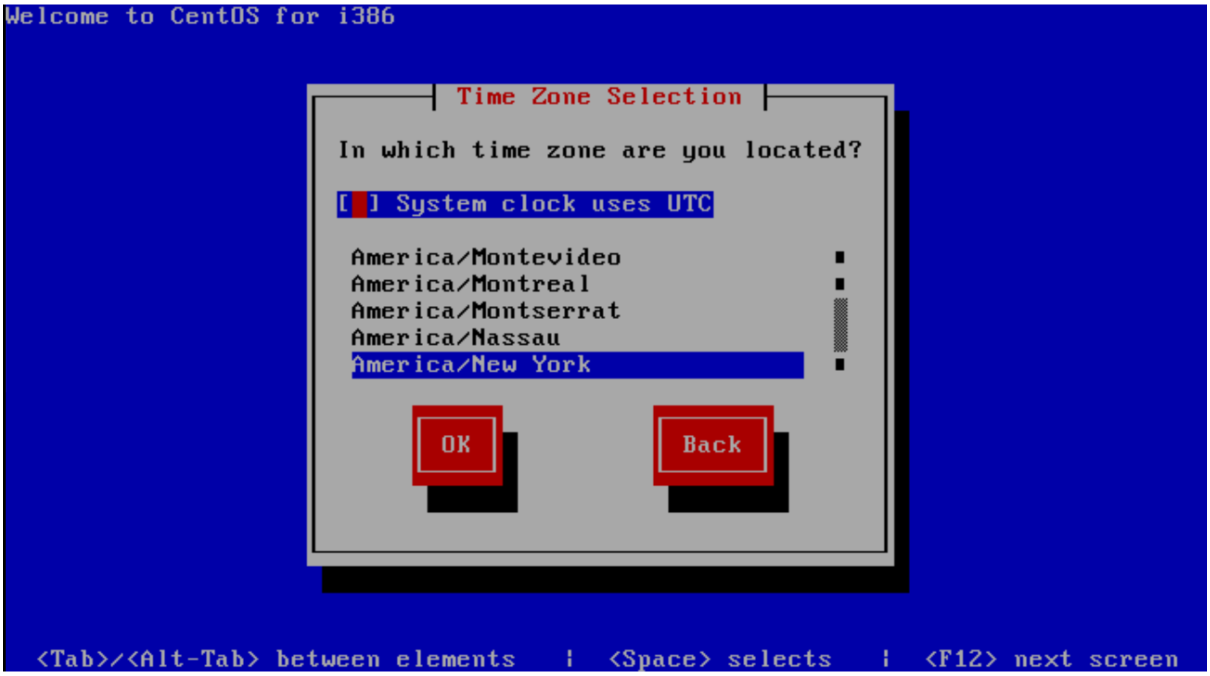
2. The installer will begin with a prompt to select the Asterisk Version you wish to install. We will go with the generic “Full Install” option for Asterisk 11.



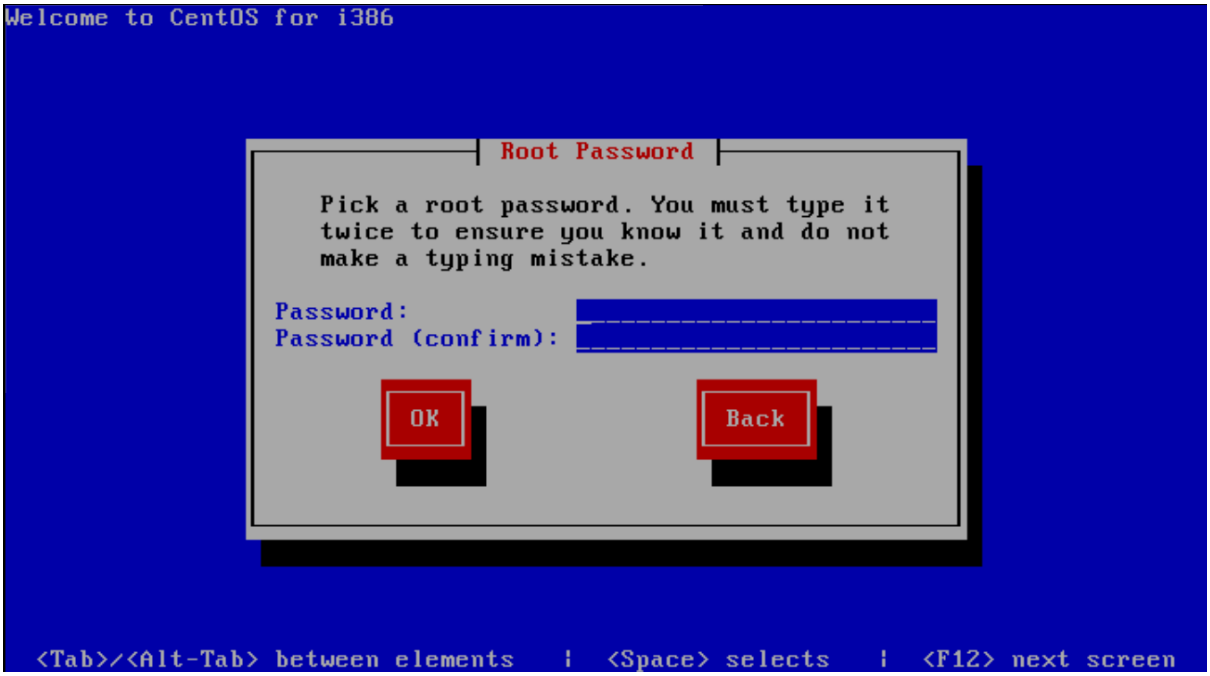
3. After the system boots, we will see the options to configure the network. Default option works fine for us.



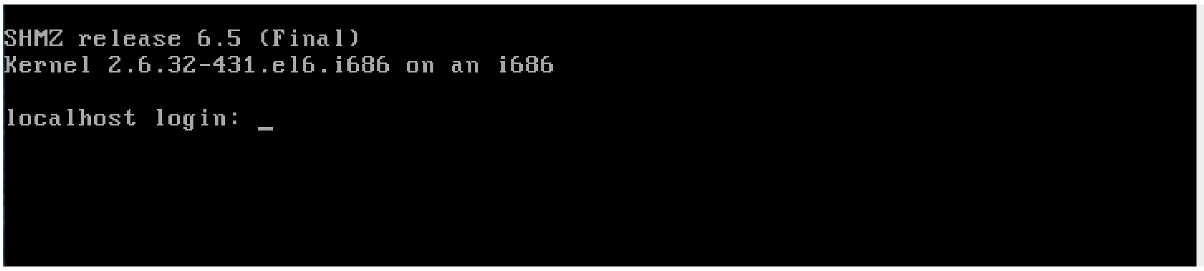
4. After that in time zone setting we will choose ASIA/KOLKATA



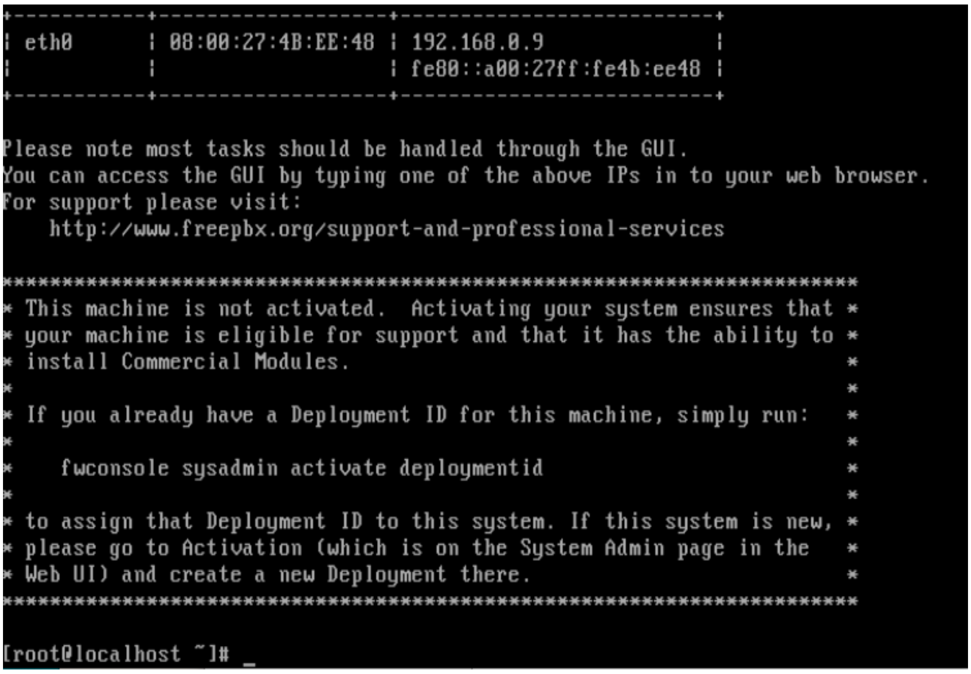
5. Then, we have to set the root password for the distro being installed



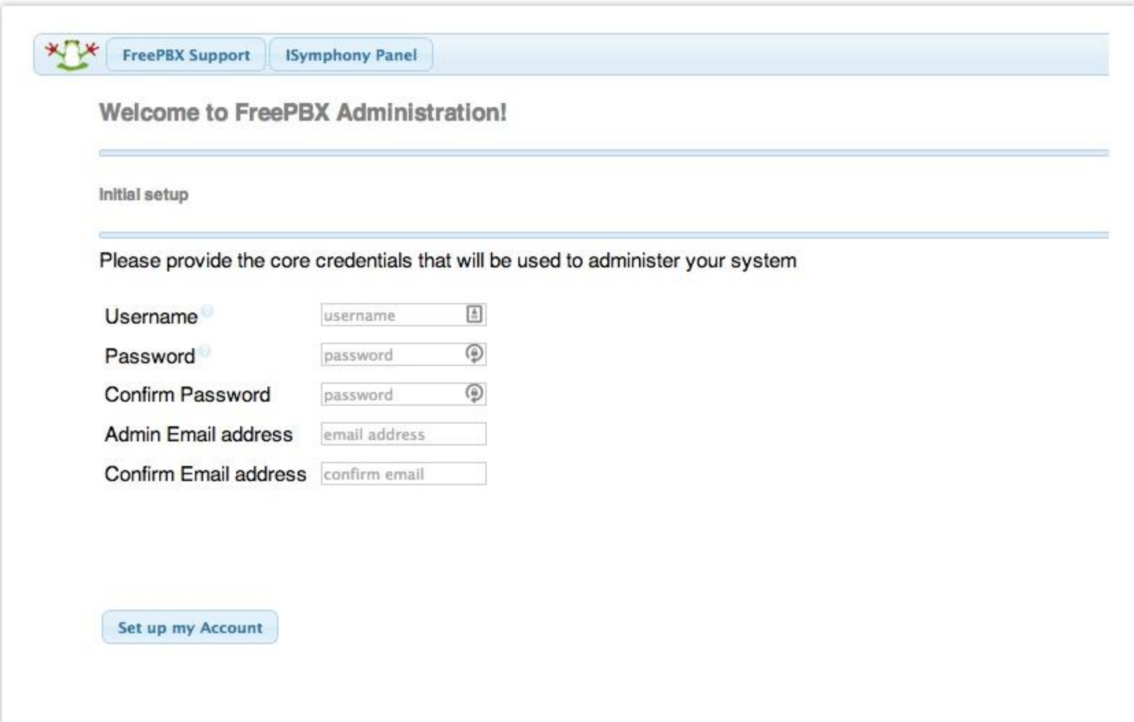
6. Now after the installation completes, we can login to the shell using the root password we set during the installation process.



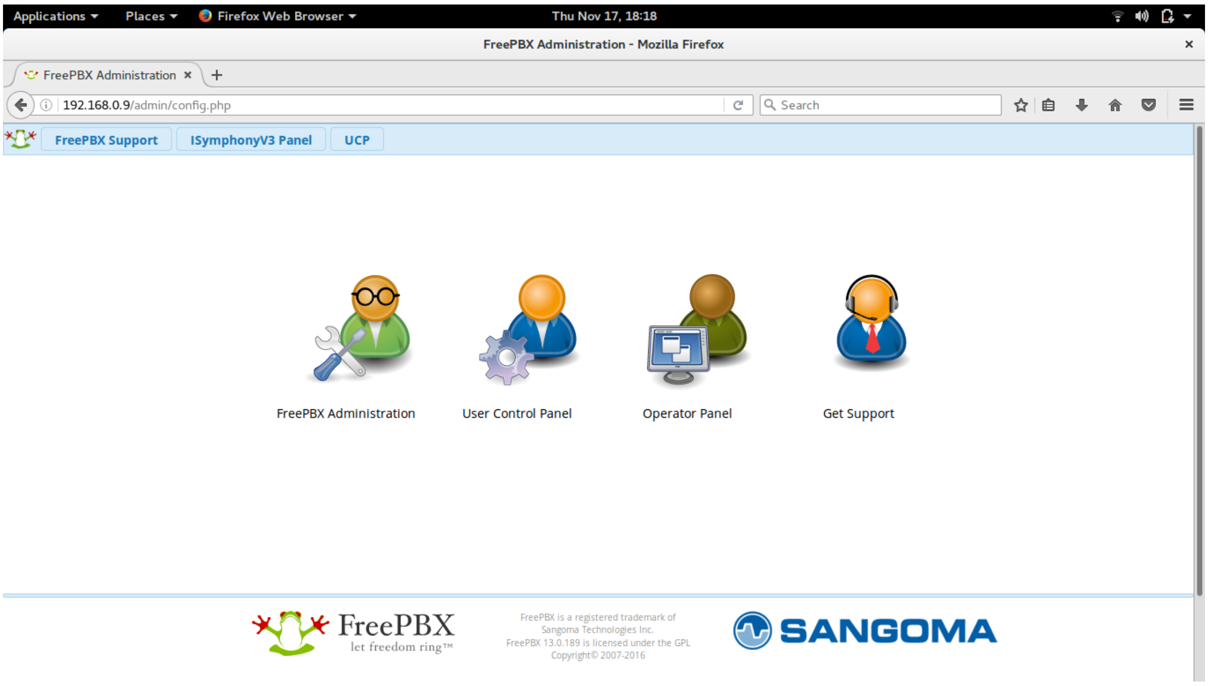
7. After we login, the IP address of the server is shown as below which should be noted down.



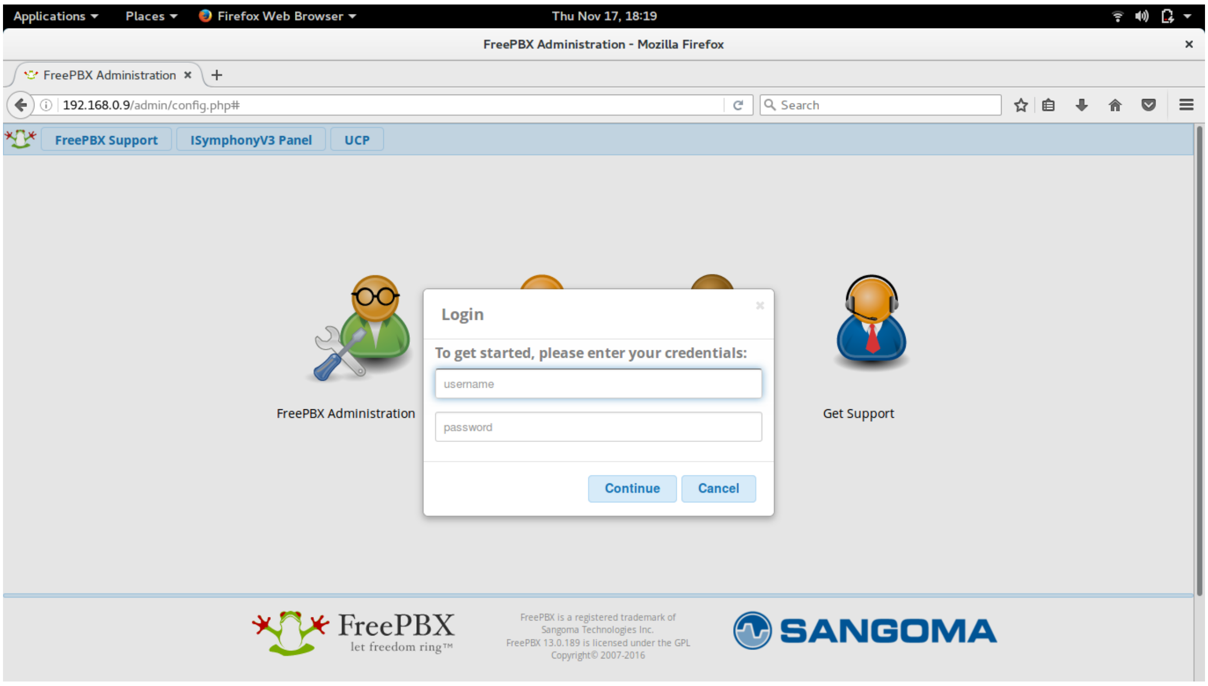
8. To access FreePBX GUI, we will open the IP address from the browser of another computer connected in the same network. When we open the GUI first time, we have to setup an admin account for FreePBX.



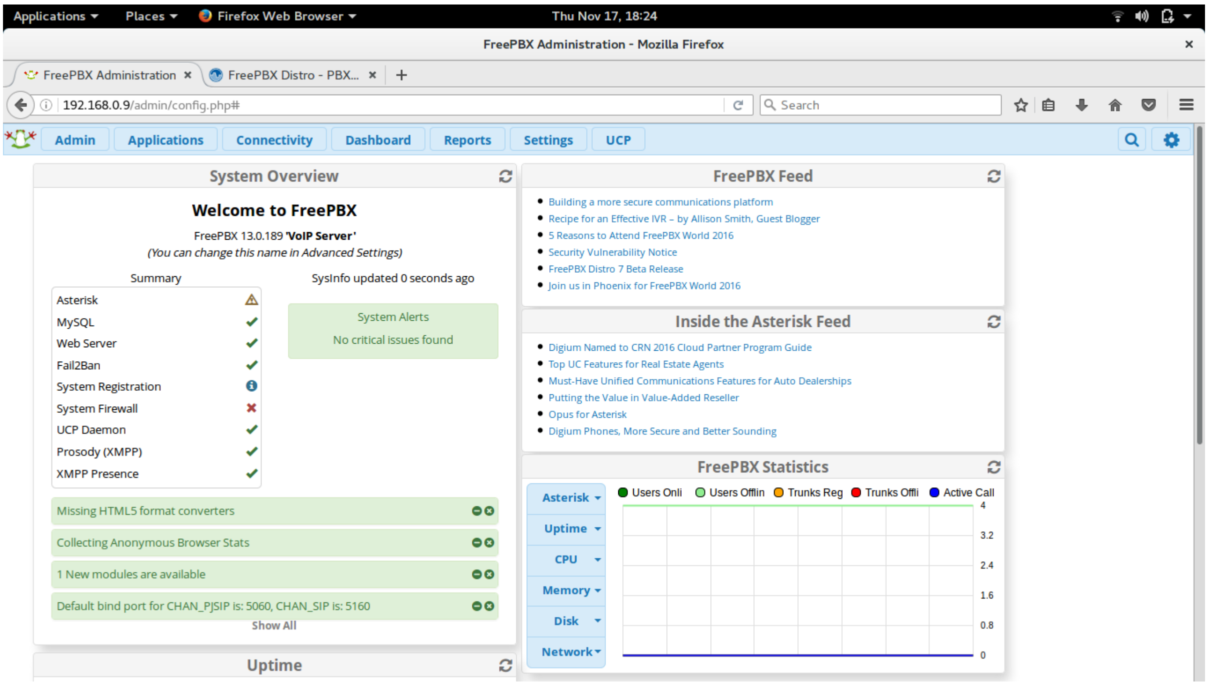
9. FreePBX GUI will offer us following four options



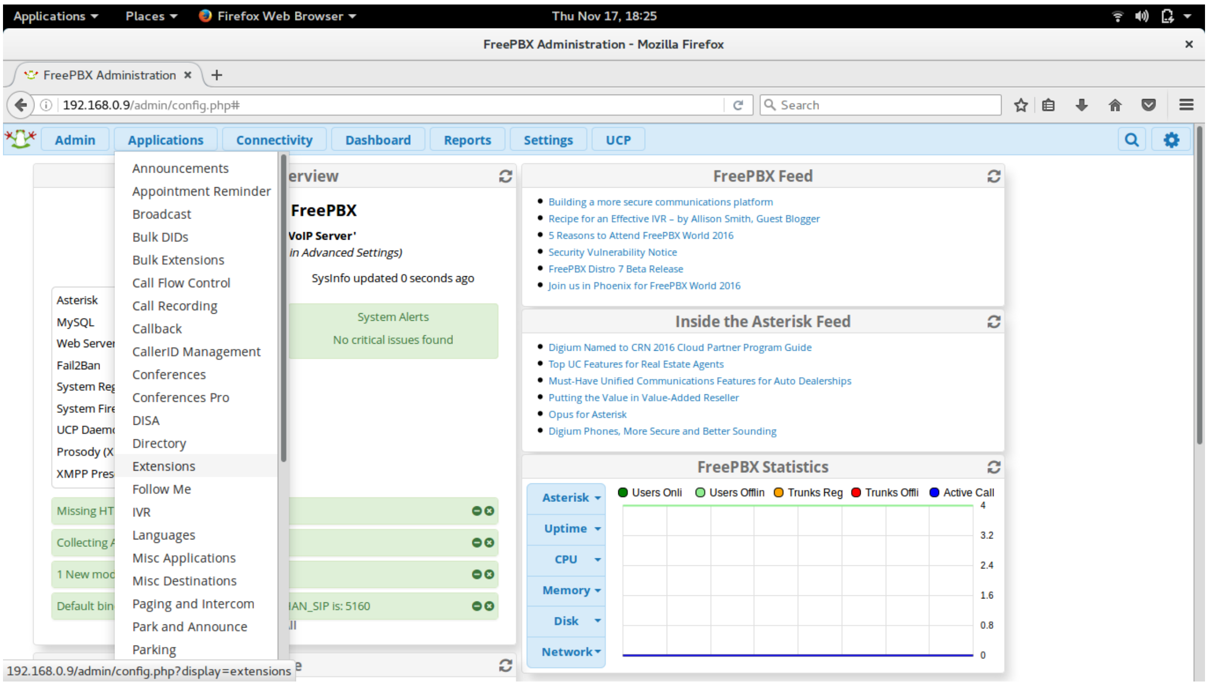
10. For Further configuration we need to login to FreePBX Administration



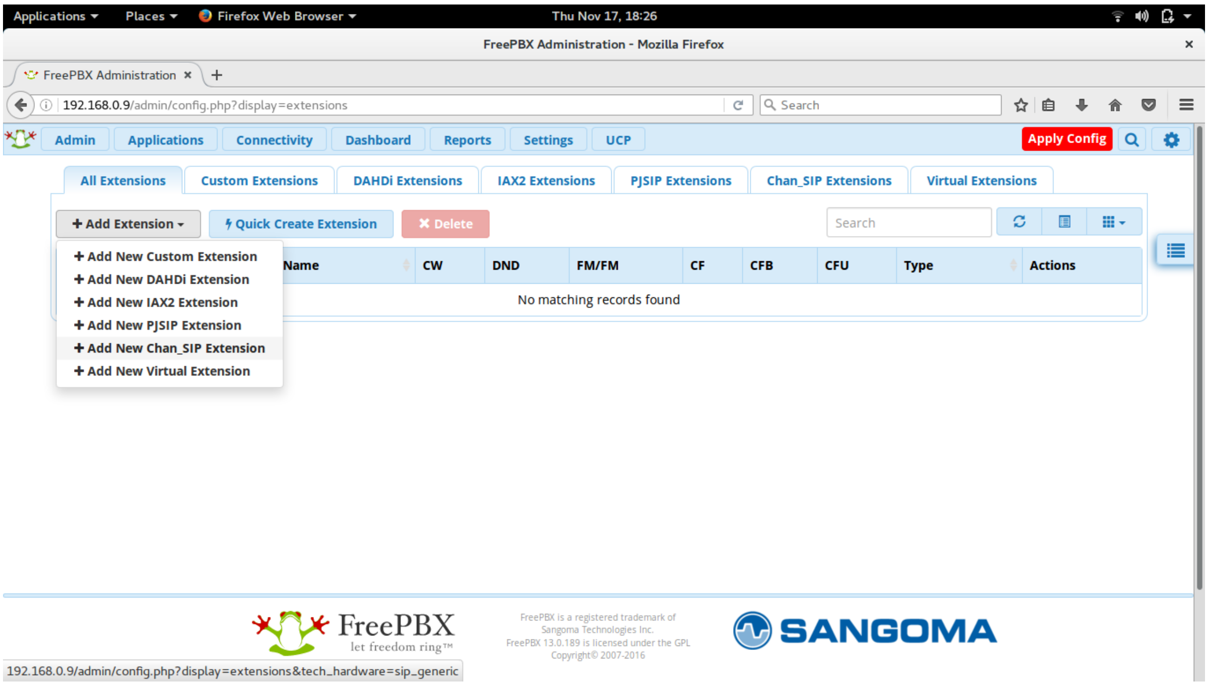
11. Following screen is the dashboard of FreePBX Administration where we can access all the options.



12. To connect phone with this FreePBX server we need to create an extension for each device. To create an extension, we need to navigate to Application>Extensions.

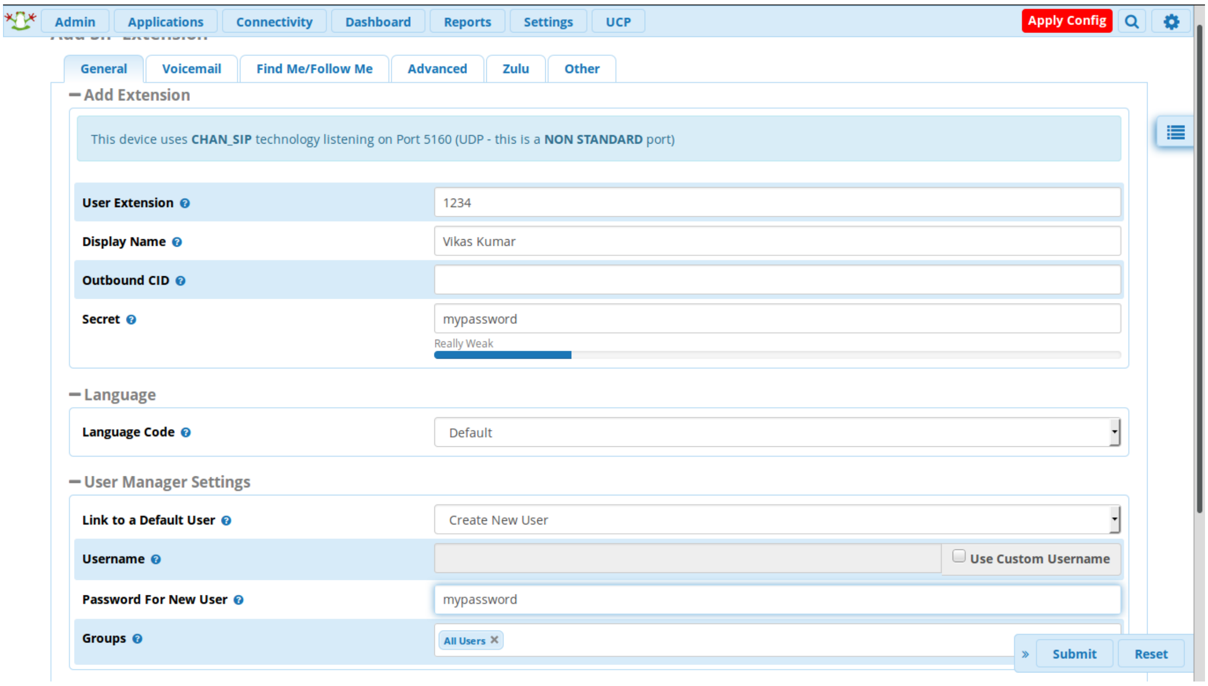


13. In the Extensions Page, we will choose **Add New Chan_SIP Extention** from Add Extension menu



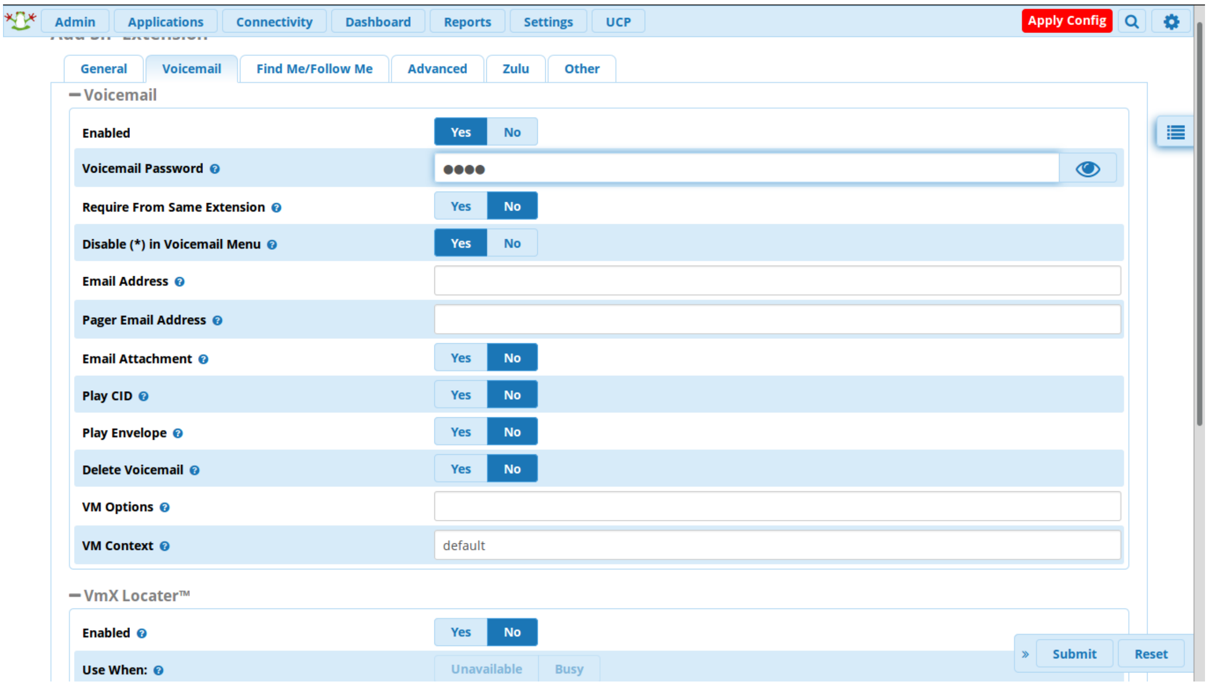
14. To, complete the Extension creation process we have to fill up the following form as follows.
<u>General Tab</u>
User Extension: It is the number on which any user of the system will dial to contact this user.
Display Name: Display Name of the user
Secret: Password for the Extension
Password for New User: FreePBX also creates a user account associated with the extension. We need to set the password for that new user too. We preferably use

the same password as secret for the ease of configuration.



Voicemail Tab

VoiceMail Password: If we want to use voicemail service, then we need to set up a PIN to access voice mailbox.

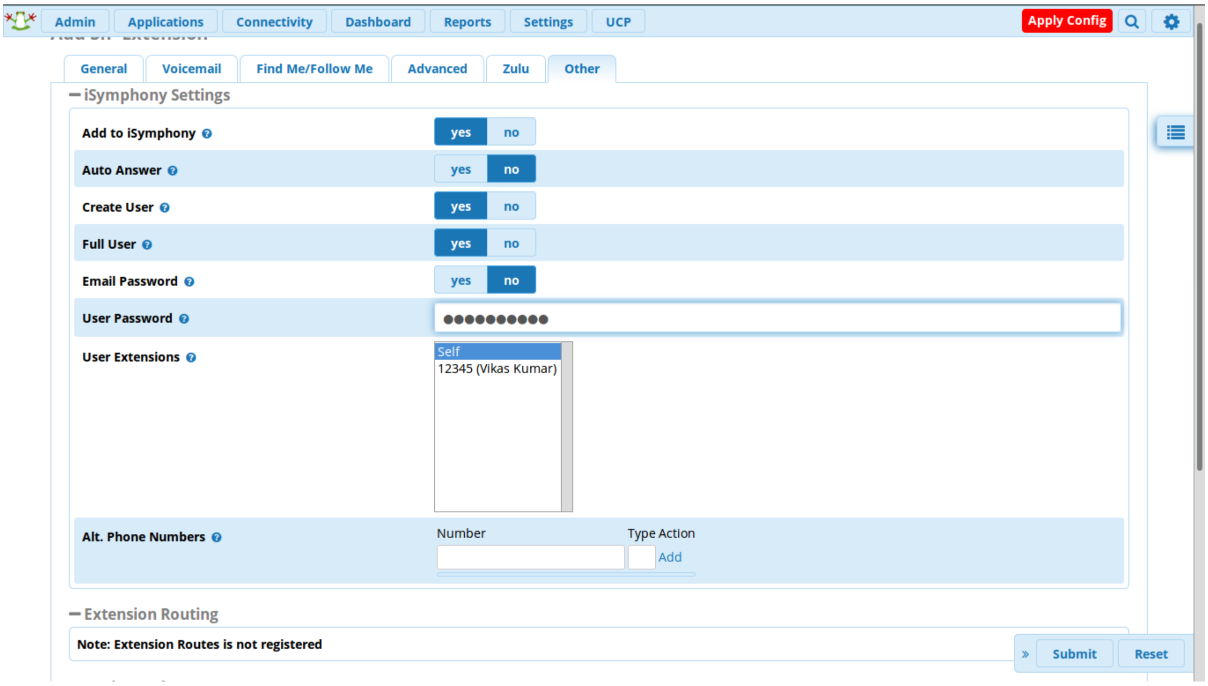


Advance Tab

Allow: We set it to blank so all devices in network can connect to it
Deny: It is also set to blank for the same reason

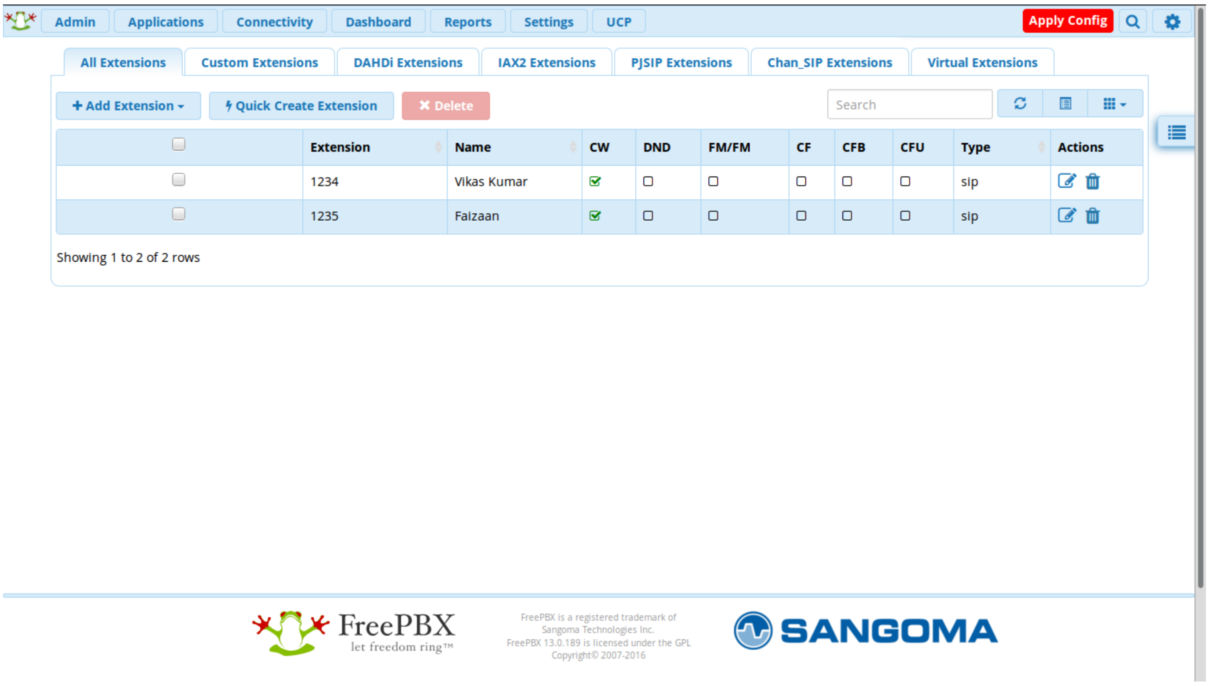
Other Tab

User Password: Set a user password for iSymphony account. It is a call manager service associated with FreePBX distro,

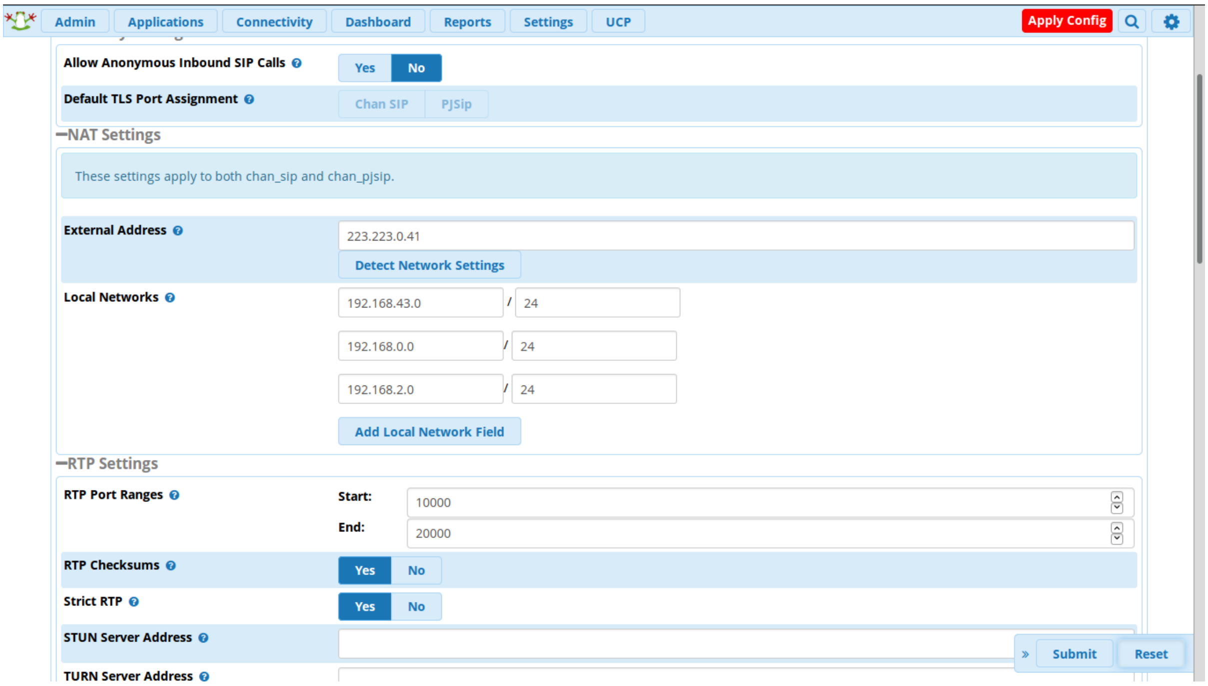


All other settings are optional for current goal.
We can click submit to create the extension. Also we can add another extension in

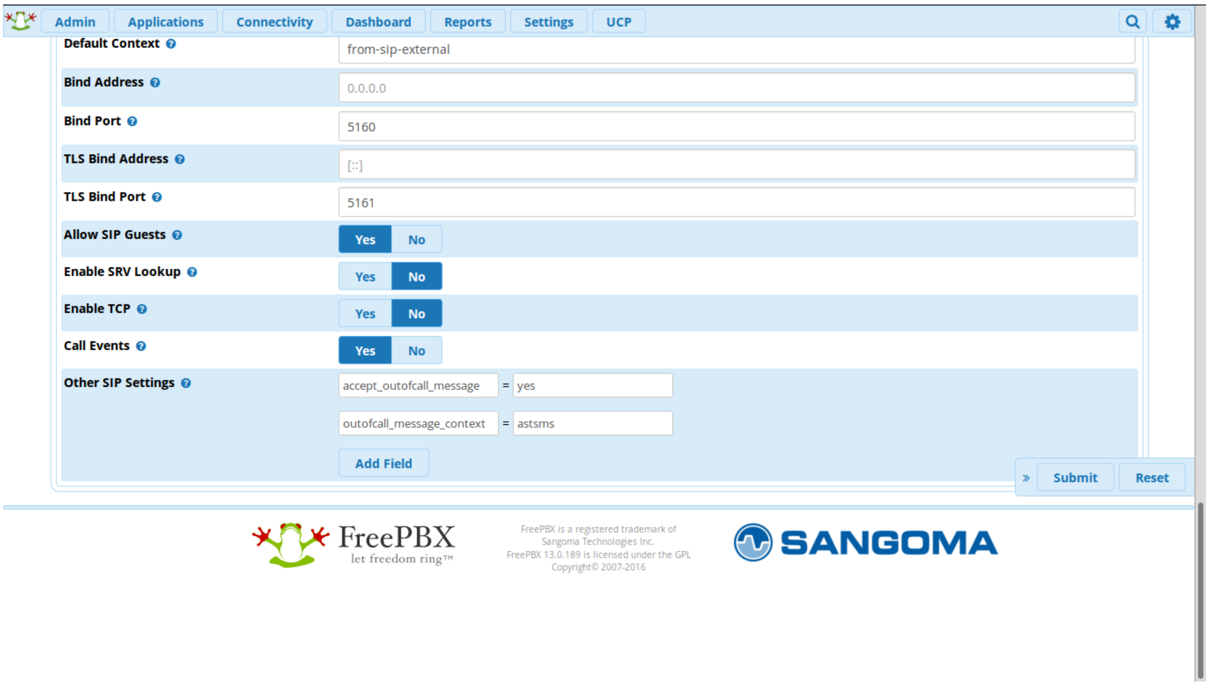
the same way but with different Extension number.



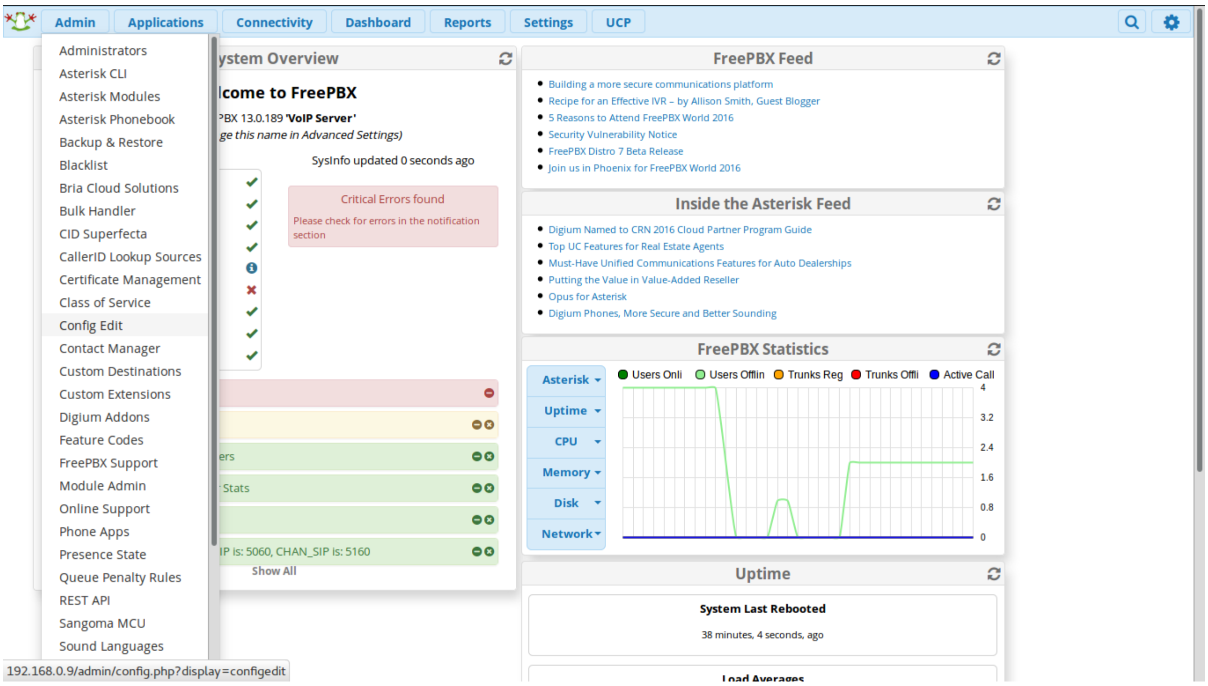
15. There could be some network error, if we move the server from on network to another. So to solve the problem, We need to check if the **local network** is configured properly in the **SIP Settings.**



16. Till now the server will work fine for voice calling, but text messages couldn't be sent. So, to solve the problem we need to create a custom dialplan. To do this, first we need to add following two lines inside "Other SIP Setting".
accept_outofcall_message=yes
outofcall_message_context=astsms

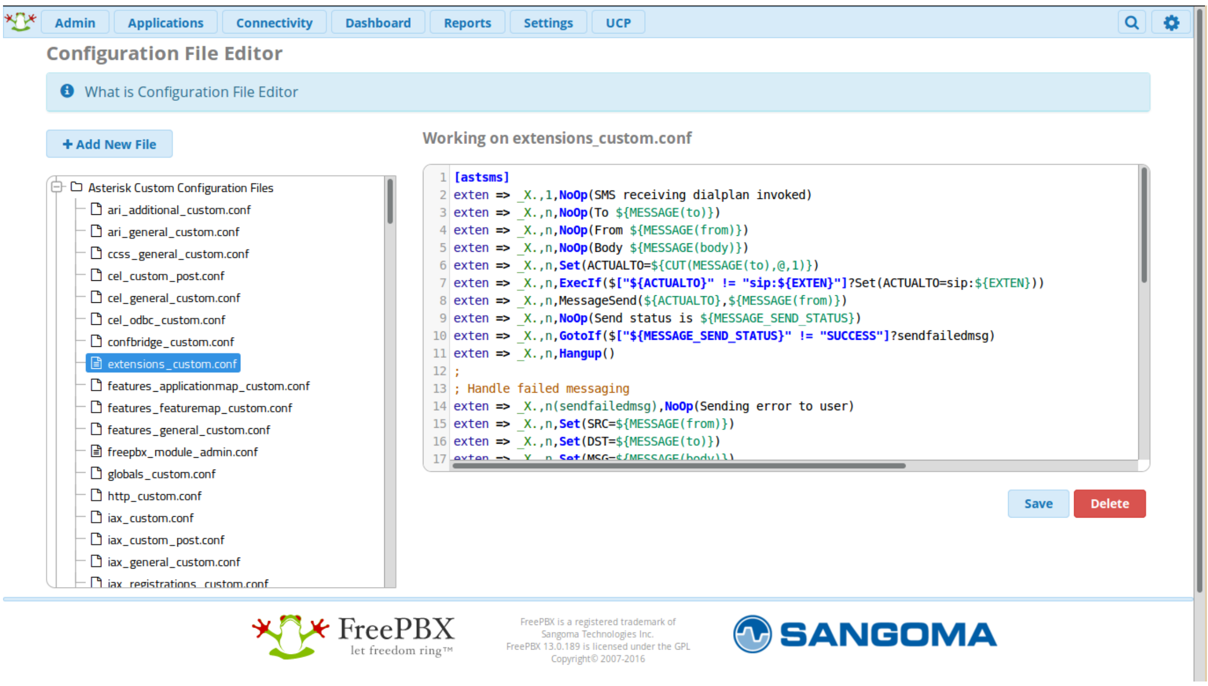


17. In file "extensions_custom.conf" , we will create a dialplan inside context **astsms.** This file can be accessed through **Config Edit** option in **Admin** menu of the dashboard.

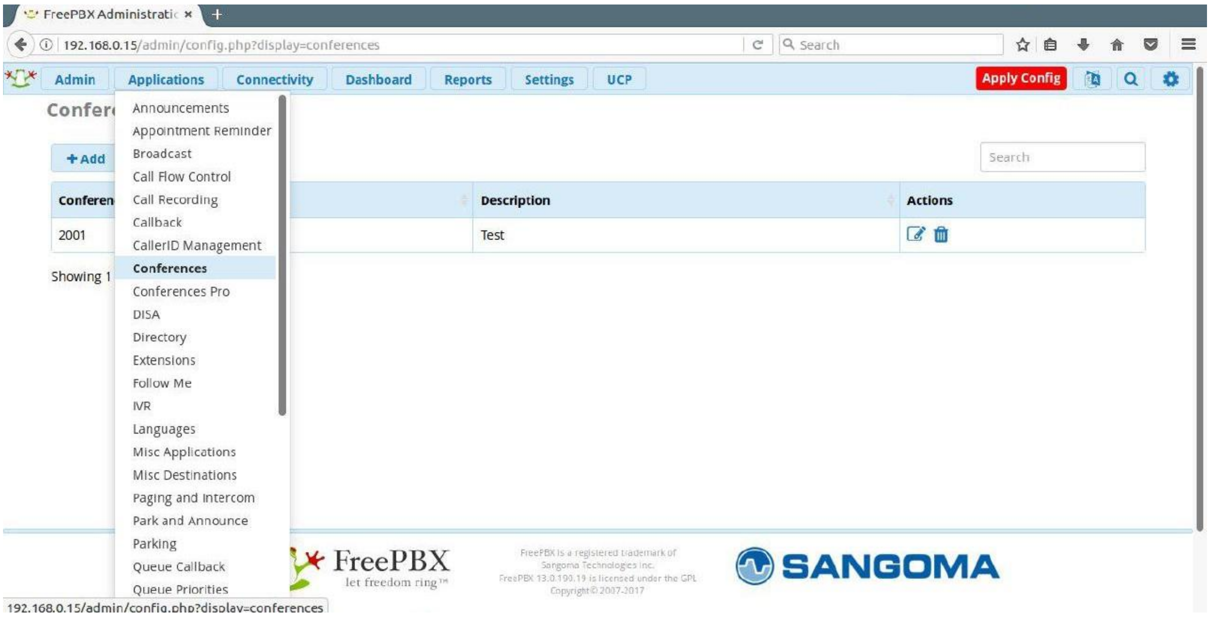


18. To create the dialplan, we will append following text to file "extensions_custom.conf"

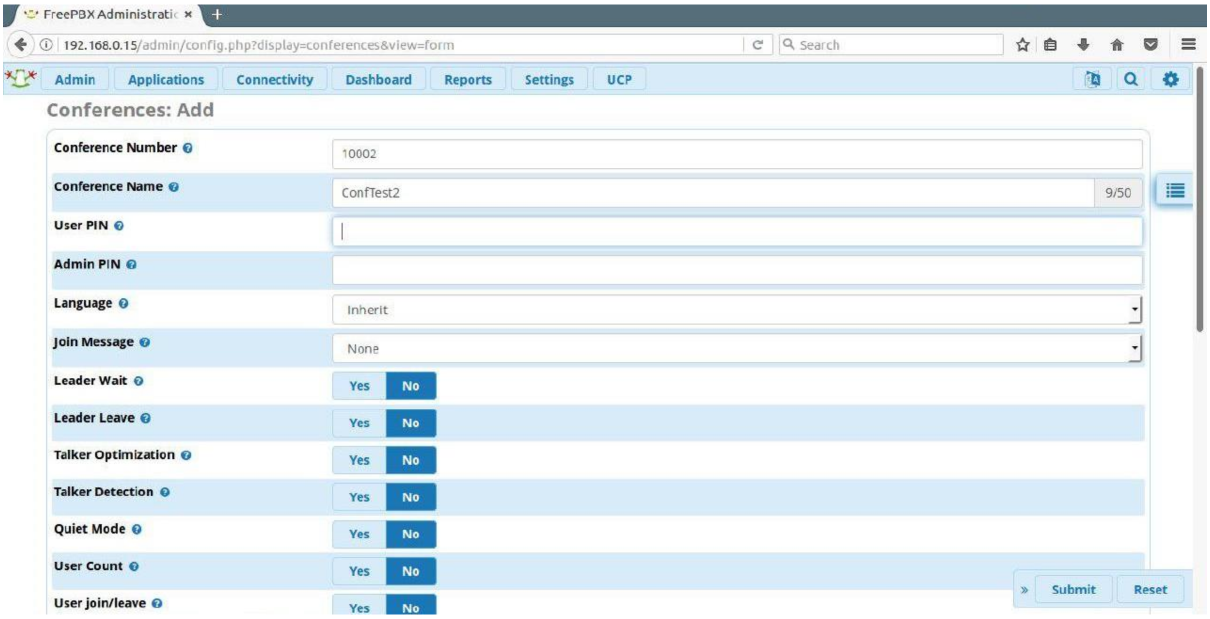
```
[astsms]
exten => _.,1,NoOp(SMS receiving dialplan invoked)
exten => _.,n,NoOp(To ${MESSAGE(to)})
exten => _.,n,NoOp(From ${MESSAGE(from)})
exten => _.,n,NoOp(Body ${MESSAGE(body)})
exten => _.,n,Set(ACTUALTO=${CUT(MESSAGE(to),@,1)})
exten => _.,n,MessageSend(${ACTUALTO},${MESSAGE(from)})
exten => _.,n,NoOp(Send status is ${MESSAGE_SEND_STATUS})
exten => _.,n,GotoIf($["${MESSAGE_SEND_STATUS}" != "SUCCESS"]?sendfailedmsg)
exten => _.,n,Hangup()
;
; Handle failed messaging
exten => _.,n(sendfailedmsg),Set(MESSAGE(body)="[${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)}] Your message to ${EXTEN} has failed. Retry later.")
exten => _.,n,Set(ME_1=${CUT(MESSAGE(from),<,2)})
exten => _.,n,Set(ACTUALFROM=${CUT(ME_1,@,1)})
exten => _.,n,MessageSend(${ACTUALFROM},ServiceCenter)
exten => _.,n,Hangup()
exten => _.,n,Hangup()
```



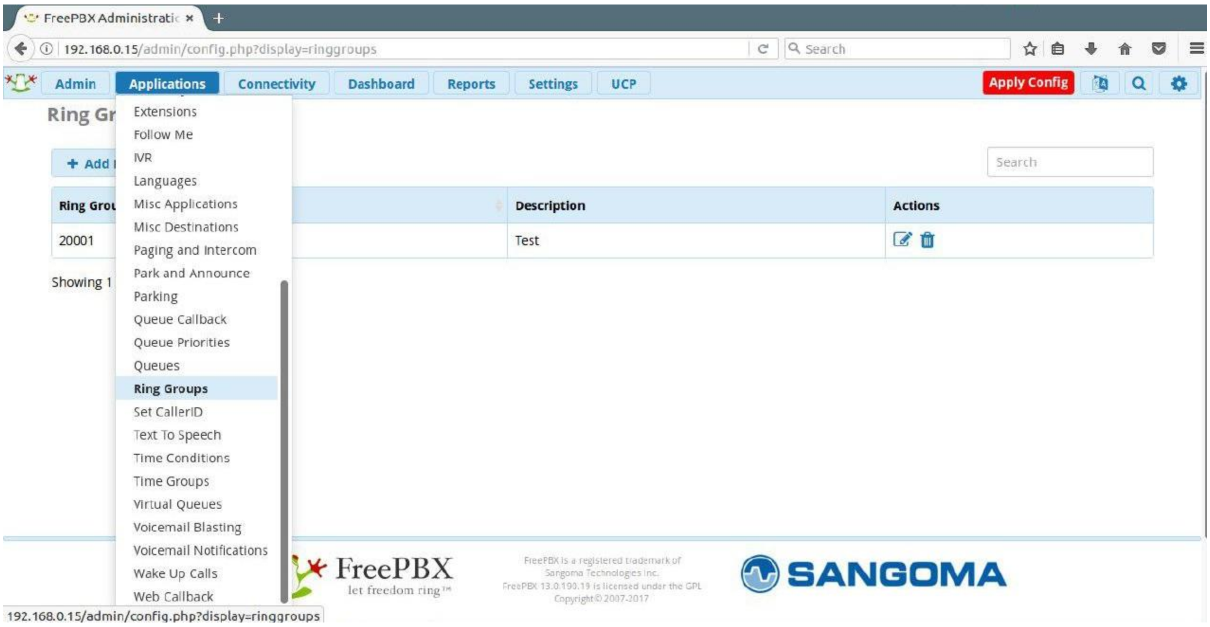
19. To set up conference call feature, go to Application -> Conferences



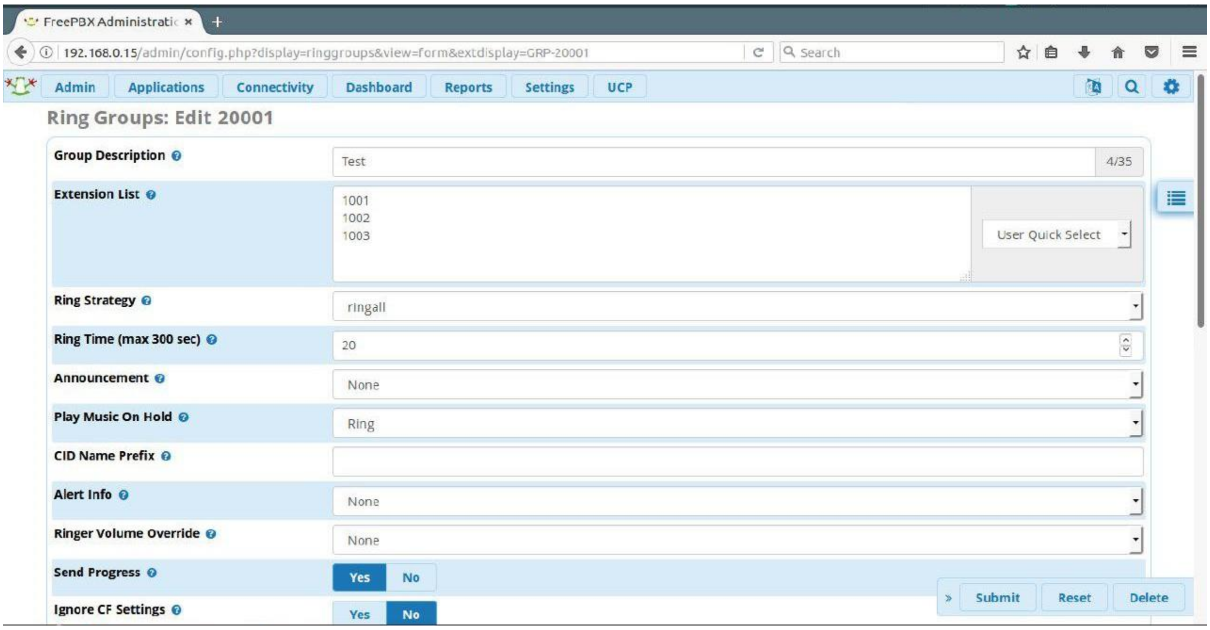
20. We click on Add button to add a conference number, (where a conference number is such that an asterisk user using the system can call in the given number and will be connected to all other users calling in the same number). We have added a conference number here (which should not match with any other number previously assigned to any user). We have added a Conference name and then we click on submit.



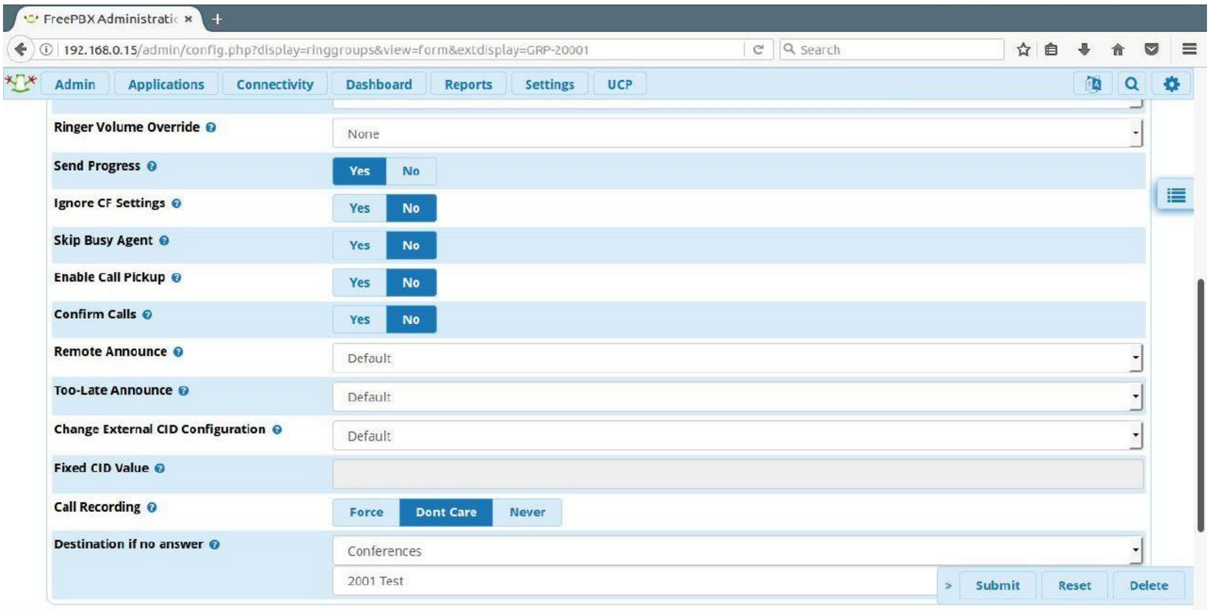
21. Another feature which we are going to implement now is calling a group of numbers at time of Emergency (we call it emergency call, or ring a group), the call is connected to the person who first picks up the phone in the group. For setting up this we go to Appplications -> Ring Groups.



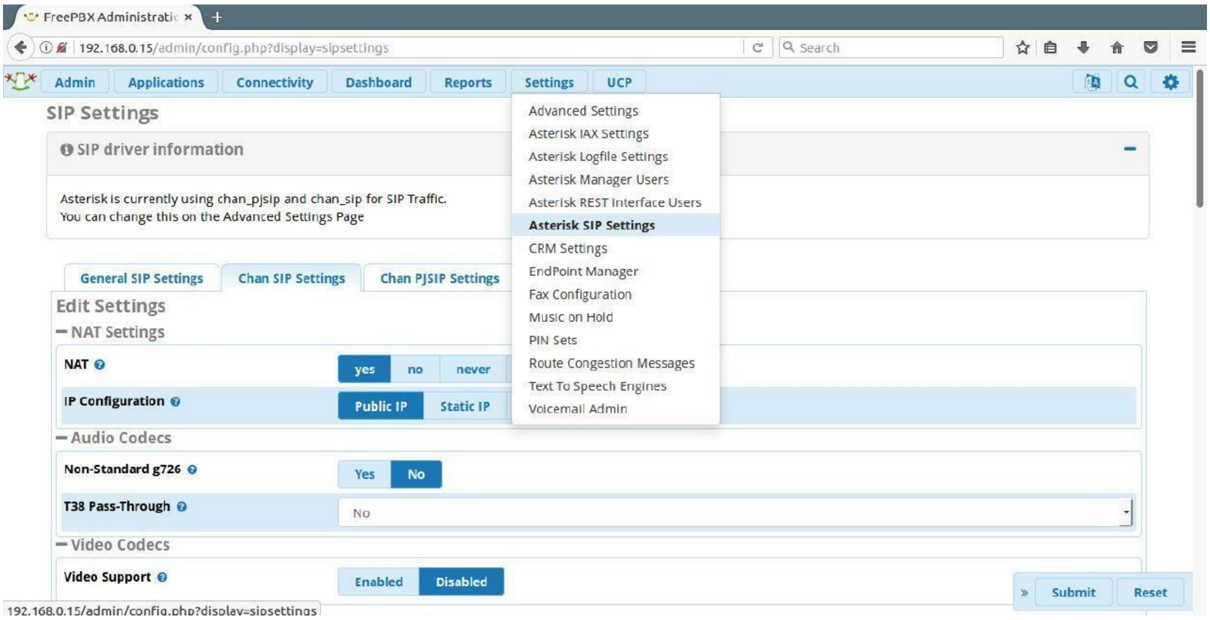
22. We will add a Group Description here, and the Asterisk Numbers of all those users for whom the emergency call feature is meant for will be added in the Extension List option given.



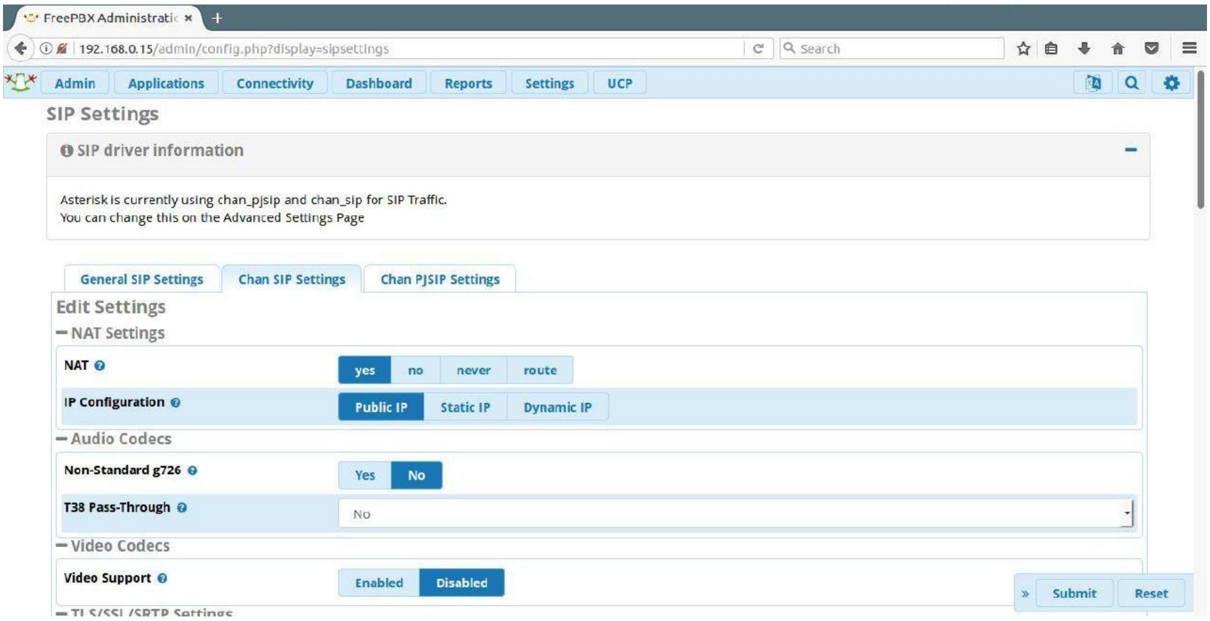
23. We scroll down in the same window and select Conferences in the option " Destination if not answered " .



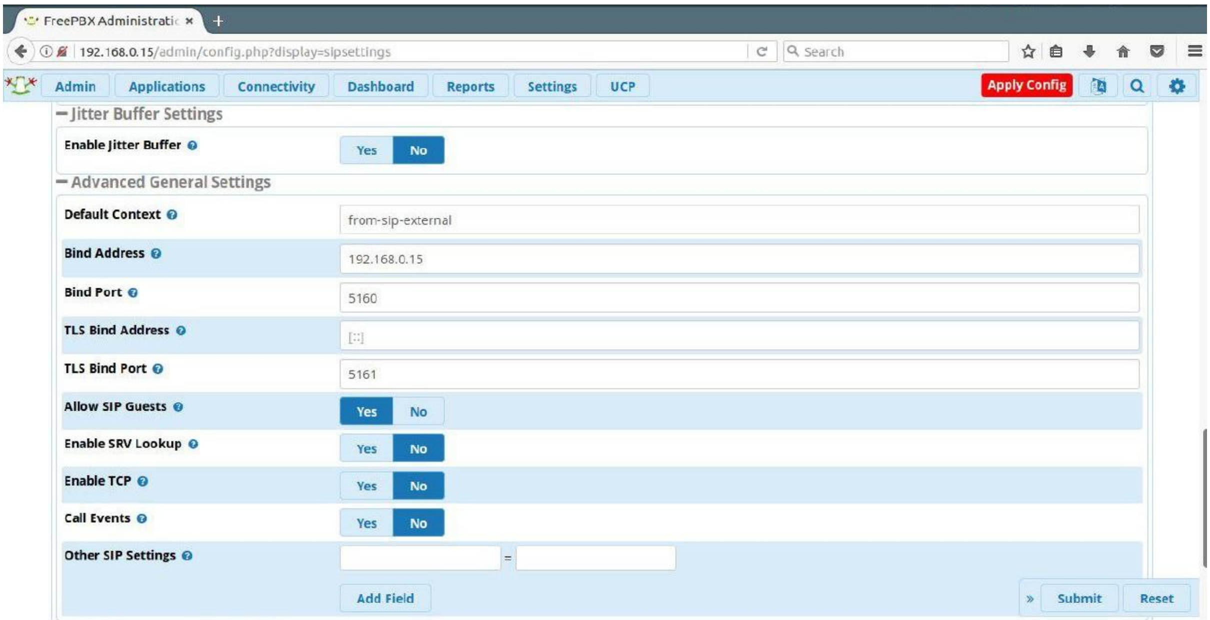
24. Now for setting up the mobile to asterisk connectivity me need to add a Binding IP address. For which we go to Settings -> Asterisk SIP Settings



25. And we select "Chain SIP Settings" tab



26. We add the binding IP address which is the local IP address of the server, one can quickly verify it in the browser url. (For Ex: It's 192.168.0.15 here)

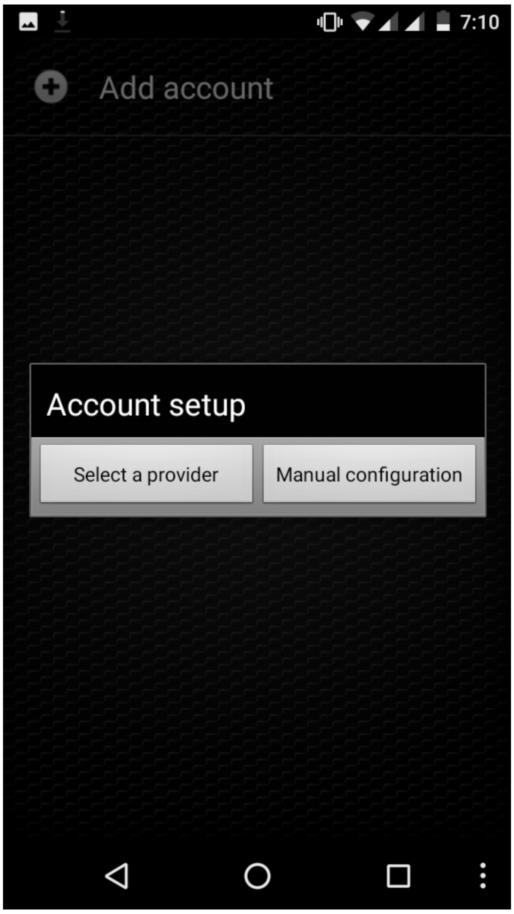


27. For few Operating Systems we need to disable the OS's firewall so that external IP connections can be set up, which is blocked by the firewall at times. For which run the following command in the Asterisk command window

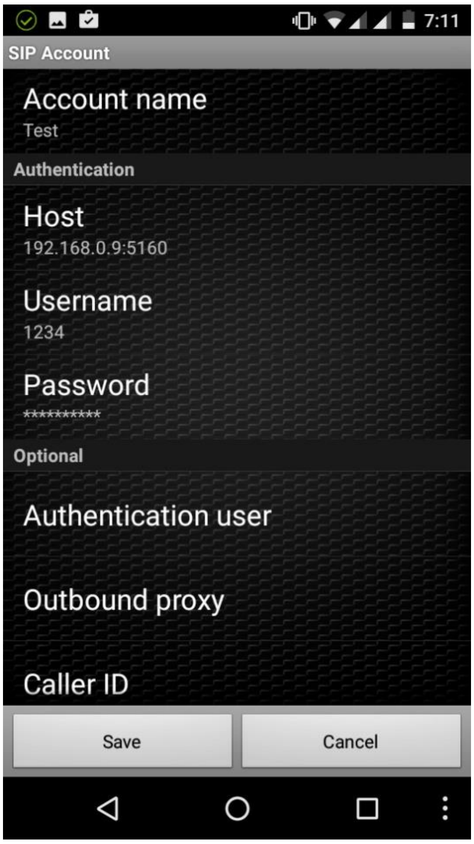
"service iptables stop"

28. Now we will save and restart the serve load the changes in action.

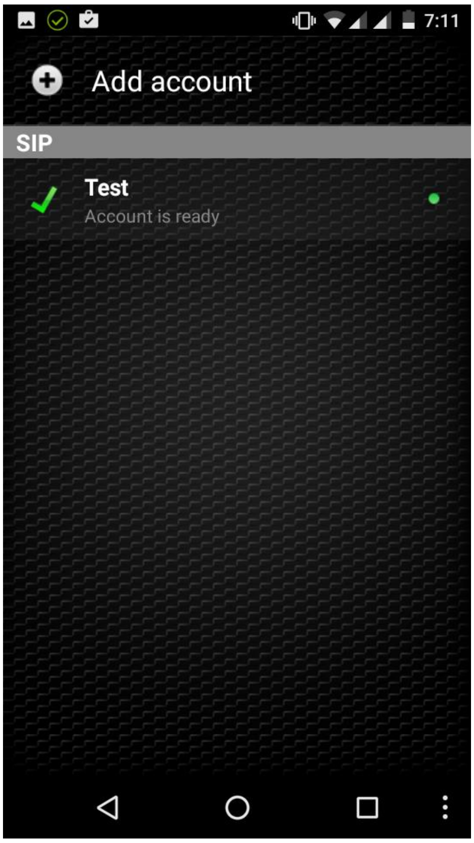
29. In Zoiper application in mobile phone, we will add an account and choose for manual configuration and select to **SIP** option

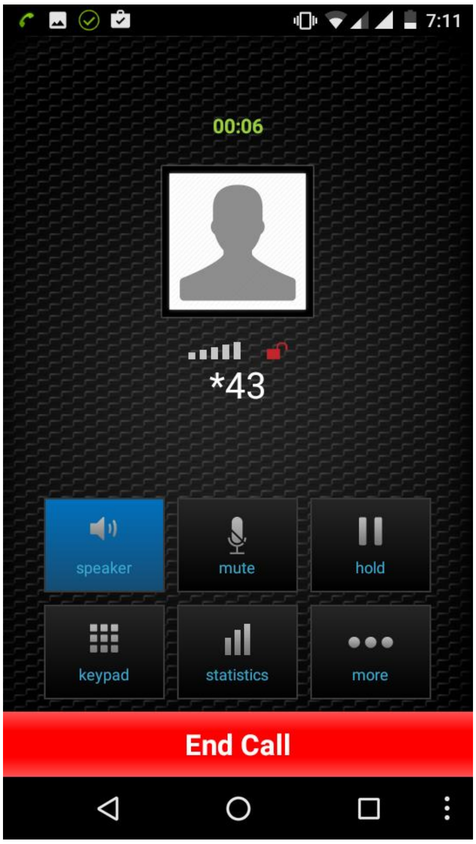


30. Account name can be any text. Host will be the ip address of server with port **5160**. Username will be sip extension number (If default user is used during the Extension creation) and password will be the user password for the extension. That's all the configuration we require.



31. Now we can save the settings. Account will be marked as ready (Green Check). We can echo test server by dialling *43





CONCLUSION:

After the project study and work we came into a conclusion that Asterisk Framework is quite flexible and easy to use when we are handling it through FreePBX. Few configurations and a bit of scripting knowledge is sufficient to handle the entire setup. This can be used by the housing apartments having existing WiFi setups in very low cost. This can be used in hospitals and offices and one doesn't need to go through the problems of setting up an IP Phone infrastructure. Our configuration is secure and robust for calls, video calls, messaging, conference calling, and emergency calling. New features can be added as per the user review and requirements.