Azblink Video/Voice/Text System – Multiple-Host Configuration Guide

Abstract

This document describes how to use multiple-host configuration to deploy Azblink Video/Voice/Text Communication Functions other than using standalone machine. It includes using LDAP for user authentication, the determination of VPN address pool, dialing rules associated with SIP trunks, XMPP cluster, the mechanism of sending FCM/APNS/PushKit, the storage space for "**http file upload**" in XMPP, recording of XMPP text messages, and recording of voice calls.

Table of Contents

Illustration Index

Introduction

Azblink Video/Voice/Text System can be deployed with a standalone host. In this document, we focus on the setting by using multiple hosts to deliver the same functionalities. The settings described in this document are related to multiple-host configuration. We did not get into the details for each specific function. We just bring up the methods for devising a multiple-host system. For details in each specific function, please refer to the **Administration Guide**.

Using multiple-host configuration is to cope with performance issues in hope to distribute the loading into several hosts. However, to make multiple hosts work as a single host requires some more planning and managing efforts.

In Azblink system, Voice and Video calls are implemented by using **SIP** along with **RTP** streams whereas Text messaging system is implemented using XMPP. To go across the hosts to place voice/video calls from **host A** to **host B**, we set up **SIP trunk** between **host A** and **host B**. For **XMPP** to send message from one host to another, there are two possible ways:

1. if **host A** and **host B** are **with different domain names**, it is necessary to establish server-to-server communication between the two hosts;

2. if **host A** and **host B** are **with the same domain name**, the two hosts should bind into a cluster in order to send messages between hosts; the host-to-host communication is counting on the internal mechanism of the cluster.

For voice or video calls using SIP, the user accounts can be created separately in each host; a host does not need to have the knowledge of the SIP accounts established on the other hosts. The behavior of hosts using XMPP with different domain names is quite similar that user accounts can be created separately on each host without the knowledge of the accounts in other hosts. However, XMPP cluster works quite differently that **each host needs to have the knowledge of all accounts in the cluster**.

Therefore, we have LDAP server to have all the XMPP accounts for the cluster scenario to avoid creating the same accounts on each host. LDAP is an industry standard protocol so that there exist many vendors to provide the tools to manage the directories established in LDAP server. Those tools are known as “**LDAP browser or editor**”. In this document we do not have the introduction of LDAP browser here. If you intend to use your own LDAP server, please remember to fetch the **DB schema of LDAP server** from Azblink.

If XMPP and SIP are intended to be used together, we provide the method to bind the two accounts together in LDAP database. Along with the VPN key, we have the client programs on mobile devices to voice/video/text messaging over VPN to achieve secure multimedia communication.

Within XMPP, it also offers “**file sharing**” between users via “**http file upload and download**”. For “**file sharing**” to work across the hosts, it is necessary to have a common storage space that can be accessed by those servers. For hosts with different domain names, there might be some political issues to hinder the existence of the storage space shared by the those domains if those domains are owned by different organizations. So, we only consider the “**file sharing**” in XMPP under the context of single host or cluster scenario.

If voice calls and text messaging shall be recording for audit purpose, extra storage space is needed. At the moment of writing this document, recording video on server side is not available. We have the client program to record video calls.

While using a client program on mobile devices, the client program might fall into sleep (**preempted by the operation system** of the mobile device) to save the battery power so that it does not have the privilege to standby all the time. The current prevailing operation systems on the mobile devices provide methods to send messages to the mobile devices while the APP falls into sleep. On Android, the mechanism is known as FCM for both sending messages and waking up the APP; on iOS of Apple's iPhones, it has APNS (to send messages) and PushKit (to wake up the APP).

The basic operation principle for FCM/APNS/PushKit is quite similar: the APP obtains a “**device token**” from the cloud maintained by Google or Apple and send this “**device token**” back to the application server that this APP is associated with. When the application server needs to send messages to the mobile device or wake up the APP on the mobile device, it sends the messages along with the “**device token**” to the cloud maintained by Google or Apple; and then, the cloud would dispatch the message to the mobile device if possible.

In Azblink Voice/Video/Text Messaging system, we offer the option to send out FCM/APNS/PushKit along with XMPP message to the callee if there is voice/video call coming to the host where the callee's account shall be registered. In this document, we only introduce the setup on Azblink server.

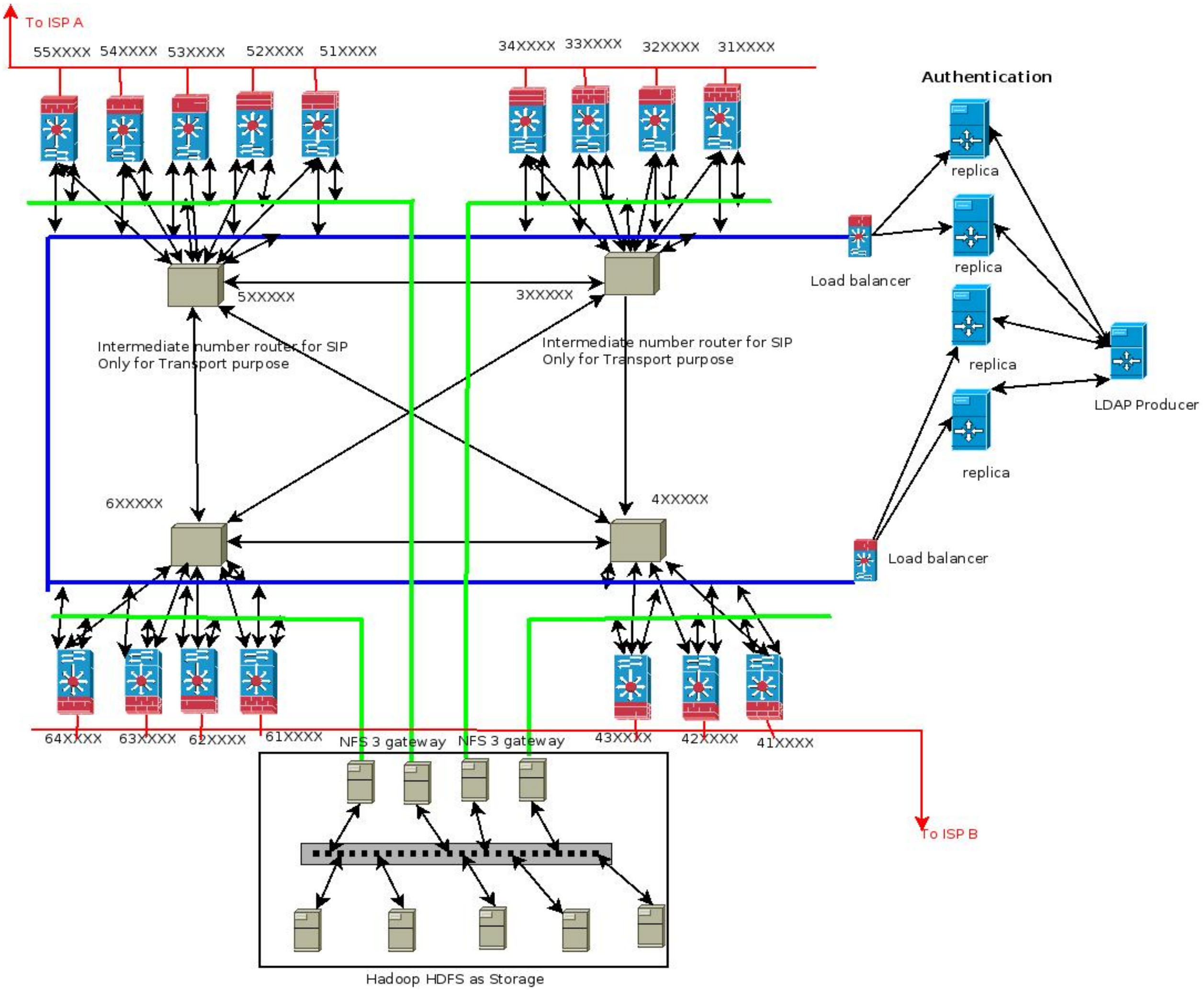


Illustration 1: Deployment Example

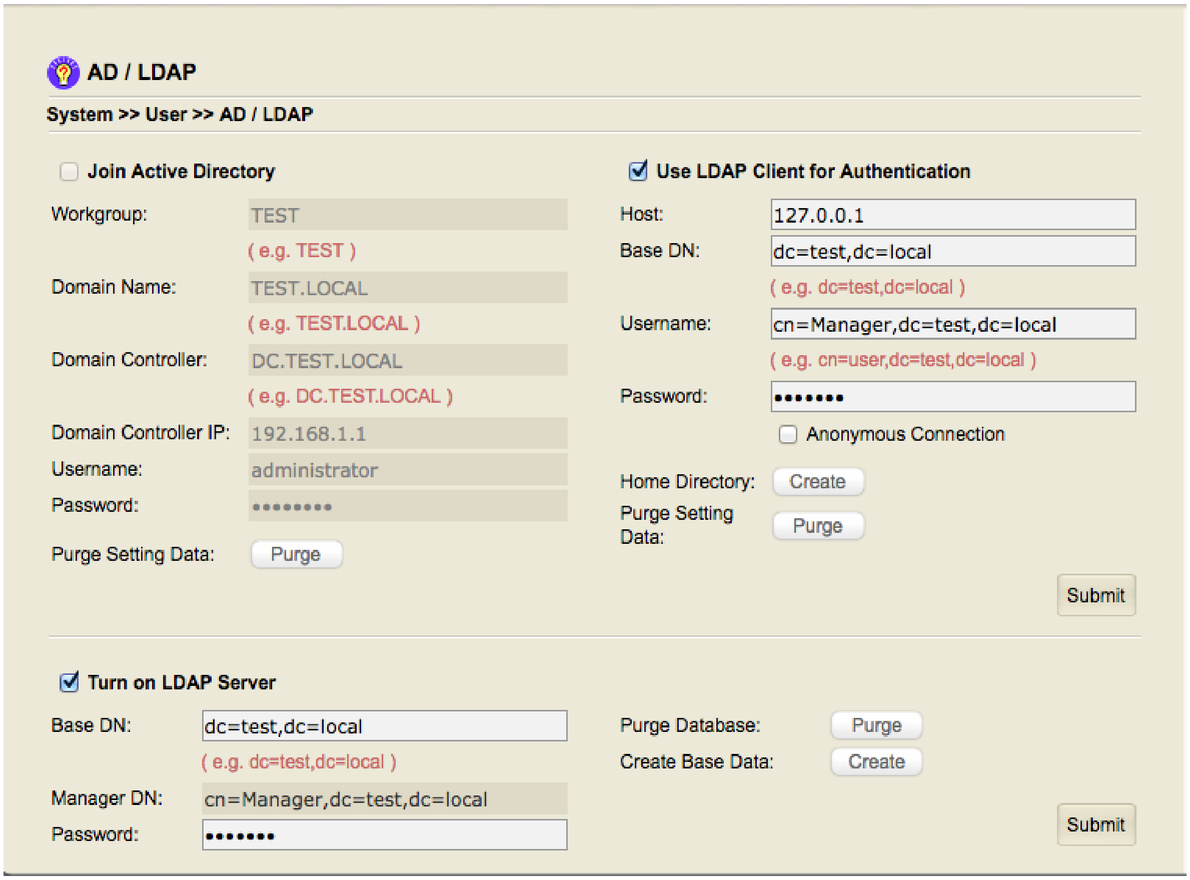
The diagram above is an example how the multiple-host configuration is deployed. In this document, we only briefly introduce the parts associated with "**Application Servers**" (Azblink Servers). For those related to **LDAP server** or **Hadoop HDFS,** you should refer their documents respectively.

LDAP Setting

LDAP configuration in Azblink Voice/Video/Text Messaging System is with two parts:

1. acting as LDAP client to fetch data from LDAP server for user authentication
2. acting as LDAP server

The server part is mainly to demonstrate how we expect the setting of LDAP server for user authentication; it can be used for small scale deployment. Our Web GUI does not provide sophisticated configuration for LDAP server; you need to turn to the documentation of LDAP for help.



AD / LDAP

System >> User >> AD / LDAP

□ Join Active Directory

Workgroup:	TEST (e.g. TEST)
Domain Name:	TEST.LOCAL (e.g. TEST.LOCAL)
Domain Controller:	DC.TEST.LOCAL (e.g. DC.TEST.LOCAL)
Domain Controller IP:	192.168.1.1
Username:	administrator
Password:	••••••••

☑ Use LDAP Client for Authentication

Host:	127.0.0.1
Base DN:	dc=test,dc=local (e.g. dc=test,dc=local)
Username:	cn=Manager,dc=test,dc=local (e.g. cn=user,dc=test,dc=local)
Password:	•••••••

□ Anonymous Connection

☑ Turn on LDAP Server

Base DN:	dc=test,dc=local (e.g. dc=test,dc=local)
Manager DN:	cn=Manager,dc=test,dc=local
Password:	•••••••

Illustration 2: LDAP Setting

The associated setting for LDAP in Azblink Voice/Video/Text Messaging System can be found at "**System >> User >> AD / LDAP**"; it is shown as the screen snapshot above. On the bottom of the screen, the button "**Create Base**

Data” is to create some basic nodes in LDAP server. For example, if “**Base DN**” is “dc=test,dc=local”, then the button will generate the following nodes in LDAP database:

```
dc=test,dc=local
ou=People,dc=test,dc=local
ou=sip,dc=test,dc=local
```

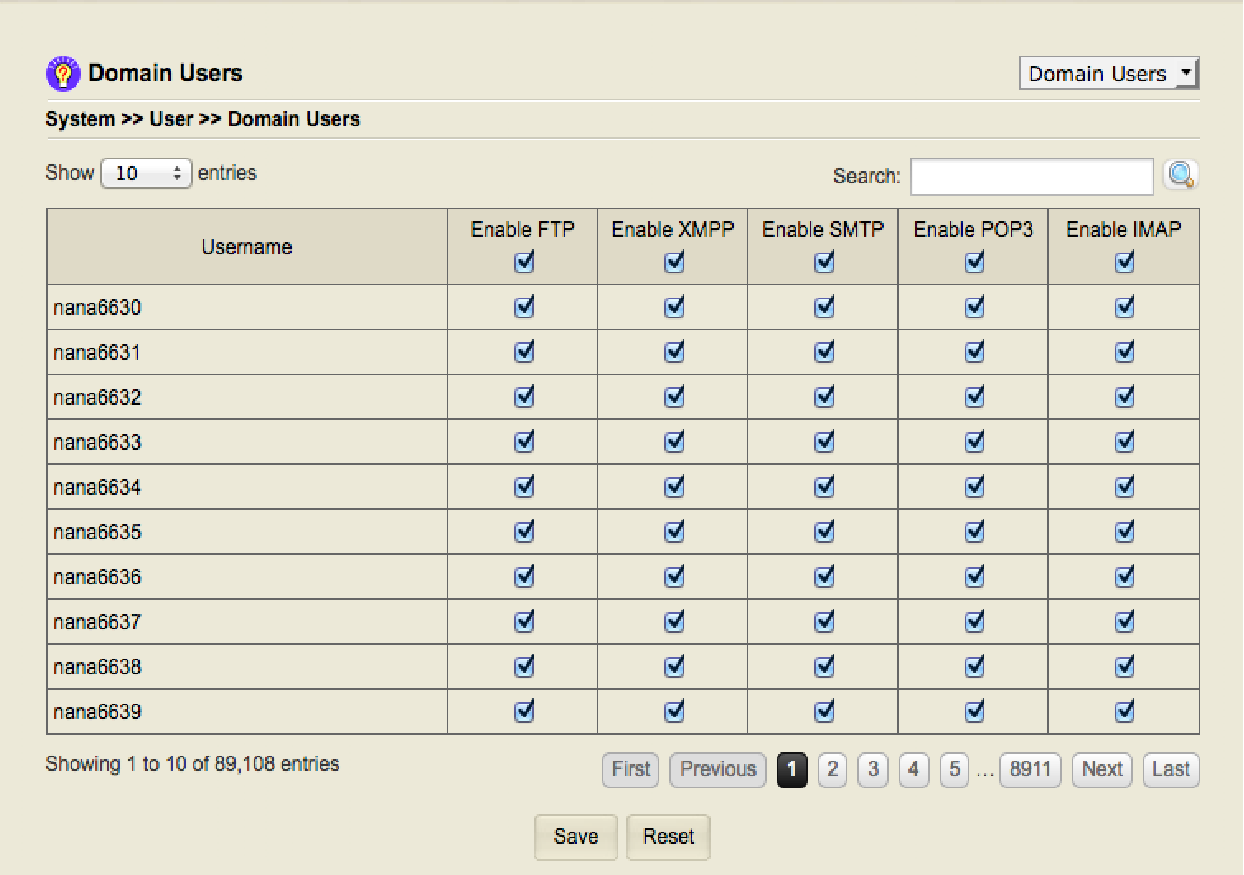
We usually create user data in LDAP server as follows:

```
dn: uid=nana6853,ou=People,dc=test,dc=local
uid: nana6853
cn: nana6853
objectClass: account
objectClass: posixAccount
objectClass: AsteriskSIPUser
objectClass: top
objectClass: inetLocalMailRecipient
userPassword:: ZTJjZGJlMDk=
loginShell: /bin/bash
uidNumber: 6853
gidNumber: 1007
homeDirectory: /home1/nana6853
gecos: Nana6853 Louis
mailHost: 192.168.11.251
AstAccountName: 6853
AstAccountHost: dynamic
AstAccountType: friend
AstAccountRealmedPassword: e2cdbe09
AstAccountCallerID: nana6853<6853>
AstAccountDirectMedia: no
AstAccountCallGroup: 1
AstAccountPickupGroup: 1
AstAccountMailbox: 6853
AstAccountQualify: yes
AstAccountContext: all
structuralObjectClass: account
entryUUID: d706f363-8ea4-4f4a-8cc0-d9a42fbeeb1c
creatorsName: cn=Manager,dc=test,dc=local
createTimestamp: 20211001012917Z
entryCSN: 20211001012917.473081Z#000000#000#000000
modifiersName: cn=Manager,dc=test,dc=local
modifyTimestamp: 20211001012917Z
```

You might use your LDAP browser or editor to create user data.

In the example provided in the screen snapshot, we just bind the LDAP Server “127.0.0.1” (localhost) for user authentication. After pressing “**Submit**” button, you should reboot the host to make the setting take effective properly.

To check if it is binding successfully, you can check from “**System >> User >> Users**” by switching the field on the top from “**Local Users**” to “**Domain Users**”:



System >> User >> Domain Users

Show 10 entries

Search:

Username	Enable FTP ☑	Enable XMPP ☑	Enable SMTP ☑	Enable POP3 ☑	Enable IMAP ☑
nana6630	☑	☑	☑	☑	☑
nana6631	☑	☑	☑	☑	☑
nana6632	☑	☑	☑	☑	☑
nana6633	☑	☑	☑	☑	☑
nana6634	☑	☑	☑	☑	☑
nana6635	☑	☑	☑	☑	☑
nana6636	☑	☑	☑	☑	☑
nana6637	☑	☑	☑	☑	☑
nana6638	☑	☑	☑	☑	☑
nana6639	☑	☑	☑	☑	☑

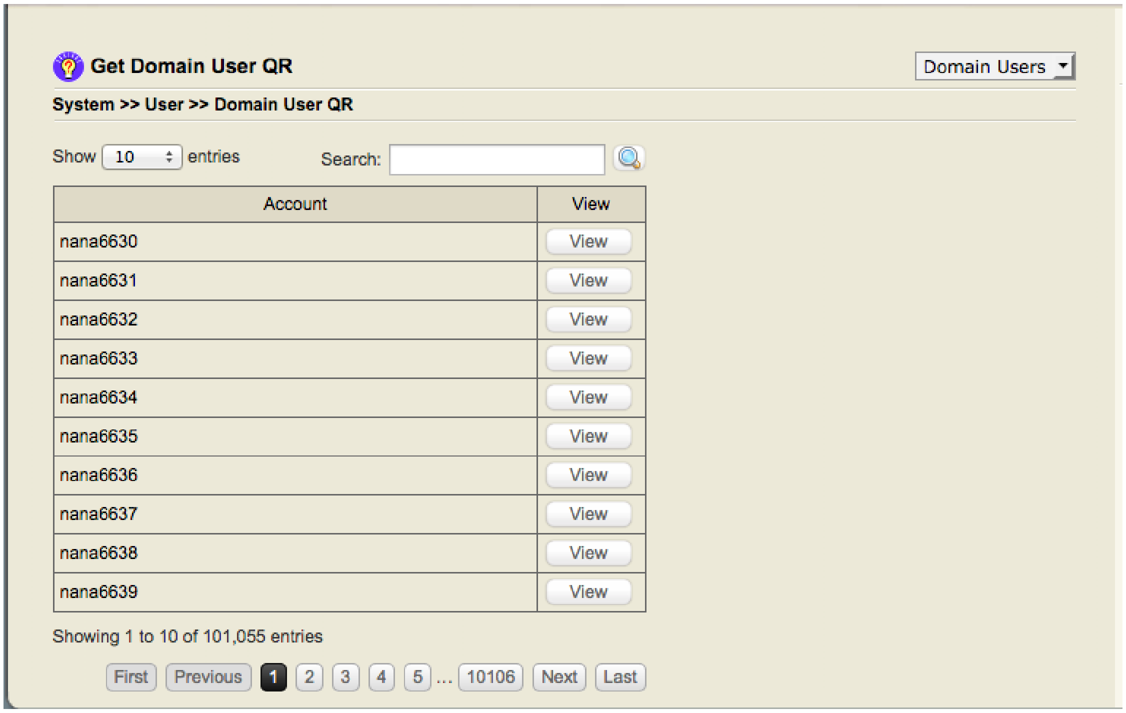
Showing 1 to 10 of 89,108 entries

Illustration 3: Listing the User Accounts from binding LDAP Server

Please note that the screen would not give the complete list of the data in LDAP server if the volume of the data is too large. The “**Search**” function is provided for you to look up specific user account. Currently, “**Search**” in LDAP is only done by “**exact match**” – it only displays the user account exactly matched the search string.

Similarly, you can view the QR code for specific user account from “**System >> User >> User QR**”. The QR code will be displayed by clicking the “**View**” button associated with user account on the left.

Those screens shown here for user account information are meant for checking if the binding is successful; it is not intended for user management. The User Management shall be done via LDAP browser/editor.



Get Domain User QR

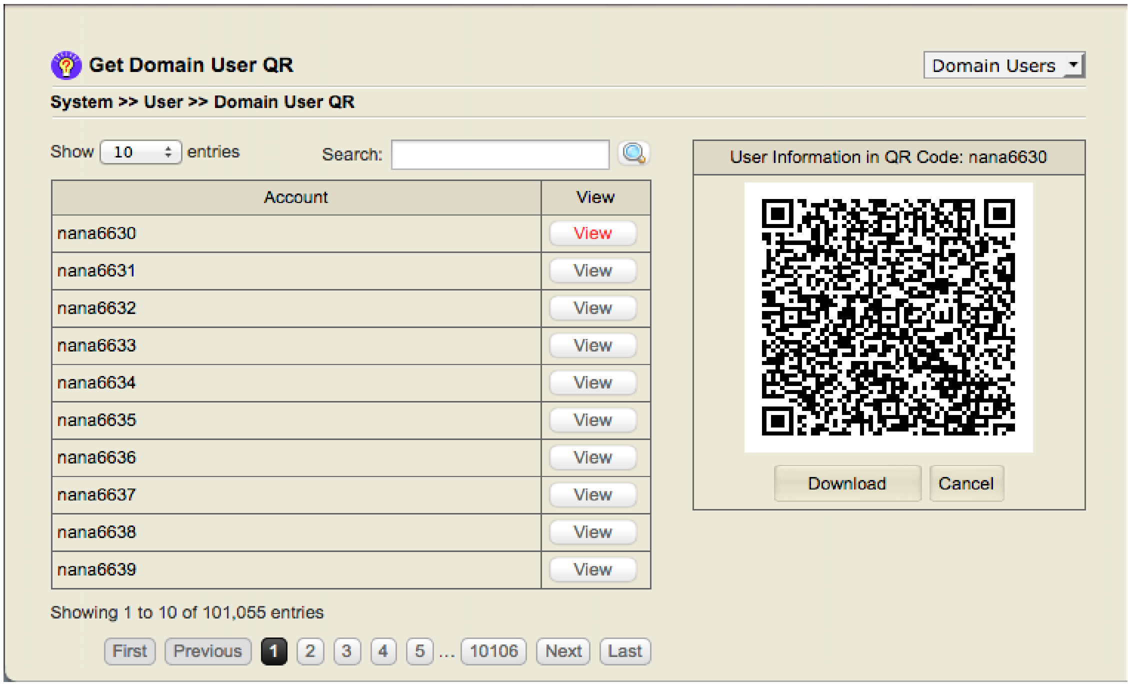
System >> User >> Domain User QR

Show 10 entries Search:

Account	View
nana6630	View
nana6631	View
nana6632	View
nana6633	View
nana6634	View
nana6635	View
nana6636	View
nana6637	View
nana6638	View
nana6639	View

Showing 1 to 10 of 101,055 entries

Illustration 4: Display the QR code associated with User Account - 1



Get Domain User QR

System >> User >> Domain User QR

Show 10 entries Search:

Account	View
nana6630	View
nana6631	View
nana6632	View
nana6633	View
nana6634	View
nana6635	View
nana6636	View
nana6637	View
nana6638	View
nana6639	View

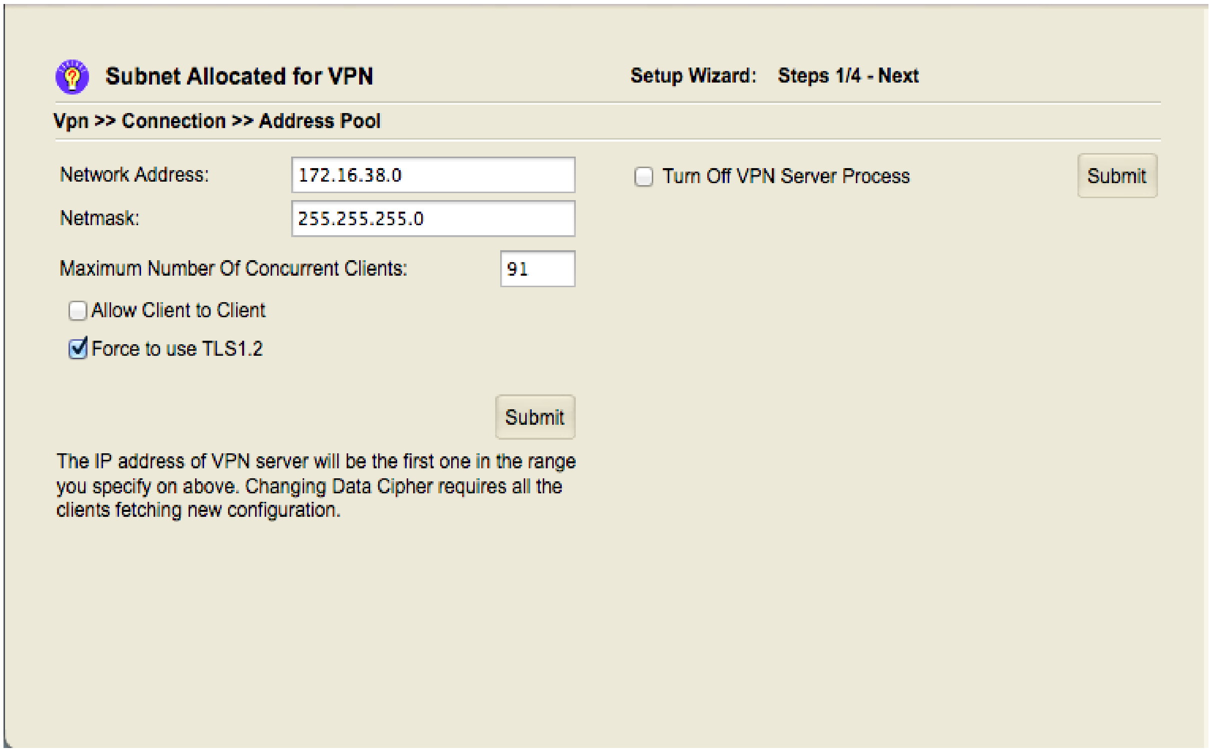
Showing 1 to 10 of 101,055 entries

User Information in QR Code: nana6630

Illustration 5: Display the QR code associated with User Account - 2

XMPP Cluster Setting

Before we go into the setting of XMPP cluster, Let's take a look at VPN address pool (“**Vpn >> Connection >> Address Pool**”):



Subnet Allocated for VPN

Setup Wizard: Steps 1/4 - Next

Vpn >> Connection >> Address Pool

Network Address: 172.16.38.0

Netmask: 255.255.255.0

Maximum Number Of Concurrent Clients: 91

□ Allow Client to Client

☑ Force to use TLS1.2

□ Turn Off VPN Server Process

The IP address of VPN server will be the first one in the range you specify on above. Changing Data Cipher requires all the clients fetching new configuration.

Illustration 6: VPN Address Pool

The diagram above reveals that the address pool for VPN is 172.16.38.0/24. And please remember that the first IP address of the address pool is the IP address of the host in VPN. In this case, it is “172.16.38.1”.

With this information at hand, let's go to the setting at “**System >> Management >> XMPP User Control**”.

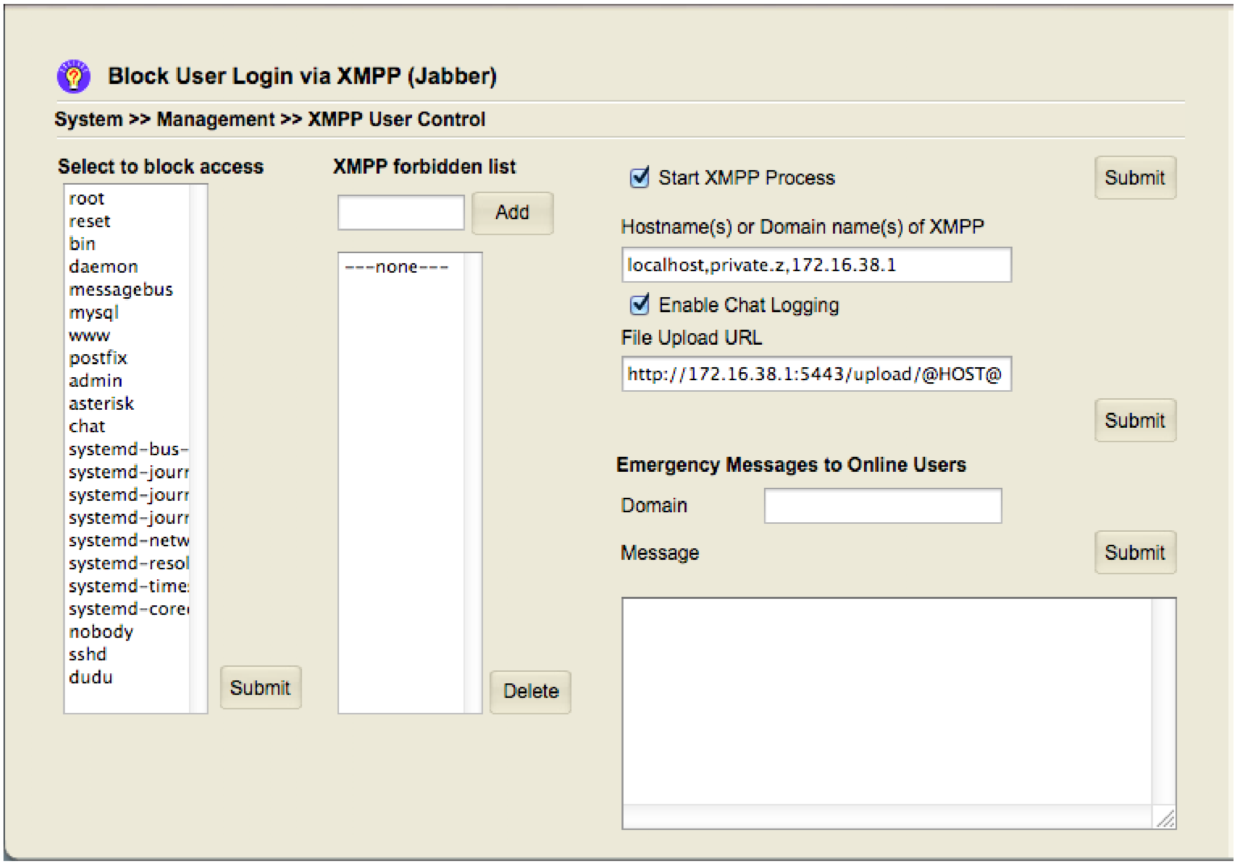
The option “**Enable Chat Logging**” is to record XMPP messages between users. The conversation log will be kept in the directory “/home/chat”.

Please note the field “**File Upload URL**”. This is provided for XMPP clients to exchange files via HTTP. In the example above, we would like to force “**HTTP File Upload**” to go through VPN. Thus, “172.16.38.1” is used in URL –

that is the IP address of the host in VPN. The uploaded files will be kept in the directory “/home/flv”.

For multiple-host configuration, we just let each host with the same IP addresses for VPN; and “**File Upload URL**” is the same on each host as well. As we know that every uploaded files will go to the directory “/home/flv” on each host. We use an external storage server such that the same storage space will be mount on the directory “/home/flv” on each host joining the XMPP cluster. Therefore, the file uploaded from host A can be accessed by the user on host B.

The recording of the XMPP text messaging is not related to VPN; but when users on host A chat with users on host B, the associated conversation shall be writing into the same place. Thus, we can also mount a remote file system to “/home/chat” to record the conversation.



Block User Login via XMPP (Jabber)

System >> Management >> XMPP User Control

Select to block access

root
reset
bin
daemon
messagebus
mysql
www
postfix
admin
asterisk
chat
systemd-bus-
systemd-jourr
systemd-jourr
systemd-jourr
systemd-netw
systemd-resol
systemd-time:
systemd-core
nobody
sshd
dudu

XMPP forbidden list

---none---

☑ Start XMPP Process

Hostname(s) or Domain name(s) of XMPP: localhost,private.z,172.16.38.1

☑ Enable Chat Logging

File Upload URL: http://172.16.38.1:5443/upload/@HOST@

Emergency Messages to Online Users

Domain

Message

Illustration 7: Setting of File Upload URL

With those issues settled, we start to configure XMPP cluster.

The operation principle for building XMPP cluster is straightforward. On each host in the cluster, it must have a "**Node Name**" to identify itself. This "**Node Name**" should associated with an IP address so that the other nodes know where to find the host by using "**Node Name**".

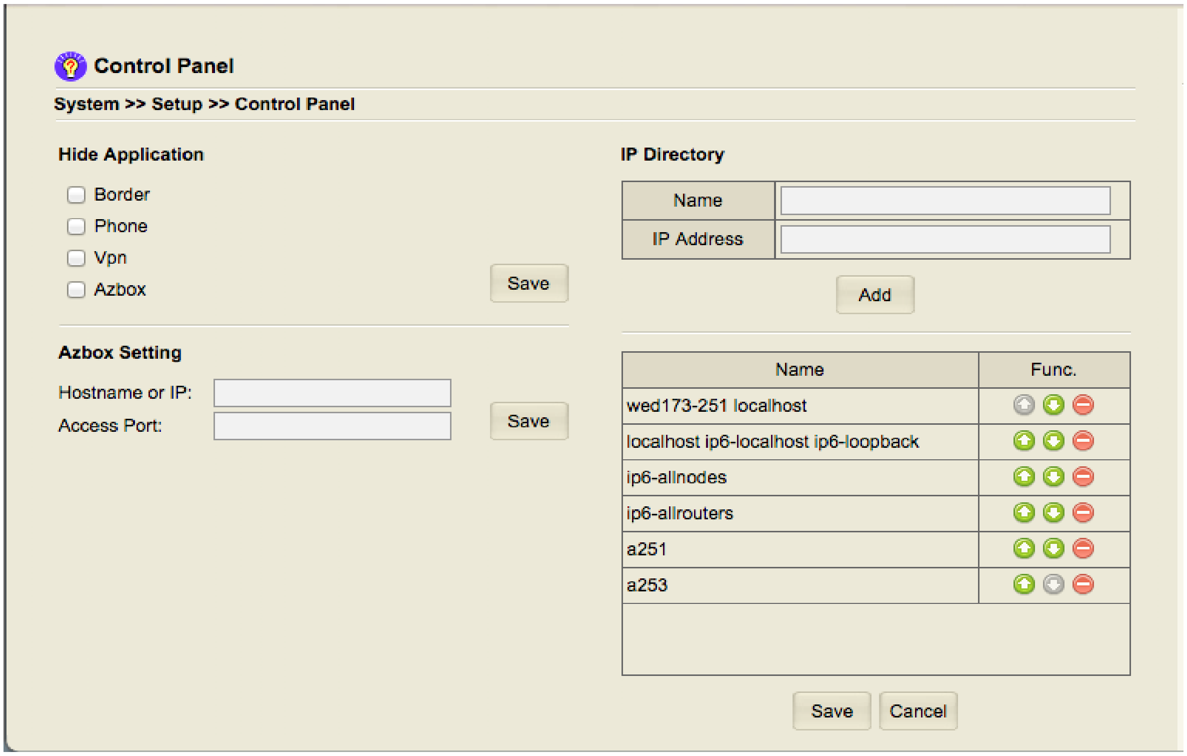
Please note: once you set "**Node Name**", all the databases for XMPP on this host will be purged, and recreated by using this "**Node Name**". Therefore, you have to decide if this host should join the cluster from the very beginning. Using this host for a while and joining the cluster later will cause the loss of the previous data (friend's list, off-line messages, ...)

For a node to join a cluster, it simply just binds any node in the cluster.

We start with one node. A host with a **Node Name** "a253" is created. Then we would like to have another node "a251" to bind "a253" in order to join the cluster.

For "a251" to join successfully, the following things shall be done in advance:

1. check "**System >> Setup >> Control Panel**" to see if the IP address records of "a251" and "a253" exist;
2. export the "**cookie file**" from "a253" and place this cookie file to "a251".
3. From "a251", start the action to join the cluster by using "a253".



Control Panel

System >> Setup >> Control Panel

Hide Application

- □ Border
- □ Phone
- □ Vpn
- □ Azbox

Azbox Setting

Hostname or IP:

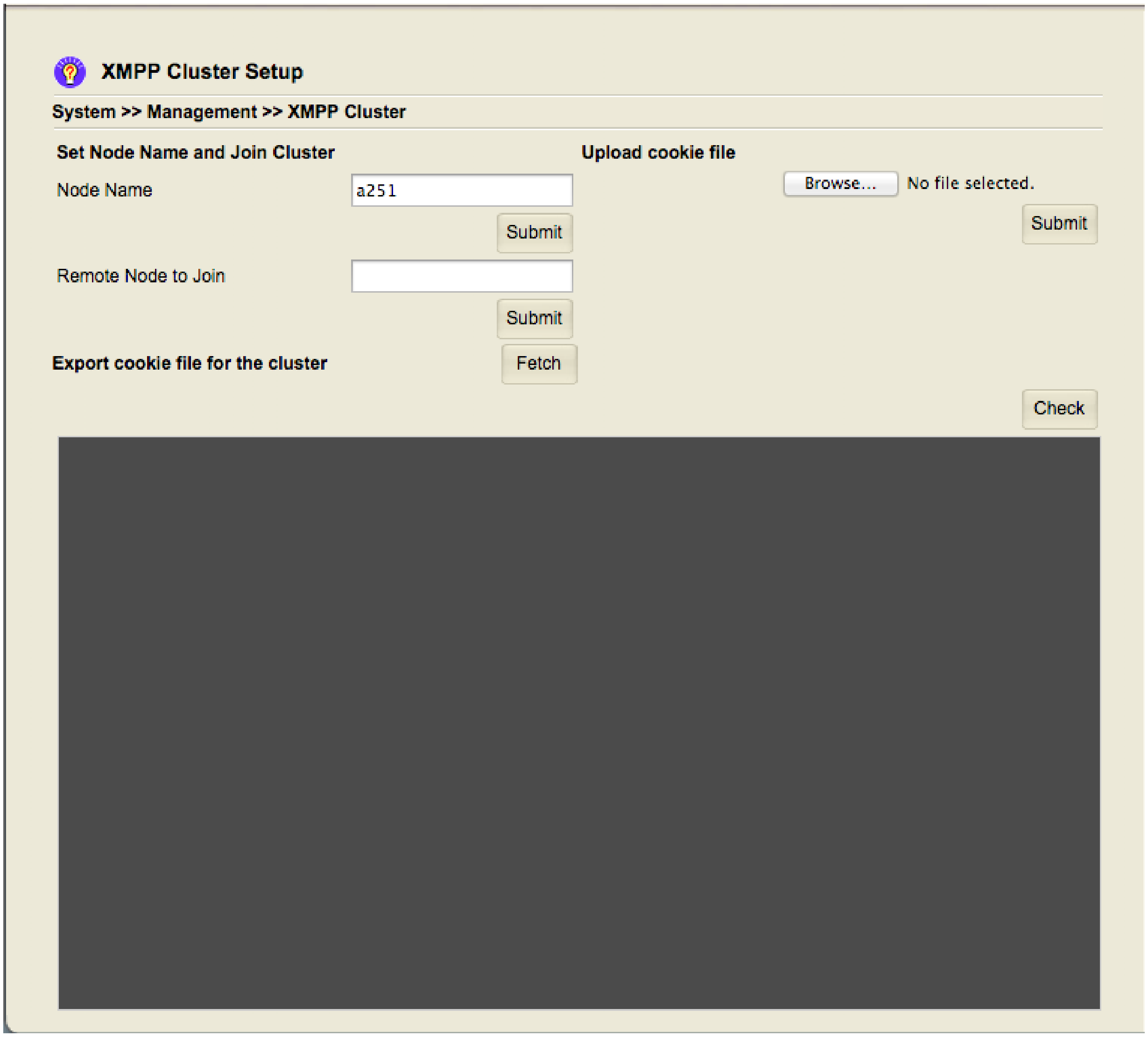
Access Port:

IP Directory

Name	
IP Address	

Name	Func.
wed173-251 localhost	↑ ↓ ⊖
localhost ip6-localhost ip6-loopback	↑ ↓ ⊖
ip6-allnodes	↑ ↓ ⊖
ip6-allrouters	↑ ↓ ⊖
a251	↑ ↓ ⊖
a253	↑ ↓ ⊖

Illustration 8: IP Address Records of Nodes in XMPP Cluster



XMPP Cluster Setup

System >> Management >> XMPP Cluster

Set Node Name and Join Cluster

Node Name: a251

Remote Node to Join:

Export cookie file for the cluster

Upload cookie file

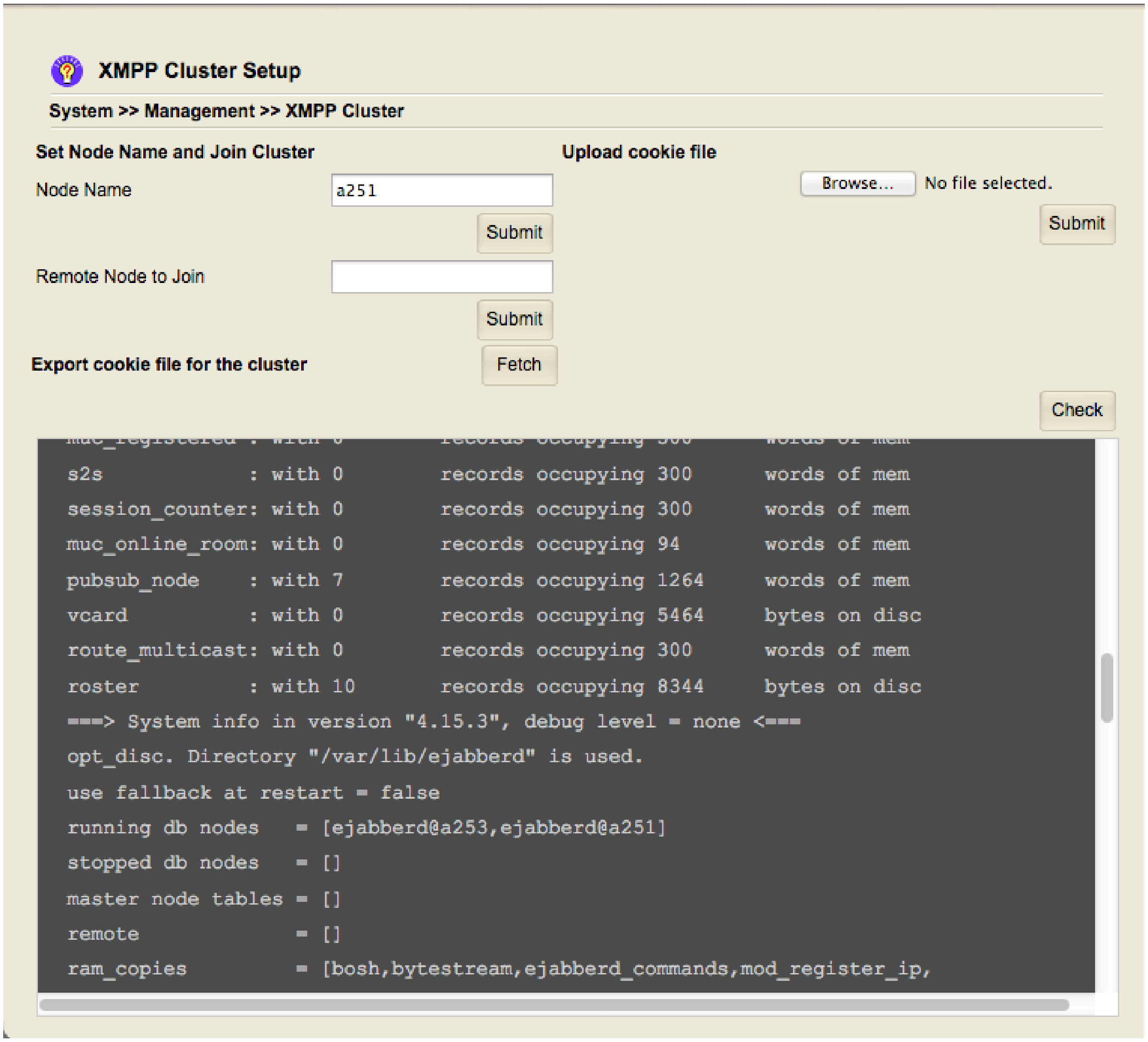
No file selected.

Illustration 9: Before Joining the Cluster

The diagram above is the screen snapshot of “a251” before joining the cluster. Enter “a253” in the field “**Remote Node to Join**” and Press “**Submit**” button. This should let “a251” join the cluster.

You might press “**Check**” button to check if it joins the cluster successfully. In the diagram below, it displays with the message that

running db nodes = [ejabberd@a253,ejabberd@a251]



XMPP Cluster Setup

System >> Management >> XMPP Cluster

Set Node Name and Join Cluster

Node Name: a251

Remote Node to Join:

Export cookie file for the cluster

Upload cookie file

No file selected.

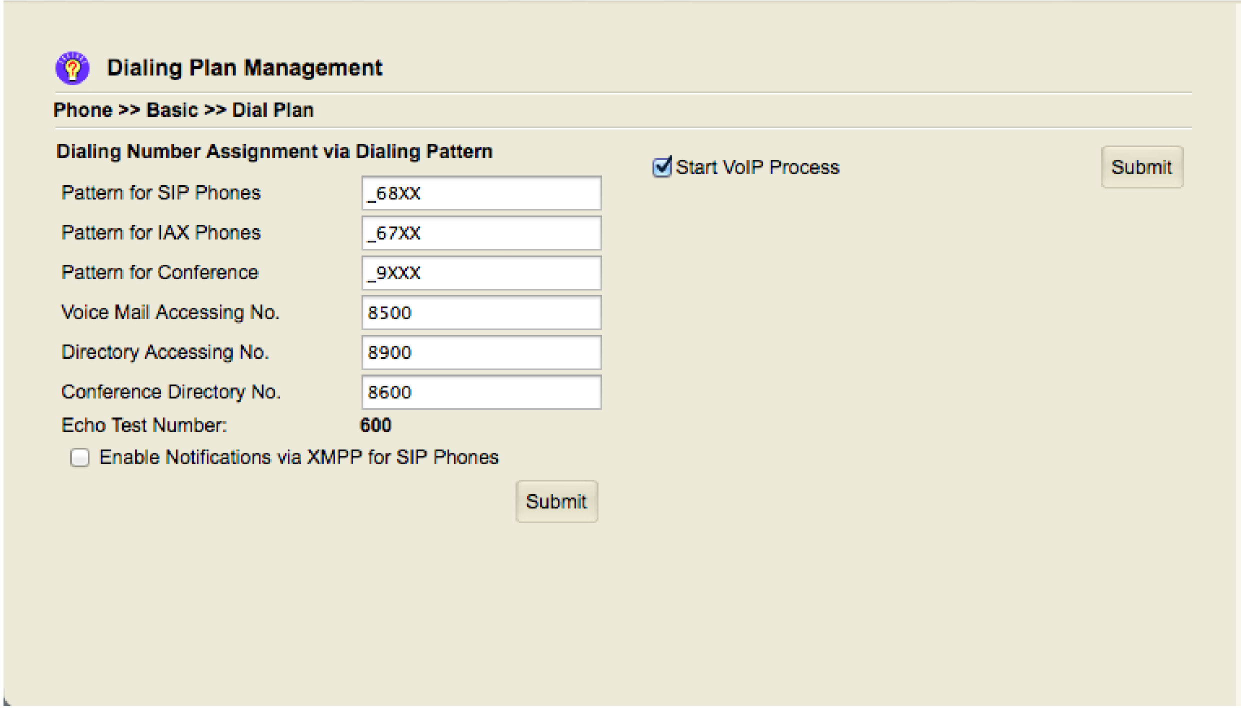
```
muc_registered : with 0         records occupying 300      words of mem
s2s            : with 0         records occupying 300      words of mem
session_counter: with 0         records occupying 300      words of mem
muc_online_room: with 0         records occupying 94       words of mem
pubsub_node    : with 7         records occupying 1264     words of mem
vcard          : with 0         records occupying 5464     bytes on disc
route_multicast: with 0         records occupying 300      words of mem
roster         : with 10        records occupying 8344     bytes on disc
===> System info in version "4.15.3", debug level = none <===
opt_disc. Directory "/var/lib/ejabberd" is used.
use fallback at restart = false
running db nodes   = [ejabberd@a253,ejabberd@a251]
stopped db nodes   = []
master node tables = []
remote             = []
ram_copies         = [bosh,bytestream,ejabberd_commands,mod_register_ip,
```

Illustration 10: After Joining the Cluster

This confirms the binding is successful.

SIP Setting

SIP is using "**number routing scheme**" between hosts. Specifically, we also use patterns of numbers to distinguish functions of the telephone system.



Dialing Plan Management

Phone >> Basic >> Dial Plan

Dialing Number Assignment via Dialing Pattern

Pattern for SIP Phones	_68XX
Pattern for IAX Phones	_67XX
Pattern for Conference	_9XXX
Voice Mail Accessing No.	8500
Directory Accessing No.	8900
Conference Directory No.	8600
Echo Test Number:	600

□ Enable Notifications via XMPP for SIP Phones

☑ Start VoIP Process

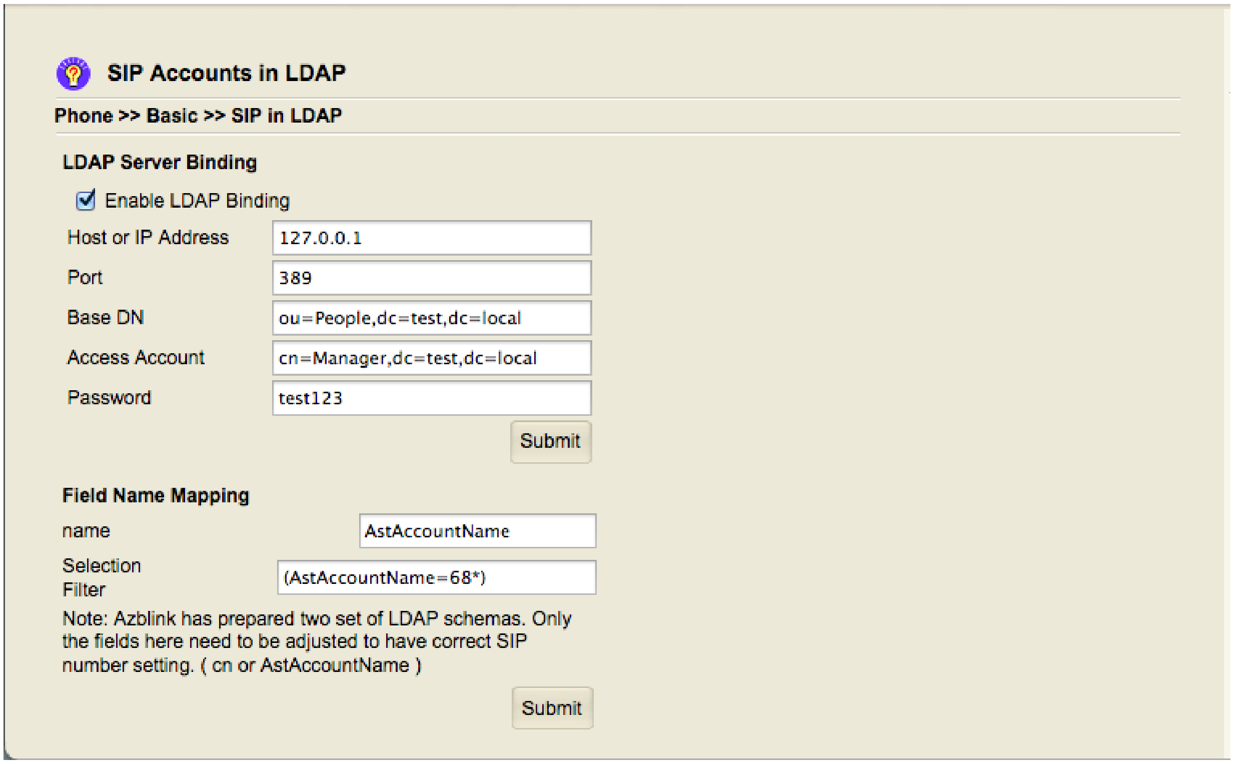
Illustration 11: Dialing Plan

While devising numbering schemes, we need to think if those numbers can be used "**locally**" or "**globally across the hosts**". For example, you can have people associated with the same host by dialing 68XX to each other, and people on the other hosts dialing to this host with 30168XX; you also can have the scheme that using 30168XX only without considering if the caller and callee are on the same host. Both schemes shall work, and users must be very clear about those dialing rules; otherwise, it would cause some confusions.

In general, we require people do **SIP registration and XMPP registration on the same host by using VPN** to that host. Theoretically, they can be on different hosts: for example, using VPN on **host C,** doing SIP registration on **host A,** and doing XMPP registration on **host B --** but this increases the complexity of the management. In the scenario of multiple-host configuration, we just require those features should be binding together.

Given on the situation described above, then we can think if we should send FCM/APNS/PushKit along with XMPP to notify the callee while there is a inbound call to the host. The notification via XMPP can be enabled by checking the box "**Enable Notifications via XMPP for SIP Phones**" along the setting in other screen.

For using the accounts established in LDAP server, it should bind the LDAP server properly. It can be done via "**Phone >> Basic >> SIP in LDAP**":



SIP Accounts in LDAP

Phone >> Basic >> SIP in LDAP

LDAP Server Binding

☑ Enable LDAP Binding

Host or IP Address	127.0.0.1
Port	389
Base DN	ou=People,dc=test,dc=local
Access Account	cn=Manager,dc=test,dc=local
Password	test123

Field Name Mapping

name	AstAccountName
Selection Filter	(AstAccountName=68*)

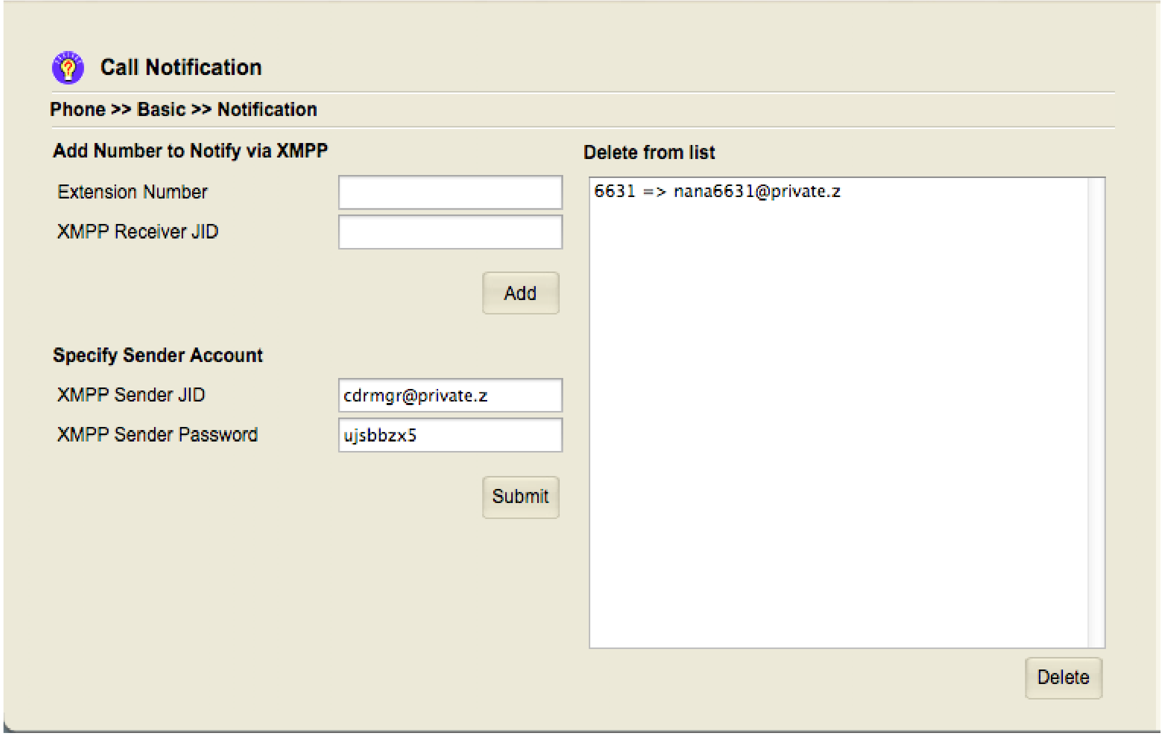
Note: Azblink has prepared two set of LDAP schemas. Only the fields here need to be adjusted to have correct SIP number setting. (cn or AstAccountName)

Illustration 12: Binding LDAP Server to get SIP Accounts

Please note that the SIP accounts on this host shall be like 68XX. Thus, we have "**Selection Filter**" set for extracting the SIP accounts we need for this host.

We continue with the topic to set up XMPP notification when there is an inbound call. On "**Phone >> Basic >> Notification**", you would notice a

“**Sender Account**” is required. You should have that “**Sender Account**” created at first in order to use that account to send “**XMPP notification**” from this host. Then, enter the phone number with associated the JID (the receiver's XMPP account). Thus, when the phone number is called, the system will send a notification to that XMPP account.



XMPP Sender JID: cdrmgr@private.z

XMPP Sender Password: ujsbbzx5

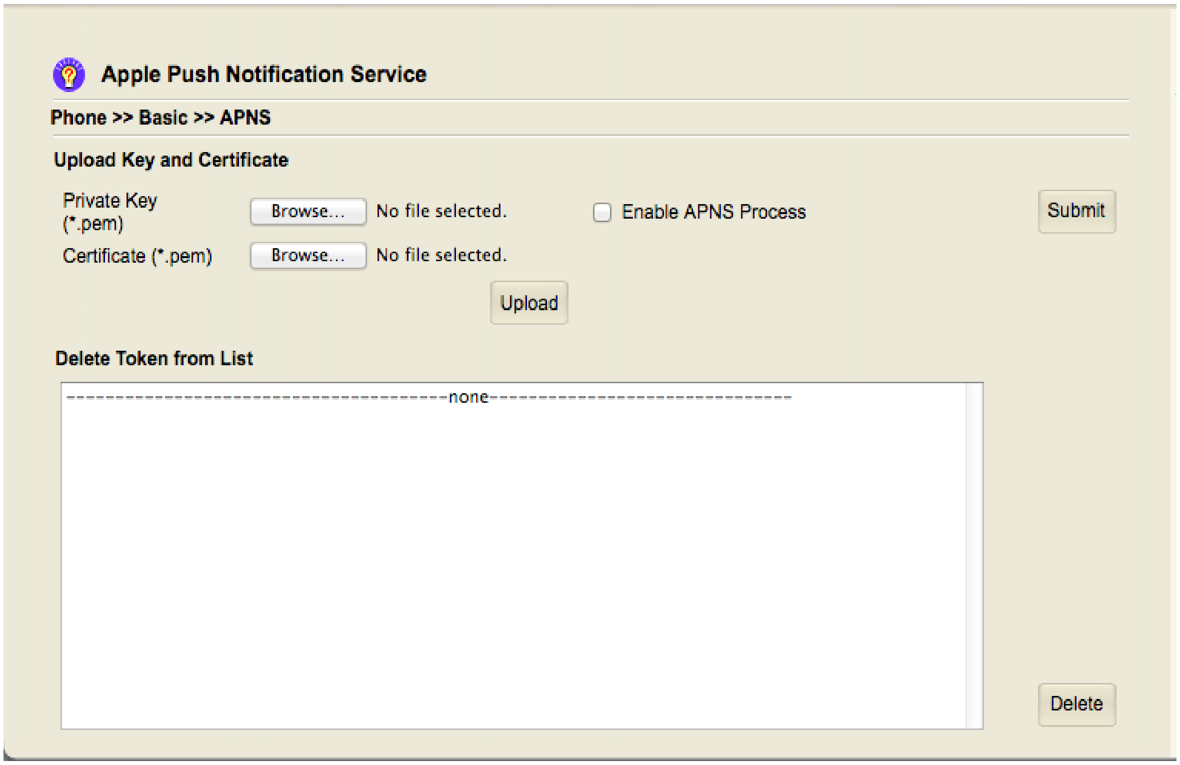
6631 => nana6631@private.z

Illustration 13: Notify via XMPP while there is Inbound Call

And FCM/APNS/PushKit can be enabled via “**Phone >> Basic >> APNS**” and “**Phone >> Basic >> FCM**”.

You can check if the associated device token exists or not. If devices tokens are not there, of course there is no FCM/APNS/PushKit.

The PushKit tokens are not listed; enabling APNS will enable PushKit as well. Please also not that **Apple requires the key and certificate of using APNS and PushKit be updated every year**. Once the certificate is expired, the developer of APP on mobile device should go to Apple's development website to fetch new certificate.



Apple Push Notification Service

Phone >> Basic >> APNS

Upload Key and Certificate

Private Key (*.pem) Browse... No file selected.

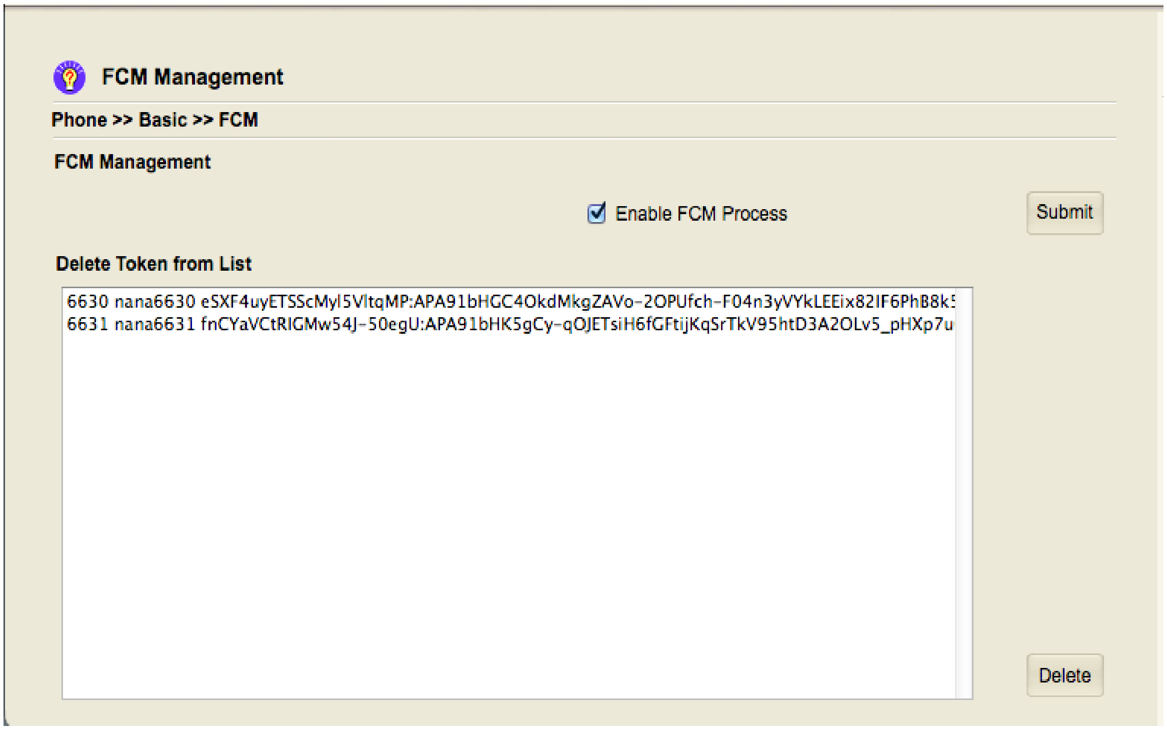
Certificate (*.pem) Browse... No file selected.

☐ Enable APNS Process

Delete Token from List

--------------------------------------none------------------------------

Illustration 14: APNS/PushKit Setting



FCM Management

Phone >> Basic >> FCM

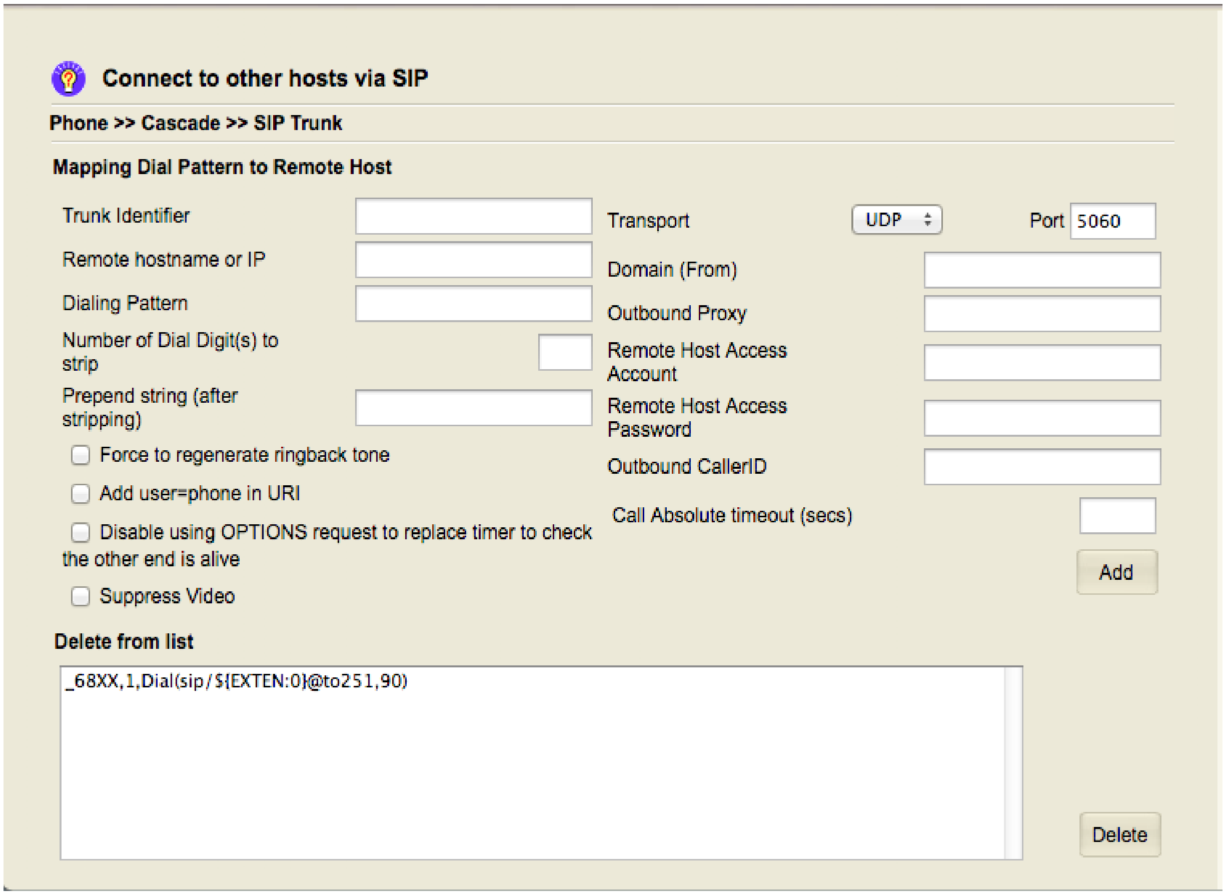
FCM Management

☑ Enable FCM Process

Delete Token from List

```
6630 nana6630 eSXF4uyETSScMyl5VltqMP:APA91bHGC4OkdMkgZAVo-2OPUfch-F04n3yVYkLEEix82IF6PhB8k
6631 nana6631 fnCYaVCtRIGMw54J-50egU:APA91bHK5gCy-qOJETsiH6fGFtijKqSrTkV95htD3A2OLv5_pHXp7
```

Illustration 15: FCM Setting



Connect to other hosts via SIP

Phone >> Cascade >> SIP Trunk

Mapping Dial Pattern to Remote Host

Trunk Identifier

Remote hostname or IP

Dialing Pattern

Number of Dial Digit(s) to strip

Prepend string (after stripping)

☐ Force to regenerate ringback tone

☐ Add user=phone in URI

☐ Disable using OPTIONS request to replace timer to check the other end is alive

☐ Suppress Video

Transport UDP

Port 5060

Domain (From)

Outbound Proxy

Remote Host Access Account

Remote Host Access Password

Outbound CallerID

Call Absolute timeout (secs)

Delete from list

```
_68XX,1,Dial(sip/${EXTEN:0}@to251,90)
```

Illustration 16: SIP Trunk

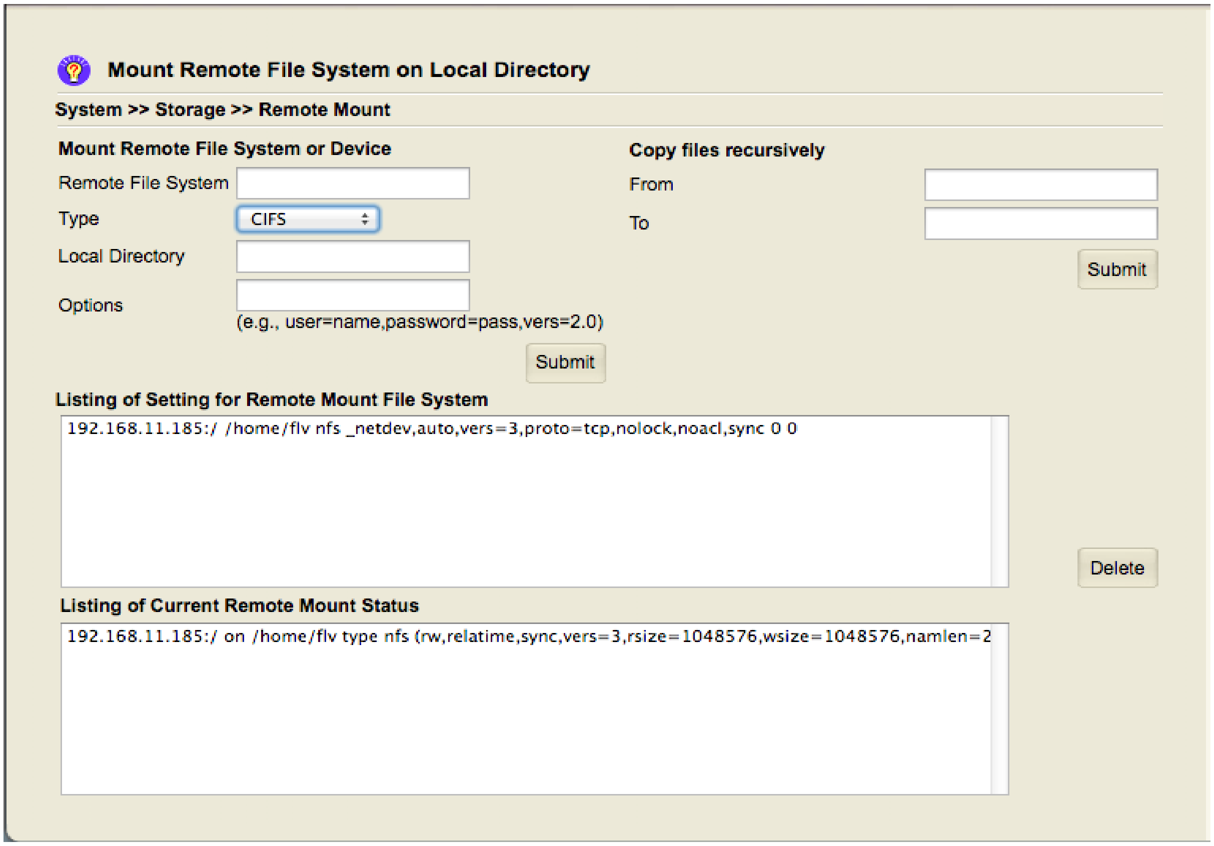
SIP trunk can be set via “**Phone >> Cascade >> SIP Trunk**”. For two Azblink hosts to set up SIP trunk between them, the simplest way is just using IP address to point to each other without further authentication.

Storage Space

The following two features would use huge amount of storage space: the **file transfer via HTTP in XMPP** and the **recording of voice calls and text messaging**. File transfer via HTTP in XMPP will use the directory "/home/flv" as deposit for the uploaded/downloaded files; the recording of the voice calls and text messaging will be kept under the directory "/home/chat". In the scenario of single-host deployment, we may or may not resort to the external storage for help.

However, in multiple-host configuration, mounting remote file system to those directories on each host is not only the issue regarding to storage space issue, but also that we have the need for multiple hosts to access a common spot in order to exchange files or merging the data from different hosts. Thus, if we intend to use "**File Transfer via HTTP in XMPP**" or "**Recording Voice or Text Messaging**", it is necessary to have external storage in place.

In "**System >> Storage >> Remote Mount**", Samba (CIFS), NFS, or GlusterFS (in "wed" build) can be used to mount remote file system to a local directory.



Mount Remote File System on Local Directory

System >> Storage >> Remote Mount

Mount Remote File System or Device

Remote File System

Type: CIFS

Local Directory

Options

(e.g., user=name,password=pass,vers=2.0)

Copy files recursively

From

To

Listing of Setting for Remote Mount File System

```
192.168.11.185:/ /home/flv nfs _netdev,auto,vers=3,proto=tcp,nolock,noacl,sync 0 0
```

Listing of Current Remote Mount Status

```
192.168.11.185:/ on /home/flv type nfs (rw,relatime,sync,vers=3,rsize=1048576,wsize=1048576,namlen=2
```

Illustration 17: Mount Remote File System to a Local Directory

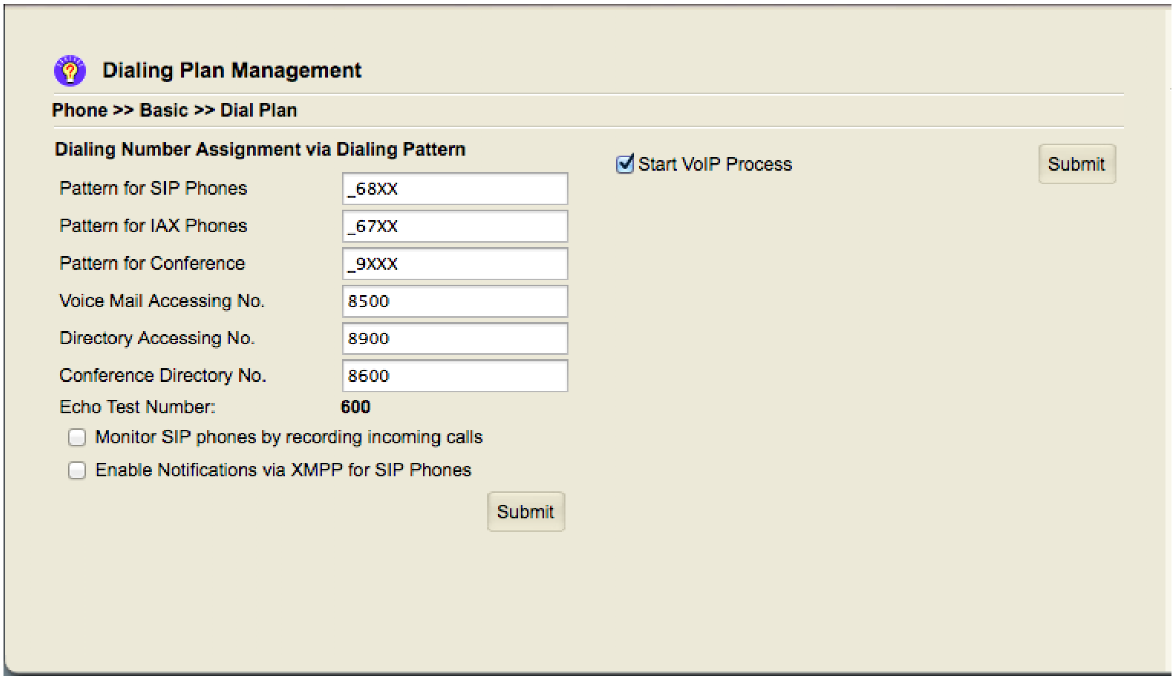
Samba (CIFS) allows using a user account to mount a file system. And please make sure that the directory on the remote storage should allow **READ and WRITE** privilege via this user account – the directory should be writable/readable right after the login process.

On NFS, we use “root” user to mount remote file system and **READ/WRITE** actions will be done via “root” user as well. The remote system has to allow “root” user to perform those actions, for example, doing “**no_root_squash**” while exporting the file system or changing the ownership of that directory.

At the end of the day (the recording day), the directories or files under “/home/flv” will be moved to “/home/chat” for archive purpose. Thus, the same remote file system should not be mount on “/home/flv” and “/home/chat” at the same time; it needs another storage system if both functions are in use. And we do not have automatic scheme to purge the directories or files under "/home/chat". Thus, it needs proper capacity planning for using these functions.

Voice Recording

Some variants of our software provide the function of recording voice calls. In “**Phone >> Basic >> Dial Plan**”, if you see the check box “**Monitor SIP phones by recording incoming calls**”, the recording function will be triggered when there is a inbound call to those SIP phones specified in the dialing pattern when the box is checked.



Dialing Plan Management

Phone >> Basic >> Dial Plan

Dialing Number Assignment via Dialing Pattern

Pattern for SIP Phones	_68XX
Pattern for IAX Phones	_67XX
Pattern for Conference	_9XXX
Voice Mail Accessing No.	8500
Directory Accessing No.	8900
Conference Directory No.	8600
Echo Test Number:	600

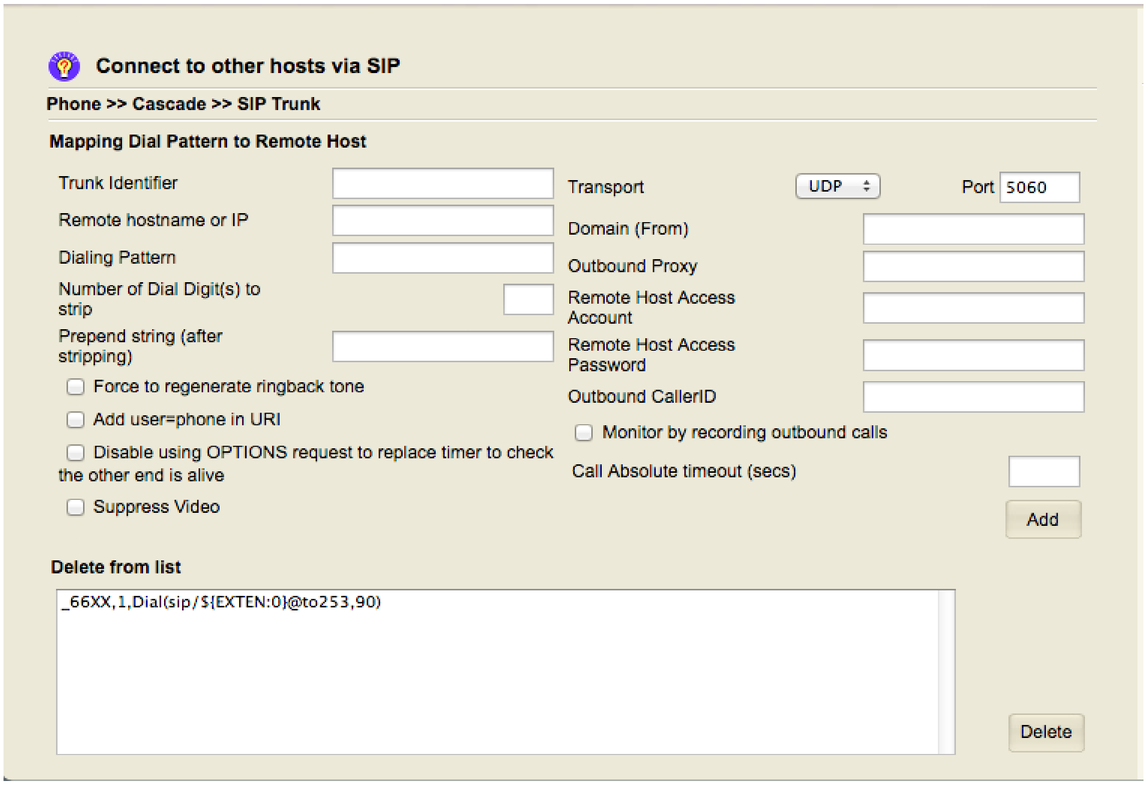
□ Monitor SIP phones by recording incoming calls

□ Enable Notifications via XMPP for SIP Phones

☑ Start VoIP Process

Illustration 18: Recording the Inbound Calls

The recorded files can be found via Web GUI “**Phone >> Audit >> Voice Records**” at the same day. By the end of the day, those files will be moved to the directory “/home/chat” to archive.



Connect to other hosts via SIP

Phone >> Cascade >> SIP Trunk

Mapping Dial Pattern to Remote Host

Trunk Identifier		Transport	UDP Port 5060
Remote hostname or IP		Domain (From)	
Dialing Pattern		Outbound Proxy	
Number of Dial Digit(s) to strip		Remote Host Access Account	
Prepend string (after stripping)		Remote Host Access Password	
		Outbound CallerID	

□ Force to regenerate ringback tone

□ Add user=phone in URI

□ Disable using OPTIONS request to replace timer to check the other end is alive

□ Suppress Video

□ Monitor by recording outbound calls

Call Absolute timeout (secs)

Delete from list

```
_66XX,1,Dial(sip/${EXTEN:0}@to253,90)
```

Illustration 19: Recording Outbound Calls While Setting Up SIP Trunk

The SIP trunk setting can also provide the function for recording. In "**Phone >> Cascade >> SIP Trunk**", if the check box "**Monitor by recording outbound calls** " is shown and checked, the outgoing calls via this SIP trunk will be recorded. Similarly, the recorded files will be shown in "**Phone >> Audit >> Voice Records**" at the same day; by the end of the day, they will be moved to "/home/chat".

Please note that we do not have automatic mechanism about how often to purge those files under the directory "/home/chat". You shall be prudent about the capacity planning on those recording functions.