Integrate Asterinic Call Center with Yeastar S series

User Manual

Yeastar Information Technology Co. Ltd.
Integrate Asterinic Call Center with Yeastar S series

Catalog

ABOUT THIS GUIDE ........................................................................................................... 3
SYSTEM REQUIREMENTS .................................................................................................. 3
CLIENT REQUIREMENTS ..................................................................................................... 3
PREPARE THE LINUX SERVER WITH LAMP ................................................................. 4
  Step One — Install Apache ......................................................................................... 4
  Step Two — Install MySQL (MariaDB) ...................................................................... 5
  Step Three — Install PHP ......................................................................................... 6
  Step Four — Test PHP Processing on your Web Server ........................................... 7
INSTALLING ASTERNIC ON LINUX SERVER ................................................................. 9
CONFIGURATION OF S SERIES PBX .......................................................................... 11
  Enable the queue log and AMI used by Asternic .................................................... 11
  Create SSH key .......................................................................................................... 14
  Start the Log replication on Yeastar ......................................................................... 15
  Manager real-time view .............................................................................................. 17
ENJOY THE PERFECT CALL CENTER SOLUTION ...................................................... 18
About This Guide

Asternic Call Center Stats is a queue reporting solution to get the best breed and metrics available for the open source Asterisk© PBX. If you have a Yeastar S series PBX embedded machine for your telephony solution, you can still use Asternic Call Center Stats to have good reporting on your queue activity. The only catch is that you must install Asternic on a separate Linux server as the little box does not have a robust and large enough permanent storage for the data.

So, first of all, you must have a Linux box ready with Apache, PHP and MySQL installed. You can use a virtual machine with at least 512Mb of RAM for that purpose. That should be enough for small/medium size call centers. If you have a large call center and have thousands of calls per day, then you need at least 2 GB or RAM.

In this guide, Linux server (Centos) has the IP address 192.168.9.69 and the Yeastar S PBX has the IP address 192.168.9.67.

System Requirements

Asternic Call Center Stats PRO is a PHP application, encoded with ioncube. There are certain requirements to be met in order to successfully run the software.

Server Requirements

On the server side, you need:
- A web server (Apache/Nginx)
- 32 or 64 bits Linux Operating System
- PHP 5.1 or above
- MySQL Server 5 or above
- ioncube Loader

It also has a log processing daemon that is written in Perl, and requires the Perl-Time-HiRes module to be installed. The log processing daemon should run on the background at all times, and it will feed any data that is written into the Asterisk queue log files in its own database.

Client Requirements

You access reports using your web browser. For a full experience, you have to use a modern one:
- Modern Web Browser (IE8 or above, Google Chrome, Safari, Firefox)
- Javascript enabled
Prepare the Linux server with LAMP

In this example, Centos 7 is recommended, it has better compatibility and stability. You can install it first. LAMP stack is the short name of Linux, Apache web server, MySQL database and PHP. You can refer to this manual to do it.

Since a part of the content is used in the manual, I list the useful commands here.

Step One — Install Apache

We can install Apache easily using CentOS's package manager, yum. A package manager allows us to install most software pain-free from a repository maintained by CentOS.

* sudo yum install httpd *

Since we are using a sudo command, these operations get executed with root privileges. It will ask you for your regular user's password to verify your intentions.

Afterwards, your web server is installed.

Once it installs, you can start Apache on your Linux server:

* sudo systemctl start httpd.service *

You can do a spot check right away to verify that everything went as planned by visiting your server's public IP address in your web browser (see the note under the next heading to find out what your public IP address is if you do not have this information already):

http://your_server_IP_address/

You will see the default CentOS 7 Apache web page, which is there for informational and testing purposes. It should look something like this:
If you see this page, then your web server is now correctly installed. The last thing you will want to do is enable Apache to start on boot. Use the following command to do so:

```
sudo systemctl enable httpd.service
```

### Step Two — Install MySQL (MariaDB)

Now that we have our web server up and running, it is time to install MariaDB, a MySQL drop-in replacement. MariaDB is a community-developed fork of the MySQL relational database management system. Basically, it will organize and provide access to databases where our site can store information. Again, we can use yum to acquire and install our software. This time, we'll also install some other "helper" packages that will assist us in getting our components to communicate with each other:

```
sudo yum install mariadb-server mariadb
```

When the installation is complete, we need to start MariaDB with the following command:

```
sudo systemctl start mariadb
```

Now that our MySQL database is running, we want to run a simple security script that will remove some dangerous defaults and lock down access to our database system a little bit. Start the interactive script by running:

```
sudo mysql_secure_installation
```

The prompt will ask you for your current root password. Since you just installed MySQL, you most likely won't have one, so leave it blank by pressing enter. Then the prompt will ask you if you want to set a root password. Go ahead and enter Y, and follow the instructions:

```
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB root user without the proper authorization.

New password: password
Re-enter new password: password
Password updated successfully!
Reloading privilege tables..
```
... Success!
For the rest of the questions, you should simply hit the “ENTER” key through each prompt to accept the default values. This will remove some sample users and databases, disable remote root logins, and load these new rules so that MySQL immediately respects the changes we have made.
The last thing you will want to do is enable MariaDB to start on boot. Use the following command to do so:

```sh
sudo systemctl enable mariadb.service
```

At this point, your database system is now set up and we can move on.

**Step Three — Install PHP**

PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display.
We can once again leverage the `yum` system to install our components. We’re going to include the `php-mysql` package as well:

```sh
sudo yum install php php-mysql
```

This should install PHP without any problems. We need to restart the Apache web server in order for it to work with PHP. You can do this by typing this:

```sh
sudo systemctl restart httpd.service
```

If we decided that `php-fpm` is something that we need, we could type:

```sh
sudo yum install php-fpm
```
Step Four — Test PHP Processing on your Web Server

In order to test that our system is configured properly for PHP, we can create a very basic PHP script. We will call this script info.php. In order for Apache to find the file and serve it correctly, it must be saved to a very specific directory, which is called the "web root". In CentOS 7, this directory is located at /var/www/html/. We can create the file at that location by typing:

```
sudo vi /var/www/html/info.php
```

This will open a blank file. We want to put the following text, which is valid PHP code, inside the file:

```php
<?php phpinfo(); ?>
```

When you are finished, save and close the file.

If you are running a firewall, run the following commands to allow HTTP and HTTPS traffic:

```
sudo firewall-cmd --permanent --zone=public --add-service=http
sudo firewall-cmd --permanent --zone=public --add-service=https
sudo firewall-cmd --reload
```

Now we can test whether our web server can correctly display content generated by a PHP script. To try this out, we just have to visit this page in our web browser. You'll need your server's public IP address again. The address you want to visit will be:

```
http://your_server_IP_address/info.php
```

The page that you come to should look something like this:
Integrate Asterinic Call Center with Yeastar S series

This page basically gives you information about your server from the perspective of PHP. It is useful for debugging and to ensure that your settings are being applied correctly.

If this was successful, then your PHP is working as expected.

Finally, please double check if you can found this info with ionCube PHP loader (enabled), if not, you will need to install it. If you use Centos7, it’s installed when PHP is installed. This is very important for Asternic.
You probably want to remove this file after this test because it could actually give information about your server to unauthorized users. To do this, you can type this:

```
sudo rm /var/www/html/info.php
```

You can always recreate this page if you need to access the information again later.

## Installing Asterinic on Linux Server

Follow the steps below for an installation. Basically download, extract and run "make".

```
yum -y install wget
cd /usr/src
wget http://download.asternic.net/asternic-stats-pro-2.2.2.tgz
```

**Note:**
Please ensure your CentOS server has network connection, otherwise you cannot download via wget command. Alternatively, you may download ‘asternic-stats-pro-2.2.2.tgz’ via link http://www.asternic.net/download.php and then upload to Linux server via SFTP tool under path /usr/src.

```
tar zxvf asternic-stats-pro-2.2.2.tgz
cd asternic-stats-pro-2.2.2
```

**Note:**
You must know beforehand the MySQL root password in order for the installation to succeed. The database and tables need to be created, and you must enter that password when prompted. Otherwise, installation will be partial and not working.

```
make
```

**Note:**
The installation script above will do several things:
• Check if the software is already installed, if so it will abort and exit
• Check if the database qstats exists, if so, it will abort and exit.
• It will check the ionCube loader for your platform and PHP version
• It will install an init script for the asterniclog parser daemon
• It will install the perl-Time-HiRes module
• It will create a MySQL database and tables named 'qstats'
Integrate Asterinic Call Center with Yeastar S series

- It will install both the parselog tool in /usr/local/parselog and the web application in /var/www/html/stats

Be sure to change owner of html directory to the same user your web server is running as. That user might change depending on your linux distribution of choice.
For Centos distribution, it could be the user 'apache'

```
chown -R apache:apache /var/www/html/stats/
```

For Ubuntu distribution it could be the user 'www-data'
```
```

Since Centos has a little issue about the privilege, we need to disable selinux.
```
vi /etc/selinux/config
```

Change the content like this.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.

#SELINUX=enforcing
SELINUXTYPE=targeted
SELINUXTYPE=targeted
```

Save it and reboot the system to take effect.

As this is a standalone Linux box with no Asterisk installation in it, you will have to create the asterisk log directory by hand like this:

```
mkdir /var/log/asterisk
```

That's it. You completed the Asternic server side of things. Now you can login the asterinic login page by http://your_server_IP_address/stats
Configuration of S series PBX

You will have to perform some configuration on the Yeastar S series PBX, some directly via command line, login into it via SSH, and some via the web interface:

Enable the queue log and AMI used by Asternic

Login web interface of Yeastar S PBX, enable SSH and AMI, you can set the user name and password as your wish, it’s used for AMI connection. You need to input Linux server’s IP or IP range to permitted list like this. When done, save it and apply changes on web.
Integrate Asterinic Call Center with Yeastar S series

Login PBX through SSH using default port 8022.
Username: support
Password: iyeastar

vi /ysdisk/support/customcfg/logger_custom.conf
Integrate Asterin Call Center with Yeastar S series

[general]
queue_log=yes

Save it then do apply changes on web to take effect.

If you have calls arrive in queue, you will find the queue_log in folder /ysdisk/syslog

You can also check the content inside directly.
Create SSH key

We will use a neat trick to replicate the queue_log from the Yeastar S PBX into our Linux server. For that we will use passwordless SSH. So again, inside the Yeastar S PBX run these commands to create an SSH key:

```bash
mkdir /ysdisk/support/tmp/.ssh
chmod 0700 /ysdisk/support/tmp/.ssh
dropbearkey -t rsa -f /ysdisk/support/tmp/.ssh/id_rsa
chmod 0600 /ysdisk/support/tmp/.ssh/id_rsa
```

Now we must retrieve the public identity just created, use this command to print it out on screen:

```bash
dropbearkey -y -f /ysdisk/support/tmp/.ssh/id_rsa | grep "^ssh-rsa"
```

You will have to copy the output and then paste it inside the file /root/.ssh/authorized_keys on your Linux server. So be sure to open that file in the Linux server and paste the contents produced by the above command into it.

```bash
mkdir /root/.ssh
chmod 0700 /root/.ssh
vi authorized_keys
```

Once that's done, you can test if the connection works by running this command
Integrate Asterin Call Center with Yeastar S series

inside the Yeastar S PBX (in this example 192.168.9.69 is Linux server IP)

```bash
ssh -y -i /ysdisk/support/tmp/.ssh/id_rsa root@192.168.9.69
```

That should connect you to the Linux box with no password prompts. If that is ok be sure to close that session by typing "exit" to return to the Yeastar CLI. As for the error inside the screen shot, just ignore that.

**Start the Log replication on Yeastar**

We will use the extensibility options the Yeastar S PBX provides via custom scripts. First, we create a script to startup a new command when the box boots up, named /ysdisk/support/add.sh:

```bash
vi /ysdisk/support/add.sh
```

Write this content in that file, via vi insertion mode (i)

```bash
#!/bin/sh
chmod +x /ysdisk/support/tmp/asterniclog
/ysdisk/support/tmp/asterniclog &
```

Save the file and exit (:wq)

Now we will create a script to actually feed log updates on the remote server by creating a script /ysdisk/support/tmp/asterniclog:
Integrate Asterin Call Center with Yeastar S series

```
vi /ysdisk/support/tmp/asterniclog
```

With this content, be sure to set vi in insert mode (i)

```
#!/bin/sh
sleep 60
tail -f /ysdisk/syslog/queue_log | ssh -y -i /ysdisk/support/tmp/.ssh/id_rsa root@192.168.9.69 "cat >>/var/log/asterisk/queue_log" &
```

Save the file and exit (:wq)

Finally, you will have to reboot the Yeastar S PBX for all the changes to take effect. When done, the queue_log will be synced to linux server every 60 seconds.

You can check the log in linux server to check the queue_log as well.

```
[root@localhost ]# ls /var/log/asterisk -al
total 6
drwxr-xr-x. 2 root root 22 Nov 6 02:16 .
drwxr-xr-x. 23 root root 4096 Nov 5 23:59 ..
-rw-r--r--. 1 root root 2563 Nov 6 00:35 queue_log
```

You can also read the content to confirm that.
Manager real-time view

In order for Asterinic real-time view to work, we have enable the AMI and allow the Linux server IP to access. Then we need to edit `/var/www/html/stats/config.php` and set the correct manager user, secret and host.

```
vi /var/www/html/stats/config.php
```

Change the content like this. Save it, then restart the Asterinic

```
[root@localhost /]# service asteriniclog restart
Restarting asteriniclog (via systemctl): [ OK  ]
```
Enjoy the perfect call center solution

Login asternic’s web interface by  http://your_server_IP/stats
You can apply for trial license for 30 days. Then you can login using default username and password.
Default username: admin
Default password: admin

Choose the queue and agent number you want, then click Display Report

The example data will show it like this for detailed check.
If there is incoming call arrived, the agent status will change from Not in use to Ringing and Busy finally.
You can also do some operation through this web page. Find more surprise by install one for your server as well.

<Finish>