



**Turning Asterisk@Home into a small call-center  
with QueueMetrics**

AUTHOR:	LOWAY RESEARCH
VERSION:	1.0
DATE:	MAY 4, 2006

---

## Index

Turning AAH into a call-center solution .....	3
Call centers 101: the very basics .....	3
Prerequisites .....	4
Installing QueueMetrics .....	4
Setting up AAH .....	8
Creating agents.....	9
Configuring a queue .....	10
What if I change the queue definition in AMP?.....	13
Do I have to create the queue with AMP in any case? .....	13
Allowing agents to log in and out as needed .....	13
Why don't we use the standard AAH logins?.....	15
Logging in .....	15
Configuring the agents and the queue in QueueMetrics.....	16

## Turning AAH into a call-center solution

If you are reading this, you probably already know that using Asterisk@Home (AAH) you can install a fully featured Asterisk-based PBX in a matter of minutes. This PBX is very good for most users as it is preconfigured to handle the most common scenarios one can find in a personal environment or in a small to medium sized office.

The typical usage patterns of a small call center are a bit different from the ones you are likely to find in a classical PBX, because:

- Agents spend almost their whole working day available to answer the phone (as opposed to standard PBX users who use the phone occasionally while doing other work); handling and answering calls for them has to be made as easy as possible
- A call center is usually a high-density commercial enterprise; therefore it has to be run and monitored using tools that are able to quickly see how things are going, identify bottlenecks and address them.

In our opinion, running a successful call center is more a state of mind than a given set of telephone gear. What makes a difference is not the number of extensions, queues or agents you have; is a mindset where you consider that the customers calling in are *actually very important* and you do your best to serve them well within given budget limits.

Maybe you just run a small computer-repair shop and have a couple of lines coming in. What is the cost of having people wait or call multiple times because they cannot talk to anybody within a reasonable time? What will your clients think about you? On the contrary, what will your clients think of your customer service if it always answers on the very first ring? And how do you know if your technicians actually answer the phone when the calls come in or wait five minutes because they are doing other things? These are the questions you should ask. If you follow the guidelines in this document, you'll find an easy way to start answering to questions like these.

Running a call-center, therefore, is not a matter of having multiple PRIs or special hardware. You will not even need a separate box from your main PBX running AAH. You will only need some software and a bit of configuration to set it up correctly.

### ***Call centers 101: the very basics***

Before we start building a small call center, we have to focus a bit on the terminology:

- A *campaign* is a set of calls that belong to the same scope, e.g. your technical support versus commercial support line are different campaigns, though they may be
- An *inbound* campaign is devoted to answering people calling in, while an *outbound* campaign is made up by agents dialing out. Call centers often mix inbound and outbound activities in order to optimize the use of personnel.

- A *queue* is the physical implementation of an inbound campaign. The queue receives calls and pipes them to the available agents according to a predefined logic (usually, FIFO for the calls and round-robin for the agents). In call center terminology, this functionality is often referred to as the ACD (Automated Call Distribution).
- An *agent* is a person working at a call center. The agent is different from a casual user as an agent logs in and out, in order to tell the system when he is available or not. In this way, the ACD searching logic minimizes agent searching time, as it almost never has to ring up an agent who is not available. An agent can be working on one or more queues: whenever he is available, all calls coming in to any queue he's working on will be piped to him.

In this tutorial, we will learn how to create an inbound queue and proper agents to handle it.

### ***Prerequisites***

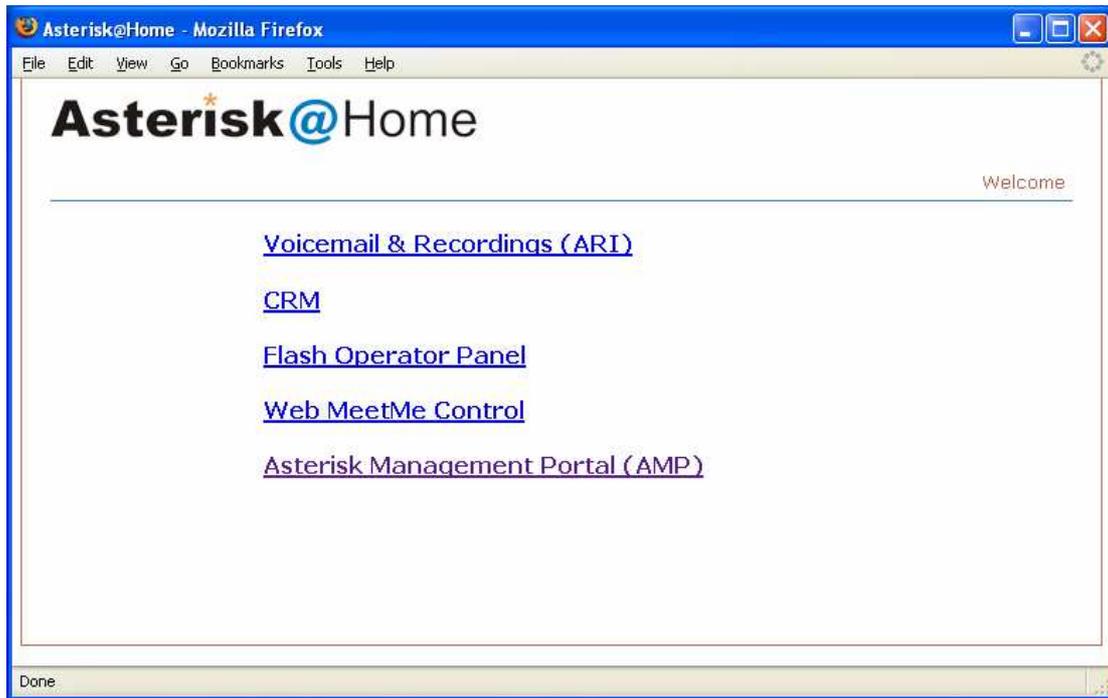
To follow this guide, you will need an already-installed, reasonably modern AAH instance. It may be your home or office PBX. You can follow this guide completely while having your PBX running, so there is no need for a prolonged downtime. You will need at least a couple of telephones to test your setup, and land line you can use.

You should be basically familiar on how to use AAH as a basic PBX: creating extensions, connecting to external lines and such things.

## **Installing QueueMetrics**

QueueMetrics is a full-fledged call center monitoring solution – see <http://queuemetrics.loway.it> . It is a commercial product that is available free of charge to smaller call centers, home users and SOHO's.

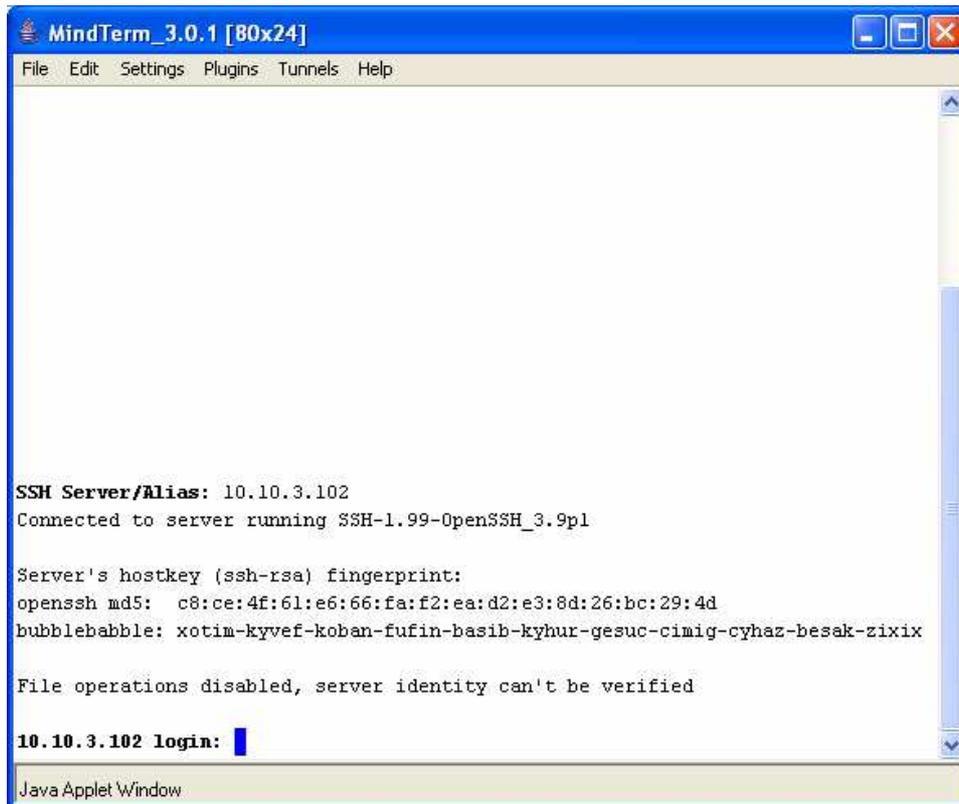
Log in to your AAH system using your web browser and the standard credentials (they are login: "maint" password: "password" if you have not changed them manually).



Click on “Asterisk Management Portal” and then click “Maintenance”.



Click on “Java SSH” from the column on the left. A terminal window will open up (it may take a while).



Type the Return key when you see the line “SSH Server/Alias”. The supplied value should already be correct.

Then log in as “root” with password “password”.

When you are logged in, type the following commands:

```
wget -P /etc/yum.repos.d http://yum.loway.it/loway.repo
yum install queuemetrics
```

The *yum* command will download QueueMetrics and all of its dependences and install it on your system. This may take a while, depending on your internet connection speed. When asked to confirm the installation, type “y” to confirm.

After the installation is done, you have to install the sample MySQL database that will be used to initialize the system by executing the following commands:

```
cd /usr/local/queuemetrics/webapps/queuemetrics-1.1.1/WEB-INF/README
./installDb.sh
```

Follow the on-screen instructions (it is a matter of typing in a couple of passwords as detailed by the *installDb* utility itself) and the database will be created.

To test that everything is okay, you'll have to point your browser to the address <http://myserver:8080/queuemetrics> and you should see a screen like the following one.



If you see this screen, you know that QueueMetrics is working fine. As you'll be curious to check it out, you can login immediately with the credentials "demoadmin" password "demo".

You may notice that the first time you load up a new page, it may take a while to display it, depending on your AAH box CPU speed; this is normal, as QueueMetrics uses Java Server Pages technology and such pages are internally compiled the first time they are accessed.



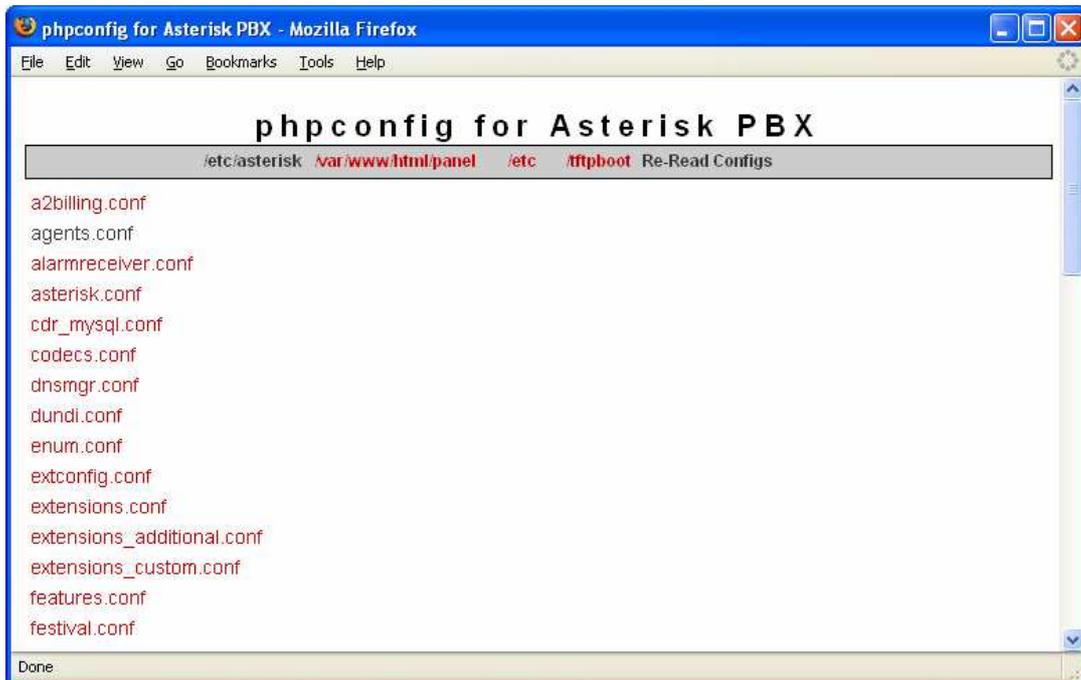
This is how the first page looks like. Feel free to look around a bit before logging off. You may now also close the SSH window as we won't need it anymore for this tutorial.

## Setting up AAH

What we need to do now is setting up AAH so that:

- We can have actual Asterisk agents, both in the *always-on* and *callback* fashion
- We can configure a queue to run agents

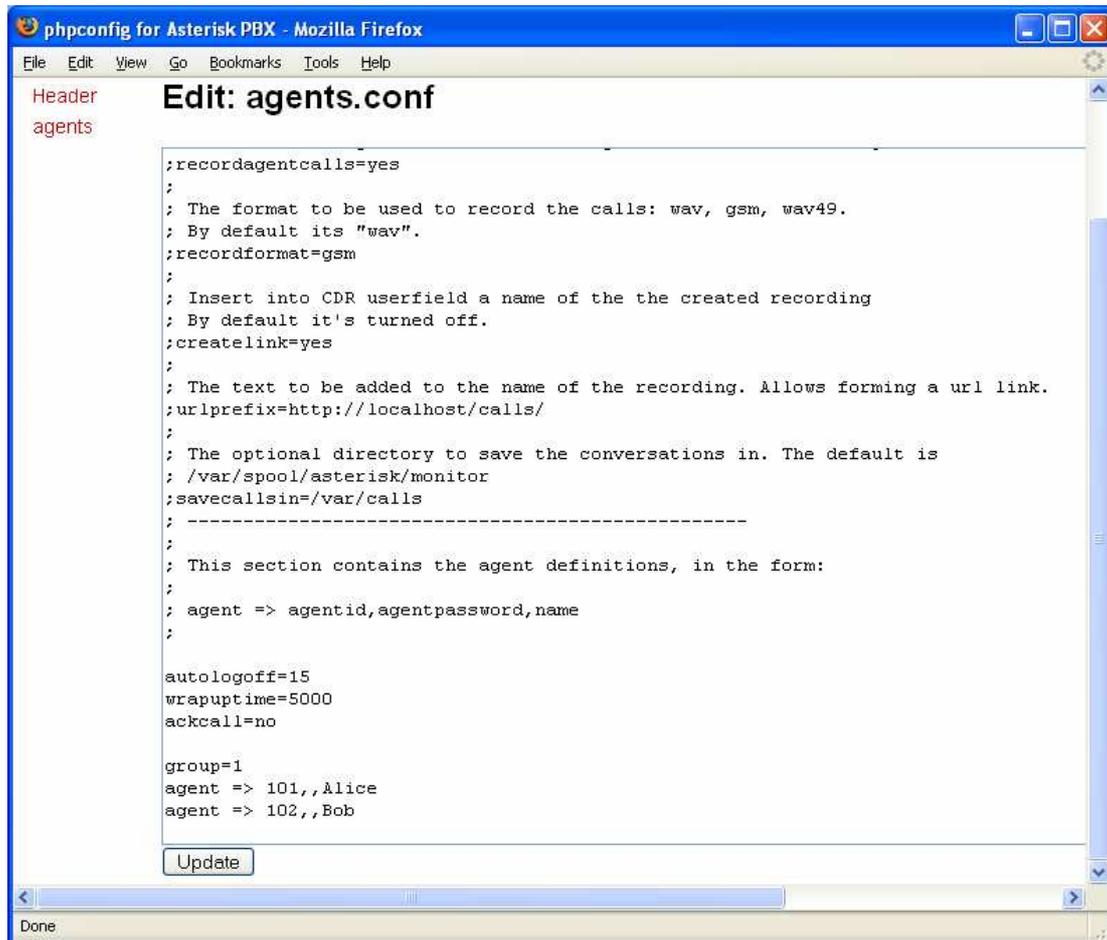
To perform this, we will have to edit Asterisk's configuration file. This is very easy, as under the maintenance menu we find the "Config edit" label; clicking on it will lead you to a screen like the following one:



You will be able to edit each file just by clicking on it and then pressing “Update” to save (do not forget to click on “Reload config” to have Asterisk load up the new configuration).

### ***Creating agents***

We’ll start by editing the *agents.conf* file; go to its bottom, it should look kind of similar to the one in the picture.



Now add the following lines:

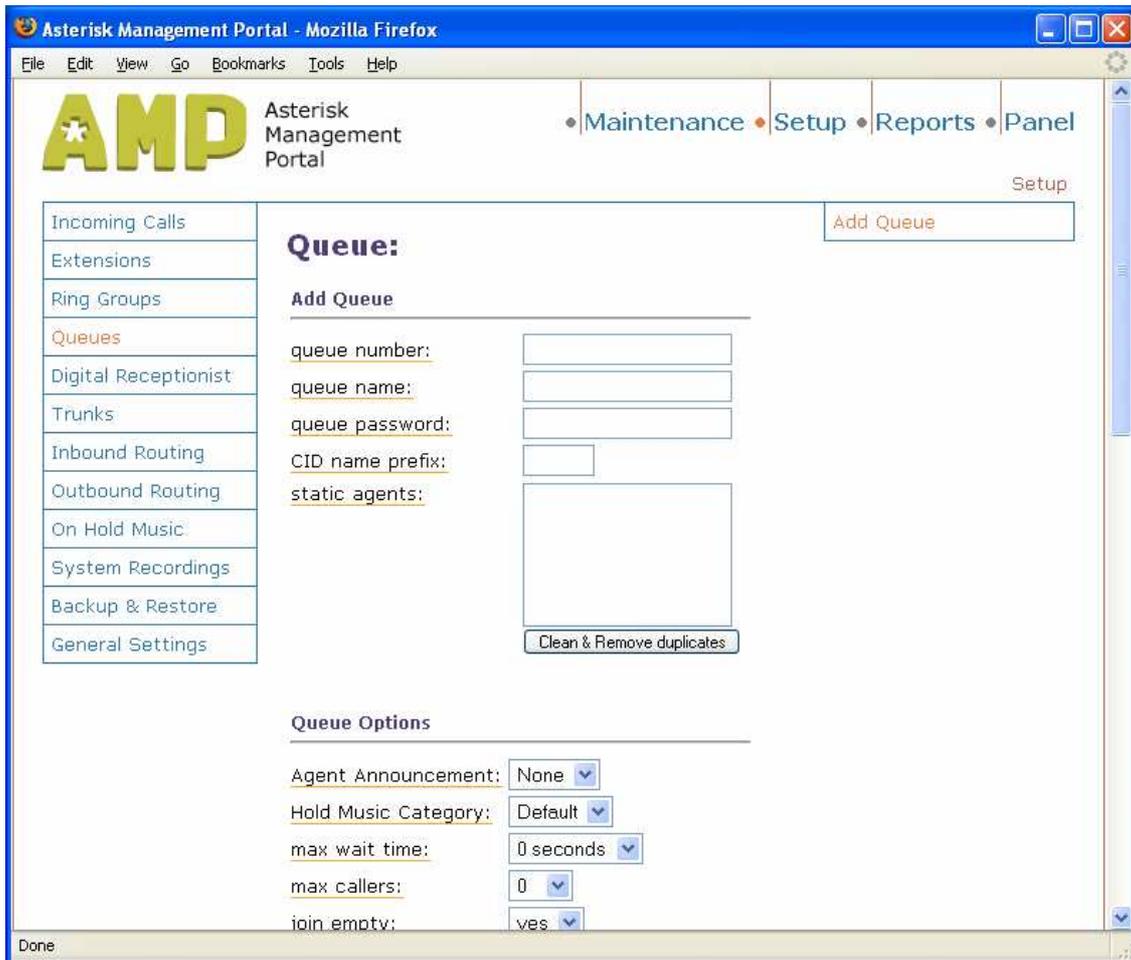
```
agent => 101,,Alice
agent => 102,,Bob
```

Make sure that you enter the double commas before the agent's name correctly – they are not a typo).

We have just successfully created a couple of agents; Agent/101 is Alice and Agent/102 is Bob.

### **Configuring a queue**

To add a queue to your system, from the main menu click on “Setup” and then “Queues”.



You will see that a queue is highly configurable; but for the moment we make ourselves content of entering the following data:

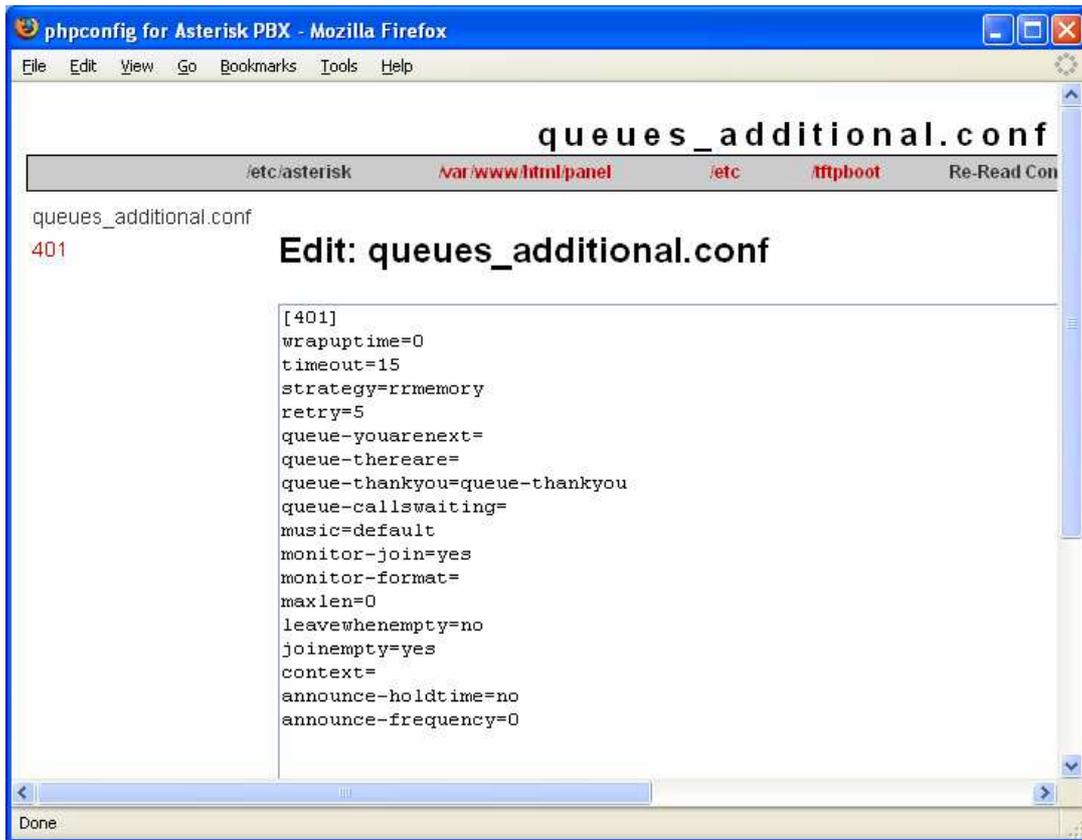
- Queue number: *401*
- Queue name: *Oper*
- Ring strategy: *rrmemory* (This is the basic call center setup where the ACD system tries one agent after the other until someone answers)

When you are done, click on the red line and have AAH update the system.

Now comes the trickiest part.

The current versions of AMP will not allow you to enter a list of “true” agents in the form Agent/101, but that is what you have to do.

To perform this, you use the editor to locate the queue definition you have just created that is written to the file *queues\_additional.conf*; it will look more or less like this:

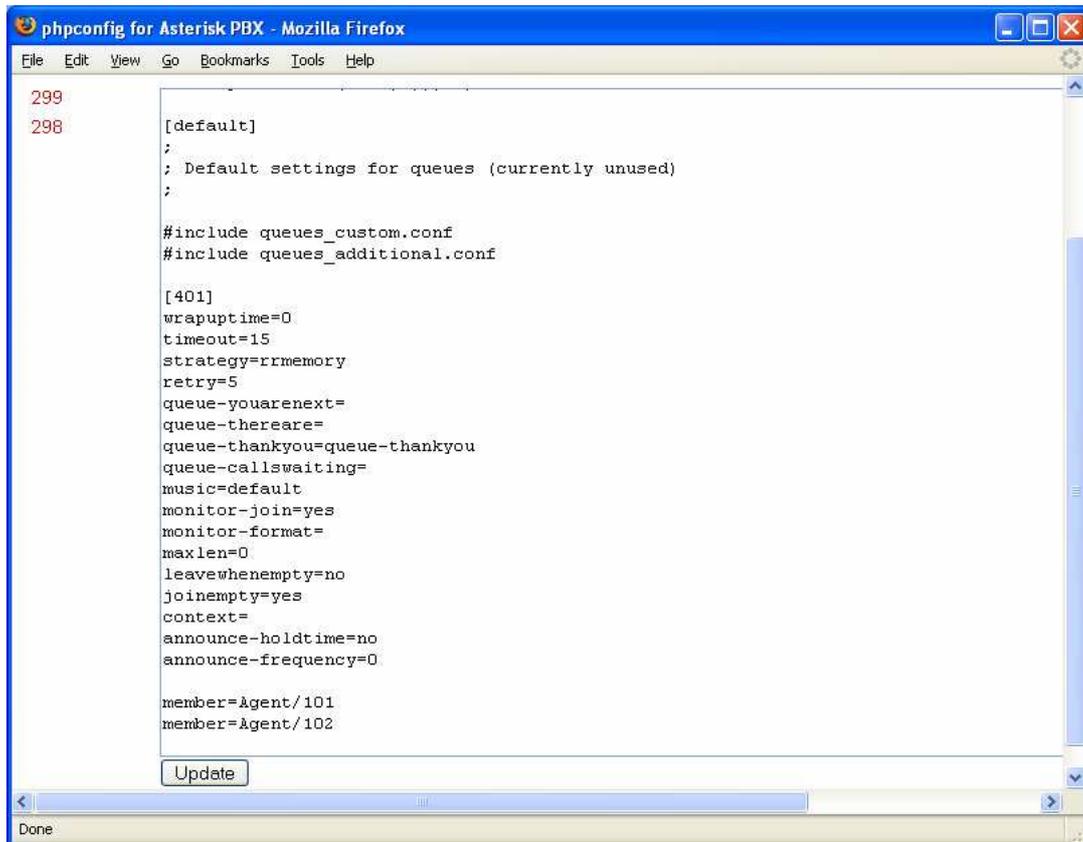


Now copy all the queue definition for queue 401 and paste into the file *queues.conf*, after all the other values. After that add two lines that read:

```

member=Agent/101
member=Agent/102
    
```

The result should look very similar to the following screenshot.



Now save this file and reload Asterisk. You have just created a new queue that has two members, Alice and Bob!

### What if I change the queue definition in AMP?

If you modify the queue through AMP, the results will be ignored by Asterisk until you copy and paste them back to the file *queues.conf* and you restart. That's because we actually redefine it in order to be useful to us.

### Do I have to create the queue with AMP in any case?

Yes, you do, because otherwise AMP will not know that the queue exists and will not route calls to it.

### ***Allowing agents to log in and out as needed***

What makes an agent different from a static terminal is that the agent logs in, i.e. tells Asterisk he is ready to handle calls, and then logs out when he is not ready anymore.

There are two kinds of agents in Asterisk: *always-on* agents and *callback* agents.

An *always-on agent* will sit all day with his headphones on, listening to music, until a call comes in. When he is available, the incoming call is routed to him immediately, he does not have to answer it. When he is done with the call, he presses “\*” to disconnect it and goes back to listening to music on hold. When he needs to take a break, he hangs up the phone; this way Asterisk knows he is not available.

A *callback agent*, instead, will simply notify Asterisk that he is available at a given extension. When a call comes in for him, Asterisk will ring his own extension. He will handle the call normally, and then will have to tell Asterisk when he is not available. To do this, he will have to perform an explicit logoff.

The first scenario is only suited for high-volume call handling, while the second is usually better suited to smaller installations.

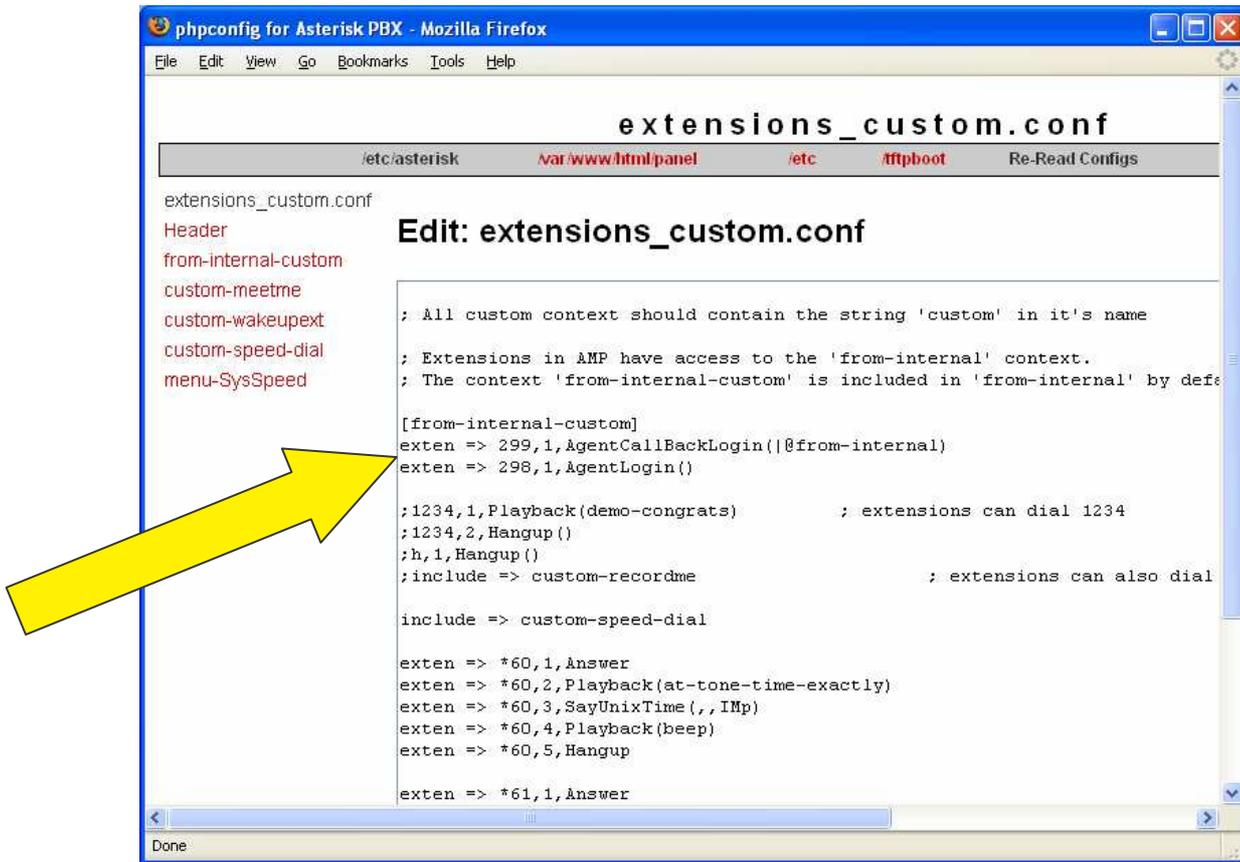
To choose whether your agents will be callback or not, you just log them in by making them dial a different extension. This means that the very same agent can be always-on on a high-volume Monday morning, and callback in the afternoon, with no change in the configuration.

So we now have to create a couple of extensions that will allow agents to log on and off in both ways. You have to find a couple of unused extensions that will allow the log-in. Let's say they are 299 and 298 on our test system.

We modify the file *extensions\_custom.conf*, by adding a couple of lines after the [from-internal-custom] section:

```
exten => 299,1,AgentCallbackLogin(|@from-internal)
exten => 298,1,AgentLogin()
```

The result should look more or less like the following screenshot:



Save and reload. AAH is now configured fully.

### Why don't we use the standard AAH logins?

Instead of actual agents, AAH usually uses either static extensions or fully dynamic callback agents, where you log in to a queue by entering the queue number followed by \* and log off by entering a queue number followed by \*\*.

This works, but for the way AAH is configured, the logging in and logging off are not traced correctly; this way it is impossible to gather data on agent sessions, i.e. how much time the agent was available. Such information is essential to the smooth running of a call center, because it lets us compare our actual staff versus the flow of calls at any given moment in time.

Static extensions, instead, should not be used, as they are never aware of the agent's availability, and therefore may end up in empty extensions ringing while other agents are sitting idle waiting for the ACD to pass over the call.

### Logging in

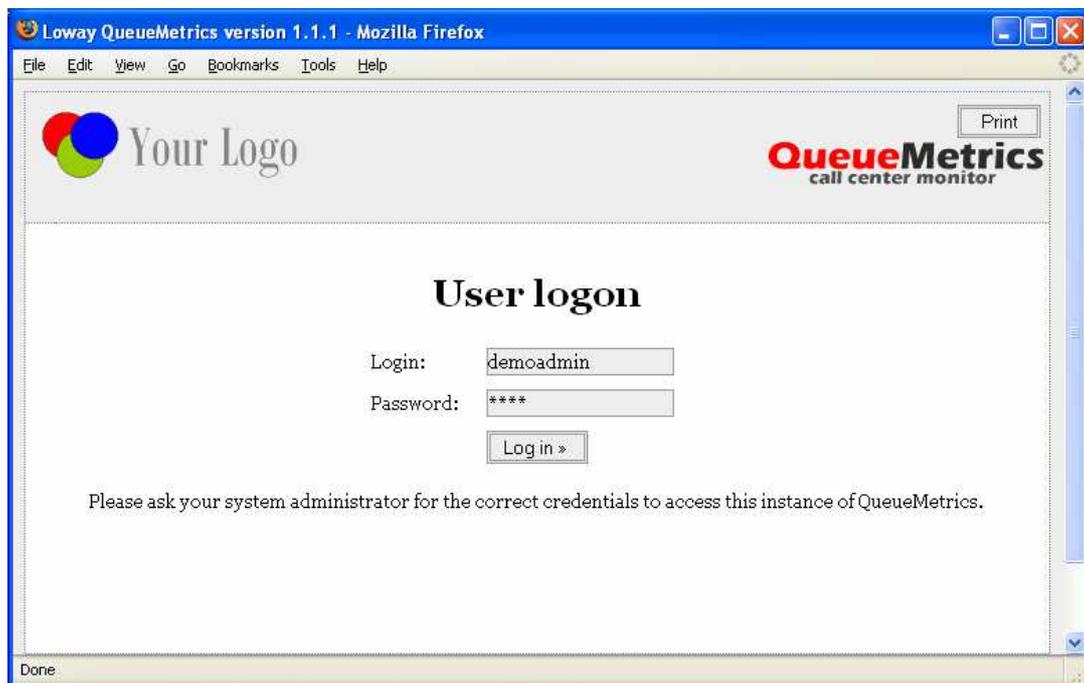
You may have your agents log in by dialing 299 for the callback or 298 for the always-on mode.

When you log in as a callback agent, you have to enter your agent number (like, 101 for Alice) and then your current extension (like, 243). Each number has to be followed by pound (#).

To logoff a callback agent, you dial 299, then enter your agent code followed by pound, and then press pound again. The voice will tell you that you are logged off.

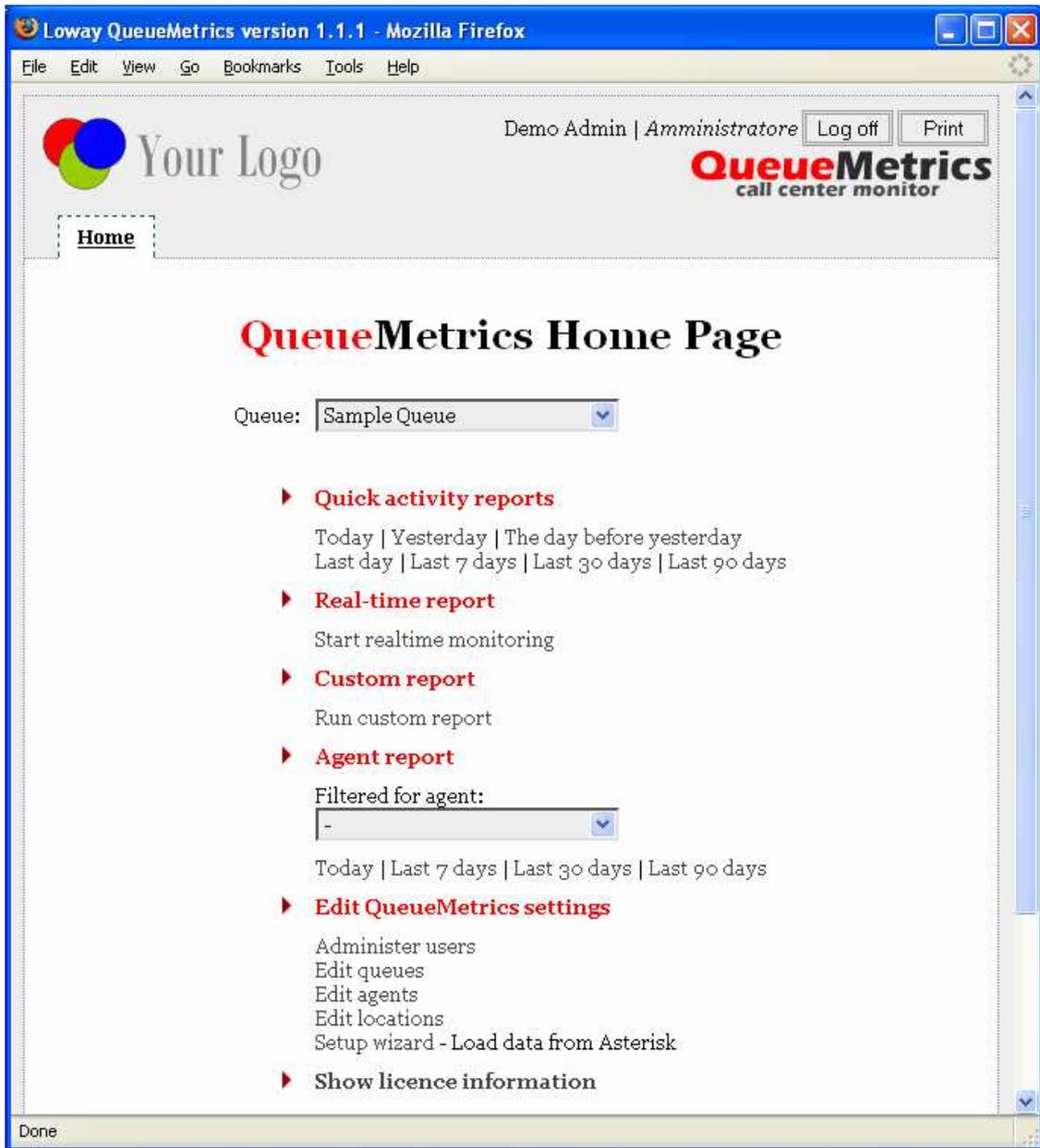
## Configuring the agents and the queue in QueueMetrics

Go back to QueueMetrics.



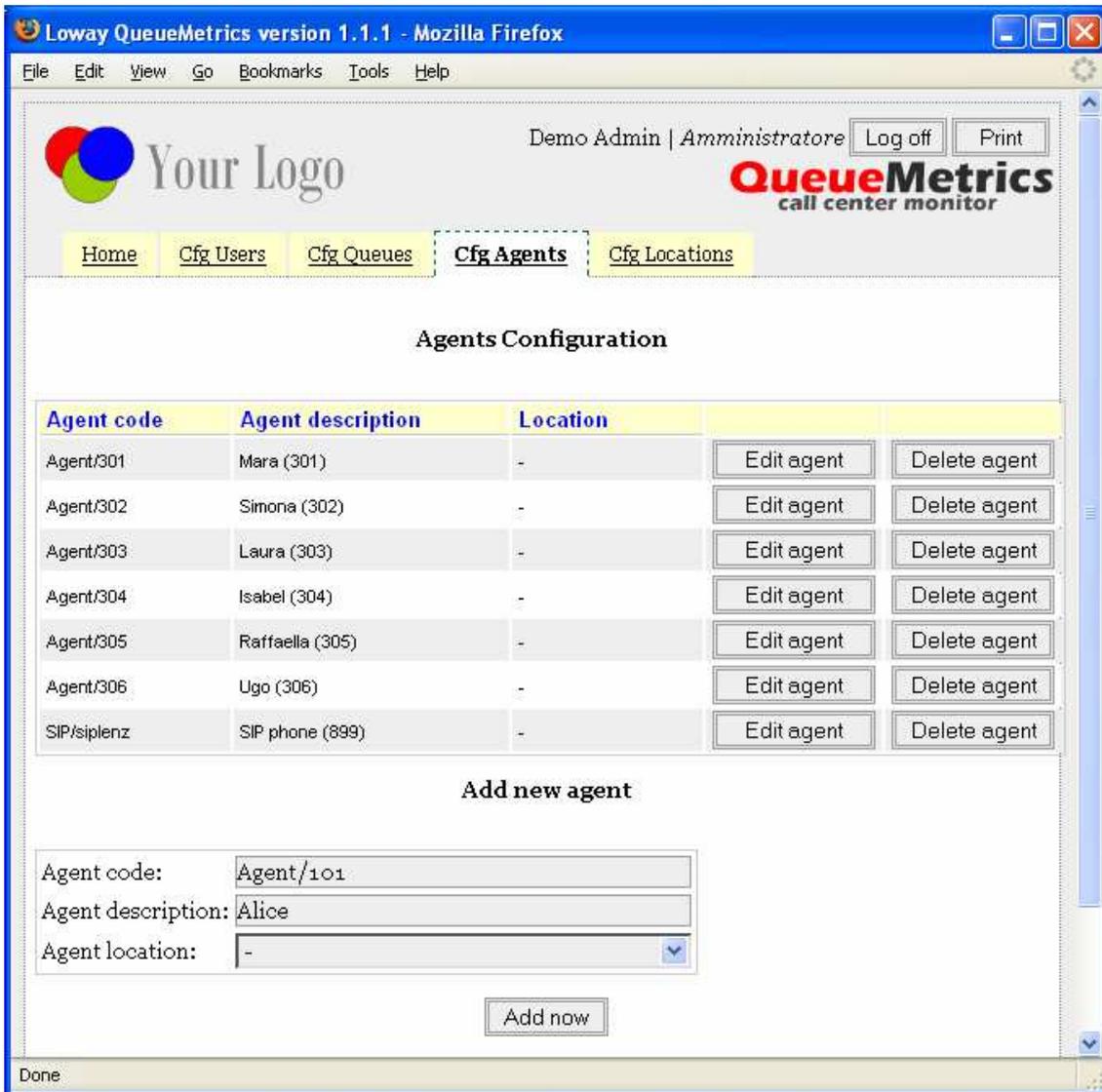
Login as “demoadmin” password “demo”.

You should see a screen like the following one:



First of all, click on “Edit agents”

Add your new two agents, by entering the agent code as “Agent/101” and the description as “Alice”. Delete existing agents you do not need – a number of agents are preloaded when the system starts up, but we may safely delete them.



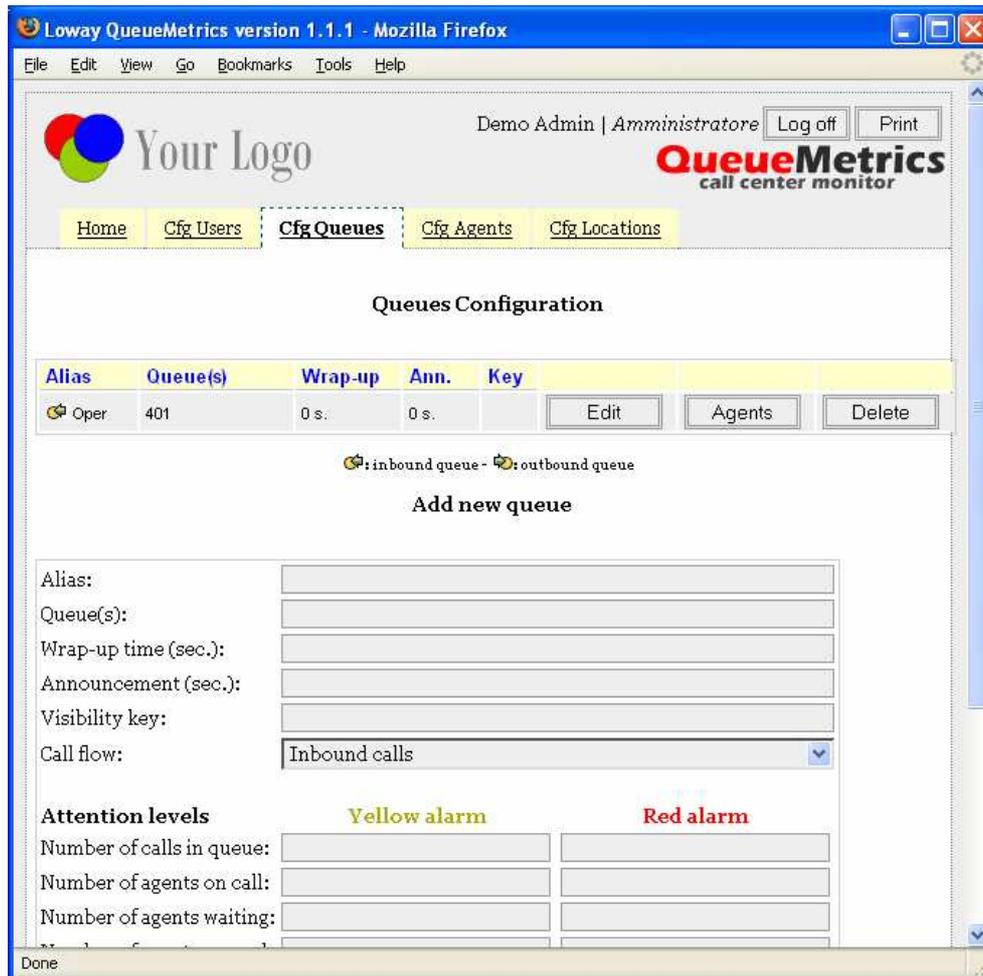
Now we add a new queue.

First of all, click on the “Cfg queues” label and delete the existing queues that were preloaded in the sample database.

Now create a new queue by filling in the form at the bottom of the page.

- Enter “Oper” as the queue Alias.
- Enter “401” in the “Queues” field.

The result should be similar to the following:

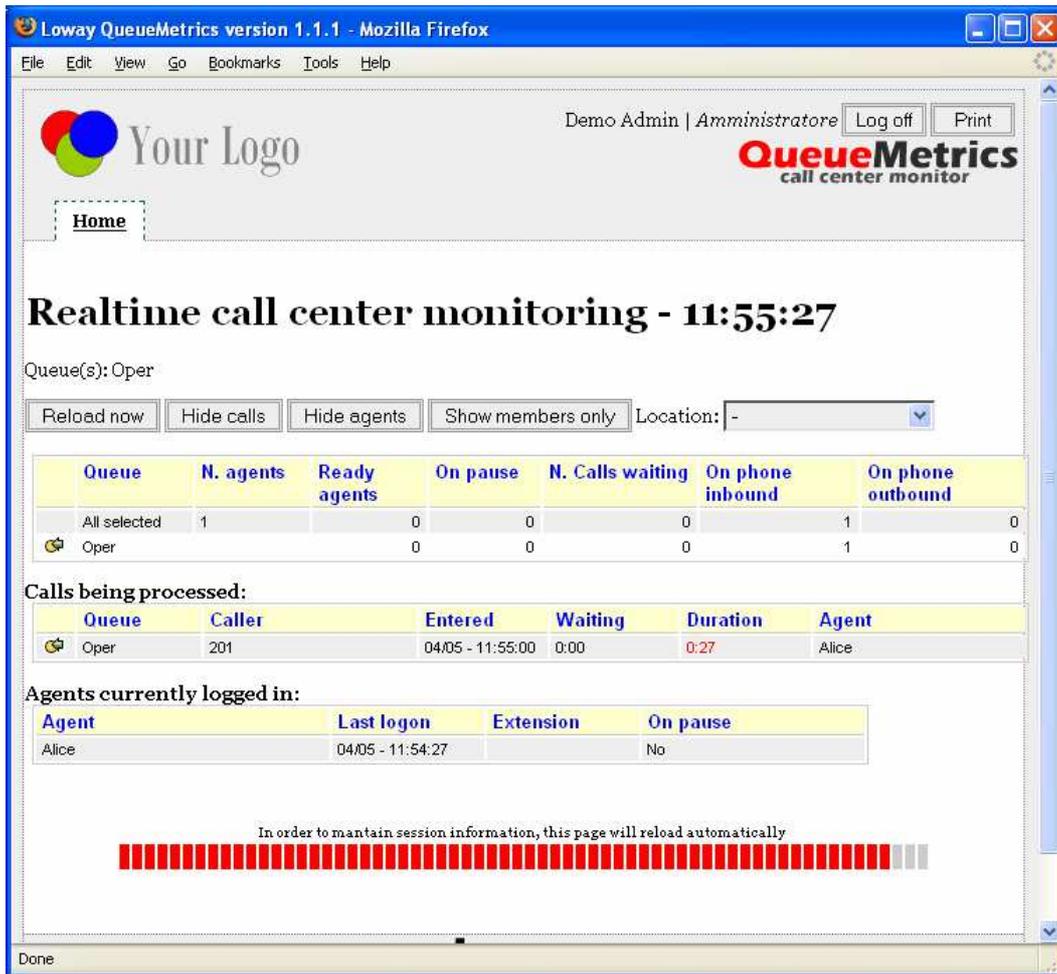


As you can see, you may set up yellow and red alarms on each queue, so that in the real-time monitoring you can have lines turn red or yellow when specific events happen (please consult the user manual for a thorough explanation of how to set this up).

Go back to the home page.

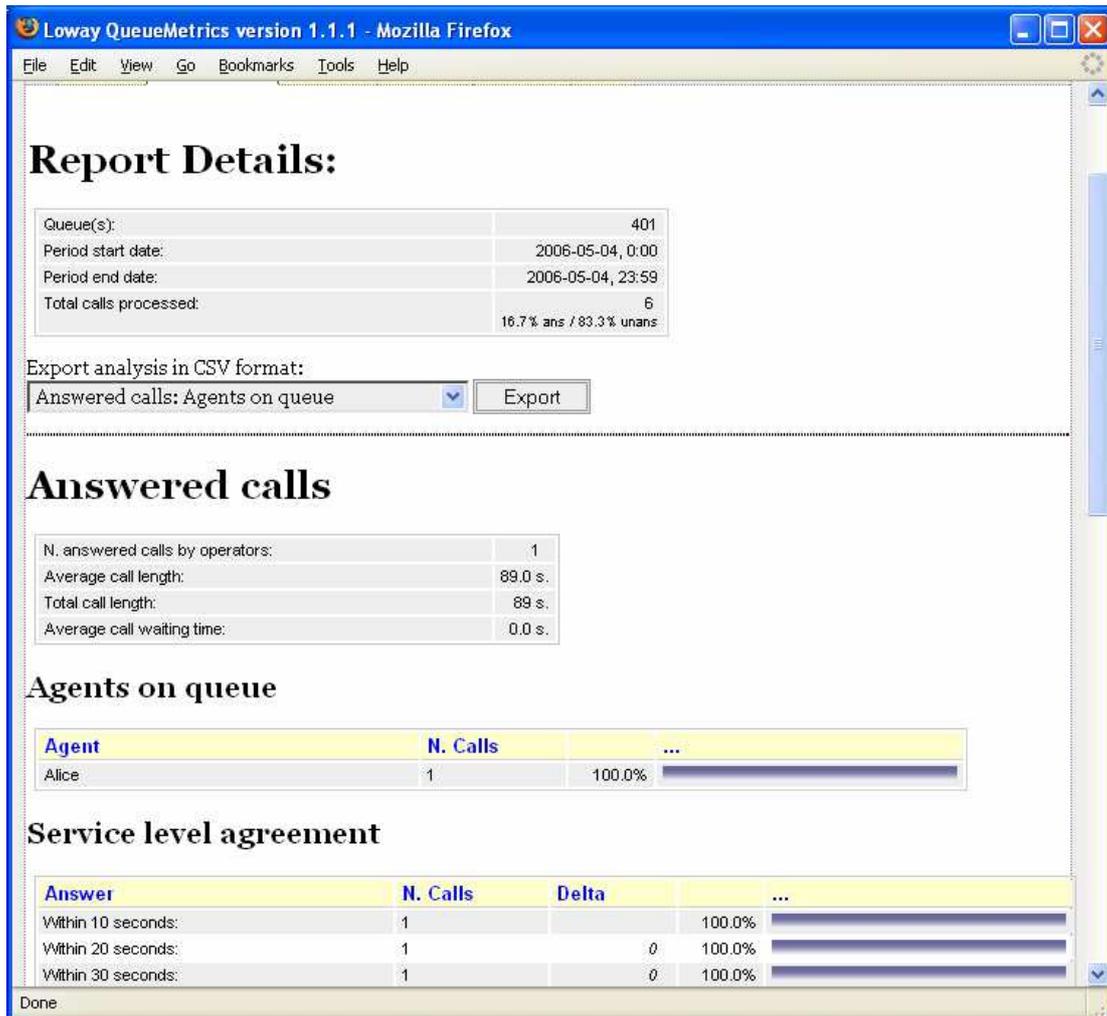
Have an agent log in from one telephone, by dialing 299 and then their agent number. Place a call to the queue (dial 401 from another extension, for example).

Click on the “Realtime reporting” and you should see a screenshot very similar to this one:



Here we can see that there is only one agent logged in as an always-on, and one call from 201 is ongoing with Alice.

Once the call is over. if we click on the “Home” tab and then “Quick activity reports” -> “Today”, we will be shown today’s stats



From the report page, you can access a number of reports; they are well detailed in QueueMetrics user manual, but the following should be a working introduction:

- The “Answered” tab gives statistics about the calls that were answered correctly: how any were handled, who handled them, average duration and average wait times
- The “Detail” button on the bottom of the “Answered” page gives a call-by-call activity detail
- The “Unanswered” tab gives information on calls that were lost, i.e. the user disconnected before talking to a living person
- The “Detail” button on the bottom of the “Unanswered” page gives a call-by-call detail of lost calls
- The “Distrib” tab gives a daily, hourly and day-of-week breakdown of taken and lost calls

- The “Agents” tab gives detail of agent sessions and calls handled during those sessions
- The “Detail” button on the bottom of the “Agents” page gives a session-by-session detail of agent activity
- The “All” tab is a convenient way to have all the stats on a single page so that it can be saved or printed for future reference.