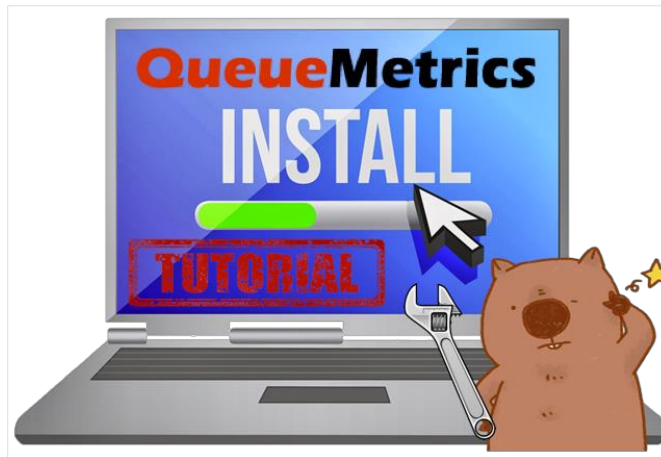


Установка QueueMetrics на Systemd (Debian/Ubuntu)



Руководство по установке Uniloader QueueMetrics на системе Debian/Ubuntu

На операционной системе **Systemd**, такой как **Debian** или **Ubuntu**, нужно установить **Uniloader** вручную. Это может показаться немного сложным, но следуя данному руководству будет легко это сделать.

Что такое QueueMetrics?

QueueMetrics – это масштабируемый набор инструментов для мониторинга и отчетности, который удовлетворяет потребности тысяч контакт-центров по всему миру, и предлагает широкий спектр интегрированных функций, таких, как мониторинг производительности агентов, контроль целевых показателей, отслеживание конверсий, анализ статистики кампании в реальном времени, и простой в использовании интерфейс. Он доступен как на отдельном сервере, так и в качестве облачного сервиса.

Что такое Uniloader?

Uniloader – программа, устанавливаемая на Asterisk PBX. Она выгружает данные в локальный или удаленный экземпляр QueueMetrics, и принимает задания для выполнения на локальной АТС.

Uniloader разворачивается из единственного двоичного файла, который должен быть установлен на самой АТС. Он разработан как очень легкое приложение, так что он может работать без негативного влияния даже на низкопроизводительной оборудовании; и он является очень безопасным, поэтому данные не будут потеряны даже в тех случаях, когда удаленный сервер QueueMetrics становится недоступным.

В процессе работы он выгружает данные, используя либо протокол HTTP/S, либо протокол MySQL (в зависимости от того, куда идет выгрузка (back-end)). HTTP/S предназначен для работы с удаленными экземплярами QueueMetrics, особенно с QueueMetrics Live (смотрите <http://queuemetrics-live.com> для более полной информации), тогда как MySQL предназначен для локальных систем; любой способ подходит, если QueueMetrics размещен на той же самой машине.

При работе через HTTP, если сервер QueueMetrics не имеет прямого соединения с АТС, Uniloader имеет возможность работать как прокси-сервер, и будет получать задания для выполнения на сервере Asterisk через AMI (Asterisk

Manager Interface). Этот способ позволяет удаленно выполнять QueueMetrics, и по-прежнему использовать возможность подключать и отключать агентов, приостанавливать их, прослушивать вызовы через ChanSpy и т.д.

Uniloader также используется для выполнения других административных/дополнительных задач, которые выполняют полезные функции в системе Asterisk, подключенной к QueueMetrics; например, может генерировать события «музыка на удержание» (music-on-hold) в очереди, и может помочь диагностировать проблемы.

Установка вручную на Systemd (CentOS, Debian/Ubuntu)

Uniloader можно загрузить с сайта: <https://www.queuemetrics-live.com/uniloader.jsp>

Вначале войдите в систему как root, затем загрузите пакет с расширением TGZ, распакуйте его в каталог `/opt`, и создайте ссылку, позволяющую вызвать команду из любого места.

```
cd /opt

wget http://downloads.loway.ch/qm/uniloader-0.4.1.tar.gz

tar zxvf uniloader-0.4.1.tar.gz

cd /usr/bin/

ln -s /opt/uniloader-0.4.1/bin/uniloader_amd64 uniloader
```

Теперь, если запустить Uniloader из командной оболочки shell, получим следующее сообщение по умолчанию.

Tip:

All files in the examples below are available within the Uniloader package, under `init-scripts/systemd`, so you can just copy them to the right location.

Uniloader

Создайте файл конфигурации `/etc/uniloader` и установите свои значения (отредактируйте в соответствии с установленной системой):

```
QUEUELOG=/var/log/asterisk/queue_log

# On-premise QueueMetrics instance
URI="mysql:tcp(127.0.0.1:3306)/queuemetrics?allowOldPasswords=1"
LOGIN=queuemetrics
PASS=javadude
TOKEN=P001

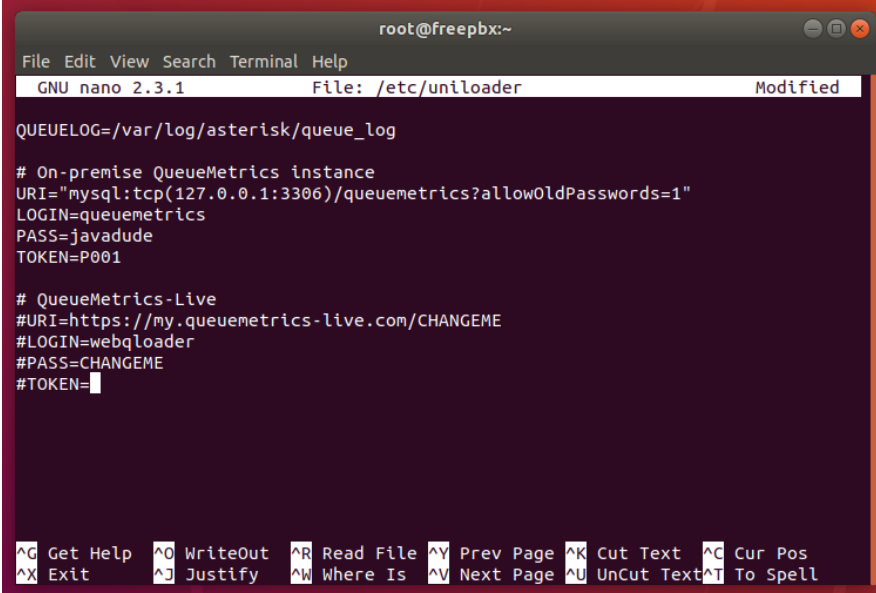
# QueueMetrics-Live
```

```
#URI=https://my.queuemetrics-live.com/CHANGEME
```

```
#LOGIN=webqloader
```

```
#PASS=CHANGEME
```

```
#TOKEN=
```



```
root@freepbx:~  
File Edit View Search Terminal Help  
GNU nano 2.3.1 File: /etc/uniloader Modified  
QUEUELOG=/var/log/asterisk/queue_log  
  
# On-premise QueueMetrics instance  
URI="mysql:tcp(127.0.0.1:3306)/queuemetrics?allowOldPasswords=1"  
LOGIN=queuemetrics  
PASS=javadude  
TOKEN=P001  
  
# QueueMetrics-Live  
#URI=https://my.queuemetrics-live.com/CHANGEME  
#LOGIN=webqloader  
#PASS=CHANGEME  
#TOKEN=
```

Теперь создайте unit файл `/lib/systemd/system/uniloader.service` :

```
[Unit]
```

```
Description=Loway Uniloader
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=simple
```

```
EnvironmentFile=/etc/uniloader
```

```
Environment=UPASSWD=${PASS}
```

```
Nice=15
```

```
KillMode=process
```

```
PIDFile=/var/run/uniloader.pid
```

```
ExecStart=/usr/bin/uniloader --src "${QUEUELOG}" \
```

```
upload --uri "${URI}" --login "${LOGIN}" --token "${TOKEN}" \
```

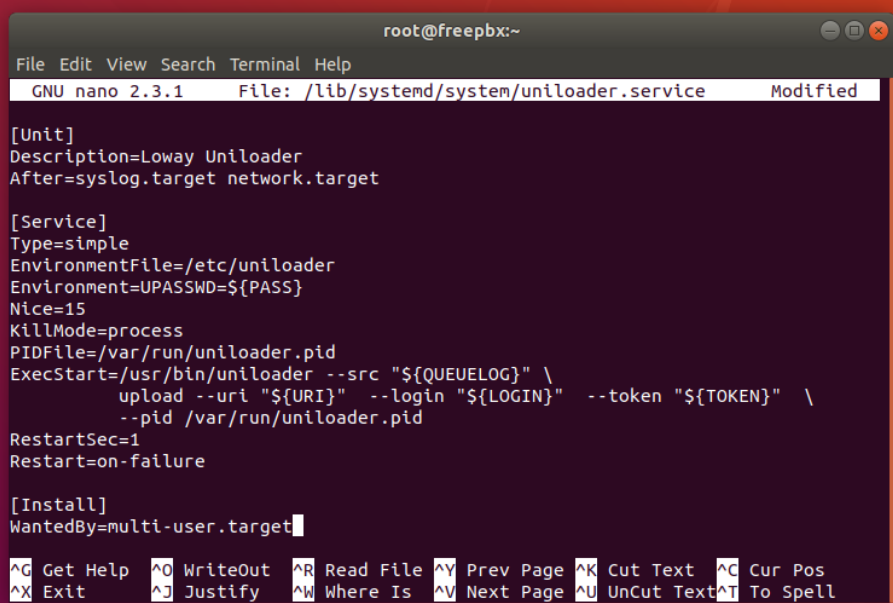
```
--pid /var/run/uniloader.pid
```

```
RestartSec=1
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=multi-user.target
```



```
root@freepbx:~
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /lib/systemd/system/uniloader.service Modified

[Unit]
Description=Loway Uniloader
After=syslog.target network.target

[Service]
Type=simple
EnvironmentFile=/etc/uniloader
Environment=UPASSWD=${PASS}
Nice=15
KillMode=process
PIDFile=/var/run/uniloader.pid
ExecStart=/usr/bin/uniloader --src "${QUEUELOG}" \
        upload --uri "${URI}" --login "${LOGIN}" --token "${TOKEN}" \
        --pid /var/run/uniloader.pid
RestartSec=1
Restart=on-failure

[Install]
WantedBy=multi-user.target

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Unitracker

Если также требуется служба трекинга, создайте файл конфигурации `/etc/unitracker` и установите свои значения (отредактируйте в соответствии с установленной системой):

```
AMIHOST=127.0.0.1
```

```
AMIPORT=5038
```

```
AMIUSER=admin
```

```
AMISECRET=amp123
```

```
#Uncomment to enable event logging
```

```
#DEBUGFILE=/root/unitracker_events.log
```

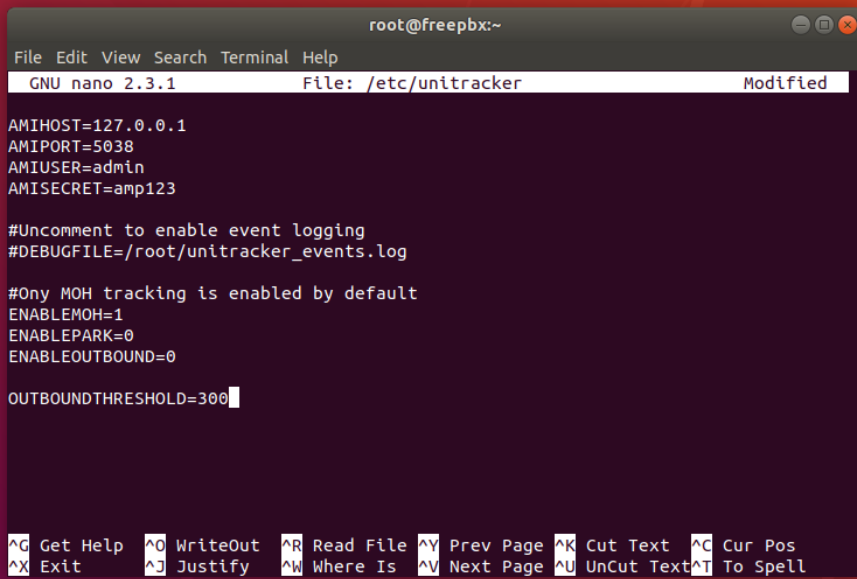
```
#Only MOH tracking is enabled by default
```

```
ENABLEMOH=1
```

ENABLEPARK=0

ENABLEOUTBOUND=0

OUTBOUNDTHRESHOLD=300



```
root@freepbx:~
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /etc/unittracker Modified
AMIHOST=127.0.0.1
AMIPORT=5038
AMIUSER=admin
AMISECRET=amp123
#Uncomment to enable event logging
#DEBUGFILE=/root/unittracker_events.log
#Only MOH tracking is enabled by default
ENABLEMOH=1
ENABLEPARK=0
ENABLEOUTBOUND=0
OUTBOUNDTHRESHOLD=300
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Теперь создайте unit файл `/lib/systemd/system/unittracker.service` :

[Unit]

Description=Loway Unitracker (Uniloader)

After=syslog.target network.target

[Service]

Type=simple

EnvironmentFile=/etc/unittracker

Environment=AMISECRET=\${AMISECRET}

Nice=15

KillMode=process

PIDFile=/var/run/unittracker.pid

ExecStart=/usr/bin/uniloader track --host "\${AMIHOST}" --port "\${AMIPORT}" --login "\${AMIUSER}" \

--debugfile "\${DEBUGFILE}" \

```
--moh "${ENABLEMOH}" \
```

```
--parkedcalls "${ENABLEPARK}" \
```

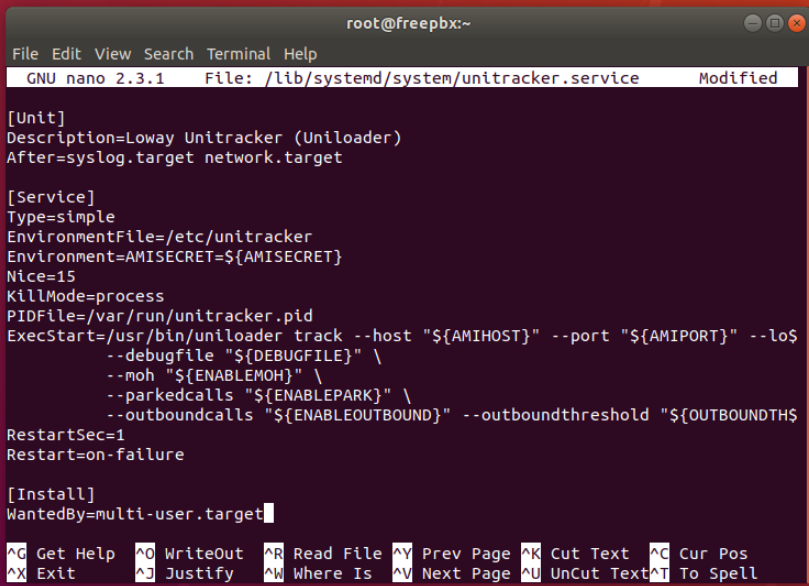
```
--outboundcalls "${ENABLEOUTBOUND}" --outboundthreshold "${OUTBOUNDTHRESHOLD}"
```

```
RestartSec=1
```

```
Restart=on-failure
```

```
[Install]
```

```
WantedBy=multi-user.target
```



```
root@freepbx:~
File Edit View Search Terminal Help
GNU nano 2.3.1 File: /lib/systemd/system/unitracker.service Modified

[Unit]
Description=Loway Unitracker (Unloader)
After=syslog.target network.target

[Service]
Type=simple
EnvironmentFile=/etc/unitracker
Environment=AMISECRET=${AMISECRET}
Nice=15
KillMode=process
PIDFile=/var/run/unitracker.pid
ExecStart=/usr/bin/uniloader track --host "${AMIHOST}" --port "${AMIPORT}" --lo$
--debugfile "${DEBUGFILE}" \
--moh "${ENABLEMOH}" \
--parkedcalls "${ENABLEPARK}" \
--outboundcalls "${ENABLEOUTBOUND}" --outboundthreshold "${OUTBOUNDTH$
RestartSec=1
Restart=on-failure

[Install]
WantedBy=multi-user.target

^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^U UnCut Text ^T To Spell
```

Запуск и доступность служб

В Systemd необходимо уведомить демон (daemon), что есть новые init файлы, сообщить, что их нужно выполнять при загрузке, и запустить их. Выполните следующие команды для Uniloader:

```
systemctl daemon reload
```

```
systemctl enable uniloader
```

```
systemctl start uniloader
```

Проверьте, что uniloader запущен:

```
systemctl status uniloader
```

```
[root@freepbx ~]# systemctl status uniloadер
● uniloadер.service - SYSV: uniloadер - QueueMetrics Data Loader.
   Loaded: loaded (/etc/rc.d/init.d/uniloadер; bad; vendor preset: disabled)
   Active: active (running) since Fre 2018-08-10 13:57:08 UTC; 8min ago
     Docs: man:systemd-sysv-generator(8)
    Main PID: 1682 (uniloadер)
```

И перезапустите службу, после того, как сделаете изменения в файле конфигурации:

```
systemctl restart uniloadер
```

Все логи будут посланы в системный журнал, и доступны при помощи команды:

```
journalctl -u uniloadер
```

Та же самая процедура используется для подключения unitracker.

Ссылки

Более подробная техническая информация о решении для колл-центров QueueMetrics находится в [User Manual](#).

Посетите www.queuemetrics.com для получения 30-дневной бесплатной триальной версии.

Обратите внимание на [Free Webinars](#) с живой демонстрацией QueueMetrics.