dcron Documentation

Release 0.9

Pim Witlox

Aug 28, 2019

Contents:

The aim of dcron is to offer cron like behaviour spanning multiple machines. The system offers a web interface to manage your jobs, and reports the health of the cluster. Everything is self contained, so you only need to start the system to have a working setup. We do however recommend that you run the system behind a reverse proxy, since there is no authentication mechanism. Please check the configuration page regarding installation, configuration and options.

CHAPTER 1

Installing the package

You need python 3.7 including development libraries or higher to run this package. The package can be installed using `pip install dcron`.

CHAPTER 2

Running the package

Our package is self contained, so you can start it by simply calling dcron.

2.1 Configuration

The only package specific configuration is given through the command line options. You may however want to run a more 'robust' setup then simply starting the application in a screen session. In order to do so, we will use systemd for starting our application and use apache2 as a reverse proxy. We recommend not running this system on a broad subnet, because it is broadcasting over UDP, use a private subnet for the nodes if possible.

2.1.1 python3.7 on Ubuntu 18

Install python 3.7 on Ubuntu from source (or from your package manager if it's included), you need the development headers (Python.h):

```
update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
update-alternatives --install /usr/bin/python python /usr/bin/python3.6 2
update-alternatives --install /usr/bin/python python /usr/bin/python3.7 3
```

Now run *update-alternatives –config python* and select 3.7 as the default interpreter. You need to install pip, so run:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python get-pip.py
```

You should now be able to install dcron from pip.

2.1.2 systemd

Download our systemd example service file *dcron.service* from the repository and adapt where necessary. Note that the most important thing is the user the service runs as, the user will need full application access for the cronlike jobs.

The usual spot for the file is */etc/systemd/system/dcron.service*. After downloading and editing run *systemctl daemon-reload* for the service to show up. Now run *systemctl start dcron* to check if everything is working. The webservice should be available under port 8080 (or whatever you configured).

2.1.3 apache2

Our system doesn't do any authentication, so we will configure apache2 as a reverse proxy with authentication.

Install apache2: *apt install apache2*

Configure apach2 modules:

```
a2enmod proxy
a2enmod ssl
a2enmod proxy_http
a2ensite default-ssl.conf
systemctl restart apache2
```

Edit /etc/apache2/sites-available/default-ssl.conf:

```
<IfModule mod_ssl.c>
   <VirtualHost _default_:443>
            #ServerAdmin admin@example.com
            #ServerName www.example.com

            ErrorLog ${APACHE_LOG_DIR}/error.log
            CustomLog ${APACHE_LOG_DIR}/access.log combined

            SSLEngine on
            SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
            SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

            <FilesMatch "\.(cgi|shtml|phtml|php)$">
                           SSLOptions +StdEnvVars
            </FilesMatch>
            <Directory /usr/lib/cgi-bin>
                           SSLOptions +StdEnvVars
            </Directory>
            BrowserMatch "MSIE [2-6]" \
                           nokeepalive ssl-unclean-shutdown \
                           downgrade-1.0 force-response-1.0
            BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

            ProxyRequests On
            ProxyPreserveHost On

            <Proxy />
                Order deny,allow
                Allow from all
            </Proxy>


            <Location />
              Order deny,allow
              Allow from all

              ProxyPass http://localhost:8080/
```

(continues on next page)

(continued from previous page)

```
            ProxyPassReverse http://localhost:8080/

            AuthType Basic
            AuthName "dcron"
            AuthBasicProvider file
            AuthUserFile /etc/apache2/.htpasswd

            Require valid-user
          </Location>
     </VirtualHost>
</IfModule>
```

Edit /etc/apache2/sites-available/000-default.conf:

```
<VirtualHost *:80>
   #ServerName www.example.com

   ServerAdmin webmaster@localhost

   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined

   Redirect / https://external.machine.address
</VirtualHost>
```

For every user you want to give access, run the following command:

htpasswd -c /etc/apache2/.htpasswd <user>

and enter a password.

2.1.4 Log file rotation

Check if logrotate is installed on your system, if not install it *apt-get install logrotate*. Create the file /etc/logrotate.d/dcron:

```
/var/log/dcron.log {
    weekly
    size 100M
    rotate 12
    compress
    delaycompress
    missingok
    notifempty
    create 644 root root
}
```

You should now be good to go.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search