Installation

- Linux?
  - As easy as 'pip install supervisor'
- Windows?
  - No not possible
- Other Installation methods

Kick Start? Yes!

- See what options exists

  <echo_supervisord_conf>

- Create your config file like this

- What are we waiting for ???

  <supervisord -c /etc/supervisor/supervisord.conf>

Alright. So what does it do.

- Not a replacement for upstart/systemd/init

- Compliments service manager. How?
  - Service manager start your process on boot, but do not monitor crash
  - Does not depend of pid files
  - Allows non root users to start/stop/restart user level services
  - Ordering of service start up
  - Daemonizes your script and logs its <stdout> and <stderr>
  - Provides hooks you could exploit in your code

- Read it yourselves

Whats it got?

- The server <supervisord>
- The client <supervisorctl>
- What else?
  - 'XML-RPC API'
  - 'Web UI'
  - 'Event management'

Lets Process!

- Your process need not be a daemon
- Could be as simle as printing a few lines

•Yes just print. Supervisord will log it for you.

```
import os
import time

sin, sout, serr = os.popen3('ps -ef ')
while 1:
    print sout.read()
    print serr.read()
    time.sleep(10)
```

- Tell supervisord to manage your process

```
[program:process_check]
command=python /home/akandavelu/spyder/supervisord_learning/process_check.py
```

Start ! Stop ! Restart !

- Relax ! supervisorctl will do it for you.

```
supervisorctl -u user -p 123 status process_check
supervisorctl -u user -p 123 restart process_check
supervisorctl -u user -p 123 stop process_check
supervisorctl -u user -p 123 tail process_check
```

That is not all

- Process will be started in the order in which they are specified
- More control on process is available through events
- Process can be started/stopped in groups
- Complete list of process options

Lets do it on Web

- Enable XML RPC and HTTP in supervisor config file

```
[unix_http_server]

file=/tmp/supervisor.sock   ; (the path to the socket file)
chmod=0700                  ; socket file mode (default 0700)
chown=nobody:nogroup        ; socket file uid:gid owner
username=user               ; (default is no username (open server))
password=123                ; (default is no password (open server))

[inet_http_server]          ; inet (TCP) server disabled by default
port=127.0.0.1:9001         ; (ip_address:port specifier, *:port for all iface)
username=user               ; (default is no username (open server))
password=123                ; (default is no password (open server))
```

Do it from another program

- The XML RPC api allows for interaction to supervisord programatically

```
import xmlrpclib

server = xmlrpclib.Server('http://localhost:9001/RPC2')

server.system.listMethods()
server.supervisor.getAllProcessInfo()
```

Child Process Logging

- By default stdout and stderr of child process are stored in /tmp
- Can be changed using 'childlogdir' param in config file
- Options for logging stdout and stderr to separate files, along with max file size are available.

Events ??

- To detect process crash and respond accordingly
- Generate crash report and notify admin via mail
- How does it happen?
  - User specifies event listeners in supervisor config file
  - User specifies what event types the listener should be sent
  - Supervisord talks to the listener via its stdin and stdout in pre defined format

Create a listener

- Configure a listener that subscribes for crash event from a process

```
[eventlistener:check_process_crash]
command=python /home/akandavelu/spyder/supervisord_learning/check_process_crash.py

events=PROCESS_STATE_EXITED
```

- Started listener would be invoked every time any process exits
- Listeners could aslo be configured to run at a specified frequency

More ?

- Writing listeners is easiers in python with 'supervisor.childutils'
- Supervisord XML RPI can be customized
- Configure supervisord to send mails on events
- Automate you build
- Third party applications using supervisord.

More ??

- Supervisord is opensource so you can contribute
- Its extensible, so you can add more features on the fly

Thank You

Ageeleshwar K