

- [Introduction](#)
- [Presentation](#)
- [Hands-On Exercise](#)
  - [Host 1 \(ACS Services + Containers\)](#)
  - [Host 2 \(Containers\)](#)
  - [Host 3 \(Or any of the previous hosts\)](#)
- [Discussion](#)

When doing the ACS workshop and demos experiences, executing the commands directly on the hosts makes sense and doesn't impose a major burden. However, when the systems start to grow and operational requirements appear, the system need better ways to handle the deployment of containers and services. For this reason ACS Daemons were created, being the master processes acsservicesdaemon and acscontainerdaemon.

The case of the acscontainerdaemon process is a CORBA object that can be used to command the start, stop, get a core dump, etc. of containers. It is possible to configure it as a system daemon using /etc/init.d, systemctl, supervisor or other technologies.

The acsservices daemon offers the same kind of functionality as a CORBA object, but it is a bit more complex, creating child processes called Imps to monitor their related services. If services that should be running are killed or stop for any reason, then these imp processes try to restart them as soon as possible.

ACS doesn't provide scripts to simplify the configuration of the ACS Daemons on the operating system, this is a task that at this point has to be performed by the developers and/or sysadmin specialists.

- Scope
  - ACS Services Daemon
  - ACS Container Daemon
  - ACS Daemons Architecture
- Duration: 10 minutes

Starting 2 (or more) instances of the virtual machine, make them interact with each other.

## Host 1 (ACS Services + Containers)

In one machine configure the daemons:

```
acsservicesdaemon &
acscontainerdaemon &
```

Later, to stop the daemons:

```
fg
Ctrl+C
fg
Ctrl+C
```

or:

```
ps u |grep acscontainerdaemon |grep -v grep |awk '{print $2}' |xargs kill
ps u |grep acsservicesdaemon |grep -v grep |awk '{print $2}' |xargs kill
```

## Host 2 (Containers)

In the other machine configure only the container daemon:

```
acscontainerdaemon &
```

Later, to stop the daemon:

```
fg
Ctrl+C
```

or:

```
ps u |grep acscontainerdaemon |grep -v grep |awk '{print $2}' |xargs kill
```

## Host 3 (Or any of the previous hosts)

Interact with the daemons on the other hosts

```
acsdaemonStartAcs -i 0 -H <host1>  
acsdaemonStartContainer -t cpp -c bilboContainer -i 0 -H <host1>  
acsdaemonStartContainer -t java -c frodoContainer -i 0 -H <host2>
```

Later, to bring down the system:

```
acsdaemonStopContainer -c bilboContainer -i 0 -H <host1>  
acsdaemonStopContainer -c frodoContainer -i 0 -H <host2>  
acsdaemonStopAcs -i 0 -H <host1>
```

- On-Demand Containers + Deploy Info
- How to improve ACS Daemons