How to write Instance profile

▼ Details

In this tutorial, we will see how to write an Instance Profile

SlapOS defines several conventions about Software Instances.

Content

What is an instance profile

▼ Details

As a reminder, the instance profile, "instance.cfg", is run by buildout when slapgrid asks for deployment of an instance. This is a simple Buildout profile describing how to deploy the instance:

- Create a directory
- Generate the MySQL configuration
- Create a wrapper from the mysqld binary installed with the Software Release
- ...

Think "Promise based"

As we saw earlier, SlapOS Node processes (i.e run Buildout) each instance at least once per day. It means that Instance profiles and recipes have to be "promise based" and shouldn't blindly initialize the instance and should take into consideration that it will be run very often.

One practical way to handle this problem is to check for any existing data before overwriting anything that can possibly break instance.

Example of pseudocode taking care of existing data:

```
def install(self):
  if not password_already_set:
    set_password()
```

Here, it is safe to run often buildout for the instance because it won't overwrite your password.

▼ Details

Define list of processes to run

Generally speaking, a service (== a Software Instance) is composed of one or several executables (Apache, Mariadb, Varnish, ...) that are run by the operating system.

In SlapOS, you can define such executables that will be run when starting the instance, then terminated when stopping the instance.

There are two types of executables:

- **${buildout:directory}/etc/services**: All executables that are needed for the service (mysqld, apache, ...) should go to ${buildout:directory}/etc/services. If one of them exits, it will trigger the sending of an alert (bang) to the SlapOS Master and cause buildout to run for this instance.
- **${buildout:directory}/etc/run**: All executable scripts designed to run for a short time then exit when done without disturbing the service should go to ${buildout:directory}/etc/run. Scripts are usually needed to bootstrap the service but are not necessary for the service itself: generation of SSL keys, helper tools needed to inject commands to the executable acting as service, ...

Those executables can be anything: shell scripts, Python scripts, symbolic links to executable located into the Software Release directory (/opt/slapgrid/0123456789abcdef), ...

Usually, it is a simple shell script that runs (exec) an executable located into the Software Release directory, with all the needed arguments: location of configuration file, option to stay in foreground, ...

▼ Details

SlapOS Node, after having run buildout for a Software Instance, will add all executables present in those two directories to the Supervisor (a.k.a process manager) configuration then ask start or stop of them. Supervisor is then responsible of reporting to SlapOS Node any suspect "service" process exit.

Set default instance parameters

▼ Details

You will usually need your users to specify a few parameters for their instance. It can be anything; in the case of KVM (virtual machine), we allow the user to define amount of RAM, CPU, disk, etc.

Of course, by default, we want the instance to just work. So we provide default parameters. They are useful for two things:

1. Obviously define a default value if the user didn't specify it;
2. Prevent Buildout to exit due to "parameter not found", with just an empty value.

A generic rule is that an empty parameter is equivalent to no parameter at all.

Example

In the KVM instance profile, we can add, in the slap-parameter section representing all the parameters given by the user:

```
[slap-parameter]
# Default value if no second disk is specified. Will be ignored by the kvm recipe
second-disk-location =
# Default value for RAM amount, in MB
ram-size = 1024
```

Then, if the user specify ram-size, it will just overwrite the value here.