

Allocate SlapOS instance from API

by [SlapOS Team](#).

▼ Details

Agenda

- What is slapconsole
- Request an instance A from API (console)
- Request an instance B from API (console) on your own SlapOS Node
- Request an instance C from API (console) with parameters
- More informations about API/console

What is slapconsole

slapconsole is included with SlapOS client (and SlapOS Node).

It is a python console with the necessary modules already imported.

It will allow you to request and manage your instances, as if you were on slapos.org, but from the console.

Request from console "In the cloud"

```
$ slapos console ~/.slapos/slapos-client.cfg
>>> instance_a = request(wordpress, "Wordpress A from console")
>>> instance_a.getState()
Traceback (most recent call last):
  File "", line 1, in
  File "/Library/Python/2.6/site-packages/.../slapos/console.py", line 108, in
    slap.registerOpenOrder().request(software_release, reference)
  File "/Library/Python/2.6/site-packages/slapos/slap/slap.py", line 162, in request
    self._connection_helper.POST('/requestComputerPartition', request_dict)
  File "/Library/Python/2.6/site-packages/slapos/slap/slap.py", line 469, in POST
    raise ResourceNotReady("%s - %s" % (path, parameter_dict))
ResourceNotReady: /requestComputerPartition
# Wait for a few minutes and retry until it works.
# Calling "request" with an existing instance name just fetch/update parameters from existing.
>>> instance_a = request(wordpress, "Wordpress A from console")
>>> instance_a.getState()
Started
>>> instance_a.getConnectionParameter('url')
[URL appears here]
```

▼ Details

Slapconsole allows you to interact with the SlapOS Master using the SLAP API. Launch slapconsole, then ask the Master to deploy an instance of your software. First, it will raise an exception meaning that the instance is not ready yet. You will have to wait until the instance is ready. From time to time, run the same commands again to fetch up-to-date informations. When it no longer throw an exception, your instance is ready. You can then fetch the informations you need and connect to your instance.

Request from console in specific Node

```
$ slapos console ~/.slapos/slapos-client.cfg
>>> instance_b = request(wordpress, "Wordpress B from console",
    filter_kw={"computer_guid": "COMP-12345" })
>>> instance_b.getState()
Traceback (most recent call last):
  File "", line 1, in
  File "/Library/Python/2.6/site-packages/.../slapos/console.py", line 108, in
    slap.registerOpenOrder().request(software_release, reference)
  File "/Library/Python/2.6/site-packages/slapos/slap/slap.py", line 162, in request
    self._connection_helper.POST('/requestComputerPartition', request_dict)
  File "/Library/Python/2.6/site-packages/slapos/slap/slap.py", line 469, in POST
    raise ResourceNotReady("%s - %s" % (path, parameter_dict))
ResourceNotReady: /requestComputerPartition
# Wait for a few minutes and retry until it works:
>>> instance_b = request(wordpress, "Wordpress B from console",
    filter_kw={"computer_guid": "COMP-12345" })
>>> instance_b.getState()
Started
>>> instance_b.getConnectionParameter('url')
[URL appears here]
```

▼ Details

Here we are going to request a new instance on a specific computer.

Request with parameters

```
$ slaposconsole ~/.slapos/slapos-client.cfg
# We will modify our first instance:
>>> instance_a = request(wordpress, "Wordpress A from console",
partition_parameter_kw={"domain":"mydomain.com"})
# Wait for a few minutes for the change to be effective:
>>> instance_a = request(wordpress, "Wordpress A from console",
partition_parameter_kw={"domain":"mydomain.com"})
>>> instance_a.getConnectionParameter('url')
[New URL appears here]
```

▼ Details

Here we are going to modify an existing instance to add some parameters. Of course, you can directly define parameters in new instances.

Next Steps

Read slapconsole documentation

[Link to documentation](#)