

# Using KumoFS with SlapOS

by [Jean-Paul Smets](#).

▼ Details

## Agenda

- Request from web UI, connect
- Request from console, connect
- Request on own computer from web UI, connect
- Request on own computer from console, connect

## Web Request (1)

□

▼ Details

Register to vifib.net, then go to "My Services".

To request an instance using vifib.net web interface, go to vifib.net, go to My Services, click on the "New service" button. Put the wanted name of the service in "Service title", then select "Kumofs" in the list of software. Make sure you have only one selected software. Send the form by clicking on "Order". On the next page, select the latest version of KumoFS available, and click on "order". Review and fill your informations, and click on the "Continue" button. You will be redirected to the home page. You will then have to wait for a few minutes for the instance to be deployed. You can know the state of the instantiation by going to "My Services" page. Your instance will appear after a few seconds, if you click on it, you can see the state of it.

## Web Request (2)

□

▼ Details

Click on "New service".

## Web Request (3)

□

▼ Details

Choose a name for your instance, and select "Kumofs" service type.

## Web Request (4)

□

▼ Details

Choose the latest version of the software.

## Web Request (5)

□

▼ Details

Fill in your personal informations.

## Web Request (6)

□

▼ Details

Instance has been requested, you are redirected to home. Wait for a few seconds, then go back to "My Services".

## Web Request (7)

□

▼ Details

You can see your instance : click on it.

## Web Request (8)

□

#### ▼ Details

You can see your instance being deployed. Reload to have up-to-date informations.

## Web Request (9)

□

#### ▼ Details

Your instance is ready. All the needed informations can be found in the "Connection" xml.

## Configure stunnel

Install, (re)configure and launch stunnel to connect to your instance :

```
# apt-get install stunnel
$ mkdir ~/slapos
$ echo "foreground = yes
syslog = no
pid = /home/${whoami}/slapos/stunnel.pid
debug = debug

[service]
accept = 127.0.0.1:11211
connect = REPLACE_ME_BY_YOUR_STUNNEL_IP:12345
client = yes" > /home/${whoami}/slapos/stunnel.conf

$ stunnel4 /home/${whoami}/slapos/stunnel.conf
```

#### ▼ Details

Install stunnel on your machine (on debian, do apt-get install stunnel). Configure it, and launch it in foreground. Open a new terminal before going to the next step, and let this one opened with stunnel running.

## Connect to Kumofs

```
$ telnet localhost 11211
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
set hoge 0 0 3
Foo
STORED
set bar 0 0 22
This is a sample text.
STORED
END
Connection closed by foreign host.
```

#### ▼ Details

Connect to your instance using memcached utility.

## Request from console

```
# In your slapos client directory :
$ bin/slapconsole slapos.cfg
>>> kumo_instance = request(kumofs, "second_kumo")
Traceback (most recent call last):
  File "", line 1, in
  File "/Library/Python/2.6/site-packages/slapos.core-0.4_dev-py2.6.egg/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 :
>>> kumo_instance = request(kumofs, "second_kumo")
>>> kumo_instance.getState()
Started
>>> kumo_instance.getConnectionParameter('stunnel_ip')
[IP appears here]
>>> kumo_instance.getConnectionParameter('stunnel_port')
[Port appears here]
```

#### ▼ Details

Slapconsole allows your 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 again the same command to fetch up-to-date informations. When it does not longer throw an exception, your instance is ready. You can then fetch the informations you need to connect to your instance

## Configure stunnel

Install, (re)configure and launch stunnel to connect to your instance :

```
# apt-get install stunnel
$ mkdir ~/slapos
$ echo "foreground = yes
syslog = no
pid = /home/${whoami}/slapos/stunnel.pid
debug = debug

[service]
accept = 127.0.0.1:11211
connect = REPLACE_ME_BY_YOUR_STUNNEL_IP:12345
client = yes" > /home/${whoami}/slapos/stunnel.conf

$ stunnel4 /home/${whoami}/slapos/stunnel.conf
```

### ▼ Details

Install stunnel on your machine (on debian, do apt-get install stunnel). Configure it, and launch it in foreground. Open a new terminal before going to the next step, and let this one opened with stunnel running.

## Connect to Kumofs

```
$ telnet localhost 11211
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^J'.
set hoge 0 0 3
Foo
STORED
set bar 0 0 22
This is another sample text.
STORED
END
Connection closed by foreign host.
```

### ▼ Details

Connect to your instance using memcached utility.

## Request from console

```
# In your slapos client directory :
$ bin/slapconsole slapos.cfg
# Wait for a few minutes and retry until it works :
>>> kumo_instance = request(kumofs, "first_kumo")
>>> kumo_instance.getState()
Started
>>> kumo_instance.getConnectionParameter('stunnel_ip')
[IP appears here]
>>> kumo_instance.getConnectionParameter('stunnel_port')
[Port appears here]
```

### ▼ Details

If you request an existing instance, vifib won't create anything, but it will give you the instance parameters if you have sufficient rights. Here we already requested first\_kumo within Web UI, and we see that we fetch the same instance than before.

## Configure stunnel

Install, (re)configure and launch stunnel to connect to your instance :

```
# apt-get install stunnel
$ mkdir ~/slapos
$ echo "foreground = yes
syslog = no
pid = /home/${whoami}/slapos/stunnel.pid
debug = debug

[service]
```

```
accept = 127.0.0.1:11211
connect = REPLACE_ME_BY_YOUR_STUNNEL_IP:12345
client = yes" > /home/$(whoami)/slapos/stunnel.conf
```

```
$ stunnel4 /home/$(whoami)/slapos/stunnel.conf
```

#### ▼ Details

Install stunnel on your machine (on debian, do apt-get install stunnel). Configure it, and launch it in foreground. Open a new terminal before going to the next step, and let this one opened with stunnel running.

## Connect to Kumofs

```
$ telnet localhost 11211
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
set hoge 0 0 3
Foo
STORED
set bar 0 0 22
This is another sample text.
STORED
END
Connection closed by foreign host.
```

#### ▼ Details

Connect to your instance using memcached utility.

## Request with SLA (Req.)

#### ▼ Details

Before trying to request an instance on a specific computer, make sure the software you want to use is deployed on this machine. If not sure, go to :

[Install and Configure SlapOS Node](#)

## Request with SLA (1)

□

#### ▼ Details

Register to vifib.net, then go to "My Services".

To request a Kumofs instance using vifib.net web interface, go to vifib.net, go to My Services, click on the "New service". Put the desired name of the service in "Service title", then select "Kumofs" in the list of softwares. Make sure you have only one selected software. Send the form by clicking on "Order". On the next page, select the latest version of kumofs available, and click on "order". Review and fill your informations, and click on the "Continue" button. You will be redirected to the home page. You will then have to wait for a few minutes for the instance to be deployed. You can know the state of the instantiation by going to "My Services" page. Your instance will appear after a few seconds, if you click on it, you can see the state of it.

## Request with SLA (2)

□

#### ▼ Details

Click on "New service".

## Request with SLA (3)

□

#### ▼ Details

Choose a name for your instance, and select "Kumofs" service type. Make sure to select the computer where you want the instance to be deployed.

## Request with SLA (4)

□

#### ▼ Details

Choose the latest version of the software.

## Request with SLA (5)

□

### ▼ Details

Fill in your personal informations.

## Request with SLA (6)

□

### ▼ Details

Instance has been requested, you are redirected to home. Wait for a few seconds, then go back to "My Services".

## Request with SLA (7)

□

### ▼ Details

You can see your instance : click on it.

## Request with SLA (8)

□

### ▼ Details

You can see your instance being deployed. Reload to have up-to-date informations.

## Request with SLA (9)

□

### ▼ Details

Your instance is ready. All the needed informations can be found in the "Connection" xml.

## Configure stunnel

Install, (re)configure and launch stunnel to connect to your instance :

```
# apt-get install stunnel
$ mkdir ~/slapos
$ echo "foreground = yes
syslog = no
pid = /home/$(whoami)/slapos/stunnel.pid
debug = debug

[service]
accept = 127.0.0.1:11211
connect = REPLACE_ME_BY_YOUR_STUNNEL_IP:12345
client = yes" > /home/$(whoami)/slapos/stunnel.conf

$ stunnel4 /home/$(whoami)/slapos/stunnel.conf
```

### ▼ Details

Install stunnel on your machine (on debian, do apt-get install stunnel). Configure it, and launch it in foreground. Open a new terminal before going to the next step, and let this one opened with stunnel running.

## Connect to Kumofs

```
$ telnet localhost 11211
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
set hoge 0 0 3
Foo
STORED
set bar 0 0 22
This is another sample text.
STORED
END
Connection closed by foreign host.
```

### ▼ Details

Connect to your instance using memcached utility.

## SLA from console

```
# In your slapos client directory :
$ bin/slapconsole slapos.cfg
>>> kumo_instance = slap.registerOpenOrder().request(kumofs, "fourth_kumo",
    filter_kw={"computer_guid": "COMP-12345" })
Traceback (most recent call last):
  File "", line 1, in
  File "/Library/Python/2.6/site-packages/slapos.core-0.4_dev-py2.6.egg/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 :
>>> kumo_instance = slap.registerOpenOrder().request(kumofs, "fourth_kumo",
    filter_kw={"computer_guid": "COMP-12345" })
>>> kumo_instance.getState()
Started
>>> kumo_instance.getConnectionParameter('stunnel_ip')
[IP appears here]
>>> kumo_instance.getConnectionParameter('stunnel_port')
[Port appears here]
```

### ▼ Details

Slapconsole allows you to interact with the SlapOS Master using the SLAP API.

Replace the COMP-12345 by your Computer ID.

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 again the same command to fetch up-to-date informations.

When it does not longer throw an exception, your instance is ready. You can then fetch the informations you need to connect to your instance.

## Configure stunnel

Install, (re)configure and launch stunnel to connect to your instance :

```
# apt-get install stunnel
$ mkdir ~/slapos
$ echo "foreground = yes
syslog = no
pid = /home/$(whoami)/slapos/stunnel.pid
debug = debug

[service]
accept = 127.0.0.1:11211
connect = REPLACE_ME_BY_YOUR_STUNNEL_IP:12345
client = yes" > /home/$(whoami)/slapos/stunnel.conf

$ stunnel4 /home/$(whoami)/slapos/stunnel.conf
```

### ▼ Details

Install stunnel on your machine (on debian, do apt-get install stunnel). Configure it, and launch it in foreground. Open a new terminal before going to the next step, and let this one opened with stunnel running.

## Connect to Kumofs

```
$ telnet localhost 11211
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
set hoge 0 0 3
Foo
STORED
set bar 0 0 22
This is another sample text.
STORED
END
Connection closed by foreign host.
```

#### ▼ Details

Connect to your instance using memcached utility.

## Next Steps

## How to Contribute?

- Add your software to SlapOS
- Add fonctionnalités to the core of SlapOS

[Fork us!](#)