Installing SlapOS Slave Node from the Sources

by SlapOS Team.
▼ Details

The goal of this tutorial is to teach how to install a SlapOS Slave node from the sources. This tutorial is useful to understand the architecture of SlapOS from a practical point of view.

This tutorial is only focus on development. **/!\ DO NOT USE THIS INSTALLATION ON PRODUCTION /!** unless you know very well what you are doing.

This tutorial will guide you step by step through the installation of a SlapOS node on your own infrastructure for development purpose. You can install a SlapOS node everywhere : on your servers, on your desktop computer or laptop. SlapOS nodes on laptops are particularly efficient for running tests or any other non mission-critical service through SlapOS.

As a general convention use all commands from this tutorial with superuser privileges (root user).

Agenda

- Requirements
- SlapOS Installation
- Additionnal configuration

▼ Details

This tutorial will first list the requirements to install SlapOS Slave node from the sources. Then we will explain how to install SlapOS from the sources.

Requirements

▼ Details

Let us now review SlapOS installation requirements.

The Requirements

- A GNU/Linux Server
- python 2.7
- gcc, g++, make, linux-headers, patch
- uml-utilities, bridge-utils
- wget
- A slapos.org Account

▼ Details

To follow this tutorial, you will need a server with a GNU/Linux distribution, python2.7 installed

You will also need the "gcc" and "g++" compiler with "make" and "patch" tools and linux headers, "uml-utilities" for tunneling tools, and "bridge-utils" for bridge tools. In Debian-like distribution, you can install it with the following command (with root privileges) :

```
# apt-get install python gcc g++ make uml-utilities bridge-utils linux-headers-$(uname -r) patch wget
```

Your GNU/Linux server should either be connected to the Internet with native IPv6 global address. Or it should have a public IPv4 address. For those who only have a public IPv4 address, we shall explain here how to create a tunnel to get IPv6.

SlapOS Installation

▼ Details

Now that we have an IPv6 address, we can start the installation of SlapOS Slave Node software components.

A SlapOS Slave Node is composed of computer partitions. As a reminder, a Computer Partition on a Slave Node consists of a UNIX user, its home directory, a dedicated tap interface and a dedicated IPv6 address.

A computer partition can host a SlapOS software instance. Whenever a software instance is requested to SlapOS Master, SlapOS Master looks for free partitions on every registered Slave Node and chooses one depending on specific "service level" parameters specified by the user. The chosen SlapOS Node is then requested to instantiate the user specified software into an free computer partition. Whenever a software instance is no longer used, it is deleted from the Slave Node it was previously allocated to and the computer partition is freed and again available.

The installation process of SlapOS Slave node installs a couple of python scripts. Most important ones are:

- "slapos node format" will be invoked to create users, directories, tap interfaces and IPv6 addresses for every of each computer partition.
- slapos node instance/software, relies on slapgrid core API which implements the SLAP protocol and requests to SlapOS Master Node which software to install, to instanciate, to account and to delete.
- "slapos node console" provides a command line environment to contact SlapOS Master.
- "slapos node" provides access to the list of running processes in all computer partitions.

SlapOS Bootstrap

▼ Details

Copy/Paste the following shell script in your terminal :

```
# Download the example script, to build slapos on /opt/slapos

wget https://lab.nexedi.com/nexedi/slapos/raw/master/component/slapos/build.sh

# After download just run it.
bash build.sh
```

▼ Details

SlapOS is installed using buildout technology. Buildout downloads the source code, the patches and the binaries required to build a software on a given architecture. It handles depencies between software components automatically.

First, we are going to create the directory for SlapOS software by invoking mkdir /opt/slapos. We then move to the SlapOS software directory. We then create a buildout file which extends SlapOS's own buildout.cfg. There is nothing to download here, only create a text file with a link to a URL.

Before starting the boostrap process, we unset some environment variables. This is mostly useless for most GNU/Linux distributions or UNIX-like operaring systems but it prevents certain errors in a few GNU/Linux distributions with uncommon environment settings.

It is now possible to start the bootstrap process. We need to make sure that we use a recent version of python such as python 2.7. The python invocation bootstraps buildout. The buildout invocation interpretes the buildout.cfg file and installs SlapOS on the server.

Before you go to the next paragraph, please make sure that you have tried and finished installing of SlapOS with buildout.

Crontab (optional)

```
# Download default cron entries to slapos works automatically

wget https://lab.nexedi.com/nexedi/slapos.package/raw/master/obs/slapos/debian/cron.d -O /etc/cron.d/slapos
```

▼ Details

We can also now define the cron configuration. Simply copy the example above. Adapt parameters if you did not follow the exact tutorial (not recommended). slapformat is invoked every hour so that if we any changes happen on the network (something not supposed to happen), the master will eventually be notified. This is very useful in case server hardware is move from one house to another, from one data center to another, etc.

slapos node will run automatically every minute if you add those lines to your crontab. If you want to manually launch slapos node, do not add those lines.

Where to go next?

- Setup SlapOS to run on your computer

▼ Details

Now that your install is complete, it is time to move to the tutorial on running SlapOS !