

Install Slackware on a CloudVPS by ArubaCloud

ArubaCloud offers excellent low-cost VPS services, starting at 1 € / month (*at least when this article was created, feb. 2018, see ArubaCloud limitations*). This article refers to installing Slackware on [CloudVPS](#) hosting service, which is based on VMware virtualization hypervisor and uses several OS templates, Slackware not being among them. *As a low-cost service, custom OS installation, management with VMware Client are not available.* You can request a free trial [here](#).

This guide does not cover Cloud Pro or other services by ArubaCloud, as these services may offer additional tools that make this guide unnecessary. Alternately, this guide may be used for other similar services.

Prerequisites

Assuming you want a Slackware server hosted at ArubaCloud, here is what you need:

- ArubaCloud account
- enough credit to create the CloudVPS you want.
- a FTP hosting space with several GB of available space and reasonable speed.
- VMware Player on your computer and at least as much free space as the CloudVPS virtual HDD

Initialization

Once you get your free trial voucher or you purchased credit, you can follow ArubaCloud's web interface to create the CloudVPS of your choice. Using the free trial voucher you can get 2 months free trial for CloudVPS Small (*1xCPU, 1GB RAM, 1xETH, 20GB HDD, 2TB traffic/month*).

1. load credit into your account (free trial voucher or purchase credit)
2. create VPS based on predefined OS template

This guide has been written using an **Ubuntu Server 14.04 LTS 64bit** template.

Information gathering

Having the VPS up and running, and after receiving your access credentials, first write down some key information:

- HDD size and layout
- Public IP (of the VPS), including netmask, gateway

Preparations

Log in to the CloudVPS using SSH.

Show IP and netmask (write it down):

```
ip addr list
```

Show default gateway (write it down):

```
ip route list
```

Show nameservers (write it down):

```
cat /etc/resolv.conf
```

Show disk layout (write down the total disk size in blocks):

```
fdisk -l /dev/sda
```

Install **ncftp** package:

```
apt-get install ncftp
```

Optimize disk by overwriting all free space with zeros (helps with compression):

```
dd if=/dev/zero of=/ZERO bs=1M; rm /ZERO
```

Export an image of the Aruba created server

This step is optional. It will familiarize you with the transfer operation, will test your connection and FTP account storage, but it is not required.

Perform it if you want to play at home with the Aruba generated server

Aruba has predefined OS installations as templates, and such a template is deployed into your VPS upon creation. It is personalized with IP address, hostname and you are given an automatically generated password.

Make sure you have enough free space in your FTP account. This will take up several GB of data.

```
cat /dev/sda | bzip2 | ncftpput -u USERNAME -p PASSWORD -c  
FTP_SERVER_ADDRESS /path/to/aruba_image_file.bz2
```

This may take a while, even hours, depending on the available bandwidth between Aruba's datacenter and your FTP server.

It is a rather dirty trick, but it works. The template contains only a bare-bones installation with

minimal services running. You may check and see if there is any background daemon that might write much to the disk and stop it, but other than a few log entries, there isn't anything worth mentioning.

Make sure the process finishes without error before proceeding!

You can use this image to make tests on your local VMware Player, and also as a test for the FTP transfer. You need either a Linux Live CD or a running virtual machine with an installed Linux that includes **ncftp** package and a separate virtual HDD to fill with the image.

Slackware virtual machine

Create your own, local, virtual machine with VMware Player.

- Start VMware Player
- Create new virtual machine
- "I will install the operating system later"
- Choose "Linux" and "Other Linux 3/4.x 64bit"
- Name it (ie. aruba_slack)
- **Maximum disk size: same as VPS HDD size (in GB);** store as single file
- proceed to finish.

Modify the virtual machine you just created as follows:

- Memory = 1GB
- Processors = 1 core
- CD/DVD = use ISO image (select slackware64-14.2-install-dvd.iso locally), connected at power-on
- Network adapter = NAT, connected at power-on
- Sound card = REMOVE
- Printer = REMOVE

Start the virtual machine.

- Enter the console (click in the VM display area)
- Proceed to install Slackware as you would on a physical server with the same characteristics of your cloud server. *There are no exotic hardware or settings.*

For example, you could use a single partition for both system and data, and a small partition (ie. 1-4GB) for swap. Or you could use about a partition for the system, a partition for swap, a partition for data. *Your choice.*

For a server installation you might want to skip KDE, T, TCL, X, XAP, XFCE package series. After you finish your installation I strongly recommend you add your SSH public key for the root user for remote access. Follow this [How to use SSH keys to connect without a password](#). Either use the Virtual Machine's console, or ssh from your local machine to do it. Note: you may want to temporarily allow root login via SSH into the Virtual Machine, but remember to restore the default config afterwards.

You are running locally, with the machine behind a NAT router (done by VMware Player) with regards to the internet. You have connectivity, but it is different than the one you'll have at Aruba.

During the configuration phase, assign the proper hostname, but set it to obtain IP address via DHCP.

After the first start of your fresh Slackware installation, you should set up **slackpkg** and perform an update.

I'll leave additional software for later, to minimize the data transfer.

Exporting Slackware VM image

Make sure you have enough free space in your FTP account. This will take up several GB of data.

In your Slackware virtual machine:

- Stop all unnecessary services (inetd,sendmail,saslauthd,ntpd,httpd,sshd,syslog, etc)
- Optimize disk by overwriting all free space with zeros (helps with compression, *the resulting image, as described here will be about 1.5GB*):

```
dd if=/dev/zero of=/ZERO bs=1M; rm /ZERO
```

- Export disk image

```
cat /dev/sda | bzip2 | ncftpput -u USERNAME -p PASSWORD -c  
FTP_SERVER_ADDRESS /path/to/aruba_slack_image_file.bz2
```

This may take a while, even hours, depending on the available bandwidth between your computer and the FTP server.

It is a rather dirty trick, but it works.

Make sure the process finishes without error before proceeding!

Importing Slackware VM image into ArubaCloud

With the above step completed without error, you can proceed to inject the image into the ArubaCloud VPS.

- Log in to your ArubaCloud Control Panel.
- Go to "Cloud servers" → Management
- If your VPS is turned off, turn it on and wait until startup is completed (see progress indicator)
- Under "Actions", select "Manage → Sign in"
- Open the recovery console. You'll get to see your server's console.
- log in to your server, as root
- ALTERNATELY you could ssh into it. Also use this method if you encounter problems with recovery console key-mappings.
- check running processes and stop any unnecessary daemons (cron, acpid, atd, fail2ban, rsyslog, open-vm-tools)
- **you must also disable swap memory.**
- run the following command:

```
ncftppget -u USERNAME -p PASSWORD -c FTP_SERVER_ADDRESS
```

```
/path/to/aruba_slack_image_file.bz2 | bzip2 -d > /dev/sda
```

This may take a while, even hours, depending on the available bandwidth between Aruba's datacenter and your FTP server. At the end of the transfer, ncftpget will report the number of bytes transferred and speed. Make sure "bytes transferred" is exactly equal to the size of the compressed image (see your FTP account). You may get a "Segmentation fault" after the end of the transfer. This is acceptable and expectable.

Make sure the process finishes without error before proceeding!

IMMEDIATELY after it finished successfully:

- Go to "Cloud servers" → Management
- Under "Actions", select "Manage → Force switch off"

Very important: just "switch off" will try to perform a normal shutdown procedure, which won't work properly and might even damage the disk data you injected.

Start your Slackware ArubaCloud server:

- Go to "Cloud servers" → Management
- Under "Actions", select "Manage → Switch on"
- wait for it to start
- Under "Actions", select "Manage → Sign in"
- Open the recovery console. You'll get to see your server's console.
- log in to your server, as root
- run netconfig to set up the network according to ArubaCloud IP settings (*your wrote them down earlier*)
- just to make sure everything works properly, reboot your machine.
- restart the recovery console after reboot
- check that everything is ok
- check that you can log in remotely

Slackware default configuration won't allow remote root login without SSH key, so take the appropriate steps to ensure your remote access.

Now you have a running Slackware64-14.2 in an ArubaCloud VPS.

Additional configuration

Since it runs under VMware hypervisor, it is useful to have the VMware-tools installed. I recommend you install the open-source version from Slackbuilds: [open-vm-tools](#).

After your install the package and start the tools, you can verify that it works properly by:

- Go to "Cloud servers" → Management
- Under "Actions", select "Manage → Switch off"

This is the normal shutdown procedure initiated at the request of the hypervisor, received by open-vm-tools and performed in the guest OS to obtain a clean system shutdown.

The "Force switch off" is equivalent to pressing the power button on a physical computer.

Enjoy your Slackware VPS!

References

- [ArubaCloud](#) by Aruba S.p.A. (IT) - cloud services provider.
- [open-vm-tools](#) from Slackbuilds.org

Sources

- Originally written by [Andrei B.](#)

[howtos](#), [slackware](#), [VPS](#), [virtualization](#), [ArubaCloud](#), [Aruba](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:slackware_admin:slackware-arubacloud

Last update: **2018/02/20 19:06 (UTC)**

