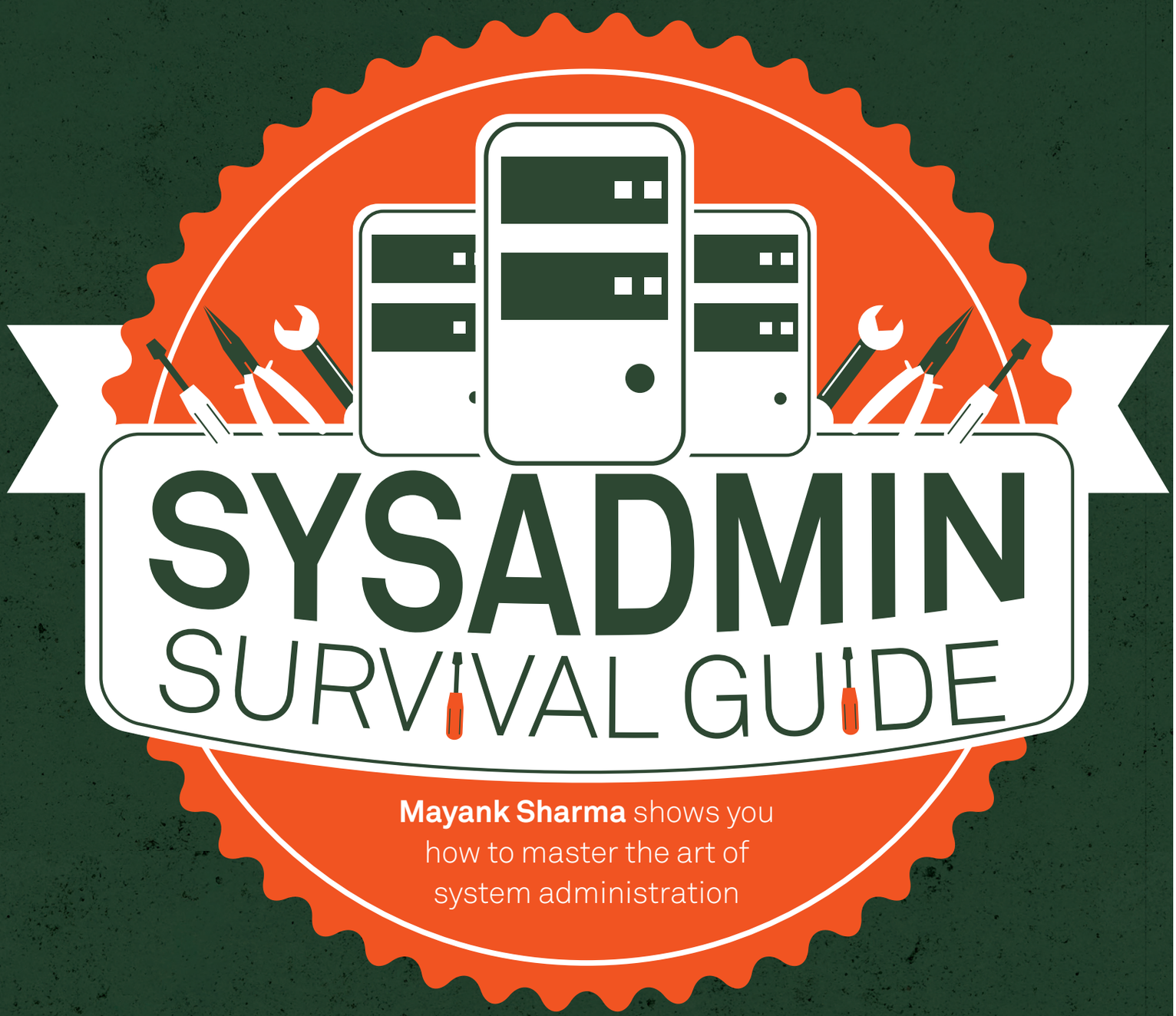


Mayank Sharma shows you how to master the art of system administration

AT A GLANCE

The field of Linux system administration has changed tremendously over the last few years. Be it the exodus to cloud computing or the proliferation of virtualised components and containerised apps, modern IT infrastructure has gone through a paradigm shift in a short span of time. No surprise, then, that organisations are on the look-out for administrators who are equipped with the necessary skills to handle this shift in enterprise technology.

Matt Simmons, Linux Systems Administrator for a leading American aerospace provider, believes system administration can be classified into three broad categories. What he calls traditional system admin is "where you only program to automate small things on a per-machine basis. The kind of role where you only write quick shell scripts to back things up, and maybe to grab some monitoring data that you throw into Nagios, Icinga, or something like Cacti. Your machines are probably mostly physical." He adds that admins of such networks spend most of their time on managing resources per machine and a sizeable chunk of their budget goes towards procuring hardware or updating them: "This is the minority, and it is shrinking."

The other end of the spectrum is all cloud: "You have automated jobs creating images regularly according to controlled rulesets, so that when you need to create a machine, it happens without any doubt as to what is on it and what it is doing, and the machine shuts down as soon as is possible; maybe because that service isn't needed any more, or maybe because the next image is ready to take over for it," says Matt. He adds that admins of such deployments spend their time looking for ways to optimise spending and consolidate and eliminate provided services. According to Matt this kind of administration is the minority as well, but is growing rapidly.

> "Today rarely do people have to get acquainted with boot loaders, kernels, file systems and device drivers, which limits them"

The majority that seems to be holding steady, in Matt's opinion, is a meeting of the two. He says the role in this segment has "probably shifted from 'systems administration' to what might be considered 'IT Operations'." Admins here use configuration management, and will surely have some services in the cloud: "You write automation whenever you get the chance. You probably work almost exclusively from tickets, you have CI/CD workflows for at least some of your code, and if you have a team, you probably do some sort of team programming and code review. Your machines are almost all virtual, and provisioned automatically without much, if any, interaction from you."

Abhas Abhinav, founder and hacker-in-charge of DeepRoot Linux who supports over 250 servers for clients all over India, says that as system admin is getting easier, the level of commitment and respect for the "craft of system admin" isn't what it once used to be. "On the other hand, [because of] the fact that systems are now easier to provision or procure because of the ubiquitous nature of VMs, many system admin skills are not getting enough attention. Today rarely do people have to get acquainted with boot loaders, kernels, filesystems and device drivers, and this limits the sort of problems they can manage and solve."

If you are a new system admin, this feature shows you what you need to know to get started. For the more experienced, it equips you with knowledge of the tech you need to be familiar with to stay relevant.

Storage & recovery

A good system admin uses storage systems efficiently and also ensures hardware failures don't bring down the house

Data is one of the most critical parts of any business, which means managing it is one of the core responsibilities of any system administator. The key to a good storage management policy is that it should strike the right balance between keeping the data safe yet accessible.

Before you can design a storage policy for an organisation, you need to understand how Linux deals with storage and familiarise yourself with the tools at your disposal. All it takes is a couple of clicks to add storage to a cloud server; provision a drive and then attach the disk to an existing VM. In the real world however, you'll first to decide on the type of device (e.g. hard disk, SSD) along with the interface to connect it to your server (e.g. USB, SATA, PCIe). When the Linux kernel detects a device, it tells udev, which then creates a representation of that device in the `/dev` directory. These device files are the way the kernel provides access to devices for applications and services. The storage devices are called block devices, and a good practice is to refer to them using their Universally Unique Identifiers or UUIDs. The next piece of the puzzle is partitioning – or rather its more dexterous cousin, logical volume management (LVM). While the traditional partitioning schemes MBR and GPT are much simpler, they are rather inflexible. LVM, on the other hand, combines one or more devices into a single logical volume group. You can then dynamically create, resize and delete volumes in a volume group to reallocate space. It also offers other benefits such as snapshot management.

> "Mastering rsync is one of those evergreen skills that'll never go out of vogue"

Linux systems give you several options for partitioning, with `parted` and its graphical frontend `gparted` leading the pack. To manage volumes you need to master `lvm`.

While you can (and should) supplement all storage mechanisms with backups, it's always a good idea to use a RAID system to distribute data across multiple disks. With the plummeting cost of storage, a RAID setup helps avoid data loss and also minimises the downtime associated with hardware failures. While RAID can be implemented by dedicated hardware, it can also be implemented by software. Use the `mdadm` command to build and use software RAIDs. No matter how you implement it, make sure you are well versed in the different RAID levels to pick the one that works best for your organisation.

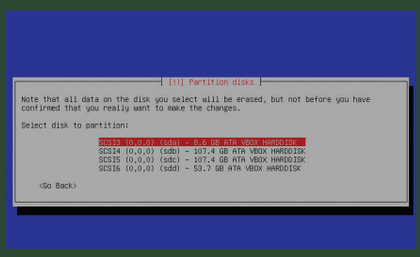
Another popular storage deployment skill that should be in the repertoire of an admin is the ability to set up a network-attached storage (NAS) device. OpenMediaVault (**www.openmediavault.org**) can make this process effortless. It supports all the popular deployment mechanisms, including software RAID, and can be accessed using

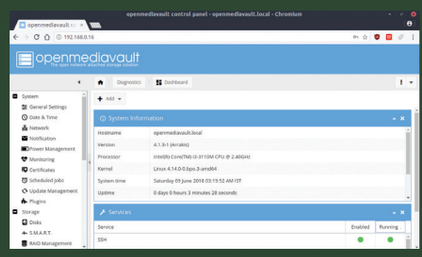
EXPERT TIP

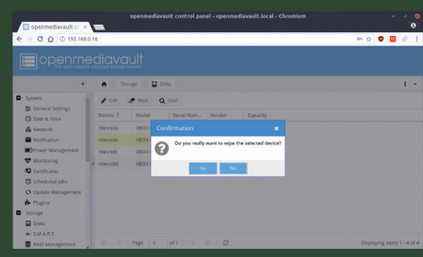
RAID is not backup

Don't use Mirrored RAID as an excuse to avoid taking backups. You don't want to be in a situation where a fried machine renders both disks useless.

HOW TO
Create a NAS with OpenMediaVault







1 Install OMV

Installing OMV is pretty straightforward. Download the ISO, transfer it to a USB disk and run through the basic steps. Since you have multiple disks connected to the NAS box, make sure you select the right installation target during the partitioning step.

2 The web interface

When you're done, reboot the computer, which will drop you to the login shell. Don't log in but instead head to the web address displayed on the console. This brings up OMV's web interface, from which you can manage all aspects of the NAS server.

3 Set up storage

You'll first need to format the drives before you can use them. Head to Storage > Disks to view all the attached disks. Select the drive and click the 'Wipe' button. After you've erased a drive, head to Storage > File Systems to create a file system on the drive.

Below Grsync is a graphical front-end to rsync that exposes virtually all of rsync's command-line options

all the leading network protocols (see the walkthrough below).

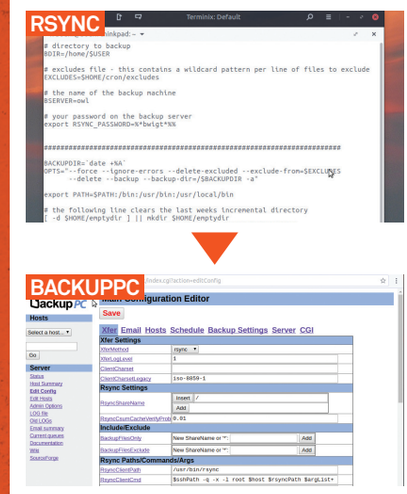
Before you can press the disk into active service, it needs to have a file system on it. Most Linux distributions default to the EXT4 journalled filesystem, but there are several others that you should be familiar with. There's NTFS, which is useful for interoperability with Windows machines, and XFS, which works well for housing database files. The needs of organisations are, however, much better satisfied with next-generation file systems, such as ZFS and Btrfs, that perform better than traditional file systems and are more serious about data integrity.

Once you've got a handle on the storage systems in the organisation, it's time to work out a plan for backing up all the data. Irrespective of the technology that powers it you'll first have to spend some time answering questions like:

- What data is to be backed up?
- Where will backup data be stored?
- How often will backups be performed?
- How long will backups be retained?
- How will backups be accessed or restored?
- What system or technology will perform the backups/restoration?

The answers to these (and other) questions depend on various factors such as cost and the amount of acceptable downtime. For example, the availability advantage of remote cloud storage diminishes with an increase in the amount of backed-up data.

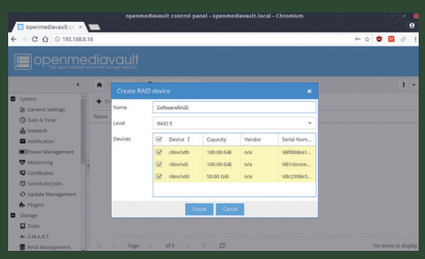
There are a plethora of reputable open source backup tools and a number are designed specifically to handle large amounts of data. One of the industry favourites is rsync, which scales well and offers enough dexterity to power desktop backup apps as well as enterprise ones like BackupPC. Mastering rsync is one of those skills that'll never go out of vogue.



QUICK TIP

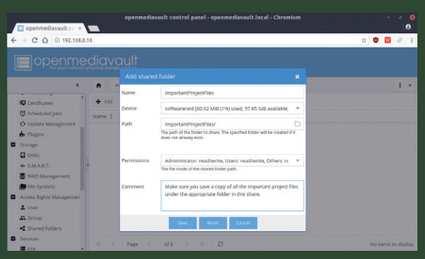
rsync and BackupPC

rsync is one of the handiest tool to copy and sync files across the network. But if writing scripts isn't your forte you can deploy and master BackupPC, which is a backup app written for the enterprise and uses rsync in the background.



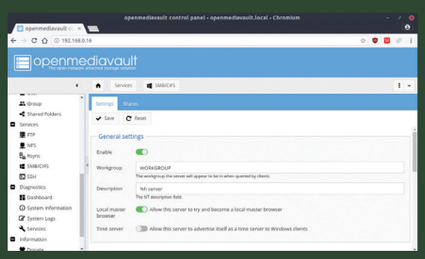
4 Use RAID

Instead of using the disks individually, OMV can tie them into a software RAID. Head to Storage > RAID Management and select the disks you want to use in the RAID, as well as the RAID level. Wait for the RAID to initialise before creating a file system.



5 Users and shares

Head to Access Rights Management > User to add or import users. Next, you'll have to add shared folders from the Access Rights Management > Shared Folders menu. Use the pull-down menu to select the volume for the folder.



6 Enable shares

Finally, enable a network service – which users will use to access the shared folders – under the Services section, and register the shared folders with the service under the Shares tab. Your users can now access these shared folders across your network.

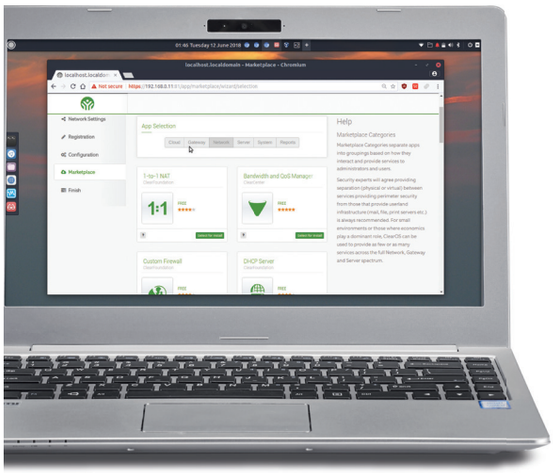
Network essentials

Networks are a crucial building block of a modern day organisation, and managing them well is an essential skill

Although administrators interact with real-world network hardware less frequently than they once did, familiarity with traditional networking still remains a crucial skill. In addition to grasping the underlying technology and the protocols at play, you must know how to configure your network peripherals and how to troubleshoot the connections. You should be familiar with the process of configuring networking in Linux and also have knowledge of the configuration files involved.

The nature and composition of your organisation will heavily influence how you set up your network. A business that is starting out often has only one main server that pretty much provides all the networking infrastructure: a DNS, DHCP, file, mail and web server all rolled into one. But very few larger businesses would trust their entire IT infrastructure to an individual host. Most admins would avoid such a single point of failure, but a small business can't afford to have a host for the individual service.

Irrespective of the size of the network you manage, the one networking skill that you must grasp firmly is tuning the firewall. The Linux kernel uses Netfilter to facilitate key network process such as Network Address Translation (NAT), packet filtering and packet mangling. The `iptables` command is the user-space management tool for Netfilter. Many distributions have graphical tools for configuring the Linux firewall and the excellent Shorewall utility (**http://shorewall.org**) makes it easy to configure the firewall.



Above You can use server distributions such as ClearOS to deploy network services with ease

> "The one networking skill that you must grasp firmly is tuning the firewall"

Still, do yourself a favour and spend time tinkering with the `iptables` command to really appreciate its usefulness.

Another aspect of the modern workspace is that it extends beyond the confines of the physical office. If your users are not located locally, you need some way of connecting them as if they were local by setting up a virtual private network (VPN). A VPN is, in essence, a private network that runs over a public network, and one of the best tools to help you get the job done is OpenVPN. The package is available in the official repositories of all mainstream Linux server distributions. Once you have it up and running, spend some time configuring it to securely expose the resources in your organisation's network to other users over a public network.

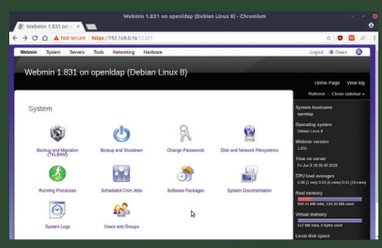
While it's always a good idea to configure and roll out these key network infrastructure components manually, there are several specialist distributions such as ClearOS (**www.clearos.com**) and NethServer (**www.nethserver.org**) that are designed to deploy many of these common services.

Shape traffic

Netfilter is a complex firewall application that can regulate network traffic and also shape it. Shaping network traffic or packets involves altering the stream of packets passing through the network. Admins usually use it to regulate the bandwidth for certain types of connections. A simpler utility to

QUICK GUIDE
Master Webmin

Webmin (**www.webmin.com**) is a configuration tool that can be used to control all aspects of your remote server, such as setting up a cron job, reading logs and managing running processes. Using it you can dispense your admin duties from a web interface. Instead of manually editing configuration files and fiddling with command line switches, Webmin helps you configure different aspects of your system, which then automatically update the relevant underlying config files. Webmin can manage network services as well as the host system. For instance, you can use the tool's interface to create and configure virtual hosts for the Apache Web server, and set up a Samba file-sharing server just as easily as you can create and manage user accounts and set up disk quotas. Webmin's intuitive dashboard contains a list of modules each of which is responsible for managing some service or server, such as the Apache web server, the firewall, software packages and so on. When installed, Webmin reads config files for all servers and services on your system from their standard install locations.



Above Webmin saves you the trouble of having to memorise numerous parameters

shape bandwidth is trickle, with which you can manage the upload and download speeds of applications in order to prevent one from hogging all the bandwidth.

Trickle regulates speeds by delaying the data transferred over a socket. Trickle provides an alternate version of the BSD socket API, with the effect that socket calls are now handled by trickle. Speed is limited by controlling the amount of the data written to or read from a socket.

You can run trickle in either the standalone mode for shaping individual downloads, or configure the trickle daemon to set global limitations and priorities. For example, the following command

```
$ trickle -s -d 30 ncftpget ftp://ftp.foo.bar/Linux/somedistro.iso
```

invokes trickle in standalone mode (`-s`) and sets the download rate to 30KB/s for downloading the specified ISO image file, using the `ncftp` non-interactive FTP download client. You can use the trickle daemon to set global limitations and priorities. When you invoke trickle without the `-s` switch it checks to see if the trickled daemon is running, reads the `/etc/trickled.conf` file and applies any rules it finds listed there. By default the file has an example entry that'll make sense with a cursory look at trickle's man page. Browsing through the file, you'll realise that trickle can do a lot more than just set upload and download speeds. It can also define a per-application priority, along with time- and length-smoothing parameters. Time-smoothing enables you to define the time interval for the application to transfer data. Large values will produce bursts in sending and receiving data, while smaller values ensure a smooth and continuous data transfer.

> "Shaping network traffic or packets involves altering the stream of packets"

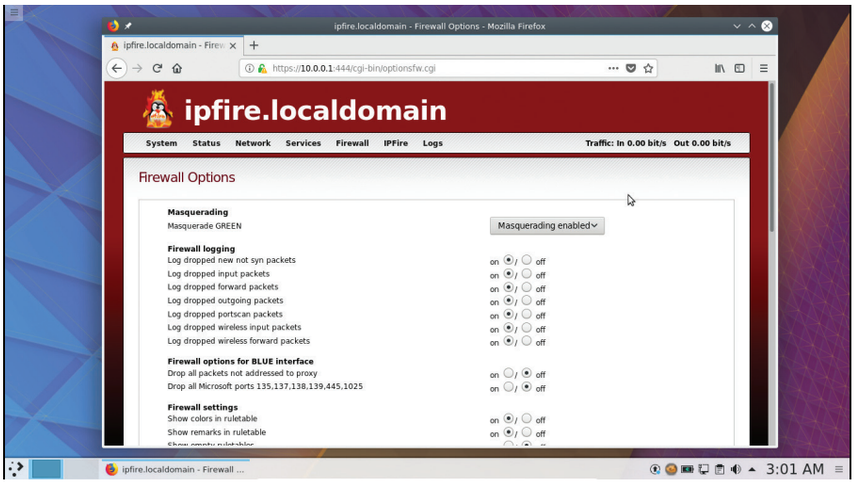
Trickle does have some limitations. For one it can only work on TCP connections, so you can't use it to regulate a UDP stream connection such as DNS. Furthermore, since it uses dynamic linking and loading, it can only work with applications that use dynamic libraries (Glibc). Statically linked applications are thus not compatible with trickle. To determine if you can use trickle with an app, use the `ldd` command, which gives you a list of all shared libraries. For example:

```
$ ldd /usr/bin/ncftp | grep libc.so
```

EXPERT TIP

No slashing

When using rsync, don't put a trailing slash at the end of the source directory. This copies the directory along with its contents, instead of scattering the files inside the backup directory.

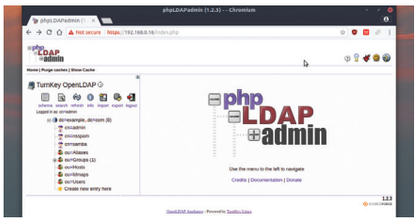


Above Instead of configuring iptables manually, you can use a specially tuned firewall distribution such as IPFire (pictured above, **www.ipfire.org**) or the graphical Shorewall utility

PRODUCTS

Key infrastructure tech

Learn to administer these four useful server apps



1 OpenLDAP

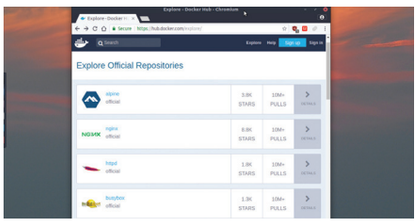
www.openldap.org

All companies need a directory server and OpenLDAP is the popular open source implementation of the Lightweight Directory Access Protocol (LDAP). Despite being lightweight, it's a fully featured suite of apps and development tools.

2 Kolab

www.kolab.org

Kolab is an open source, unified communications, collaboration and groupware suite. It's built with security, privacy and user control in mind and provides solutions for an email server, directory service, web service, Calendar, Tasks, address books and more.



3 Docker

www.docker.com

The shiny new and popular piece of tech for managing and running apps in virtual containers is also a boon for admins. You can use it to deploy conflict-free and production-like development environments for your users.

4 MariaDB

https://mariadb.org

Developed by the original authors of MySQL, MariaDB is one of the most popular database servers in the world. Many companies use it directly or via apps built around it, so it pays to make sure you are familiar with it.

Automation

Smart admins are lazy – they automate mundane repetitive tasks using scripts

As a savvy admin you should try to automate all your regular tasks, such as flushing caches and removing temp files, locking out idle accounts, backups and so on, using scripts. Scripts standardise these administrative chores and free up admins' time for more pressing tasks. You should spend time hashing out a plan to perform repetitive tasks with as little involvement or intervention as possible.

Bash and Python are two of the most popular scripting languages for automating system administration tasks. Generally speaking, lengthier scripts for important tasks such as backups and user resets are executed on a schedule. To that end, spend some time getting familiar with the time-based job scheduler, cron.

> "The best work is work you don't have to do. Eliminating work isn't lazy, it's smart"

Another task that you must automate is the installation and maintenance of the computers in your network. Manual installs don't scale and are prone to errors, especially when done repeatedly. Fully Automatic Installation (FAI, **https://fai-project.org**) is a set of Perl scripts that enables you to run an unattended Debian install. Similarly, Kickstart helps you automate the installation of RPM-based systems such as CentOS and Fedora.

EXPERT TIP

Matt Simmons on scripting

The best work is work that you don't have to do. Eliminating work isn't lazy, it's smart. Cut extraneous work and concentrate on only that which needs to be done.

Kickstart can install not only the operating system but also all the applications you expect to run.

To make everything run as quickly as possible, you can store Kickstart files and Linux packages on a local Apache web server, which then serves as your network installation server. To build your installation server, start by installing a barebones CentOS system. Once the hardware and operating system are set up, make sure the server has a fixed IP address, say 192.168.1.100. Then install the Apache web server from the distribution's repositories. Next, copy over the installation files from the CentOS DVDs to this server with something like `rsync -arv /media/CentOS_7 /install`.

Ideally you should set up the `/install` partition on a separate disk or a logical volume for more flexibility. If you create the install directory outside Apache's document root directory (`/var/www/html/`), you must create a configuration file to point the web server to the correct directory, such as:

```
# nano /etc/httpd/conf.d/install.conf
Alias /install/ /install/
<directory /install>
        Options Indexes
        AllowOverride None
        Order allow,deny
        Allow from all
</directory>
```

Before you can access your install server you'll have to make sure the firewall on the machines in your network allow HTTP traffic over port 80 using `iptables -I INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT`.

Other machines on your network will now be able to connect to the installation server. Next you need a kickstart file to automate the installation. The most convenient way to create one is to use the graphical Kickstart Configurator tool from the repositories. Besides the package selection, you need to pay attention to two

PRODUCTS

Automate admin tasks

Conduct admin business over several machines with Fabric's Python library

```
from fabric.api import run
def uptime():
        run ('uptime')
```

1 Define a function
Enter these lines in a file called fabfile.py and then call the function with `fab uptime`. You can also run functions on multiple remote servers with `fab -H localhost,192.168.0.100`

2 Change context
Fabric's context managers are used with Python's `with` statement. You can use the settings context manager if you temporarily need to run a command as a different user, such as `with settings (sudo_user='mysql'):`.

```
from fabric.api import *
env.hosts = ['192.168.0.100','192.168.0.101']
def upgrade_all ():
        sudo ("apt update")
        sudo ("apt -y upgrade")
```

3 Update several machines
The env.hosts variable lists all computers in your network. When you call the function with `fab upgrade_all` it'll connect to the remote machines, refresh their repos and install any updates.

```
from datetime import datetime, date, time
now = datetime.now()
def fetch_logs():
        get (remote_path="/tmp/logs.tar.gz", local_path="/logs/{}_log.tar.gz".format(now))
```

4 Download remote logs
The get command is useful for downloading logs from a remote system. This example saves them to the local PC with a timestamped file name.

important things in the kickstart files to work with an installation server: the installation method and the network setup.

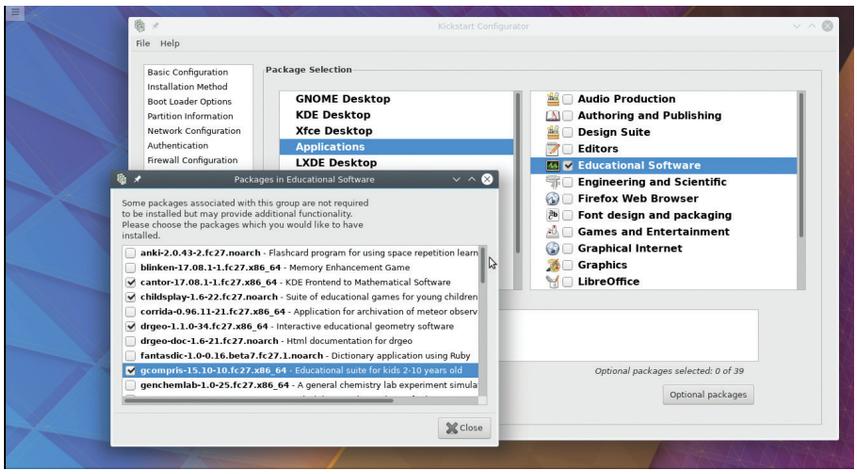
The clients we'll be setting up should fetch packages from the network server we've just set up, instead of from the CentOS installation DVD. To change the installation medium, head to the Installation Method section in the Kickstart Configurator, toggle the HTTP setting and enter the location of the installation server – **http://192.168.1.100/install/** – in the adjacent textbox. Then jump to the Network Configuration section and make sure that the network device you add uses the DHCP server.

Place the generated kickstart files under the web server's document root directory, in a directory such as `/var/www/html/kickstarts`. To install a client using the kickstart files, boot from the minimal boot CD, and at the boot screen point to the kickstart file you want to use, such as `linux ks=http://192.168.1.100/kickstarts/gnome-workstation.cfg`. Once you have everything set up, rolling out new machines

> "The clients we'll be setting up should fetch packages from the network server"

becomes just a matter of pointing to the correct kickstart file. You can set up new machines, reset old ones for new employees, or change a web server into a mail server with a single command.

While the installation is automated, you still have to initiate it manually. The key to fully automating the process is the Pre-Execution Environment (PXE). A PXE server acts as the network boot server and broadcasts a DHCP 'discover' request that includes the name of the boot server and the boot file. The PXE-enabled remote client responds to the request, downloads the boot file and then executes it to begin the installation, which itself has already been automated by you.



Above Generate as many kickstart files as you want, each with different package selections

Configuration tools

That takes care of the deployment. But what if you have to make a configuration change in a deployed machine, such as the firewall change we listed earlier to allow incoming HTTP traffic? There are several tools including Puppet, Chef, Foreman, SpaceWalk, Ansible and Fabric that help you orchestrate various configuration operations.

The advantages of Fabric (**www.fabfile.org**) and Ansible (**www.ansible.com**) is that unlike the other solutions these two have no server daemons and are agent-less. In essence they're really just a set of commands that you install on any system from which you wish to manage clients. Both Fabric and Ansible rely on SSH to conduct their business, and you can install them using your distribution's package manager.

One difference between Fabric and Ansible is that you can get started with Fabric without much delay, while with Ansible you'll have to spend some time setting it up. However, many experienced admins consider Ansible to be ideal for large and complex networks since it can better model multi-tier infrastructure. On the other hand, Fabric uses Python for authoring which is much simpler to understand

EXPERT TIP

Abhas Abhinav, DeepRoot Linux

Automate everything that you would need to do more than once. After five times, generalise it. And then, share it under a free-software licence.

PRODUCTS

Run automated installs

Save time and effort by automatically provisioning machines using these solutions

1 MAAS
www.ubuntu.com/server/maas
Canonical's MAAS is easy to set up and can provision Ubuntu, CentOS, RHEL and SUSE servers. It was designed to compliment Canonical's service orchestration framework called Juju that enables you to easily deploy services with its charms architecture.

2 Foreman
www.theforeman.org
Foreman's a popular choice for provisioning machines and can deploy lots of mainstream distributions. One of the reasons for Foreman's popularity is interoperability; it can work with a host of popular application provisioning tools such as Puppet and Chef.

3 Fog
https://fogproject.org
With Fog you can image an installation and deploy it to other computers on the network. It also makes light of regular administration tasks such as installing software, and can manage large networks that may be spread over multiple locations.

Cloud & virtualisation

Learn and master the new ubiquitous tech that has completely revamped modern IT infrastructure

Perhaps the one biggest reason for the move towards computing in the cloud is convenience. You get technically superior access without the overhead of maintenance and upkeep. Cloud computing offers several other advantages to the modern IT infrastructure, and like any other resource on the network, it has to be looked after by the system administrator.

There are various models of cloud services. Of these, Infrastructure-as-a-Service (IaaS) is the most pertinent to system administrators. IaaS enables businesses to purchase resources on demand, as and when required, instead of having to buy hardware outright. These are typically delivered in the form of virtual private servers (VPS) that are made up of virtualised peripherals. You should have the skills to evaluate IaaS providers and select the one that best meets the requirements of your organisation.

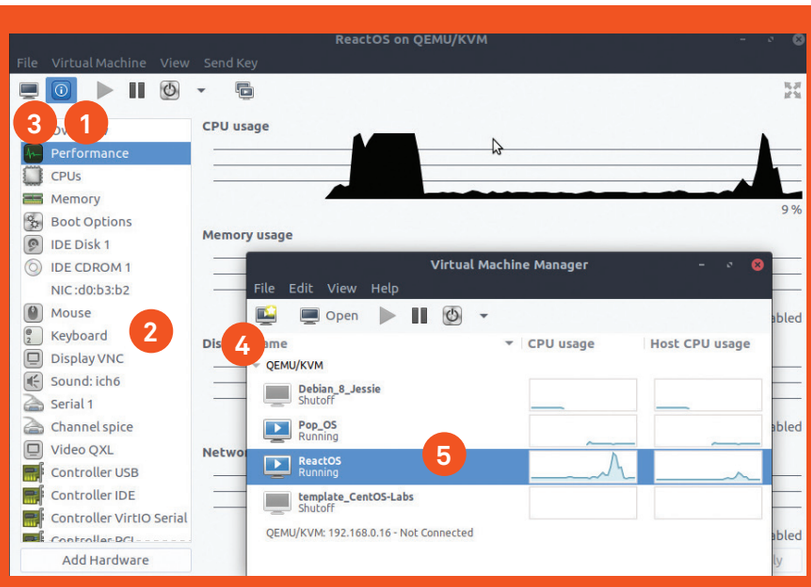
Virtually all cloud vendors have a web-based graphical interface for interacting with the virtualised resources. The good ones even have APIs that enable you to access all their functionality via scripts for automation.

> "VirtualBox's VBoxManage CLI can be used to automate essential VM tasks"

The best way to get to grips with this new dimension of the IT infrastructure is to sign up for an account and leaf through their APIs to understand how to integrate them with your scripts and other automation tasks.

KVM, or Kernel-based Virtual Machine, has been the default hypervisor on Linux since 2007. It is a set of kernel modules that when loaded converts a Linux server into a hypervisor. KVM depends on libvirt, which provides a convenient way to manage VMs and other virtualisation functionality, such as storage and network interface management. You can only create KVM-based VMs on servers that have processors with hardware virtualisation extensions, either AMD-V or Intel VT-x. VirtualBox (**www.virtualbox.org**) is another hypervisor that you'll need to familiarise yourself with for the sake of end users on your network. The good thing about VirtualBox is its VBoxManage CLI that you can use to automate essential VM tasks such as snapshots.

The more useful virtualisation tech you must get to grips with is Linux containers, especially Docker (**www.docker.com**). Docker enables you to bundle any Linux app, with all its dependencies, into its own environment.



QUICK GUIDE
Manage KVM with virt-manager

The Virtual Machine Manager is the open source graphical front-end for creating KVM-based VMs, and is available as `virt-manager` in the repositories of most distributions. The app uses the qemu-kvm hypervisor, which is a version of the QEMU machine emulator modified by the KVM developers.

The virt-manager app is written in Python and is very intuitive to operate. You can use the app to easily create new VMs, monitor them and make configuration changes. virt-manager also includes a VNC and SPICE client that displays a full graphical console to the running VM.

1 **VM details** Click this button to view details about the virtual hardware attached to the VM

2 **Virtual Hardware** You can configure virtual peripherals or add new ones

3 **Graphical console** Use this to peek inside a running VM via VNC or SPICE

4 **Create VM:** This button brings up a five-step wizard to create new VMs

5 **List of VMs:** virt-manager's main dashboard displays all added VMs running on local and remote hypervisors

Above You can grab a host of Debian-based appliances that have been optimised for virtual machines and the cloud from **www.turnkeylinux.org**

You can then run multiple instances of the containerised app, each as a completely isolated and separated process, with near-native runtime performance.

Provision virtual

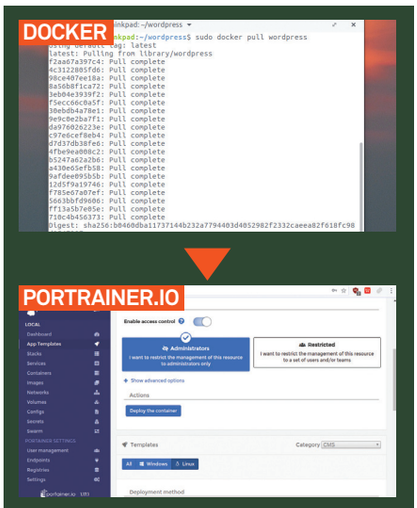
Next you should familiarise yourself with Vagrant (**www.vagrantup.com**), which helps you make consistent virtual environments available to your users with a few keystrokes. Vagrant supports all major virtual platforms such as VirtualBox and VMWare, and plays nicely with all the well-known software configuration tools such as Chef, Puppet, Ansible, Fabric and more.

To get started with Vagrant, fetch and install the latest binary from its website. You'll now have to create a Vagrant configuration file that defines all the characteristics of the VM from a template. You can search for templates based on various operating systems and pre-defined purposes from **https://app.vagrantup.com/boxes/search**. So for example, `vagrant init centos/7` creates a VM based on the CentOS 7.5.1804 distro.

The command creates a file called Vagrantfile under the current directory, and is the main configuration file that defines all the attributes of the VM. If you want to make changes to a VM, you'll need to edit it. You'll need to be well-versed with the anatomy of a Vagrantfile in order to modify or create one as per your needs. For now just use `vagrant up` to create an actual VM from this file. This tells Vagrant to create a new VirtualBox machine based on the base image specified in the Vagrantfile. It'll do so by copying the virtual hard disk files from the remote server.

> "You'll need to be well-versed with the anatomy of a Vagrantfile to edit it"

Once it's done, you'll have a fully featured CentOS 7 VM running headless in the background. Use the `vagrant ssh` command to connect to the machine via `ssh`. You can now interact with the VM like any other CentOS installation. When you're done, type `exit` to drop back to your host and use `vagrant halt` to turn off the VM.



QUICK TIP

Docker and Portainer.io

Using Docker via the terminal isn't all that cumbersome and the tool is well documented. To make your life easier you can use Portrainer.io (**https://portainer.io**), which is an open source, web-based graphical front-end that supports all features exposed by the Docker API.

You can now modify the Vagrantfile to customise the installation. If you are running a web server inside the VM, you'll want to forward port 80 by adding this line:

```
config.vm.network "forwarded_port", guest: 80, host: 8080
```

You'll now be able to access port 80 on the guest via port 8080 on the host. Type `vagrant up` to start the VM once again. You can also make changes to the Vagrantfile while the VM is running; in that case, type `vagrant reload` to load the VM with the modified settings. Similarly, you can put all the operations you might do with a freshly minted CentOS box – such as installing some apps – inside a Bash shell script and then point to it from the Vagrantfile with:

```
config.vm.provision "shell", path: "post-install.sh"
```

Now bring up or reload the VM. Thanks to this line, Vagrant will execute the `post-install.sh` script after the VM is up and running. There's a lot more you can do with Vagrant, though, so make sure you read through its documentation.

Logging & monitoring

Be thorough and take steps to ensure all the IT infrastructure in your realm keeps chugging along smoothly

Everything from databases to daemons keep logs of their activity and managing these is as important a function as deploying servers. It's the responsibility of system administrators to gather actionable bits of information from this deluge of messages.

Log data usually ends up in the `/var/log` directory in a variety of files usually placed by the syslog daemon. Syslog has been the standard Unix format for several years, but there have been a couple of replacements, with rsyslog being the most recent and popular one. Make sure you are fully versed with the configuration of the logging daemon on your servers.

Another aspect of managing logs is to control their size and archive duration, which for some types files such as access logs is mandated by regulatory policies. To help you keep the logs manageable you need to rotate them, which involves flushing the old ones to an archive file. The logrotate utility is a popular choice for this task as it implements a variety of log management policies.

A good sysad should be fully aware of the limited usefulness of this data, in terms of time, and must automate the process of filtering and summarising it using tools such as Logwatch (see right). Furthermore, it's important to note that the **syslogd** daemon is a passive tool that awaits inputs from apps and doesn't go out and actively query them. For this you need a monitoring tool

> "Syslog has been the standard Unix format for several years"

like Nagios (**www.nagios.org**), one of the most popular and extensively used network monitoring tools, which you can use to streamline the management and monitoring of your network. You can use it monitor nearly all devices and services that have an address and that can be contacted via TCP/IP. ■

QUICK GUIDE

Analyse with Logwatch

One of the least interesting part of an admin's job is to scroll through logs in order to spot potential issues. This is where log filtering tools like Logwatch come in. It parses, analyses and filters logs, and then generates daily reports on your system's log activity. The utility is available in the official repositories of all popular server distributions, including CentOS and Debian.

Above Logwatch reports are categorised by services; you can control the level of verbosity

HOW TO

Use osquery

```
SELECT * from logged_in_users;
SELECT * from last;
```

1 Check on users
You can use osquery to keep an eye on the users logged into your servers with the first command shown above, while the second shows the list of previous logins. Look for logins from unknown IP addresses, and even more so if there are multiple users logging in from an unfamiliar host, which should be a red flag.

```
SELECT * from iptables;
SELECT * from listening_ports;
```

2 Check firewall
These queries help keep an eye on the firewall. If the first one doesn't produce any output it means there's no firewall – which isn't a good thing, especially on a server. The second lists all the listening ports and will help you find back-doors on the server.

```
SELECT * from kernel_info;
SELECT name, size, used_by, status from kernel_modules where status="Live" order by size;
```

3 Check kernel
You'll also want to run these queries periodically and compare their output against older results for any changes. The first retrieves information about the current kernel to help identify outdated kernels, while the second lists all loaded kernel modules.

```
SELECT pid, name, uid, resident_size from processes order by resident_size desc limit 10;
SELECT name, path, pid FROM processes WHERE on_disk = 0;
```

4 Watch processes
The first query displays the 10 largest processes arranged by size, while the second displays processes that don't have a binary associated with them. You should immediately terminate that process.