Deployment - post Xserve

Pascal Robert
Miguel Arroz
David LeBer

The Menu

- Deployment options
- Deployment on CentOS Linux
- Deployment on Ubuntu Linux
- Deployment on BSD

Hardware/environment options

Choices

- Using your own hardware
- Leasing the hardware
- Virtual machines (VMWare ESXi/Xen) / VPS (Slicehost, Linode)
- Cloud hosting (Amazon EC2/Windows Azure/RackSpace)

Your own hardware

- Pros
  - It can be cheaper, if you use the hardware to its full potential.
  - You can resell it.
  - You do whatever you want.
- Cons
  - You have to manage everything yourself.
  - Must get a good support contract in case of hardware problems
  - Not cost effective if you don't need a lot of processing power.

Leasing the hardware

- Pros
  - The provider will take care of hardware problems, with resonable SLA.
  - You can buy software support, including backup solutions.
  - No big upfront cost, can pay per month.
- Cons
  - Still have to manage the operating system yourself.
  - Less hardware and software support.
  - Can cost more in the long run.

Virtual machines/VPS

- Pros:
  - You can isolate customers by using virtual machines.
  - Can create your own virtual environment on your own or leased hardware (Xen, VMWare ESX, KVM, etc.), or get VMs (VPS) on a hosted partner (Slicehost, Linode, etc.)
  - Easy to allocate more ressources to the VMs.
  - Snapshots!
- Cons:
  - Can get pricy, especially for Virtual Private Server.
  - CPU is shared for all hosts on the physical server.

Cloud hosting (virtual machines on steroids)

- Pros:
  - Tons of options. Example: load balancer,
  - Can be cheap if you don't need CPU or bandwith all of time.
- Cons:
  - Can get very pricy if you use a lot of resources (bandwith, CPU, memory)

Price comparaison

- One CPU, 2 GB of RAM, 64 GB disk space, 700 GB bandwith/ month
  - Leased hardware (iWeb.com): $99 (with 320 GB of storage)
  - VPS (Linode.com): $79.95
  - Amazon EC2: $125.96 (1.7 GB of RAM)

Other things to check

- 32 bits vs 64 bits
- "Commercial software"
- Adding volumes (LVM)
- ... memory

Memory

- If using a virtual machine, be it Amazon EC2, Xen or otherwise, check for memory usage of your app!
- Amazon Linux don't have a swap partition!
- On a 64 bits system, a single instance of an application can take up to 1.5 GB of memory!
- A "micro" instance of Amazon Linux (32 bits) with Apache, wotaskd and JavaMonitor will eat up 187 MB of RAM.

Memory

- Use the Xmx parameters to make sure your apps would not start using all "real" and "virtual" memory.
- Monitoring the heap space of your instances to see if you need more memory.
- For Amazon Linux: add a swap partition.
- Use a 32 bits system if you only need a VM with less than 1.5 GB of RAM.

RedHat/CentOS/Amazon vs Ubuntu/Debian

- RedHat Enterprise Linux is a "stable" release of work done in the Fedora project + support.
- CentOS is the "free as in beer" clone of RedHat.
- Amazon Linux is based on RedHat.
- Debian is another distribution that is there for a long time.
- Ubuntu is a deriative of Debian.

Which distro to use?

- If you need to install commercial software, go with RedHat or CentOS.
- CentOS is also more « stable » but packages can be very old (ex: PHP).
- Ubuntu is the cool kid, and packages are more current.
- Ubuntu Server LTS have support for 5 years. RedHat have support for 7 years.
- CentOS major releases take more time to get out than RedHat.

RedHat/CentOS Linux Primer

Installing software on RedHat/ CentOS

- Use the RPM package when possible.
  - rpm --install software.rpm
- You can find other software on RPM Forge (http://rpmrepo.org/RPMforge)
- On CentOS, you can also use « yum » to get software from the CentOS and other repositories.
  - yum info sofware-name
  - yum install software-name

Starting/stopping services

- Init scripts are in /etc/init.d
- To start a service:
  - service serviceName start
- To stop it:
  - service serviceName stop
- To mark it to start at reboot:
  - chkconfig serviceName on

Network configuration

- Network scripts are in /etc/sysconfig/network-scripts
  - If you do change, you have to restart the network script:
    - sh /etc/init.d/network restart
- DNS resolver configuration file is /etc/resolv.conf (put your nameservers IP in there).
- You can use the Network control panel too.
  - command line: system-config-network-tui
  - GUI (X11): system-config-network

GUI

- By default, RedHat/CentOS will start in GUI mode, which will use some RAM. To disable the GUI when starting up, edit /etc/inittab to put it in level 3 instead of 5.
- Even if the GUI is not started, you can still start GUI apps remotely.
  - ssh -X user@host

User management

- To create a user:
  - useradd -d /path/to/user/home -g main_group -G other_groups username
  - passwd username
- To modify a user, use « usermod », to delete one, use « userdel ».
- To change a password of another user:
  - passwd username
    (with no argument, it will change your own password)
- GUI tool: system-config-users

Unneeded packages

- Check that you are not running extra stuff that you don't need (sendmail, Samba, etc.)
- You can get a list of started services with:
  - chkconfig --list | grep "on"
- Check their description in the init.d script to see if you really need it.

Unneeded Apache modules

- You should also disable unneeded Apache modules. Get the list of modules with:
  - httpd -M
- You can delete unneeded module installed by RedHat/CentOS with Yum:
  - yum provides "mod_cgi.so"
  - yum erase mod_perl
- Apache configuration files are in /etc/httpd/conf and /etc/httpd/conf.d

Installing WO on RedHat/CentOS Linux

Installing a JVM

- You can use OpenJDK 1.6
  - yum install java-1.6.0-openjdk
  - ... but some other software (ex: Atlassian) doesn't work well with OpenJDK, so it's better to get the JVM from Oracle.
- Oracle JVM install itself into /usr/java
- To manage the JVMs, use « alternatives ».
  - alternatives --install /usr/bin/java java /usr/java/default/bin/java 2
  - alternatives --config

Installing wotaskd and Monitor

- Make sure you have Apache on the system. If not, you can install it with:
  - yum install httpd httpd-devel
  - Amazon Linux: beware, Apache is not installed by default
- Follow the rest of the instructions from the wiki

Monitoring performance

top/free/vmstat

- top: shows which processes are taking the most memory or CPU. Nice summary of load.
- free: shows how much RAM and swap space is available.
- vmtstat: good way to monitor RAM and I/O.
- lsof: finding which resources are used by a process

JMX

- Use JMX to monitor CPU and heap space usage.
- Nagios is your friend (again).

Security

SSH

- Configuration file on the server is /etc/ssh/sshd_config
- Disable root login ("PermitRootLogin" directive)
- Disable SSH v1 ("Protocol 2")
- Allow only specific users
  - AllowUsers user1 user2 user3
- Run the server on a different port ("Port 2345")
- Disable password authentification and use public/private keys.
  - PasswordAuthentication no

iptables

- Software firewall included in RedHat/CentOS for a long time.
- To list firewall rules:
  - /sbin/iptables --list
- To save them in a text file:
  - /sbin/iptables-save > somefile.txt
- To restore them from the text file:
  - /sbin/iptables-restore < somefile.txt

iptables

- To block 1085 from the external network:
  - /sbin/iptables -A INPUT -i eth0 -p tcp -m tcp --dport 1085 -j DROP
  - /sbin/iptables -A INPUT -i eth0 -p udp -m udp --dport 1085 -j DROP

Protecting from brute force attacks

- SSH password brute force attacks are common
  - ... and IMAP/POP3 brute force attacks are more and more popular too
- If you can't disable SSH password authentification, use iptables to block IPs that are doing too much SSH requests for a given period

logwatch

- Useful tool to get a summary of common hack attempts
- Will generate a nightly summary of various system logs, including Apache error log
- It's also available for other platforms than Linux

SSH tunnels

- Don't allow access to JavaMonitor and your database servers from the outside world! Use SSH tunnels instead
- SSH tunnel will map a local port with a remote server
- Example, to access a remote PostgreSQL server and make it available on port 55432 on your system:
  - ssh -fNg -L 55432:127.0.0.1:5432 user@yourserver.com

SELinux

- Policies-based security system
- Apps are allowed to read/write only to specific paths
- Can be a PITA to configure
- Put SELinux in permissive mode first, check the warnings, fix them, put it on enforcing mode.

chroot

- Basic isolation
- Put a user into its own environnement
- User won't be able to navigate to other users or system directories, think FTP chroot
- Use "jailkit" to ease the pain a bit
- Is a PITA when doing OS updates (you have to update the libs and binaries of each user's chroot)

OpenVZ

- chroot on steroids
- Think of Solaris Zones and BSD jails
- Will run a copy of Linux userland for each "VZ" , including its own root user
- Can only run Linux

Resources

- http://wiki.centos.org/HowTos/Network/SecuringSSH
- http://www.nsa.gov/ia/_files/os/redhat/rhel5-guide-i731.pdf
- http://wiki.objectstyle.org/confluence/x/CYE5
- http://wiki.openvz.org/Main_Page
- http://olivier.sessink.nl/jailkit/
- http://sourceforge.net/projects/logwatch/