Mar 30th, 1:15 PM - 2:00 PM

Managing Self-Hosted Open Source Systems with Virtual Machine Test Environments

Jeremiah Kellogg
Eastern Oregon University, jkellogg@eou.edu

EASTERN OREGON UNIVERSITY

Building Virtual Test Environments for Open Source Library Web Applications

Jeremiah Kellogg
Systems Librarian
Pierce Library

jkellogg@eou.edu

Helpful Resources for Beginners

Learn the Command Line at Code Academy:
https://www.codecademy.com/learn/learn-the-command-line

Tutorials on YouTube:
https://www.youtube.com/channel/UCVzKEg6a7Ads2VZE9_y2Aaw

Get Free Training on Linux from the Linux Foundation:
https://training.linuxfoundation.org/free-linux-training

Read the VirtualBox User Manual:
https://www.virtualbox.org/manual/

Familiarize Yourself with Linux apt Repositories using Ubuntu Documentation:
https://help.ubuntu.com/community/Repositories/CommandLine

Learn more about Linux Repositories at How-To Geek:
https://www.howtogeek.com/117579/htg-explains-how-software-installation-package-managers-work-on-linux/

Why Open Source?

Linux

APACHE HTTP SERVER

MySQL

php

WordPress

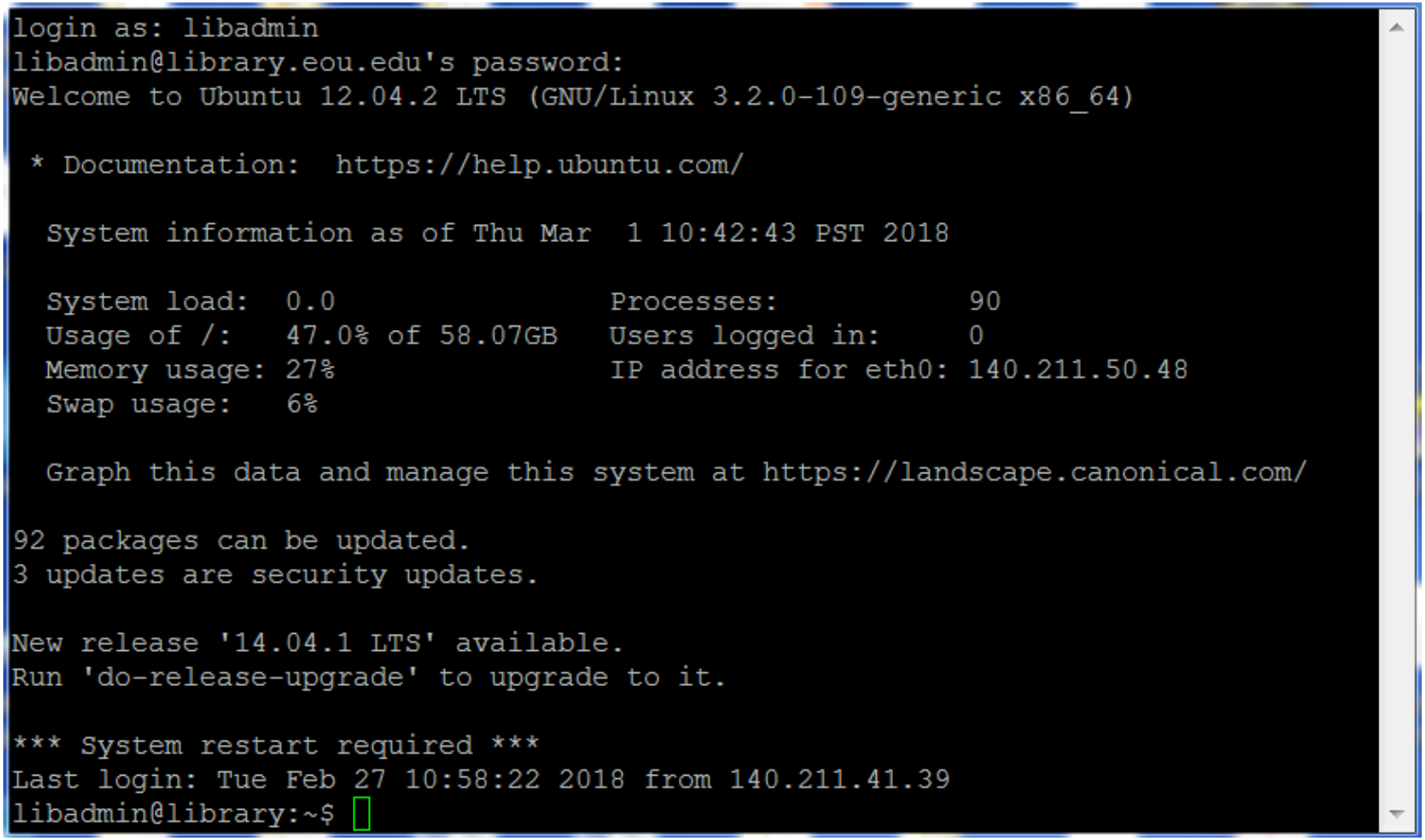
D SPACE

Guide on the Side

ARIZONA

omeka

Why Use Test Environments?



Don’t let your updates make your web app go from this…

…to this

EOU Digital Archives

Enter search terms...

Browse Items Browse Collections

Welcome to the Eastern Oregon University Digital Archives. Here you will find digital copies of materials that are historically important to the University and the surrounding region.

LEARN MORE ABOUT OUR DIGITAL PHOTO COLLECTIONS

Fred Hill WWII Photos

See World War Two through the lens of local resident Fred Hill's camera.

Fred Hill Historical Photos

Experience history with images captured by Eastern Oregonians and collected by local resident Fred Hill over his lifetime.

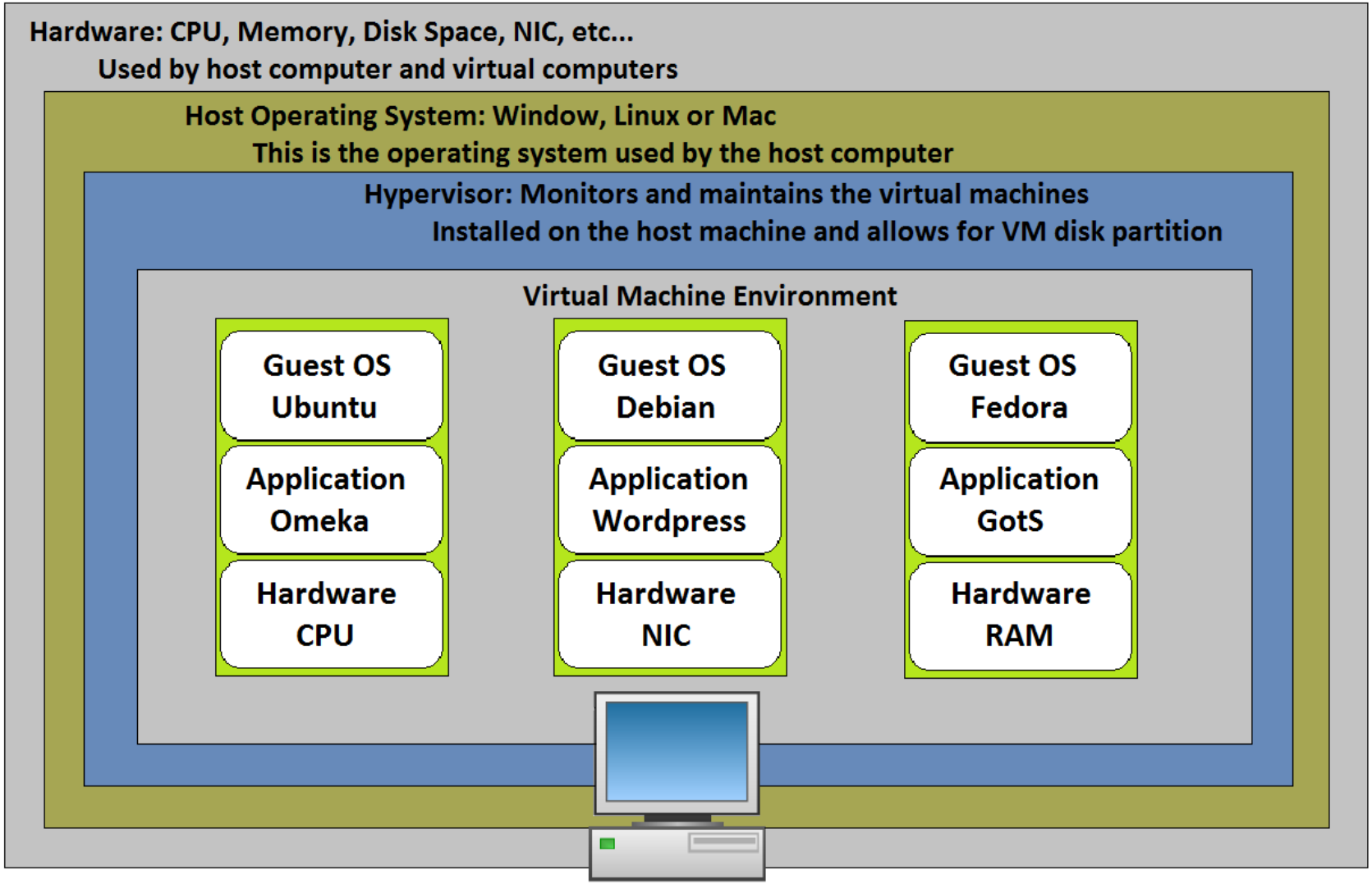
Regional Historical Photos

Explore images of Eastern Oregon between the years of 1860 and 1945.

EOU Historical Photos

Follow EOU's photographic history from as far back as 1929.

What is a Virtual Machine?



An Introduction to VirtualBox

Download the most recent version from the VirtualBox wiki:
https://www.virtualbox.org/wiki/Downloads (Be sure to select the Operating System of your Host computer and not the Virtual Machine).

Follow the Setup Wizard (you'll need admin privileges). There's no need to change default settings so you can keep clicking "Next."

Take a tour of VirtualBox on YouTube:
TechGumbo: https://youtu.be/sB_5fqiysi4 (This video gets right to the point and is very succinct)
My Channel: https://youtu.be/zv51whtHYv8 (This video goes into more detail and covers more concepts related to installing virtual machines).

Build Your Test Environment: Linux

Useful Commands:

It's a good idea to have a look at the install page for your webapp

* Show the IP address of your server:$ **ifconfig –a**
* Show information on the version of Linux you're using: $ **cat /etc/*release**
* Show what repositories you're using: $ **apt-cache policy**
* Find Processor Architecture: $ **uname –a**
* Find Hard Drive size in GB:$ **df –H**
* Find Base Memory (RAM in kb):$ **free**
* Check sources.list on each server:$ **sudo nano /etc/apt/sources.list**
* Move copy of sources.list from production server to test server:$

sudo scp libadmin@10.3.20.27:/etc/apt/sources.list /etc/apt/sources.list

Be sure to run this in your test environment and to replace libadmin with the username for your production server and the 10.3.20.27 with the IP address of your production server.

* Update your repository:$ **sudo apt-get update**
* Run software upgrades:$ **sudo apt-get upgrade**
* Compare lists of installed software: **dpkg -l**

Build Your Test Environment: Apache2

Useful Commands:

*Check what version of Apache2 your repository will install:$

apt-cache showpkg apache2

*Install Apache2 on your test server:$

sudo apt-get install apache2

*Enable mod_rewrite on the test server:$

sudo a2enmod rewrite

*Configure Apache to allow overrides:$

sudo nano /etc/apache2/sites-enabled/000-default

Find the section that looks like this:

```
<Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

Change this line to AllowOverride All

* Restart Apache2:$ **sudo service apache2 restart**

*Fix domain name error (specific to this presentation):

sudo nano /etc/apache2/conf.d/fqdn

Add this text to the empty file: **ServerName localhost**

Build Your Test Environment: MySQL

Useful Commands:

*Check what version of MySQL Server your repository will install:$

apt-cache showpkg mysql-server

*Install MySQL Server on your test server:$

sudo apt-get install mysql-server

*Log in to mysql as root:$

mysql -u root –p

*Create the database your web app requires:$

CREATE DATABASE guide_on_the_side;

*Set permissions for the database required by your web app:$

GRANT ALL ON guide_on_the_side.* TO gots_user@localhost IDENTIFIED BY 'password';

*Exit MySQL:

exit

Build Your Test Environment: PHP

Useful Commands:

*Check what version of PHP your repository will install:$

apt-cache showpkg php*

*Install PHP and necessary components. These commands are specific to PHP5:

Install PHP5:$ **sudo apt-get install php5**

Install GD support:$ **sudo apt-get install php5-gd**

Install Tidy:$ **sudo apt-get install php5-tidy**

Install Apache support (Includes Mbstring):$

sudo apt-get install libapache2-mod-php5

Install JSON support:$ **sudo apt-get install php5-json**

Install MySQL support:$ **sudo apt-get install php5-mysql**

Build Your Test Environment: WebApp

*This part can be tricky and is dependent on how the webapp is installed. For our demonstration we're using Guide on the Side. These steps may work for other applications, too, but it's highly recommended that the webapp documentation is consulted to figure out the best approach.

*Start by copying the MySQL database in your production server to the webapps primary directory. Most web apps will be in the /var/www/html directory. My example is in the /var/www directory because it's on an older server, so I'm going to backup my MySQL database there using this command: $

mysqldump -u root -p guide_on_the_side > guide_on_the_side.sql

*Copy the webapp directory from the production server to the test server. From the test server fun this command:$

sudo scp -r libadmin@10.3.20.27:/var/www /var/

*Populate your test server's MySQL database with the production server's backup:$

mysql -u root -p guide_on_the_side < /var/www/guide_on_the_side.sql

*Set your Linux user as the owner of the /var/www directory:$

sudo chown -R libadmin:libadmin /var/www/

*At this point we should follow instructions on the Guide on the Side website:

https://github.com/ualibraries/Guide-on-the-Side/blob/master/README.md

*Once finished, be sure to run the webserver restart one more time:$

sudo service apache2 restart