Test Environment and Open Source Infrastructure

By

Nate Frary

Submitted to
The faculty of the Information Technology Program
In Partial Fulfillment of the Requirements for
The Degree of Bachelor of Science
In Information Technology

University of Cincinnati
CECH

June 2012
Test Environment and Open Source Infrastructure

By

Nate Frary

Submitted to
The faculty of the Information Technology Program
In Partial Fulfillment of the Requirements for
The Degree of Bachelor of Science
In Information Technology

© Copyright 2012 Nate Frary

The author grants to the Information Technology Program permission to reproduce and distribute copies of this document in whole or in part.

______________________________   __________________
Nate Frary                                               Date

______________________________   __________________
Mark Stockman, Faculty Advisor                                            Date
Acknowledgements

I’d like to extend a special thanks to David Heberling of Peak 10 for his generosity in scheduling my shifts and allowing me space time and equipment to finish this project. My colleague Jim Domask who very willingly helped to cover my work shifts that allowed me to finish. I’d also like to thank Professor Stockman for his advice, assistance, and guidance throughout the project.
Dedication

I’d like to dedicate this project to my longsuffering and understanding wife Amy Frary.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>i</td>
</tr>
<tr>
<td>Dedication</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Illustrations</td>
<td>iv</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>1. Statement of Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Definition of Need</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Provisioning</td>
<td>3</td>
</tr>
<tr>
<td>1.1.2 Customer Options</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Solution Definition</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Users</td>
<td>7</td>
</tr>
<tr>
<td>2. Design Protocols</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Dedicated Test Lab</td>
<td>10</td>
</tr>
<tr>
<td>3.2 Open Source Infrastructure</td>
<td>12</td>
</tr>
<tr>
<td>3. Deliverables</td>
<td>16</td>
</tr>
<tr>
<td>4. Project Plan</td>
<td>18</td>
</tr>
<tr>
<td>4.1 Hardware</td>
<td>18</td>
</tr>
<tr>
<td>4.2 Costs</td>
<td>19</td>
</tr>
<tr>
<td>4.3 Gantt Chart</td>
<td>20</td>
</tr>
<tr>
<td>5. Proof of Design</td>
<td>23</td>
</tr>
<tr>
<td>5.1 Dedicated Testing Environment</td>
<td>23</td>
</tr>
<tr>
<td>5.2 Open Source Small Business Infrastructure</td>
<td>27</td>
</tr>
<tr>
<td>6. Testing</td>
<td>34</td>
</tr>
<tr>
<td>6.1 Testing Scenarios</td>
<td>34</td>
</tr>
<tr>
<td>7. Conclusion</td>
<td>38</td>
</tr>
<tr>
<td>References</td>
<td>39</td>
</tr>
</tbody>
</table>
List of Illustrations

Figure 1. Dedicated Test Environment IT Technician Use Case 8
Figure 2. Small Business Infrastructure Use Case 10
Figure 3. Network Diagram of Testing Environment 11
Figure 4.1. Small Business Infrastructure Network Diagram 13
Figure 4.2. Small Business Infrastructure with Distros 15
Figure 5.1 Page one of Gantt chart 20
Figure 5.2 Page two of Gantt chart 21
Figure 5.3 Page three of Gantt chart 22
Figure 6.1. vSphere Client Login 23
Figure 6.2. vSphere Main Screen 24
Figure 6.3. New Virtual Machine Window 25
Figure 6.4. New Virtual Machine Finished 26
Figure 7.1. Firewall/Router – Untangle 27
Figure 7.2. CentOS 6.2 27
Figure 7.3. Web Server – CentOS 6.2 27
Figure 7.4. Desktop Chromium 27
Figure 8.1. Untangle Front Screen 28
Figure 8.2. Untangle Virtual Rack 28
Figure 8.3. Untangle Firewall Config 29
Figure 8.4. Untangle Networking Config 29
Figure 9.1. Fileserver – NFS Restart 30
Figure 9.2. Fileserver – nfsstat result 30
Figure 9.3 Desktop Access to the Fileserver
Figure 9.4. Remote File on the Fileserver
Figure 10.1. Apache installed on Webserver
Figure 10.2. phpinfo() Page: test.php
Figure 11.1 Page one of test scenarios
Figure 11.2 Page two of test scenarios
Figure 11.3 Page three of test scenarios
Abstract

Dedicated Test Environment and Open Source Infrastructure is a project utilizing virtualization to create a safe testing environment and explore the inexpensive open source options available to IT infrastructures. In the datacenter business, being able to effectively provision new equipment and software for customers is critical to our success. In our target demographic of small to medium businesses, IT budgets are tight and hardware/software can be quite expensive. Virtualization and “The Cloud” have increased the possibility of providing safe, economical, and more flexible solutions to meet everyone’s needs. The ease of setting up virtual servers, networks, and environments provides an ideal solution to set up a dedicated testing environment for the testing of potential customer solutions. The environment also allows us to explore open source software as a viable alternative to commercial software by setting up and testing a proof of concept IT infrastructure made up of entirely free of cost open source software targeted at the budget conscious business owner.
1. Statement of the Problem

I work at the Cincinnati location for a company called Peak 10. Peak 10 builds and manages data centers and has sixteen data centers in nine different markets, mostly in the southeast United States. The customer we typically target is a small to medium business in the local market looking for a secure and reliable data center to host their equipment and software. We also do our best to sell what are called managed services. Managed services encompass such services as monitoring a customer’s equipment for outages, setting up and maintaining their operating systems, firewall setup and monitoring, antivirus installation/maintenance, network configuration, and backups. We also have a robust and growing cloud environment. Our cloud environment allows us through the use of powerful servers and specialized software to create virtual servers for customers that want all the benefits of a physical server but not the hassle of actually having to purchase and maintain it. The nature of our business means that we have multiple companies occupying our data centers and there is no one solution that fits every customer’s needs. (In Cincinnati alone we have 90+ different customers with 90+ different sets of needs). Since we are a customer centered company we take great pains to tailor our solutions to the specific customer needs and their ability to pay for these solutions.

As our business volume has grown, we’ve reached a level of maturity here in Cincinnati where I feel we need to focus on a couple of things:

1. Streamlining how we provision solutions for customers.

2. Exploring other potential software options that might better suit a customer’s needs.
1.1 Definition of the Need

1.1.1 Provisioning

One of the major hurdles to the smooth provisioning a customer is the lack of a dedicated testing environment in which to safely and effectively set up and test potential solutions for customers. The lack of a dedicated testing environment has resulted in us testing possible solutions live or on an ad-hoc basis. The ad-hoc nature also means that we essentially need to spend time building a new testing environment for every new customer solution implemented. Then once implemented, we move onto the next customer and set up an entirely new testing environment and so on. This adds time and money to the cost of provisioning the customer. (Not to mention, the potential cost of failure associated with an inadequately tested solution. (3)(6))

It goes without saying that in business money is critical.

A separate dedicated testing environment would also be very helpful setting up customers in our customer cloud. In Cincinnati, whenever we stand up a new virtual server and operating system in our “Enterprise Cloud” we are setting it up in the production cloud environment. This is where existing customers are already up and running. This presents at least two problems:

1. You have to be very careful with the resources you allocate as anything you take to test the new equipment could be shared by other production servers.

2. However remote, there is the possibility that while changing a test server/OS configuration, or power-cycling a new (or test) server you could accidentally click on the wrong one altogether and change something belonging to a completely different customer.
1.1.2 Customer Options

As I mentioned before, we take great care to tailor potential solutions to customers. Each customer has their own unique and different set of needs. I feel that, while currently, we’ve been able to meet our customer’s needs with the software we offer, we are limited to largely commercial software options that may not always be the perfect solution for their particular needs or ability to pay. For example, of the current operating systems we offer customers all but one of them (CentOS) is an expensive commercial product. While they are all very good products they may not provide the best overall solution for a particular customer. There will always be a place for commercial software in the business world but it’s worth noting that when you buy into the commercial framework you buy into the problems inherent to it.

First of all, commercial software is expensive, especially in licensing and support contracts. The first IT job I ever had was working on a Unisys A-Series mainframe. For the privilege of using the operating system on the mainframe hardware (which my company owned) Unisys charged us $1 million dollars a year. Keep in mind that this was just the software lease and didn’t cover any kind of support.

How about something slightly less intimidating than a mainframe like the flagship of the Microsoft server world, Windows Server 2008 R2? To understand the costs of licensing a Windows Server 2008 R2 instance you can download from Microsoft the “Windows Server, System Center, and Forefront Licensing Guide”. (13) It’s a 58 page .pdf document that spends the first 29 pages explaining all the possible configurations and the variety of ways they can charge for said configurations before it even shows actual prices. Those prices aren’t entirely clear cut either. Page 30 – 34 lists tables of all the various configurations and options you can purchase. However, it seems that to get in on the ground floor with Windows Server 2008 R2
Standard it will cost you $1029.00 retail. This includes five user licenses, support for a single processor, and the ability to run the OS on one physical server and one virtual server on the same physical server simultaneously. The license is good for three years. (13) What if you want Windows 2008 but only have 32 bit hardware? Well, you can always choose Server 2008 which supports 32bit processors. How about pricing? You can download another .PDF called “Microsoft Windows Server 2008 Licensing Guide”. It’s only 39 pages. (It’s the “Quick Reference” after all.) On page 35 you find the retail price of $999.00 for the same setup as before, (five user licenses, single processor, and a three year license contract.)(14) So, for about $1000 dollars you can get in on the ground floor with one server (two if you virtualize another instance on the same physical server). This doesn’t even include other costs like application software, hardware, internet, and electricity. Taking all that into account it would make sense to try to save where you could and if you could save $1000 every three years wouldn’t it be worth it? (11)

Second, when you choose a commercial product you face the potential problem of locking yourself into one type of architecture that may not scale the way you like or provide adequate solutions down the road. I also worked, for three years, on several IBM AS/400 systems. AS/400 is a great system that can run a ton of information and applications. However, the OS is (again) leased to you and it only runs on their expensive hardware. What database does it use? DB2, IBM’s own. (19) The character set used is not ASCII but EBCDIC, developed at IBM. (20) When you go AS/400 you go all in.

Third, locking into a vendor also puts you at their mercy even if you’re perfectly happy with what you have. They still can force you to upgrade and spend even more money. There’s
also a real possibility they might discontinue an unprofitable product that you’re using or, even worse, go out of business. (5)

Fourth, how does the commercial vendor handle stability and security? If and when bugs and exploits are found in commercial software the users of the product are at the mercy of the vendor, “The pattern … is typically that a defect report needs to be filed and then there will be a delay before the vendor determines when or whether to issue an updated release. Users of the software are much more at the mercy of the vendor's internal processes…” (5)(9)

1.2 Solution Definition

For step one to streamlining the provisioning process and increasing customer options, I have set up a dedicated virtual machine based test environment in Cincinnati. It is a combination of hardware and software that allows us to set up, test, and tailor specific hardware and software configurations to meet the exacting needs of our customers and has the flexibility to test almost any configuration of equipment and software that might be needed to meet the aforementioned customer’s needs. (6)(8)(12) It also allows us to experiment with new technologies in a safe and controlled space.

For step two to streamlining the provisioning process and increasing customer options I have utilized the virtualization component of the dedicated testing environment to set up a simple IT infrastructure. In order for it to be relatable and applicable as a potential solution for our customers, the IT infrastructure is built with a small business in mind. In the infrastructure I have loaded (free-of-charge) open source software at each of the critical locations (firewall/router, servers, desktops).
By utilizing all free of cost open source software we directly address the four failings of commercial software:

1. Price
2. Rigid Architecture
3. Loss of Software Support
4. Security and Stability

Luckily, the open source software movement has made that possible. (5)(9) What do I mean by “open source”? Well, the two things that really distinguish open source are these:

1. The source code the software was written in is open and available to anyone to study, and use. (In fact, it is copyrighted in such a way that its source code will always remain open)(1)
2. It’s usually free of cost or very reasonably priced compared to commercial (or “closed source) alternatives. For example, even if the vendor has a commercial version of their product, if it’s based on open source code they typically release a free version that’s slightly scaled back, typically in administrator tools, user interface, and support.)

As I just mentioned, open source software is a viable option to address the four failings of commercial software:

1. Price (5)
   - The upfront cost? $0. There are many free of charge open source options.
   - The need to track number of copies of software to pay expensive license fees is unlikely at best. They may ask you to register an account when downloading directly from the vendor but that’s free of charge too.
• Open source software usually runs quite well on older hardware thus eliminating the need to spend a lot of money on expensive new hardware.

2. Rigid Architecture (5)

• Open Source adheres to common standards in IT making it able to be flexible as business needs change.
• There’s no monetary gain to open source authors to create proprietary standards or formats.

3. Forced upgrades (5)

• In open source software the “availability of source code provides greater continuity and security against financial collapse of vendors of key products.”
• Since no money is being made, the chance of “vendors choosing to withdraw support for unprofitable products” isn’t really a concern either.

4. Security and Stability (5)(9)

• Closed source means only the original developers have the ability to fix a software bug or security hole. Open source software typically has many users, all with the ability to examine the code and note potential flaws or bugs. If any are discovered they are often dealt with in a matter of hours and a fix immediately posted for download.

1.3 Users

There are three types of users:

1. IT technician for the testing environment.
2. An end user in the small business IT infrastructure
3. An IT administrator for the small business IT infrastructure.
The IT technician has at least three years of experience with server hardware, operating systems, and VMware products including setting up and modifying new VMs. They have some experience, as well, with alternatives to VMware products like Hyper-V, Virtual PC, or Xenserver. The frequency of use will depend largely on the customer’s needs but the technicians access the test environment several times a week for several hours at a time.

As you will see in Figure 1 they connect to the environment by VPN and all other interactions will be through vSphere including configuring the ESXi server as well as setting up the virtual machines.

![Figure 1 – Dedicated Test Environment IT Technician Use Case](image)

The IT administrator is responsible for setting up and maintaining the environment, and the end user, whose interaction with the environment will be exclusively through their desktop.
The IT administrator at a small business should have the flexibility to manage and configure each level of the infrastructure as they will most likely be the only administrator or one of a few administrators.

The end user has experience with GUI based computer operating systems such as Windows, business productivity software (word processing, spreadsheets etc.), email, and accessing the internet.

Shown in Figure 2 the two users in the open source infrastructure have several of the same interactions with the environment, with the Admin ultimately having total control. The desktop user’s interaction with the infrastructure will be from their PC as they use their PC to access files, applications, or the internet. The administrator will be heavily involved in administering the network, servers, and desktops.
2. Design Protocols

2.1 Dedicated Test lab

The environment is on a dedicated network segment and is accessed via a point-to-point tunneling protocol (PPTP) VPN making it accessible from the office or home.
I chose a PPTP VPN due to its simplicity of access and setup. (SSL-VPN was also an option but it required a separate software installation and far more complicated setup on the firewall. The need for extra software would also limit our ability to access the environment to computer systems that allowed us to install the necessary SSL-VPN client software.)

The network setup is relatively straightforward, at the top of the diagram you’ll see the remote PC that accesses via VPN tunnel through the internet to one of two firewalls. After the proper authentication to the subnet, the user starts up the vSphere client to access the hypervisor.
Since Peak 10 already uses VMware products extensively, the server is loaded with the virtual machine hypervisor ESXi v4.1. ESXi allows us to simulate different types of hardware and load a virtual machine with just about any kind of computer operating system; configure it according to the customer’s needs; and do so very quickly. (12) It also has a very nice GUI management interface called vSphere.

From personal experience, I can attest that one of the great benefits of a virtual machine hypervisor is that you can take a single physical server (like we have) and run multiple virtual servers on it, each containing whatever operating system or software you feel like installing.

2.2 Open Source Infrastructure

With a small business on a tight budget in mind and with the goal of utilizing all free open source products, this means that the firewall, router, and servers all have open source software loaded. To simulate the kind of inexpensive equipment a small business might utilize I have created virtual machines to reflect older hardware. This means that I am simulating:

- Single core processors
- Memory of 1GB or less.
- 40GB of storage per virtual machine. (The fileserver will also have a second 100GB hard disk mounted at the /home directory for the network storage.)
- All the operating systems will be the 32bit versions.
Figure 4.1 best illustrates what I mean when I say a small business IT infrastructure:

![Small Business Open Source Infrastructure Network Diagram](image)

You will note that on Figure 4.1 I’ve also included desktops as part of the environment. Here I run into a problem. Peak 10 doesn’t set up or support desktops for customers. However, I feel that in order to effectively test an end-to-end solution I need to include desktops into the equation to show how the rest of the infrastructure interfaces with the desktop and how a desktop user is able to use the infrastructure.

You will also note that I have separated the Router and the Firewall in this diagram when in actuality both functions will be performed by the same piece of software on the same virtual server.
Luckily, we are in a golden age of open source options. In my research I found a number of potential solutions for each stage of the IT infrastructure. These are the ones I have used:

- **FIREWALL AND ROUTER** – After considering several options I decided to go with **Untangle**. Untangle is a Debian Linux variant and has a very nice GUI in addition to the command line interface (CLI) which may appeal more to the small business administrator with a lot on his/her plate.

- **FILE SERVER - CentOS 6.2.** (15) CentOS is well regarded, stable, secure, and robust. It doesn’t hurt that it is essentially Red Hat Linux in all but name. (15) (Red Hat is a very good and very mature Linux distribution that comes packaged with support contracts and their own suite of software administrator tools which they, of course, charge for.) (15) The protocol used to share files across the network is NFSv4 (which is also backward compatible to earlier versions.) I considered SAMBA for it greater security controls, integration with LDAP, and compatibility with the Windows environment but the setup was more complicated than the scope of this project and consequently I opted for the easier to configure NFS protocol for this proof of concept. I will implement SAMBA at a later date.

- **WEB SERVER - CentOS 6.2.** The web server is set up in a LAMP configuration. LAMP stands for Linux, Apache, MySQL, and PHP. These are the basic services you need installed for any modern day webserver. Apache is the actual web server, MySQL functions as the back- end database and PHP is a very common programming language for the web. (22)
• **DESKTOP** – I’ve considered several options and initially planned on four options each with their strengths and weaknesses. Due to considerations such as the overall user experience and the amount of time left in the project I have chosen **Chromium OS**.

Chromium is a version of Google’s Chrome OS compiled and built in Open SuSE. It has all the advantages of SuSE (plenty of admin tools, very recognizable Windows-like interface) but is overlaid with a version of Chrome OS. (21) Anyone familiar with an internet browser should have no problem working in Chromium. (Each Chromium desktop will also be installed with the NFS protocols to allow them to access the fileserver.)

To help visualize the environment with the distros, here’s the figure from earlier with the distros chosen in the place of the computer icons.

---

**Small Business Open Source Infrastructure**

- **Internet**
- **Firewall**
- **Router**
- **Switch**
- **CentOS Fileserver**
- **LAMP Server [Web server]**
- **User Desktop**
- **Admin Desktop**

*Figure 4.2 – Small Business Infrastructure with distros.*
3. Deliverables

This project has delivered a fully functional private cloud for the testing and setup of potential customer solutions. Within the cloud there is an IT infrastructure made up of entirely open source software. This is a proof of concept showing the feasibility of a low cost open source environment for a small business.

More specifically the project contains:

- Virtual Machine based dedicated testing environment (essentially a private cloud.)
  
  a. It is accessible via PPTP VPN through two Fortigate 60B firewalls that also act as switches.
  
  b. The testing environment resides on an HP DL 360 G4 server with dual Xeon processors, (each @3.0GHz), 3GB memory, 1.36TB storage, and 2 network connections.
  
  c. The VM software is VMware ESXi 4.1.
  
  d. The environment allows us to create and setup virtual machines of almost any kind of OS. It also allows us to create virtual networks between the different virtual machines.

- A complete IT infrastructure (for a small business) based on open source and free of cost software.
  
  a. There are five virtual machines that make up the following segments of the infrastructure:

     1. A router/firewall
     2. File server
     3. Application Server
4. User Desktop

5. Administrator Desktop

b. The router/firewall control and direct traffic on the internal network. It filters traffic that enters and leaves the network. It also logs traffic and allows the administrator to access those logs. The software chosen is called Untangle and is based on Debian Linux.

c. The file server hosts an NFS fileshare and allows the users to connect to store files on the network. It is based on CentOS 6.2 Linux.

1. Users can access a group fileshare on the fileserver from their individual desktops.

d. The application (web) server is in a LAMP configuration (Linux, Apache, MySQL, and PHP) which means that each of these protocols will be installed. The Linux distribution used is also be CentOS 6.2.

e. The two desktops are each be loaded with Chromium OS Linux and have internet access and access to the servers.

f. The IT infrastructure functions as any should in that the users are given access to the resources they need (files, applications, internet) and limited from the ones they don’t need.

g. There are domain based names for each device either via the DNS server on the firewall/router so that each may be accessed via name resolution and not just IP address.

h. Servers and Firewall are accessible via SSH from the desktops.
4. Project Plan

4.1 Hardware

Hardware needs are relatively simple for the entire project:

- 1 server to host the hypervisor and all the virtual machines.
- 2 multi-function network devices that provide firewall, router, and switch all in one device.
- A NAS device with dual 1TB SATA drives to provide additional storage.

For the server I have used a single HP DL 360 G4.

Server Specifications:

- Dual processors, each running at 3.0GHz.
- 8 GB memory @ 800MHz.
- 36 GB storage. The storage will be split between a pair of 36 GB SCSI drives in a RAID 1 array providing the system OS drive. 2 NIC ports.

The multi-function network devices used are a pair of Fortigate 60B firewalls.

Specifications:

- Fully configurable as firewalls, routers, and 6 port switches.
- Two external internet connections and a dedicated DMZ port.
- Able to be paired with an identical Fortigate device in a HA pair for failover redundancy.

The dual firewalls each have their own separate network connection. This allows us, in the future, to test failover capabilities of ESXi. The 12 ports total between the two Fortigates allow us to connect more dedicated servers as needed or temporarily connect new customer equipment for configuration.
The NAS used is a D-Link DNS-320

Specifications:

- Dual 1TB 7200rpm Western Digital drives paired in RAID 1 for a total of 1TB of storage.
- Connectable to the physical network and IP addressable.
- Configured with the protocol NFS (network file system) to allow ESXi access.

The NAS is necessary as the server only supports old SCSI drives and is currently limited to 36GB of total storage space. This is obviously inadequate for our need to create virtual machines, each with their own storage space needs that go well beyond the limits of 36GB.

4.2 Costs

The total cost of the entire project has been $119.00 which was covered out of my company’s budget. The cost went towards the NAS device. Otherwise, everything else was either donated or free of charge.
4.3 Gantt Chart

Below in figures 5.1 – 5.3 is shown the Gantt chart detailing the specific milestones completed in the project.
<table>
<thead>
<tr>
<th>Task Description</th>
<th>Start Date</th>
<th>End Date</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow SSH access to webserver</td>
<td>4/21</td>
<td>4/23</td>
<td>100%</td>
</tr>
<tr>
<td>Test Scenarios</td>
<td>4/26</td>
<td>4/28</td>
<td>100%</td>
</tr>
<tr>
<td>Perform and record results of testing scenarios</td>
<td>4/26</td>
<td>4/28</td>
<td>100%</td>
</tr>
<tr>
<td>Evaluate results</td>
<td>4/26</td>
<td>4/28</td>
<td>100%</td>
</tr>
<tr>
<td>Fix any problems</td>
<td>4/27</td>
<td>4/29</td>
<td>100%</td>
</tr>
<tr>
<td>Turn in testing data</td>
<td>4/23</td>
<td>4/25</td>
<td>100%</td>
</tr>
<tr>
<td>Final Report</td>
<td>5/15</td>
<td>5/20</td>
<td></td>
</tr>
<tr>
<td>Tech Expo</td>
<td>5/18</td>
<td>5/20</td>
<td></td>
</tr>
<tr>
<td>Prepare for Tech Expo</td>
<td>5/18</td>
<td>5/20</td>
<td>100%</td>
</tr>
<tr>
<td>Setup for Tech Expo</td>
<td>5/14</td>
<td>5/16</td>
<td>100%</td>
</tr>
<tr>
<td>Tech Expo</td>
<td>5/15</td>
<td>5/16</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 5.2 - Page two of Gantt chart.
Figure 5.3 - Page one of Gantt chart.
5. Proof of Design

Now that I’ve outlined the problem and proposed solution it’s time to provide the proof of concept.

5.1 Dedicated Testing Environment

Here’s the initial login screen through the vSphere client which allows access to the ESXi server. Until I establish a domain I use the NATted IP address after connecting through the PPTP VPN.

Figure 6.1 - vSphere Client Login
Once logged in we’re presented with this screen. From here we can administer the entire environment via the tabs. From here we can also create a new VM by simply clicking on the icon in the corner.
Now that we’re logged in we can create a new VM.

After clicking on the “New Virtual Machine” icon we’re presented with a series of pop-up windows.

Figure 6.3 - New Virtual Machine Window

Here you can take the “Typical” setup or the “Custom” setup. The custom is, of course, more complex. We’ll select the typical setup.
After a series of more questions regarding where and how to store the VM files, network adapter, which OS will be loaded, we’re brought back to the main screen but with our new VM now visible.

![Figure 6.4 - New Virtual Machine Finished](image)

From there we power it on by clicking on one of the green arrows.
5.2 Open Source Small Business Infrastructure

Here in figures 7.1 – 7.4 are the desktops for each of the elements in the environment.

Figure 7.1 - Firewall/Router - Untangle

Figure 7.2 - Fileserver - CentOS 6.2

Figure 7.3 - Web Server - CentOS 6.2

Figure 7.4 - Desktop - Chromium
Here at the firewall it can be administered from the GUI client or the terminal. (It’s worth noting that any changes made from the terminal are not supported if you do end up paying for a support contract with the vendor.)

If we click on the “Launch Client” button a login screen is presented. After authenticating you’re presented with the virtual rack shown here.
They provide, for free, a number of items such as spyware protection, IPS, VPN, and firewall. Clicking on the firewall we’re taken to this simple screen where you can add delete and change firewall rules.

On the left bar if we select the “Config” tab and the “Networking” button we can then configure, the interfaces, the DHCP Server, the DNS Server, IP addresses, Hostnames, etc.

Moving over to the fileserver:
Here on the fileserver the network file system service can be restarted with a simple command.

And the fileserver stats can be shown with an “nfsstat” command.
desktop we can check to make sure the fileserver works.

In Figure 11.3, with a “df” command, we can see the fileserver mounted to a local directory (/mnt/nfs) on the desktop. With a “touch” command we create a blank file in the local directory mounted on the fileserver.
Back to the fileserver, by listing the shared directory (/home/share) we can see in Figure 11.4 the file just created from the desktop is residing on the fileserver.
Now to test the webserver, we can go back to the desktop and attempt to connect to it as seen in figures 12.1 and 12.2 below.

By simply connecting to the IP of the webserver we can see that apache is successfully installed and displaying and info page.

I also created a “phpinfo()” page test.php:

Figure 10.1 - Apache Installed on Webserver

Figure 10.2 - phpinfo() Page: test.php
6. Testing

6.1 Testing Scenarios

The testing scenarios to help establish the proof of concept are listed in figures 11.1 – 11.3 below. All testing was done by myself alone.

<table>
<thead>
<tr>
<th>Step #</th>
<th>Action</th>
<th>Data / Input</th>
<th>Expected Results</th>
<th>Pass/Fail</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preconditions</strong></td>
<td>1. Logon via PPTP VPN</td>
<td>74.126.83.148</td>
<td></td>
<td>Pass</td>
<td>Login took just a few seconds as expected.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>2. Enter username and password</td>
<td>admin/password</td>
<td>VPN will connect and network icon in system tray will show being connected to VPN</td>
<td>n/a</td>
<td>Username and password were already saved and submitted automatically.</td>
</tr>
</tbody>
</table>

**Scenario 1 - Test Environment Admin connects to vSphere**

1. Open vSphere client and logon with username and password root/[password]
The vSphere client will access the ESXi box and the interface GUI will appear.

2.01 Click on "squares" icon on upper left part of window.
One click.
New Virtual Machine dialogue box will appear.

2.03 Select "typical" VM setup and click right arrow.
One radial selected and one click
The VM wizard will take you to the next step of a "typical" setup.

2.03 Name virtual machine then click 'Next'.
Name of virtual machine and one click
It will accept the name and take you to the next step of the setup.

2.04 Select "Guest Operating System" type radial.
Click on radial
The guest OS will remain selected.

2.05 Select "Version:" from drop-down menu. Then click 'Next'.
Click on dropdown box and highlight OS. One click
The chosen OS will be accepted by ESXi.

2.06 Select amount of memory by typing in amount in field. Click 'Next'.
Amount of memory desired in MBs. One click.
That the amount chosen won't exceed the maximum available.

Figure 11.1 – Page one of test scenarios
<table>
<thead>
<tr>
<th>Scenario 3 - Test Environment Assigns OS to VM Shell</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.01 Right click on new VM from left panel and select &quot;Edit Settings&quot;.</td>
</tr>
<tr>
<td>3.02 Selects &quot;CD/DVD drive&quot; from hardware window.</td>
</tr>
<tr>
<td>3.03 Select &quot;Connect at power on&quot; checkbox</td>
</tr>
<tr>
<td>3.04 Select &quot;Datastore ISO file&quot; radial</td>
</tr>
<tr>
<td>3.05 Click on 'Browse' button.</td>
</tr>
<tr>
<td>3.06 Select .iso file and click 'Ok' button.</td>
</tr>
<tr>
<td>3.07 Right click on new VM from left panel and select &quot;Power On&quot;.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 4 - Open Source Infrastructure User Access PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.01 Enter logon and password</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 5 - Open Source Infrastructure User Accesses Internet to access company gmail account</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.01 Single click on Gmail icon on Chrome desktop.</td>
</tr>
<tr>
<td>Test Case</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>5.02</td>
</tr>
<tr>
<td><strong>Scenario 6 - Open Source Infrastructure Admin Checks Firewall Rules</strong></td>
</tr>
<tr>
<td>6.01</td>
</tr>
<tr>
<td>6.02</td>
</tr>
<tr>
<td>6.03</td>
</tr>
<tr>
<td>6.04</td>
</tr>
<tr>
<td>6.05</td>
</tr>
<tr>
<td><strong>Scenario 7 - Open Source Infrastructure Admin brings down Apache service running on LAMP server.</strong></td>
</tr>
<tr>
<td>7.01</td>
</tr>
<tr>
<td>7.02</td>
</tr>
<tr>
<td>7.04</td>
</tr>
<tr>
<td>7.05</td>
</tr>
</tbody>
</table>

---

**Scenario 1** The test environment administrator logging into the ESX server via vSphere. Basic access to the environment is critical and will show that the VPN, hypervisor, and management software are all set up correctly to allow access.

*This was successful. I was able to easily connect to the environment and login to vSphere.*
Scenario 2 - The test environment administrator creating a virtual machine shell. This will showcase the basic element of a cloud environment, a virtual machine, can be easily setup within the new environment.

--This was successful. Creating a VM shell is easy and intuitive with their wizard.

Scenario 3 - The test environment administrator loading the virtual machine shell with an operating system and turning on the virtual machine. The next logical step will be for the administrator to load and install an operating system.

--This was successful. It’s important to remember to think of the VM shell as physical hardware. To load the OS all that’s needed is to mount an ISO image to the virtual optical drive and set it to connect at startup.

Scenario 4 - The open source small business infrastructure user logging into their PC and accessing their Gmail account. This will prove basic internet connectivity and that the firewall is passing internet traffic.

--This was successful. Internet connectivity was immediate from the user desktop and the Chrome OS made the connection as simple as clicking on an icon.

Scenario 5 - The small business infrastructure user placing a file into their NFS fileshare folder. This will show that network traffic inside the subnet is working and that the file server is available to the desktop users.

--This was successful. After mounting the fileshare to a directory on the user desktop I created a file in it. I then went to the fileserver and accessed the directory being shared and found the new file I just created.
Scenario 6 - The open source small business infrastructure administrator checking his firewall rules. This will show that he has access to the firewall and can check and change access rules to the network as needed.

--This was successful. Untangle makes firewall administration very easy and user friendly. Three clicks and user authentication were all that was needed to access the firewall rules.

Scenario 7 - The open source small business infrastructure administrator bouncing the apache service on their webserver. The apache service is the critical component to the webserver and being able to restart it is a simple but important task for the admin.

--This was successful. CentOS makes server administration very easy with the “service” command. All that was needed to restart the web services was to ssh to the server, elevate to super user status, and restart the apache daemon (httpd) with the command “service httpd restart”.

7. Conclusion

Peak 10 Cincinnati has reached a level of maturity in the market that a dedicated testing environment makes sense to allow us to fully test and setup potential IT solutions for customers. Harnessing the power of virtualization has allowed us to do so in a safe and controlled environment and consequently more efficiently address their needs.

As part of our maturity it also makes sense for us to better serve our customers by exploring more free and open source software options. Open source software presents many viable options to replace all or part of an IT infrastructure and exploring some of those options gives us a larger tool bag to draw from when meeting a customer’s needs. Utilizing the
dedicated testing environment I have set up a proof of concept completely open source IT infrastructure in order to explore budget-conscious options for our customers.
References


    http://www.techradar.com/news/software/operating-systems/10-best-linux-distros-for-2011-

17. OSXDaily. “Run Android using a virtual machine on a Mac or Windows PC”.
   http://osxdaily.com/2010/12/14/run-android-using-a-virtual-machine-on-a-mac-or-windows-pc/.
   December 14, 2010.

18. Kurlyandchik, Mark. “Android Overtakes Symbian as World's No. 1 Smartphone Platform”. DAILY
    TECH.


    November 30, 2011.
