Northeastern University – Silicon Valley

CS 6620 Cloud Computing

Homework Set #2 [100 points]

INSTRUCTIONS: ***Please provide clear explanations in your own sentences, directly answering the question, demonstrating your understanding of the question and its solution, in depth, with sufficient detail. Submit your solutions [PDF preferred]. Include your full name. Do not email the solutions.***

PART I: Concepts and Theory, Algorithms [60 points]

Answer the below questions in your own words (1 – 2 small paras each; Diagrams as needed).

Please provide accurate, concise (maximum 10 sentence) answers. **All questions carry 4 points.**

1. Explain, in your own words, what Systems Virtualization is, and 4 advantages of it giving a scenario example for each. (use specific, concrete scenario to illustrate the benefit.)

2. Why is virtualizing the X86 processor necessary for systems Virtualization?

3. What are user mode instructions and kernel mode (privileged) instructions? Give 2 examples for each.

4. What are the challenges involved in virtualizing the X86 processor? How does Virtualization address them?

5. How is virtual memory implemented in a real machine? Explain how it would be implemented in a VM? Why is it very costly?

For the below questions, first consider a public cloud (IaaS) platform like AWS, where multple tenants can rent infrastructure and build their own IT environments. Then, explain what the below technologies (aka ~ features) are and why you need them to build the above public cloud (what functions does the feature serve?) Please provide 2 use cases in each case which need the particular feature. Provide detailed answers (2- 3 para What, How, Why) with a diagram in every case. **All questions carry 8 points**.

6. VPC
7. VPN
8. VLAN
9. DHCP
10. NAT

PART II: LAB (You can use AWS Cloud or any other available tools) [40 points]

A. Create a Docker Container following this tutorial https://docker-curriculum.com/

B. What are the differences between a VM and a Container?

C. Your startup Stitchfix.com uses a 3 tier App architecture. You need to scale the platform up (lot of business demand from users.) In this context, what do the below functions mean? (please explain)
   i. automating deployment,
   ii. scaling,
   iii. management of containerized applications

D. What is Kubernetes? Why dod you need it to run the above startup platform?

ANS (Qichen An)

Part 1: Concepts and understanding

1. Explain, in your own words, what Systems Virtualization is, and 4 advantages of it giving a scenario example for each. (use specific, concrete scenario to illustrate the benefit.)

Systems virtualization is a widely used technology for customizable computing environments. It's one of the principle part of cloud computing. Virtualization allows the creation of a secure, customizable,

and isolated execution environment for running applications with minimum costs. Even in the condition that the applications are untrusted, VM will not affect other users' applications.

Four advantages of virtualization:

a. Managed execution and isolation. It offers more reliability and security compared to non-virtualized environment.

Example: In the Virtualized scenario of the environment dying, virtualization makes the disaster recovery process very simple. The virtual machines, taking regular snapshots and providing up-to-date data can be easily backed up and recovered.

b. Portability. Since the virtualized environment is segmented into virtual machines, your developers can quickly spin up a virtual machine without impacting a production environment.

Example: a new software patch has been released, someone can clone the virtual machine and apply the latest software update, test the environment, and then pull it into their production application. This increases the speed and agility of an application.

c. Cost. Visualization could help reduce the costs of maintenance. Using a virtualization system is actually cheaper due to the fact that it doesn't require any hardware components.

Example: In the scenario, choosing to establish a visual environment reduces number of hardware, and eventually reduces the costs of purchasing and maintaining them. It would be cheaper than physical servers.

d. Efficiency. Visualization could be helpful for efficient use of resources. Virtualization allows automatic updates to the hardware and software by installing on their third-party provider and virtualization reduces the load of resource management for facilitating the efficiencies in the virtual environment.

Example: In the scenario that one programmer makes a major mistake, the programmer can access the previous snapshot and resume the work.

2. Why is virtualizing the X86 processor necessary for systems Virtualization?

The central processor is the core of a virtual machine. Virtual PC for Windows is able to make direct use of the host's x86 processor. With x86 computer virtualization, a virtualization layer is added between the hardware and operating system.

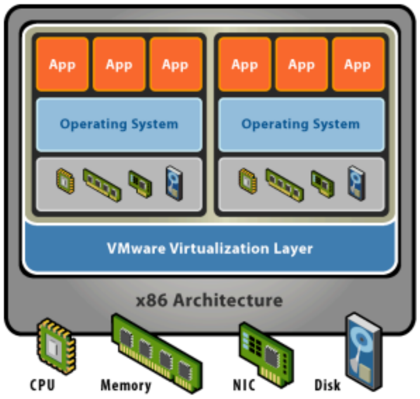


Figure 2 – x86 virtualization layer

By combination of binary translation and direct execution on the processor that allowed multiple guest OSes to run in full isolation, the market for x86 virtualization was created. This could be helpful to save costs for companies. The savings is further driving the rapid adoption of virtualized computing from the desktop to the data center. It turns out that the X86 processor is crucial for systems virtualization.

3. What are user mode instructions and kernel mode (privileged) instructions? Give 2 examples for each.

The Instructions that can run only in User Mode are called user mode instructions or Non-Privileged Instructions. Similarly, the Instructions that can run only in Kernel Mode are called kernel mode Instructions or Privileged Instructions. Dual Mode Operation could ensure the protection and security of the System from unauthorized or errant users.

Examples:

User mode instructions:

Reading the status of Processor;

Reading the System Time

Kernel mode instructions:

I/O instructions and Halt instructions;

Turn off all Interrupts;

Clear the Memory or Remove a process from the Memory

4. What are the challenges involved in virtualizing the X86 processor? How does Virtualization address them?

The challenge involved in virtualizing the X86 processor is how to trap and translate sensitive and privileged instruction. X86 operating systems are designed to run upon the computer hardware. As shown the figure, the x86 architecture offers four levels of privilege known as Ring 0, 1, 2 and 3 to operating systems and applications. user level applications typically run in Ring 3, but the operating system (direct access to the memory and hardware) executes its privileged instructions in Ring 0. Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (Ring 0). However, some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0.

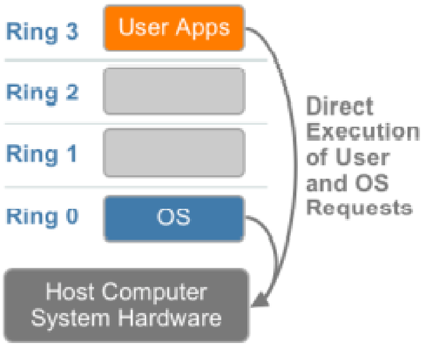

Ring 3
User Apps
Ring 2
Ring 1
Ring 0
OS
Host Computer System Hardware
Direct Execution of User and OS Requests

Figure 4 – x86 privilege level architecture without virtualization

This challenge was resolved in 1998 by VMware. VMware developed binary translation techniques that allow the VMM to run in Ring 0 for isolation and performance. Now, there are three alternative techniques: Full virtualization using binary translation; OS assisted virtualization or paravirtualization; Hardware assisted virtualization.

5. How is virtual memory implemented in a real machine? Explain how it would be implemented in a VM? Why is it very costly?

In a real machine, virtual memory is implemented by MMU (memory management unit) performing address translation for a “logical” address to a “physical” address conversion. In general, virtual memory is either paged or segmented.

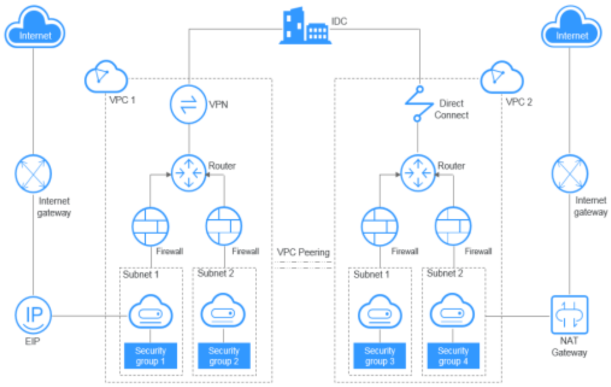
In a VM, Virtual PC makes use of the host processor’s MMU to perform address translation.

To provide maximum performance, using virtual memory could be costly. RAM cells have a limited number of writes, so using them for virtual memory often reduces the lifespan of the drive. Also, data must be mapped between virtual and physical memory, which requires extra hardware support for address translations, slowing down a computer further.

6. VPC

The Virtual Private Cloud (VPC) service enables to provision logically isolated, Configurable, and manageable virtual networks for Elastic Cloud Servers (ECSs). As follows is the figure showing VPC components.

Figure 1-1 VPC components



By the feature serving, functions contain improving cloud resource security and simplifying network deployment.

Use cases:

a. Hosting web applications
Users can host web applications and websites in a VPC and use the VPC as a regular network.

b. Hosting services that demand high security
Users can create a VPC and security groups to host multi-tier web applications in different security zones.

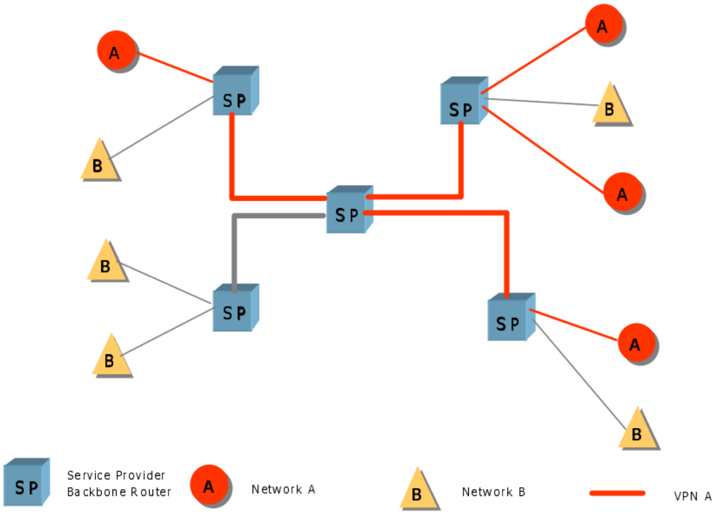
7. VPN

According to the documents from lecture 2, a definition of Virtual Private Networks (VPN) perhaps is the least difficult term to define and understand.

The common and somewhat formal characterization of the VPN, and perhaps the most straightforward and strict definition, is: a VPN is a communications environment in which access is controlled to permit peer connections only within a defined community of interest, and is constructed though some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis.

A simpler, more approximate, and much less formal description is: a VPN is private network constructed within a public network infrastructure, such as the global Internet.

As follows is a simple example.



By the feature serving, VPN make some portion of the communications essentially "invisible" to external observers, while taking advantage of the efficiencies of a common communications infrastructure.

Use cases:

a. an online banking service

one-to-one VPN's (single dial-up users establishing a VPN connection) could be helpful for a secure application

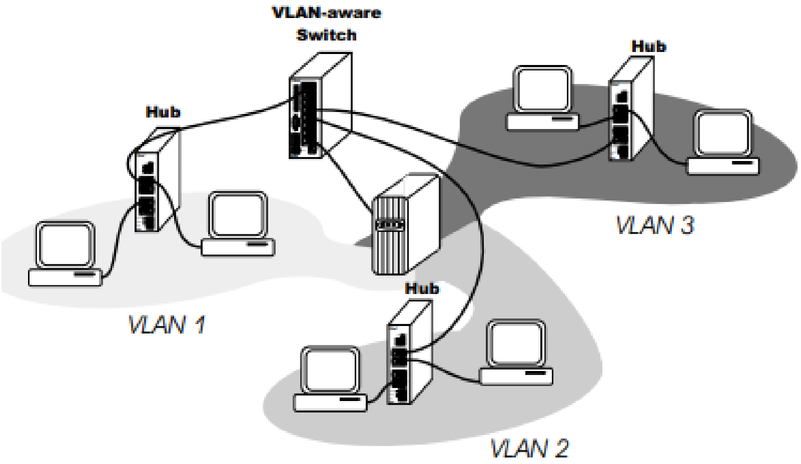
b. encrypted e-mail transactions

VPN's can be implemented at the transport and application layers of the protocol stack

8. VLAN

Before the definition of VLANs, the definition of LAN should be understood. A local area network (LAN) is a private network usually confined to one plant. Thus, Virtual LANs (VLANs) allow a single physical LAN to be partitioned into several smaller logical LANs.

As follows is a VLAN application.



VLANs are effective because a large LAN is portioned into manageable subsets. VLANs restrict the broadcast domain, improve performance and security, and so on.

Use cases:

a. port VLAN

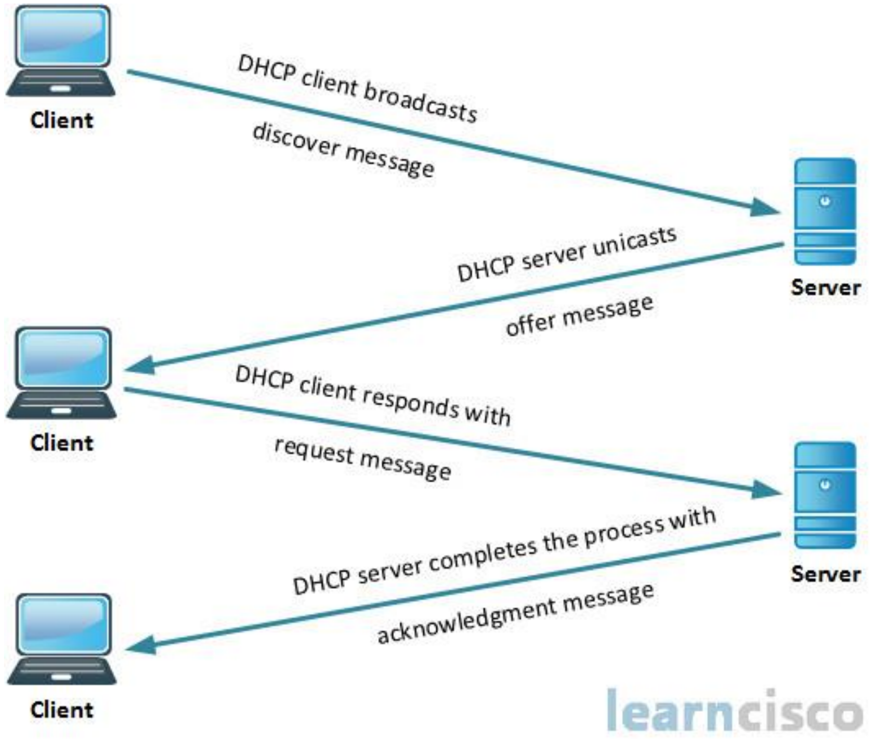
The above figure of VLAN application is port VLAN. Switches create an association of MAC address and port numbers, and then VLAN associations are added.

b. VLAN core switches

Core switches understand VLAN tags and reside in the backbone of the LAN and are usually only connected to edge switches.

9. DHCP

The Dynamic Host Configuration Protocol (DHCP) is a network management protocol. It automatically assigns IP addresses and other communication parameters. The DHCP employs a connectionless service model, using the User Datagram Protocol (UDP). DHCP is an important part of the DDI solution (DNS-DHCP-IPAM).



The technology eliminates the need for individually configuring network devices manually.

Use cases:

There are four key DHCP usage scenarios: 1. Initial Client Connection: the client requests from the DHCP server an IP address and other parameter values for accessing network services 2. IP Usage Extension: the client contacts the DHCP server to extend usage of its current IP address 3. Client Connection After Reboot: the client contacts the DHCP server for confirmation that it can use the same IP address being used before reboot 4. Client Disconnection: the client requests the DHCP server to release its IP address.

10. NAT

Network address translation (NAT) is a method of mapping an IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device.

It has become a popular and essential tool in conserving global address space in the face of IPv4 address exhaustion.

Use cases:

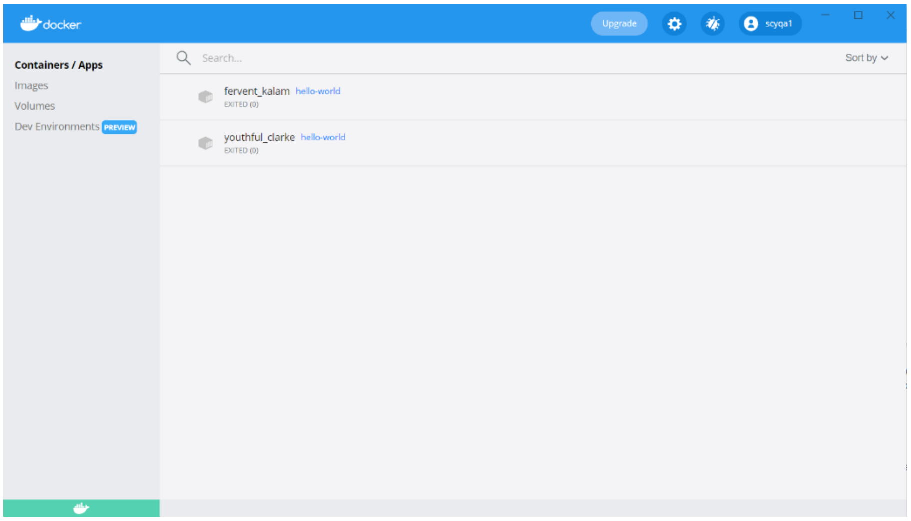
a. Enable a Server on the Inside Network to Reach the Internet Using a Public IP address

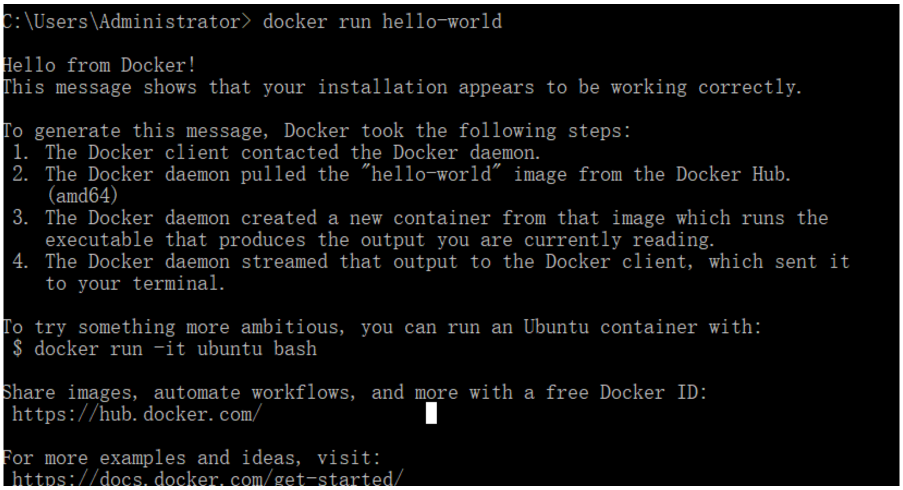
b. Enable Users on the Inside Network to Access the Internet Using the Outside Interface's Public IP Address

Part 2: LAB

A. Create a Docker Container

After creating an account (scyqa1) and download the Docker Desktop, I create a Docker Container (Hello World).



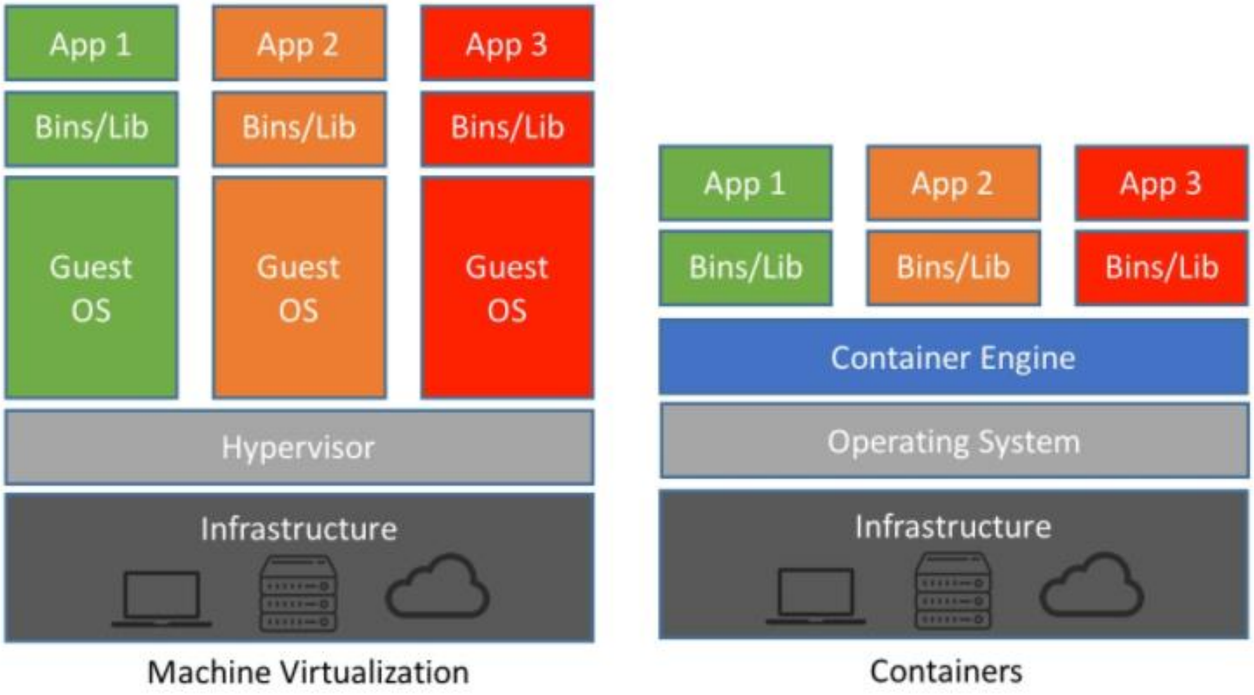


```
C:\Users\Administrator> docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

B. The differences between a VM and a Container?

Within each virtual machine runs a unique guest operating system. Each VM has its own binaries, libraries, and applications that its services, and the VM may be many gigabytes in size.

While containers sit on top of a physical server and its host OS—for example, Linux or Windows. Each container shares the host OS kernel and, usually, the binaries and libraries, too. Shared components are read-only. So, they are only megabytes in size and take just seconds to start, versus gigabytes and minutes for a VM.



Containers also reduce management overhead. Because they share a common operating system, only a single operating system needs care and feeding for bug fixes, patches, and so on.

In summary, containers provide a way to virtualize an OS so that multiple workloads can run on a single OS instance. With VMs, the hardware is being virtualized to run multiple OS instances.

C. Scenario: startup of Stitchfix.com

i. automating deployment

Deployment automation refers to a software deployment approach that allows organizations to increase their velocity by automating build processes, testing, and deployment workflows for developers. In other words, it allows organizations to release new features faster and more frequently.

In the scenario, by Continuous testing, Configuration management for environments and Monitoring and logging, lead time could be reduced by saving time on environment setup and manual testing for the whole project. Also, for developers of the website, they can catch bugs faster, find more complex bugs, and reduce the code feedback loop.

ii. scaling

Scaling is a common concept in life. For the scenario, to scale up the project of 'Stitchfix.com', I think there are two directions could be favorable. For the frontend, launching new sub-website with different functionality. For the backend, the responding is to add servers and update backend systems that contain confidential or personally identifiable data.

iii. management of containerized applications

Containerization is an approach to software development that isolates processes that share an OS kernel. In the context, IT teams of the web could use containers for cloud-native, distributed applications, and to package legacy applications for increased portability and efficient deployment. As benefit of container management, management for clusters of container hosts is simplified.

D. What is Kubernetes? Why do you need it to run the above startup platform?

In 2015, Google introduced the container orchestration platform Kubernetes, which was based on its internal data center management software called Borg. At its most basic level, open source Kubernetes automates the process of running, scheduling, scaling and managing a group of Linux containers. With more stable releases throughout 2017 and 2018, Kubernetes rapidly attracted industry adoption, and today it is one of advanced container management technology.

As the above introduction of Kubernetes shows, Kubernetes is a kind of container management technology. Thus, it could be helpful for the third point (management of containerized applications) of startup of the web Stitchfix.com. It could be easier to deploy and manage applications on IBM Cloud Kubernetes Service.