CS-524 Homework # 2

2. (10 points) Explain the advantage that paravirtualization provides for handling timers in virtual machines.

Solution:

Para-virtualization is a virtualization technique that presents a software interface to virtual machines that is similar but not identical to that of the underlying hardware.

- Just as the user expects its machine to maintain real time, similarly, in a virtual system, the user expects that the virtual machine maintains the real time, not virtual time.
- As most operating systems rely on timer interrupts to maintain their time, the system expects timer interrupts even when idle.
- If timer interrupts are missed, it can affect the time keeping of the operating system.
- Without para-virtualization, the VMM needs to continue injecting timer interrupts or to inject back-to-back timer interrupts when the guest operating system is scheduled back to run. This is not a reliable or scalable way of virtualization.
- With para-virtualization, a typical modification is to change the idle code to request the VMM to notify itself in a specified time period. Then time is re-calculated and restored in the guest.

Source: [Hybrid-Virtualization—Enhanced Virtualization for Linux by Jun Nakajima and Asit K. Mallick (Intel Open Source Technology Center)]

3. (10 points) Explain how paravirtualization helps in minimizing access to APIC.

Solution:

- In x86 or x86- 64, local Advanced Programmable Interrupt Controller (APIC) is required to support Symmetric Multi-processing (SMP) especially because the operating systems needs to send IPI (InterProcessor Interrupt).
- Each access to the APIC registers needs to be intercepted for virtualization, causing overhead (often a transition to the VMM).
- This is expensive because of the transitions into and out of the hypervisor.
- Para-virtualization (which has full-view of the code) can replace such multiple implicit requests with a single explicit hypercall, achieving faster, simpler, and more efficient implementations.

Source: [Hybrid-Virtualization—Enhanced Virtualization for Linux by Jun Nakajima and Asit K. Mallick (Intel Open Source Technology Center)]

4. (5 points) Find out if Linux (like Unix) has both the user-mode and system-mode stacks for each process it runs.

Solution:

Yes, Linux (like Unix) has both the user-mode and system-mode stacks for each process it runs.

- CPUs may have more than one set of identical registers. As a minimum, one register set is reserved for the user mode—in which application programs execute– and the other for the system (or supervisory, or kernel) mode, in which only the operating system software executes.
- An operating system can now maintain **two** stacks—one for executing the system code and one for executing the user code.
- If and when the CPU gets interrupts, if it is in the User mode, the execution is stopped and the system or kernel mode handles the interrupt.

Source: [Cloud Computing: Business Trends and Technologies]

5. (10 points) Find out what "unscrambled" means in the description of the Intel LSL instruction (you can, for example, use the Intel manual referenced in the lecture).

Solution:

LSL (Load Segment Limit) has Opcode: 0F 03 /r.

- It loads the unscrambled segment limit from the segment descriptor specified with the second operand (source operand) into the first operand (destination operand) and sets the ZF flag in the EFLAGS register.
- The "unscrambled" refers to the granularity flag, G.
- The segment limit is a 20-bit value contained in bytes 0 and 1 and in the first 4 bits of byte 6 of the segment descriptor.
- If the descriptor has a byte granular segment limit (the granularity flag is set to 0), the destination operand is loaded with a byte granular value (byte limit).
- If the descriptor has a page granular segment limit (the granularity flag is set to 1), the LSL instruction will translate the page granular limit (page limit) into a byte limit before loading it into the destination operand.
- The translation is performed by shifting the 20-bit "raw" limit left 12 bits and filling the low-order 12 bits with 1s.
- Operation:
  temp ← SegmentLimit([SRC]);
  IF (G ← 1)
  THEN temp ← ShiftLeft(12, temp)
  OR 00000FFFH.

Source: [https://software.intel.com/sites/default/files/managed/ad/01/253666-sdm-vol-2a.pdf]

6. (25 points) Read the following two papers:

• Carl Waldspurger and Rosenblum, M. (2012) I/O Virtualization. Communications of the ACM, vol. 55, No 1. January 2012. Pages 66-72; and

• Muli Ben-Yehuda; Xenidis, J.; Ostrowski, M.; Rister, K.; Bruemmer, A.; Van Doorn, L. (2007). The Price of Safety: Evaluating IOMMU Performance. Proceedings of the Linux Symposium on June 27th–30th, 2007. Ottawa, Ontario. Pages 225-230.

6.1) Explain the advantages and disadvantages of using I/O MMU by citing the appropriate text from the paper;

Solution:

In computer architecture, an IOMMU (I/O memory-management unit) translates I/O-virtual memory addresses to corresponding physical memory addresses.

Advantages of using I/O MMU:

- **Increase in system reliability**: Isolation capable IOMMUs perform a valuable system service by preventing rogue devices from performing errant or malicious DMAs, thereby substantially increasing the system's reliability and availability
- **Secure Access**: Operating systems utilize IOMMUs to isolate device drivers; hypervisors utilize IOMMUs to grant secure direct hardware access to virtual machines
- **Efficiency**: I/O MMU allows decoupling which enables time- and space-multiplexing of I/O devices, allowing multiple logical devices to be implemented by a smaller number of physical devices. The ability to multiplex logical I/O devices onto physical ones allows both administrators and automated systems to drive I/O devices at higher utilization and achieve better hardware efficiency.
- **Ease in Live Migration**: Decoupling enables popular VM features such as the ability to suspend and resume a VM and the ability to move a running VM between physical machines, known as live migration.
- **Device aggregation**: Multiple physical devices can be combined into a single more capable logical device that is exported to the VM. Example: combining multiple disk storage devices exported as a single larger disk.
- **Feature Enhancement**: New features can be added to existing systems by interposing and transforming virtual I/O requests, transparently enhancing unmodified software with new capabilities. Example: A disk write can be transformed into replicated writes to multiple disks, so that the system can tolerate disk-device failures.
- **Improvement in System Security**: When an encryption function is run over the I/O to and from a disk to implement transparent disk encryption, interposing on network traffic allows virtualization layers to implement advanced networking security, such as firewalls and intrusion-detection systems employing deep packet inspection.

Disadvantages of using I/O MMU:

- **Penalty in Performance**: Although they provide valuable services, IOMMUs can impose a performance penalty due to the extra memory accesses required to perform DMA operations. The performance of two IOMMUs, DART on PowerPC and Calgary on x86-64, presented through running IO-intensive benchmarks with and without an IOMMU on the IO path. CPU utilization rises when the IOMMU is enabled, leading to up to 60% more CPU utilization.
- **Risk in Safety**: Direct memory access (DMA) illustrates additional safety and performance issues. It enables an I/O device to read and write host RAM directly without involving the CPU, which is critical for achieving high-performance I/O rates.
- **Complexity in resource-management**: Multiplexing physical devices across multiple VMs pose complexities in resource-management like Scheduling and prioritization.
- **Additional challenges**: Although each VM is presented with the illusion of having its own dedicated virtual hardware, in reality the hypervisor must multiplex limited physical hardware across multiple VMs of varying importance, mapping their virtual resources onto available physical resources.
- **Example**: CPU utilization, could be as much as 60% more in a hypervisor environment and 30% more in a bare-metal environment, when the IOMMU was enabled; according to performance of two IOMMUs, DART on PowerPC and Calgary on x86-64

Source: [https://cacm.acm.org/magazines/2012/1/144808-i-o-virtualization/fulltext#comments]

[The Price of Safety: Evaluating IOMMU Performance:Ben-Yehuda, Xenidis, Mostrows, Rister, Bruemmer, Van Doorn: https://www.kernel.org/doc/mirror/ols2007v1.pdf]

6.2) Research the Web to find what is meant by "carrier-grade hypervisors". What products are available?

Solution:

- According to Wikipedia, a "carrier-grade" or "carrier-class" refers to a system, or a hardware or software component that is extremely reliable, well tested and proven in its capabilities.
- SCOPE, an industry alliance whose mission is to help, enable and promote availability of open carrier grade base platforms based on COTS) hardware / software and FOSS building blocks, and to promote interoperability has defined the following characteristics of Carrier Grade Hypervisors:
    - High-performance scaling with the amount of hardware, and supporting large number of transactions and simultaneous sessions.
    - Small and controlled error recovery domains supporting effective error isolation.
    - Real-time behavior for signaling and call control domains.

- Failover capabilities to let user sessions survive failures.
- Hardware and software upgrade and error correction capabilities with minimal service interruption.
- Configurable security to provide high level of safety and protection Controlled and extended life cycle of hardware and software components.
- Efficient and uniform management interfaces.
- Cost effective operation and maintenance.
- Speedy application development and testing.
- Easy to learn and well-documented.

Source: [http://www.linuxpundit.com/documents/CGV_WP_Final_FN.pdf]

Examples of "carrier-grade" hypervisors:

- VirtualLogix Carrier Grade Hypervisor
- Wind River "Carrier Grade Linux"
- Oracle Solaris
- NEC Carrier Grade HyperVisor

7. (5 points) Find out what hypervisors Amazon is using in EC2, and describe their major characteristics.

Solution:

- ✓ All new instance types in EC2 will use the Nitro Hypervisor, but in the near term, some new instance types will use Xen depending on the requirements of the platform.
- ✓ Initially, EC2 used Xen virtualization, but since 2017, Nitro Hypervisor is used.
- ✓ As a component of the Nitro system, the Nitro Hypervisor primarily provides CPU and memory isolation for EC2 instances. VPC networking and EBS storage resources are implemented by dedicated hardware components, Nitro Cards that are part of all current generation EC2 instance families. The Nitro Hypervisor is built on core Linux Kernel-based Virtual Machine (KVM) technology, but does not include general-purpose operating system components.

The major characteristics of Nitro Hypervisors are:

- ➢ **Identical with Bare-metal hypervisor**: The Nitro Hypervisor is a lightweight hypervisor that manages memory and CPU allocation and delivers performance that is indistinguishable from bare metal.
- ➢ **Increased Performance**: The Nitro Hypervisor provides consistent performance and increased compute and memory resources for EC2 virtualized instances by removing host system software components.
- ➢ **Supports larger instances**: It allows AWS to offer larger instance sizes (like c5.18xlarge) that provide practically all of the resources from the server to customers.

- **Other features**: Previously, C3 and C4 instances each eliminated software components by moving VPC and EBS functionality to hardware designed and built by AWS. This hardware enables the Nitro Hypervisor to be very small and uninvolved in data processing tasks for networking and storage.

Source: [https://aws.amazon.com/ec2/faqs/]

8. (10 points) Examine the Amazon EC2 VM offer capabilities and particularly the Amazon Machine Image (AMI) ((https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html) and answer the following questions:

a. How (i.e., in what units) does EC2 measure the CPU power of a virtual machine and how is the unit in question translated into the power of the physical processors)?

b. What kinds of machine instances are there as characterized by the power of their respective CPUs, platform (i.e., 32-bit or 64-bit), memory, storage, etc.? Please list all the instances in the nomenclature along with their respective characteristics;

c. Which operating systems are available on the above systems?

d. What is an AMI and what is its relationship to an instance?

e. What are the components of an AMI?

Solution:

a.

- Amazon EC2 uses a variety of measures to provide each instance with a consistent and predictable amount of CPU capacity. The amount of CPU that is allocated to a particular instance is expressed in terms of these EC2 Compute Units. The EC2 Compute Unit (ECU) provides the relative measure of the integer processing power of an Amazon EC2 instance.
- Although physical processors are measured in GHz unit of frequency, Amazon does not disclose the exact vCPU or EC2 Compute Units to GHz conversions.
- Amazon EC2 instances support multithreading, which enables multiple threads to run concurrently on a single CPU core. Each thread is represented as a virtual CPU (vCPU) on the instance. An instance has a default number of CPU cores, which varies according to instance type. Default vCPUs= Default CPU cores* Default threads per core
- For example, for a f1.4xlarge instance type has 8 default CPU cores and 2 default threads per core; the default vCPUs will be 8*2=16. In this case, the valid number of CPU cores will be 1, 2, 3, 4, 5, 6, 7, 8 and valid number of threads per core will be 1,2.

b.

When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities.

Types of instances:

- **General purpose instances** provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.
    - A1 instances are the first EC2 instances powered by AWS Graviton Processors that feature 64-bit Arm Neoverse cores and custom silicon designed by AWS.
    - T3 instances are the next generation burstable general-purpose instance type that provide a baseline level of CPU performance with the ability to burst CPU usage at any time for as long as required.
    - T3a instances are the next generation burstable general-purpose instance type that provide a baseline level of CPU performance with the ability to burst CPU usage at any time for as long as required.
    - T2 instances are Burstable Performance Instances that provide a baseline level of CPU performance with the ability to burst above the baseline.
    - Amazon EC2 M6g instances are powered by Arm-based AWS Graviton2 processors.
    - M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175 processors.
    - M5a instances are the latest generation of General Purpose Instances powered by AMD EPYC 7000 series processors.
    - M5 instances are ideal for workloads that require a balance of compute, memory, and networking resources including web and application servers, small and mid-sized databases, cluster computing, gaming servers, and caching fleet.
    - M4 instances provide a balance of compute, memory, and network resources, and it is a good choice for many applications.

- **Compute Optimized instances** are ideal for compute bound applications that benefit from high performance processors.
    - C5 instances are optimized for compute-intensive workloads and deliver cost-effective high performance at a low price per compute ratio.
    - C5n instances are ideal for high compute applications (including High Performance Computing (HPC) workloads, data lakes, and network appliances such as firewalls and routers) that can take advantage of improved network throughput and packet rate performance.
    - C4 instances are optimized for compute-intensive workloads and deliver very cost-effective high performance at a low price per compute ratio.

- **Memory optimized instances** are designed to deliver fast performance for workloads that process large data sets in memory.
    - R5 instances deliver 5% additional memory per vCPU than R4 and the largest size provides 768 GiB of memory.
    - R5a instances are the latest generation of Memory Optimized instances ideal for memory-bound workloads and are powered by AMD EPYC 7000 series processors.
    - R5 instances are ideal for memory-bound workloads including high performance databases, distributed web scale in-memory caches, mid-sized in-memory database, real time big data analytics, and other enterprise applications.
    - R4 instances are optimized for memory-intensive applications and offer better price per GiB of RAM than R3.
    - X1e instances are optimized for high-performance databases, in-memory databases and other memory intensive enterprise applications.
    - X1 instances are optimized for large-scale, enterprise-class and in-memory applications, and offer one of the lowest prices per GiB of RAM among Amazon EC2 instance types.
    - High memory instances are purpose built to run large in-memory databases, including production deployments of SAP HANA, in the cloud.
    - Amazon EC2 z1d instances offer both high compute capacity and a high memory footprint.
- **Accelerated computing instances** use hardware accelerators, or co-processors, to perform functions, such as floating-point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.
    - P3 instances are the latest generation of general purpose GPU instances
    - P2 instances are intended for general-purpose GPU compute applications.
    - Amazon EC2 Inf1 instances are built from the ground up to support machine learning inference applications.
    - G4 instances are designed to help accelerate machine learning inference and graphics-intensive workloads.
    - G3 instances are optimized for graphics-intensive applications.
    - F1 instances offer customizable hardware acceleration with field programmable gate arrays (FPGAs).
- **Storage optimized instances** are designed for workloads that require high, sequential read and write access to very large data sets on local storage
    - I3 provides Non-Volatile Memory Express (NVMe) SSD-backed instance storage optimized for low latency, very high random I/O performance, high sequential read throughput and provide high IOPS at a low cost.
    - I3en: This instance family provides dense Non-Volatile Memory Express (NVMe) SSD instance storage optimized for low latency, high random I/O performance, high sequential disk throughput, and offers the lowest price per GB of SSD instance storage on Amazon EC2.

- D2 instances feature up to 48 TB of HDD-based local storage, deliver high disk throughput, and offer the lowest price per disk throughput performance on Amazon EC2.
- H1 instances feature up to 16 TB of HDD-based local storage, deliver high disk throughput, and a balance of compute and memory.

Source: [https://aws.amazon.com/ec2/instance-types/]

c.

Amazon EC2 currently supports a variety of operating systems including: Amazon Linux, Ubuntu, Windows Server, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Fedora, Debian, CentOS, Gentoo Linux, Oracle Linux, and FreeBSD.

- ✓ General purpose instances: The following AMIs are supported on A1 instances: Amazon Linux 2, Ubuntu 16.04.4 or newer, Red Hat Enterprise Linux (RHEL) 7.6 or newer, SUSE Linux Enterprise Server 15 or newer. Additional AMI support for Fedora, Debian, NGINX Plus are also available through community AMIs and the AWS Marketplace.

- ✓ Compute-optimized instances: The following AMIs are supported on C5: Amazon Linux 2014.03 or newer, Ubuntu 14.04 or newer, SUSE Linux Enterprise Server 12 or newer, Red Hat Enterprise Linux 7.4 or newer, CentOS 7 or newer, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, FreeBSD 11.1-RELEASE

- ✓ Memory-optimized instances: SUSE Linux 12, SUSE Linux 12 SP1, SLES for SAP 12 SP1, SLES for SAP 12 SP2, and RHEL 7.2 for SAP HANA are used.

- ✓ Accelerated computing instances: On P3, P2, G3 instances, Windows Server, SUSE Enterprise Linux, Ubuntu, and Amazon Linux AMIs on P2 and G3 instances

Source: [https://aws.amazon.com/ec2/faqs/]

d.

- An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications).
- An instance is a copy of the AMI running as a virtual server in the cloud.
- An instance can be launched from an AMI; different types of instances can be launched from a single AMI.

e.

An Amazon Machine Image (AMI) provides the information required to launch an instance. Components of an AMI are:

i. One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
ii. Launch permissions that control which AWS accounts can use the AMI to launch instances.
iii. A block device mapping that specifies the volumes to attach to the instance when it's launched.

Source: [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html]

9. (10 points) Find out about the pricing of the EC2 platforms and provide a few examples.

Solution:

There are five ways to pay for Amazon EC2 instances:

- **On-Demand:** You pay for compute capacity by the hour or the second depending on which instances you run. No longer-term commitments or upfront payments are needed. Example: t3.medium running Windows with 2 virtual CPUs and 4 GiB memory would cost around $0.056 per hour.
  Source: [https://aws.amazon.com/ec2/pricing/on-demand/]
- **Savings Plans**: This plan offers significant savings over On Demand, just like EC2 Reserved Instances, in exchange for a commitment to use a specific amount of compute power (measured in $/hour) for a one or three year period. For t3.medium running Windows would cost $0.0484 in Savings plan rate, whereas the same would have cost $0.06 if bought on-demand. That's a 19% in savings.
  Source: [https://aws.amazon.com/savingsplans/pricing/]
- **Reserved Instances:** Amazon EC2 Reserved Instances (RI) provide a significant discount (up to 75%) compared to On-Demand pricing and provide a capacity reservation when used in a specific Availability Zone. Example: For t3.medium, if the payment is made upfront, it would cost $0.024 which is 42% savings over on-demand.
  Source:[https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/]
- **Spot Instances**: User pays the Spot price that's in effect for the time period instances are running. Spot Instance prices are set by Amazon EC2 and adjust gradually based on long-term trends in supply and demand for Spot Instance capacity. Example: for t3.medium for Windows usage would cost $0.0309 per hour
  Source: [https://aws.amazon.com/ec2/spot/pricing/]
- **Dedicated Hosts**: Provide users with EC2 instance capacity on physical servers dedicated for use. The price for a Dedicated Host varies by instance family, region, and payment option. Regardless of the quantity or the size of instances that user chooses to launch on a particular Dedicated Host user pays hourly for each active Dedicated Host, and user is not billed for instance usage. Example: The table shows pricing for p2 instance for Dedicated host pricing model:

p2

STANDARD 1-YEAR TERM					
Payment Option	Upfront	Monthly*	Effective Hourly**	Savings over On-Demand	On-Demand Hourly
No Upfront	$0	$7,892.03	$10.811	32%	$15.8400
Partial Upfront	$45,096	$3,758.04	$10.296	35%	
All Upfront	$88,389	$0.00	$10.090	36%	

Source: [https://aws.amazon.com/ec2/dedicated-hosts/pricing/]

- Per-second billing: With per-second billing, you pay for only what you use. It takes cost of unused minutes and seconds in an hour off of the bill, so you can focus on improving your applications instead of maximizing usage to the hour.
  Source: [https://aws.amazon.com/ec2/pricing/]
- Free Tier: AWS Free Tier includes 750 hours of Linux and Windows t2.micro instances each month for one year. To stay within the Free Tier, use only EC2 Micro instances.
  Source: [https://aws.amazon.com/ec2/pricing/]

10. (15 points) From the above exercise, you will learn that it is possible to create a free machine instance. Please, do the following:

a. Find out and document the essence of the respective Service Level Agreement (SLA); in particular write down what one needs to do in order to maintain this service free;
b. Describe the process (i.e., what exactly one needs to do) to create a free machine instance that could be used as a server. (Do not, however, create anything yet!)
c. Can you create a machine instance equivalent to your own PC and then transfer your own PC image there? If so, how would you achieve that?

Solution:

a. The Amazon Compute Service Level Agreement (the "SLA") is a policy governing the use of the Amazon EC2 and applies separately to each account using the Amazon EC2. Depending on the product being used, there are three types of offers available under Free Tier: Always free, 12 months free and Trials. Free Tier includes 750 hours of Linux and Windows t2.micro instances each month for one year. To stay within the Free Tier, use only EC2 Micro instances.
   i. An account needs to be created on Amazon EC2.
   ii. Credit card information has to be entered in the billing section.
   iii. In order to remain in the "Free Tier", only EC2 Micro instances can be used and the limit is 750 hours per month.

Source: [https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc]

b. An instance is a virtual server in the AWS cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

   i. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   ii. From the console dashboard, choose **Launch Instance**.
   iii. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instance. Select an HVM version of Amazon Linux 2. Notice that these AMIs are marked "Free tier eligible."
   iv. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the t2.micro type, which is selected by default. Notice that this instance type is eligible for the free tier.
   v. Choose **Review and Launch** to let the wizard complete the other configuration settings for you. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
   vi. Choose **Edit security groups**.
   vii. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
   viii. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
   ix. On the **Review Instance Launch** page, choose **Launch**.
   x. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**.
   xi. When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

Source:[https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html]

c. Yes, I can create a machine instance equivalent to my own PC and then transfer my own PC image there using Amazon EC2.

   i. First, I will create an AWS account.
   ii. I will then launch an Amazon EC2 instance as described in 10.(b)
   iii. Then, I will configure my instance by choosing an Amazon Machine Image (AMI); since my PC is Windows, I will select a Windows AMI.

iv. Then I will choose t2.micro (Since I want to be in Free Tier)
v. After Launching, I will create a new key pair and download and store it in and as C:\user\Tehreem\.ssh\MyKeyPair.pem
vi. After these steps, I will launch the instance.
vii. I will then Connect to my instance.
viii. Then I will connect my PC to the virtual machine and transfer its image.

Source: [https://aws.amazon.com/getting-started/tutorials/launch-a-virtual-machine/]