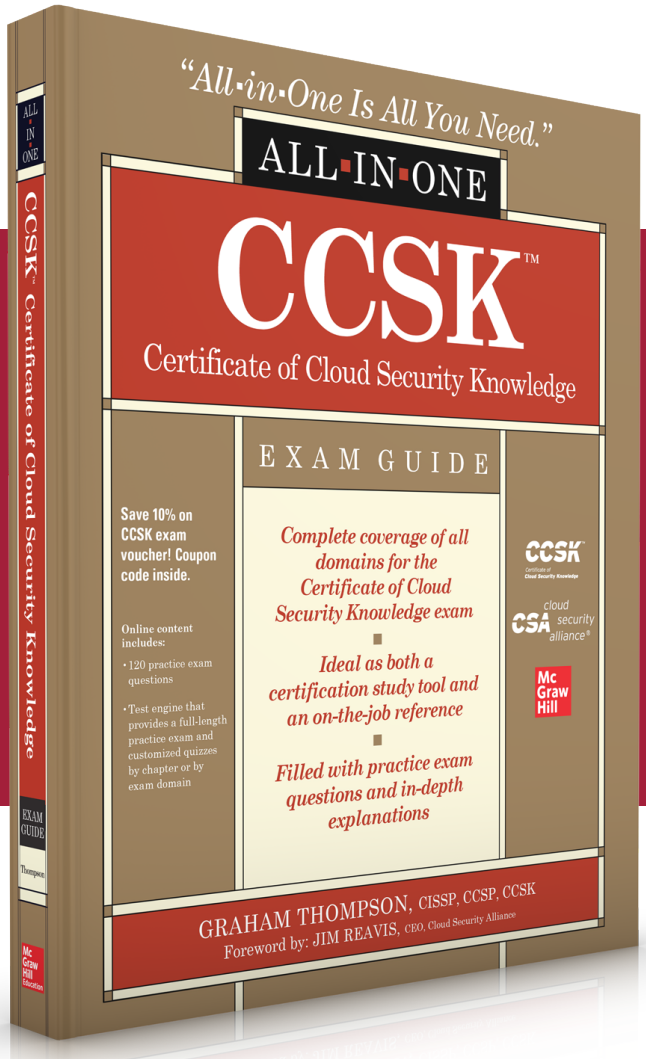
Mc Graw Hill

CCSK™
Certificate of Cloud Security Knowledge

Sample Chapter

CHAPTER 8:
Virtualization and Containers



LEARN MORE

BUY NOW

CHAPTER 8

Virtualization and Containers

This chapter covers the following topics from Domain 8 of the CSA Guidance:

- Major Virtualization Categories
- Security of Compute Virtualization
- Security of Network Virtualization
- Security of Storage Virtualization
- Container Components
- Security of Containers and Related Components

Companies in every industry need to assume that a software revolution is coming.
—Marc Andreessen

Yes, the opening quote in this chapter is the same quote used in Chapter 7. I did this intentionally to highlight the fact that this chapter builds on the topic of the software revolution introduced in Chapter 7. It expands the discussion to address the security issues surrounding virtualization and the responsibility split between the provider and the customer. The material is presented in this fashion to provide consistency with the CSA Guidance and the CCSK exam. If a subject has been addressed as part of another chapter, however, I'll point you there as a refresher.

NOTE As always, backgrounder material is included in this chapter to address any subject material that isn't covered as part of the CSA Guidance. This material is presented to improve your understanding of core technologies and concepts that the CSA Guidance assumes you already know; it's not required for the CCSK exam.

As you now know, virtualization is a core enabling technology in the cloud, and it's not just about virtual machines. Virtualization is how compute, network, and storage pools are created and is the enabling technology behind the multitenancy aspect of cloud services. In this chapter, we look at the responsibility split for securing various virtualized technologies—what to look for from the provider and what you need to do as

the consumer. You'll also learn about the various components involved with containers and some approaches to securing them.

NOTE Virtualization provides the abstraction needed for resource pools, which are then managed using orchestration. Without virtualization, there is no cloud.

Much of the security we use today in an IT world assumes that we have physical control of the underlying infrastructure. This hasn't changed; what has changed with the cloud is that the provider is responsible for securing the physical infrastructure. In addition, virtualization adds two new layers for security controls:

- Security of the virtualization technology itself, such as hypervisor security. This rests with the provider.
- Security controls for the virtual assets. The responsibility for implementing available controls rests with the customer. Exposing controls for the customers to leverage is the provider's responsibility.

Major Virtualization Categories Relevant to Cloud Computing

The main areas of virtualization that you need to know for your exam are straightforward: Compute, Network, Storage. Each of the three creates its own storage pools, and those pools are possible only as a result of virtualization; it makes sense, then, that these are areas of focus for the CCSK exam. The following sections will refresh your memory on each of these pools. Then I will cover how the CSA splits the security responsibilities of each technology.

Compute Virtualization

You know about virtual machines and hypervisors already, so consider this a refresher. Compute virtualization abstracts the running of code (including operating systems) from the underlying hardware. Instead of running code directly on the hardware, it runs on top of an abstraction layer (such as a hypervisor) that isolates (not just segregation!) one virtual machine (VM) from another. This enables multiple operating systems (guest OSs) to run on the same hardware.

EXAM TIP For the exam, remember that compute virtualization abstracts the running of code (including operating systems) from the underlying hardware.

Although compute virtualization is generally tied to virtual machines, there is more to it than VMs (or, more appropriately, instances, when discussing the cloud). An older form

of virtualization that you may be aware of is the Java Virtual Machine (JVM). Rather than go too deeply into JVMs, I'll limit this discussion by simply saying the JVM creates an environment for a Java application to run in. The JVM abstracts the underlying hardware from the application. This allows for more portability across hardware platforms, because the Java app does not need to communicate directly with the underlying hardware, only with the JVM. There are, of course, many other examples of virtualization out there, but the big takeaway is that virtualization performs abstraction.

NOTE Java Virtual Machines are called out in the CSA Guidance as a form of compute virtualization.

The next era of compute virtualization that needs to be addressed is about containers and serverless computing technologies. Both technologies also perform some form of compute abstraction. I cover serverless computing in greater depth in Chapter 14.

Cloud Provider Responsibilities

The primary security responsibilities of the cloud provider in compute virtualization are to enforce isolation and maintain a secure virtualization infrastructure. Isolation ensures that compute processes or memory in one virtual machine/container are not visible to another. This isolation supports a secure multitenant model, where multiple tenants can run processes on the same physical hardware (such as a single server). The cloud provider is also responsible for securing the underlying physical infrastructure and the virtualization technology from external attack or internal misuse. Like any other software, hypervisors need to be properly configured and will require the latest patches installed to address new security issues.

Cloud providers should also have strong security in place for all aspects of virtualization for cloud users. This means creating a secure chain of processes from the image (or other source) used to run the virtual machine all the way through a boot process, with security and integrity being top concerns. This ensures that tenants cannot launch machines based on images that they shouldn't have access to, such as those belonging to another tenant, and that when a customer runs a virtual machine (or another process), it is the one the customer expects to be running.

Finally, cloud providers should also assure customers that volatile memory is safe from unapproved monitoring, since important data could be exposed if another tenant, a malicious employee, or a bad actor is able to access running memory belonging to another tenant.

EXAM TIP Remember that volatile memory contains all kinds of potentially sensitive information (think unencrypted data, credentials, and so on) and must be protected from unapproved access. Volatile memory must also have strong isolation implemented and maintained by the provider.

Cloud Consumer Responsibilities

The primary responsibility of the cloud user is to implement security properly for everything deployed and managed in a cloud environment. Cloud customers should take advantage of the security controls exposed by their providers for managing their virtual infrastructures. Of course, there are no rules or regulations as to what a provider must offer customers, but some controls are usually offered.

Cloud providers offer security settings such as identity and access management (IAM) to manage virtual resources. When you're considering the IAM offered by the provider, remember that this is generally at the management plane, not the applistructure. In other words, we're talking about the ability for your organization's users accessing the management plane to be given the appropriate permissions required to start or stop an instance, for example, not log on to the server itself. For a refresher, review the "Securing the Management Plane" section in Chapter 6.

Cloud providers will also likely offer logging of actions performed at the metastructure layer and monitoring of workloads at the virtualization level. This can include the status of a virtual machine, performance (such as CPU utilization), and other actions and workloads.

IAM is as important in a cloud environment as it is in a traditional data center. Cloud compute deployments are based on master images—a virtual machine, container, or other code—that are then run as an instance in the cloud. Just as you would likely build a server in your data center by using a trusted, preconfigured image, you would do the same in a cloud environment. Some Infrastructure as a Service (IaaS) providers may have "community images" available. But unless they are supplied by a trusted source, I would be very hesitant to use these in a production environment, because they may not be inspected by the provider for malicious software or back doors being installed by a bad actor who's waiting for someone to use them. Managing images used by your organization is one of your most vital security responsibilities.

Another option that providers may offer is that of "dedicated instances" or "dedicated hosting." This usually comes at an increased cost, but it may be a useful option if the perceived risk of running a workload on hardware shared with another tenant is deemed unacceptable, or if there is a compliance requirement to run a workload on a single-tenant server.

TIP Dedicated instances may have various limitations associated with them. For one thing, although the workload may be running on single-tenant hardware, your data is likely stored in a multitenant storage environment. You may also have other technical restrictions in place, such as not all services being supported or available for dedicated instances. This area requires that you fully understand what the provider is really offering when they offer "dedicated" anything.

Finally, the customer is responsible for the security of everything within the workload itself. All the standard stuff applies here, such as starting with a secure configuration of

the operating system, securing any applications, updating patches, using agents, and so on. The big difference for the cloud has to do with proper management of the images used to build running server instances as a result of the automation of cloud computing. It is easy to make the mistake of deploying older configurations that may not be patched or properly secured if you don't have strong asset management in place.

Other general compute security concerns include these:

- Virtualized resources tend to be more ephemeral and can change at a more rapid pace. Any corresponding security, such as monitoring, must keep up with the pace.
- Host-level monitoring/logging may not be available, especially for serverless deployments. Alternative log methods such as embedding logging into your applications may be required.

Network Virtualization

This section covers much of what was covered in Chapter 7, so I'm going to make this as brief as possible for you. You know there are multiple network virtualization technologies out there, ranging from virtual LANs (VLANs) to software defined networking (SDN). By now, you understand that "software-driven everything" is the way the industry is going. The software-driven aspect is a key contributor for the resource pooling, elasticity, and all other aspects that make the cloud work at the scale it does.

I'll be touching on a few items regarding security of virtual environments and will then cover the responsibilities of each party (provider and customer). First up is filtering and monitoring of virtual networks.

Filtering and Monitoring

If network traffic between two VMs never leaves a physical computer, is it inspected and filtered by an external physical firewall? It's not, so where does that leave us? We still have a requirement to perform inspection and filtering of network traffic, but we can no longer use the same security controls we have used in the past. Back in the early days of virtualization, some people thought it was a good idea to send all virtual network traffic out of the virtual environment, inspect the traffic using a physical firewall, and then reintroduce it back to the virtual network. Newer virtual approaches to address this problem could include routing the virtual traffic to a virtual inspection machine on the same physical server or routing the network traffic to a virtual appliance on the same virtual network. Both approaches are feasible, but they still introduce bottlenecks and require less efficient routing.

Remember that all is not lost. The provider will most likely offer some form of filtering capability, be it through the use of an SDN firewall or within the hypervisor.

NOTE Remember that any appliance, virtual or physical, can be a bottleneck and/or a single point of failure.

From a network monitoring perspective, don't be surprised if you can't get the same level of detail about network traffic from the provider that you had in the past in your own environment. This is because the cloud platform/provider may not support access for direct network monitoring. They will state that this is because of complexity and cost. Access to raw packet data will be possible only if you collect it yourself in the host or by using a virtual appliance. This accounts only for network traffic that is directed to, or originates from, a system that you control. In other environments, such as systems managed by the provider, you will not be able to gain access to monitor this network traffic, because this would be a security issue for the provider.

Management Infrastructure

By default, the virtual network management plane is available to the entire world, and if it's accessed by bad actors, they can destroy the entire virtual infrastructure in a matter of seconds via an API or web access. It is therefore paramount that this management plane be properly secured. See "Securing the Management Plane" in Chapter 6 if you need a refresher on this critical subject.

Cloud Provider Responsibilities

As with compute virtualization in a cloud environment, virtual networks have a shared responsibility. I'll begin with the responsibilities of the provider and then move to the customer responsibilities for network virtualization.

The absolute top security priority is segregation and isolation of network traffic to prevent tenants from viewing another tenant's traffic. At no point should one tenant ever be able to see traffic from another tenant unless this is explicitly allowed by both parties (via cross-account permissions, for example). This is the most foundational security control for any multitenant network.

Next, packet sniffing (such as using Wireshark), even within a tenant's own virtual networks, should be disabled to reduce the ability of an attacker to compromise a single node and use it to monitor the network, which is common in traditional networks. This is not to say that customers cannot use some packet-sniffing software on a virtual server, but it means the customers should be able to see traffic sent only to a particular server.

In addition, all virtual networks should offer built-in firewall capabilities for cloud users without the need for host firewalls or external products. The provider is also responsible for detecting and preventing attacks on the underlying physical network and virtualization platform. This includes perimeter security of the cloud itself.

Cloud Consumer Responsibilities

The consumer is ultimately responsible for adhering to their own security requirements. Quite often, this will require consuming and configuring security controls that are created and managed by the cloud provider, especially any virtual firewalls. Here are several recommendations for consumers when it comes to securing network virtualization.

Take advantage of new network architecture possibilities. For example, compartmentalizing application stacks in their own isolated virtual networks to enhance security can be performed at little to no cost (aside from operational costs, which will likely

go up). Such an implementation may be cost prohibitive in a traditional physical network environment.

Next, software defined infrastructure (SDI) includes the ability to create templates of network configurations. You can essentially take a known-good network environment and save it as software. This approach enables you to rebuild an entire network environment incredibly quickly if needed. You can also use these templates to ensure that your network settings remain in a known-good configuration.

Finally, when the provider doesn't expose appropriate controls for customers to meet their own security requirements, customers will need to implement additional controls (such as virtual appliances or host-based security controls) to meet their requirements.

Cloud Overlay Networks

I'm including this section because I want to make sure the mappings to the actual CSA Guidance document remain intact. Like much of the content of this chapter, cloud overlay networks were covered in Chapter 7. It's important for you to remember that cloud overlay networks are a function of the Virtual Extensible LAN (VXLAN) technology, and they enable a virtual network to span multiple physical networks across a wide area network (WAN). This is possible because VXLAN encapsulates packets in a routable format. Note that the CSA Guidance specifically states that this technology is beyond the scope of the necessary material and it will therefore not be part of your CCSK exam.

Storage Virtualization

Storage virtualization sounds like new technology, but it really isn't: I present to you, by way of example, the classic redundant array of independent disks (RAID), a storage virtualization method that has been around for many years (since the 1970s) and can be implemented in any operating system today. I'm going to spare you a discussion of the various RAID levels, but I am going to say that RAID 0 (stripe set), for example, enables you to take three 1TB hard drives and make them look like a single 3TB hard drive. What could you call that? How about a pool of storage? Yes, that should work. Using software RAID, you virtualize your storage by joining drives together virtually to form a storage pool. (This example is courtesy of Windows NT in the mid-'90s—look it up if you need to.)

You know that the concept of storage virtualization has been around for quite a while, but how is it done in a cloud environment? Well, chances are pretty good that providers aren't using just a general-purpose server with a handful of drives installed (aka direct attached storage using "just a bunch of drives"). They are likely using network attached storage (NAS) or storage area networks (SANs) to form these pools. As SAN is often a bit of a gray area for many people, the next section is a high-level backgrounder on the subject. If you are familiar with SAN, or you just don't care, you can skip it.

Storage Area Network Backgrounder

Research shows that SANs account for about two-thirds of the network storage market. The key word here is "network." The SAN is a dedicated network of storage devices. It is a combination of hardware and software that offers a block-level storage mechanism.

It doesn't offer a file system, but it stores and send blocks of data as requested by authorized servers. As such, it cannot be accessed directly by a user on a workstation; like most everything virtualized, the access is abstracted, and actions are orchestrated.

From a high-level architectural perspective, the SAN reference architecture can generally be divided into three layers: host layer, fabric layer, and storage layer. Figure 8-1 shows the three layers typically involved in a SAN.

Each layer serves a specific purpose:

- **Host layer** This is where servers (or hosts) take calls from the local area network (LAN) and enable access to/from the underlying SAN fabric.
- **Fabric layer** This is where all the networking components live. SAN network devices include switches, routers, bridges, gateways, and even cables.
- **Storage layer** As you can imagine, this is where the actual storage devices live. Direct physical communication generally (still!) occurs using Small Computer System Interface (SCSI).

SAN uses its own protocols to do its job. These protocols are Fibre Channel, Fibre Channel over Ethernet (FCoE), Internet SCSI (iSCSI), and InfiniBand. All of these are purpose-built to transfer blocks of data at high speeds and have higher throughput than Transmission Control Protocol (TCP) networks do. Fibre Channel is considered the most popular protocol in large SAN environments, with some reports stating that up to 80 percent of the SANs in use today use Fibre Channel, whose speeds can theoretically reach up to 128 Gbps.

As the name might imply, Fibre Channel uses fiber-optic cables. To connect to the SAN, host bus adapters (HBAs) or converged network adapters (CNAs) are used. Both of these network cards will have a fiber-optic connector. The CNA will have both a fiber-optic connector and a standard Ethernet network adapter.

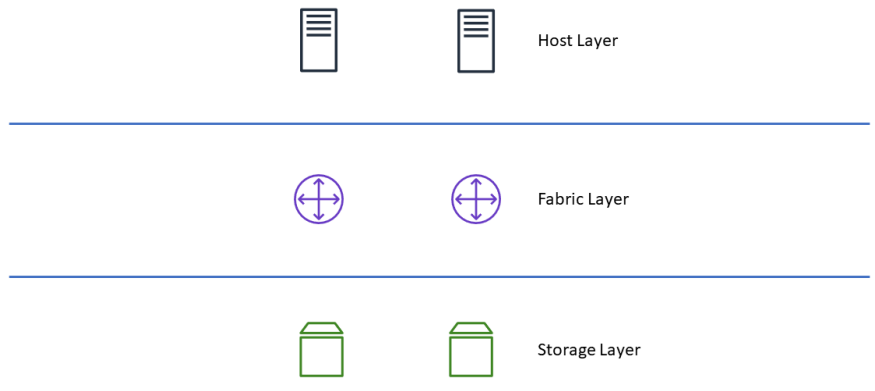


Figure 8-1 Three layers of a SAN

To minimize expense and additional optical network cables from being required, companies may opt to use standard Ethernet to transmit SAN traffic in a *converged network*. The two main protocols that allow for this are FCoE and iSCSI. Both encapsulate SCSI commands in an Ethernet frame and use standard Ethernet cables for transport. iSCSI is generally considered useful in smaller environments, but it's not appropriate for high-volume SANs.

Logical Unit Numbers Now that you know the high-level architecture of a SAN and some of the components and the protocols used in storage virtualization, let's look at another SAN item that you should be aware of: the *logical unit number* (LUN). A LUN is assigned to every drive and partition in a SAN. Let's say, for example, that you have a large SAN that's 1000TB in size. You don't want everyone in your company (or cloud) to have access to a 1000TB drive that everything is just dumped into. Instead, you want to divide that large SAN into smaller 1TB volumes. You would do this by creating the 1TB logical volumes and associating a LUN to them so they can be accessed (and controlled, which you'll see in a little bit). The LUN is used to present a logical drive to a host server, giving the host abstracted access to a reserved space in a pool of storage.

NOTE LUN is not just a SAN thing. The concept of LUNs goes back to the SCSI technology standard that defines how storage is accessed, so it is applicable in many forms of storage, be it storage arrays from the '90s all the way up to the latest SAN technology.

You may wonder how you would limit access to these virtual drives to appropriate requestors. The answers for restricting access to storage come in zoning and LUN masking.

Zoning allows for a logical grouping of ports or nodes that restricts certain hosts to accessing only specified storage devices. This is usually configured on the Fibre Channel switch in the SAN fabric. In an FCoE implementation, this would be performed in the same manner as a VLAN. There are two different ways to set up zoning, soft zoning and hard zoning:

- Soft zoning is performed in software on the Fibre Channel switches to prevent ports from being seen from outside of their assigned zones. The security issue with soft zoning is that views are filtered and not physically blocked. Therefore, unauthorized ports can be accessed if a Fibre Channel address is spoofed.
- Hard zoning implements restrictions in hardware and physically blocks access to a zone from any unauthorized device. Hard zoning is considered the more secure approach to block communication between devices not in the same zone.

LUN masking is performed in addition to the zoning to provide additional security to the storage environment. Remember that zoning establishes which hosts and storage devices are grouped together, and it restricts access to zone members. However, as you know, a storage device could contain multiple logical drives. LUN masking is used to

identify the virtual drives that can be accessed within a zone. LUN masking can be performed at the host bus adapter (HBA) of a host server or on a storage controller. From a security perspective (because of spoofing), it is generally considered that LUN masking should be enforced at the storage controller.

Storage Virtualization Security

Most cloud platforms use highly redundant and durable storage mechanisms that make multiple copies of data and spread those copies across multiple storage locations. This is called *data dispersion*. This approach enables the provider to offer incredible levels of resiliency (some providers even offer "11 9's," or 99.999999999 percent, resiliency SLAs).

NOTE Resiliency and availability aren't the same thing. Data can be inaccessible if the network is down. The data is still there (resiliency), but it cannot be accessed (availability).

Providers will usually encrypt all customer data at the physical level, which doesn't protect data at the virtual level, but does protect data on a drive that is decommissioned and is awaiting destruction (and, of course, it protects the data if the drive is stolen by a rogue administrator).

I will address additional security measures that customers can use (such as encryption, access controls, and more) to protect stored data in Chapter 11.

Containers

In Chapter 7, I mentioned that containers could help address portability but that this technology relies on more than just source code and that all components need to be properly secured. This section covers the various components of a container system and the high-level security recommendations from the CSA. Note that although container technology is fairly mature, it is a rapidly evolving technology.

You know that containers are a compute virtualization technology and that they differ from virtual machines in that only the application and required dependencies are bundled in a container, which is then run in an isolated user space on a shared kernel. Containers can run directly on a physical server (even a laptop), or they can run in a virtual machine.

NOTE A container is an abstraction at the application layer that isolates software from its environment. Containers don't necessarily provide full-stack security isolation, but they do provide task segregation. On the other hand, virtual machines typically do provide security isolation. You can put tasks of equivalent security context on the same set of physical or virtual hosts to provide greater security segregation.

Container systems always have the following components:

- **Container** This is the execution environment itself. The container provides code running inside a restricted environment with access only to the processes and capabilities defined in the container configuration via a configuration file (covered later in this chapter). While a VM is a full abstraction of an operating system, a container is a constrained place to run segregated processes while still utilizing the kernel and other capabilities of the base OS.
- **Engine** Also referred to as the *container runtime*, this is the environment on top of which a container is run. A very popular example of a container runtime is Docker Engine. This isn't to say it's the only container runtime, but it is arguably the first container runtime (as we know containers today) and it is the most well-known.
- **Orchestration and scheduling controller** Container orchestration deals with managing the lifecycle of containers. Orchestration deals with items such as provisioning and deployment of containers, scaling, movement of containers, and container health monitoring. When a container needs to be deployed, the orchestration tool schedules the deployment and identifies an appropriate system to run the container on. It knows how to deploy and manage containers based on a configuration file that tells the orchestration software where to find the container image (repository) and configuration items such as networking, mounting of storage space, and where to store container logs. Examples of container orchestration and scheduling tools include Kubernetes and Docker Swarm.
- **Image repository** This is where all of the images and code that can be deployed as containers are stored. Docker Hub is a popular example of a container image repository. Image repositories can be public or private.

EXAM TIP For image repository, I'm using the naming used in the CSA Guidance, but you should know about two related concepts—image registries and image repositories. An *image registry* is used to host and distribute images. An *image repository* is technically different, because it is defined as a collection of related images. Long story short, this means that an image registry can contain multiple repositories. You'll often see these terms used interchangeably. Your CCSK exam will use the term "image repository."

Keeping all of these elements in mind, I hope you can appreciate how there may be some proprietary dependencies in place that make portability a bit more difficult than you may have expected. For example, what about moving a container from Windows to Linux runtimes (and vice versa)? What if you presently use Kubernetes as the orchestration and scheduling service and then decide to use a cloud provider's orchestration service

instead? Are the runtimes backward-compatible? As I said, a container can help with portability, but it isn't a guaranteed magic bullet for portability.

Container Definitions Backgrounder

As you know, containers are built and managed according to a definition file you create. The definition file is passed to a service (daemon) to build an image so it is properly allocated resources and other configuration settings are implemented.

NOTE Container definition files are not in the CSA Guidance and therefore will not be covered in your CCSK exam. They are covered here to give you a better understanding of how containers are configured and managed.

Following is a list of some of the available options in configuration files used by Amazon Elastic Container Service (Amazon ECS) to build and manage containers:

- **Name** The name of the container
- **Image** The name of the image in the repository that should be used to build the container
- **Memory** The amount of memory to be allocated to the container
- **Port mappings** The required network ports for the container
- **Protocol** The required network protocol (TCP or UDP)
- **Health checks** Monitors the health of the container; if the container is unreachable, it is removed and replaced
- **CPU** The required CPU capacity of the container
- **Working directory** The directory in the container where commands are run
- **Secrets** Credential storage location outside of the container
- **DNS servers** A list of DNS servers for the container to use
- **Mount points** Supplies data volumes
- **Log configuration** Where the container should store logs
- **User** The username to use in the container; the "privileged" user can run everything as root (administrator)

Now you should have a pretty good idea of how containers are built and how the orchestration and scheduling service launches and maintains containers. Again, you won't be tested on any of these items.

Container Security Recommendations

As I mentioned, container technology is maturing, but many products out there have their own security requirements. Following is a list of security recommendations from

the CSA as to general security best practices that you should consider when deploying container technology internally or within a cloud environment:

- **Securing the underlying infrastructure** Security always begins in the container, and in a cloud environment, this is the provider's responsibility. Just as the provider is responsible for security of the physical infrastructure and the hypervisors in a virtual machine world, the provider is responsible for the physical infrastructure and the container platform hosting consumer containers.
- **Securing the orchestration and scheduling service** You know that orchestration and scheduling are critical components of container deployments and management. CSA Guidance refers to this as the "management plane" for containers.
- **Securing the image repository** The image repository for containers can be considered in the same way as images for virtual machines. Images need to be stored in a secure location, and appropriate access controls should be configured to ensure that only approved access is granted to modify images or configuration files.
- **Securing the tasks/code in the container** Containers hold software code. Weak application security will be weak regardless of whether it is run in a container or on a VM. Weak security isn't limited to the code in the container; it can also apply to the definition files you read about in the "Container Definitions Backgrounder." Appropriate network ports, file storage, secrets, and other settings can increase security of the container environment and therefore the application as a whole.

NOTE These are general best practices. Always consult vendor documentation for the latest product-dependent security recommendations. Check out the Cloud Security Alliance web site for more in-depth container security recommendations, such as the "Best Practices for Implementing a Secure Application Container Architecture." Also, the Center for Internet Security provides industry recommendations on securing specific products such as Docker and Kubernetes.

A final takeaway for security of a container environment is that tools will offer varying degrees of security. At a bare minimum, all products should have strong access controls and authentication capabilities. They should also support secure configurations that isolate file system, process, and network access.

Chapter Review

This chapter expanded on the coverage related to the virtualization of Compute, Network, and Storage. It also included additional information on the security responsibilities split between the providers and customers. As always, backgrounder information was included

to address any knowledge gaps that you may have, but that information will not be part of your exam.

When preparing for your CCSK exam, make sure you are comfortable with the following items:

- Cloud providers must make strong isolation of workloads their primary duty.
- Providers are responsible for all physical security and any virtualization technologies that customers use. They must keep hypervisors secured and implement any required security patches.
- Providers must implement all customer-managed virtualization features with a "secure-by-default" (aka deny-by-default) configuration.
- Providers must ensure that any volatile memory is properly secured to prevent unintended access by other tenants or administrators.
- Providers must implement strong networking controls to protect customers at the physical level as well as virtual networking the customers cannot control.
- Providers must isolate virtual network traffic, even when networks are controlled by the same customer.
- Providers must secure the physical storage system in use. This can include encryption at the physical layer to prevent data exposure during drive replacements.
- Consumers always need to know what security is offered by the provider and what they need to do in order to meet their own security requirements.
- For container security, remember that all the various components (engine, orchestration, and repository) need to be properly secured.
- Containers offer application isolation, but not complete isolation. Containers with similar security requirements should be grouped together and run on the same physical or virtual host to provide greater security segregation.
- Proper access controls and strong authentication should be in place for all container components.
- Ensure that only approved, known, and secure container images or code can be deployed.

Questions

1. Why must the provider encrypt hard drives at the physical layer?

A. It prevents data from being compromised as a result of theft.

B. It prevents data from being accessed by others via the virtual layer.

C. It prevents data from being compromised after the drive is replaced.

D. Answers A and C are correct.

2. How do containers perform isolation?
   - A. They perform application layer isolation.
   - B. They perform isolation at all layers like a virtual machine does.
   - C. They perform isolation of the repository.
   - D. All of the above are correct.
3. Which of the following is the number one security priority for a cloud service provider?
   - A. Implementing SDN firewalls for customers
   - B. Isolating tenant access to pools of resources
   - C. Securing the network perimeter
   - D. Offering network monitoring capability to customers
4. Which of the following are examples of compute virtualization?
   - A. Containers
   - B. Cloud overlay networks
   - C. Software templates
   - D. A and C
5. Nathan is trying to troubleshoot an issue with a packet capture tool on a running instance. He notices clear-text FTP usernames and passwords in the captured network traffic that is intended for another tenant's machine. What should Nathan do?
   - A. This is normal behavior in a cloud. He should contact the other tenant and advise them that using clear-text credentials in a cloud is a bad idea.
   - B. Nathan should contact the other tenant and submit his finding for a bug bounty.
   - C. This is not possible because FTP is prohibited in a cloud environment.
   - D. He should contact the provider and advise them that he will be canceling his use of their cloud services because the provider has failed to isolate the network.
6. What is/are benefits of a virtual network compared to physical networks?
   - A. You can compartmentalize application stacks in their own isolated virtual networks, which increases security.
   - B. An entire virtual network can be managed from a single management plane.
   - C. Network filtering in a physical network is easier.
   - D. All of the above are true.

7. How is a storage pool created?
   - **A.** The provider uses direct storage with a bunch of hard drives attached to a server.
   - **B.** The provider uses a storage area network.
   - **C.** The provider uses a NAS.
   - **D.** The provider builds the storage pool however they want.
8. A provider wants to ensure that customer data is not lost in the event of drive failure. What should the provider do?
   - **A.** Use a SAN and copy the data across multiple drives in a storage controller.
   - **B.** Replicate the data to an offshore third party.
   - **C.** Make multiple copies of the data and store the copies on multiple storage locations.
   - **D.** Store client data using solid state drives (SSDs).
9. Why is volatile memory a security concern for providers?
   - **A.** It isn't. Volatile memory protection is the customer's responsibility.
   - **B.** Volatile memory may contain unencrypted information.
   - **C.** Volatile memory may contain credentials.
   - **D.** B and C are correct.
10. Which of the following components in a container environment require access control and strong authentication?
    - **A.** Container runtime
    - **B.** Orchestration and scheduling system
    - **C.** Image repository
    - **D.** All of the above

Answers

1. **D.** Answers A and C are correct. Providers encrypt hard drives so that the data cannot be read if the drive is stolen or after it is replaced. Encryption at the physical layer does not protect data that is requested via the virtual layer.
2. **A.** Containers perform isolation only at the application layer. This is unlike a virtual machine that can offer isolation for all layers. Repositories require appropriate controls to be put in place to restrict unauthorized access to the code and configuration files held within.
3. **B.** The top priority for providers is ensuring that they implement strong isolation capabilities. All of the other answers are possible priorities, but B is the best answer.

4. **A.** Of the list presented, only containers can be considered as compute virtualization. Software templates are used to build an entire environment quickly. Although you could use these templates in infrastructure as code (IaC) to build or deploy containers and VMs, this is not considered a compute virtualization. A cloud overlay network enables a virtual network to span multiple physical networks.
5. **D.** Nathan is able to see network traffic destined for other machines, so there has been a failure of network isolation, and this should be the provider's top security priority. If I were Nathan, I would change cloud providers as soon as possible. All the other answers are not applicable (although writing a bunch of screen captures to the other tenant's FTP directory to advise them of their exposure would be pretty funny).
6. **A.** The only accurate answer listed is that virtual networks can be compartmentalized, and this can increase security; this is expensive, if not impossible, in a physical network. SDN can offer a single management plane for physical network appliances, and the "ease" of filtering is quite subjective. Filtering in a virtual network is different, but it may or may not be more difficult.
7. **D.** It is completely up to the provider as to how they build a storage pool. They can use any of the other technologies listed in the answers, or they can use something completely different and proprietary.
8. **C.** To offer increased resiliency, the provider should make multiple copies of customer data and store copies across multiple storage locations. Answer A looks good, but it's not the best answer, because a SAN is not required and, more importantly, writing data to multiple drives in the same controller will not protect against the single point of failure in the controller (or the controller corrupting the data). Finally, we haven't discussed the difference between "normal" magnetic storage drives versus solid state drives, but SSDs can fail just like magnetic ones, so D isn't the best answer either.
9. **D.** The correct answer is that volatile memory can contain sensitive information such as credentials and data that needs to be unencrypted in order to be processed. Both the provider and the customer play a role in ensuring security related to volatile memory. The provider needs to ensure that volatile memory from one tenant is never seen by another tenant (an even better way to think of it is that one workload shouldn't have access to another workload). The customer needs to make sure that volatile memory is wiped from a system prior to it being imaged. This can be achieved by rebooting the instance prior to creating the image.
10. **D.** Yes, all of the above is the right choice this time. But wait! There's a good story here that I'm including for those of you still with me. In February 2018, Tesla (the car company) was breached. Thankfully for Tesla, the attackers only wanted

to use Tesla cloud resources for bitcoin mining. How was Tesla breached? Was it a zero-day attack? Was it advanced state-sponsored agents? Nope! Its container orchestration software (Kubernetes in this case) was accessible from the Internet and didn't require a password to access it! Not only did this give the attackers the ability to launch their own containers, paid for courtesy of Tesla, but inside the Kubernetes system was a secrets area that had Amazon S3 keys stored in it. The keys were used to access nonpublic information from Tesla. Again, container security involves much more than just application security within a container.