EC-Council

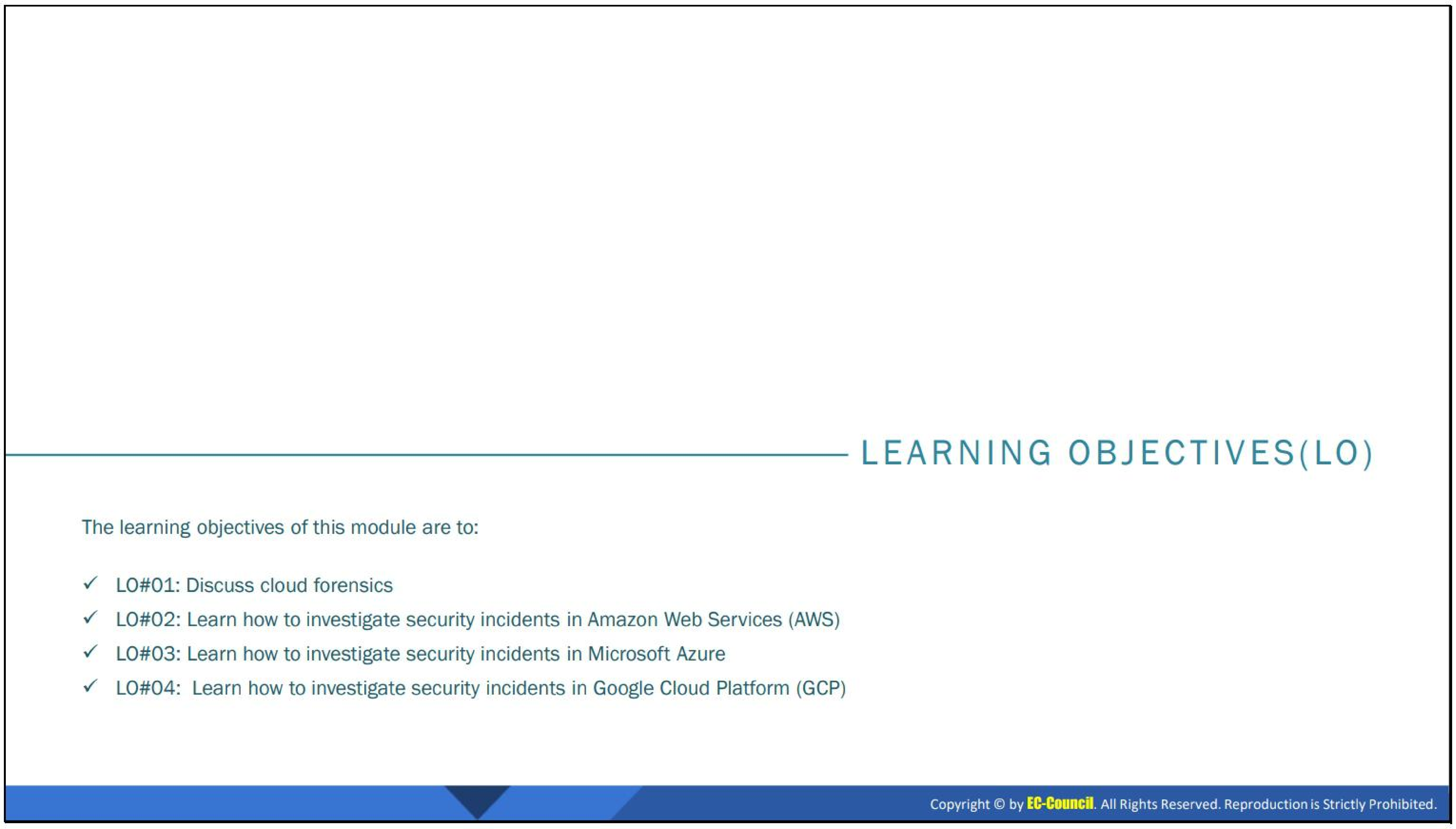
C|CSE™

Certified Cloud Security Engineer

Forensic Investigation in Cloud

Module 08

This page is intentionally left blank.



LEARNING OBJECTIVES(LO)

The learning objectives of this module are to:

- ✓ LO#01: Discuss cloud forensics
- ✓ LO#02: Learn how to investigate security incidents in Amazon Web Services (AWS)
- ✓ LO#03: Learn how to investigate security incidents in Microsoft Azure
- ✓ LO#04: Learn how to investigate security incidents in Google Cloud Platform (GCP)

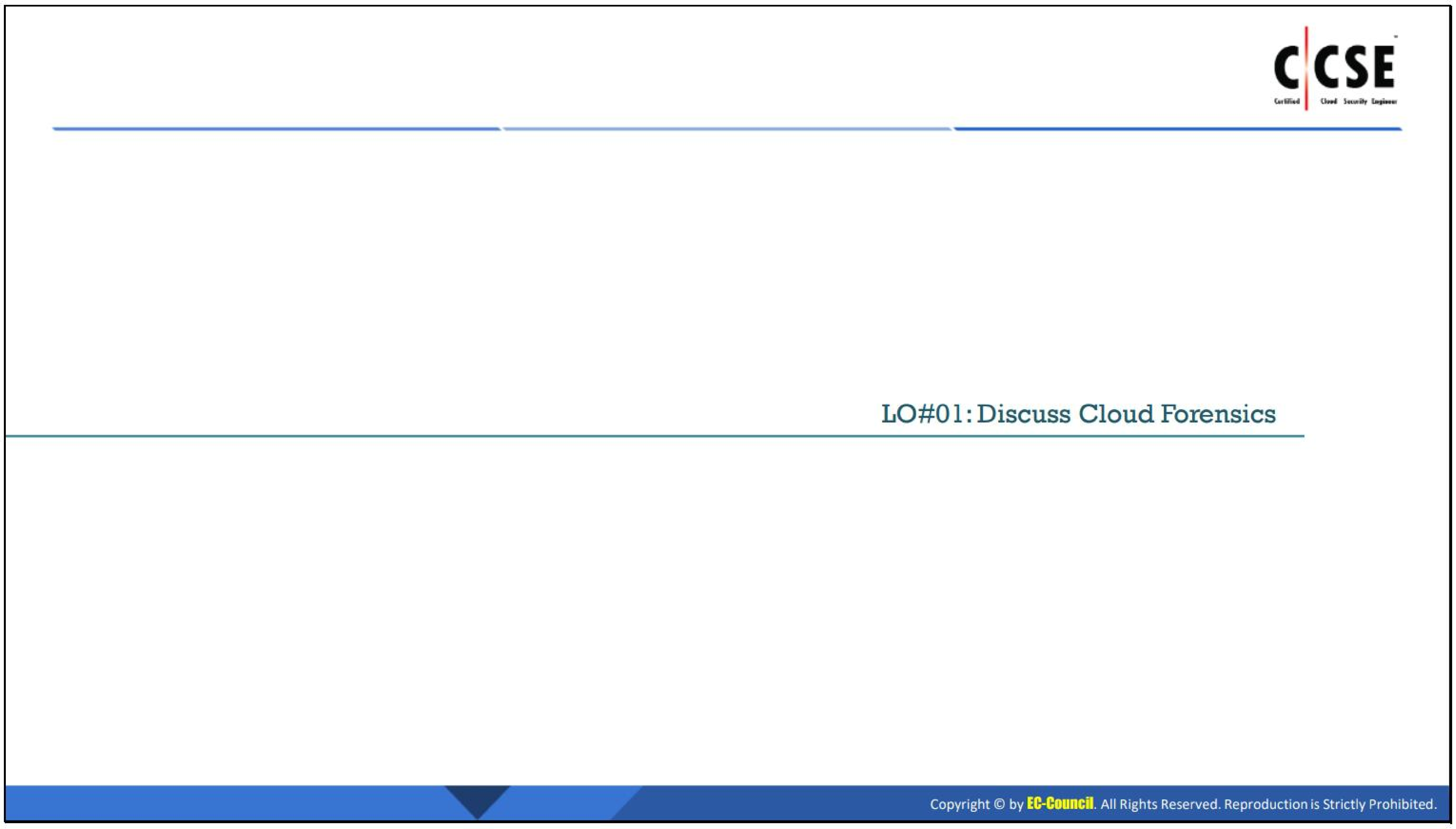
Learning Objectives

Cloud computing is an emerging technology that delivers computing services such as online business applications, data storage facilities, databases, and virtualized OSes. Cloud implementation enables a distributed workforce, reduces organizational expenses, and provides data security. As many enterprises are adopting the cloud, attackers make the cloud as their target of exploit to gain unauthorized access to the valuable data stored in it. Therefore, a there is a need of having a proactive incident response plan in the cloud environment that can facilitate faster forensic investigation process and effective risk mitigation.

This module discusses the basic cloud computing concepts such as types of cloud deployment models and cloud computing threats. It introduces cloud forensics and elaborates on various cloud forensic challenges. This module specifically focuses on two top cloud service providers, namely, Amazon Web Services (AWS) and Microsoft Azure, and demonstrates specific processes of performing forensic acquisition and analysis on their cloud environments.

At the end of this module, you will be able to:

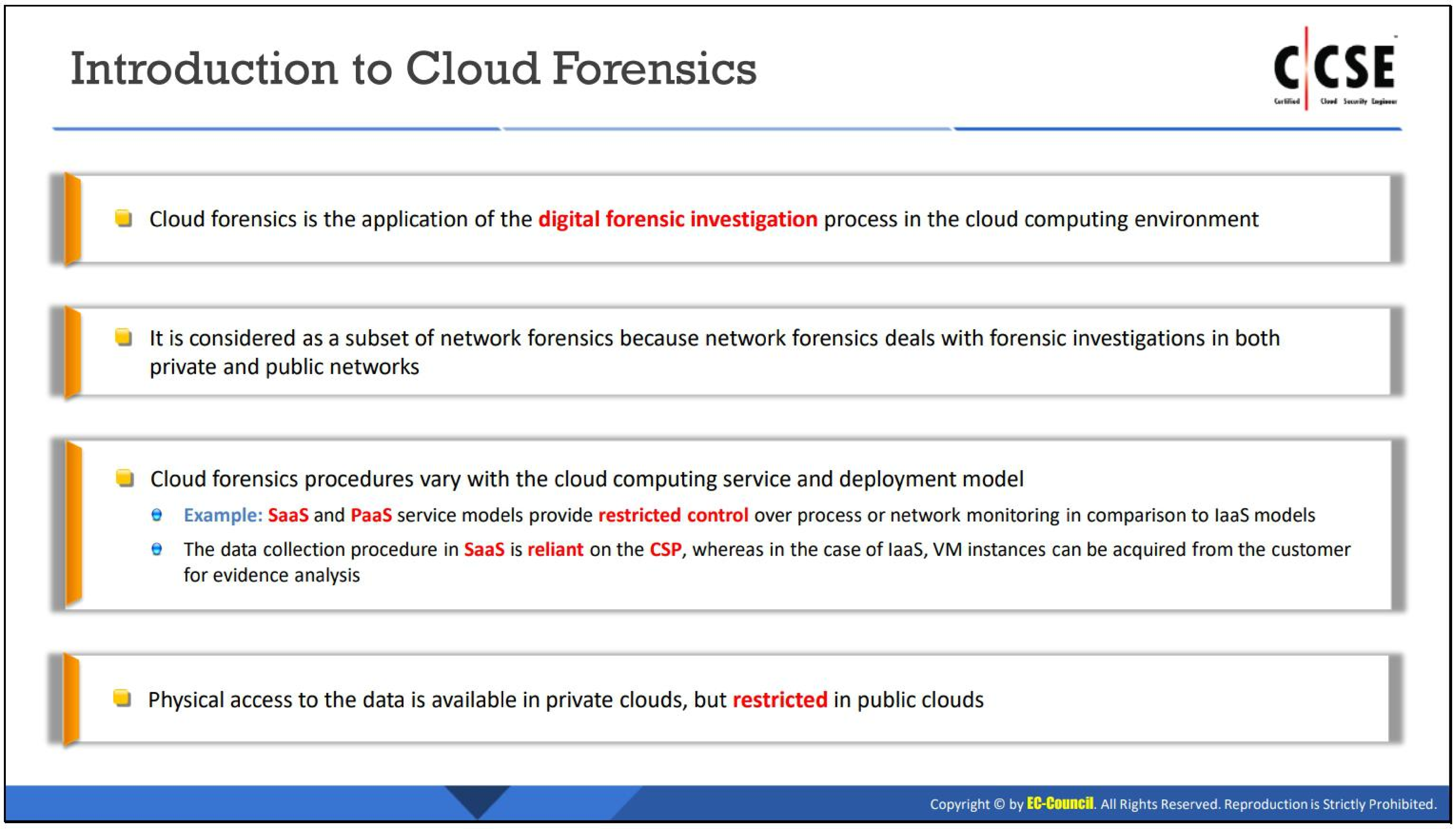
- Understand cloud forensics
- Determine how to investigate security incidents in AWS
- Determine how to investigate security incidents in Azure
- Determine how to investigate security incidents in GCP



LO#01: Discuss Cloud Forensics

LO#01: Discuss Cloud Forensics

Cloud forensics seeks to apply the principles and methods of digital forensics within the cloud environment to investigate any security incident. To obtain digital evidence in a cloud environment, investigators must know the data location and the access level exercised by a specific organization on that data. This section discusses cloud forensics, its importance and usage, the role of stakeholders during investigation, and the multiple challenges encountered by investigators while performing forensics in the cloud.



Introduction to Cloud Forensics

- Cloud forensics is the application of the **digital forensic investigation** process in the cloud computing environment

- It is considered as a subset of network forensics because network forensics deals with forensic investigations in both private and public networks

- Cloud forensics procedures vary with the cloud computing service and deployment model
  - **Example:** **SaaS** and **PaaS** service models provide **restricted control** over process or network monitoring in comparison to IaaS models
  - The data collection procedure in **SaaS** is **reliant** on the **CSP**, whereas in the case of IaaS, VM instances can be acquired from the customer for evidence analysis

- Physical access to the data is available in private clouds, but **restricted** in public clouds

Introduction to Cloud Forensics

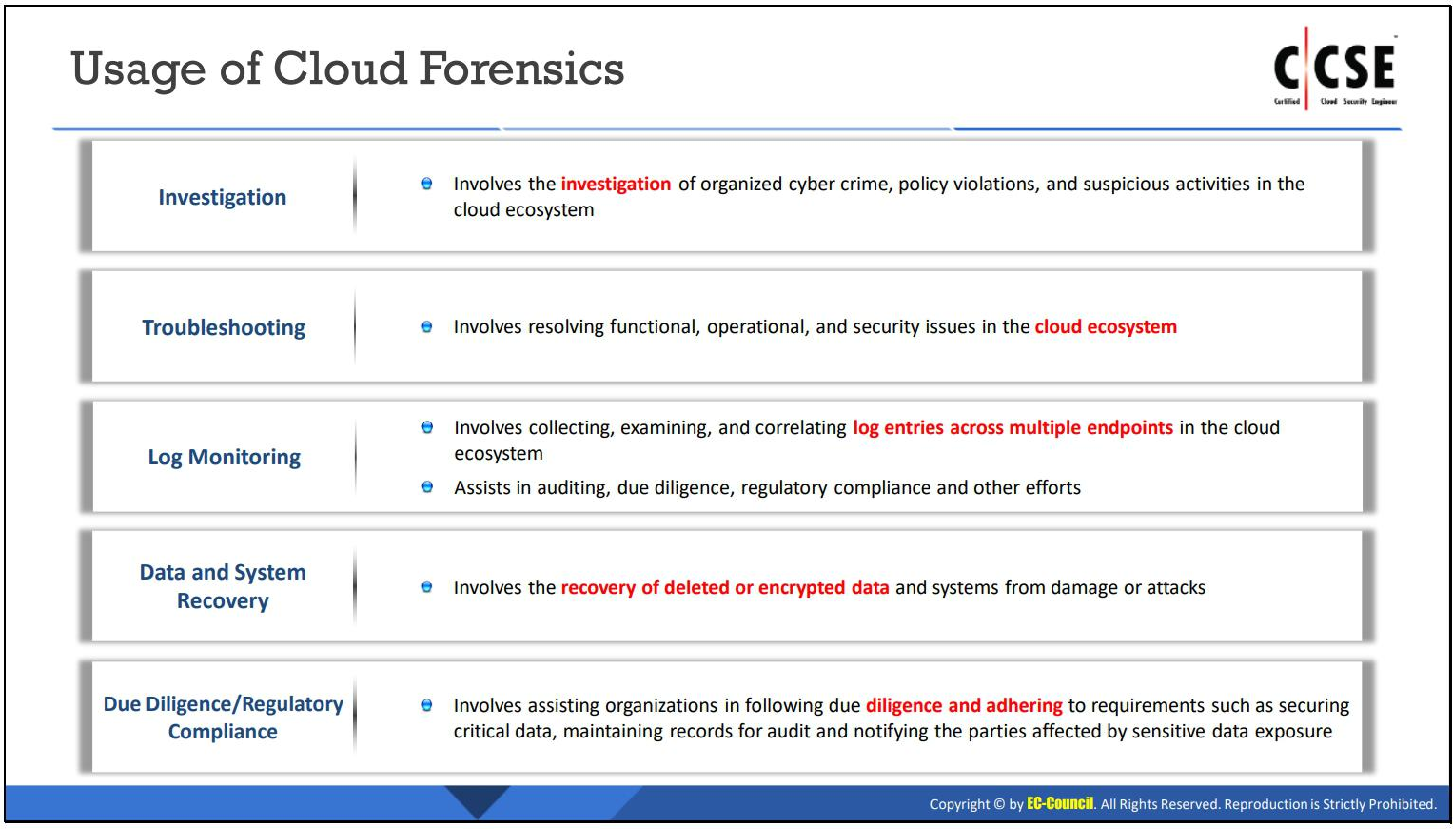
Cloud forensics is the application of digital forensic investigation in a cloud environment and a division of network forensics; it involves the management of public and private networks.

According to the NIST, "Digital Forensics is the application of science to the identification, examination, collection, and analysis of data while preserving the information and maintaining a strict chain of custody for the data."

Cloud computing is spread across a large network and has customized principles. Therefore, the forensic procedures in cloud computing differ according to the service provided and the deployment model.

The initial phases of evidence collection vary for each model. In the SaaS model, the investigators completely depend on the CSP for collecting application logs. However, in IaaS, the investigators can acquire the virtual disk of the OS from the clients and initiate the forensics examination and analysis process.

Similarly, the cloud forensic examiners can have physical access to the digital evidence in private cloud services because they provide more control to the customer over the data and hardware infrastructure. However, it is difficult to gain physical access to the data in SaaS models over public cloud, and the investigation depends on the audit reports and log data provided by the CSP.



Usage of Cloud Forensics

Investigation	Involves the **investigation** of organized cyber crime, policy violations, and suspicious activities in the cloud ecosystem
Troubleshooting	Involves resolving functional, operational, and security issues in the **cloud ecosystem**
Log Monitoring	Involves collecting, examining, and correlating **log entries across multiple endpoints** in the cloud ecosystem Assists in auditing, due diligence, regulatory compliance and other efforts
Data and System Recovery	Involves the **recovery of deleted or encrypted data** and systems from damage or attacks
Due Diligence/Regulatory Compliance	Involves assisting organizations in following due **diligence and adhering** to requirements such as securing critical data, maintaining records for audit and notifying the parties affected by sensitive data exposure

Usage of Cloud Forensics

The cloud technology enables users to conveniently access the configurable computing resources (such as servers, applications, and services) on demand. Therefore, attackers target the cloud to gain unauthorized access to this private information. Cloud forensic techniques help forensic practitioners and everyday users in handling and protecting themselves from such security incidents.

Cloud forensics has many uses as listed below:

- **Investigation**

  Cloud forensics helps in finding the source of different cloud-based crimes and solving organized cloud crimes, policy violations in a public environment, and suspicious activities in a cloud environment. In the investigation process, all sources, including manual and mechanical, are analyzed and the results are revealed. This helps the clients and service providers to secure their cloud services.

- **Troubleshooting**

  Cloud forensic techniques assist users in troubleshooting by determining the data and hosts that are physically and virtually present in a cloud environment. They allow users to find and resolve any errors or security issues in the cloud. They help in understanding the trends of past security attacks to tackle any incident in the future.

- **Log Monitoring**

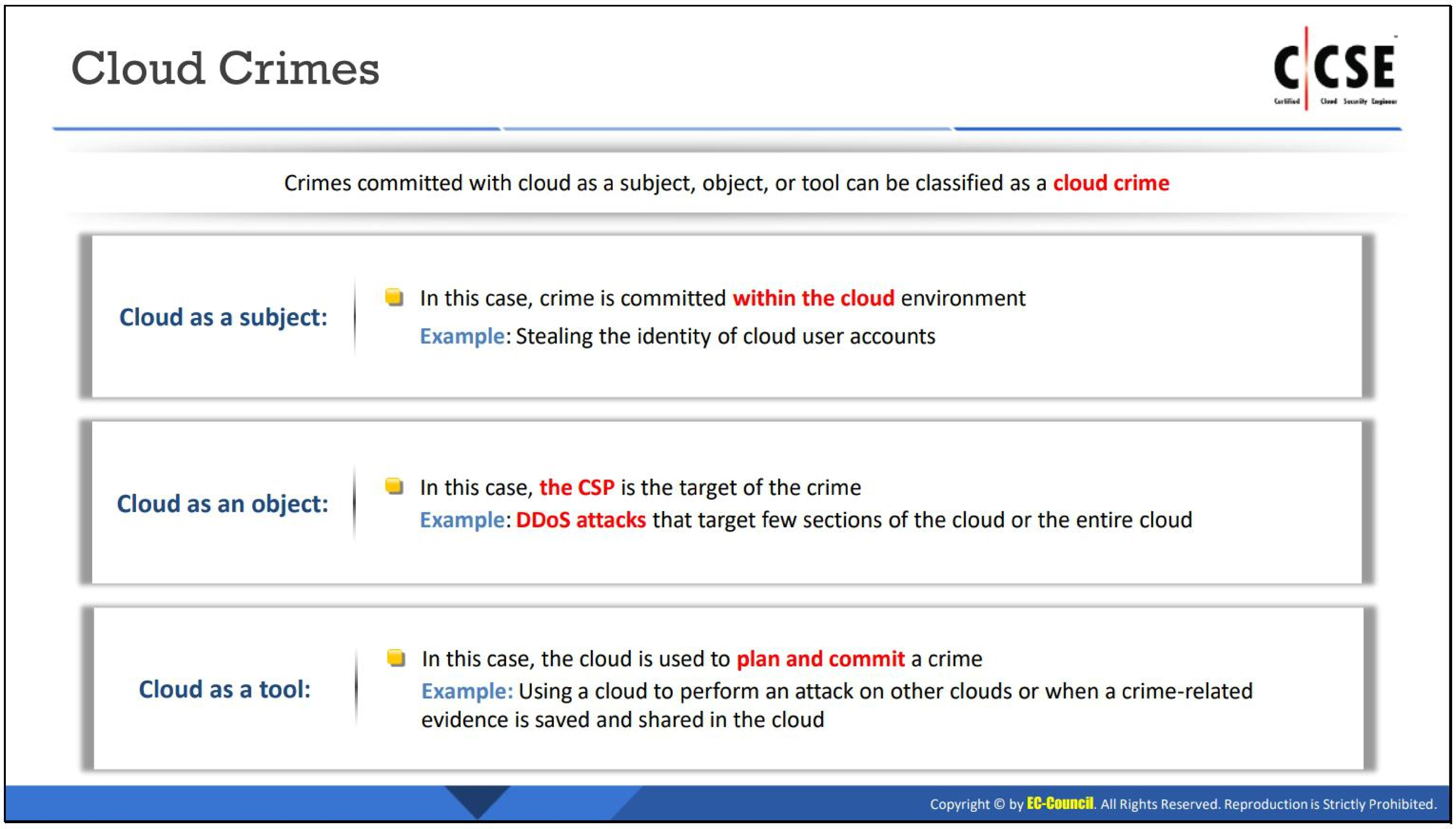
  Cloud forensic techniques include processes for generating, storing, analyzing, and correlating the massive volumes of log data created within a cloud environment. These data help the users and service providers to audit, analyze, and calculate various aspects of the cloud environment; they also help security officials in checking whether a cloud system complies with the regulatory standards.

- **Data and System Recovery**

  Cloud forensics involves recovery procedures that help forensic practitioners in recovering lost, accidentally deleted, corrupted, and inaccessible data. It also enables the data acquisition of cloud systems and the creation of a forensic copy of the data that can be used by the service providers as back up; forensic experts can use this copy as evidence in the court of law.

- **Due Diligence/Regulatory Compliance**

  Cloud forensics also deals with the security aspects of an organization in securing critical data, maintaining necessary records for auditing purposes, and notifying the concerned team when any suspicious activity is reported; for instance, if private data have been misused or exposed. It also helps in finding the sections that miss a regulatory compliance and fixes them.



Cloud Crimes

Crimes committed with cloud as a subject, object, or tool can be classified as a **cloud crime**

Cloud as a subject:

- In this case, crime is committed **within the cloud** environment
  Example: Stealing the identity of cloud user accounts

Cloud as an object:

- In this case, **the CSP** is the target of the crime
  Example: **DDoS attacks** that target few sections of the cloud or the entire cloud

Cloud as a tool:

- In this case, the cloud is used to **plan and commit** a crime
  Example: Using a cloud to perform an attack on other clouds or when a crime-related evidence is saved and shared in the cloud

Cloud Crimes

Any criminal activity that involves a cloud environment used as a subject, object, or tool can be considered as a cloud crime.

- **Cloud as a Subject**

  It refers to a crime in which the attackers attempt to compromise the security of a cloud environment to steal data or inject malware.

  Example: Stealing the identity of a cloud user, unauthorized modification or deletion of stored data, and installation of malware on the cloud.

- **Cloud as an Object**

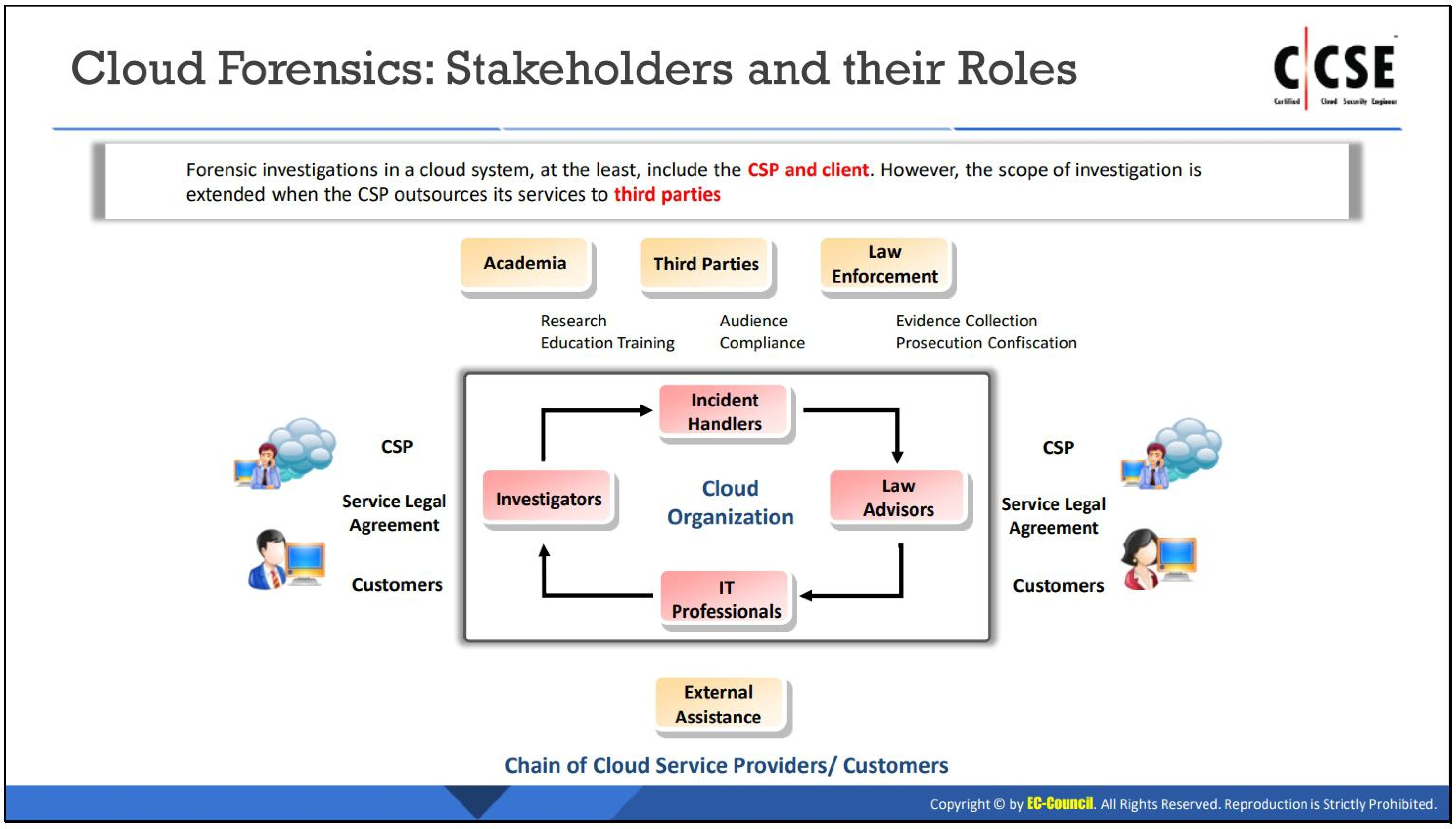
  In this type of crime, attackers use a cloud system to commit a crime against the CSP; here, the cloud behaves like an object. In this case, the main objective of the attacker is to impact the CSP instead of the cloud environment.

  Example: DDoS attacks over the cloud that can cause failure in the entire cloud environment.

- **Cloud as a Tool**

  The cloud becomes a tool when the attacker uses a compromised cloud account to attack other accounts. In such cases, both the source and target cloud can store the evidence data.

  Example: Using a cloud to perform an attack on other clouds or when a crime-related evidence is saved and shared in the cloud.



Cloud Forensics: Stakeholders and their Roles

Forensic investigations in a cloud system, at the least, include the **CSP and client**. However, the scope of investigation is extended when the CSP outsources its services to **third parties**

Chain of Cloud Service Providers/ Customers

Cloud Forensics: Stakeholders and their Roles

A cloud forensic activity involves several stakeholders, including government members, industry partners, third parties, and law enforcement. Investigators should be able to understand the roles and responsibilities of each stakeholder for effective investigation.

This will also help the investigators in determining the technical, legal, and organizational stakeholders to accordingly allocate and document their interests and generate reports. Additionally, it will facilitate the management of different tasks in the cloud and the responsibilities involved when signing the contract.

To enable the forensic capabilities on a cloud, a proper internal structure should be established involving the CSPs and customers. A defined collaboration between the CSP and the customers should be created along with external assistance from the following roles:

- **IT Professionals**

  This team includes professionals responsible for managing and maintaining all aspects of a cloud, for example, cloud security architects, network administrators, security administrators, and ethical hackers.

  IT professionals can provide knowledge about cloud operations, assist the investigators, and help in data collection. They may also be answerable in the case of internal attacks.

- **Investigators**

  The investigators in a cloud organization are responsible for conducting forensic examinations against allegations regarding wrongdoings, vulnerabilities, and attacks over the cloud. They should also work in collaboration with the external investigators and law enforcement agencies for forensic investigations on internal assets.

- **Incident Handlers**

  The incident handlers are the first responders for all security incidents on a cloud. They are the first line of defense against cloud security attacks and their primary role is to respond against any type of security incident immediately.

- **Law Advisors**

  The key responsibility of law advisors is to ensure that all forensic activities are within the jurisdiction and they do not violate any regulations or agreements.

- **External Assistance**

  External assistance is utilized when the internal team requires external support in performing any task besides those that have already been performed such as the investigation of civil cases and e-discovery.

  Before taking external assistance, the internal team should be well-aware of the forensic activities that will be performed under the external assistance.

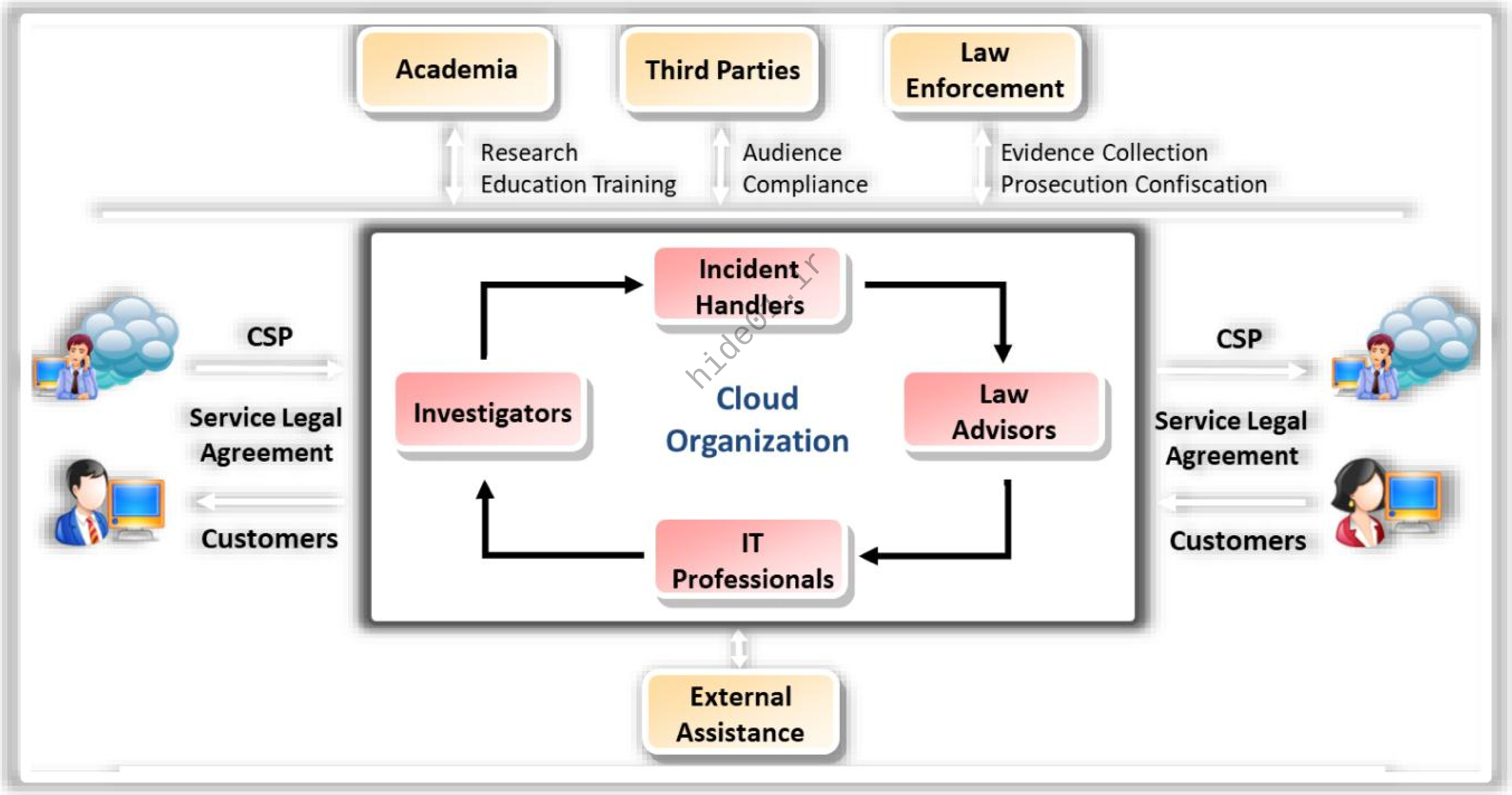
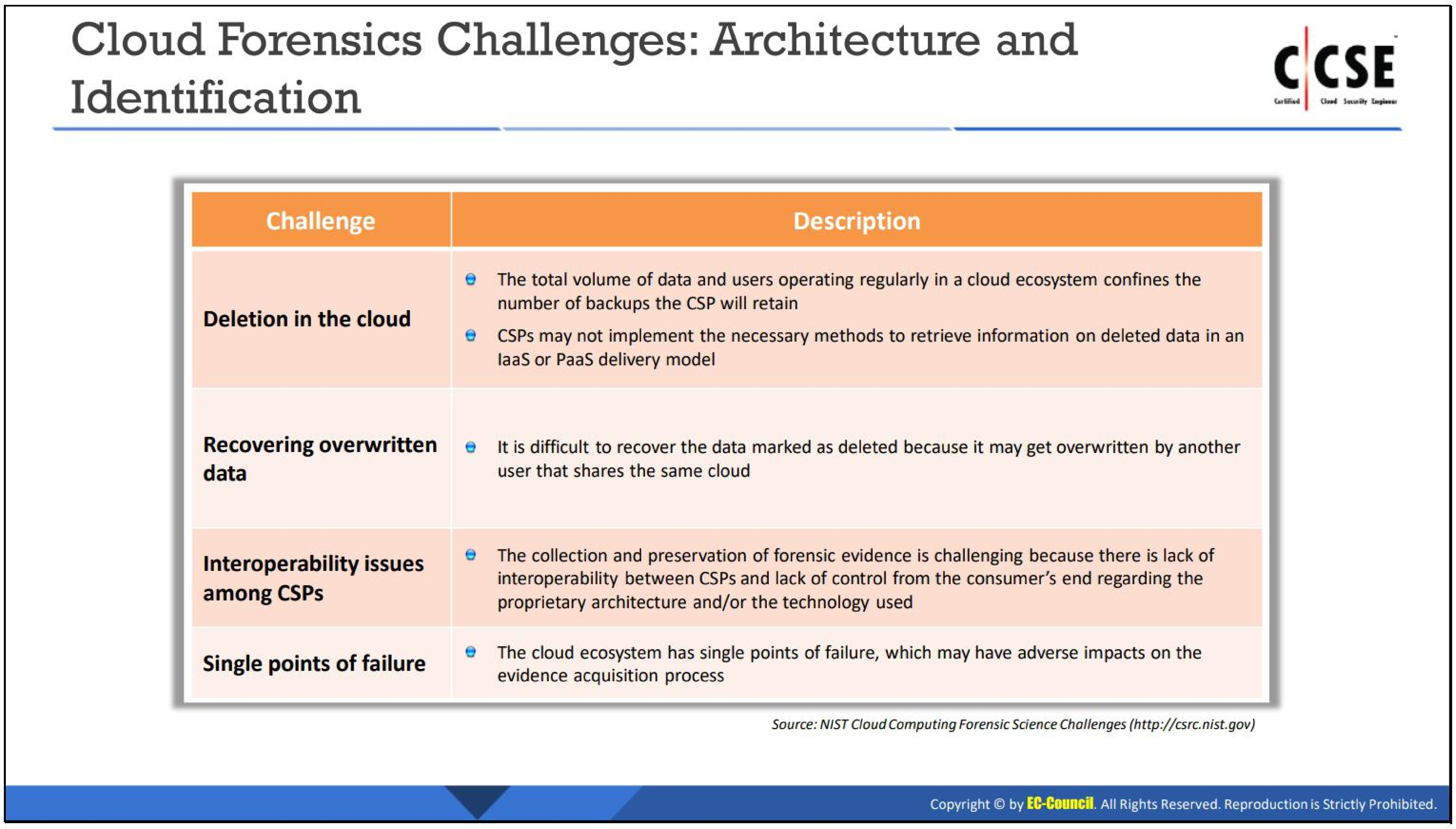


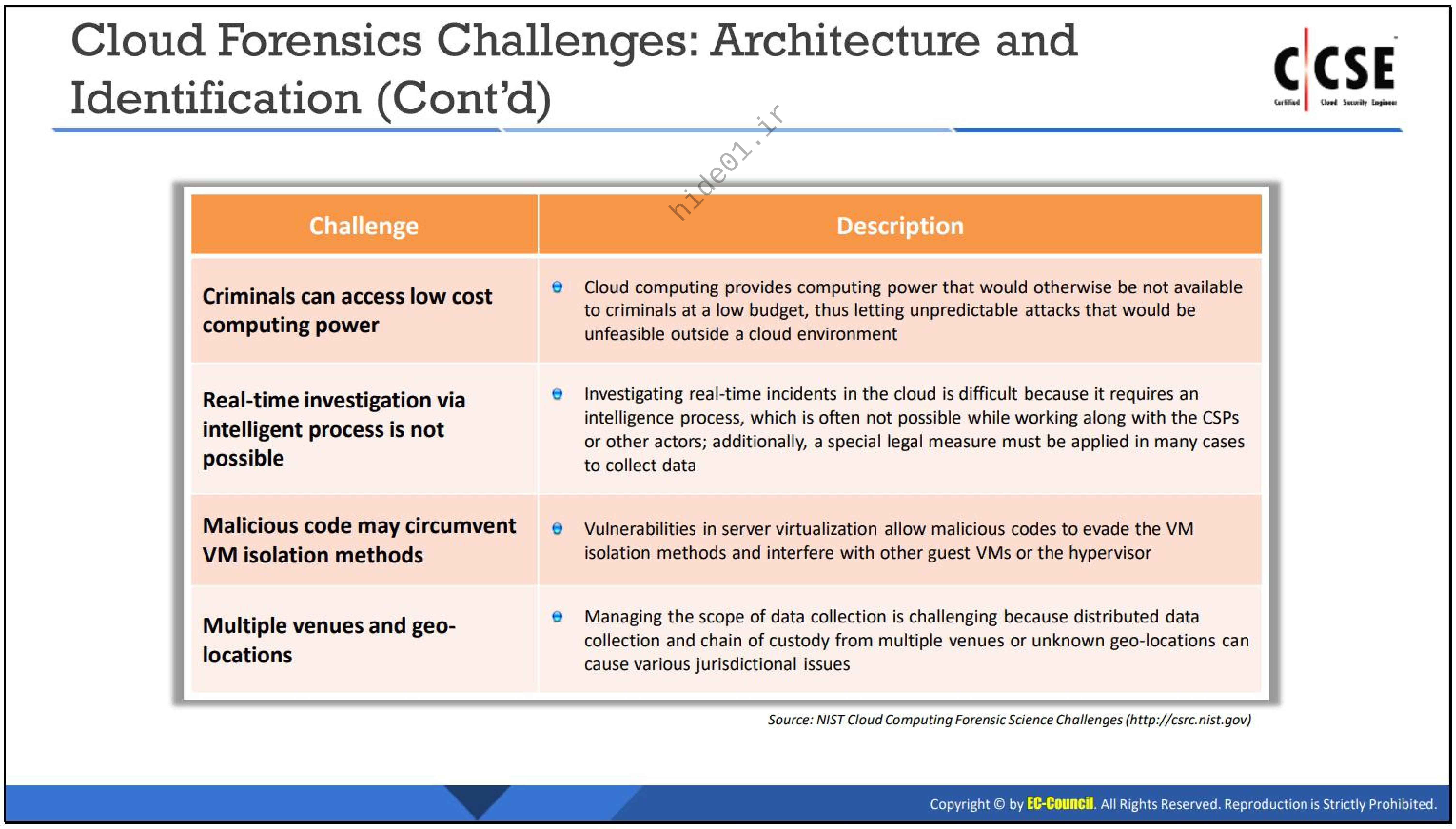
Figure 8.1: Chain of Cloud Service Providers / Customers



Cloud Forensics Challenges: Architecture and Identification

Challenge	Description
Deletion in the cloud	- The total volume of data and users operating regularly in a cloud ecosystem confines the number of backups the CSP will retain - CSPs may not implement the necessary methods to retrieve information on deleted data in an IaaS or PaaS delivery model
Recovering overwritten data	- It is difficult to recover the data marked as deleted because it may get overwritten by another user that shares the same cloud
Interoperability issues among CSPs	- The collection and preservation of forensic evidence is challenging because there is lack of interoperability between CSPs and lack of control from the consumer's end regarding the proprietary architecture and/or the technology used
Single points of failure	- The cloud ecosystem has single points of failure, which may have adverse impacts on the evidence acquisition process

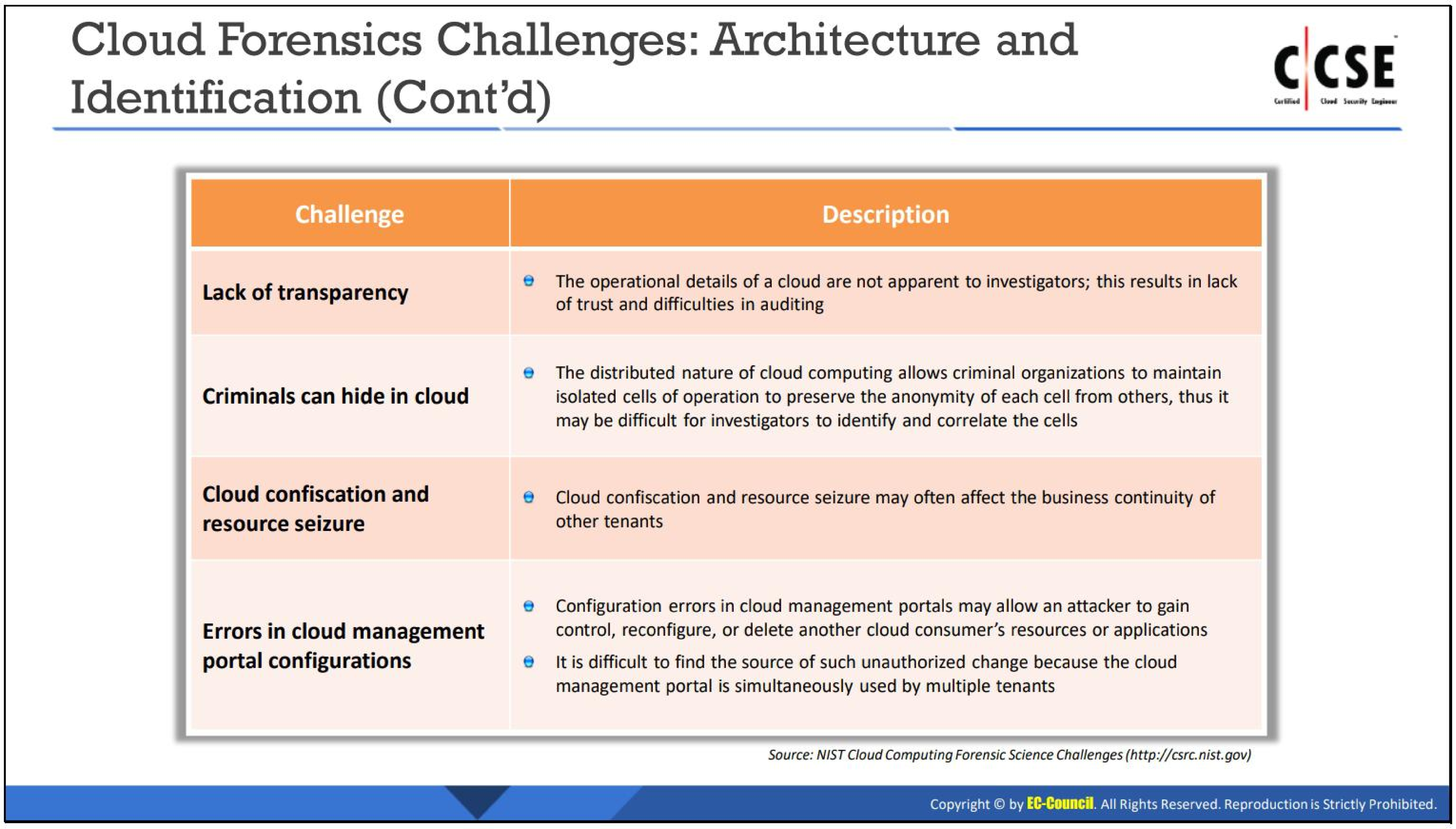
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



Cloud Forensics Challenges: Architecture and Identification (Cont'd)

Challenge	Description
Criminals can access low cost computing power	- Cloud computing provides computing power that would otherwise be not available to criminals at a low budget, thus letting unpredictable attacks that would be unfeasible outside a cloud environment
Real-time investigation via intelligent process is not possible	- Investigating real-time incidents in the cloud is difficult because it requires an intelligence process, which is often not possible while working along with the CSPs or other actors; additionally, a special legal measure must be applied in many cases to collect data
Malicious code may circumvent VM isolation methods	- Vulnerabilities in server virtualization allow malicious codes to evade the VM isolation methods and interfere with other guest VMs or the hypervisor
Multiple venues and geo-locations	- Managing the scope of data collection is challenging because distributed data collection and chain of custody from multiple venues or unknown geo-locations can cause various jurisdictional issues

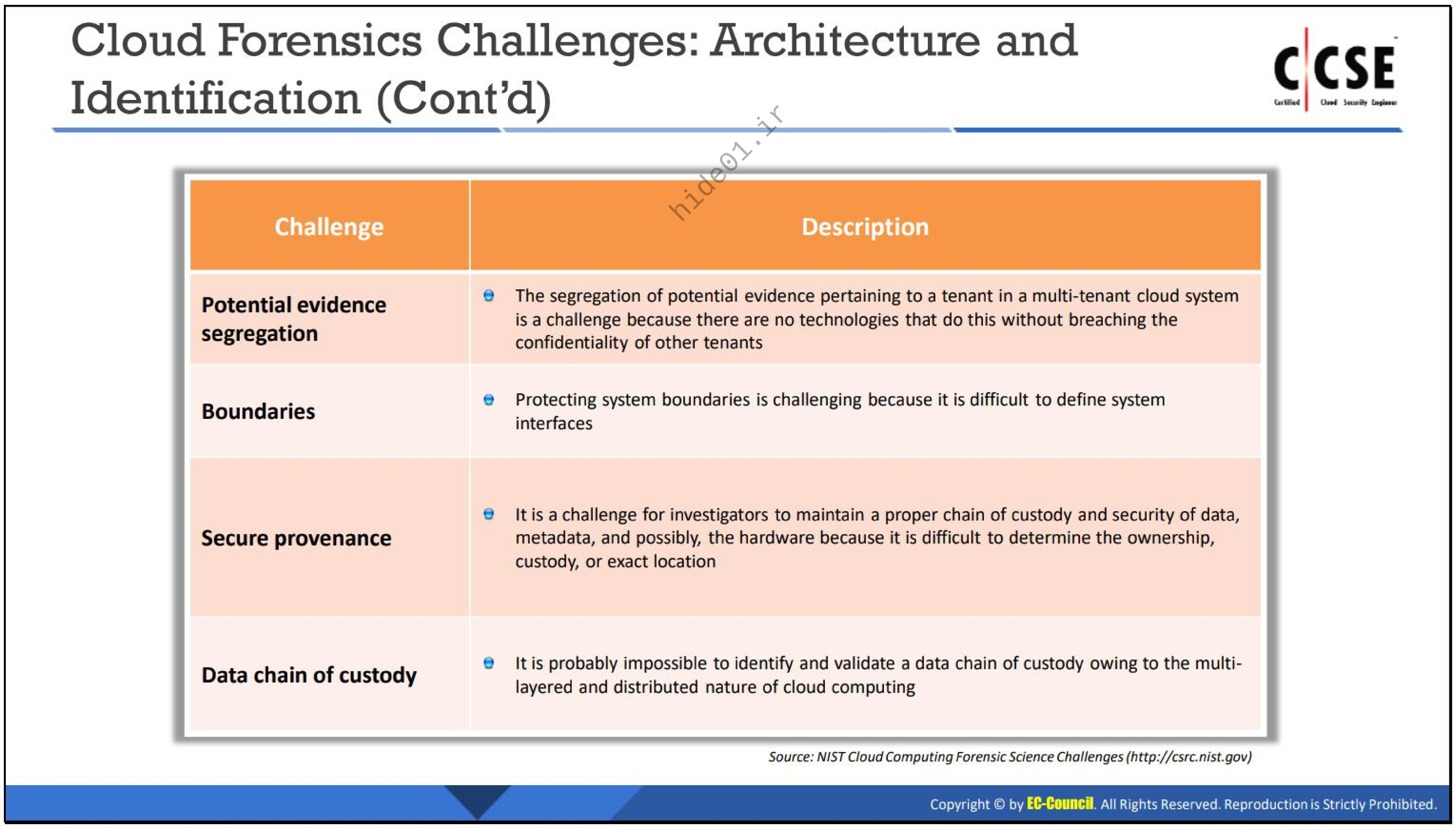
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



Cloud Forensics Challenges: Architecture and Identification (Cont'd)

Challenge	Description
Lack of transparency	The operational details of a cloud are not apparent to investigators; this results in lack of trust and difficulties in auditing
Criminals can hide in cloud	The distributed nature of cloud computing allows criminal organizations to maintain isolated cells of operation to preserve the anonymity of each cell from others, thus it may be difficult for investigators to identify and correlate the cells
Cloud confiscation and resource seizure	Cloud confiscation and resource seizure may often affect the business continuity of other tenants
Errors in cloud management portal configurations	Configuration errors in cloud management portals may allow an attacker to gain control, reconfigure, or delete another cloud consumer's resources or applications It is difficult to find the source of such unauthorized change because the cloud management portal is simultaneously used by multiple tenants

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



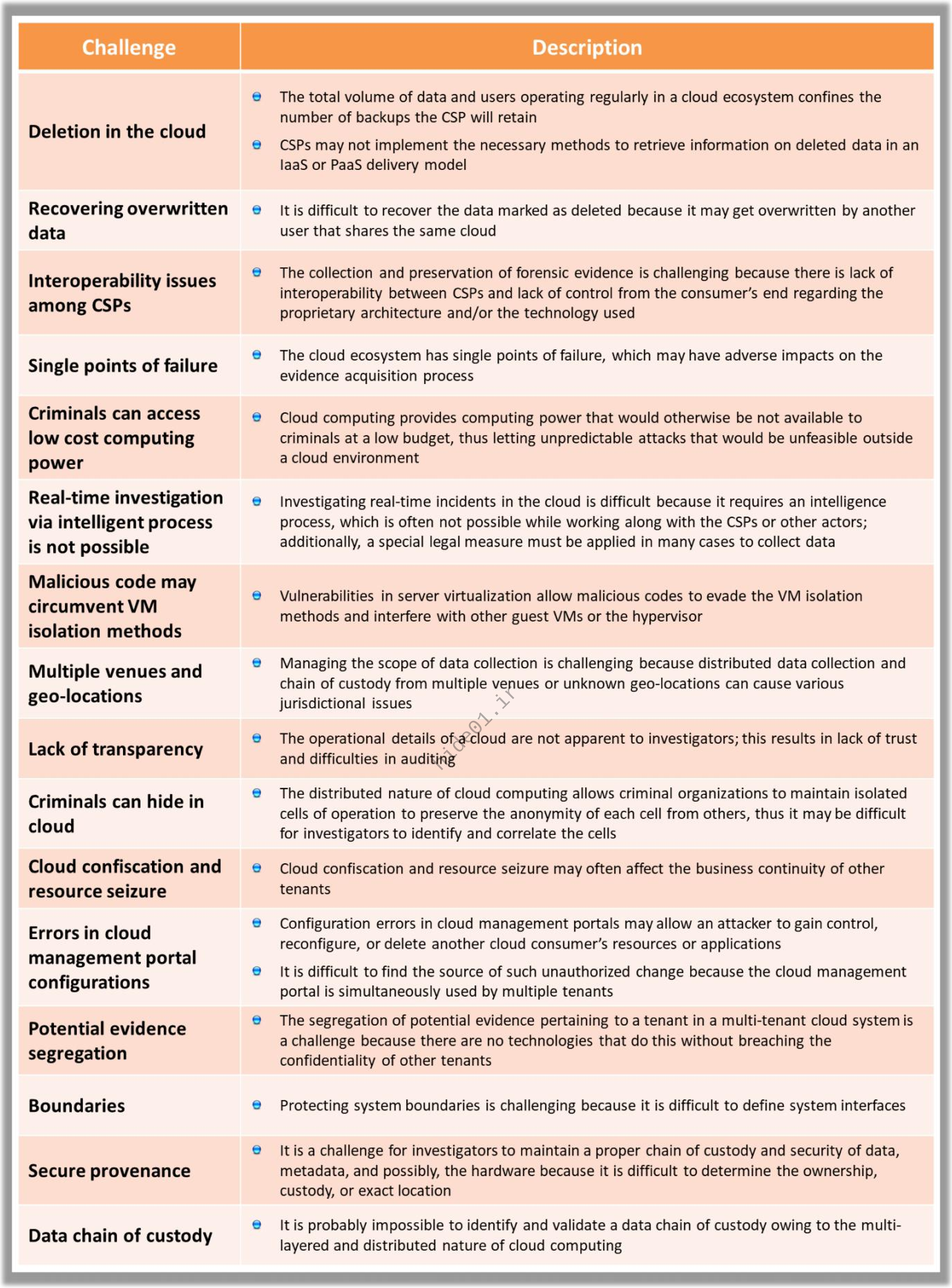
Cloud Forensics Challenges: Architecture and Identification (Cont'd)

Challenge	Description
Potential evidence segregation	The segregation of potential evidence pertaining to a tenant in a multi-tenant cloud system is a challenge because there are no technologies that do this without breaching the confidentiality of other tenants
Boundaries	Protecting system boundaries is challenging because it is difficult to define system interfaces
Secure provenance	It is a challenge for investigators to maintain a proper chain of custody and security of data, metadata, and possibly, the hardware because it is difficult to determine the ownership, custody, or exact location
Data chain of custody	It is probably impossible to identify and validate a data chain of custody owing to the multi-layered and distributed nature of cloud computing

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

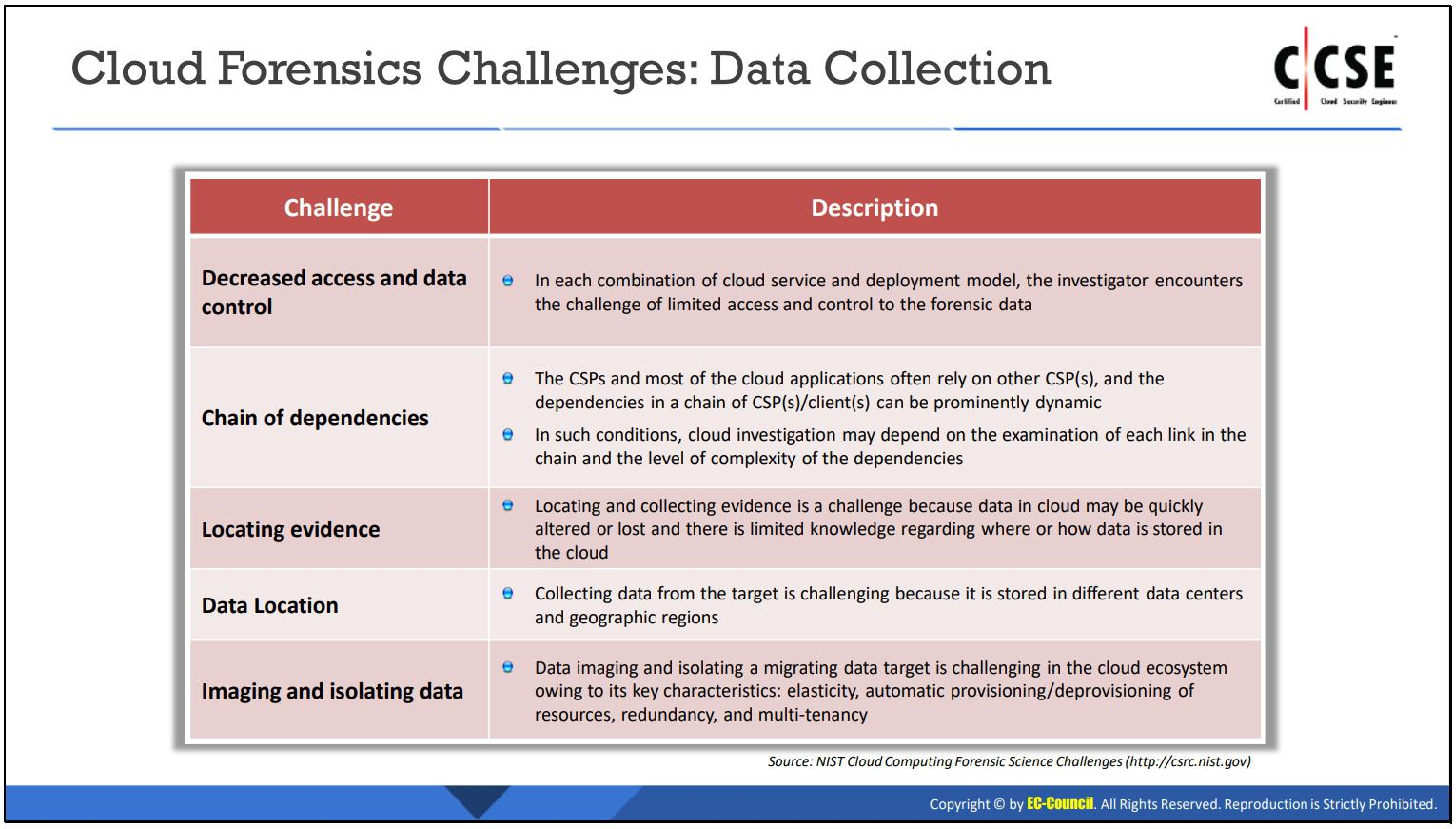
Cloud Forensics Challenges: Architecture and Identification

Source: *NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov*)



Challenge	Description
Deletion in the cloud	The total volume of data and users operating regularly in a cloud ecosystem confines the number of backups the CSP will retain CSPs may not implement the necessary methods to retrieve information on deleted data in an IaaS or PaaS delivery model
Recovering overwritten data	It is difficult to recover the data marked as deleted because it may get overwritten by another user that shares the same cloud
Interoperability issues among CSPs	The collection and preservation of forensic evidence is challenging because there is lack of interoperability between CSPs and lack of control from the consumer's end regarding the proprietary architecture and/or the technology used
Single points of failure	The cloud ecosystem has single points of failure, which may have adverse impacts on the evidence acquisition process
Criminals can access low cost computing power	Cloud computing provides computing power that would otherwise be not available to criminals at a low budget, thus letting unpredictable attacks that would be unfeasible outside a cloud environment
Real-time investigation via intelligent process is not possible	Investigating real-time incidents in the cloud is difficult because it requires an intelligence process, which is often not possible while working along with the CSPs or other actors; additionally, a special legal measure must be applied in many cases to collect data
Malicious code may circumvent VM isolation methods	Vulnerabilities in server virtualization allow malicious codes to evade the VM isolation methods and interfere with other guest VMs or the hypervisor
Multiple venues and geo-locations	Managing the scope of data collection is challenging because distributed data collection and chain of custody from multiple venues or unknown geo-locations can cause various jurisdictional issues
Lack of transparency	The operational details of a cloud are not apparent to investigators; this results in lack of trust and difficulties in auditing
Criminals can hide in cloud	The distributed nature of cloud computing allows criminal organizations to maintain isolated cells of operation to preserve the anonymity of each cell from others, thus it may be difficult for investigators to identify and correlate the cells
Cloud confiscation and resource seizure	Cloud confiscation and resource seizure may often affect the business continuity of other tenants
Errors in cloud management portal configurations	Configuration errors in cloud management portals may allow an attacker to gain control, reconfigure, or delete another cloud consumer's resources or applications It is difficult to find the source of such unauthorized change because the cloud management portal is simultaneously used by multiple tenants
Potential evidence segregation	The segregation of potential evidence pertaining to a tenant in a multi-tenant cloud system is a challenge because there are no technologies that do this without breaching the confidentiality of other tenants
Boundaries	Protecting system boundaries is challenging because it is difficult to define system interfaces
Secure provenance	It is a challenge for investigators to maintain a proper chain of custody and security of data, metadata, and possibly, the hardware because it is difficult to determine the ownership, custody, or exact location
Data chain of custody	It is probably impossible to identify and validate a data chain of custody owing to the multi-layered and distributed nature of cloud computing

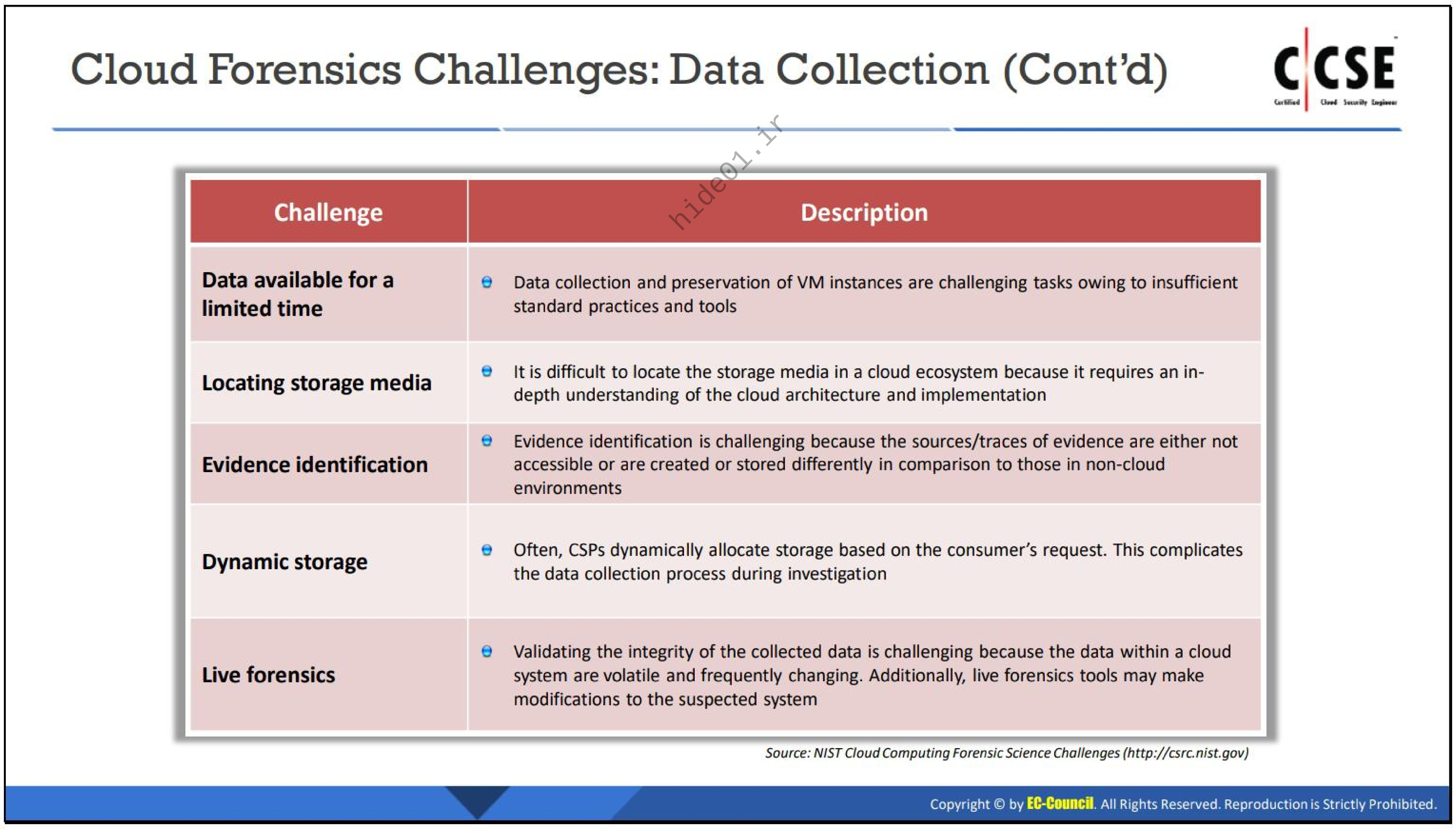
Table 8.1: Cloud Forensics Challenges - Architecture and Identification



Cloud Forensics Challenges: Data Collection

Challenge	Description
Decreased access and data control	In each combination of cloud service and deployment model, the investigator encounters the challenge of limited access and control to the forensic data
Chain of dependencies	The CSPs and most of the cloud applications often rely on other CSP(s), and the dependencies in a chain of CSP(s)/client(s) can be prominently dynamic In such conditions, cloud investigation may depend on the examination of each link in the chain and the level of complexity of the dependencies
Locating evidence	Locating and collecting evidence is a challenge because data in cloud may be quickly altered or lost and there is limited knowledge regarding where or how data is stored in the cloud
Data Location	Collecting data from the target is challenging because it is stored in different data centers and geographic regions
Imaging and isolating data	Data imaging and isolating a migrating data target is challenging in the cloud ecosystem owing to its key characteristics: elasticity, automatic provisioning/deprovisioning of resources, redundancy, and multi-tenancy

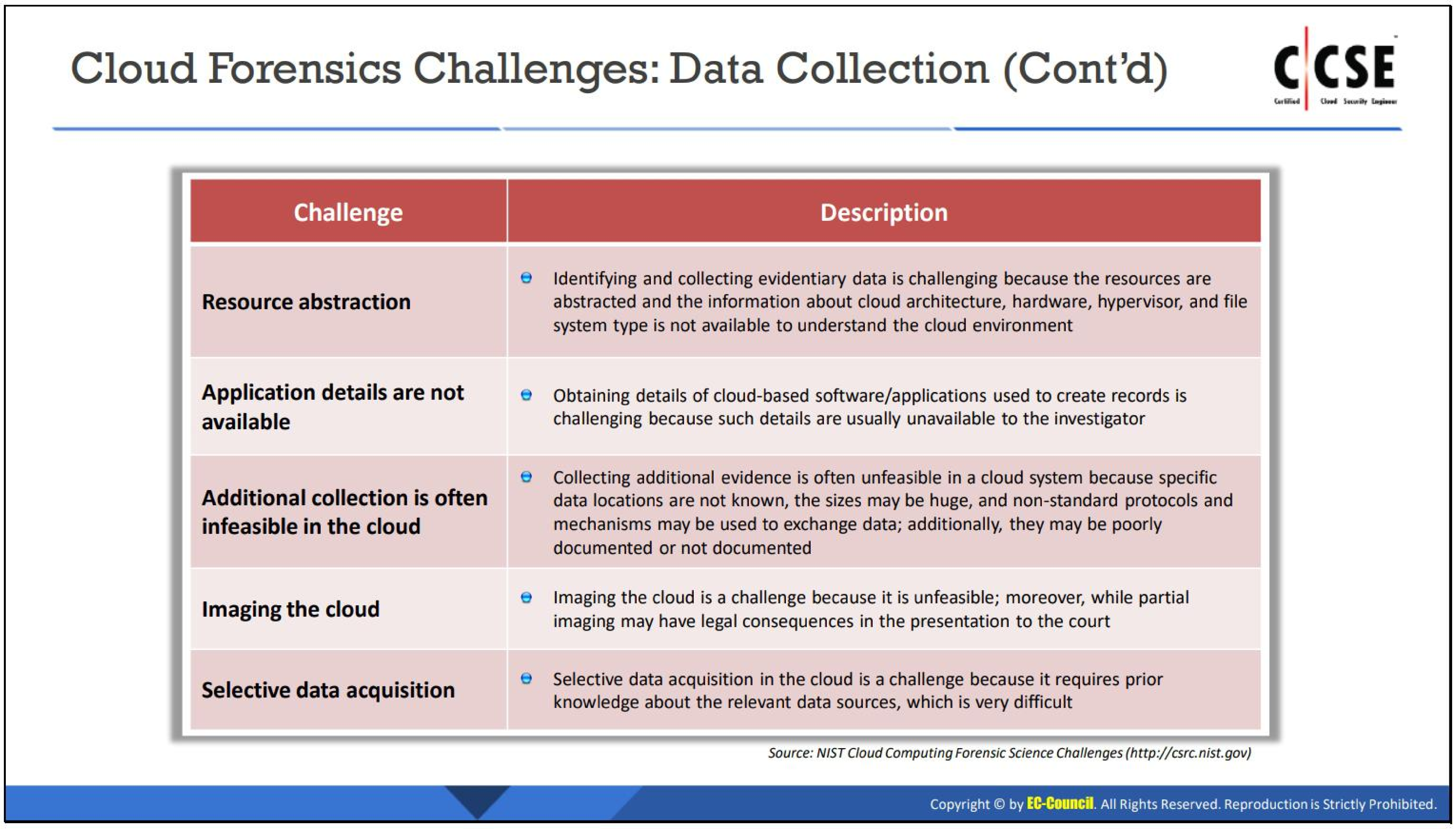
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



Cloud Forensics Challenges: Data Collection (Cont'd)

Challenge	Description
Data available for a limited time	Data collection and preservation of VM instances are challenging tasks owing to insufficient standard practices and tools
Locating storage media	It is difficult to locate the storage media in a cloud ecosystem because it requires an in-depth understanding of the cloud architecture and implementation
Evidence identification	Evidence identification is challenging because the sources/traces of evidence are either not accessible or are created or stored differently in comparison to those in non-cloud environments
Dynamic storage	Often, CSPs dynamically allocate storage based on the consumer's request. This complicates the data collection process during investigation
Live forensics	Validating the integrity of the collected data is challenging because the data within a cloud system are volatile and frequently changing. Additionally, live forensics tools may make modifications to the suspected system

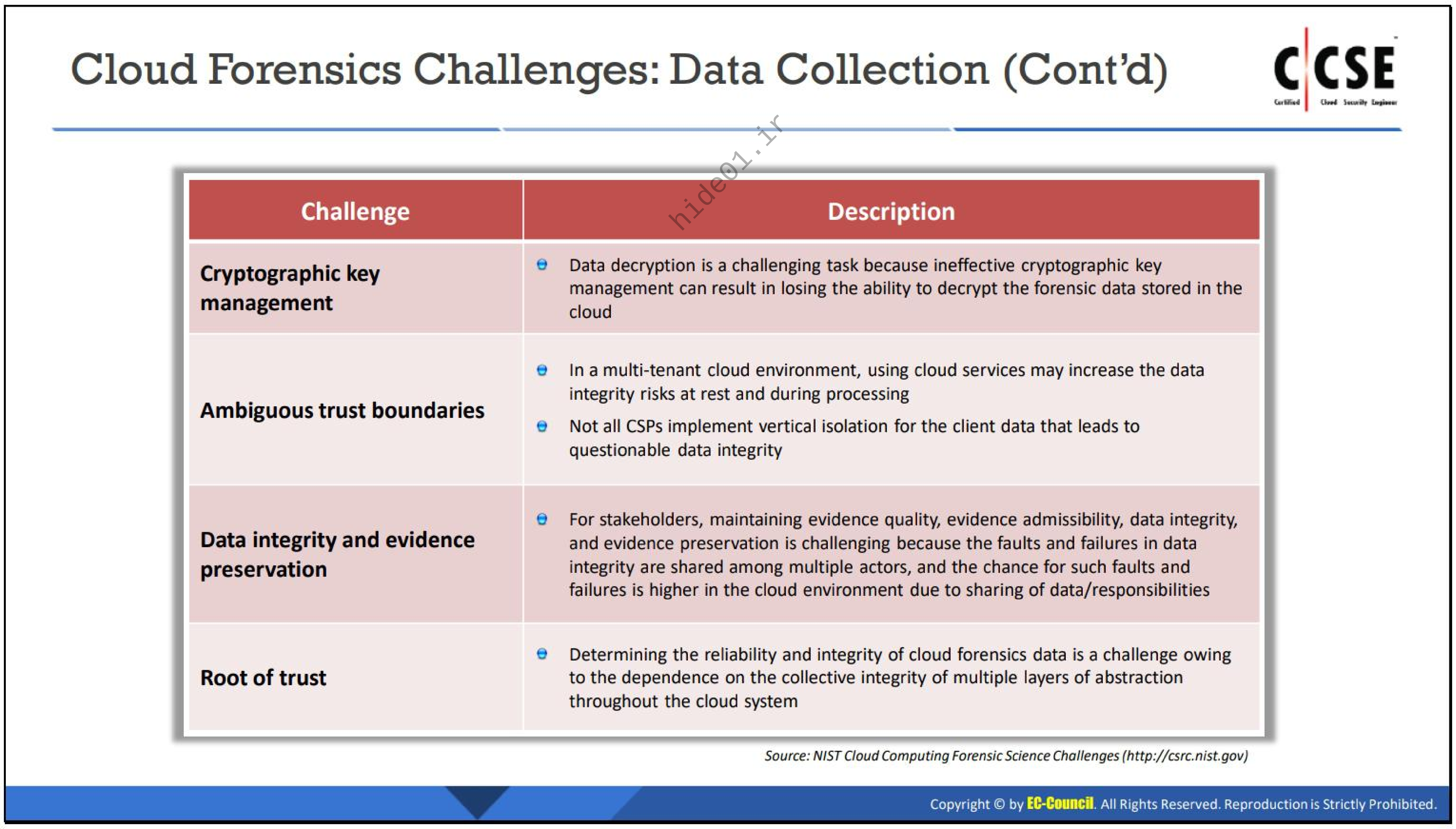
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



Cloud Forensics Challenges: Data Collection (Cont'd)

Challenge	Description
Resource abstraction	Identifying and collecting evidentiary data is challenging because the resources are abstracted and the information about cloud architecture, hardware, hypervisor, and file system type is not available to understand the cloud environment
Application details are not available	Obtaining details of cloud-based software/applications used to create records is challenging because such details are usually unavailable to the investigator
Additional collection is often infeasible in the cloud	Collecting additional evidence is often unfeasible in a cloud system because specific data locations are not known, the sizes may be huge, and non-standard protocols and mechanisms may be used to exchange data; additionally, they may be poorly documented or not documented
Imaging the cloud	Imaging the cloud is a challenge because it is unfeasible; moreover, while partial imaging may have legal consequences in the presentation to the court
Selective data acquisition	Selective data acquisition in the cloud is a challenge because it requires prior knowledge about the relevant data sources, which is very difficult

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

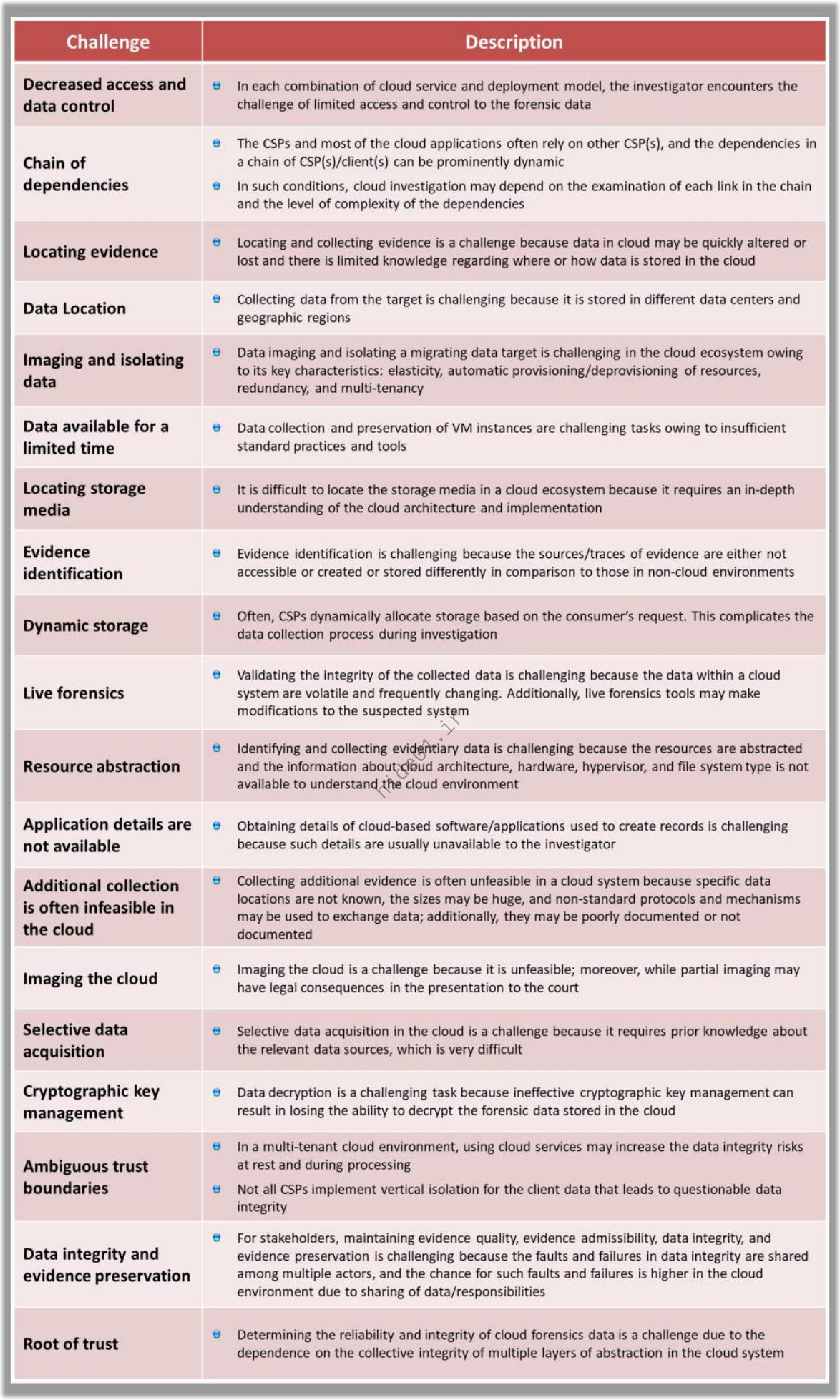


Cloud Forensics Challenges: Data Collection (Cont'd)

Challenge	Description
Cryptographic key management	Data decryption is a challenging task because ineffective cryptographic key management can result in losing the ability to decrypt the forensic data stored in the cloud
Ambiguous trust boundaries	In a multi-tenant cloud environment, using cloud services may increase the data integrity risks at rest and during processing Not all CSPs implement vertical isolation for the client data that leads to questionable data integrity
Data integrity and evidence preservation	For stakeholders, maintaining evidence quality, evidence admissibility, data integrity, and evidence preservation is challenging because the faults and failures in data integrity are shared among multiple actors, and the chance for such faults and failures is higher in the cloud environment due to sharing of data/responsibilities
Root of trust	Determining the reliability and integrity of cloud forensics data is a challenge owing to the dependence on the collective integrity of multiple layers of abstraction throughout the cloud system

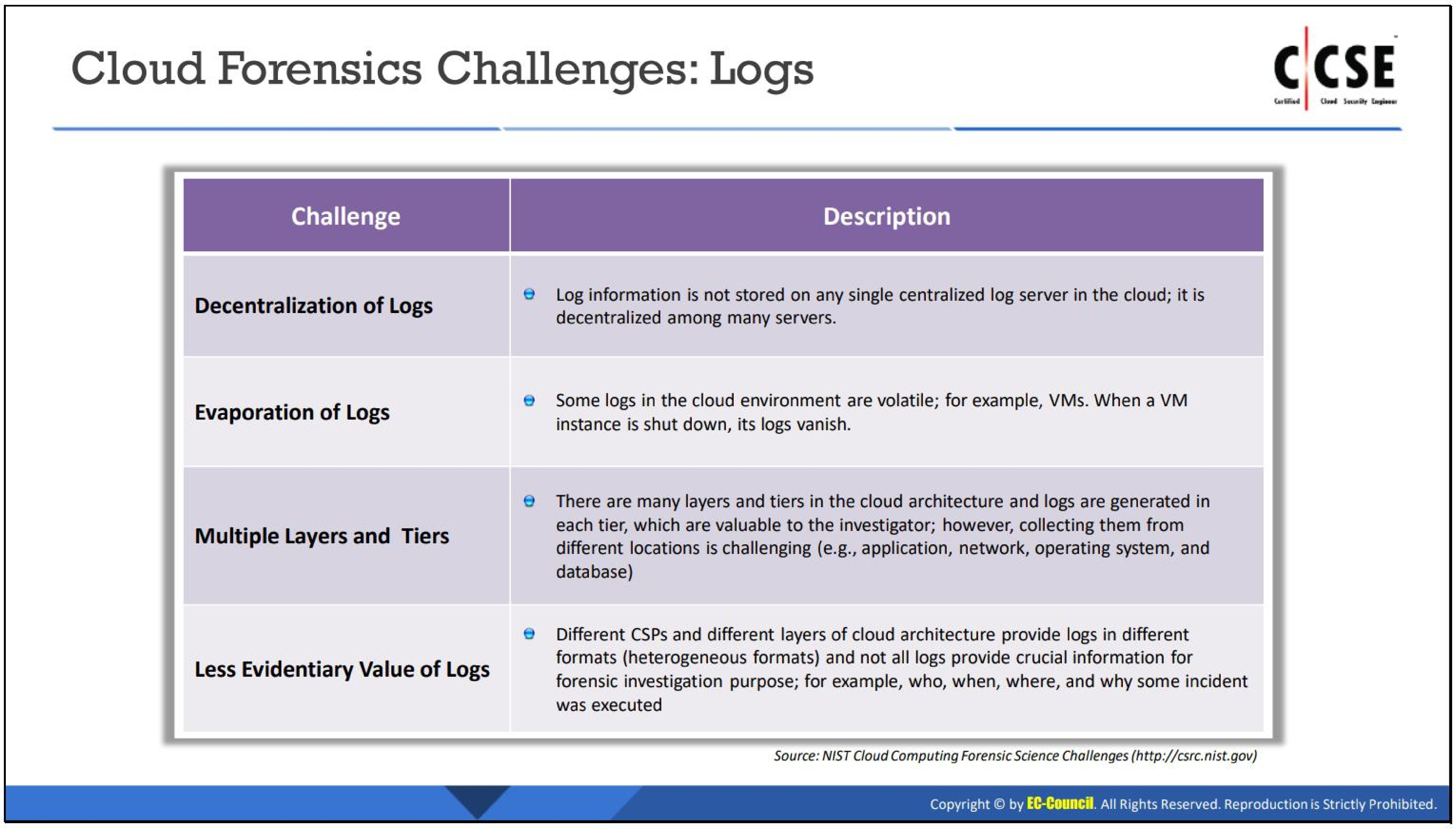
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

Cloud Forensics Challenges: Data Collection



Challenge	Description
Decreased access and data control	In each combination of cloud service and deployment model, the investigator encounters the challenge of limited access and control to the forensic data
Chain of dependencies	The CSPs and most of the cloud applications often rely on other CSP(s), and the dependencies in a chain of CSP(s)/client(s) can be prominently dynamic In such conditions, cloud investigation may depend on the examination of each link in the chain and the level of complexity of the dependencies
Locating evidence	Locating and collecting evidence is a challenge because data in cloud may be quickly altered or lost and there is limited knowledge regarding where or how data is stored in the cloud
Data Location	Collecting data from the target is challenging because it is stored in different data centers and geographic regions
Imaging and isolating data	Data imaging and isolating a migrating data target is challenging in the cloud ecosystem owing to its key characteristics: elasticity, automatic provisioning/deprovisioning of resources, redundancy, and multi-tenancy
Data available for a limited time	Data collection and preservation of VM instances are challenging tasks owing to insufficient standard practices and tools
Locating storage media	It is difficult to locate the storage media in a cloud ecosystem because it requires an in-depth understanding of the cloud architecture and implementation
Evidence identification	Evidence identification is challenging because the sources/traces of evidence are either not accessible or created or stored differently in comparison to those in non-cloud environments
Dynamic storage	Often, CSPs dynamically allocate storage based on the consumer's request. This complicates the data collection process during investigation
Live forensics	Validating the integrity of the collected data is challenging because the data within a cloud system are volatile and frequently changing. Additionally, live forensics tools may make modifications to the suspected system
Resource abstraction	Identifying and collecting evidentiary data is challenging because the resources are abstracted and the information about cloud architecture, hardware, hypervisor, and file system type is not available to understand the cloud environment
Application details are not available	Obtaining details of cloud-based software/applications used to create records is challenging because such details are usually unavailable to the investigator
Additional collection is often infeasible in the cloud	Collecting additional evidence is often unfeasible in a cloud system because specific data locations are not known, the sizes may be huge, and non-standard protocols and mechanisms may be used to exchange data; additionally, they may be poorly documented or not documented
Imaging the cloud	Imaging the cloud is a challenge because it is unfeasible; moreover, while partial imaging may have legal consequences in the presentation to the court
Selective data acquisition	Selective data acquisition in the cloud is a challenge because it requires prior knowledge about the relevant data sources, which is very difficult
Cryptographic key management	Data decryption is a challenging task because ineffective cryptographic key management can result in losing the ability to decrypt the forensic data stored in the cloud
Ambiguous trust boundaries	In a multi-tenant cloud environment, using cloud services may increase the data integrity risks at rest and during processing Not all CSPs implement vertical isolation for the client data that leads to questionable data integrity
Data integrity and evidence preservation	For stakeholders, maintaining evidence quality, evidence admissibility, data integrity, and evidence preservation is challenging because the faults and failures in data integrity are shared among multiple actors, and the chance for such faults and failures is higher in the cloud environment due to sharing of data/responsibilities
Root of trust	Determining the reliability and integrity of cloud forensics data is a challenge due to the dependence on the collective integrity of multiple layers of abstraction in the cloud system

Table 8.2: Cloud Forensics Challenges – Data Collection



Cloud Forensics Challenges: Logs

Challenge	Description
Decentralization of Logs	Log information is not stored on any single centralized log server in the cloud; it is decentralized among many servers.
Evaporation of Logs	Some logs in the cloud environment are volatile; for example, VMs. When a VM instance is shut down, its logs vanish.
Multiple Layers and Tiers	There are many layers and tiers in the cloud architecture and logs are generated in each tier, which are valuable to the investigator; however, collecting them from different locations is challenging (e.g., application, network, operating system, and database)
Less Evidentiary Value of Logs	Different CSPs and different layers of cloud architecture provide logs in different formats (heterogeneous formats) and not all logs provide crucial information for forensic investigation purpose; for example, who, when, where, and why some incident was executed

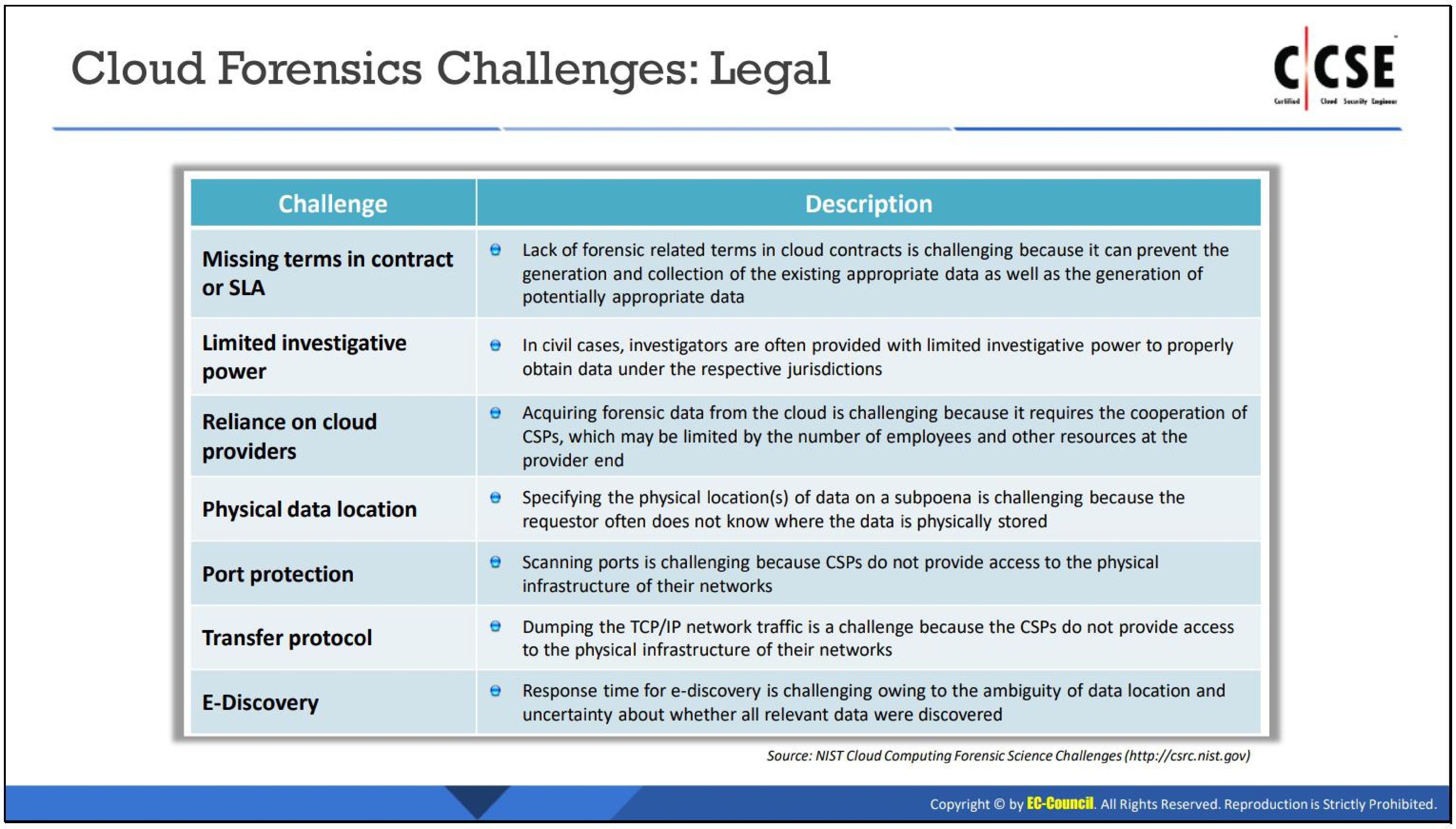
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

Cloud Forensics Challenges: Logs

***Source**: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)*

Challenge	Description
Decentralization of Logs	Log information is not stored on any single centralized log server in the cloud; it is decentralized among many servers.
Evaporation of Logs	Some logs in the cloud environment are volatile; for example, VMs. When a VM instance is shut down, its logs vanish.
Multiple Layers and Tiers	There are many layers and tiers in the cloud architecture and logs are generated in each tier, which are valuable to the investigator; however, collecting them from different locations is challenging (e.g., application, network, operating system, and database)
Less Evidentiary Value of Logs	Different CSPs and different layers of cloud architecture provide logs in different formats (heterogeneous formats) and not all logs provide crucial information for forensic investigation purpose; for example, who, when, where, and why some incident was executed

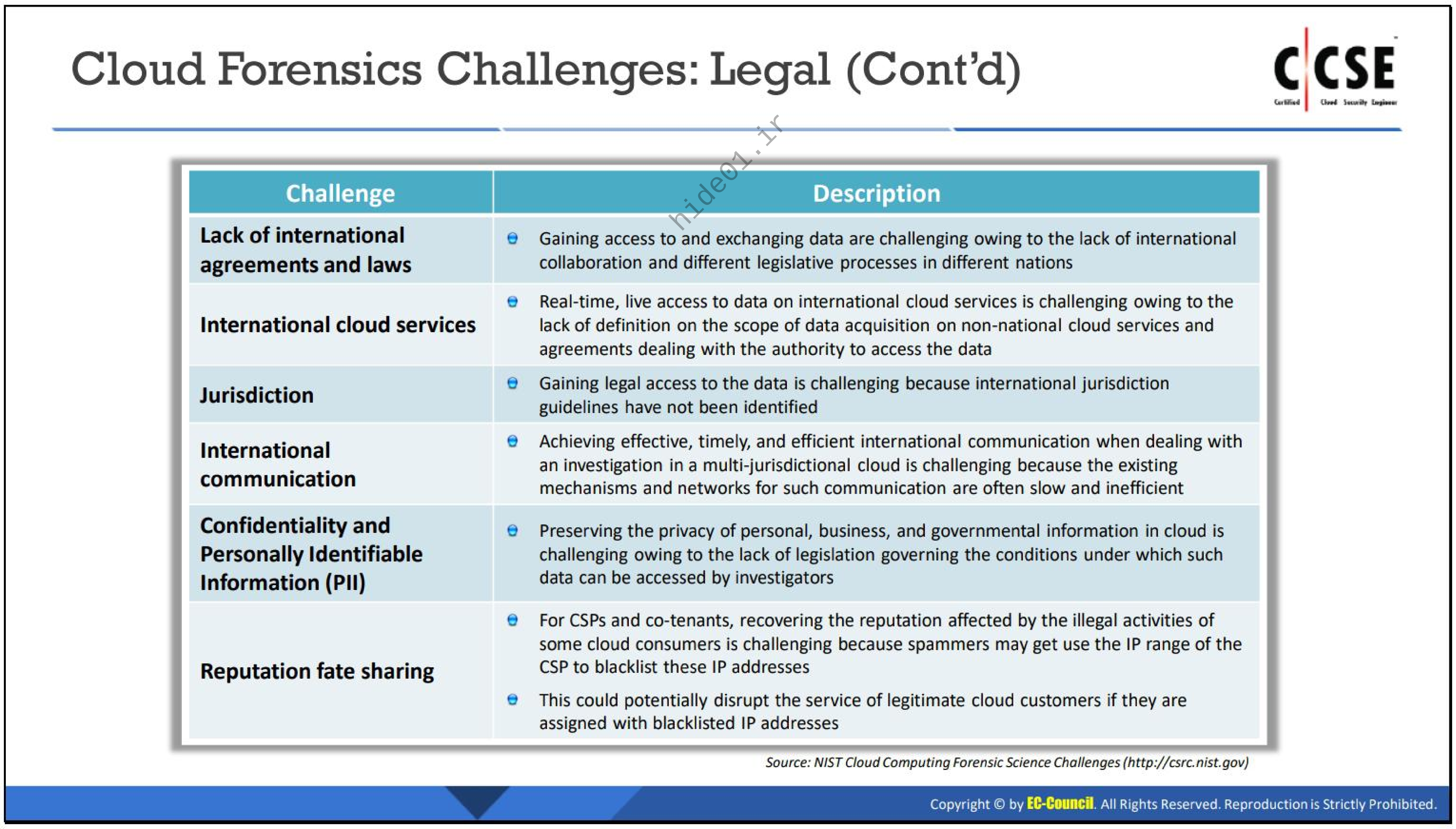
Table 8.3: Cloud Forensics Challenges – Logs



Cloud Forensics Challenges: Legal

Challenge	Description
Missing terms in contract or SLA	Lack of forensic related terms in cloud contracts is challenging because it can prevent the generation and collection of the existing appropriate data as well as the generation of potentially appropriate data
Limited investigative power	In civil cases, investigators are often provided with limited investigative power to properly obtain data under the respective jurisdictions
Reliance on cloud providers	Acquiring forensic data from the cloud is challenging because it requires the cooperation of CSPs, which may be limited by the number of employees and other resources at the provider end
Physical data location	Specifying the physical location(s) of data on a subpoena is challenging because the requestor often does not know where the data is physically stored
Port protection	Scanning ports is challenging because CSPs do not provide access to the physical infrastructure of their networks
Transfer protocol	Dumping the TCP/IP network traffic is a challenge because the CSPs do not provide access to the physical infrastructure of their networks
E-Discovery	Response time for e-discovery is challenging owing to the ambiguity of data location and uncertainty about whether all relevant data were discovered

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



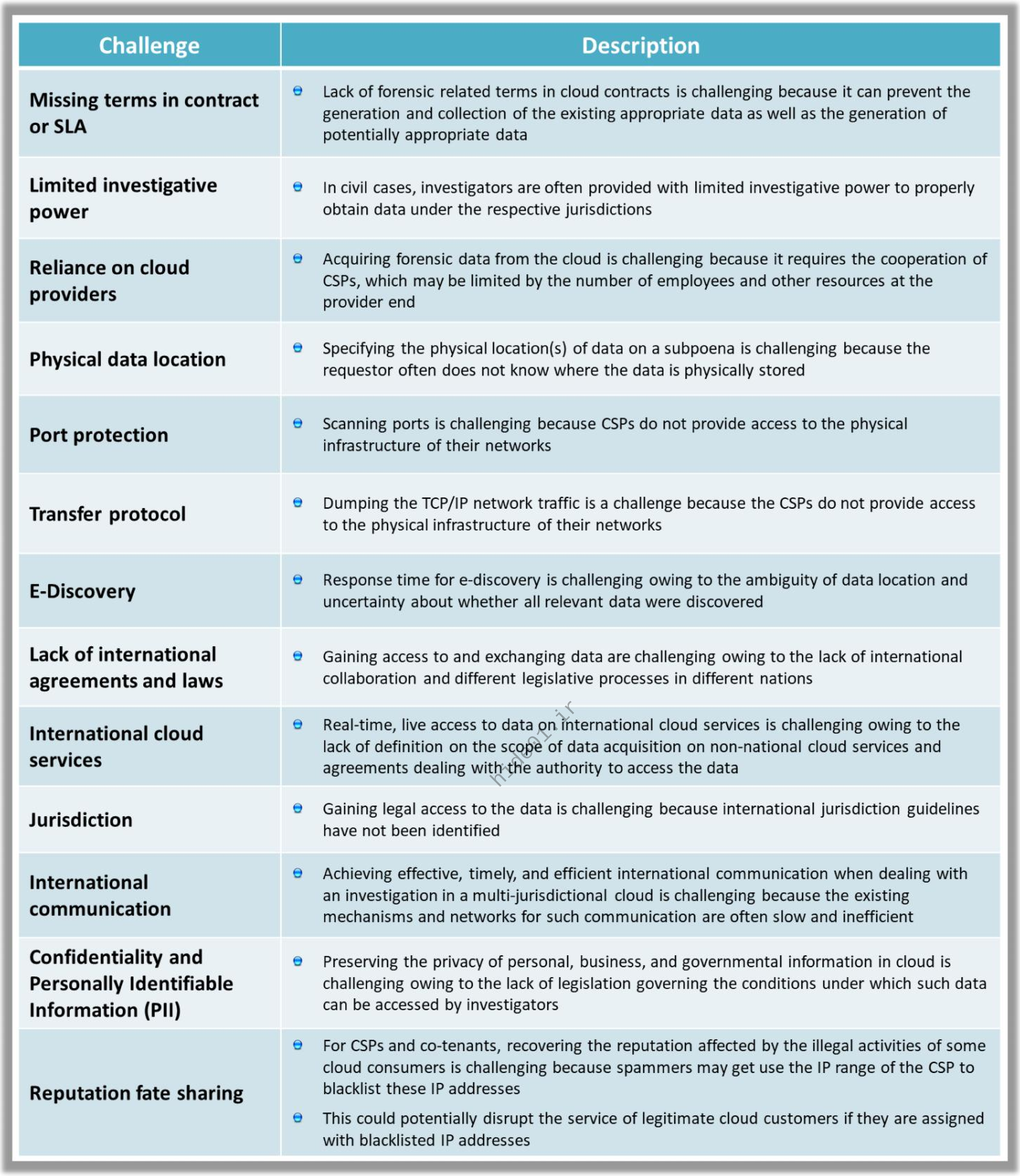
Cloud Forensics Challenges: Legal (Cont'd)

Challenge	Description
Lack of international agreements and laws	Gaining access to and exchanging data are challenging owing to the lack of international collaboration and different legislative processes in different nations
International cloud services	Real-time, live access to data on international cloud services is challenging owing to the lack of definition on the scope of data acquisition on non-national cloud services and agreements dealing with the authority to access the data
Jurisdiction	Gaining legal access to the data is challenging because international jurisdiction guidelines have not been identified
International communication	Achieving effective, timely, and efficient international communication when dealing with an investigation in a multi-jurisdictional cloud is challenging because the existing mechanisms and networks for such communication are often slow and inefficient
Confidentiality and Personally Identifiable Information (PII)	Preserving the privacy of personal, business, and governmental information in cloud is challenging owing to the lack of legislation governing the conditions under which such data can be accessed by investigators
Reputation fate sharing	For CSPs and co-tenants, recovering the reputation affected by the illegal activities of some cloud consumers is challenging because spammers may get use the IP range of the CSP to blacklist these IP addresses This could potentially disrupt the service of legitimate cloud customers if they are assigned with blacklisted IP addresses

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

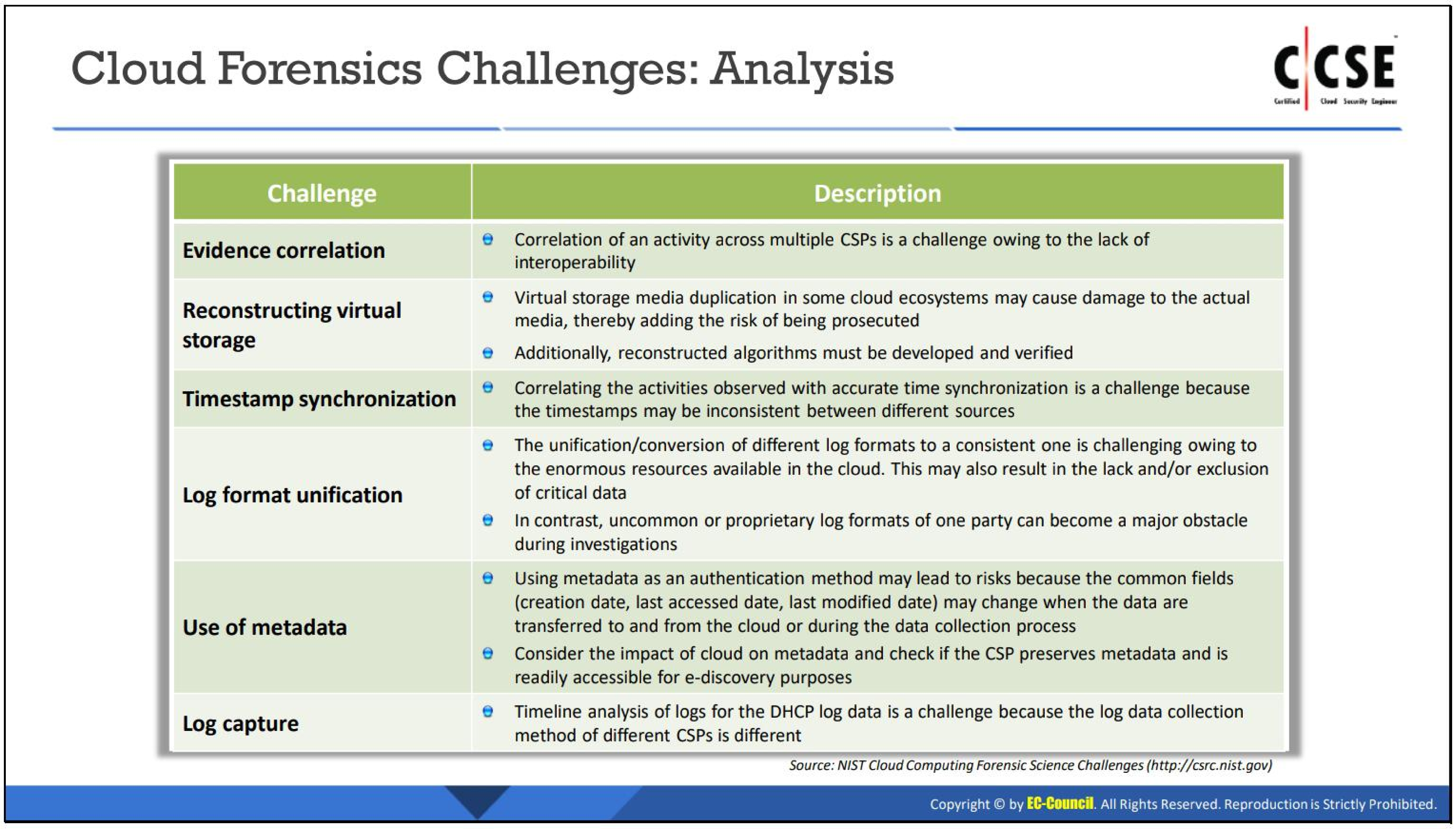
Cloud Forensics Challenges: Legal

***Source**: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)*



Challenge	Description
Missing terms in contract or SLA	Lack of forensic related terms in cloud contracts is challenging because it can prevent the generation and collection of the existing appropriate data as well as the generation of potentially appropriate data
Limited investigative power	In civil cases, investigators are often provided with limited investigative power to properly obtain data under the respective jurisdictions
Reliance on cloud providers	Acquiring forensic data from the cloud is challenging because it requires the cooperation of CSPs, which may be limited by the number of employees and other resources at the provider end
Physical data location	Specifying the physical location(s) of data on a subpoena is challenging because the requestor often does not know where the data is physically stored
Port protection	Scanning ports is challenging because CSPs do not provide access to the physical infrastructure of their networks
Transfer protocol	Dumping the TCP/IP network traffic is a challenge because the CSPs do not provide access to the physical infrastructure of their networks
E-Discovery	Response time for e-discovery is challenging owing to the ambiguity of data location and uncertainty about whether all relevant data were discovered
Lack of international agreements and laws	Gaining access to and exchanging data are challenging owing to the lack of international collaboration and different legislative processes in different nations
International cloud services	Real-time, live access to data on international cloud services is challenging owing to the lack of definition on the scope of data acquisition on non-national cloud services and agreements dealing with the authority to access the data
Jurisdiction	Gaining legal access to the data is challenging because international jurisdiction guidelines have not been identified
International communication	Achieving effective, timely, and efficient international communication when dealing with an investigation in a multi-jurisdictional cloud is challenging because the existing mechanisms and networks for such communication are often slow and inefficient
Confidentiality and Personally Identifiable Information (PII)	Preserving the privacy of personal, business, and governmental information in cloud is challenging owing to the lack of legislation governing the conditions under which such data can be accessed by investigators
Reputation fate sharing	For CSPs and co-tenants, recovering the reputation affected by the illegal activities of some cloud consumers is challenging because spammers may get use the IP range of the CSP to blacklist these IP addresses This could potentially disrupt the service of legitimate cloud customers if they are assigned with blacklisted IP addresses

Table 8.4: Cloud Forensics Challenges – Legal



Cloud Forensics Challenges: Analysis

Challenge	Description
Evidence correlation	- Correlation of an activity across multiple CSPs is a challenge owing to the lack of interoperability
Reconstructing virtual storage	- Virtual storage media duplication in some cloud ecosystems may cause damage to the actual media, thereby adding the risk of being prosecuted - Additionally, reconstructed algorithms must be developed and verified
Timestamp synchronization	- Correlating the activities observed with accurate time synchronization is a challenge because the timestamps may be inconsistent between different sources
Log format unification	- The unification/conversion of different log formats to a consistent one is challenging owing to the enormous resources available in the cloud. This may also result in the lack and/or exclusion of critical data - In contrast, uncommon or proprietary log formats of one party can become a major obstacle during investigations
Use of metadata	- Using metadata as an authentication method may lead to risks because the common fields (creation date, last accessed date, last modified date) may change when the data are transferred to and from the cloud or during the data collection process - Consider the impact of cloud on metadata and check if the CSP preserves metadata and is readily accessible for e-discovery purposes
Log capture	- Timeline analysis of logs for the DHCP log data is a challenge because the log data collection method of different CSPs is different

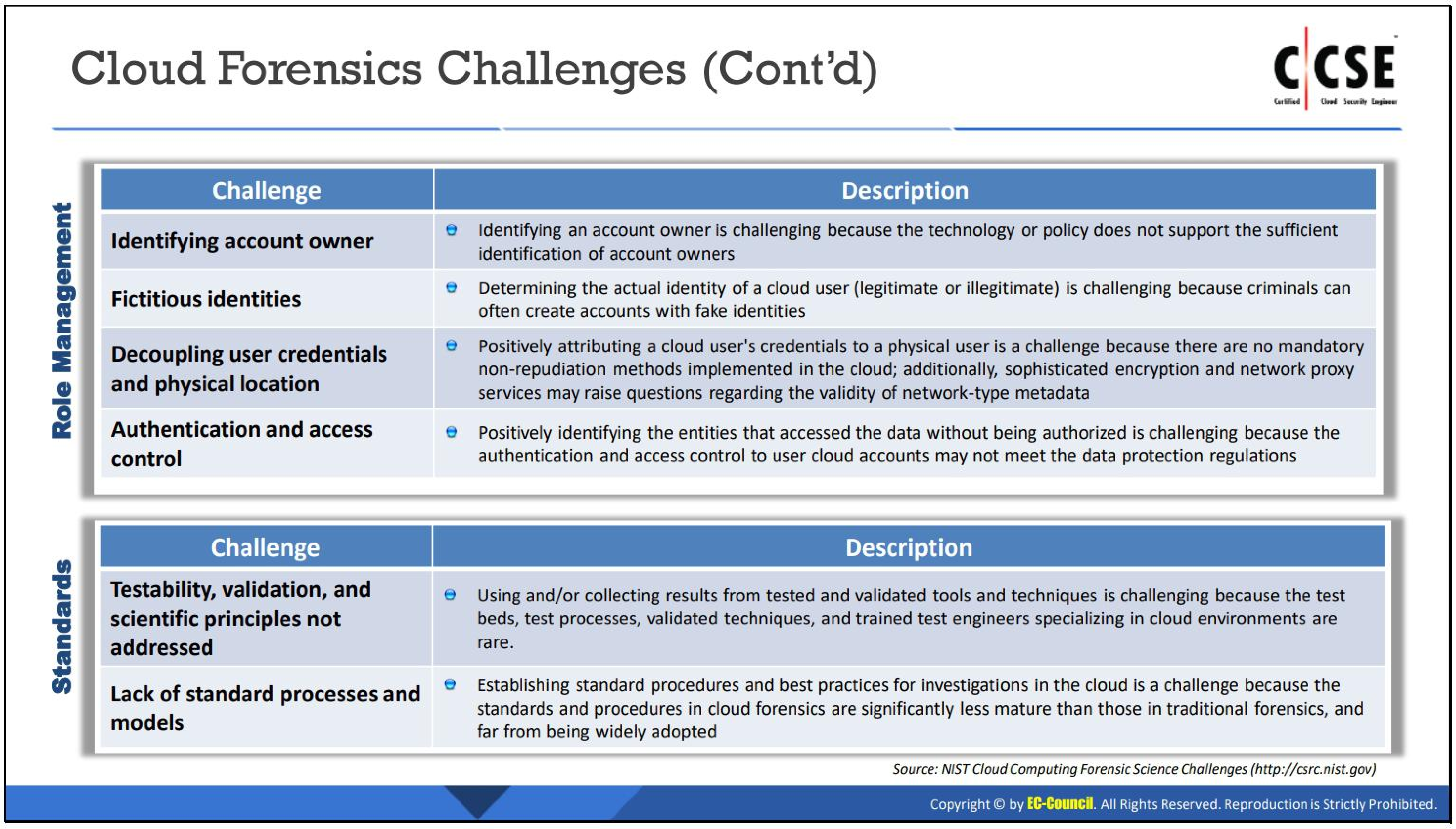
Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

Cloud Forensics Challenges: Analysis

***Source**: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)*

Challenge	Description
Evidence correlation	- Correlation of an activity across multiple CSPs is a challenge owing to the lack of interoperability
Reconstructing virtual storage	- Virtual storage media duplication in some cloud ecosystems may cause damage to the actual media, thereby adding the risk of being prosecuted - Additionally, reconstructed algorithms must be developed and verified
Timestamp synchronization	- Correlating the activities observed with accurate time synchronization is a challenge because the timestamps may be inconsistent between different sources
Log format unification	- The unification/conversion of different log formats to a consistent one is challenging owing to the enormous resources available in the cloud. This may also result in the lack and/or exclusion of critical data - In contrast, uncommon or proprietary log formats of one party can become a major obstacle during investigations
Use of metadata	- Using metadata as an authentication method may lead to risks because the common fields (creation date, last accessed date, last modified date) may change when the data are transferred to and from the cloud or during the data collection process - Consider the impact of cloud on metadata and check if the CSP preserves metadata and is readily accessible for e-discovery purposes
Log capture	- Timeline analysis of logs for the DHCP log data is a challenge because the log data collection method of different CSPs is different

Table 8.5: Cloud Forensics Challenges – Analysis



Cloud Forensics Challenges (Cont'd)

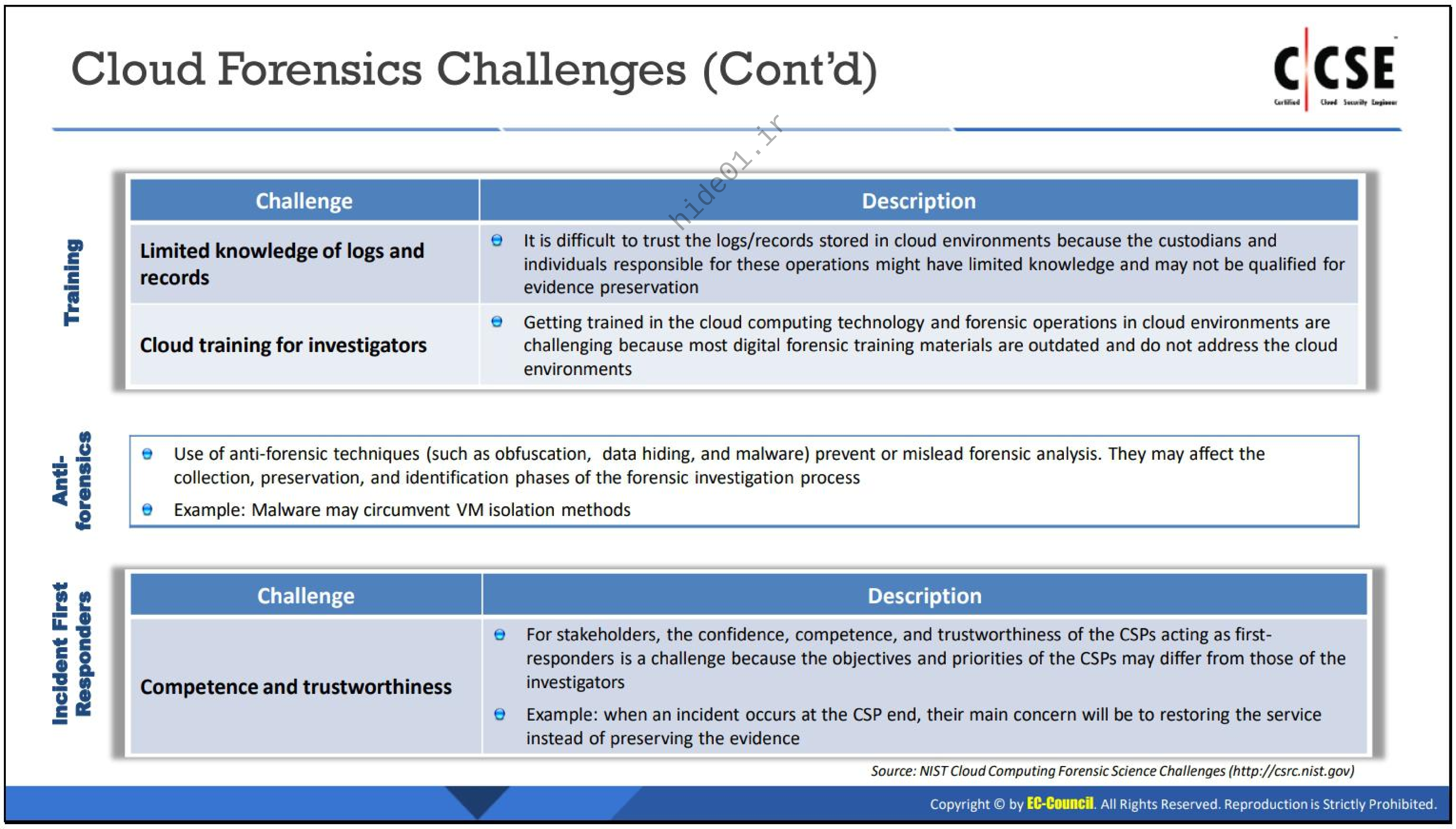
Role Management

Challenge	Description
Identifying account owner	Identifying an account owner is challenging because the technology or policy does not support the sufficient identification of account owners
Fictitious identities	Determining the actual identity of a cloud user (legitimate or illegitimate) is challenging because criminals can often create accounts with fake identities
Decoupling user credentials and physical location	Positively attributing a cloud user's credentials to a physical user is a challenge because there are no mandatory non-repudiation methods implemented in the cloud; additionally, sophisticated encryption and network proxy services may raise questions regarding the validity of network-type metadata
Authentication and access control	Positively identifying the entities that accessed the data without being authorized is challenging because the authentication and access control to user cloud accounts may not meet the data protection regulations

Standards

Challenge	Description
Testability, validation, and scientific principles not addressed	Using and/or collecting results from tested and validated tools and techniques is challenging because the test beds, test processes, validated techniques, and trained test engineers specializing in cloud environments are rare.
Lack of standard processes and models	Establishing standard procedures and best practices for investigations in the cloud is a challenge because the standards and procedures in cloud forensics are significantly less mature than those in traditional forensics, and far from being widely adopted

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)



Cloud Forensics Challenges (Cont'd)

Training

Challenge	Description
Limited knowledge of logs and records	It is difficult to trust the logs/records stored in cloud environments because the custodians and individuals responsible for these operations might have limited knowledge and may not be qualified for evidence preservation
Cloud training for investigators	Getting trained in the cloud computing technology and forensic operations in cloud environments are challenging because most digital forensic training materials are outdated and do not address the cloud environments

Anti-forensics

- Use of anti-forensic techniques (such as obfuscation, data hiding, and malware) prevent or mislead forensic analysis. They may affect the collection, preservation, and identification phases of the forensic investigation process
- Example: Malware may circumvent VM isolation methods

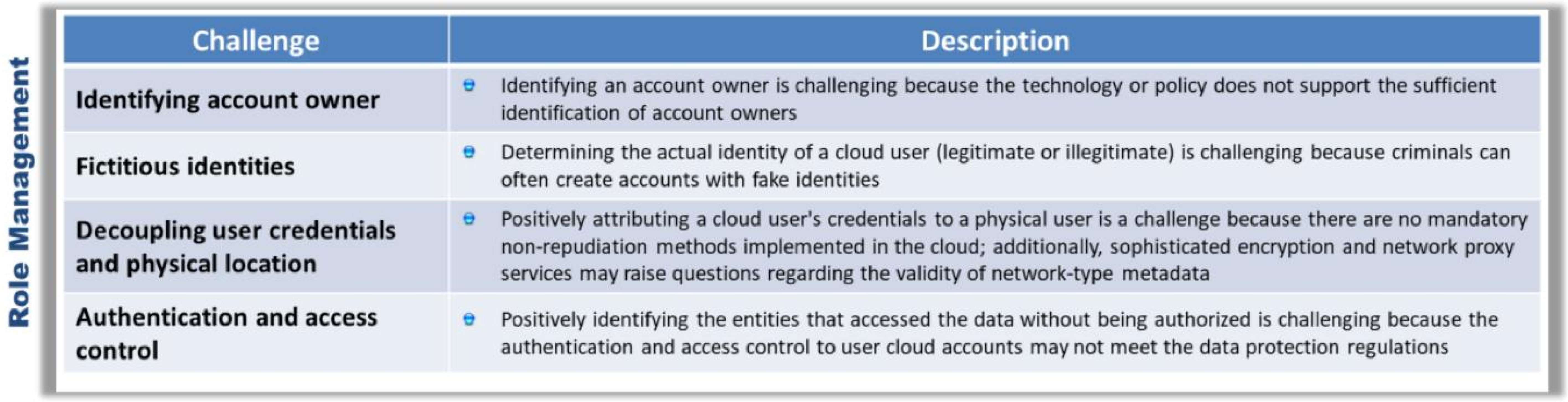
Incident First Responders

Challenge	Description
Competence and trustworthiness	For stakeholders, the confidence, competence, and trustworthiness of the CSPs acting as first-responders is a challenge because the objectives and priorities of the CSPs may differ from those of the investigators Example: when an incident occurs at the CSP end, their main concern will be to restoring the service instead of preserving the evidence

Source: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)

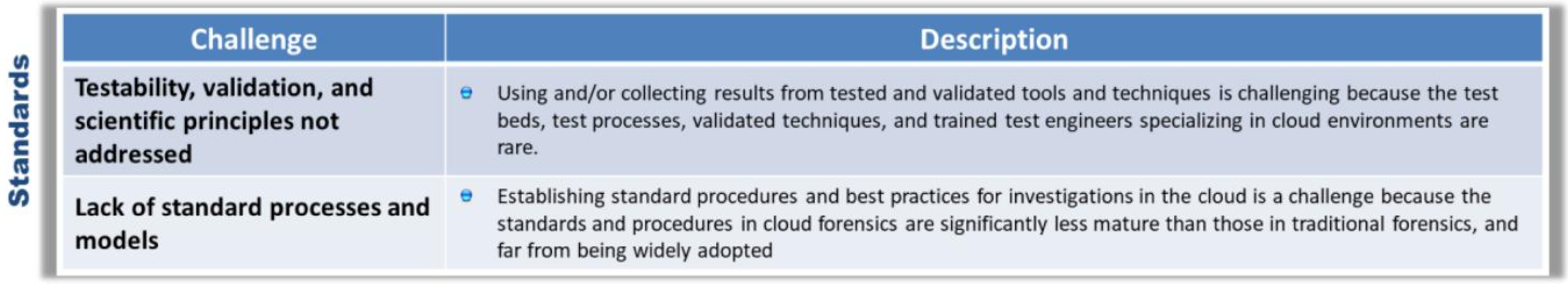
Cloud Forensics Challenges

***Source**: NIST Cloud Computing Forensic Science Challenges (http://csrc.nist.gov)*



Role Management

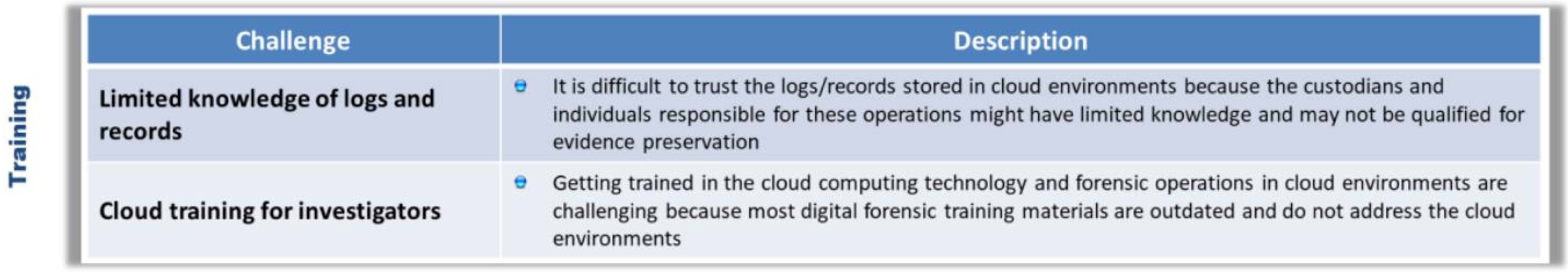
Challenge	Description
Identifying account owner	Identifying an account owner is challenging because the technology or policy does not support the sufficient identification of account owners
Fictitious identities	Determining the actual identity of a cloud user (legitimate or illegitimate) is challenging because criminals can often create accounts with fake identities
Decoupling user credentials and physical location	Positively attributing a cloud user's credentials to a physical user is a challenge because there are no mandatory non-repudiation methods implemented in the cloud; additionally, sophisticated encryption and network proxy services may raise questions regarding the validity of network-type metadata
Authentication and access control	Positively identifying the entities that accessed the data without being authorized is challenging because the authentication and access control to user cloud accounts may not meet the data protection regulations



Standards

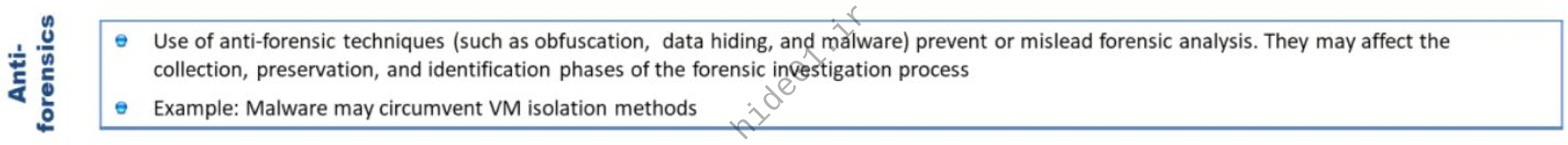
Challenge	Description
Testability, validation, and scientific principles not addressed	Using and/or collecting results from tested and validated tools and techniques is challenging because the test beds, test processes, validated techniques, and trained test engineers specializing in cloud environments are rare.
Lack of standard processes and models	Establishing standard procedures and best practices for investigations in the cloud is a challenge because the standards and procedures in cloud forensics are significantly less mature than those in traditional forensics, and far from being widely adopted

Table 8.6: Cloud Forensics Challenges – Role Management and Standards



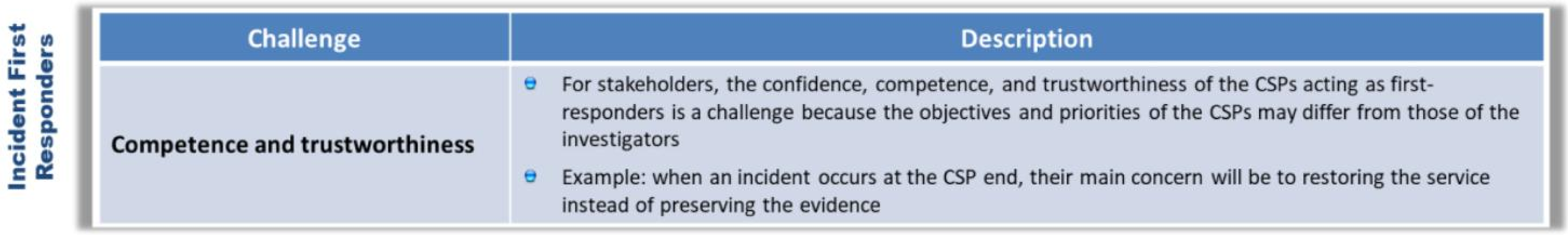
Training

Challenge	Description
Limited knowledge of logs and records	It is difficult to trust the logs/records stored in cloud environments because the custodians and individuals responsible for these operations might have limited knowledge and may not be qualified for evidence preservation
Cloud training for investigators	Getting trained in the cloud computing technology and forensic operations in cloud environments are challenging because most digital forensic training materials are outdated and do not address the cloud environments



Anti-forensics

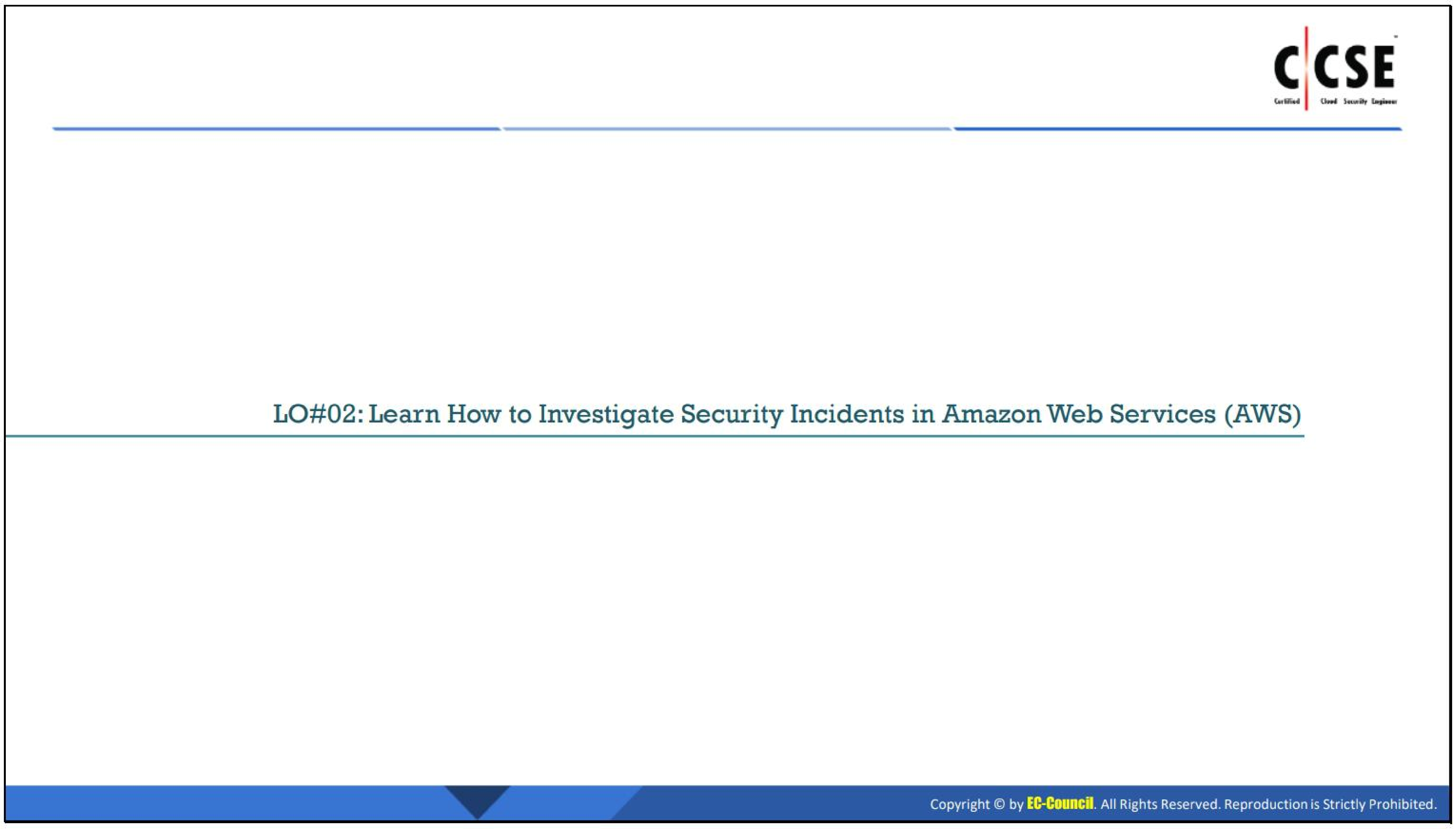
- Use of anti-forensic techniques (such as obfuscation, data hiding, and malware) prevent or mislead forensic analysis. They may affect the collection, preservation, and identification phases of the forensic investigation process
- Example: Malware may circumvent VM isolation methods



Incident First Responders

Challenge	Description
Competence and trustworthiness	For stakeholders, the confidence, competence, and trustworthiness of the CSPs acting as first-responders is a challenge because the objectives and priorities of the CSPs may differ from those of the investigators Example: when an incident occurs at the CSP end, their main concern will be to restoring the service instead of preserving the evidence

Table 8.7: Cloud Forensics Challenges – Training, Anti-Forensics and Incident First Responders

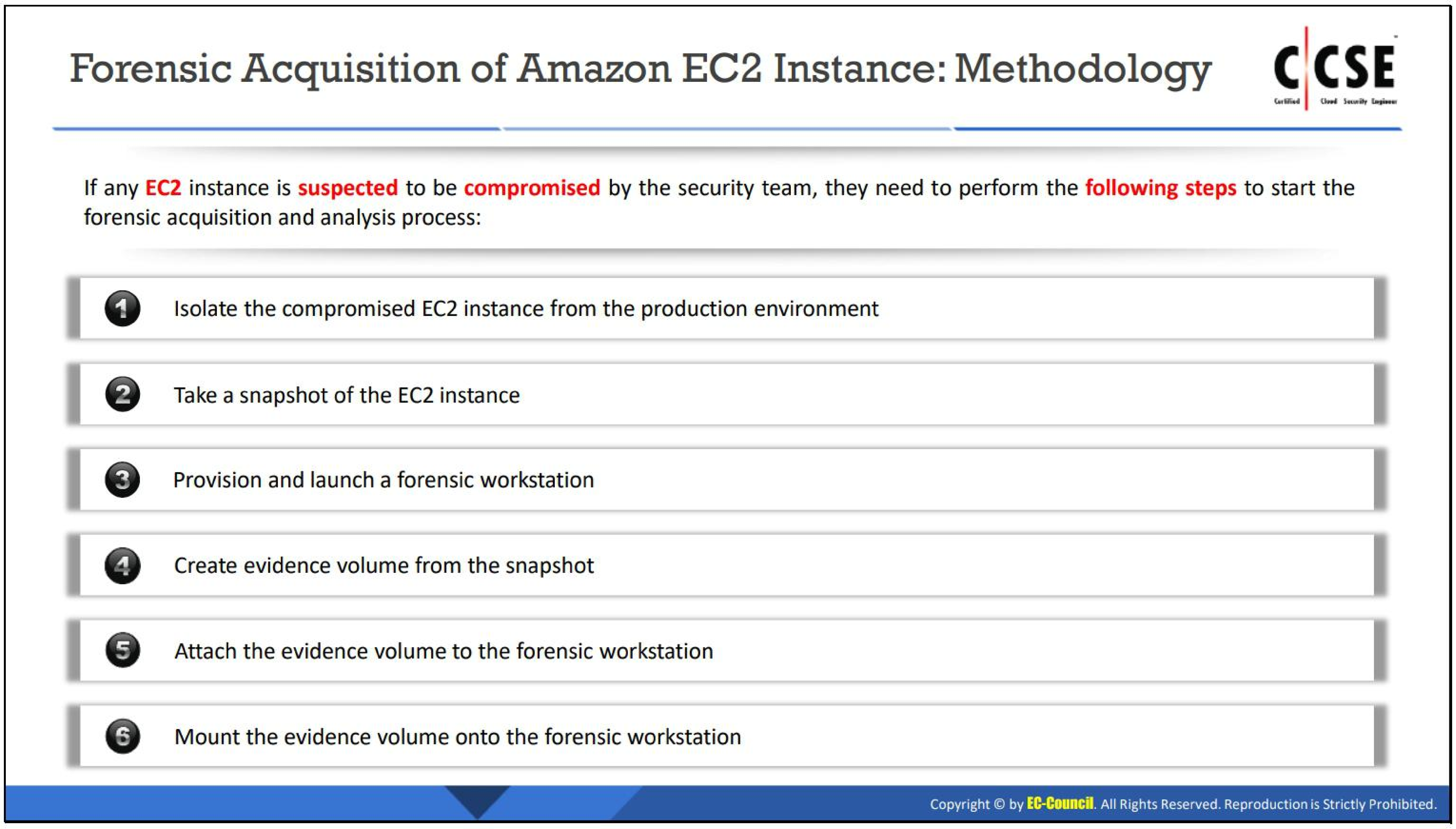


LO#02: Learn How to Investigate Security Incidents in Amazon Web Services (AWS)

LO#02: Learn How to Investigate Security Incidents in Amazon Web Services (AWS)

When a security incident occurs within an AWS environment, the incident response and forensic investigation procedures differ from those performed on an on-premise server.

This section discusses the acquisition of forensic evidence from compromised EC2 instances and the key findings obtained by investigating the log files generated by various AWS resources.



Forensic Acquisition of Amazon EC2 Instance: Methodology

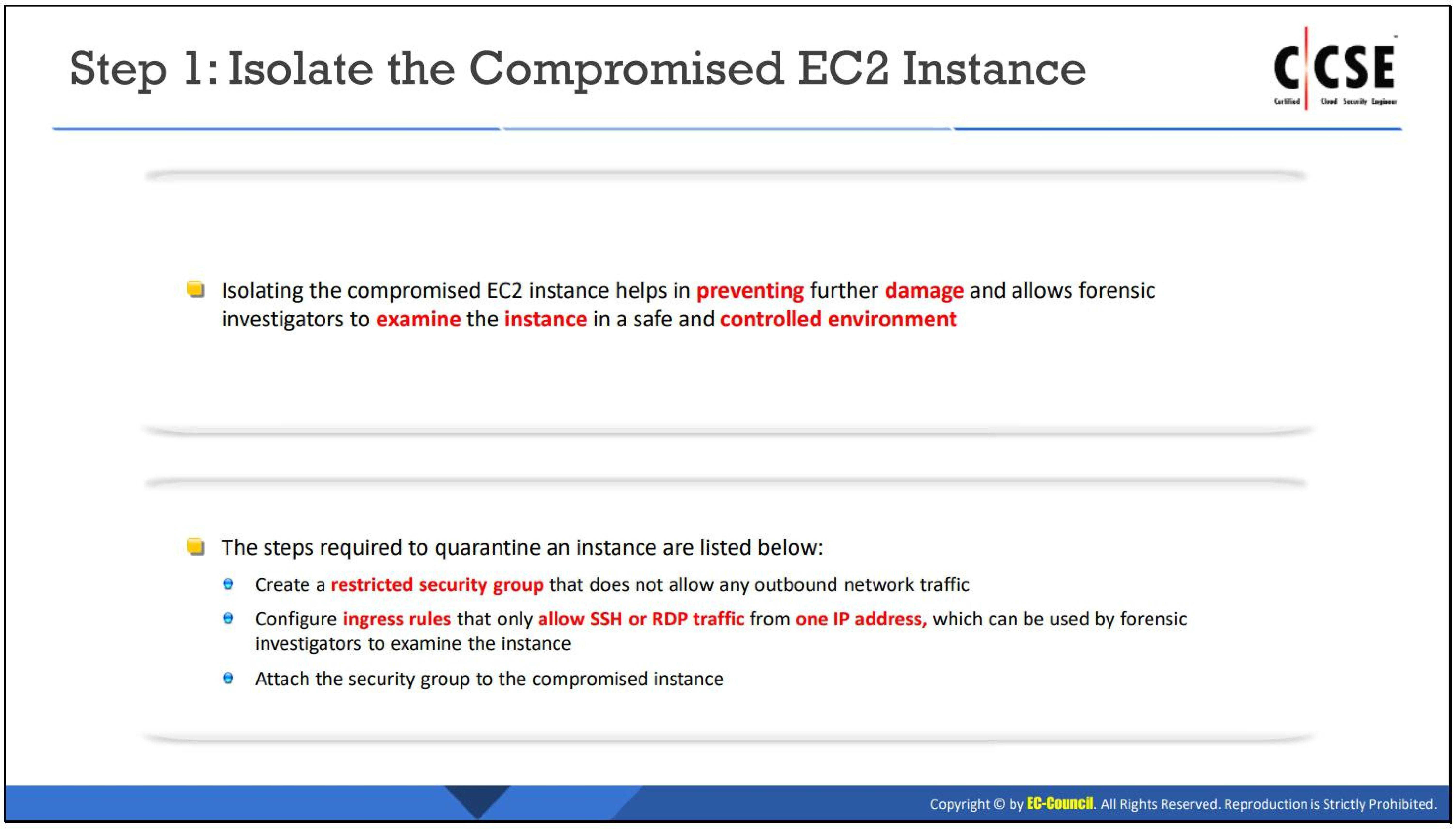
If any **EC2** instance is **suspected** to be **compromised** by the security team, they need to perform the **following steps** to start the forensic acquisition and analysis process:

1. Isolate the compromised EC2 instance from the production environment
2. Take a snapshot of the EC2 instance
3. Provision and launch a forensic workstation
4. Create evidence volume from the snapshot
5. Attach the evidence volume to the forensic workstation
6. Mount the evidence volume onto the forensic workstation

Forensic Acquisition of Amazon EC2 Instance: Methodology

Acquisition is an important part of a forensic investigation. However, the acquisition of compromised instances in a cloud environment follows a different methodology. Given below are the steps involved in the forensic acquisition of an EC2 instance if it is suspected to be compromised.

1. Isolating the compromised EC2 instance from the production environment
2. Taking a snapshot of the EC2 instance
3. Provisioning and launching a forensic workstation
4. Creating evidence volume from the snapshot
5. Attaching evidence volume to the forensic workstation
6. Mounting the evidence volume onto the forensic workstation



Step 1: Isolate the Compromised EC2 Instance

CCSE
Certified Cloud Security Engineer

- Isolating the compromised EC2 instance helps in **preventing** further **damage** and allows forensic investigators to **examine** the **instance** in a safe and **controlled environment**

- The steps required to quarantine an instance are listed below:
  - Create a **restricted security group** that does not allow any outbound network traffic
  - Configure **ingress rules** that only **allow SSH or RDP traffic** from **one IP address,** which can be used by forensic investigators to examine the instance
  - Attach the security group to the compromised instance

Step 1: Isolate the Compromised EC2 Instance

Once the compromised EC2 instance is identified, it is crucial to isolate it from the production environment. For example, isolating an instance in case of malware infection can stop it from spreading further and causing more damage. In the AWS cloud environment, it is important to remove the affected instance from the network. However, because forensic analysis is often performed in the cloud environment, completely detaching the affected EC2 instance from the network would prevent the investigators from accessing the evidentiary data. Hence, the forensic team should do the following:

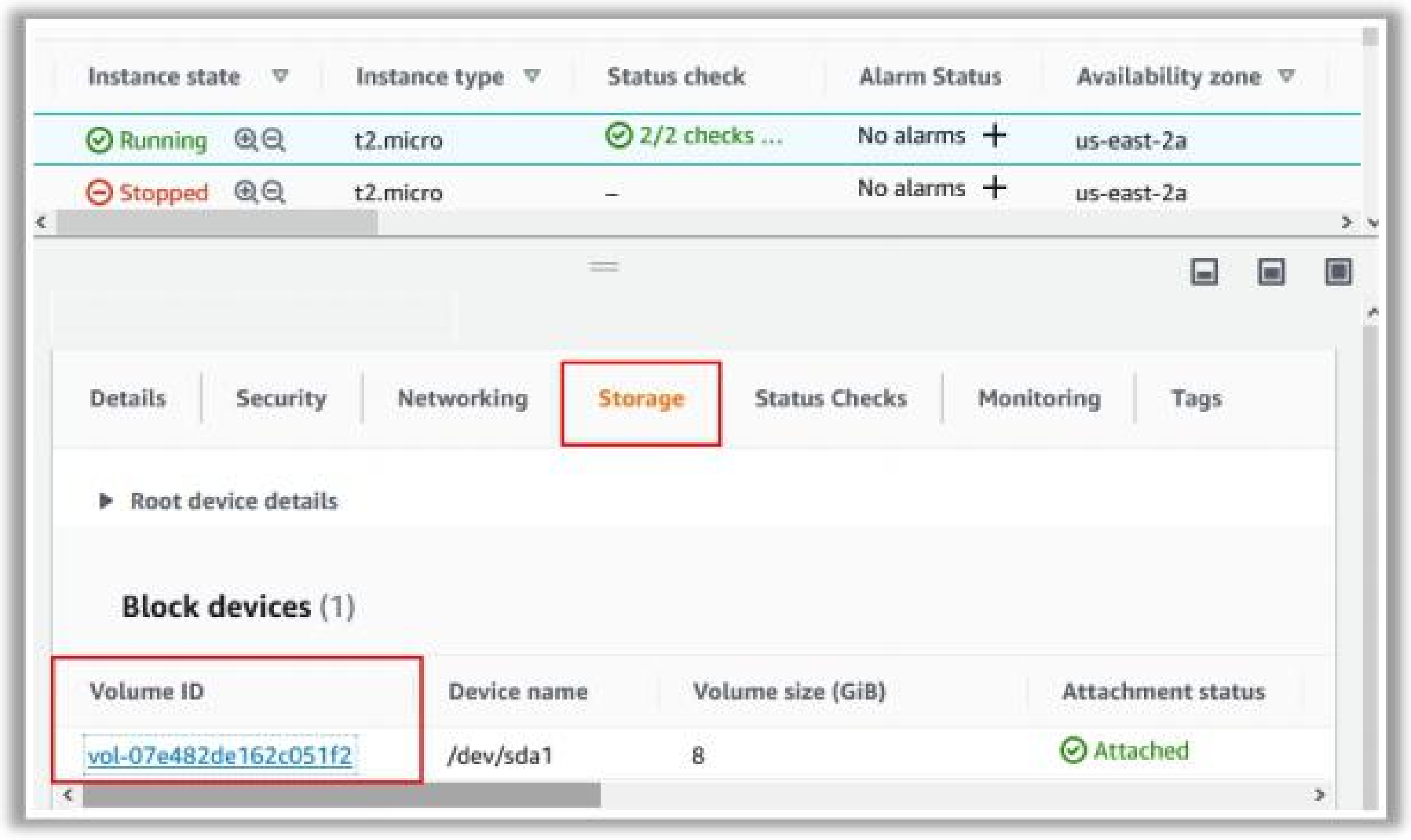
- Create a restrictive security group that denies all outbound traffic
- Make ingress rules that allow RDP or SSH traffic from only one secure IP address, which can be used later to connect to the instance
- Attach the newly configured security group immediately to the instance

Step 2: Take a Snapshot of the EC2 Instance

- Amazon EC2 instances use **EBS** volumes that act like virtual hard drives
- In the event of a security incident, investigators must take an **offline snapshot** of the **EBS volume** from the affected EC2 instance to acquire forensic evidence

Taking Snapshot of EBS volume from the Compromised EC2 Instance

- Stop the affected instance
- Select the affected instance and go to the **Storage** tab
- Locate the **volume ID** of the instance and click on it



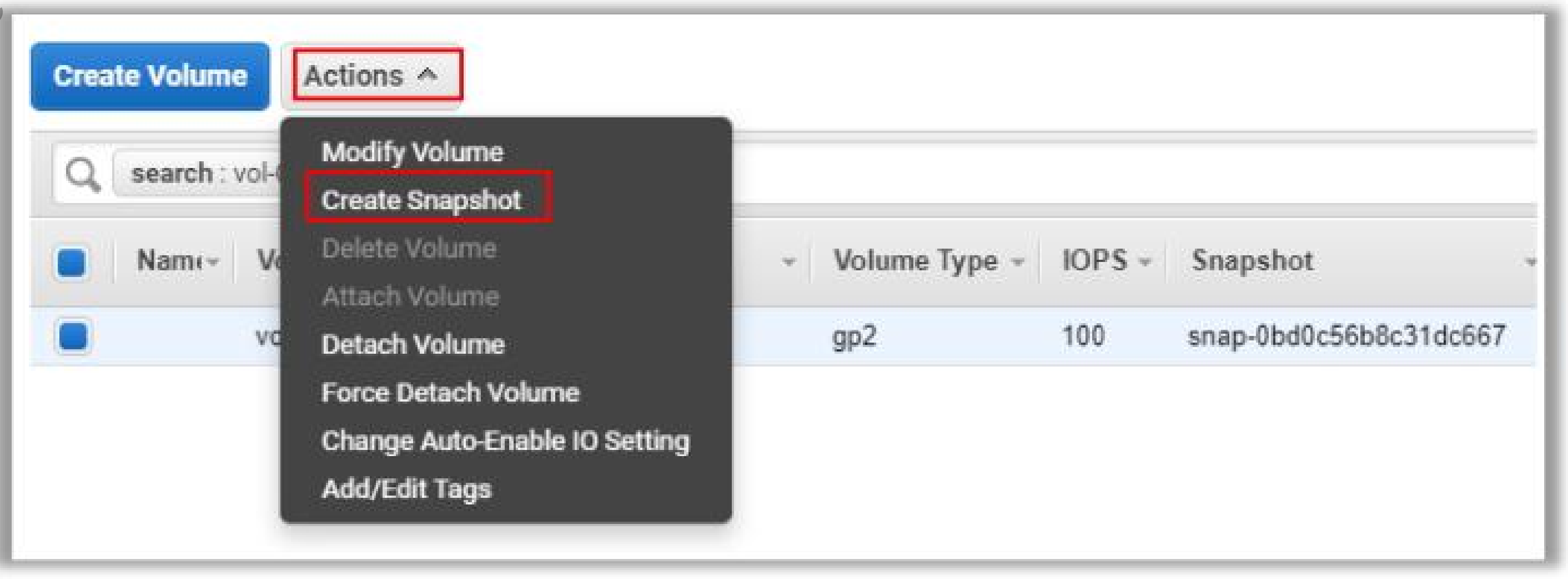
Instance state	Instance type	Status check	Alarm Status	Availability zone
Running	t2.micro	2/2 checks ...	No alarms +	us-east-2a
Stopped	t2.micro	–	No alarms +	us-east-2a

Block devices (1)

Volume ID	Device name	Volume size (GiB)	Attachment status
vol-07e482de162c051f2	/dev/sda1	8	Attached

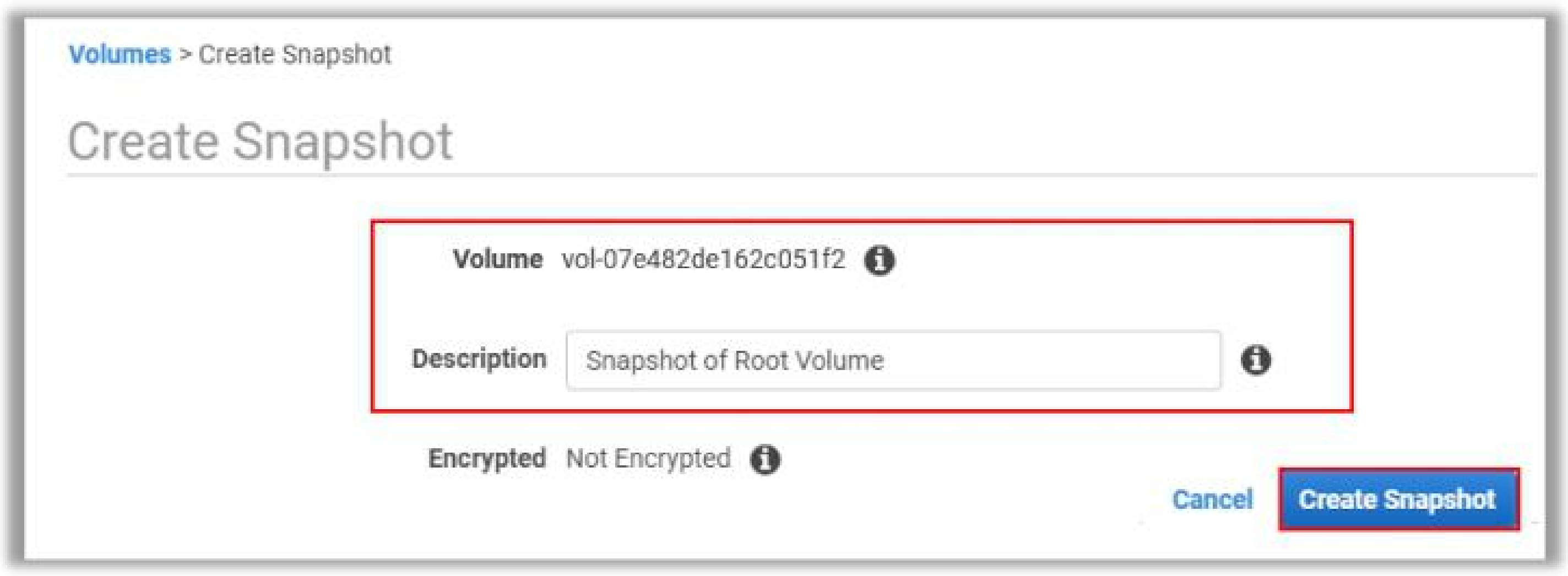
Step 2: Take a Snapshot of the EC2 Instance (Cont'd)

- Clicking on the **volume ID** will redirect you to the **Volumes** page where it will be already selected
- Click on the **Actions** button and select the **Create Snapshot** option from the drop-down menu



Name	V	Volume Type	IOPS	Snapshot
	vo	gp2	100	snap-0bd0c56b8c31dc667

- On the next page, add a description and click on the **Create Snapshot** button
- The **snapshot** of the **EBS volume** is now successfully created
- Take back up of the necessary data and **terminate** the affected EC2 instance



Volumes > Create Snapshot

Create Snapshot

Volume vol-07e482de162c051f2

Description Snapshot of Root Volume

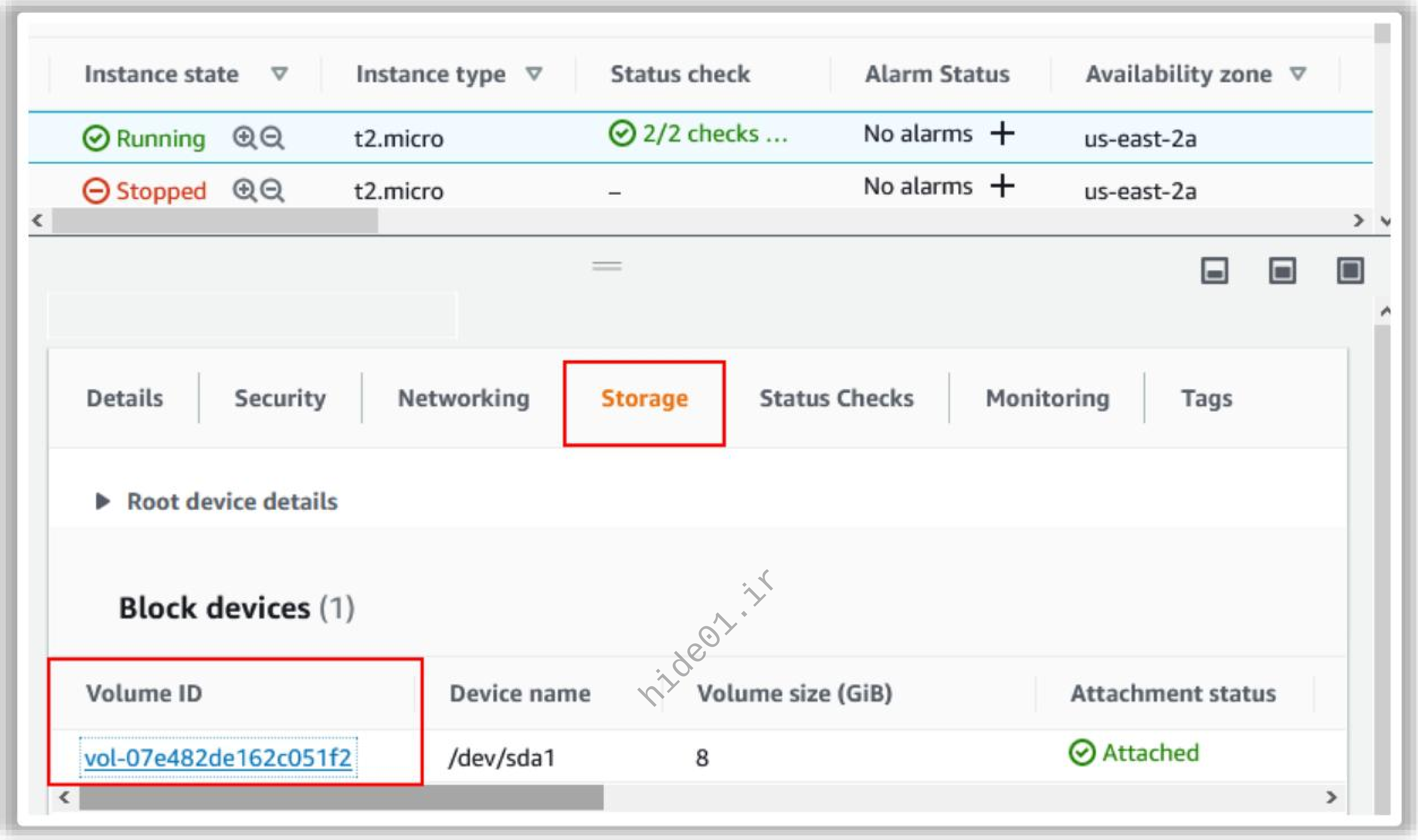
Encrypted Not Encrypted

Step 2: Take a Snapshot of the EC2 Instance

The concept of acquiring forensic evidence from the EBS volume of an affected EC2 instance for forensic examination is similar to that of creating an image file of the physical hard drive of any affected system and mounting it to the forensic workstation. However, the execution processes for on-premise and cloud devices are different. In case of a compromised EC2 instance, investigators first need to stop the instance and take an offline snapshot of the EBS volume. If the incident response team is doing this, they can share the snapshot with the forensic team for further investigation. Once it is created, the affected instance should be terminated immediately.

Following are the steps of taking a snapshot of the EBS volume of the affected EC2 instance on AWS management console:

- Stop the affected instance
- Make sure that the affected instance is selected. Now, go to the Storage tab which can be found on EC2 Management console page where all the deployed instances are listed
- Locate the volume ID of the instance under Block Devices tab and click on it



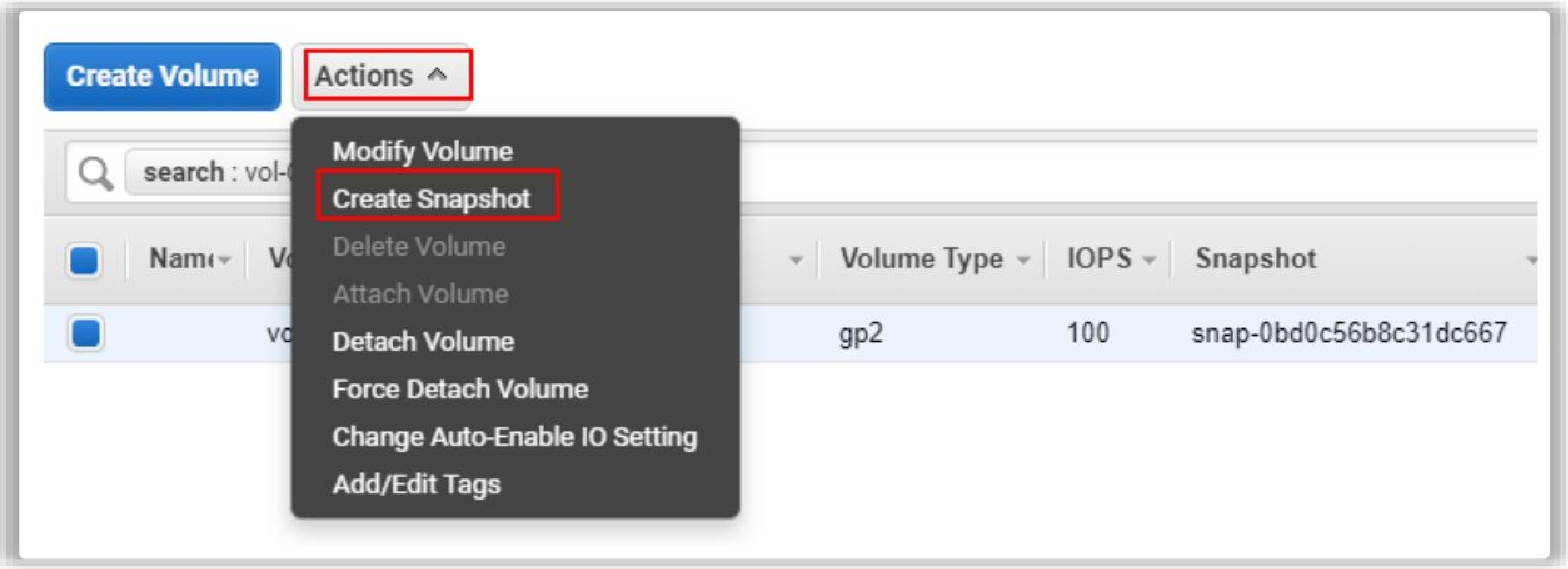
Instance state	Instance type	Status check	Alarm Status	Availability zone
Running	t2.micro	2/2 checks ...	No alarms +	us-east-2a
Stopped	t2.micro	–	No alarms +	us-east-2a

Block devices (1)

Volume ID	Device name	Volume size (GiB)	Attachment status
vol-07e482de162c051f2	/dev/sda1	8	Attached

Figure 8.2: Locating Volume ID on Storage Tab on EC2 Management Console

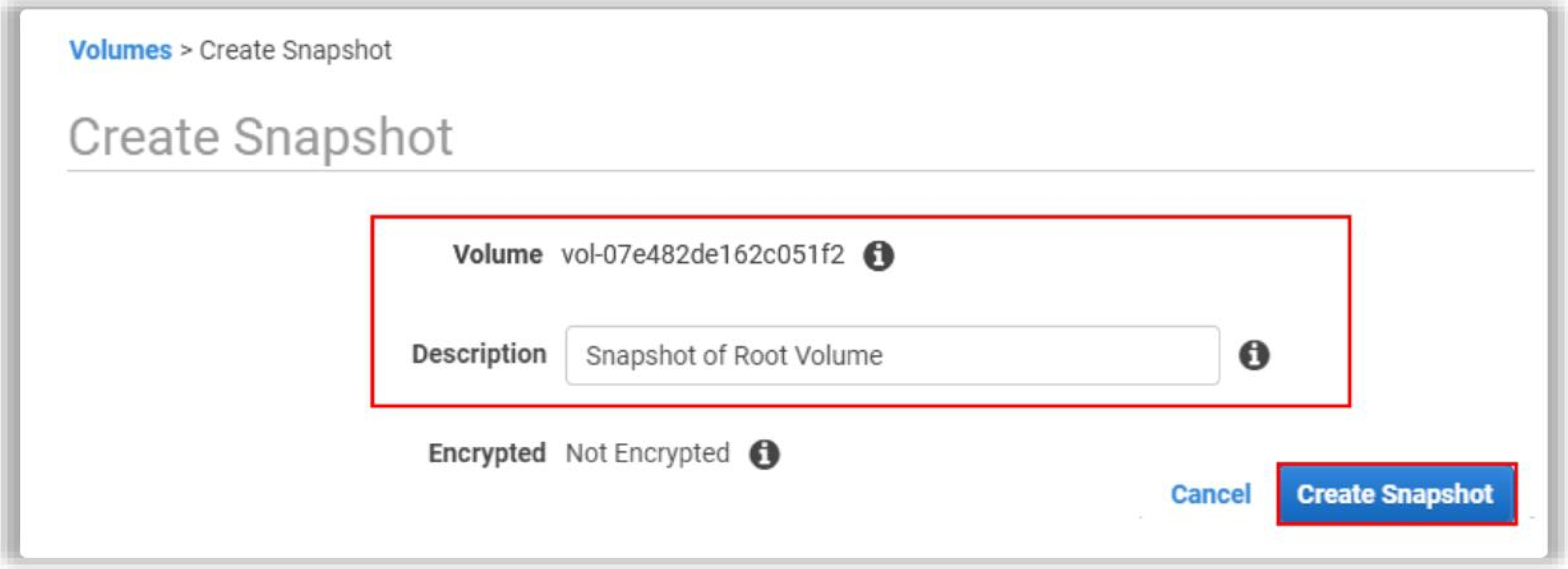
- Clicking on the volume ID will redirect you to the Volumes page where it will be already selected
- Click on the Actions button and select the Create Snapshot option from the drop-down menu



Name	V	Volume Type	IOPS	Snapshot
	vo	gp2	100	snap-0bd0c56b8c31dc667

Figure 8.3: Create Snapshot Option

- On the next page, add a description and click on the Create Snapshot button
- The snapshot of the EBS volume is now successfully created
- Take back up of the necessary data and terminate the affected EC2 instance



Volumes > Create Snapshot

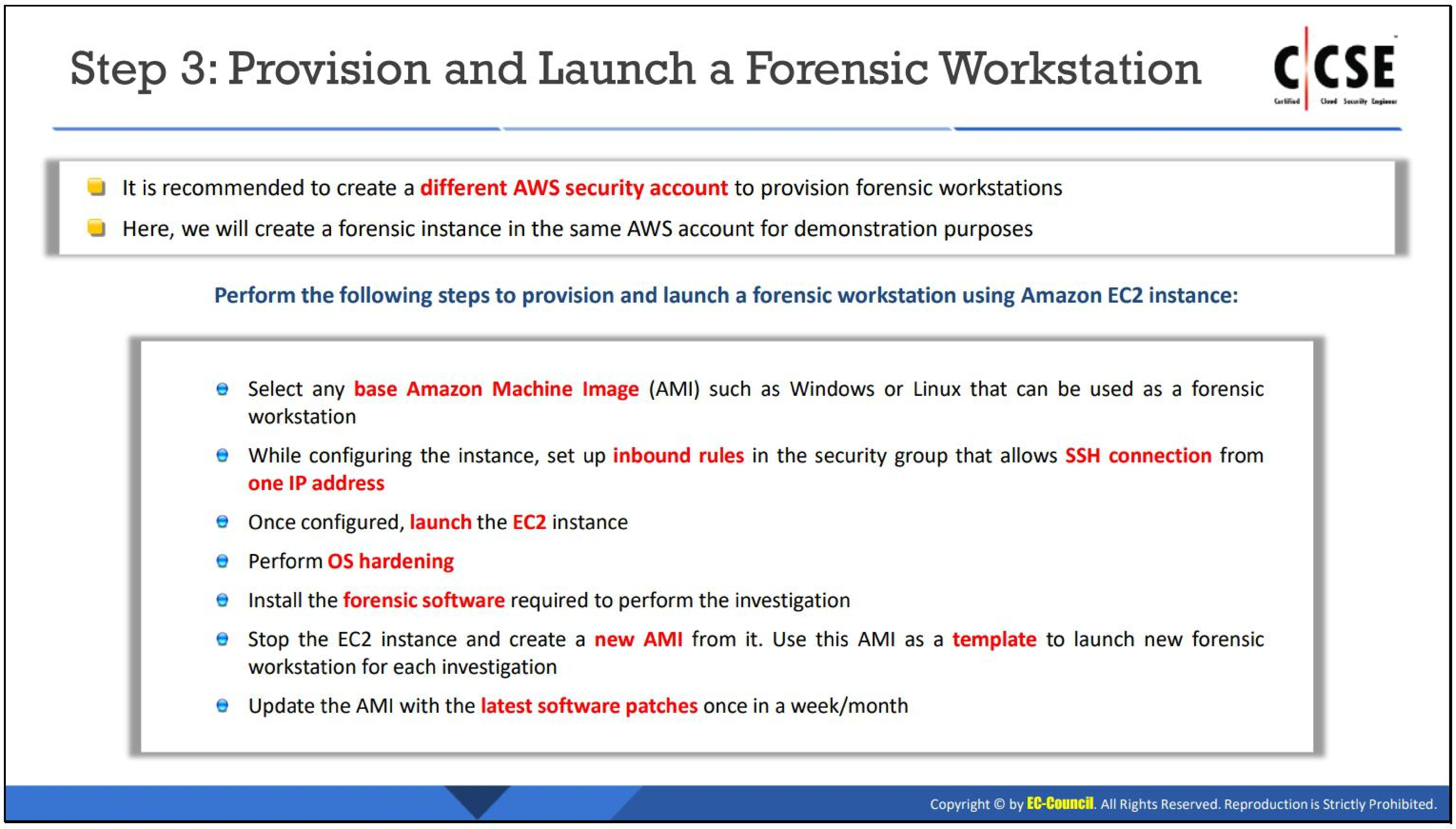
Create Snapshot

Volume vol-07e482de162c051f2

Description Snapshot of Root Volume

Encrypted Not Encrypted

Figure 8.4: Creating Snapshot of the EBS Volume



Step 3: Provision and Launch a Forensic Workstation

- It is recommended to create a **different AWS security account** to provision forensic workstations
- Here, we will create a forensic instance in the same AWS account for demonstration purposes

Perform the following steps to provision and launch a forensic workstation using Amazon EC2 instance:

- Select any **base Amazon Machine Image** (AMI) such as Windows or Linux that can be used as a forensic workstation
- While configuring the instance, set up **inbound rules** in the security group that allows **SSH connection** from **one IP address**
- Once configured, **launch** the **EC2** instance
- Perform **OS hardening**
- Install the **forensic software** required to perform the investigation
- Stop the EC2 instance and create a **new AMI** from it. Use this AMI as a **template** to launch new forensic workstation for each investigation
- Update the AMI with the **latest software patches** once in a week/month

Step 3: Provision and Launch a Forensic Workstation

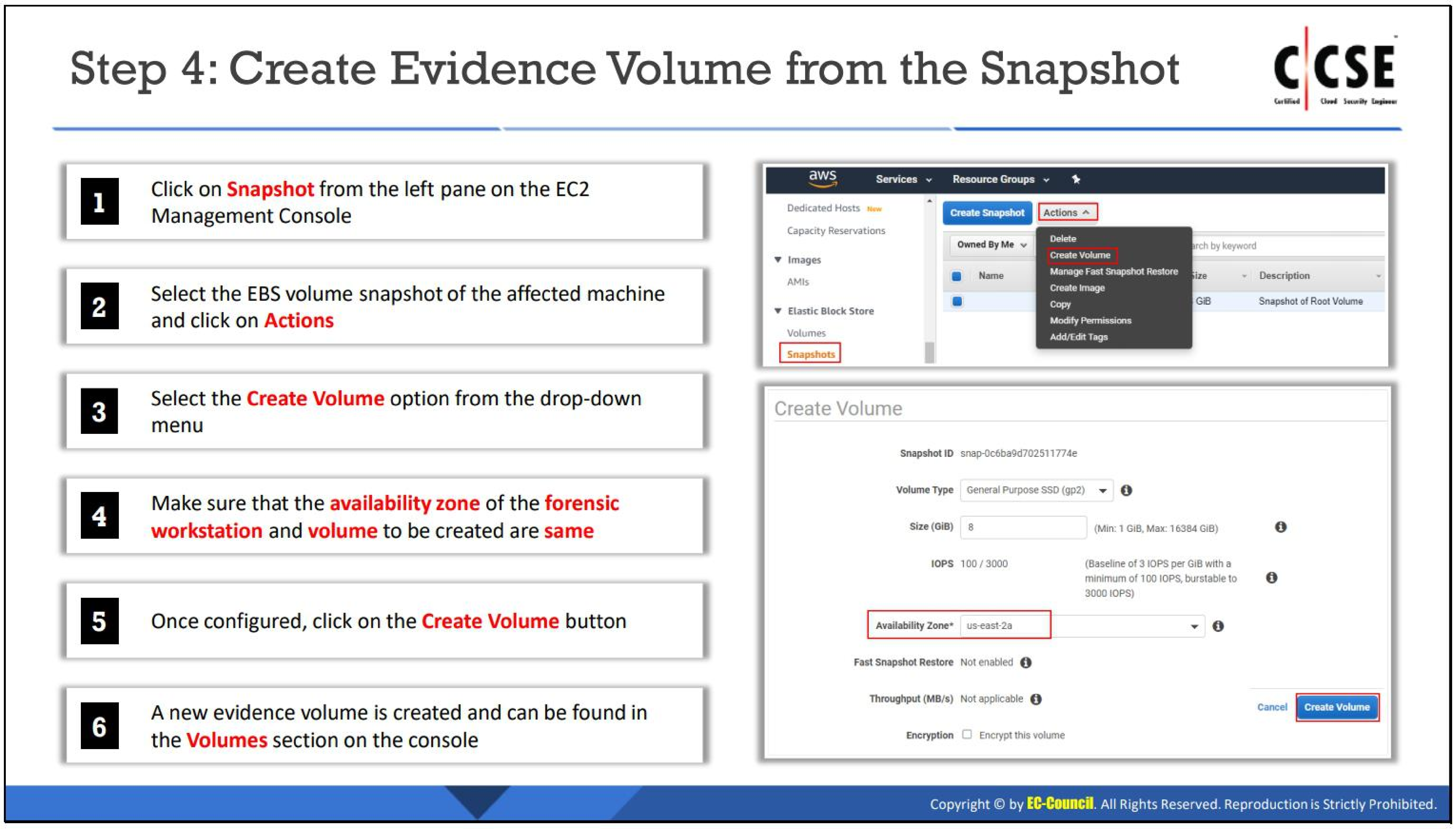
While forensic analysis can be performed with on-premise forensic workstations as well, better results can be achieved if it is executed within the cloud environment. Investigation within the cloud environment, where the data are located, is often cited as the best practice, instead of copying the data into other storage media. Organizations are recommended to maintain two separate AWS accounts, one for production and another for security operations. This often helps in satisfying the legal and compliance requirements.

It is essential to abide by the data privacy laws while provisioning any forensic instance and sending it the evidence data for analysis. To avoid any legal implications of moving data between two AWS regions, organizations can choose to create the forensic workstations in the same region where the security incident occurred. Investigators need to perform the following steps to provision and launch a forensic workstation using Amazon EC2 instance:

- Select any base Amazon Machine Image (AMI) such as Windows or Linux that can be used as a forensic workstation
- While configuring the instance, set up inbound rules in the security group that allows SSH connection from one IP address
- Once configured, launch the EC2 instance
- Perform OS hardening
- Install the forensic software required to perform the investigation
- Stop the EC2 instance and create a new AMI from it. Use this AMI as a template to launch new forensic workstation for each investigation

- Update the AMI with the latest software patches once in a week/month

Note: We will create a forensic instance in the same AWS account for demonstration purposes.



Step 4: Create Evidence Volume from the Snapshot

1. Click on **Snapshot** from the left pane on the EC2 Management Console
2. Select the EBS volume snapshot of the affected machine and click on **Actions**
3. Select the **Create Volume** option from the drop-down menu
4. Make sure that the **availability zone** of the **forensic workstation** and **volume** to be created are **same**
5. Once configured, click on the **Create Volume** button
6. A new evidence volume is created and can be found in the **Volumes** section on the console

Create Volume

Snapshot ID snap-0c6ba9d702511774e

Volume Type General Purpose SSD (gp2)

Size (GiB) 8 (Min: 1 GiB, Max: 16384 GiB)

IOPS 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)

Availability Zone* us-east-2a

Fast Snapshot Restore Not enabled

Throughput (MB/s) Not applicable

Encryption □ Encrypt this volume

Step 4: Create Evidence Volume from the Snapshot

Following are the steps to create an evidence volume from the snapshot of the affected EC2 instance:

1. Click on Snapshot from the left pane on the EC2 Management Console
2. Select the EBS volume snapshot of the affected machine and click on Actions
3. Select the Create Volume option from the drop-down menu
4. Make sure that the Availability Zone of the forensic workstation and volume to be created are same
5. Once configured, click on the Create Volume button
6. A new evidence volume is created and can be found in the Volumes section on the console

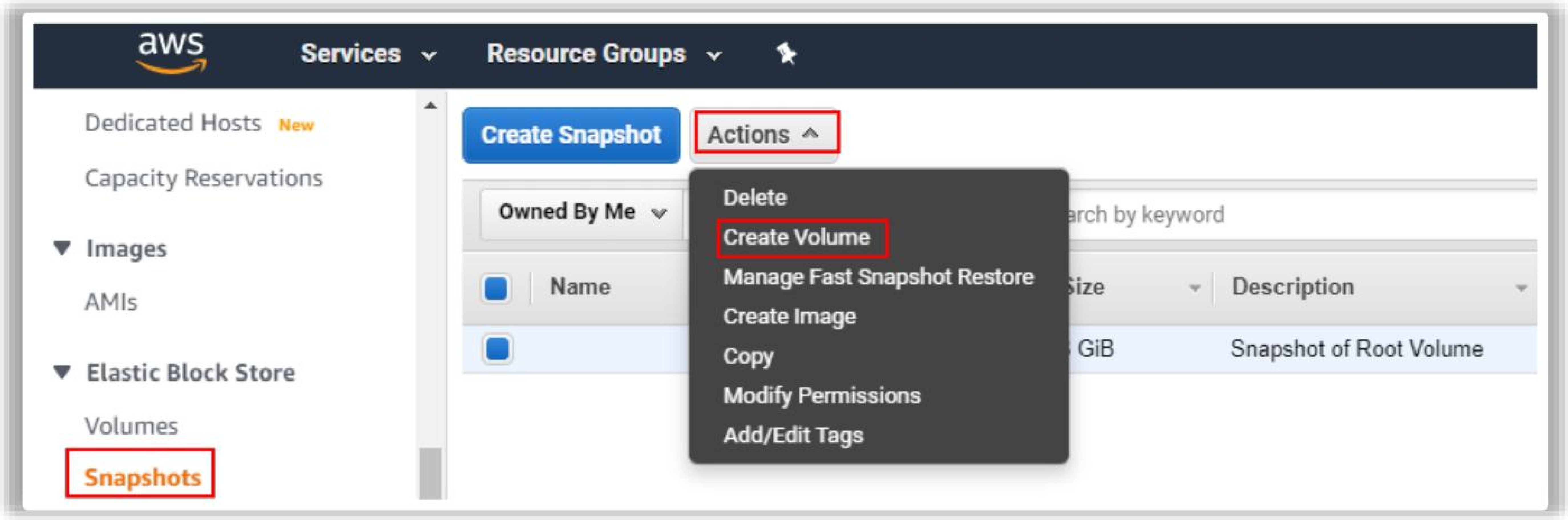
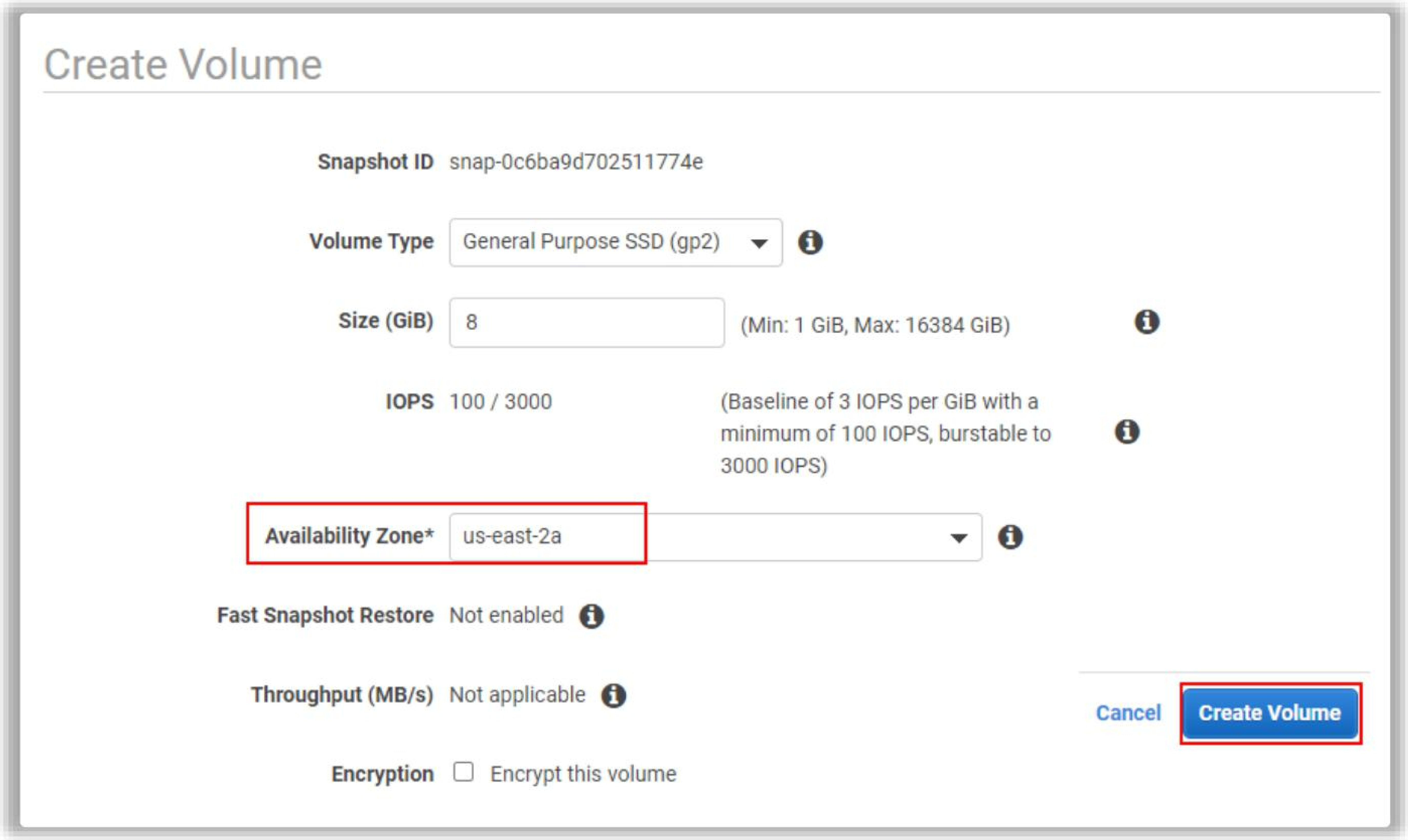


Figure 8.5: Selecting Create Volume Option on the Snapshots Page



Create Volume

Snapshot ID: snap-0c6ba9d702511774e

Volume Type: General Purpose SSD (gp2)

Size (GiB): 8 (Min: 1 GiB, Max: 16384 GiB)

IOPS: 100 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)

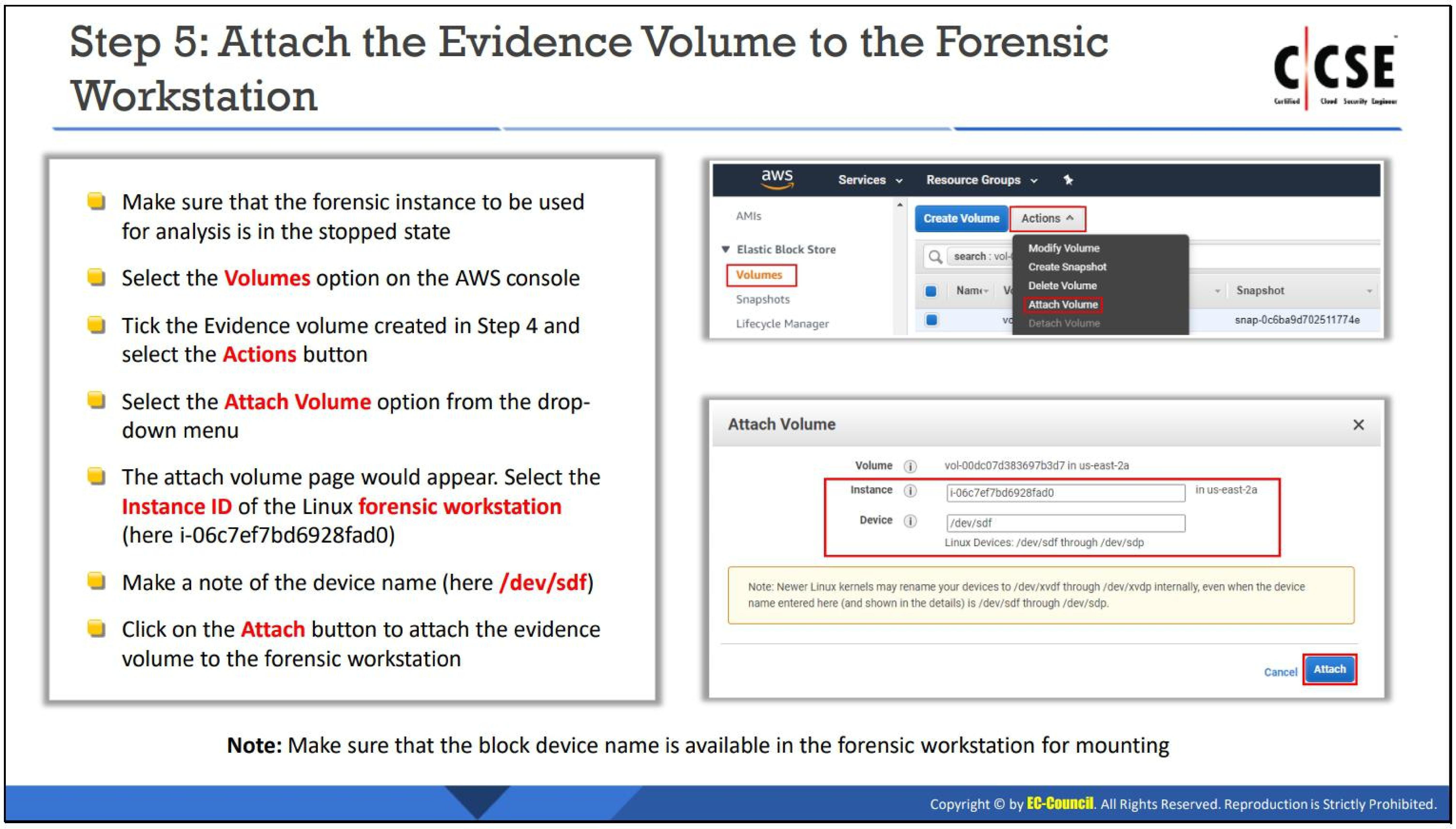
Availability Zone*: us-east-2a

Fast Snapshot Restore: Not enabled

Throughput (MB/s): Not applicable

Encryption: ☐ Encrypt this volume

Figure 8.6: Configuring Fields on the Create Volume Page



Step 5: Attach the Evidence Volume to the Forensic Workstation

- Make sure that the forensic instance to be used for analysis is in the stopped state
- Select the **Volumes** option on the AWS console
- Tick the Evidence volume created in Step 4 and select the **Actions** button
- Select the **Attach Volume** option from the drop-down menu
- The attach volume page would appear. Select the **Instance ID** of the Linux **forensic workstation** (here i-06c7ef7bd6928fad0)
- Make a note of the device name (here **/dev/sdf**)
- Click on the **Attach** button to attach the evidence volume to the forensic workstation

Attach Volume

Volume: vol-00dc07d383697b3d7 in us-east-2a

Instance: i-06c7ef7bd6928fad0 in us-east-2a

Device: /dev/sdf

Linux Devices: /dev/sdf through /dev/sdp

Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Note: Make sure that the block device name is available in the forensic workstation for mounting

Step 5: Attach the Evidence Volume to the Forensic Workstation

Before attaching the evidence volume to the forensic instance, investigators must ensure that the forensic instance is in the terminated state. The process of attaching the evidence volume to an Ubuntu forensic instance can be performed entirely on the AWS management console.

Investigators can perform the following steps to attach the evidence volume to the forensic instance:

- Make sure that the forensic instance to be used for analysis is in stopped state
- Select the Volumes option on the EC2 Management Console
- Tick the Evidence volume created in Step 4 and select the Actions button
- Select the Attach Volume option from the drop-down menu

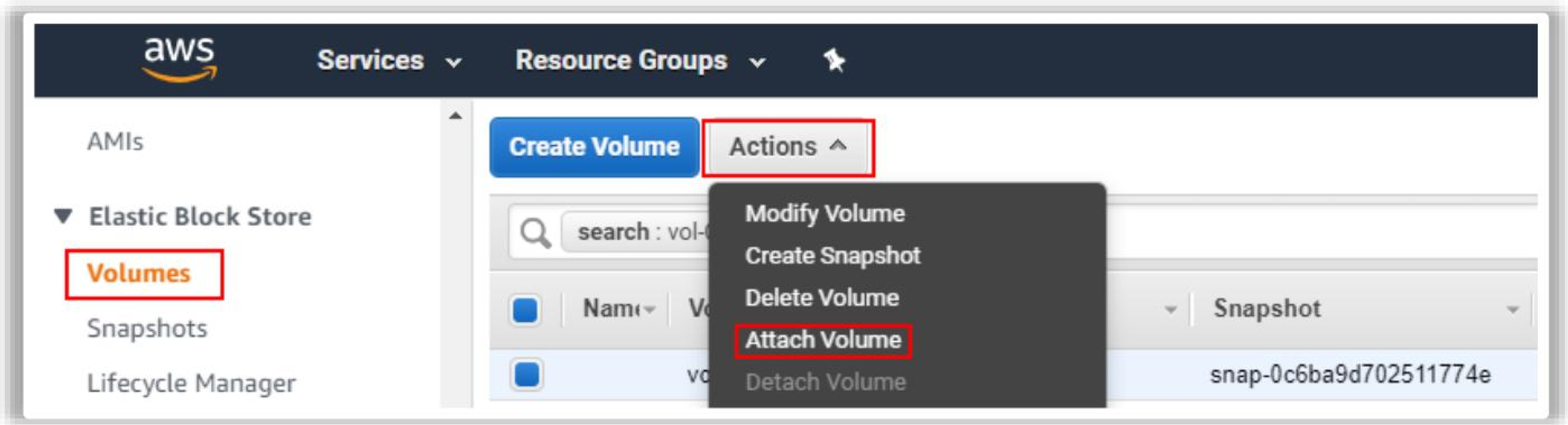
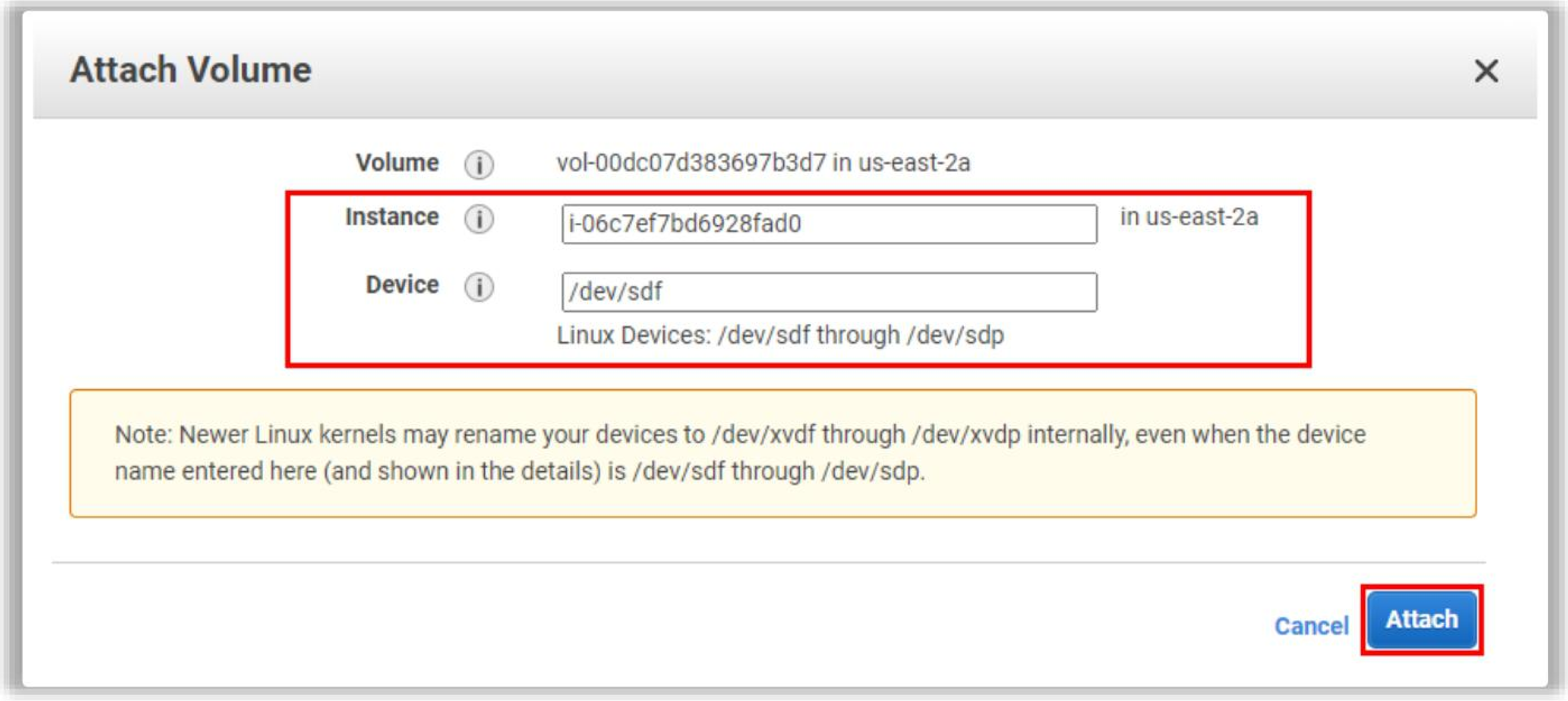


Figure 8.7: Selecting the Attach Volume Option on the Volumes Page

- The attach volume page would appear. Select the Instance ID of the Linux forensic workstation (here i-06c7ef7bd6928fad0)

- Make a note of the device name (here /dev/sdf)
- Click on the Attach button to attach the evidence volume to the forensic workstation



Volume: vol-00dc07d383697b3d7 in us-east-2a

Instance: i-06c7ef7bd6928fad0 in us-east-2a

Device: /dev/sdf

Linux Devices: /dev/sdf through /dev/sdp

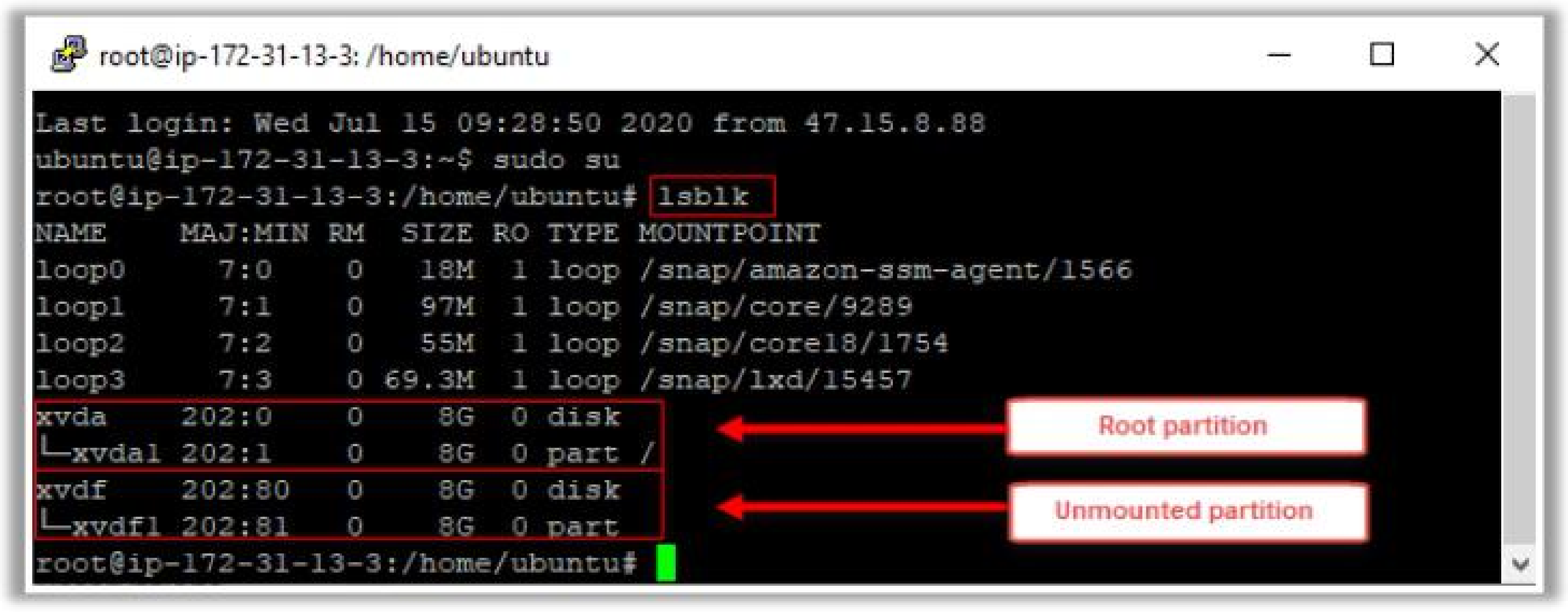
Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.

Figure 8.8: Attaching the Evidence Volume to the Forensic Instance

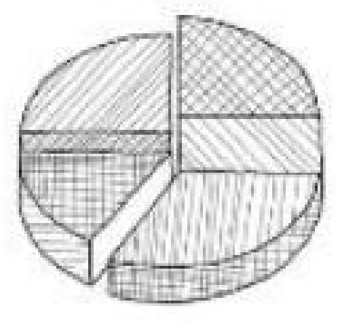
Note: Before attaching the evidence volume, investigators must check the number of partitions already present in the selected forensic instance (for example, Unix-based EC2 instances) and note down their names. While attaching the evidence volume, they must ensure that the name of the block device is available in the forensic workstation for mounting.

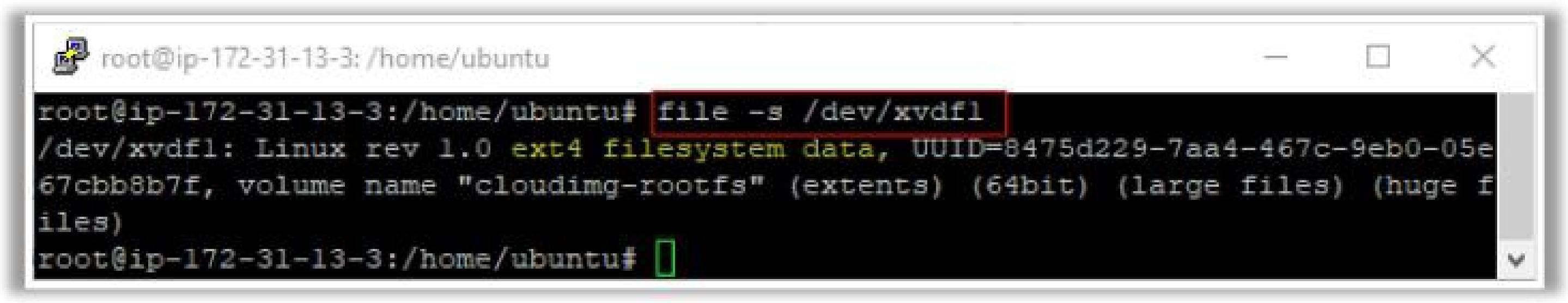
Step 6: Mount the Evidence Volume on the Forensic Workstation

- Start the forensic workstation (here, we are using an instance running on Linux)
- Run the **lsblk** command to verify whether the evidence volume was successfully attached
- The first screenshot here shows the root partition (xvda1) along with **another partition** on a new volume **(xvdf1)** which is attached but not mounted. This is the **evidence volume.**
- Run the **file** command to see the file format of the evidence volume
- The second screenshot shows that the evidence volume has Linux's **EXT4 filesystem**



```
Last login: Wed Jul 15 09:28:50 2020 from 47.15.8.88
ubuntu@ip-172-31-13-3:~$ sudo su
root@ip-172-31-13-3:/home/ubuntu# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0   18M  1 loop /snap/amazon-ssm-agent/1566
loop1     7:1    0   97M  1 loop /snap/core/9289
loop2     7:2    0   55M  1 loop /snap/core18/1754
loop3     7:3    0 69.3M  1 loop /snap/lxd/15457
xvda    202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdf    202:80   0    8G  0 disk
└─xvdf1 202:81   0    8G  0 part
root@ip-172-31-13-3:/home/ubuntu#
```



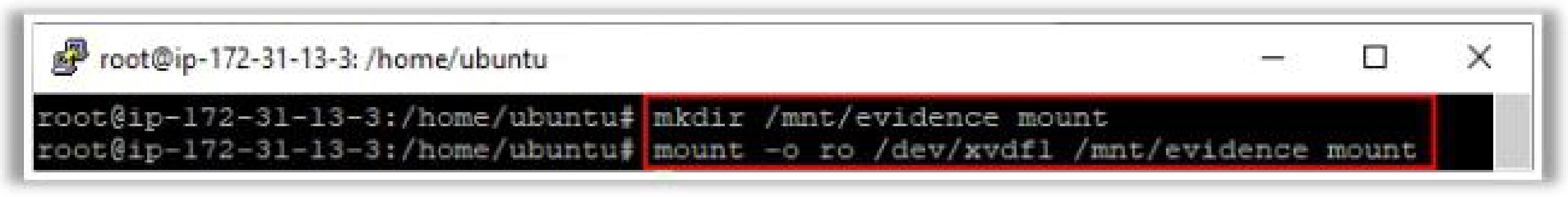


```
root@ip-172-31-13-3:/home/ubuntu# file -s /dev/xvdf1
/dev/xvdf1: Linux rev 1.0 ext4 filesystem data, UUID=8475d229-7aa4-467c-9eb0-05e
67cbb8b7f, volume name "cloudimg-rootfs" (extents) (64bit) (large files) (huge f
iles)
root@ip-172-31-13-3:/home/ubuntu#
```

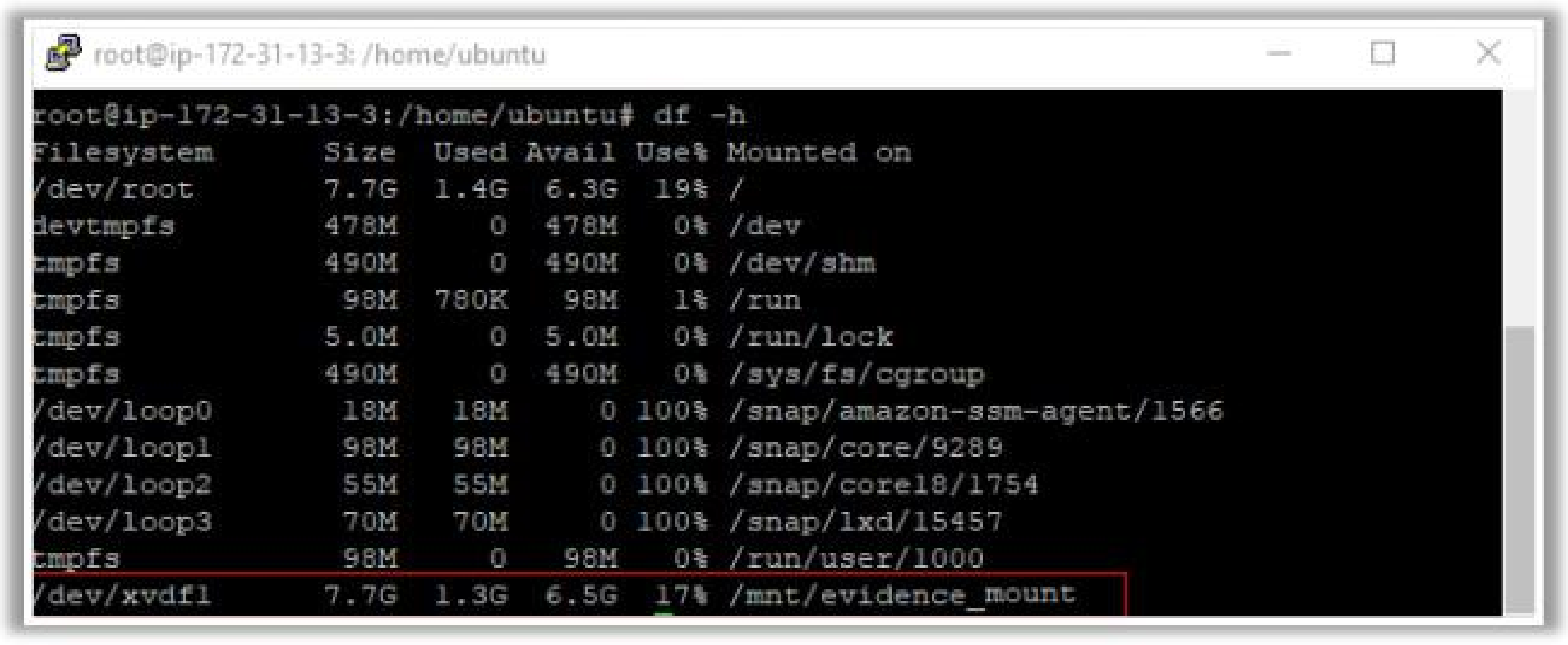
Note: Linux might rename the block devices as **/dev/xvdf** internally even though the device name is shown to be **/dev/sdf** while attaching it to an instance

Step 6: Mount the Evidence Volume on the Forensic Workstation (Cont'd)

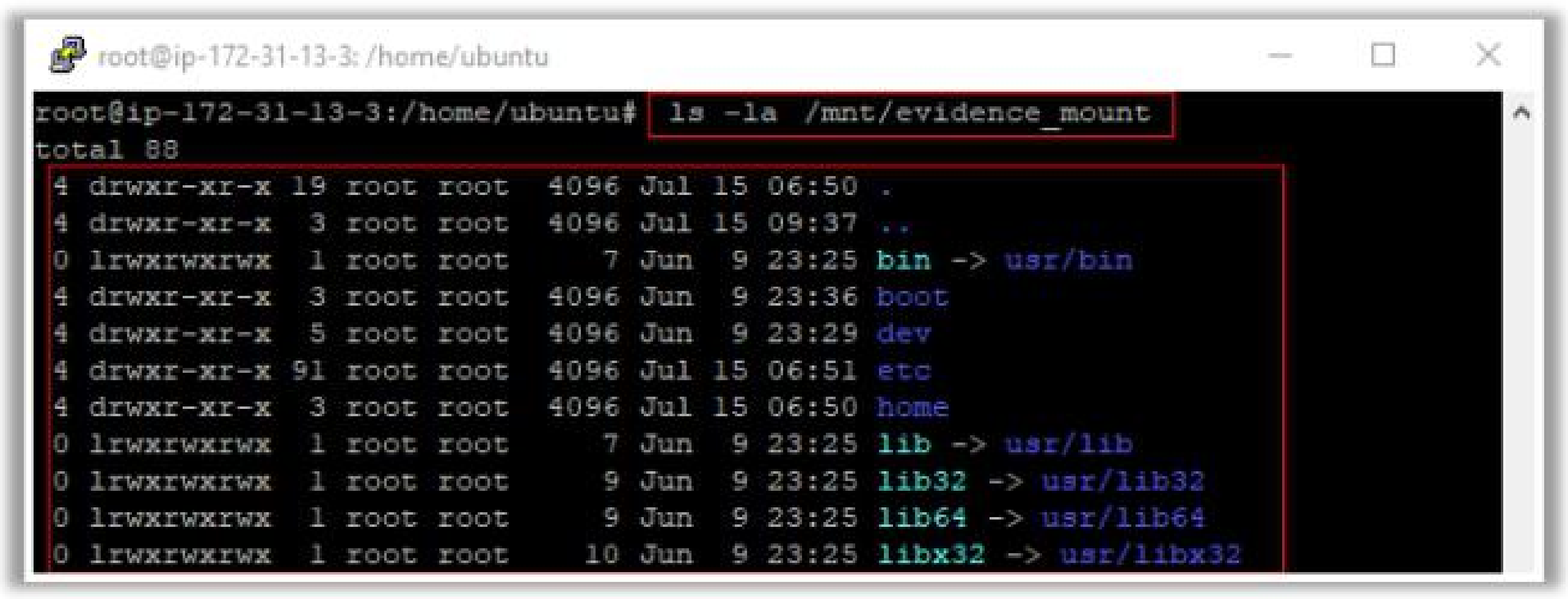
- Use the **mkdir** command to make a directory (here, /mnt/evidence_mount) and **mount** the evidence Linux file system in the read-only mode
- Verify whether the partition is mounted using the **df** command
- The second screenshot shows that **/dev/xvdf1** is mounted on **/mnt/evidence_mount**
- Get the full list of all files and directories stored in /dev/xvdf1 by using the **ls** command



```
root@ip-172-31-13-3:/home/ubuntu# mkdir /mnt/evidence mount
root@ip-172-31-13-3:/home/ubuntu# mount -o ro /dev/xvdf1 /mnt/evidence mount
```



```
root@ip-172-31-13-3:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.7G  1.4G  6.3G  19% /
devtmpfs        478M     0  478M   0% /dev
tmpfs           490M     0  490M   0% /dev/shm
tmpfs            98M  780K   98M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
/dev/loop0       18M   18M     0 100% /snap/amazon-ssm-agent/1566
/dev/loop1       98M   98M     0 100% /snap/core/9289
/dev/loop2       55M   55M     0 100% /snap/core18/1754
/dev/loop3       70M   70M     0 100% /snap/lxd/15457
tmpfs            98M     0   98M   0% /run/user/1000
/dev/xvdf1      7.7G  1.3G  6.5G  17% /mnt/evidence_mount
```



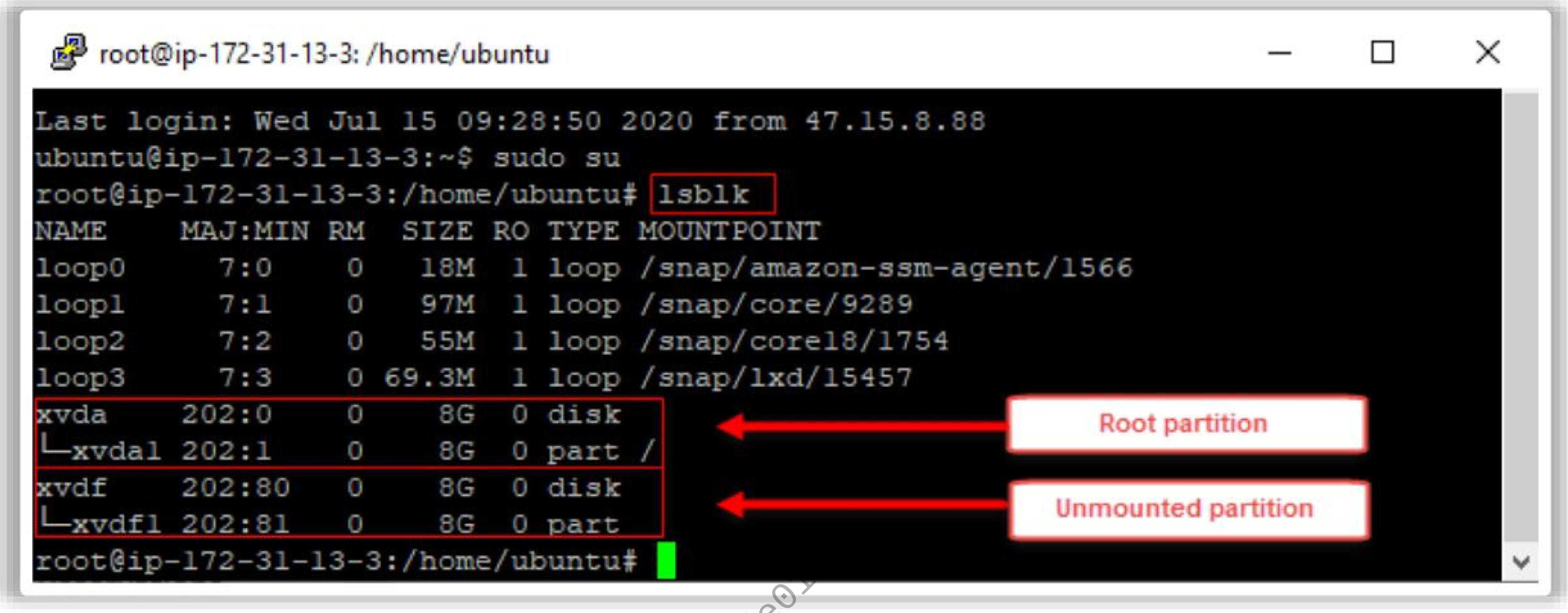
```
root@ip-172-31-13-3:/home/ubuntu# ls -la /mnt/evidence_mount
total 88
 4 drwxr-xr-x 19 root root  4096 Jul 15 06:50 .
 4 drwxr-xr-x  3 root root  4096 Jul 15 09:37 ..
 0 lrwxrwxrwx  1 root root     7 Jun  9 23:25 bin -> usr/bin
 4 drwxr-xr-x  3 root root  4096 Jun  9 23:36 boot
 4 drwxr-xr-x  5 root root  4096 Jun  9 23:29 dev
 4 drwxr-xr-x 91 root root  4096 Jul 15 06:51 etc
 4 drwxr-xr-x  3 root root  4096 Jul 15 06:50 home
 0 lrwxrwxrwx  1 root root     7 Jun  9 23:25 lib -> usr/lib
 0 lrwxrwxrwx  1 root root     9 Jun  9 23:25 lib32 -> usr/lib32
 0 lrwxrwxrwx  1 root root     9 Jun  9 23:25 lib64 -> usr/lib64
 0 lrwxrwxrwx  1 root root    10 Jun  9 23:25 libx32 -> usr/libx32
```

Step 6: Mount the Evidence Volume on the Forensic Workstation

Mounting the evidence volume on the forensic workstation is the last step of acquisition which allows investigators to get started with the investigation. It is possible to attach and mount the evidence volume as an additional volume on the forensic workstation. However, if required for investigation, they can detach the root volume of the forensic workstation and attach the evidence volume as root volume.

Investigators can perform the following steps to mount the evidence volume on the forensic instance:

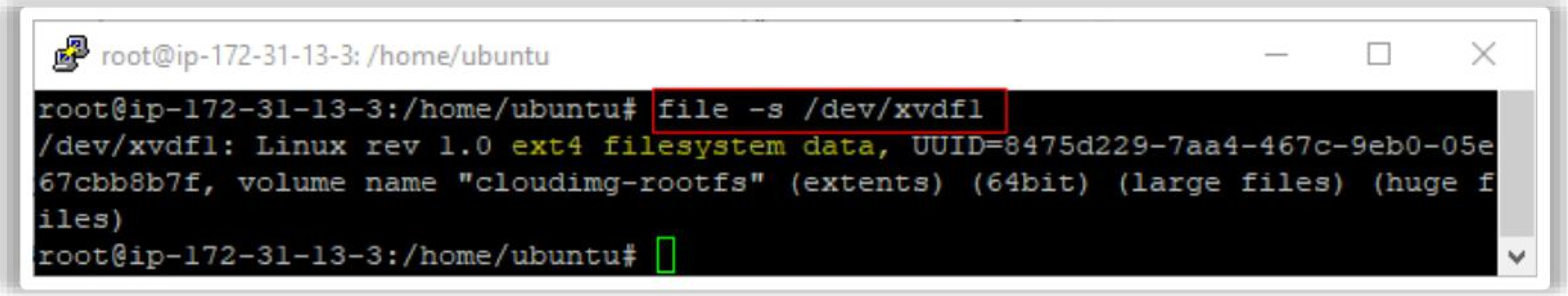
- Start the forensic workstation (here, we are using an instance running on Linux)
- Run the **`lsblk`** command to verify whether the evidence volume was successfully attached.
- The figure below shows the root partition (xvda1) of the forensic instance along with another partition on a new volume (xvdf1) which is attached but not mounted. This is the evidence volume.



```
Last login: Wed Jul 15 09:28:50 2020 from 47.15.8.88
ubuntu@ip-172-31-13-3:~$ sudo su
root@ip-172-31-13-3:/home/ubuntu# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0     7:0    0   18M  1 loop /snap/amazon-ssm-agent/1566
loop1     7:1    0   97M  1 loop /snap/core/9289
loop2     7:2    0   55M  1 loop /snap/core18/1754
loop3     7:3    0 69.3M  1 loop /snap/lxd/15457
xvda    202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdf    202:80   0    8G  0 disk
└─xvdf1 202:81   0    8G  0 part
root@ip-172-31-13-3:/home/ubuntu#
```

Figure 8.9: Running the lsblk Command

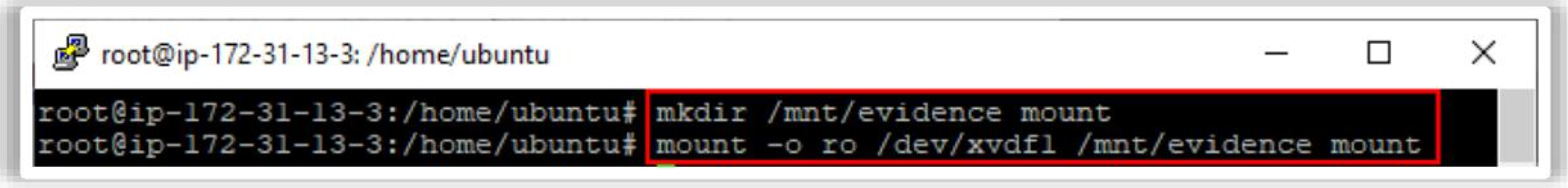
- Run the **`file`** command to see the file format of the evidence volume. The figure below shows that the evidence volume has Linux's EXT4 filesystem.



```
root@ip-172-31-13-3:/home/ubuntu# file -s /dev/xvdf1
/dev/xvdf1: Linux rev 1.0 ext4 filesystem data, UUID=8475d229-7aa4-467c-9eb0-05e
67cbb8b7f, volume name "cloudimg-rootfs" (extents) (64bit) (large files) (huge f
iles)
root@ip-172-31-13-3:/home/ubuntu#
```

Figure 8.10: Running the File Command

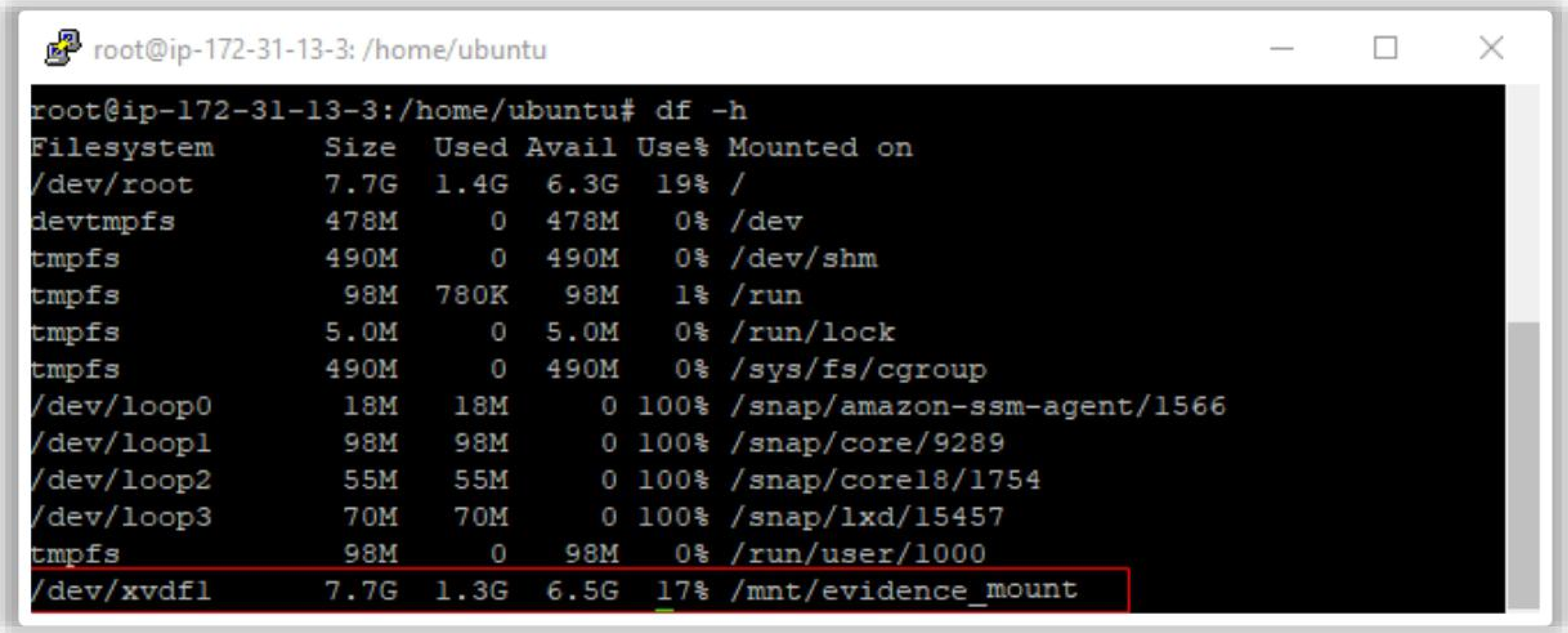
- Use the **`mkdir`** command to make a directory (here, /mnt/evidence_mount) and mount the evidence Linux file system in the read-only mode.



```
root@ip-172-31-13-3:/home/ubuntu# mkdir /mnt/evidence_mount
root@ip-172-31-13-3:/home/ubuntu# mount -o ro /dev/xvdf1 /mnt/evidence_mount
```

Figure 8.11: Making a Directory Named evidence_mount and Mounting the Evidence Volume

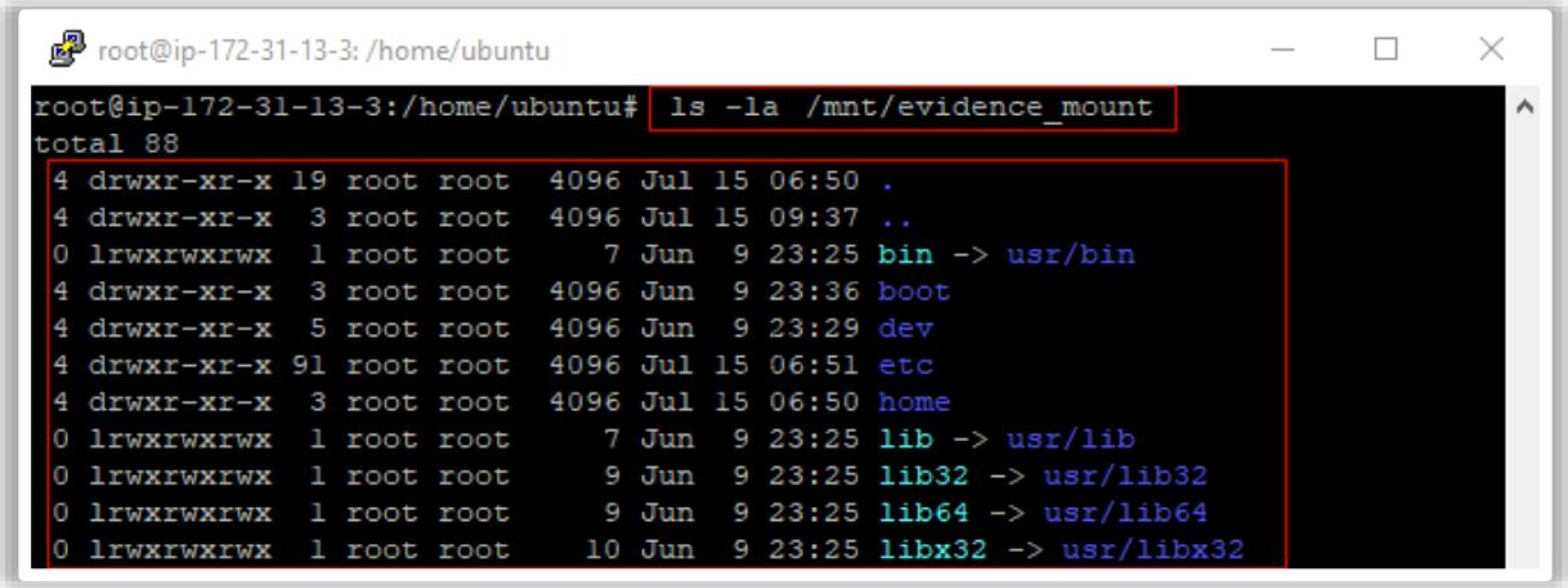
- Verify whether the partition is mounted using the **df** command. The figure below shows that /dev/xvdf1 is mounted on /mnt/evidence_mount.



```
root@ip-172-31-13-3:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       7.7G  1.4G  6.3G  19% /
devtmpfs        478M     0  478M   0% /dev
tmpfs           490M     0  490M   0% /dev/shm
tmpfs            98M  780K   98M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
/dev/loop0       18M   18M     0 100% /snap/amazon-ssm-agent/1566
/dev/loop1       98M   98M     0 100% /snap/core/9289
/dev/loop2       55M   55M     0 100% /snap/core18/1754
/dev/loop3       70M   70M     0 100% /snap/lxd/15457
tmpfs            98M     0   98M   0% /run/user/1000
/dev/xvdf1      7.7G  1.3G  6.5G  17% /mnt/evidence_mount
```

Figure 8.12: Evidence Volume is Mounted on /mnt/evidence_mount

- Get the full list of all files and directories stored in /dev/xvdf1 by using the **ls** command



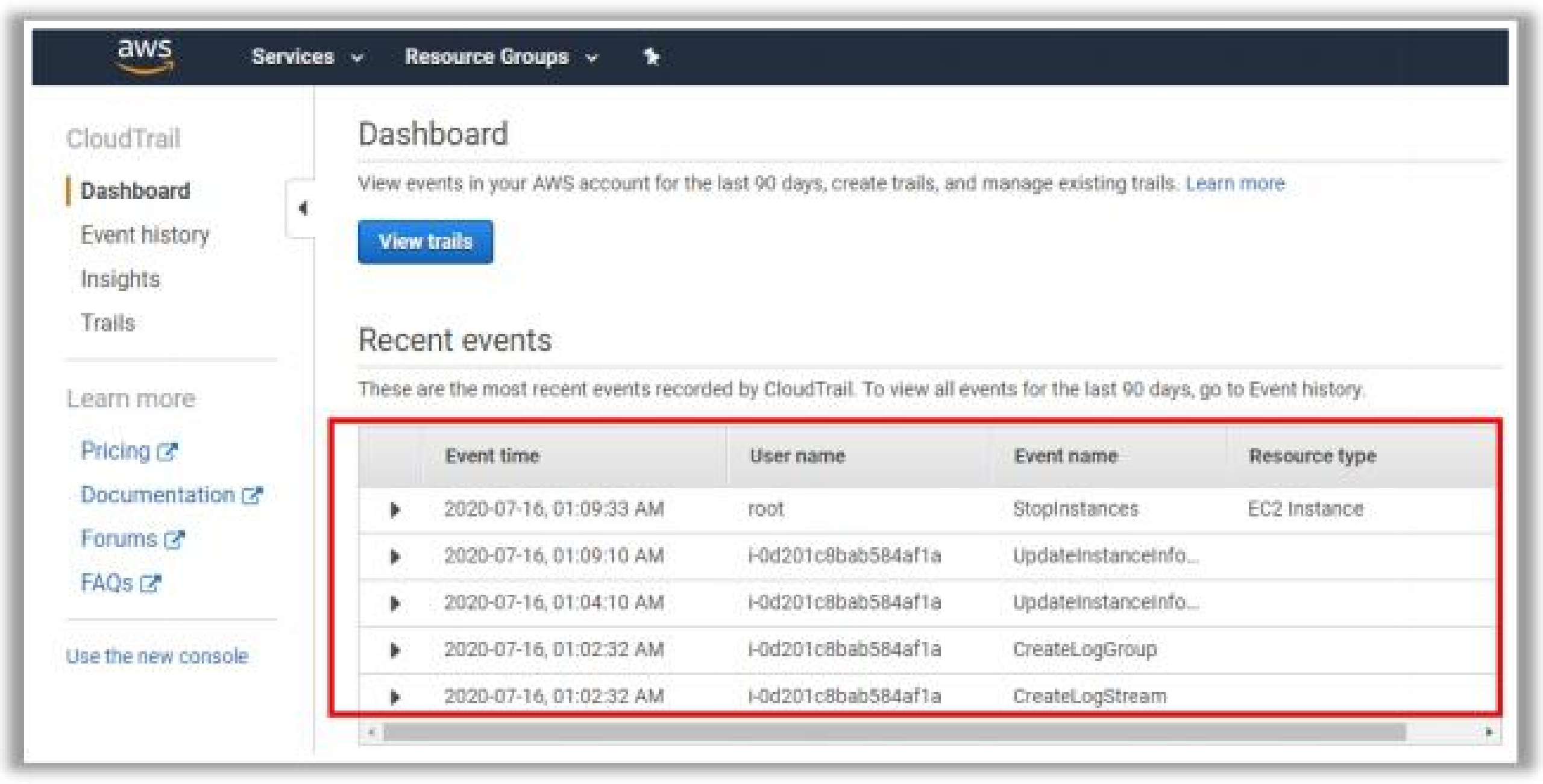
```
root@ip-172-31-13-3:/home/ubuntu# ls -la /mnt/evidence_mount
total 88
 4 drwxr-xr-x 19 root root  4096 Jul 15 06:50 .
 4 drwxr-xr-x  3 root root  4096 Jul 15 09:37 ..
 0 lrwxrwxrwx  1 root root     7 Jun  9 23:25 bin -> usr/bin
 4 drwxr-xr-x  3 root root  4096 Jun  9 23:36 boot
 4 drwxr-xr-x  5 root root  4096 Jun  9 23:29 dev
 4 drwxr-xr-x 91 root root  4096 Jul 15 06:51 etc
 4 drwxr-xr-x  3 root root  4096 Jul 15 06:50 home
 0 lrwxrwxrwx  1 root root     7 Jun  9 23:25 lib -> usr/lib
 0 lrwxrwxrwx  1 root root     9 Jun  9 23:25 lib32 -> usr/lib32
 0 lrwxrwxrwx  1 root root     9 Jun  9 23:25 lib64 -> usr/lib64
 0 lrwxrwxrwx  1 root root    10 Jun  9 23:25 libx32 -> usr/libx32
```

Figure 8.13: Examining the Mounted Volume

Logs in AWS

AWS CloudTrail

- AWS CloudTrail provides AWS **API call history for AWS accounts** including calls made via the AWS Management Console or Command Line tools, AWS Software Development Kits and other AWS services
- It **records and stores logs** that provide information on the event history related to activities in any AWS account for a period of 90 days
- API call history monitoring can help in **tracing suspicious user activities** within the AWS environment and detect source IP addresses associated with unusual API calls along with their time stamps
- Investigators can quickly track changes related to the creation, modification, and deletion of AWS resources by reviewing CloudTrail events
- **AWS CloudTrail events** can be viewed via AWS CloudTrail console which are stored in S3 buckets as log files, and delivered to Amazon CloudWatch



Dashboard

View events in your AWS account for the last 90 days, create trails, and manage existing trails. Learn more

Recent events

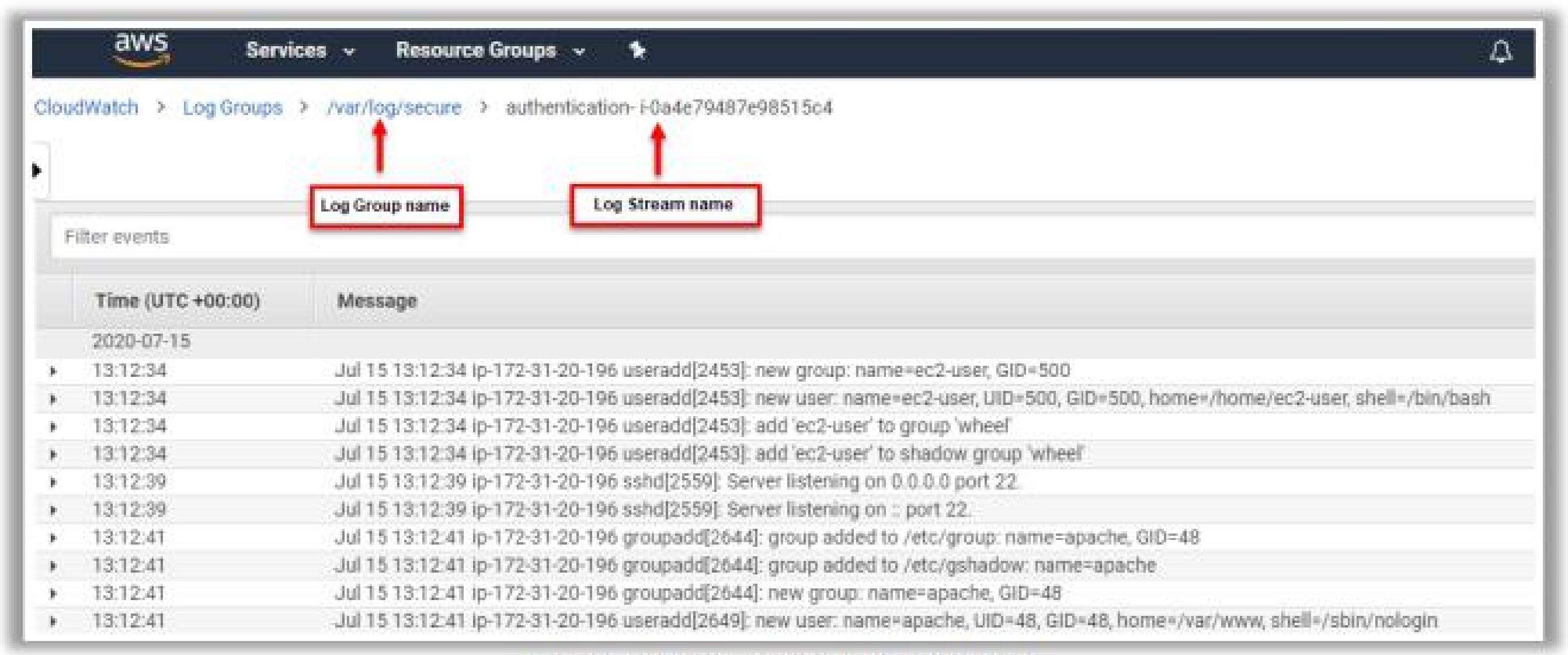
These are the most recent events recorded by CloudTrail. To view all events for the last 90 days, go to Event history.

Event time	User name	Event name	Resource type
2020-07-16, 01:09:33 AM	root	StopInstances	EC2 Instance
2020-07-16, 01:09:10 AM	i-0d201c8bab584af1a	UpdateInstanceInfo...	
2020-07-16, 01:04:10 AM	i-0d201c8bab584af1a	UpdateInstanceInfo...	
2020-07-16, 01:02:32 AM	i-0d201c8bab584af1a	CreateLogGroup	
2020-07-16, 01:02:32 AM	i-0d201c8bab584af1a	CreateLogStream	

Logs in AWS (Cont'd)

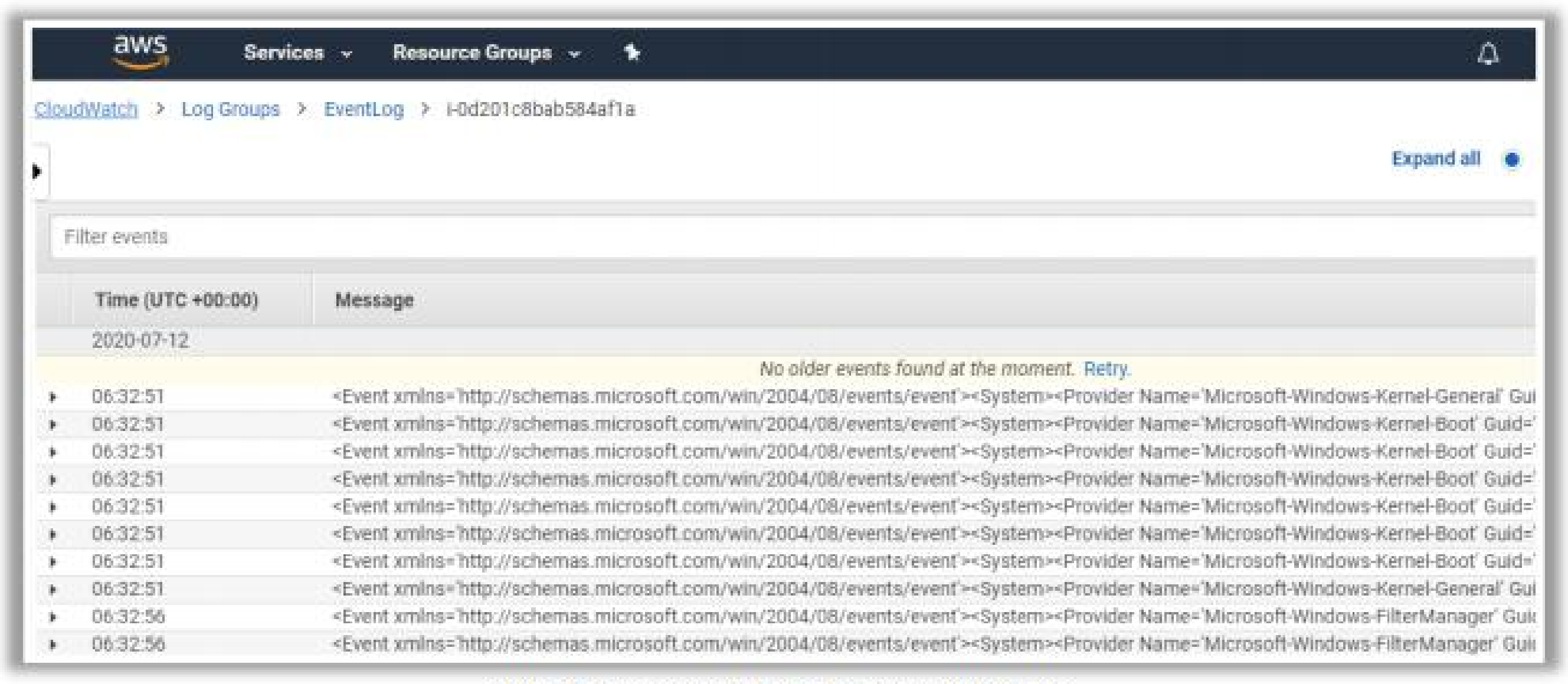
AWS CloudWatch

- Amazon **CloudWatch** allows the inspection, access and storage of log files from various AWS sources such as AWS CloudTrail, EC2 instances, and Route 53
- It **helps in collecting all log data** to a centralized location and analyzes them by performing custom search queries
- These logs can be **viewed as log streams** that capture a sequence of log events from the same instance or resource
- In Amazon CloudWatch, investigators can:
  - Examine the system and application-specific data from EC2 instances
  - Review unusual API activity as recorded by CloudTrail
  - Monitor DNS queries as received by Route 53



```
Time (UTC +00:00)   Message
2020-07-15
13:12:34   Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: new group: name=ec2-user, GID=500
13:12:34   Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: new user: name=ec2-user, UID=500, GID=500, home=/home/ec2-user, shell=/bin/bash
13:12:34   Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: add 'ec2-user' to group 'wheel'
13:12:34   Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: add 'ec2-user' to shadow group 'wheel'
13:12:39   Jul 15 13:12:39 ip-172-31-20-196 sshd[2559]: Server listening on 0.0.0.0 port 22.
13:12:39   Jul 15 13:12:39 ip-172-31-20-196 sshd[2559]: Server listening on :: port 22.
13:12:41   Jul 15 13:12:41 ip-172-31-20-196 groupadd[2644]: group added to /etc/group: name=apache, GID=48
13:12:41   Jul 15 13:12:41 ip-172-31-20-196 groupadd[2644]: group added to /etc/gshadow: name=apache
13:12:41   Jul 15 13:12:41 ip-172-31-20-196 groupadd[2644]: new group: name=apache, GID=48
13:12:41   Jul 15 13:12:41 ip-172-31-20-196 useradd[2649]: new user: name=apache, UID=48, GID=48, home=/var/www, shell=/sbin/nologin
```

Linux server logs in CloudWatch



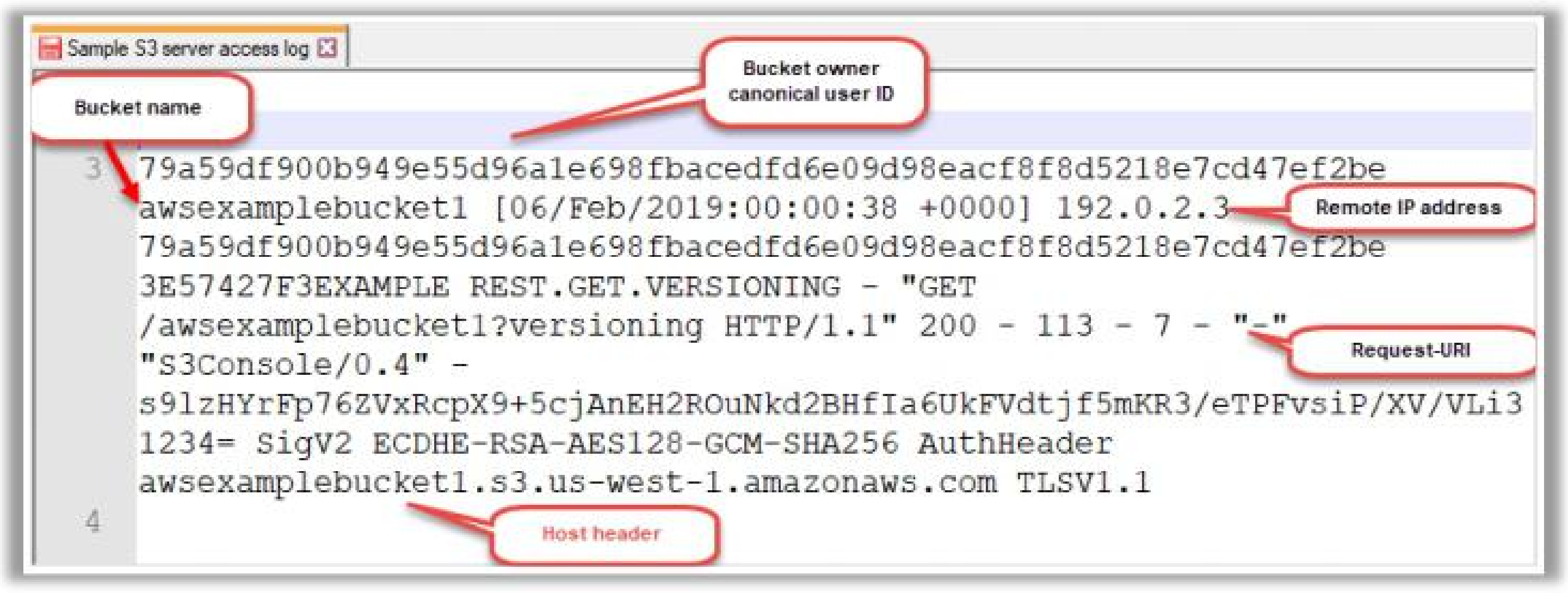
```
Time (UTC +00:00)   Message
2020-07-12
                    No older events found at the moment. Retry.
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-General' Gui
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-General' Gui
06:32:56   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-FilterManager' Gui
06:32:56   <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-FilterManager' Gui
```

Windows event logs in CloudWatch

Logs in AWS (Cont'd)

S3 Server Access Logs

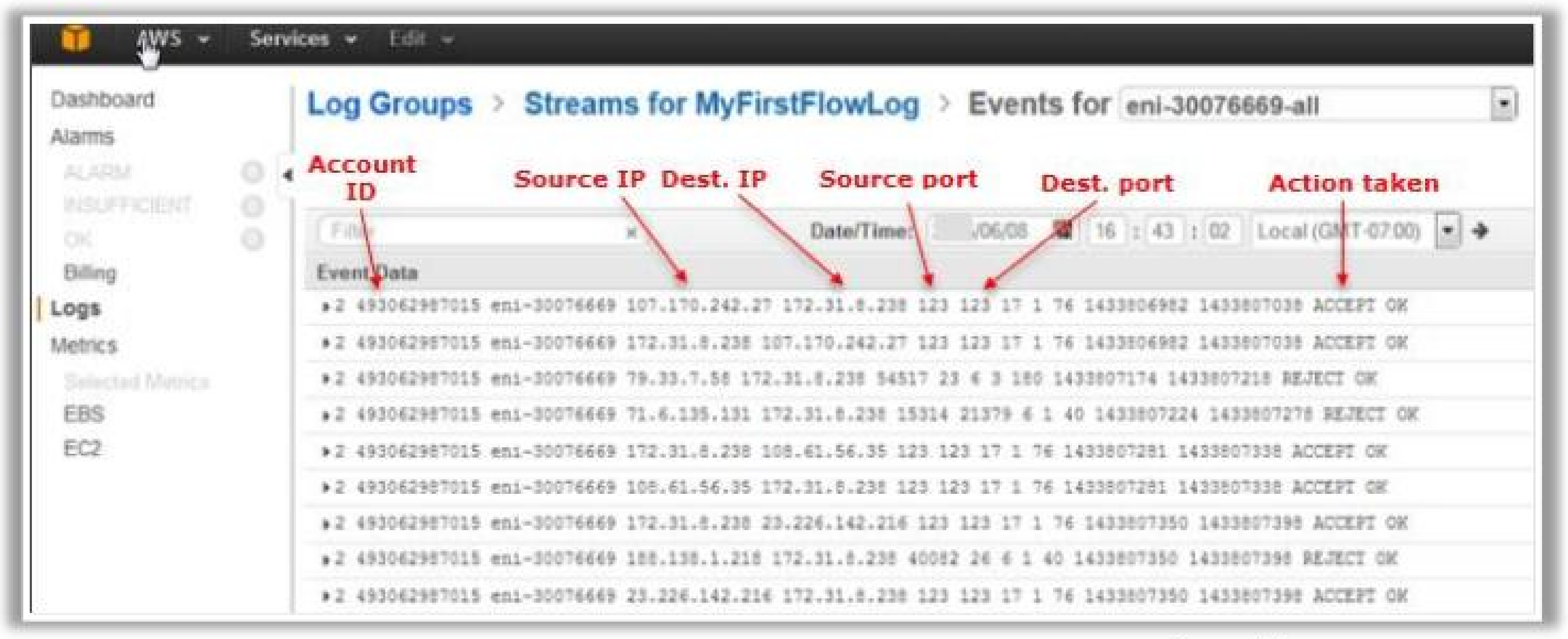
- When enabled, these logs record detailed information regarding the access requests made to any **S3 bucket**, including PUT, GET and DELETE actions
- The **log entries** contain various field, such as the bucket owner ID, bucket name, name and IP of the requester, request time and ID, Request-URI, and response status



```
3  79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
   awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
   79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
   3E57427F3EXAMPLE REST.GET.VERSIONING - "GET
   /awsexamplebucket1?versioning HTTP/1.1" 200 - 113 - 7 - "-"
   "S3Console/0.4" -
   s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi3
   1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader
   awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
4
```

VPC Flow Logs

- Flow logs record information regarding the **inbound and outbound IP traffic** on various network interfaces within a VPC environment
- The log entries contain the source and destination IP addresses and ports along with the timestamp and action performed



```
2 493062987015 eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 1433806982 1433807038 ACCEPT OK
2 493062987015 eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 1433806982 1433807038 ACCEPT OK
2 493062987015 eni-30076669 79.33.7.58 172.31.8.238 54517 23 6 3 180 1433807174 1433807218 REJECT OK
2 493062987015 eni-30076669 71.6.135.131 172.31.8.238 15314 21379 6 1 40 1433807224 1433807278 REJECT OK
2 493062987015 eni-30076669 172.31.8.238 108.61.56.35 123 123 17 1 76 1433807281 1433807338 ACCEPT OK
2 493062987015 eni-30076669 108.61.56.35 172.31.8.238 123 123 17 1 76 1433807281 1433807338 ACCEPT OK
2 493062987015 eni-30076669 172.31.8.238 23.226.142.216 123 123 17 1 76 1433807350 1433807398 ACCEPT OK
2 493062987015 eni-30076669 188.138.1.218 172.31.8.238 40082 26 6 1 40 1433807350 1433807398 REJECT OK
2 493062987015 eni-30076669 23.226.142.216 172.31.8.238 123 123 17 1 76 1433807350 1433807398 ACCEPT OK
```

https://aws.amazon.com

Logs in AWS

There are multiple log sources on AWS platform that can provide investigators with data of evidentiary value during investigation. Here, we will discuss some of the log sources in AWS.

AWS CloudTrail

CloudTrail provides the AWS API call history for AWS accounts, including calls made via the AWS Management Console or Command Line tools, AWS Software Development Kits, and other AWS services. It is enabled by default when someone makes an AWS account. CloudTrail log analysis helps investigators in easily tracking the changes made to AWS resources and performing security analysis.

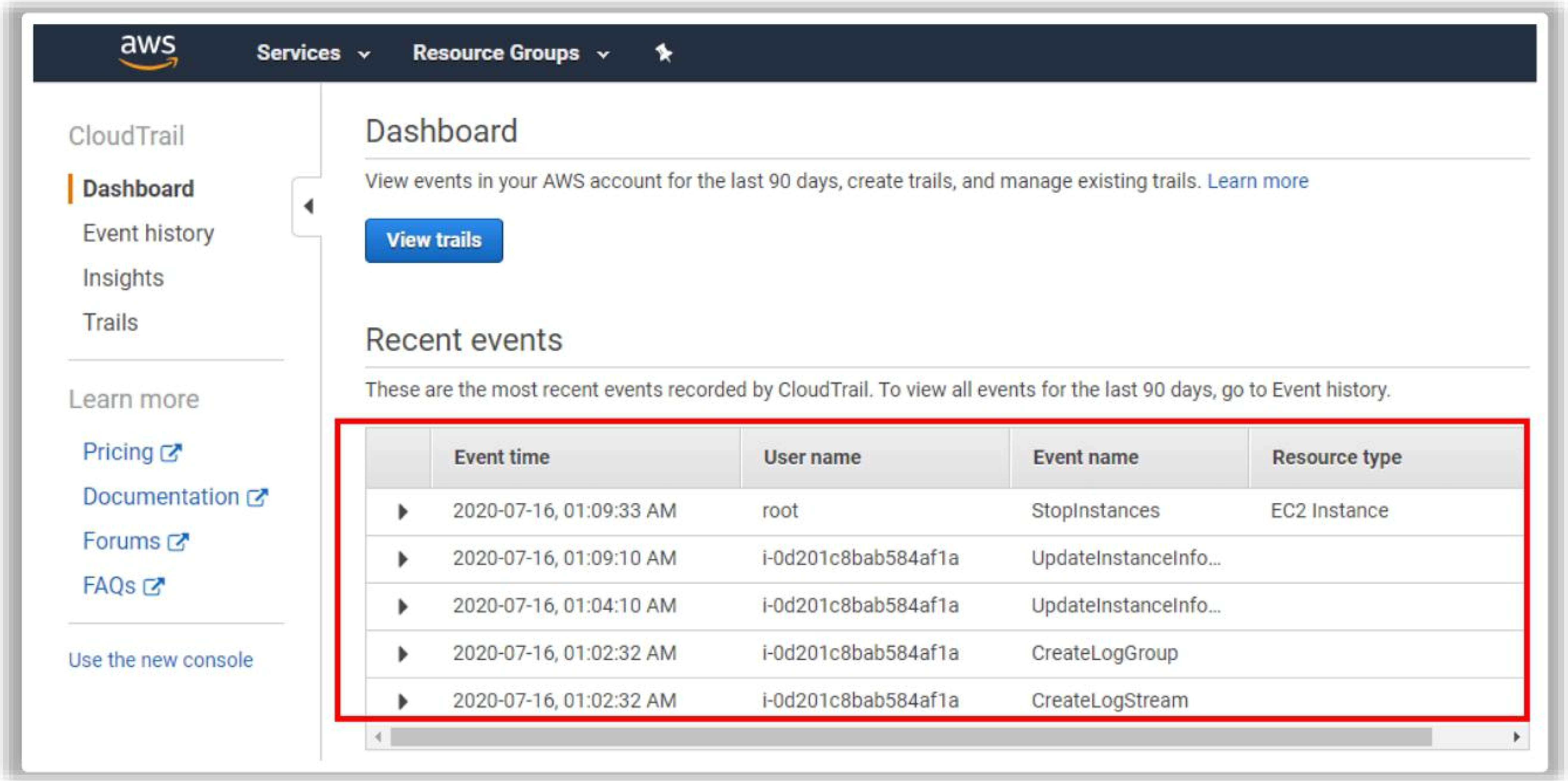
CloudTrail logs events that provide insights on the created, modified, or deleted AWS resources, along with the time stamp and user who performed these operations. The events are logged based on the region and can be viewed and downloaded from the CloudTrail Console in the JSON or CSV format.

CloudTrail logs provide enhanced visibility into a user along with details regarding resource activities such as the following:

- The time when a specific API call was made
- Source IP of the entity making the call
- Various request and response details

CloudTrail can be configured to produce management events that include log and security policy configuration as well as data events that provide details on the operations performed on a specific AWS resource. These logs are generated multiple times in an hour and can be read programmatically via the AWS CLI.

CloudTrail retains the API call events for 90 days. However, it is possible to create a trail of these CloudTrail events, send them to CloudWatch, and store them in an S3 bucket. These log trails can then be stored beyond the 90-day retention limit.



Dashboard

View events in your AWS account for the last 90 days, create trails, and manage existing trails. Learn more

Recent events

These are the most recent events recorded by CloudTrail. To view all events for the last 90 days, go to Event history.

Event time	User name	Event name	Resource type
2020-07-16, 01:09:33 AM	root	StopInstances	EC2 Instance
2020-07-16, 01:09:10 AM	i-0d201c8bab584af1a	UpdateInstanceInfo...	
2020-07-16, 01:04:10 AM	i-0d201c8bab584af1a	UpdateInstanceInfo...	
2020-07-16, 01:02:32 AM	i-0d201c8bab584af1a	CreateLogGroup	
2020-07-16, 01:02:32 AM	i-0d201c8bab584af1a	CreateLogStream	

Figure 8.14: An Example of Amazon CloudTrail Logs

Amazon CloudWatch

Amazon CloudWatch provides a platform for AWS customers to store and monitor their system and application log data in a centralized location and analyze them by performing search queries. CloudWatch log analysis helps in determining the origin of a problem and troubleshooting the system or application-specific errors.

CloudWatch logs can be exported and stored in exceptionally durable S3 buckets, thereby saving space on the hard drive. Amazon CloudWatch allows the inspection, access and storage of log files from various AWS sources such as AWS CloudTrail, EC2 instances, and Route 53.

From the perspective of a forensic investigator, CloudWatch log data analysis can be of immense value owing to the following reasons:

- **Real-Time Monitoring**

  CloudWatch helps investigators to examine the log data obtained from different data sources in real time and reconstruct how the security incident took place. Various log files can be sent to CloudWatch from both Linux and Windows-based EC2 instances such as Apache logs, authentication logs, event logs, and IIS logs.

- **Log Data Query**

  CloudWatch provides "Logs Insights" that enable investigators to find relevant log data by performing queries. These queries act as commands that filter the relevant data and save time.

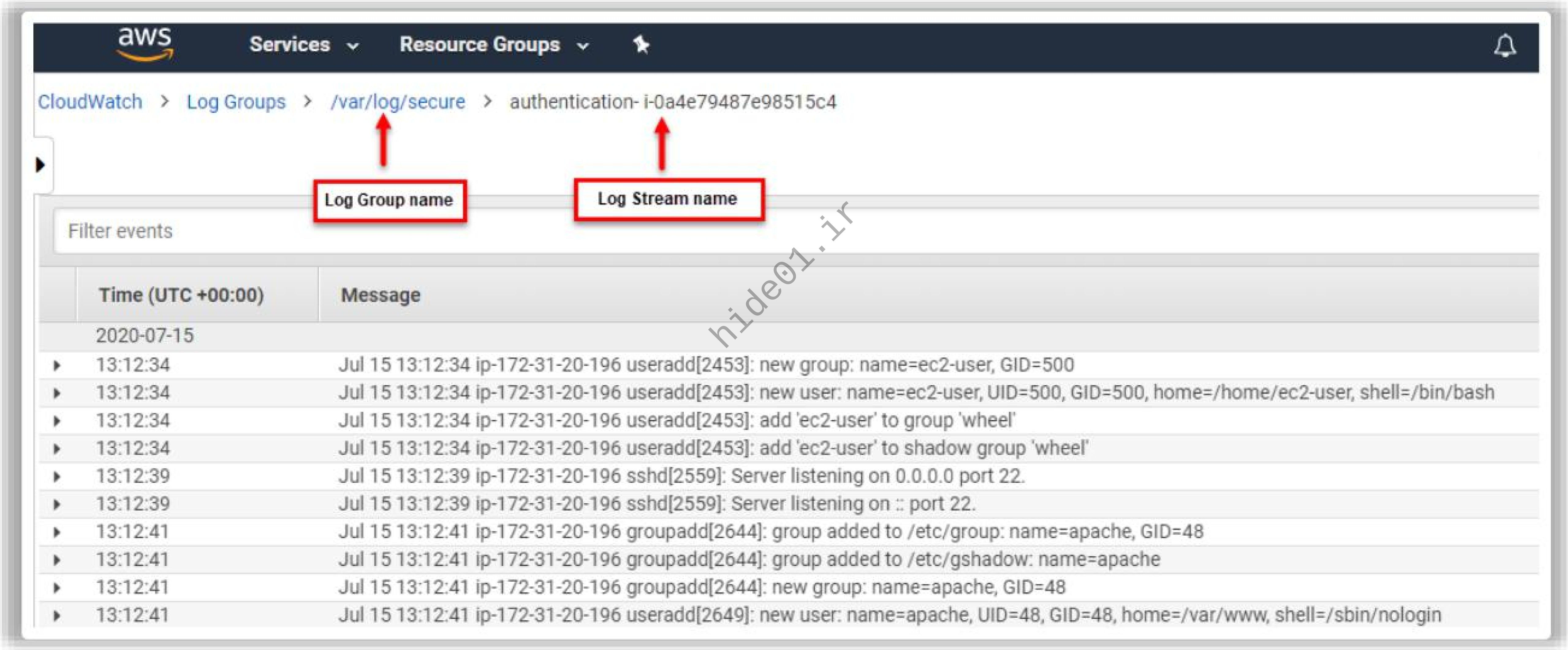
- **CloudTrail Log Monitoring**

  CloudTrail events can be sent to CloudWatch; this can help investigators in obtaining better visibility into multiple log sources at once

- **Route 53 Log Monitoring**

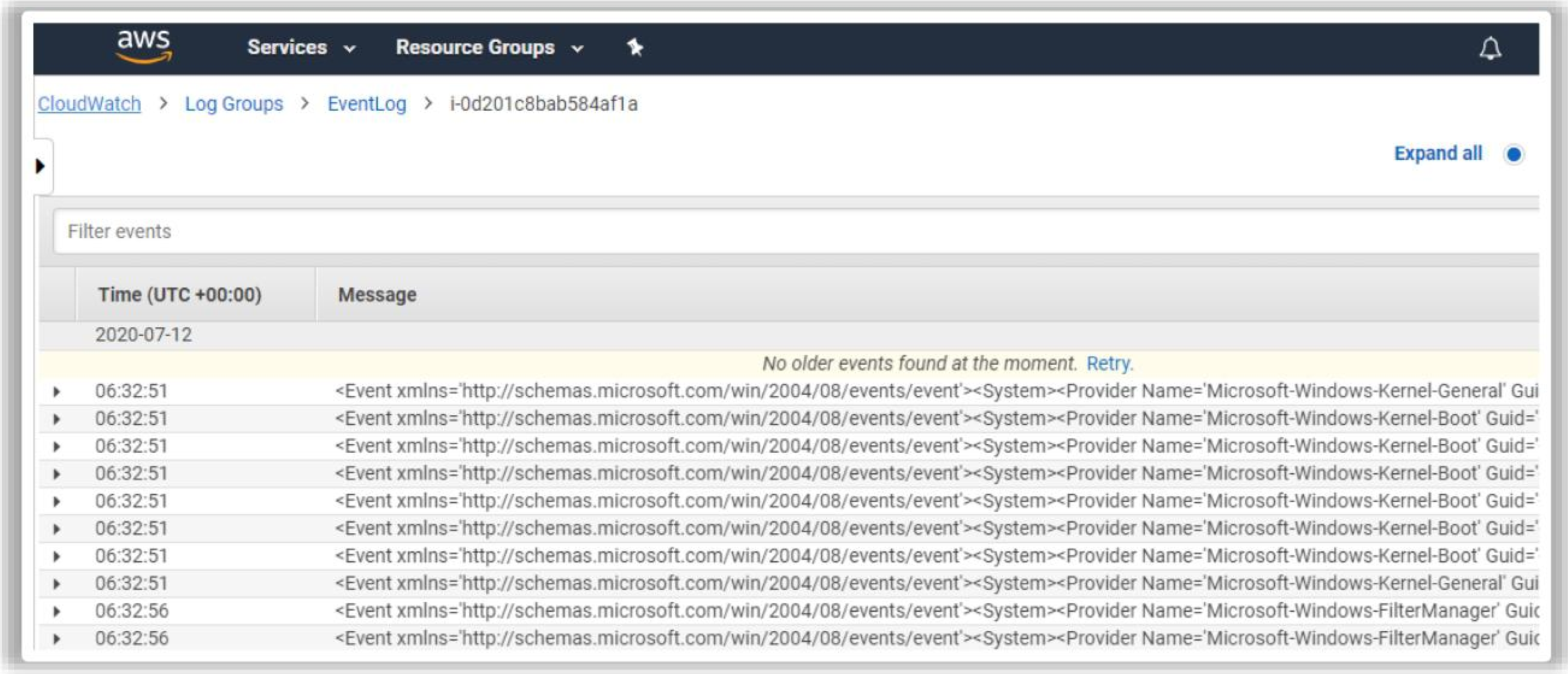
  Investigators can view and analyze all DNS queries received by Route 53 via the CloudWatch console

All types of EC2 instances send metrics related to health and performance in CloudWatch by default. However, to send system and application-specific log data from different EC2 instances, administrators must install and configure a CloudWatch agent on the respective EC2 instances and run the CloudWatch agent wizard to create specific log groups. Once a log group for a specific log source is created, all log entries of that log group are sent as a log stream. The name of a log stream can be customized. CloudWatch can store the log data for an indefinite period; however, the log retention time in CloudWatch can be customized. Administrators can enable the detailed logging of metrics and create alarms in CloudWatch for monitoring purposes. Moreover, CloudWatch log data can be searched and analyzed via Amazon CLI.


aws
Services
Resource Groups
CloudWatch > Log Groups > /var/log/secure > authentication- i-0a4e79487e98515c4
Log Group name
Log Stream name
Filter events

```
Time (UTC +00:00)    Message
2020-07-15
13:12:34             Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: new group: name=ec2-user, GID=500
13:12:34             Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: new user: name=ec2-user, UID=500, GID=500, home=/home/ec2-user, shell=/bin/bash
13:12:34             Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: add 'ec2-user' to group 'wheel'
13:12:34             Jul 15 13:12:34 ip-172-31-20-196 useradd[2453]: add 'ec2-user' to shadow group 'wheel'
13:12:39             Jul 15 13:12:39 ip-172-31-20-196 sshd[2559]: Server listening on 0.0.0.0 port 22.
13:12:39             Jul 15 13:12:39 ip-172-31-20-196 sshd[2559]: Server listening on :: port 22.
13:12:41             Jul 15 13:12:41 ip-172-31-20-196 groupadd[2644]: group added to /etc/group: name=apache, GID=48
13:12:41             Jul 15 13:12:41 ip-172-31-20-196 groupadd[2644]: group added to /etc/gshadow: name=apache
13:12:41             Jul 15 13:12:41 ip-172-31-20-196 groupadd[2644]: new group: name=apache, GID=48
13:12:41             Jul 15 13:12:41 ip-172-31-20-196 useradd[2649]: new user: name=apache, UID=48, GID=48, home=/var/www, shell=/sbin/nologin
```

Figure 8.15: Linux Server Logs on Amazon CloudWatch



```
Time (UTC +00:00)    Message
2020-07-12
                     No older events found at the moment. Retry.
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-General' Gui
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-Boot' Guid='
06:32:51             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Kernel-General' Gui
06:32:56             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-FilterManager' Guic
06:32:56             <Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-FilterManager' Guic
```

Figure 8.16: Windows Event Logs on Amazon CloudWatch

S3 Server Access Logs

S3 server access logging, when enabled, records information of all requests made to any bucket. Requests such as GET, PUT, and DELETE are captured, which helps investigators to understand the actions that were performed on a bucket object along with the users who performed these actions. If a bucket object is missing after a security breach, server access logging can help in tracing the perpetrator.

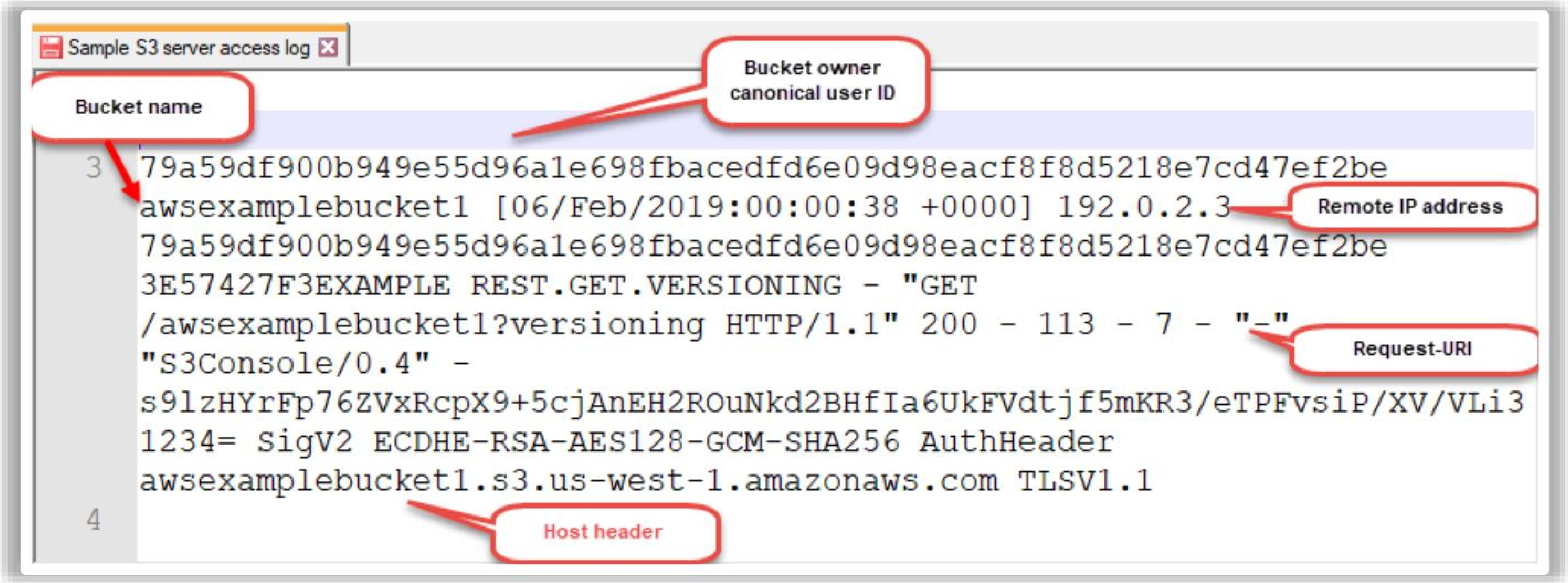
Following is an example of an S3 server log entry:

```
67d765aabc059cac8a41156d40f084b25e7363b479bf5ea23b392ef9106c7038 bucket1
[18/Jul/2020:10:54:36 +0000] 172.168.0.1
67d765aabc059cac8a41156d40f084b25e7363b479bf5ea23b392ef9106c7038
AB4B2C7CAADD7513 REST.HEAD.BUCKET - "HEAD /bucket1 HTTP/1.1" 200 - - - 13 13 "-"
"S3Console/0.4, aws-internal/3 aws-sdk-java/1.11.783 Linux/4.9.217-
0.1.ac.205.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.252-b09 java/1.8.0_252
vendor/Oracle_Corporation" -
Dq1Di4V1ymqelH6MA9CjIDiEHrSs1Zn6058v5m/1hOj8gSAnuBgWNpi832Jrv8vXwBpUjFZCiwg=
SigV4 ECDHE-RSA-AES128-SHA AuthHeader s3.us-east-2.amazonaws.com TLSv1.2
```

Descriptions of different parts of the log are listed below:

- **Bucket Owner:** It is the canonical ID of the bucket owner. Here, it is 67d765aabc059cac8a41156d40f084b25e7363b479bf5ea23b392ef9106c7038.
- **Bucket Name:** It shows the name of the bucket whose objects were requested for access. Here, it is bucket1
- **Time of Request:** The shows the time when the request was made. Here, it is 18/Jul/2020:10:54:36
- **Remote IP:** It shows the public IP of the requester. It can be forged if the requester is using VPN or proxy.
- **Requester:** This part shows the canonical ID of the requester and a blank value is returned in the case of unauthenticated requests. Additionally, if the requester has an IAM role associated with their AWS account, it will show its IAM name along with the AWS root account details. In the log entry, the canonical ID of the requester and owner is the same.
- **Request ID:** It shows the ID of the specific request. In the example log, it is AB4B2C7CAADD7513
- **Operation:** Operations include REST.HTTP_method.resource_type, SOAP operations, GET, PUT, or DELETE operations. In the above log entry, REST.HEAD.BUCKET is a sample operation.
- **Request-URI:** It shows the URI associated with the request. In the example log, it is HEAD /bucket1 HTTP/1.1
- **HTTP Status Code:** The status code 200 indicates that the server processed the request
- **Error Code:** This shows errors, if any

- **Bytes Sent:** No entry in the example log
- **Size of Object**: No object size is mentioned in the example log
- **Total Time Taken:** The server took 13 ms to send a response for the request
- **Turnaround Time:** This shows the request processing time. In the example log, it is 13 ms.
- **User Agent Referrer:** It refers to the HTTP referrer header. In the example log, the referrer header is S3Console/0.4, aws-internal/3 aws-sdk-java/1.11.783 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.252-b09 java/1.8.0_252 vendor/Oracle_Corporation.
- **Version ID:** The example log does not include any version ID
- **Host ID:** An extended request ID for Amazon, which is Dq1Di4V1ymqelH6MA9CjIDiEHrSs1Zn6058v5m/1hOj8gSAnuBgWNpi832Jrv8vXwBpUjFZCiwg=.
- **Signature Version:** It can be both SigV2 and SigV4. The sign "-" is generated for unauthenticated requests.
- **Cipher Suite Used:** ECDHE-RSA-AES128-SHA, which denotes an SSL cipher
- **Type of Authentication:** It shows the authentication request sent; AuthHeader for authenticated requests and "-" for unauthenticated ones
- **Host Header:** In the example log, the host header is s3.us-east-2.amazonaws.com, which points to the endpoint connecting to S3
- **TLS Version:** In the example log, the TLS version is TLSv1.2, which denotes the version of TLS



```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
awsexamplebucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
3E57427F3EXAMPLE REST.GET.VERSIONING - "GET
/awsexamplebucket1?versioning HTTP/1.1" 200 - 113 - 7 - "-"
"S3Console/0.4" -
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi3
1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader
awsexamplebucket1.s3.us-west-1.amazonaws.com TLSV1.1
```

Figure 8.17: Different Components of Amazon S3 Server Access Log

VPC Flow Logs

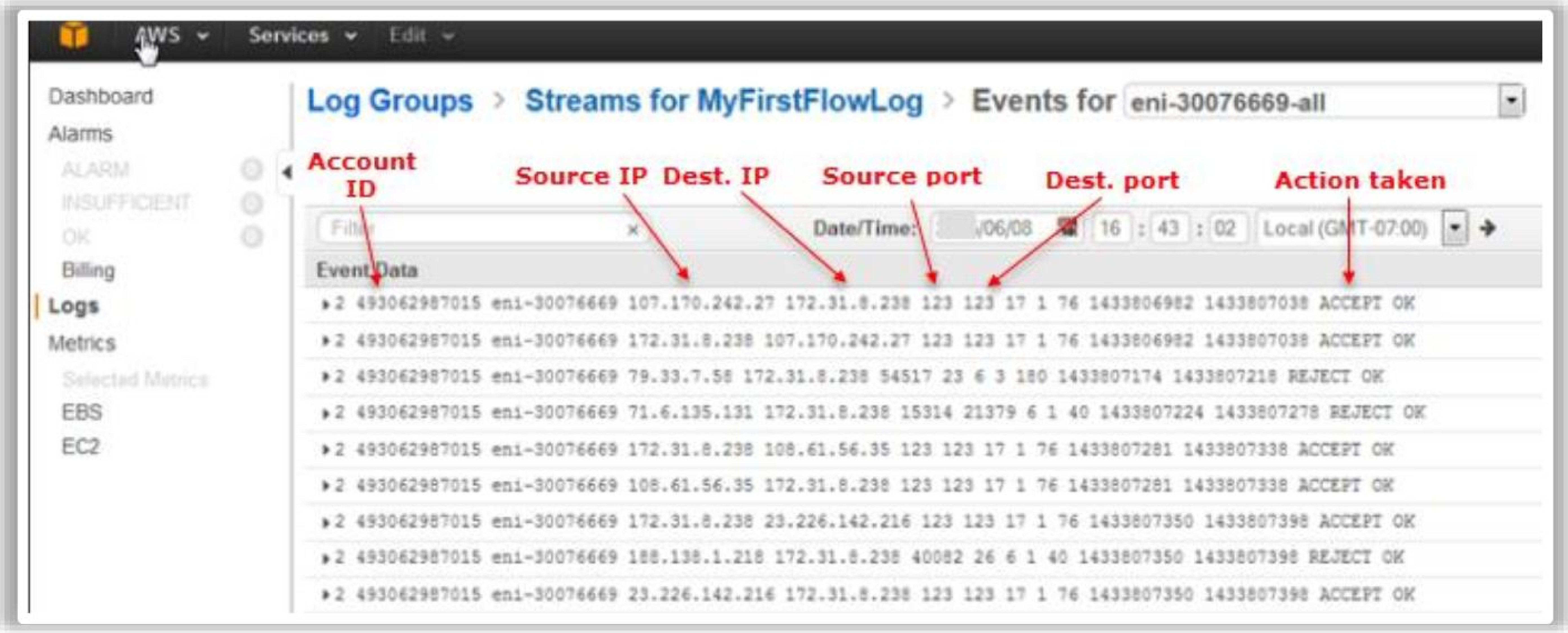
Amazon VPC flow logs show the IP traffic going in and out of a specific VPC. These log entries help in troubleshooting network-related issues and provide substantial evidential data to forensic investigators in the case of a security breach. The flow log data can be integrated with an SIEM tool or published to CloudWatch to obtain better visibility into whether the EC2 instances are communicating with any suspicious IP addresses/ports. Organizations using network flow logs can also enable alarms that get triggered if a certain type of traffic flow is identified and use metrics to better understand the flow patterns.

A sample VPC log entry for a specific VPC subnet or Elastic Network Interface is given below:

498062897015 eni-40078889 107.180.232.28 172.168.8.9 123 789 17 3 76 1433806982 1433807038 ACCEPT OK

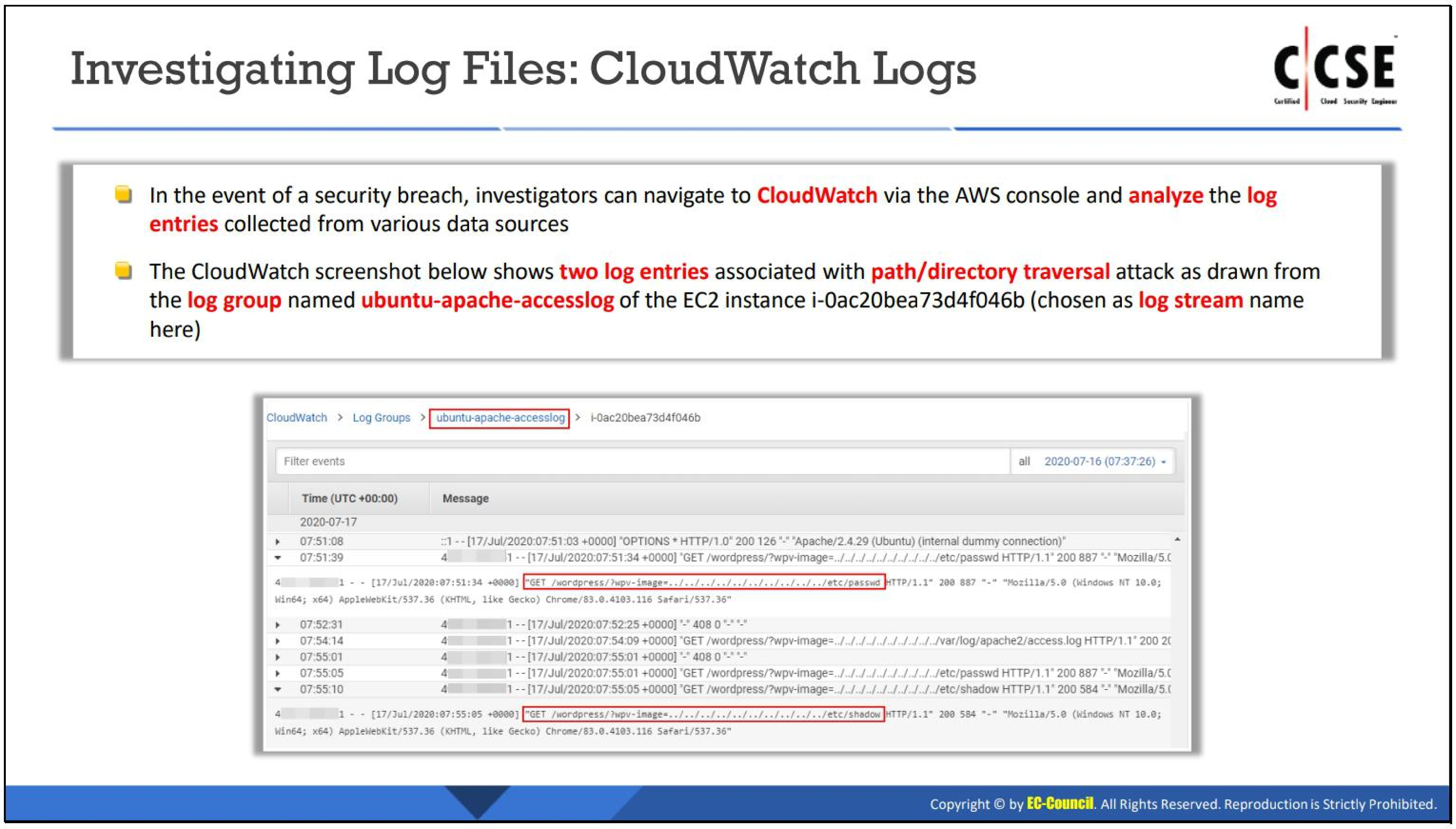
Let us analyze the different parts of the log entry -

- **Account ID:** 498062897015
- **ENI-ID:** eni-40078889
- **Source IP Address:** 107.180.232.28
- **Destination/Server IP Address:** 172.168.8.9
- **Source Port:** 123
- **Destination Port:** 789
- **Protocol Used:** 17
- **Number of Packets Transferred:** 3
- **Bytes Transferred:** 76
- **Start Time:** 1433806982
- **End Time:** 1433807038
- **Action Taken:** ACCEPT



```
▸2 493062987015 eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 1433806982 1433807038 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 1433806982 1433807038 ACCEPT OK
▸2 493062987015 eni-30076669 79.33.7.58 172.31.8.238 54517 23 6 3 180 1433807174 1433807218 REJECT OK
▸2 493062987015 eni-30076669 71.6.135.131 172.31.8.238 15314 21379 6 1 40 1433807224 1433807278 REJECT OK
▸2 493062987015 eni-30076669 172.31.8.238 108.61.56.35 123 123 17 1 76 1433807281 1433807338 ACCEPT OK
▸2 493062987015 eni-30076669 108.61.56.35 172.31.8.238 123 123 17 1 76 1433807281 1433807338 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 23.226.142.216 123 123 17 1 76 1433807350 1433807398 ACCEPT OK
▸2 493062987015 eni-30076669 188.138.1.218 172.31.8.238 40082 26 6 1 40 1433807350 1433807398 REJECT OK
▸2 493062987015 eni-30076669 23.226.142.216 172.31.8.238 123 123 17 1 76 1433807350 1433807398 ACCEPT OK
```

Figure 8.18: Different Components of Amazon VPC Flow Log



Investigating Log Files: CloudWatch Logs

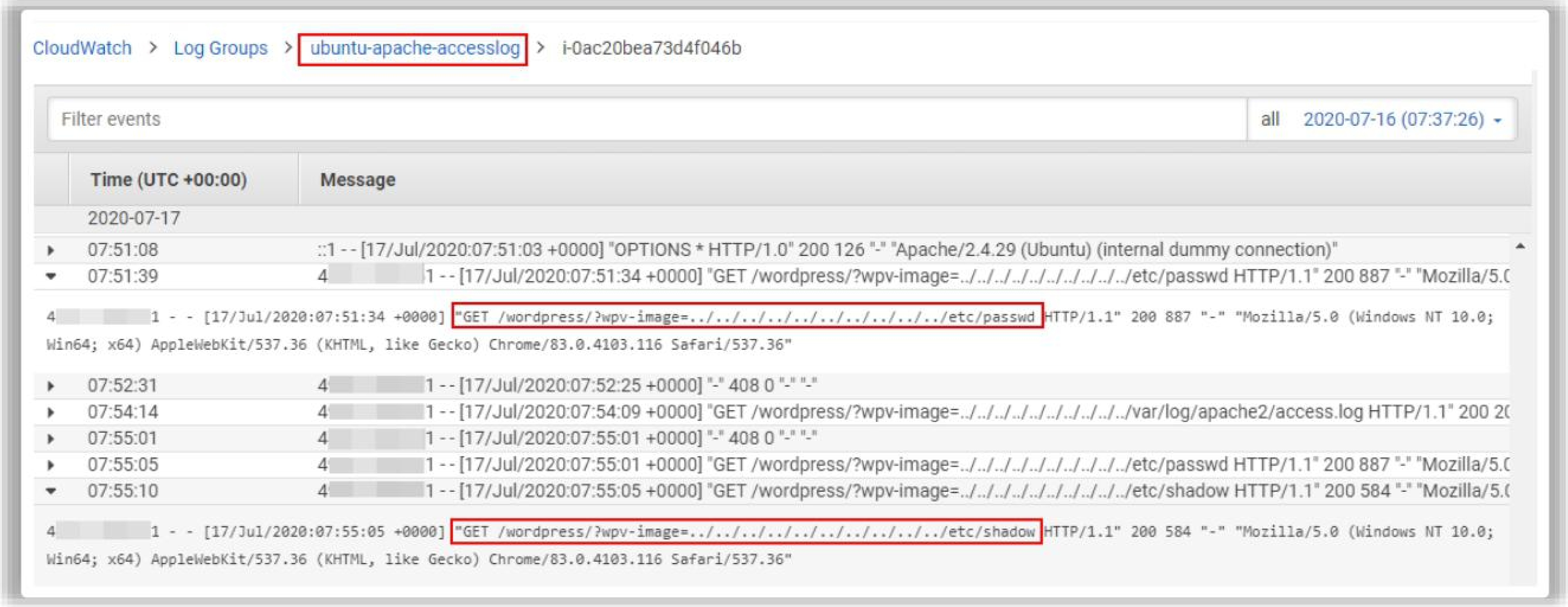
- In the event of a security breach, investigators can navigate to **CloudWatch** via the AWS console and **analyze** the **log entries** collected from various data sources
- The CloudWatch screenshot below shows **two log entries** associated with **path/directory traversal** attack as drawn from the **log group** named **ubuntu-apache-accesslog** of the EC2 instance i-0ac20bea73d4f046b (chosen as **log stream** name here)

CloudWatch > Log Groups > ubuntu-apache-accesslog > i-0ac20bea73d4f046b

```
Time (UTC +00:00)   Message
2020-07-17
07:51:08   ::1 - - [17/Jul/2020:07:51:03 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.29 (Ubuntu) (internal dummy connection)"
07:51:39   4[redacted]1 - - [17/Jul/2020:07:51:34 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/passwd HTTP/1.1" 200 887 "-" "Mozilla/5.(
4[redacted]1 - - [17/Jul/2020:07:51:34 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/passwd HTTP/1.1" 200 887 "-" "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36"
07:52:31   4[redacted]1 - - [17/Jul/2020:07:52:25 +0000] "-" 408 0 "-" "-"
07:54:14   4[redacted]1 - - [17/Jul/2020:07:54:09 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../var/log/apache2/access.log HTTP/1.1" 200 2(
07:55:01   4[redacted]1 - - [17/Jul/2020:07:55:01 +0000] "-" 408 0 "-" "-"
07:55:05   4[redacted]1 - - [17/Jul/2020:07:55:01 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/passwd HTTP/1.1" 200 887 "-" "Mozilla/5.(
07:55:10   4[redacted]1 - - [17/Jul/2020:07:55:05 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/shadow HTTP/1.1" 200 584 "-" "Mozilla/5.(
4[redacted]1 - - [17/Jul/2020:07:55:05 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/shadow HTTP/1.1" 200 584 "-" "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36"
```

Investigating Log Files: CloudWatch Logs

In the event of a security incident, examining CloudWatch log files can prove to be valuable for forensic investigators because it collects logs from various data sources and stores them on a centralized location.

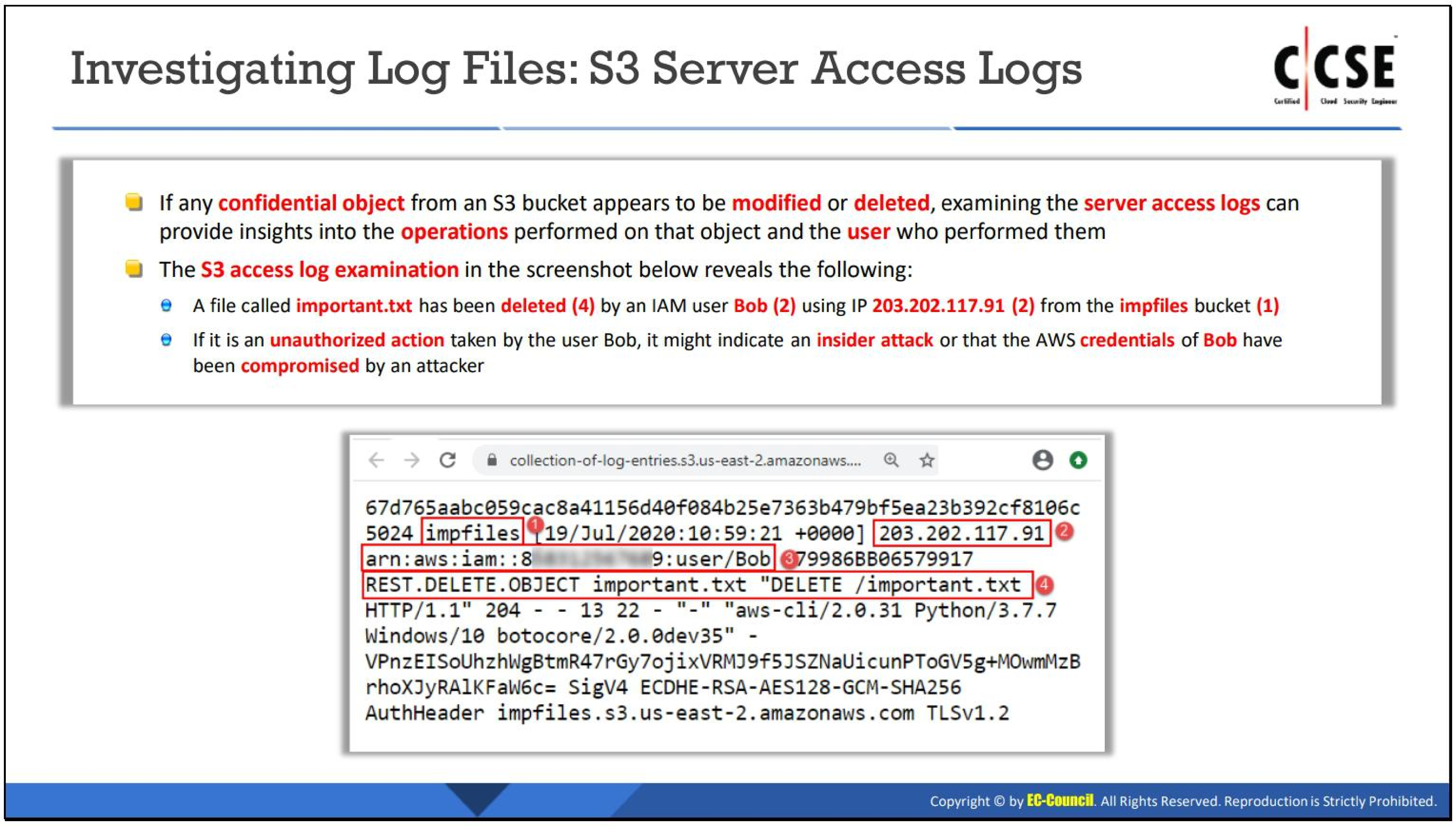


CloudWatch > Log Groups > ubuntu-apache-accesslog > i-0ac20bea73d4f046b

```
Time (UTC +00:00)   Message
2020-07-17
07:51:08   ::1 - - [17/Jul/2020:07:51:03 +0000] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.29 (Ubuntu) (internal dummy connection)"
07:51:39   4[redacted]1 - - [17/Jul/2020:07:51:34 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/passwd HTTP/1.1" 200 887 "-" "Mozilla/5.(
4[redacted]1 - - [17/Jul/2020:07:51:34 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/passwd HTTP/1.1" 200 887 "-" "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36"
07:52:31   4[redacted]1 - - [17/Jul/2020:07:52:25 +0000] "-" 408 0 "-" "-"
07:54:14   4[redacted]1 - - [17/Jul/2020:07:54:09 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../var/log/apache2/access.log HTTP/1.1" 200 2(
07:55:01   4[redacted]1 - - [17/Jul/2020:07:55:01 +0000] "-" 408 0 "-" "-"
07:55:05   4[redacted]1 - - [17/Jul/2020:07:55:01 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/passwd HTTP/1.1" 200 887 "-" "Mozilla/5.(
07:55:10   4[redacted]1 - - [17/Jul/2020:07:55:05 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/shadow HTTP/1.1" 200 584 "-" "Mozilla/5.(
4[redacted]1 - - [17/Jul/2020:07:55:05 +0000] "GET /wordpress/?wpv-image=../../../../../../../../../../etc/shadow HTTP/1.1" 200 584 "-" "Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36"
```

Figure 8.19: Investigating CloudWatch Logs

The figure above is taken from the CloudWatch console in which you can see two attempts on directory traversal attack path from a remote IP that occurred on the Apache server hosted on an EC2 instance running on Ubuntu. Two of the files requested by the attacker are /etc/passwd and /etc/shadow. Such log findings confirm the presence of an attack and in initiating measures to stop it.



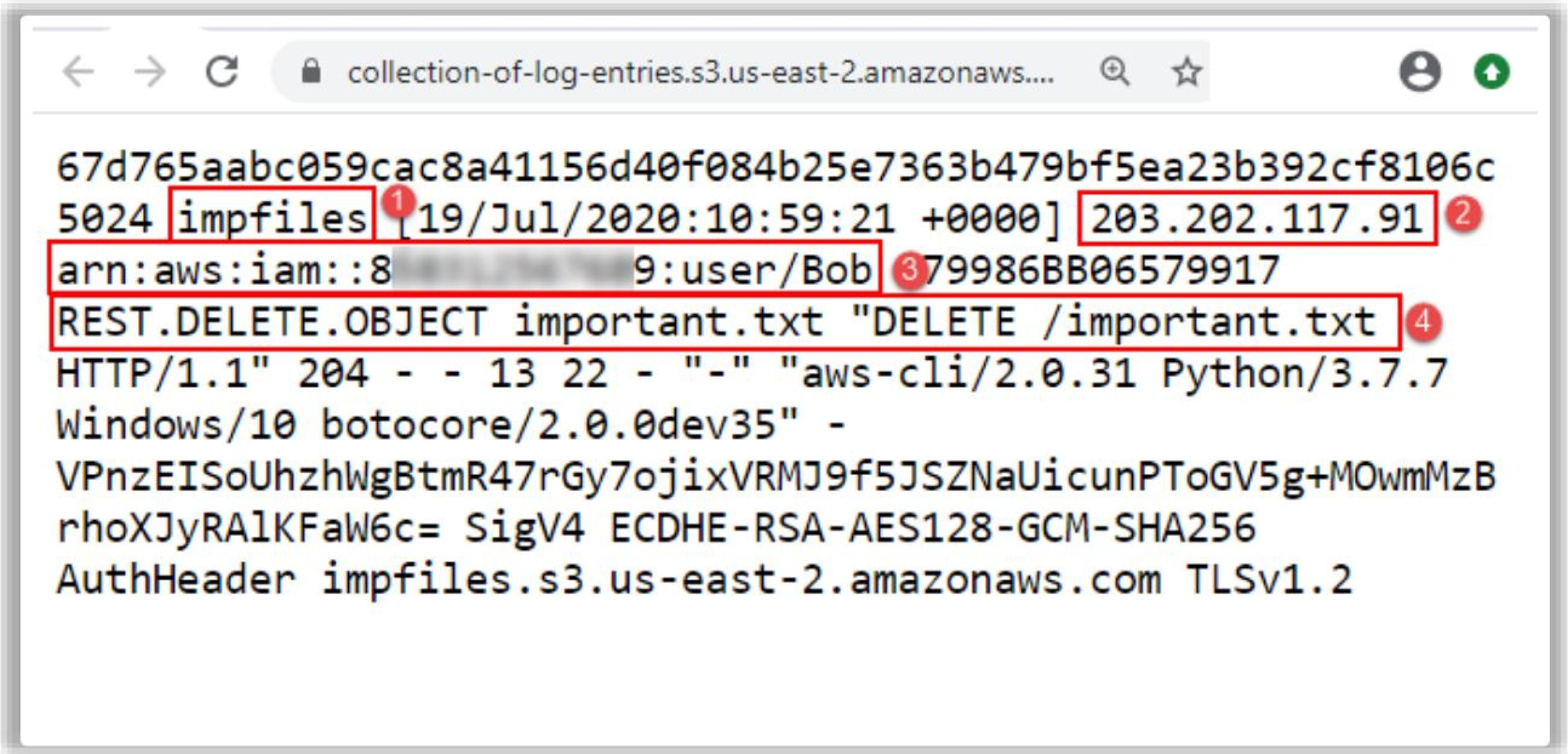
Investigating Log Files: S3 Server Access Logs

- If any **confidential object** from an S3 bucket appears to be **modified** or **deleted**, examining the **server access logs** can provide insights into the **operations** performed on that object and the **user** who performed them
- The **S3 access log examination** in the screenshot below reveals the following:
  - A file called **important.txt** has been **deleted (4)** by an IAM user **Bob (2)** using IP **203.202.117.91 (2)** from the **impfiles** bucket **(1)**
  - If it is an **unauthorized action** taken by the user Bob, it might indicate an **insider attack** or that the AWS **credentials** of **Bob** have been **compromised** by an attacker

```
67d765aabc059cac8a41156d40f084b25e7363b479bf5ea23b392cf8106c
5024 impfiles [19/Jul/2020:10:59:21 +0000] 203.202.117.91
arn:aws:iam::8[redacted]9:user/Bob 79986BB06579917
REST.DELETE.OBJECT important.txt "DELETE /important.txt
HTTP/1.1" 204 - - 13 22 - "-" "aws-cli/2.0.31 Python/3.7.7
Windows/10 botocore/2.0.0dev35" -
VPnzEISoUhzhWgBtmR47rGy7ojixVRMJ9f5JSZNaUicunPToGV5g+MOwmMzB
rhoXJyRAlKFaW6c= SigV4 ECDHE-RSA-AES128-GCM-SHA256
AuthHeader impfiles.s3.us-east-2.amazonaws.com TLSv1.2
```

Investigating Log Files: S3 Server Access Logs

If an S3 bucket object containing sensitive business data appears to be modified or has suddenly gone missing, investigators can probe into the S3 server log files to determine the state of that object and the user responsible for it.



```
67d765aabc059cac8a41156d40f084b25e7363b479bf5ea23b392cf8106c
5024 impfiles [19/Jul/2020:10:59:21 +0000] 203.202.117.91
arn:aws:iam::8[redacted]9:user/Bob 79986BB06579917
REST.DELETE.OBJECT important.txt "DELETE /important.txt
HTTP/1.1" 204 - - 13 22 - "-" "aws-cli/2.0.31 Python/3.7.7
Windows/10 botocore/2.0.0dev35" -
VPnzEISoUhzhWgBtmR47rGy7ojixVRMJ9f5JSZNaUicunPToGV5g+MOwmMzB
rhoXJyRAlKFaW6c= SigV4 ECDHE-RSA-AES128-GCM-SHA256
AuthHeader impfiles.s3.us-east-2.amazonaws.com TLSv1.2
```

Figure 8.20: Investigating S3 Server Access Logs

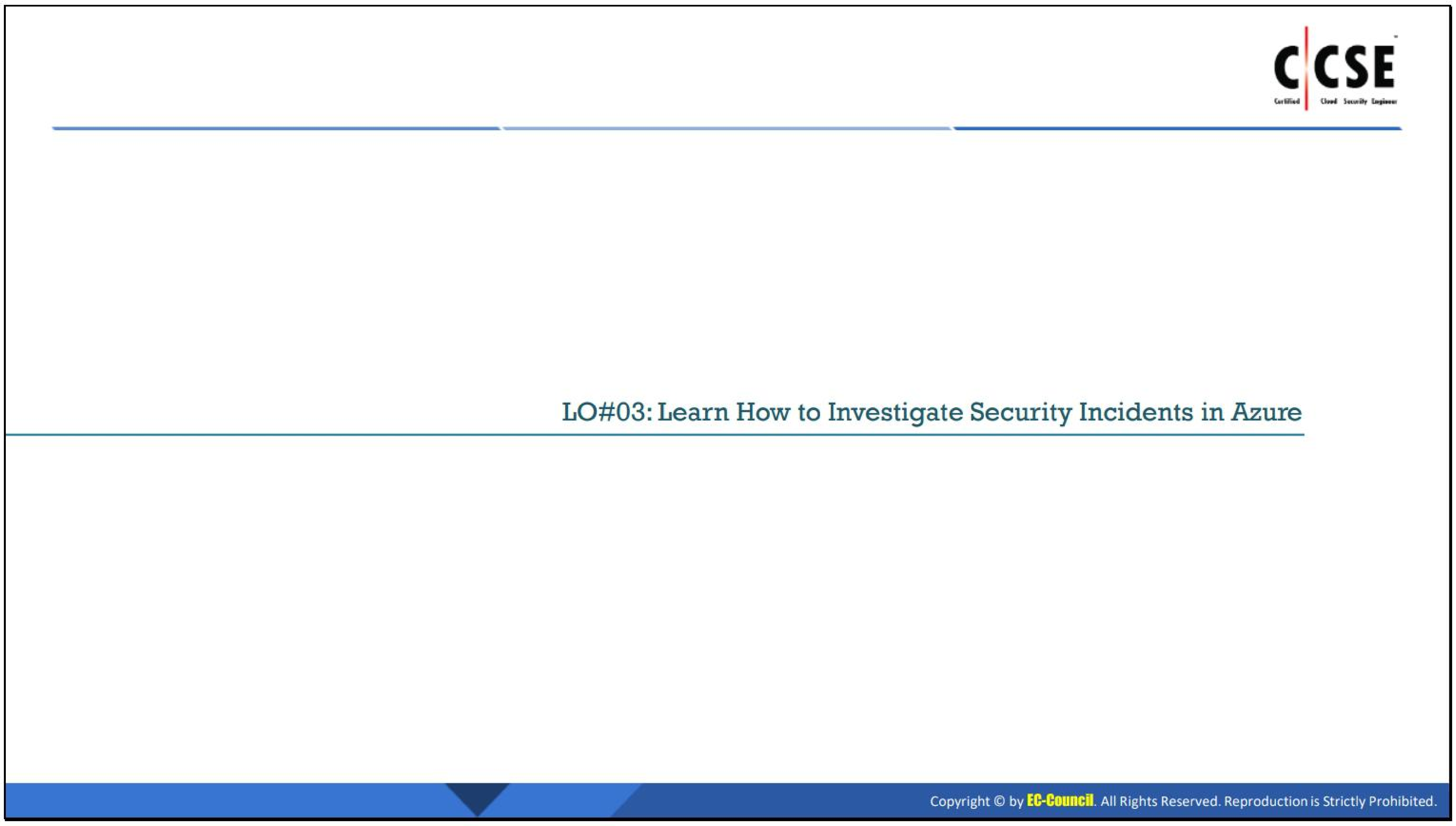
In the S3 server access log entry depicted in the figure above, it can be observed that a file named important.txt has been deleted by an IAM user, Bob, using IP 203.202.117.91 from the impfiles bucket. If this is an unauthorized action by the user, it indicates two possibilities:

- This is an insider attack

- The user credentials of “Bob” have been compromised by the attacker. The attacker logged into the AWS CLI as Bob and deleted the file using their privileges.

Other details that can be obtained from the log files include:

- Date of the attack is 19th July 2020
- The AWS CLI is used from a Windows machine

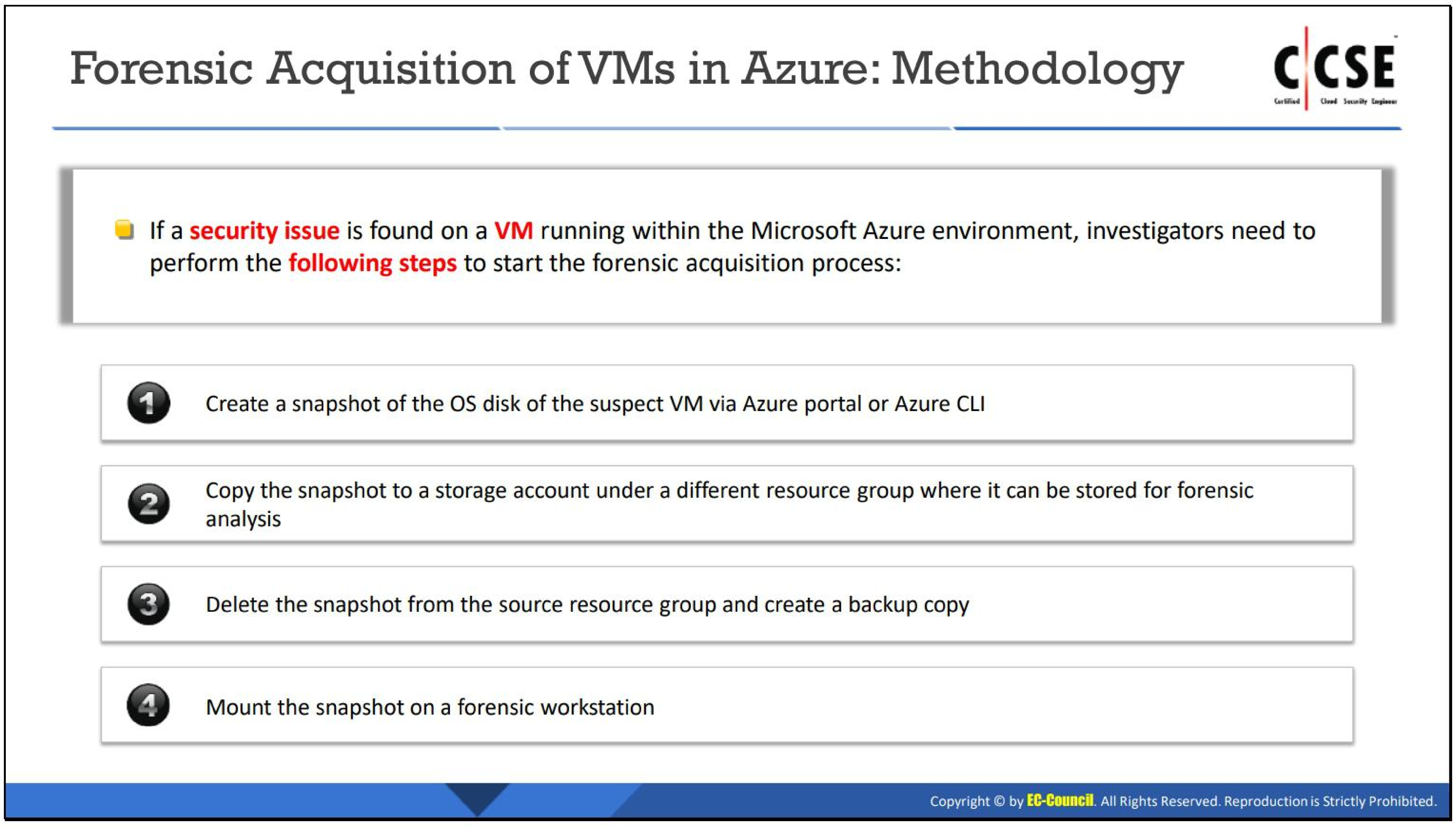


LO#03: Learn How to Investigate Security Incidents in Azure

LO#03: Learn How to Investigate Security Incidents in Azure

When a security incident is detected on the Azure cloud platform, investigators must examine the log data collected from various sources. If a VM is found to be affected, it is important to take a snapshot of the OS disk of the VM for further investigation.

This section discusses the forensic acquisition methodology of an Azure VM and discusses an assumed scenario to divide the whole process into multiple steps.



Forensic Acquisition of VMs in Azure: Methodology

- If a **security issue** is found on a **VM** running within the Microsoft Azure environment, investigators need to perform the **following steps** to start the forensic acquisition process:

1. Create a snapshot of the OS disk of the suspect VM via Azure portal or Azure CLI
2. Copy the snapshot to a storage account under a different resource group where it can be stored for forensic analysis
3. Delete the snapshot from the source resource group and create a backup copy
4. Mount the snapshot on a forensic workstation

Forensic Acquisition of VMs in Azure: Methodology

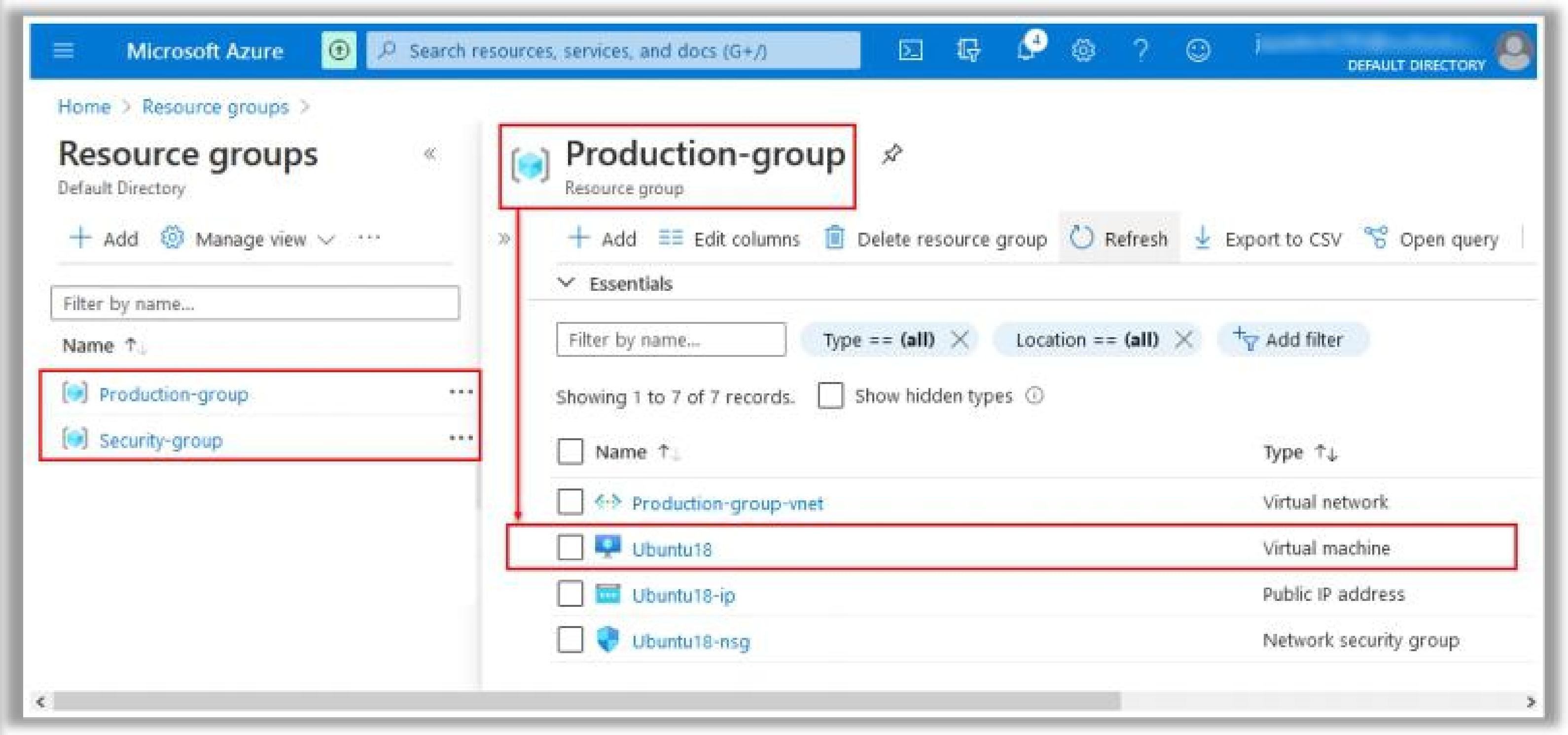
It is important to follow a stepwise process while acquiring an Azure VM that is suspected to be compromised. VM acquisition on the Azure cloud platform includes the following steps.

1. Create a snapshot of the OS disk of the suspect VM via Azure portal or Azure CLI
2. Copy the snapshot to a storage account under different resource groups where it can be stored for forensic analysis
3. Delete the snapshot from the source resource group and create a backup copy
4. Mount the snapshot onto the forensic workstation

Forensic Acquisition of VMs in Azure: The Scenario

Let us assume the following scenario:

- A company uses **two resource groups** under the same subscription ID:
  - **Production-group** which is used for its production environment
  - **Security-group** which is used for incident response and forensic investigation
- Under the Production-group, there is a **VM** called **Ubuntu18,** which is suspected to be **compromised**
- As a forensic investigator, you need to take a **snapshot of the OS disk** of the suspect VM Ubuntu18 for further investigation

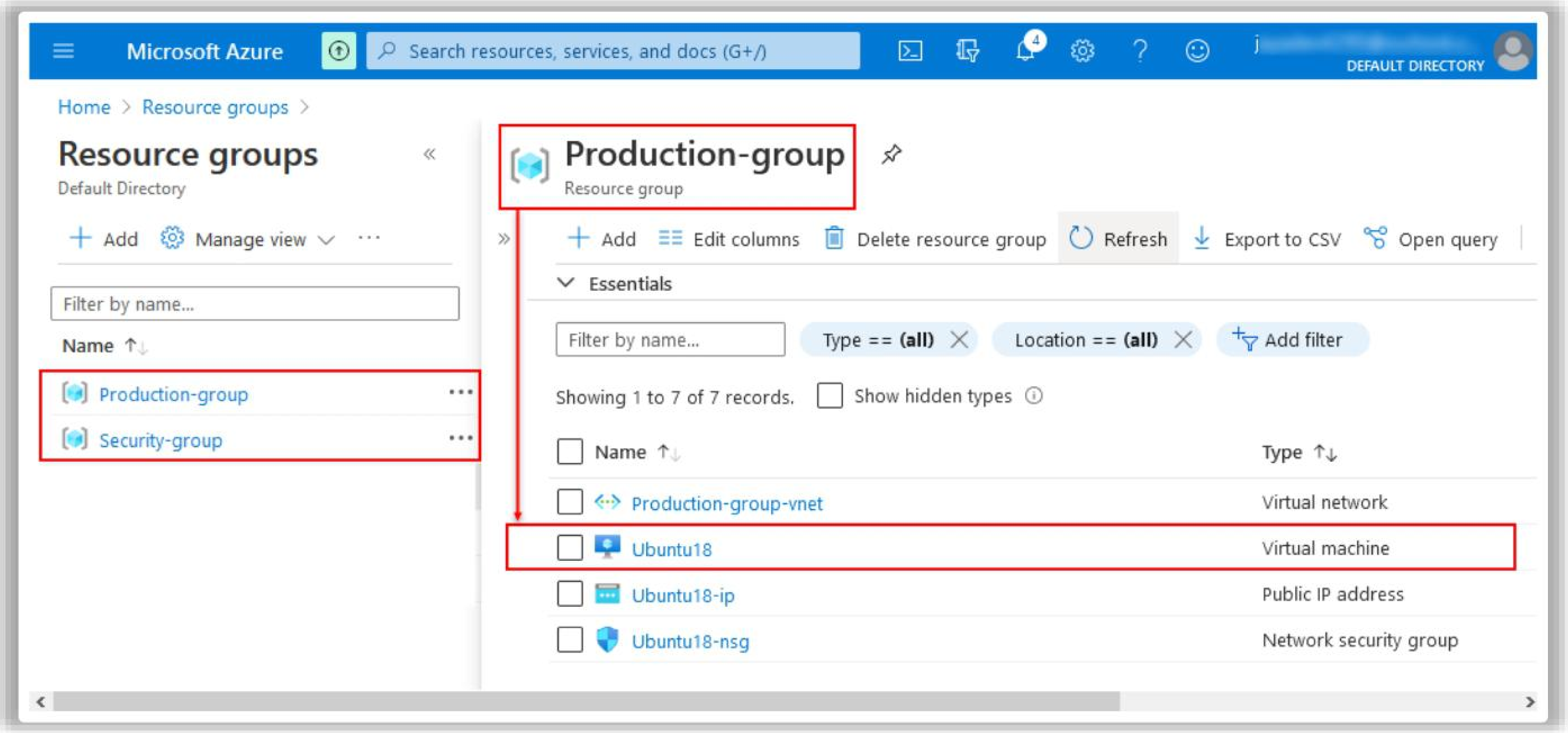


Forensic Acquisition of VMs in Azure: The Scenario

In this investigation, we assumed a scenario that a company uses two resource groups under the same subscription ID:

- Production-group which is used for its production environment
- Security-group which is used for incident response and forensic investigation

Under the Production-group, A VM called Ubuntu18 is suspected to be compromised. As a forensic investigator, you need to take a snapshot of the OS disk of the suspect VM Ubuntu18 for further investigation.



Name ↑↓	Type ↑↓
Production-group-vnet	Virtual network
Ubuntu18	Virtual machine
Ubuntu18-ip	Public IP address
Ubuntu18-nsg	Network security group

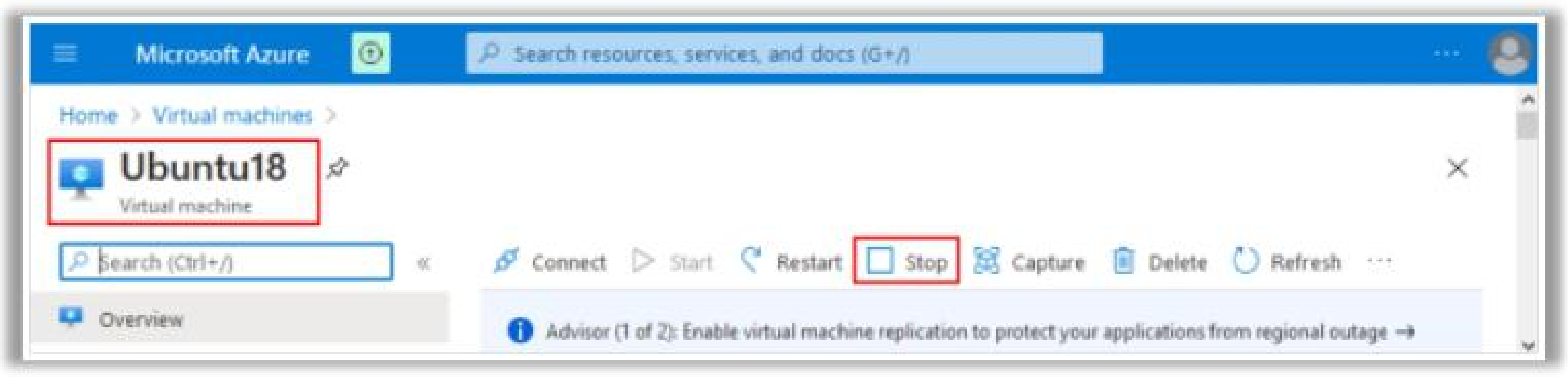
Figure 8.21: VM Named Ubuntu18 Under Production-Group

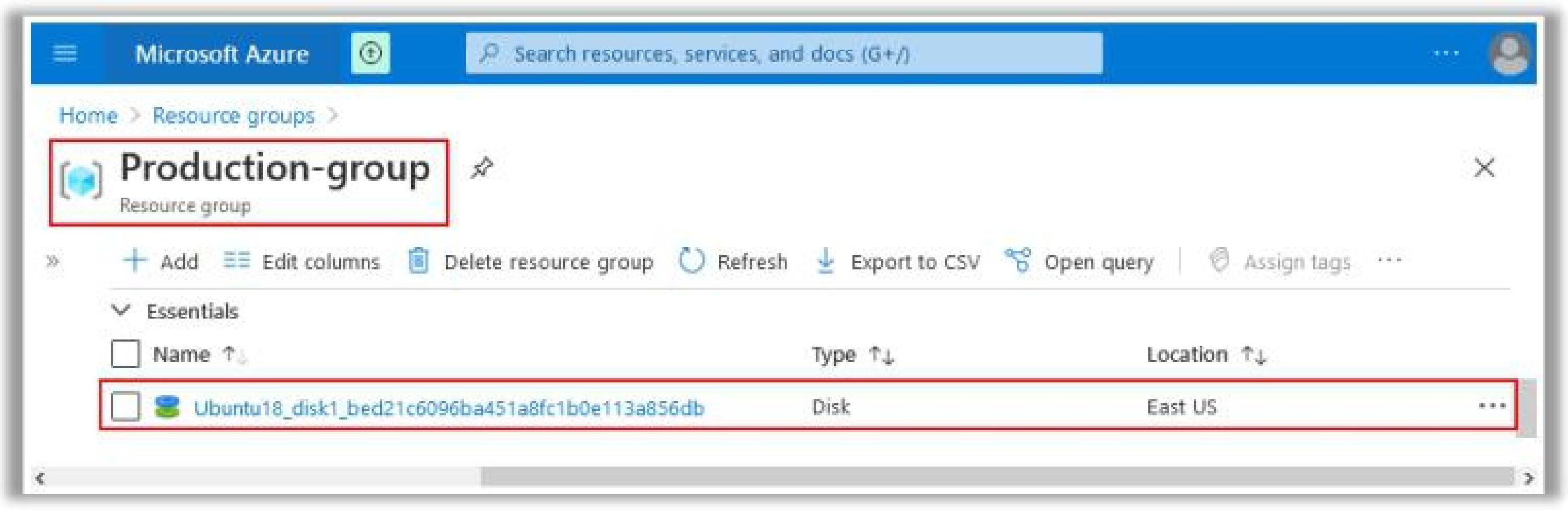
Step 1: Create a Snapshot of the OS Disk of the Affected VM via Azure Portal

1 Shut down the **Ubuntu18 VM** whose snapshot needs to be created

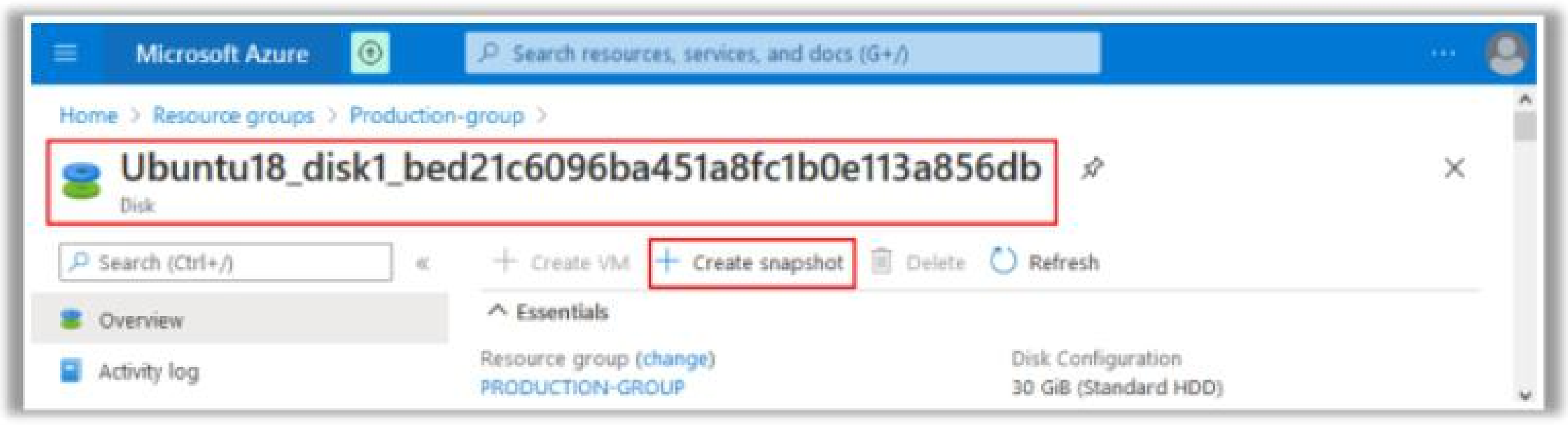
2 Locate the **Ubuntu18 OS disk** from the Production-group and click on it

3 On the next page that appears, click on the **Create Snapshot** button


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Virtual machines >
Ubuntu18
Virtual machine
Search (Ctrl+/)
Connect
Start
Restart
Stop
Capture
Delete
Refresh
Overview
Advisor (1 of 2): Enable virtual machine replication to protect your applications from regional outage →



Name	Type	Location
Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db	Disk	East US


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Resource groups > Production-group >
Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db
Disk
Search (Ctrl+/)
Create VM
Create snapshot
Delete
Refresh
Overview
Activity log
Essentials

Resource group (change): PRODUCTION-GROUP

Disk Configuration: 30 GiB (Standard HDD)

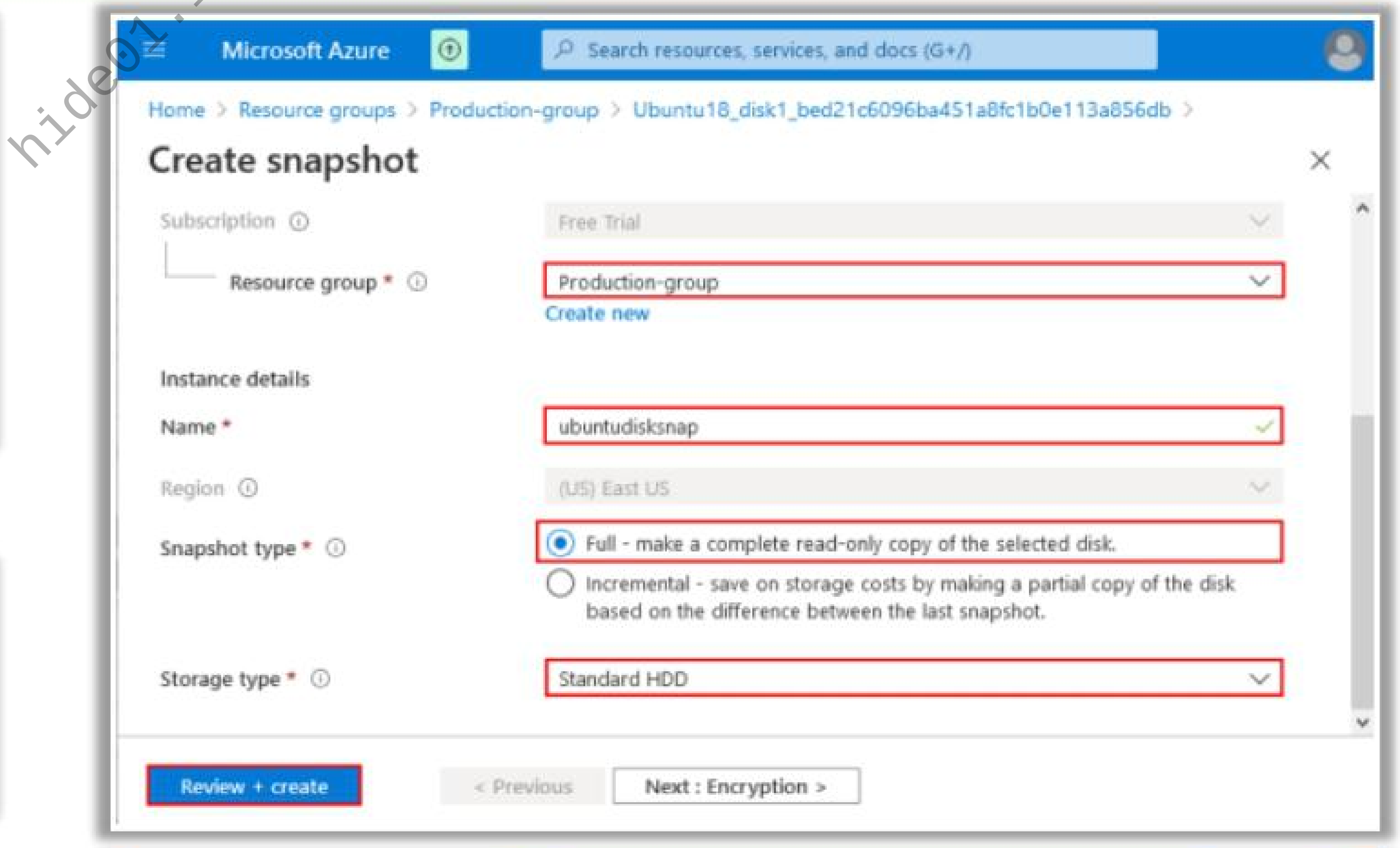
Step 1: Create a Snapshot of the OS Disk of the Affected VM via Azure Portal (Cont'd)

4 The **Create Snapshot** page would appear. Provide the following details:

- Give a desired **name** for the OS snapshot (here, it is **ubuntudisksnap**)
- Select the **snapshot type** as **read-only**
- Select a **storage type** (here, it is **standard HDD**)

5 After filling up all details, click on the **Review+create** button

6 Click on **Create** in the next page. The snapshot is now successfully created


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Resource groups > Production-group > Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db >
Create new
Review + create
< Previous
Next : Encryption >

Create snapshot

Subscription: Free Trial

Resource group *: Production-group

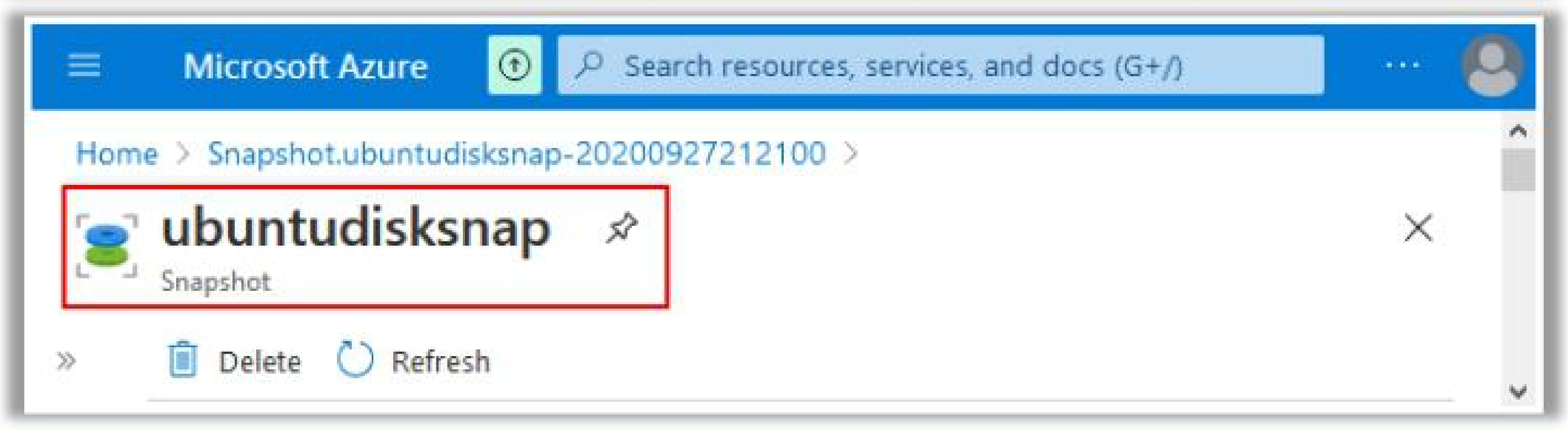
Instance details

Name *: ubuntudisksnap

Region: (US) East US

Snapshot type *:
● Full - make a complete read-only copy of the selected disk.
○ Incremental - save on storage costs by making a partial copy of the disk based on the difference between the last snapshot.

Storage type *: Standard HDD


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Snapshot.ubuntudisksnap-20200927212100 >
ubuntudisksnap
Snapshot
Delete
Refresh

Step 1: Create a Snapshot of the OS Disk of the Affected VM via Azure Portal

Following are some steps that you need to follow to create a snapshot of the OS disk of the affected VM named Ubuntu18 on Azure portal:

1. Stop the Ubuntu18 VM whose snapshot needs to be created

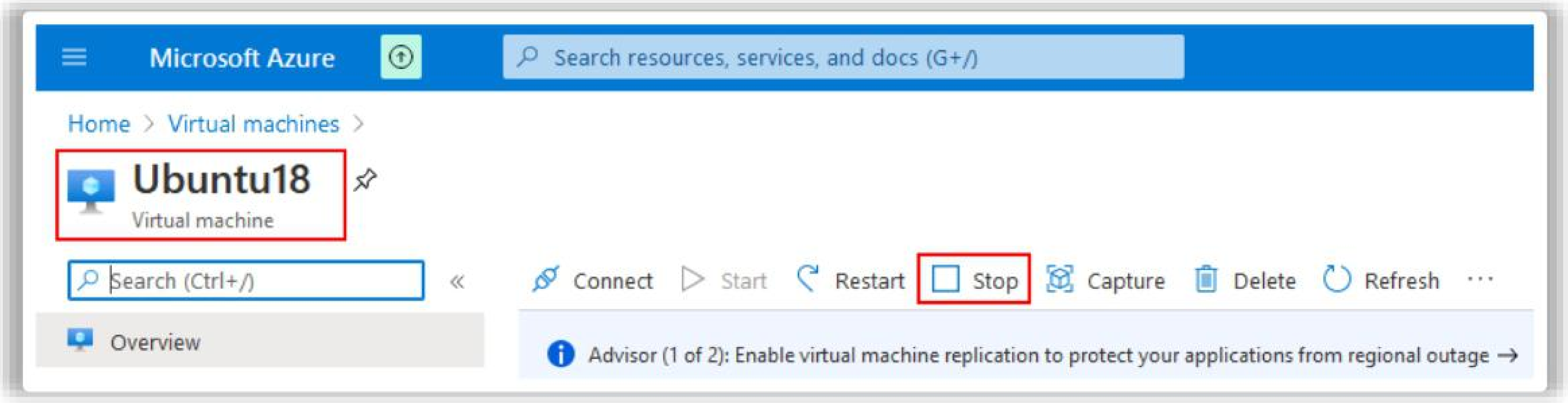

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Virtual machines >
Ubuntu18
Virtual machine
Search (Ctrl+/)
Overview
Connect
Start
Restart
Stop
Capture
Delete
Refresh
Advisor (1 of 2): Enable virtual machine replication to protect your applications from regional outage →

Figure 8.22: Stopping Ubuntu18 VM

2. Locate the Ubuntu18 OS disk from the Production-group and click on it

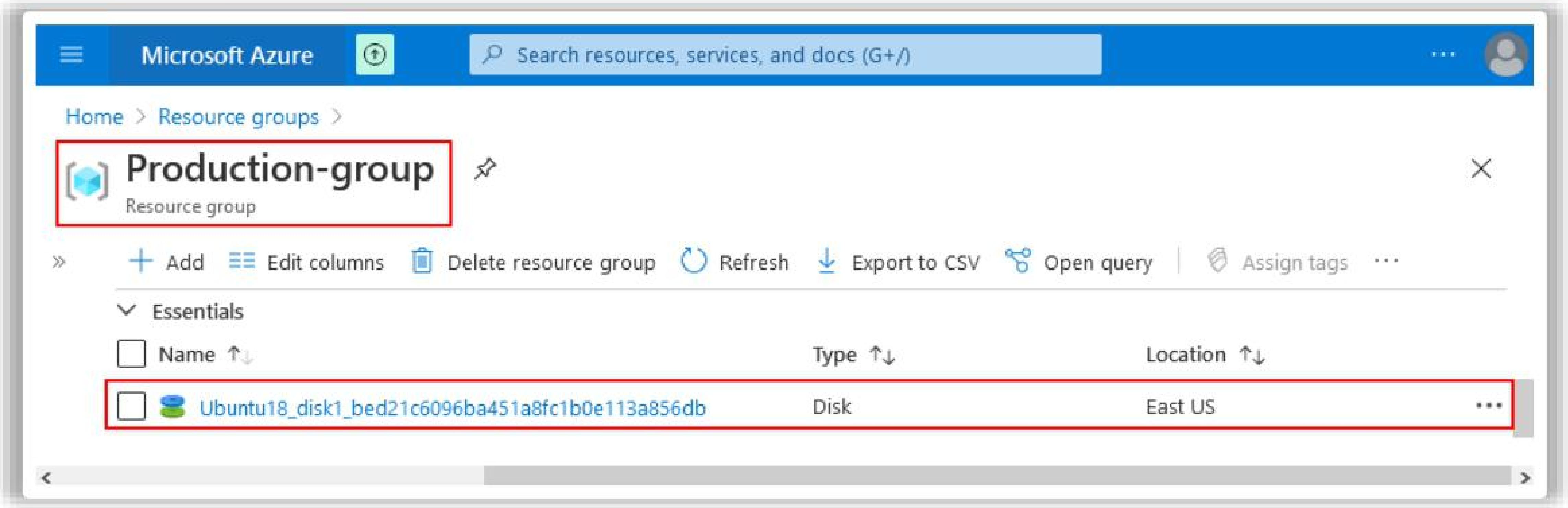


Figure 8.23: Selecting the OS Disk of Ubuntu18 VM

3. On the next page that appears, click on the Create Snapshot button

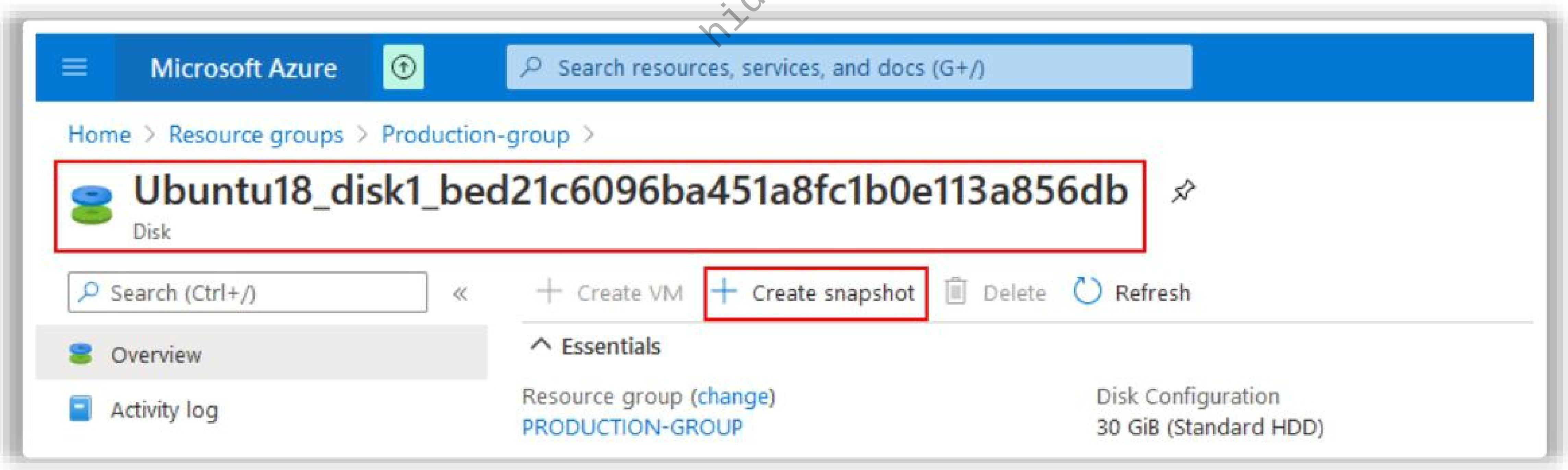
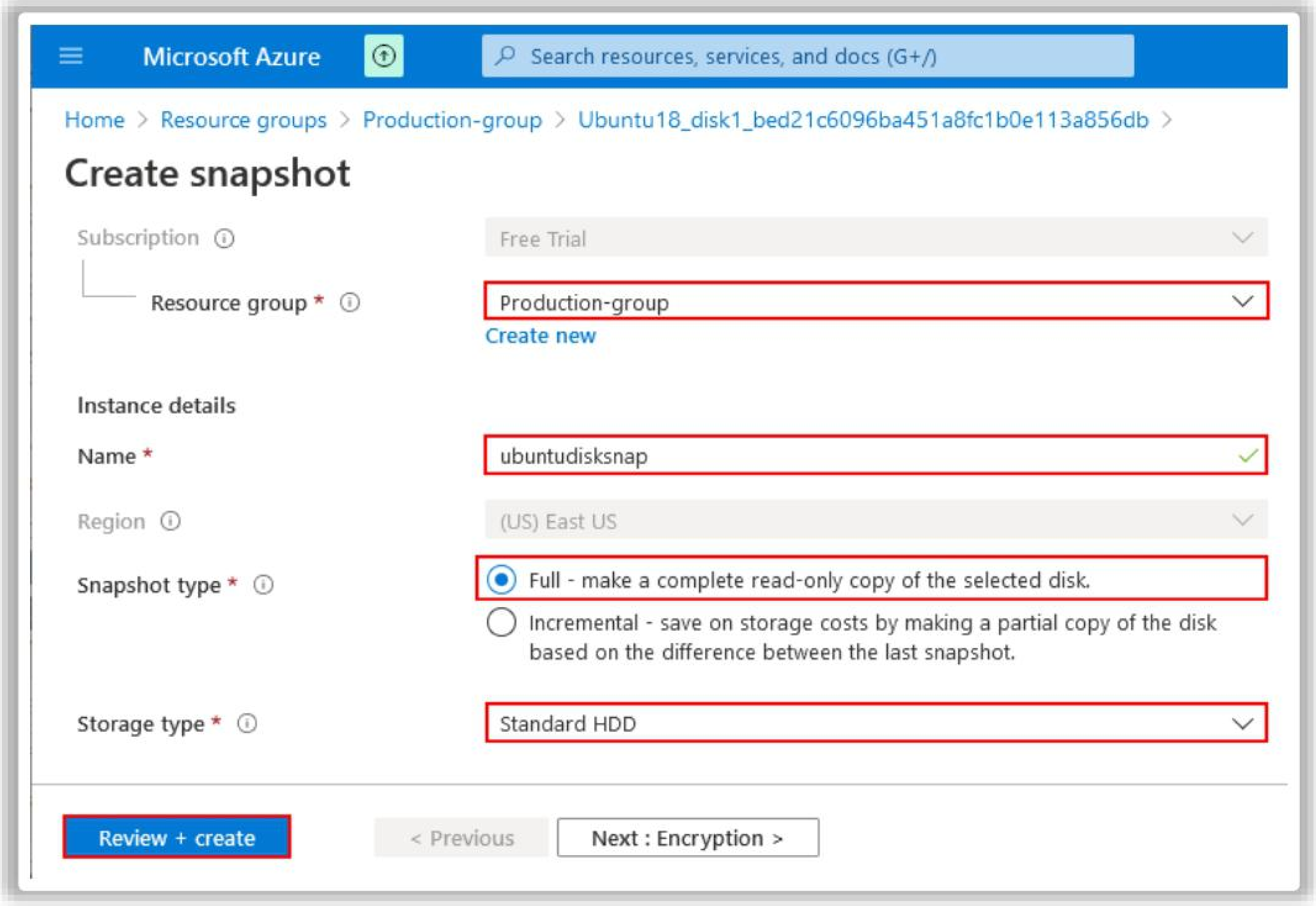


Figure 8.24: Clicking on the Create Snapshot Button

4. The Create Snapshot page would appear. Provide the following details and then click on the Review+create button
   - Give a desired name for the OS snapshot (here, it is ubuntudisksnap)
   - Select the snapshot type as read-only

- Select a storage type (here, it is standard HDD)



Home > Resource groups > Production-group > Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db >

Create snapshot

Subscription ⓘ: Free Trial

Resource group * ⓘ: Production-group

Create new

Instance details

Name *: ubuntudisksnap ✓

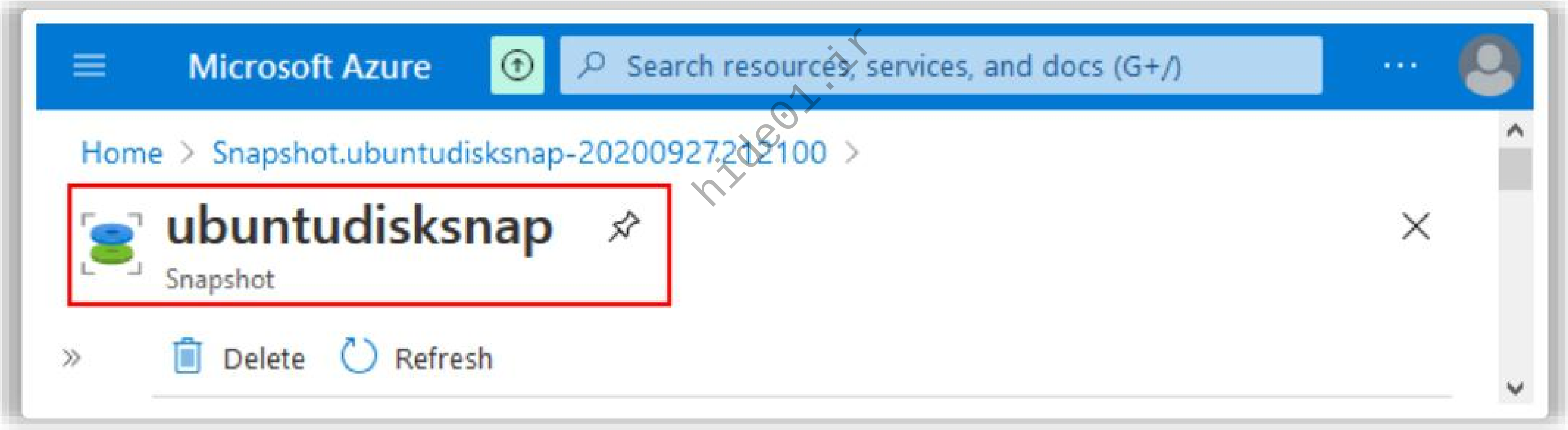
Region ⓘ: (US) East US

Snapshot type * ⓘ:
● Full - make a complete read-only copy of the selected disk.
○ Incremental - save on storage costs by making a partial copy of the disk based on the difference between the last snapshot.

Storage type * ⓘ: Standard HDD

Figure 8.25: Configuring Details on the Create Snapshot Page

5. Click on Create in the next page. The snapshot is now successfully created.

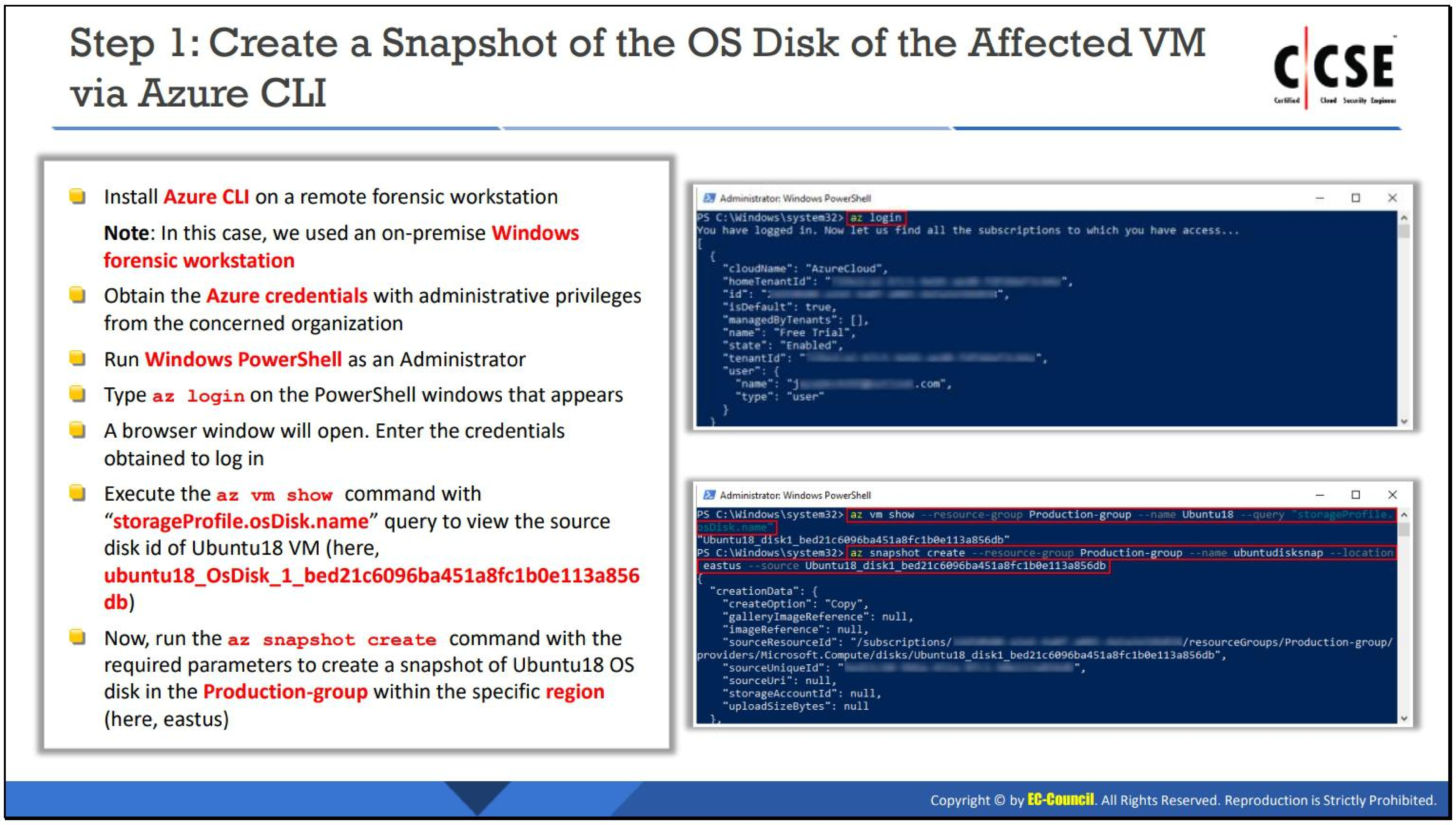


Home > Snapshot.ubuntudisksnap-20200927212100 >

ubuntudisksnap

Snapshot

Figure 8.26: Snapshot Named ubuntudisksnap is Successfully Created



Step 1: Create a Snapshot of the OS Disk of the Affected VM via Azure CLI

- Install **Azure CLI** on a remote forensic workstation

  Note: In this case, we used an on-premise **Windows forensic workstation**
- Obtain the **Azure credentials** with administrative privileges from the concerned organization
- Run **Windows PowerShell** as an Administrator
- Type **`az login`** on the PowerShell windows that appears
- A browser window will open. Enter the credentials obtained to log in
- Execute the **`az vm show`** command with "**storageProfile.osDisk.name**" query to view the source disk id of Ubuntu18 VM (here, **ubuntu18_OsDisk_1_bed21c6096ba451a8fc1b0e113a856db**)
- Now, run the **`az snapshot create`** command with the required parameters to create a snapshot of Ubuntu18 OS disk in the **Production-group** within the specific **region** (here, eastus)

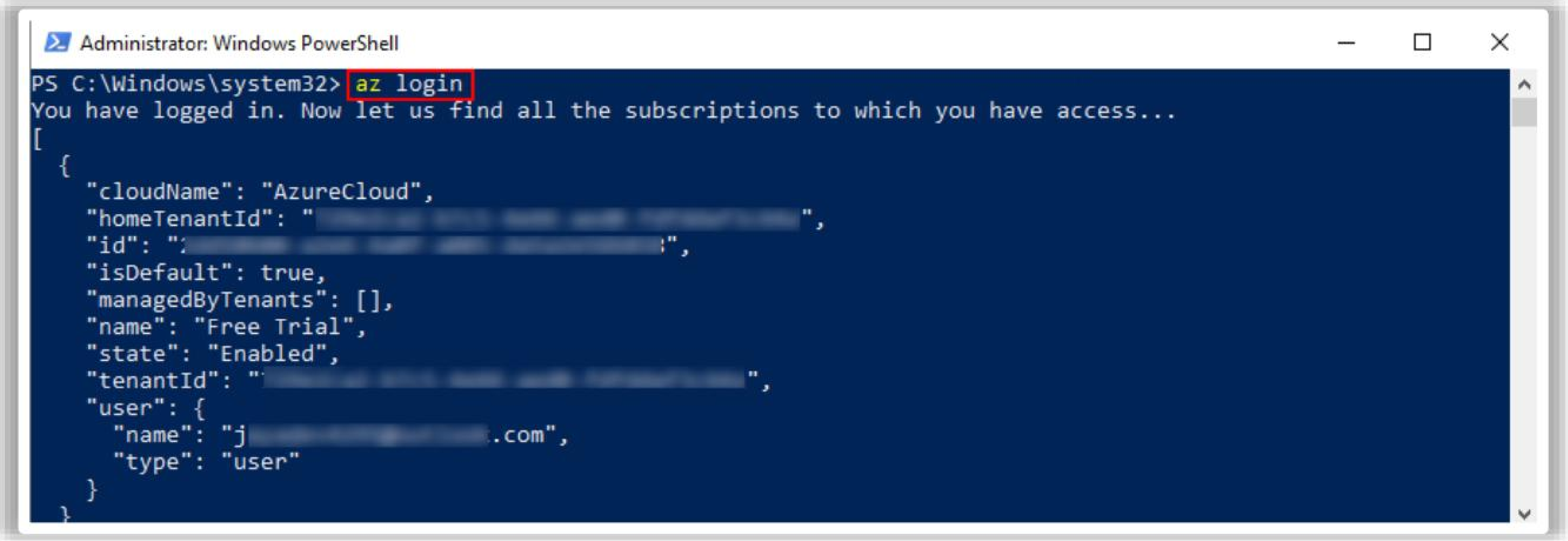
```
PS C:\Windows\system32> az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "[redacted]",
    "id": "[redacted]",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Free Trial",
    "state": "Enabled",
    "tenantId": "[redacted]",
    "user": {
      "name": "j[redacted].com",
      "type": "user"
    }
  }
```

```
PS C:\Windows\system32> az vm show --resource-group Production-group --name Ubuntu18 --query "storageProfile.
osDisk.name"
"Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db"
PS C:\Windows\system32> az snapshot create --resource-group Production-group --name ubuntudisksnap --location
 eastus --source Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db
{
  "creationData": {
    "createOption": "Copy",
    "galleryImageReference": null,
    "imageReference": null,
    "sourceResourceId": "/subscriptions/[redacted]/resourceGroups/Production-group/
providers/Microsoft.Compute/disks/Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db",
    "sourceUniqueId": "[redacted]",
    "sourceUri": null,
    "storageAccountId": null,
    "uploadSizeBytes": null
  },
```

Step 1: Create a Snapshot of the OS Disk of the Affected VM via Azure CLI

- Install Azure CLI on a remote forensic workstation

  Note: In this case, we used an on-premise Windows forensic workstation
- Obtain the Azure credentials with administrative privileges from the concerned organization.
- Run Windows PowerShell as an Administrator and type **`az login`** on the PowerShell windows that appears.
- A browser window will open. Enter the credentials obtained to log in and you will be logged into the concerned Azure account.

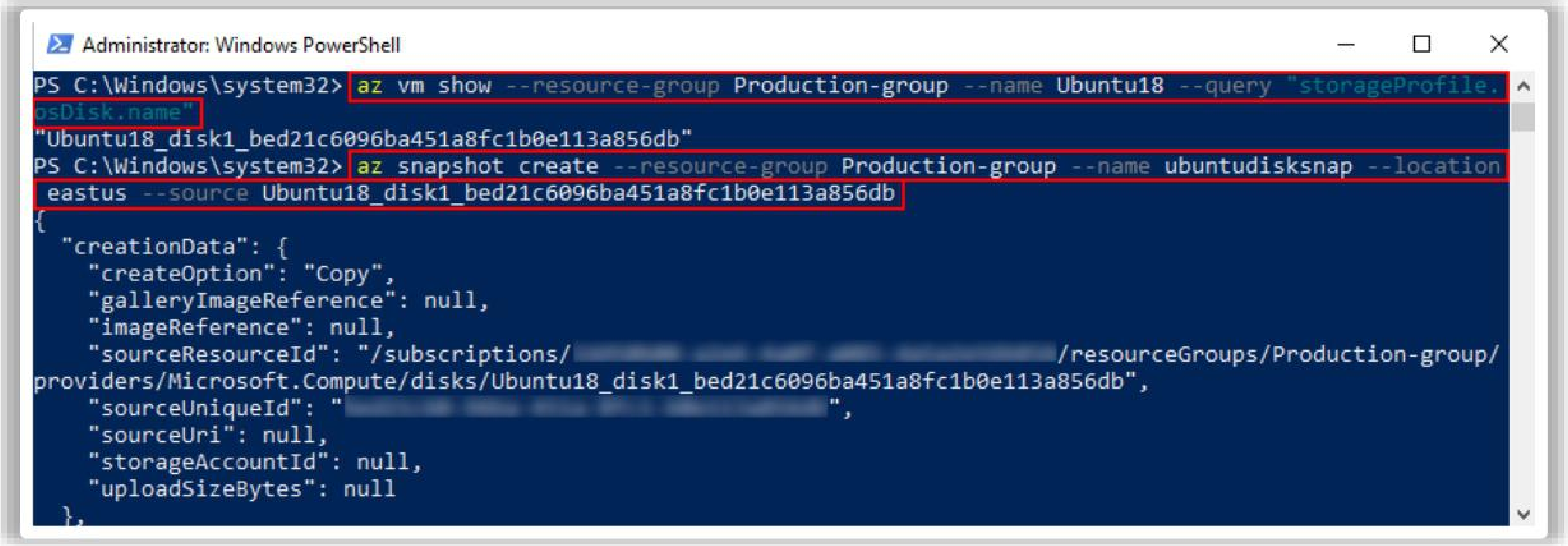


```
PS C:\Windows\system32> az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "[redacted]",
    "id": "[redacted]",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Free Trial",
    "state": "Enabled",
    "tenantId": "[redacted]",
    "user": {
      "name": "j[redacted].com",
      "type": "user"
    }
  }
```

Figure 8.27: Logging into the Azure Account Via az Login Command

- Execute the **`az vm show`** command with “storageProfile.osDisk.name” query to view the source disk id of Ubuntu18 VM. In this scenario, the source disk ID is found to be ubuntu18_OsDisk_1_bed21c6096ba451a8fc1b0e113a856db
- Now, run the **`az snapshot create`** command with the required parameters to create a snapshot of Ubuntu18 OS disk in the Production-group within the specific region.

  In this scenario, eastus region is used and the snapshot is named as ubuntudisksnap.



```
PS C:\Windows\system32> az vm show --resource-group Production-group --name Ubuntu18 --query "storageProfile.
osDisk.name"
"Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db"
PS C:\Windows\system32> az snapshot create --resource-group Production-group --name ubuntudisksnap --location
 eastus --source Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db
{
  "creationData": {
    "createOption": "Copy",
    "galleryImageReference": null,
    "imageReference": null,
    "sourceResourceId": "/subscriptions/[redacted]/resourceGroups/Production-group/
providers/Microsoft.Compute/disks/Ubuntu18_disk1_bed21c6096ba451a8fc1b0e113a856db",
    "sourceUniqueId": "[redacted]",
    "sourceUri": null,
    "storageAccountId": null,
    "uploadSizeBytes": null
  },
```

Figure 8.28: Creating a Snapshot Named ubuntudisksnap of the Affected VM Via Azure CLI

Step 2: Copy the Snapshot to a Storage Account under a Different Resource Group

- The concerned company maintains a resource group called **Security-group** for incident response; therefore, we need to copy the snapshot to a **storage account** under that resource group **using Azure CLI**
- First, the **type of storage** where the **snapshot** would be **copied** (such as Azure files or Azure blobs) must be determined
- In this case, we will be using **file share,** which can be directly **mounted** on the on-premise Windows **forensic workstation**

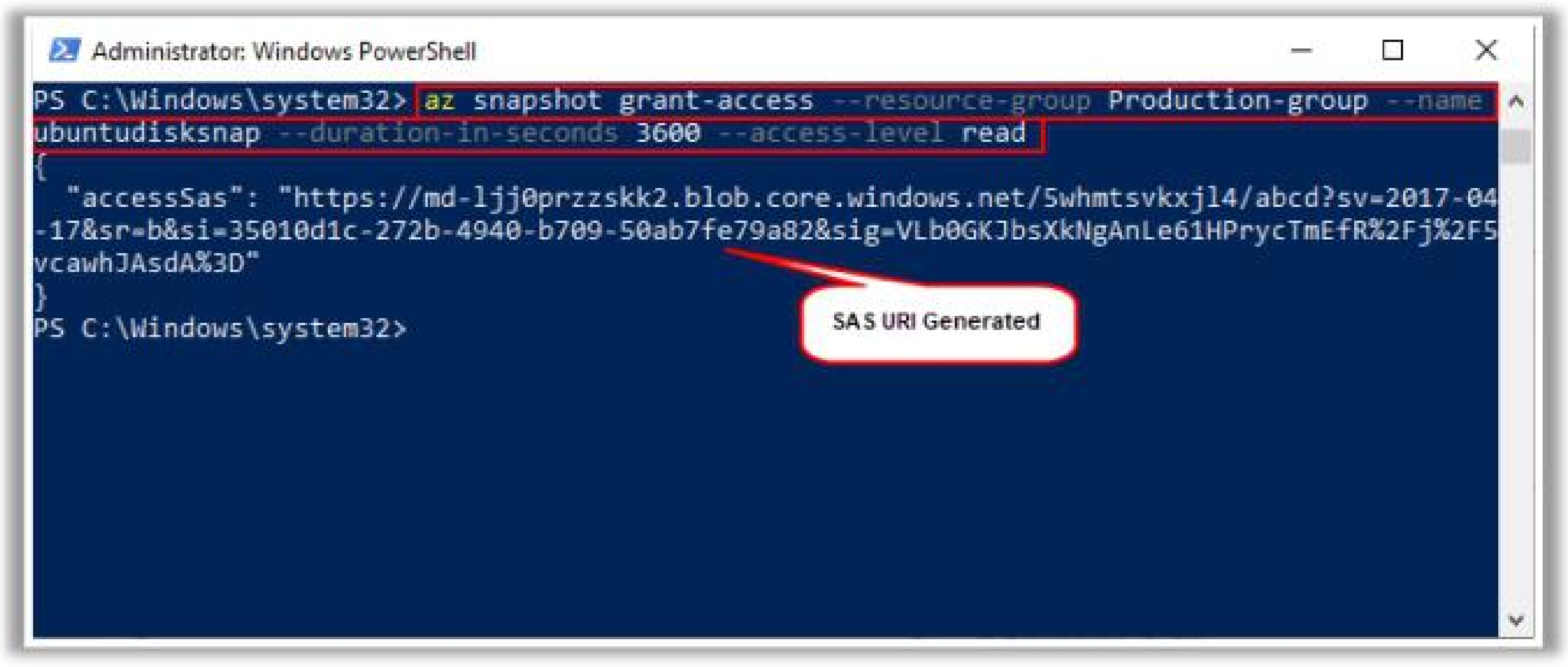
Set the required parameters

- **Source resource group name:** Production-group (name of the **group** where the **snapshot exists**)
- **Destination resource group name:** Security-group (name of the **group** where the **snapshot** would be **copied to**)
- **Account name:** chfisecurity (name of the **storage account** under **security-group**)
- **Destination file share name:** chfishare (name of the **file share** where the snapshot would be copied)
- **Destination file path:** disksnapshot.dd (name of the **snapshot in fileshare**)

Step 2: Copy the Snapshot to a Storage Account under a Different Resource Group (Cont'd)

Generate Shared Access Signature (SAS)

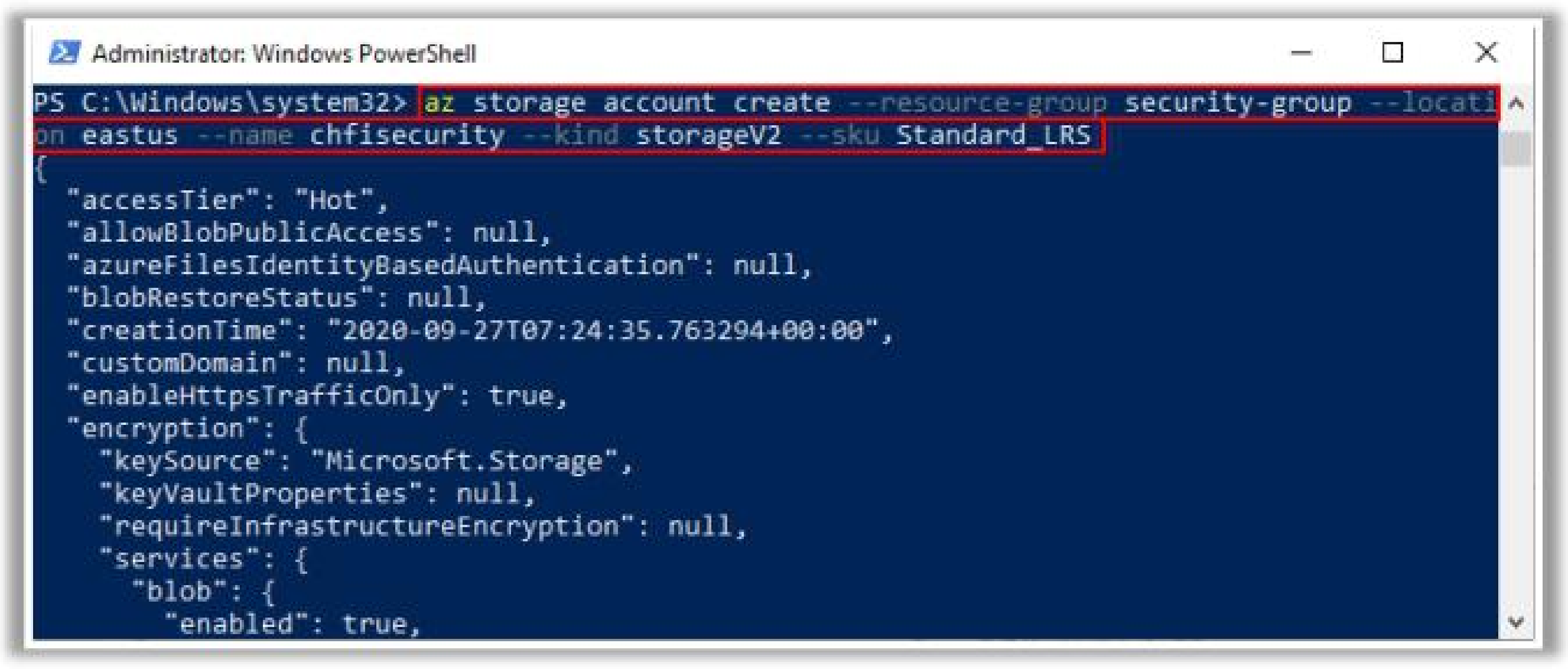
- Run the `az snapshot grant-access` command to gain time-specific access rights to the snapshot
- The command in the screenshot generated a **Shared Access Signature (SAS) token** with **read-only access** rights to the snapshot for **60 minutes**



```
PS C:\Windows\system32> az snapshot grant-access --resource-group Production-group --name
ubuntudisksnap --duration-in-seconds 3600 --access-level read
{
  "accessSas": "https://md-ljj0przzskk2.blob.core.windows.net/5whmtsvkxjl4/abcd?sv=2017-04
-17&sr=b&si=35010d1c-272b-4940-b709-50ab7fe79a82&sig=VLb0GKJbsXkNgAnLe61HPrycTmEfR%2Fj%2F5
vcawhJAsdA%3D"
}
PS C:\Windows\system32>
```

Create the storage account

- Run the `az storage account create` command to create the storage account **chfisecurity** where the snapshot would be copied
- While creating the storage account, mention the **resource group**, **region**, **storage kind,** and **type**

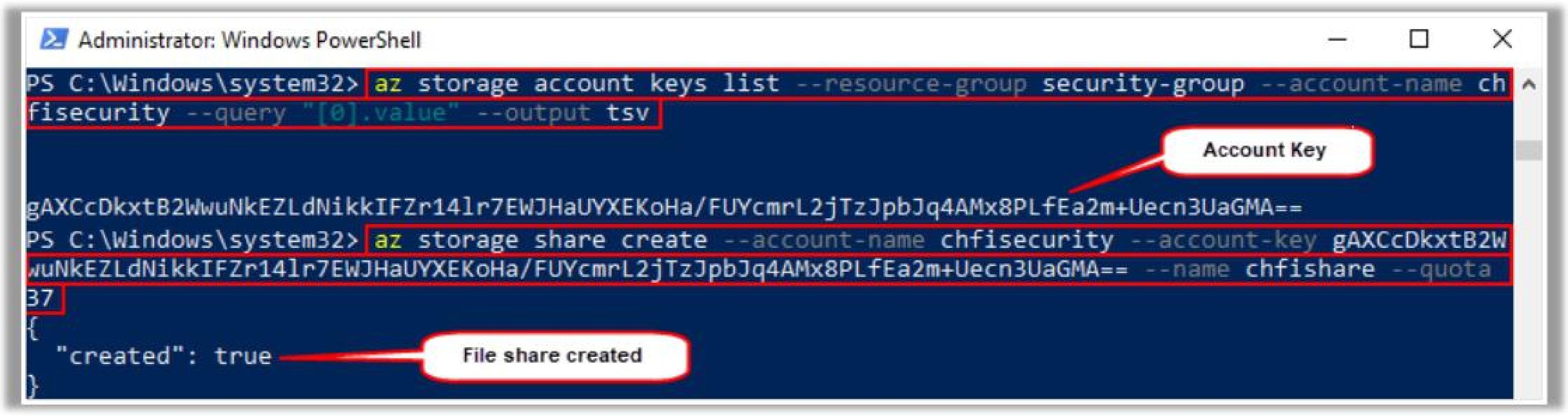


```
PS C:\Windows\system32> az storage account create --resource-group security-group --locati
on eastus --name chfisecurity --kind storageV2 --sku Standard_LRS
{
  "accessTier": "Hot",
  "allowBlobPublicAccess": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2020-09-27T07:24:35.763294+00:00",
  "customDomain": null,
  "enableHttpsTrafficOnly": true,
  "encryption": {
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
        "enabled": true,
```

Step 2: Copy the Snapshot to a Storage Account under a Different Resource Group (Cont'd)

Create file share within the storage account

- You need to obtain the **account keys** for the storage account **chfisecurity** to create the file share
- Execute the **`az storage account keys list`** command to view the account key
- Once the key is obtained, create the file share **chfishare** with the **`az storage share create`** command with required parameters

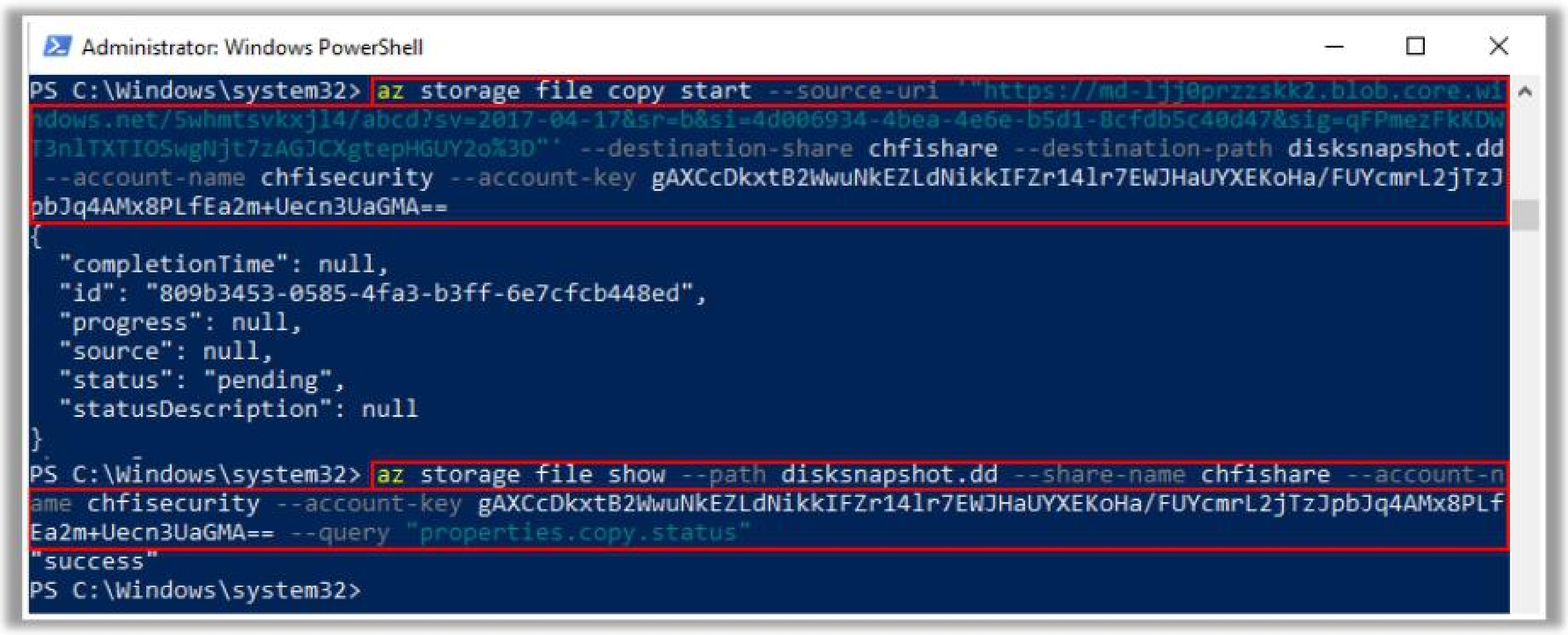


```
PS C:\Windows\system32> az storage account keys list --resource-group security-group --account-name ch
fisecurity --query "[0].value" --output tsv


gAXCcDkxtB2WwuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJpbJq4AMx8PLfEa2m+Uecn3UaGMA==
PS C:\Windows\system32> az storage share create --account-name chfisecurity --account-key gAXCcDkxtB2W
wuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJpbJq4AMx8PLfEa2m+Uecn3UaGMA== --name chfishare --quota
37
{
  "created": true
}
```

Step 2: Copy the Snapshot to a Storage Account under a Different Resource Group (Cont'd)

Copy the snapshot to the file share

- Now copy the snapshot to the **chfishare** by executing the **`az storage file copy start`** command with required parameters, such as **source URI** (the Access SAS generated for accessing the snapshot), **destination file path** (snapshot name with dd extension), **storage** account **name** and, account **key**
- Run the **`az storage file show`** command with the query '**properties.copy.status**' to determine whether the **dd** file was successfully copied to chfishare



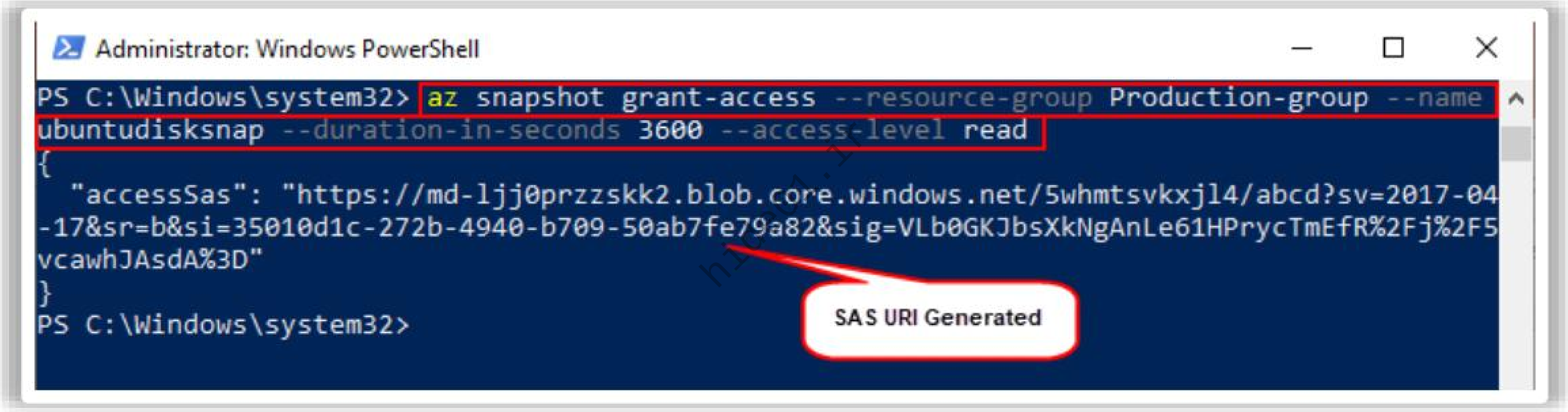
```
PS C:\Windows\system32> az storage file copy start --source-uri '"https://md-ljj0przzskk2.blob.core.wi
ndows.net/5whmtsvkxjl4/abcd?sv=2017-04-17&sr=b&si=4d006934-4bea-4e6e-b5d1-8cfdb5c40d47&sig=qFPmezFkKDW
T3nlTXTIOSwgNjt7zAGJCXgtepHGUY2o%3D"' --destination-share chfishare --destination-path disksnapshot.dd
 --account-name chfisecurity --account-key gAXCcDkxtB2WwuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJ
pbJq4AMx8PLfEa2m+Uecn3UaGMA==
{
  "completionTime": null,
  "id": "809b3453-0585-4fa3-b3ff-6e7cfcb448ed",
  "progress": null,
  "source": null,
  "status": "pending",
  "statusDescription": null
}
PS C:\Windows\system32> az storage file show --path disksnapshot.dd --share-name chfishare --account-n
ame chfisecurity --account-key gAXCcDkxtB2WwuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJpbJq4AMx8PLf
Ea2m+Uecn3UaGMA== --query "properties.copy.status"
"success"
PS C:\Windows\system32>
```

Step 2: Copy the Snapshot to a Storage Account under a Different Resource Group

As per the assumed scenario, the concerned company maintains a resource group called Security-group for incident response; therefore, we need to copy the snapshot to a storage account under that resource group using Azure CLI. First, the type of storage where the snapshot would be copied (such as Azure files or Azure blobs) must be determined. In this case, we will be using file share, which can be directly mounted on the on-premise Windows forensic workstation.

1. **Set the Required Parameters**
   - **Source Resource Group Name:** Production-group (name of the group where the snapshot exists)
   - **Destination Resource Group Name:** Security-group (name of the group where the snapshot would be copied to)
   - **Account Name:** chfisecurity (name of the storage account under security-group)
   - **Destination File Share Name:** chfishare (name of the file share where the snapshot would be copied)
   - **Destination File Path:** disksnapshot.dd (name of the snapshot in fileshare)
2. **Generate Shared Access Signature (SAS)**

   Run the **`az snapshot grant-access`** command to gain time-specific access rights to the snapshot. The command in the screenshot below has generated a Shared Access Signature (SAS) token with read-only access rights to the snapshot for 60 minutes.

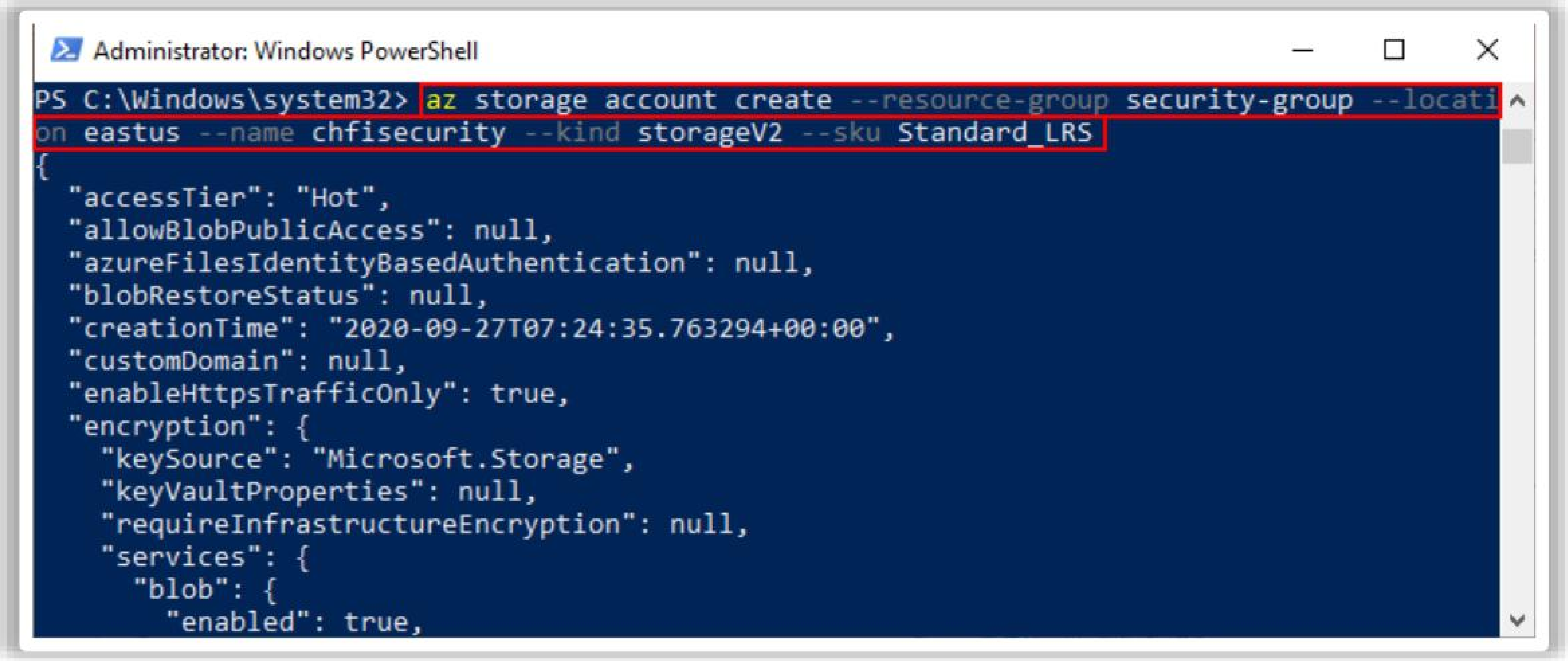


```
PS C:\Windows\system32> az snapshot grant-access --resource-group Production-group --name
ubuntudisksnap --duration-in-seconds 3600 --access-level read
{
  "accessSas": "https://md-ljj0przzskk2.blob.core.windows.net/5whmtsvkxjl4/abcd?sv=2017-04
-17&sr=b&si=35010d1c-272b-4940-b709-50ab7fe79a82&sig=VLb0GKJbsXkNgAnLe61HPrycTmEfR%2Fj%2F5
vcawhJAsdA%3D"
}
PS C:\Windows\system32>
```

Figure 8.29: Generating Shared Access Signature

3. **Create the Storage Account**

   Run the **`az storage account create`** command to create the storage account chfisecurity where the snapshot would be copied. While creating the storage account, mention the resource group, region, storage kind, and type.

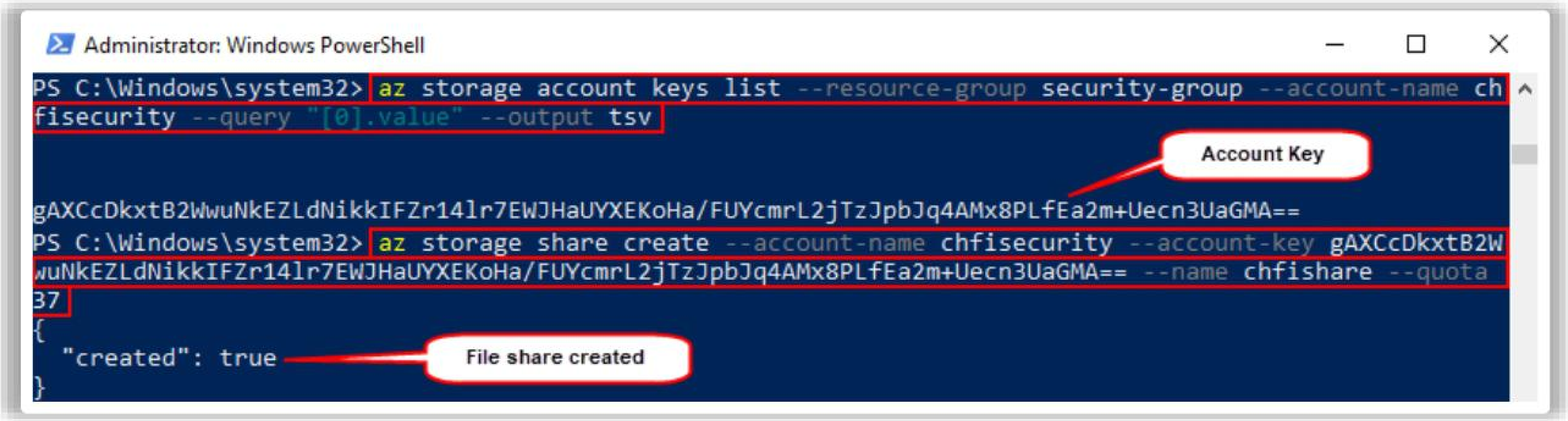


```
PS C:\Windows\system32> az storage account create --resource-group security-group --locati
on eastus --name chfisecurity --kind storageV2 --sku Standard_LRS
{
  "accessTier": "Hot",
  "allowBlobPublicAccess": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2020-09-27T07:24:35.763294+00:00",
  "customDomain": null,
  "enableHttpsTrafficOnly": true,
  "encryption": {
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
        "enabled": true,
```

Figure 8.30: Creating a Storage Account Named chfisecurity

4. Create File Share Within The Storage Account

You need to obtain the account keys for the storage account chfisecurity to create the file share. Execute the **`az storage account keys list`** command to view the account key. Once the key is obtained, create the file share chfishare with the **`az storage share create`** command with required parameters as shown below:



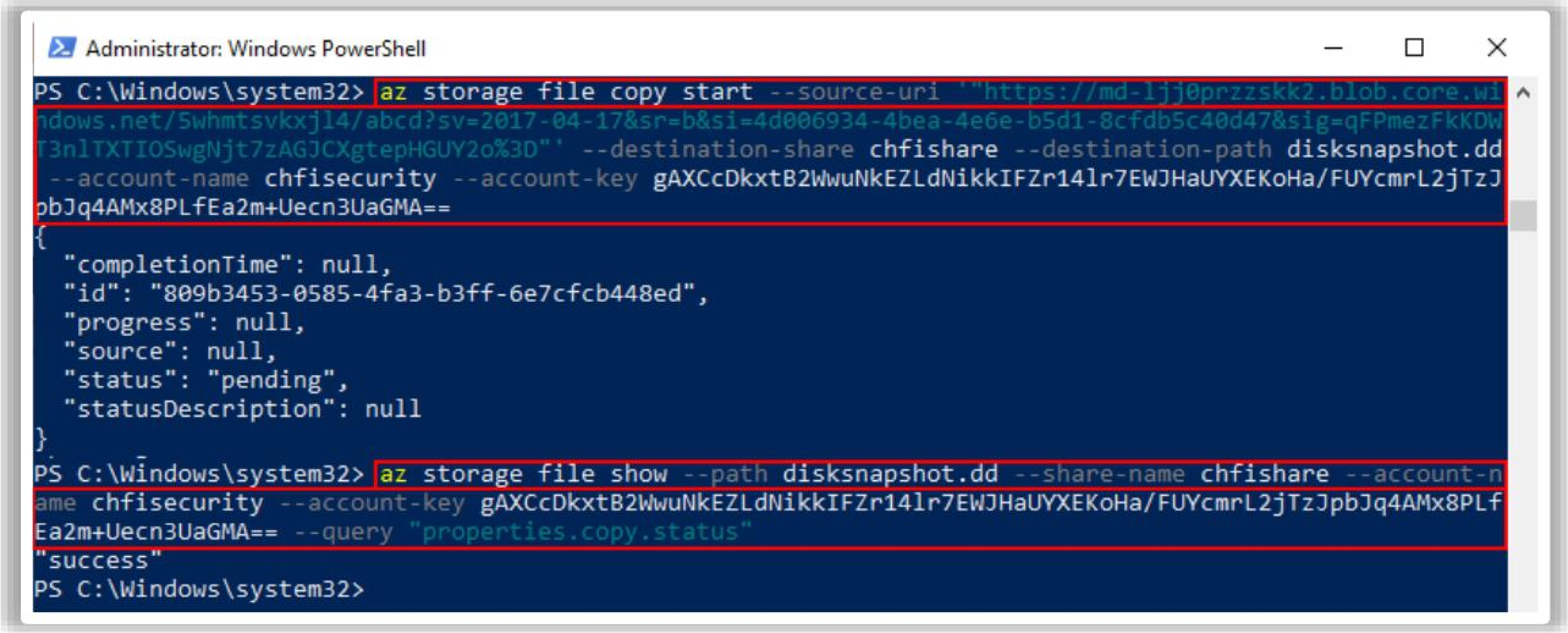
```
PS C:\Windows\system32> az storage account keys list --resource-group security-group --account-name ch
fisecurity --query "[0].value" --output tsv


gAXCcDkxtB2WwuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJpbJq4AMx8PLfEa2m+Uecn3UaGMA==
PS C:\Windows\system32> az storage share create --account-name chfisecurity --account-key gAXCcDkxtB2W
wuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJpbJq4AMx8PLfEa2m+Uecn3UaGMA== --name chfishare --quota
37
{
  "created": true
}
```

Figure 8.31: Creating a File Share Named chfishare Within chfisecurity

5. Copy the Snapshot to the File Share

Now copy the snapshot to the chfishare by executing the **`az storage file copy start`** command with required parameters, such as source URI (the Access SAS generated for accessing the snapshot), destination file path (snapshot name with dd extension), storage account name and, account key.

It might take some time to copy the snapshot to the file share. Run the **`az storage file show`** command with the query 'properties.copy.status' to determine whether the dd file was successfully copied to chfishare. When the file is successfully copied, you will get the output as 'success'.



```
PS C:\Windows\system32> az storage file copy start --source-uri '"https://md-1jj0przzskk2.blob.core.wi
ndows.net/5whmtsvkxjl4/abcd?sv=2017-04-17&sr=b&si=4d006934-4bea-4e6e-b5d1-8cfdb5c40d47&sig=qFPmezFkKDW
T3nlTXTIOSwgNjt7zAGJCXgtepHGUY2o%3D"' --destination-share chfishare --destination-path disksnapshot.dd
 --account-name chfisecurity --account-key gAXCcDkxtB2WwuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJ
pbJq4AMx8PLfEa2m+Uecn3UaGMA==
{
  "completionTime": null,
  "id": "809b3453-0585-4fa3-b3ff-6e7cfcb448ed",
  "progress": null,
  "source": null,
  "status": "pending",
  "statusDescription": null
}
PS C:\Windows\system32> az storage file show --path disksnapshot.dd --share-name chfishare --account-n
ame chfisecurity --account-key gAXCcDkxtB2WwuNkEZLdNikkIFZr14lr7EWJHaUYXEKoHa/FUYcmrL2jTzJpbJq4AMx8PLf
Ea2m+Uecn3UaGMA== --query "properties.copy.status"
"success"
PS C:\Windows\system32>
```

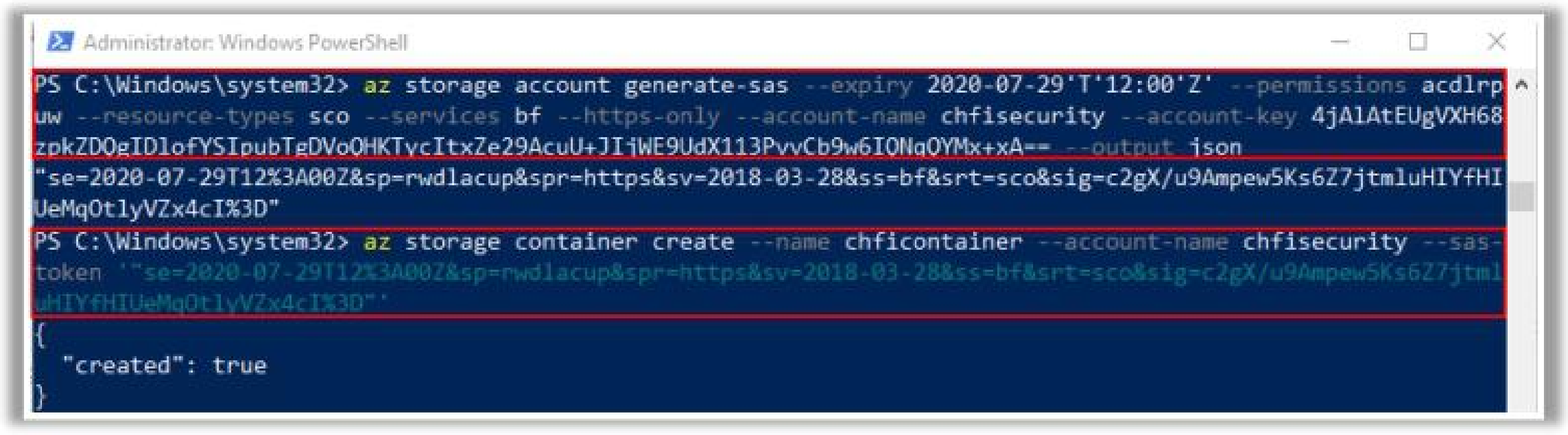
Figure 8.32: Copying the Snapshot to chfishare and Checking the Copy Status

Step 3: Delete the Snapshot from the Source Resource Group and Create a Backup Copy

- The snapshot **ubuntudisksnap** would still exist in the **Production-group** even after being copied to a storage account
- It is recommended to create a **backup copy** of the **OS snapshot** of the affected VM and **remove** it from the **source resource group** for security purposes
- Here, we will create a backup copy of the snapshot in a **blob container** as a **page blob**

Create a backup copy of the snapshot

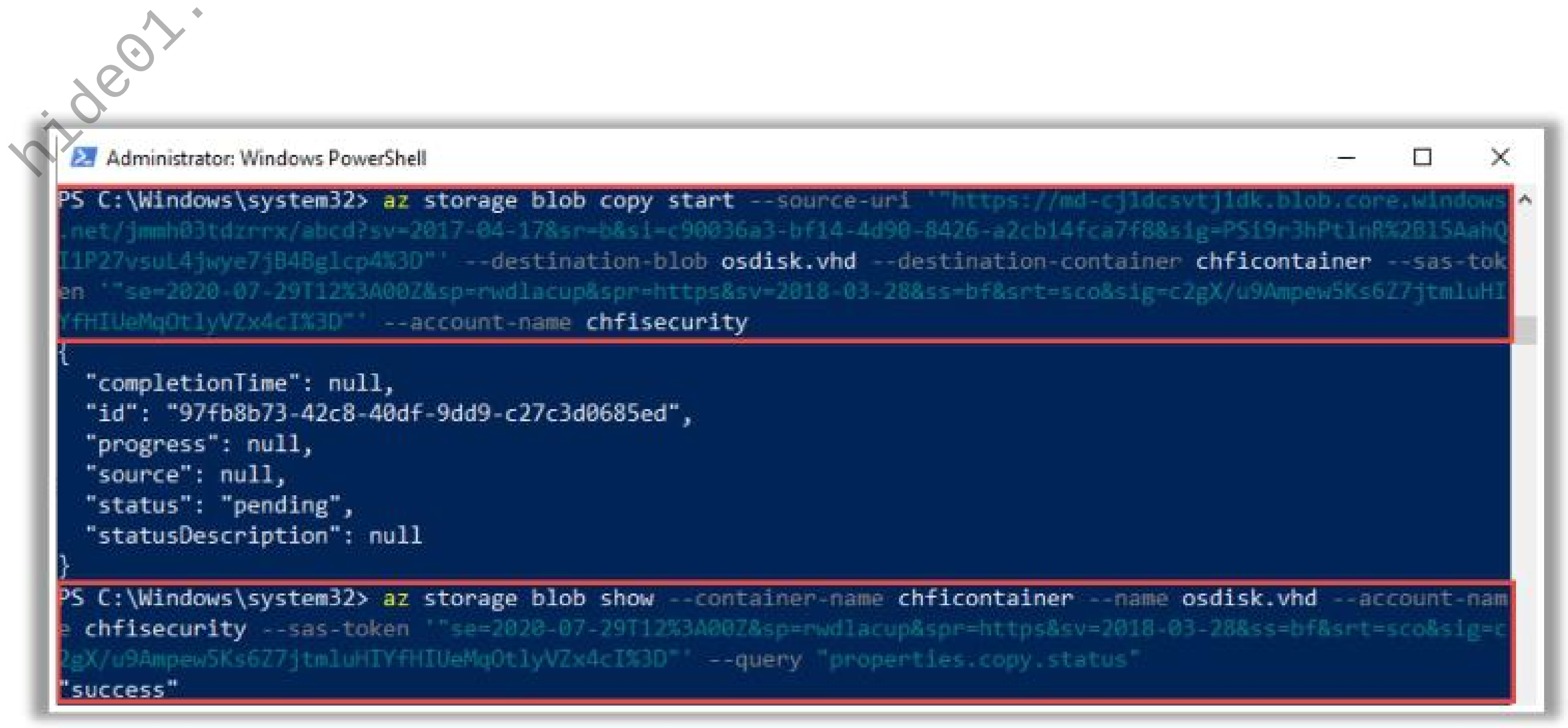
- Generate a **SAS token** for the storage account **chfisecurity** with required parameters (as demonstrated in the screenshot)
- Now, create a **blob container** (here, **chficontainer**) by using the SAS token for enhanced security



```
PS C:\Windows\system32> az storage account generate-sas --expiry 2020-07-29'T'12:00'Z' --permissions acdlrp
uw --resource-types sco --services bf --https-only --account-name chfisecurity --account-key 4jA1AtEUgVXH68
zpkZDQgIDlofYSIpubTgDVoQHKTycItxZe29AcuU+JIjWE9UdX113PvyCb9w6IQNqQYMx+xA== --output json
"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c2gX/u9Ampew5Ks6Z7jtmluHIYfHI
UeMqOtlyVZx4cI%3D"
PS C:\Windows\system32> az storage container create --name chficontainer --account-name chfisecurity --sas-
token '"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c2gX/u9Ampew5Ks6Z7jtml
uHIYfHIUeMqOtlyVZx4cI%3D"'
{
  "created": true
}
```

Step 3: Delete the Snapshot from the Source Resource Group and Create a Backup Copy (Cont'd)

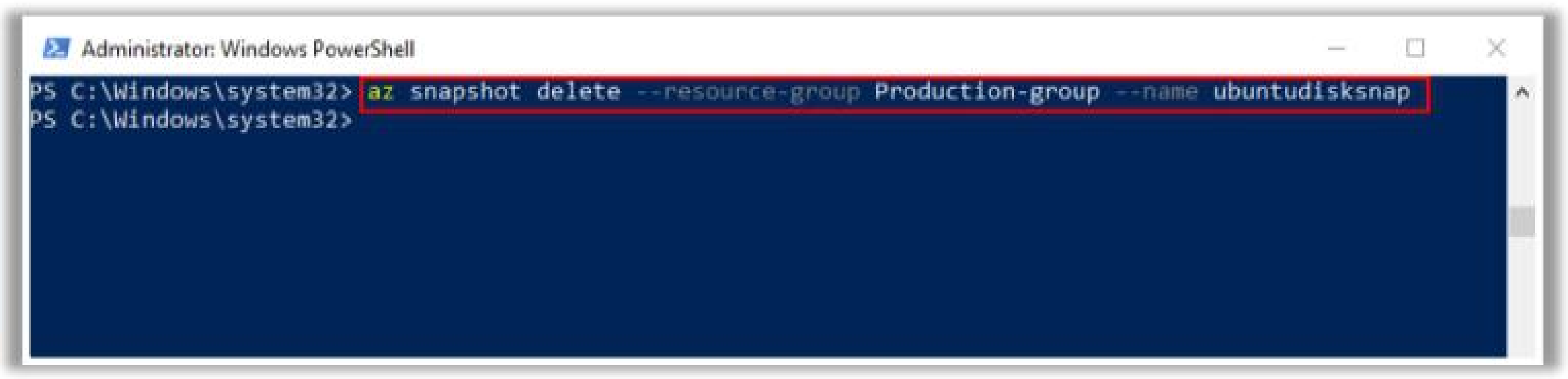
- Run the **`az storage blob copy start`** command to copy the snapshot to **chficontainer** as a page blob named **osdisk.vhd**
- Confirm whether the blob copy is created by executing the **`az storage blob show`** command with the query '**properties.copy.status**'



```
PS C:\Windows\system32> az storage blob copy start --source-uri '"https://md-cj1dcsvtj1dk.blob.core.windows
.net/jmmh03tdzrrx/abcd?sv=2017-04-17&sr=b&si=c90036a3-bf14-4d90-8426-a2cb14fca7f8&sig=PS19r3hPtlnR%2B15AahQ
I1P27vsuL4jwye7jB48glcp4%3D"' --destination-blob osdisk.vhd --destination-container chficontainer --sas-tok
en '"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c2gX/u9Ampew5Ks6Z7jtmluHI
YfHIUeMqOtlyVZx4cI%3D"' --account-name chfisecurity
{
  "completionTime": null,
  "id": "97fb8b73-42c8-40df-9dd9-c27c3d0685ed",
  "progress": null,
  "source": null,
  "status": "pending",
  "statusDescription": null
}
PS C:\Windows\system32> az storage blob show --container-name chficontainer --name osdisk.vhd --account-nam
e chfisecurity --sas-token '"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c
2gX/u9Ampew5Ks6Z7jtmluHIYfHIUeMqOtlyVZx4cI%3D"' --query "properties.copy.status"
"success"
```

Delete the snapshot from the source group

- Run **`az snapshot delete`** command to remove the snapshot from the Production-group



```
PS C:\Windows\system32> az snapshot delete --resource-group Production-group --name ubuntudisksnap
PS C:\Windows\system32>
```

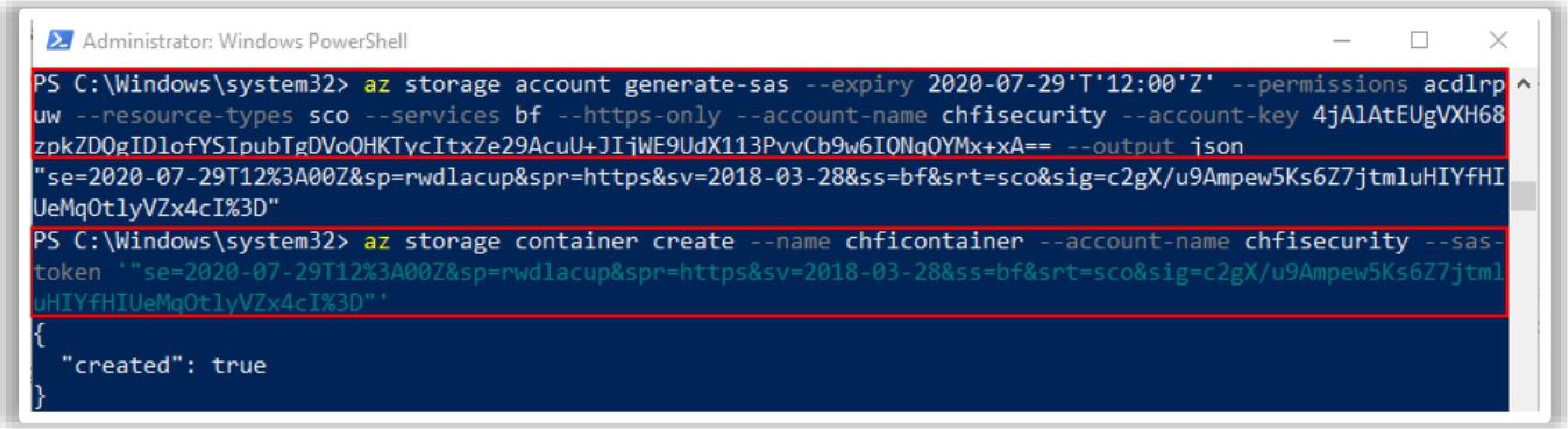
Step 3: Delete the Snapshot from the Source Resource Group and Create a Backup Copy

The snapshot ubuntudisksnap would still exist in the Production-group even after being copied to a storage account. It is recommended to create a backup copy of the OS snapshot of the affected VM and remove it from the source resource group for security purposes. Here, we will create a backup copy of the snapshot in a blob container as a page blob.

Creating a Backup Copy of The Snapshot

The steps for creating a backup copy of the snapshot are given below:

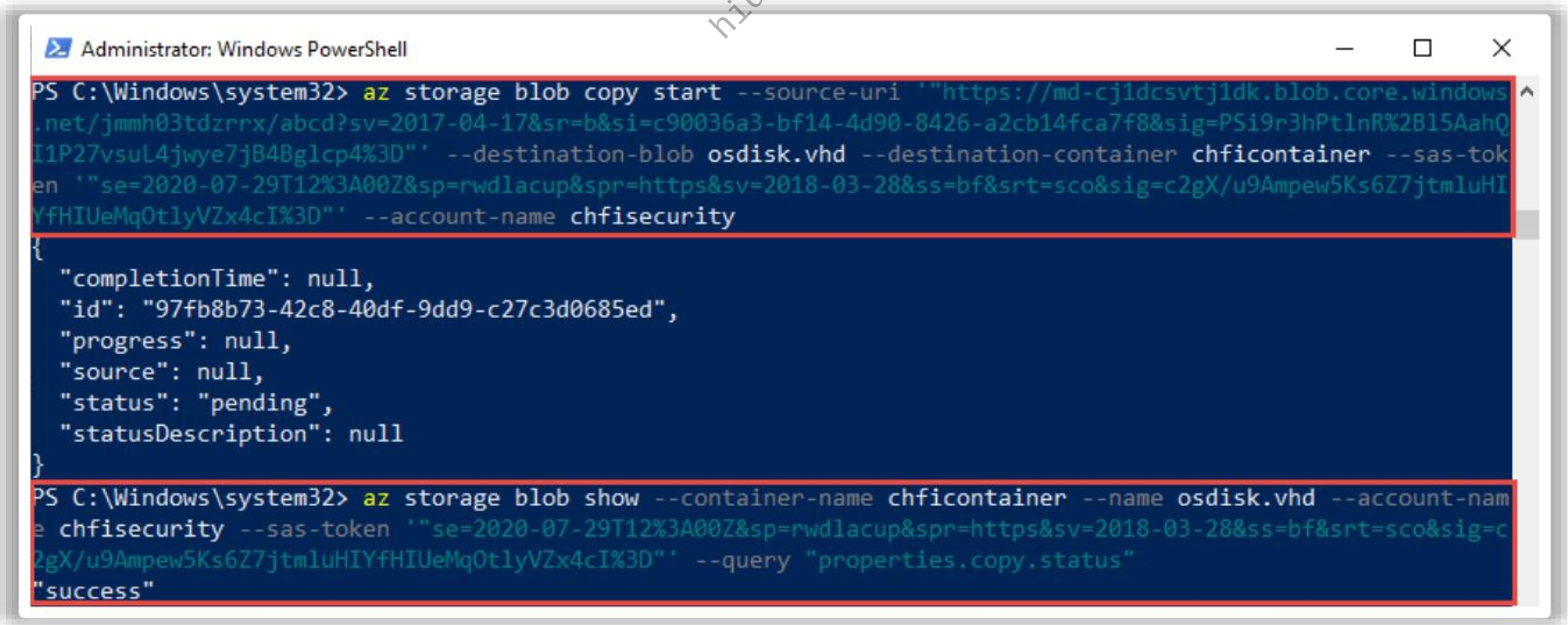
1. Generate a SAS token for the storage account chfisecurity with required parameters such as --expiry, --permissions, --resource types, --services, --account-name and --account-key
2. Now, create a blob container (here, chficontainer) by using the SAS token for enhanced security with **`az storage container create`** command



```
PS C:\Windows\system32> az storage account generate-sas --expiry 2020-07-29'T'12:00'Z' --permissions acdlrp
uw --resource-types sco --services bf --https-only --account-name chfisecurity --account-key 4jAlAtEUgVXH68
zpkZDQgIDlofYSIpubTgDVoQHKTycItxZe29AcuU+JIjWE9UdX113PvvCb9w6IQNqQYMx+xA== --output json
"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c2gX/u9Ampew5Ks6Z7jtmluHIYfHI
UeMqOtlyVZx4cI%3D"
PS C:\Windows\system32> az storage container create --name chficontainer --account-name chfisecurity --sas-
token '"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c2gX/u9Ampew5Ks6Z7jtml
uHIYfHIUeMqOtlyVZx4cI%3D"'
{
  "created": true
}
```

Figure 8.33: Generating SAS Token and Creating a Blob Container Named chficontainer

3. Run the **`az storage blob copy start`** command to copy the snapshot to chficontainer as a page blob named osdisk.vhd
4. Confirm whether the blob copy is created by executing the **`az storage blob show`** command with the query 'properties.copy.status'

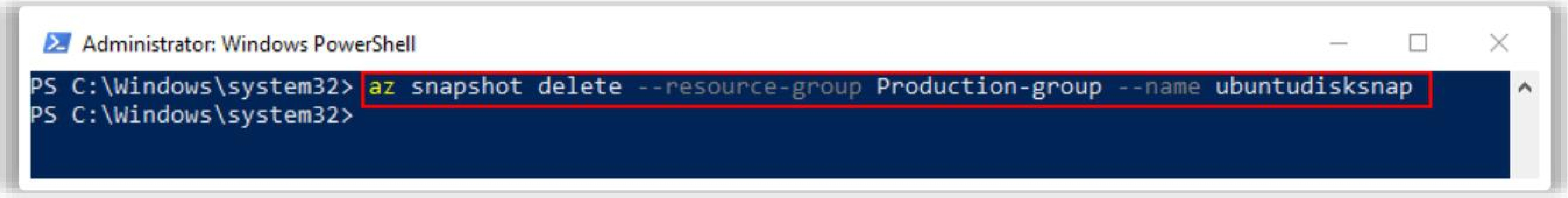


```
PS C:\Windows\system32> az storage blob copy start --source-uri '"https://md-cj1dcsvtj1dk.blob.core.windows
.net/jmmh03tdzrrx/abcd?sv=2017-04-17&sr=b&si=c90036a3-bf14-4d90-8426-a2cb14fca7f8&sig=PSi9r3hPtlnR%2Bl5AahQ
I1P27vsuL4jwye7jB4Bglcp4%3D"' --destination-blob osdisk.vhd --destination-container chficontainer --sas-tok
en '"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c2gX/u9Ampew5Ks6Z7jtmluHI
YfHIUeMqOtlyVZx4cI%3D"' --account-name chfisecurity
{
  "completionTime": null,
  "id": "97fb8b73-42c8-40df-9dd9-c27c3d0685ed",
  "progress": null,
  "source": null,
  "status": "pending",
  "statusDescription": null
}
PS C:\Windows\system32> az storage blob show --container-name chficontainer --name osdisk.vhd --account-nam
e chfisecurity --sas-token '"se=2020-07-29T12%3A00Z&sp=rwdlacup&spr=https&sv=2018-03-28&ss=bf&srt=sco&sig=c
2gX/u9Ampew5Ks6Z7jtmluHIYfHIUeMqOtlyVZx4cI%3D"' --query "properties.copy.status"
"success"
```

Figure 8.34: Copying the Snapshot to chficontainer and Checking the Copy Status

Deleting the Snapshot from the Source Group

Run **`az snapshot delete`** command to remove the snapshot from the production-group.



```
PS C:\Windows\system32> az snapshot delete --resource-group Production-group --name ubuntudisksnap
PS C:\Windows\system32>
```

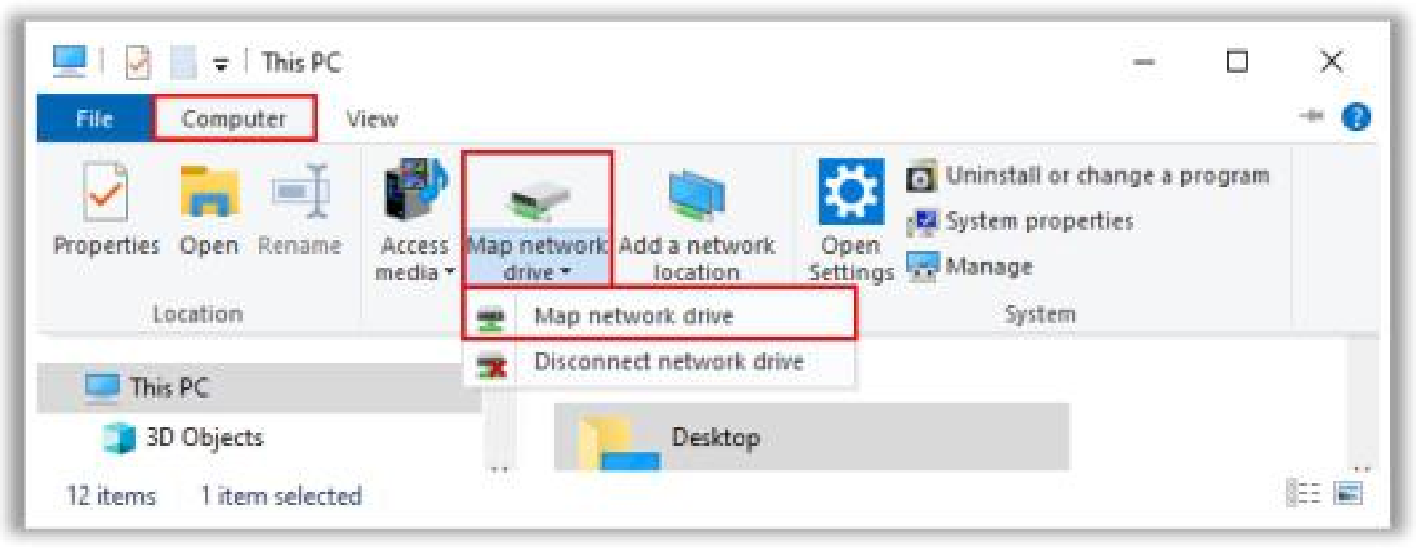
Figure 8.35: Deleting the Snapshot from Production-Group

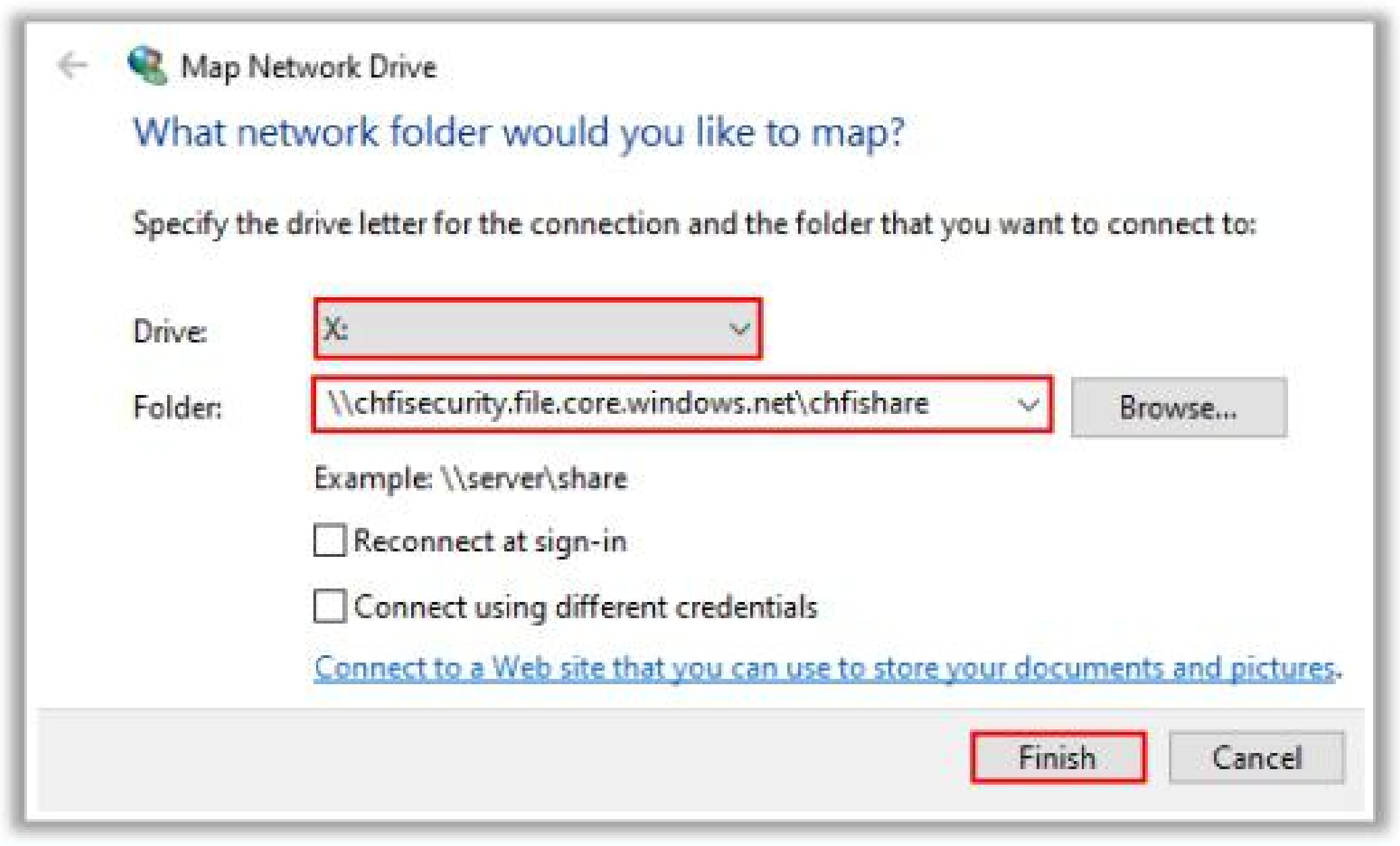
Step 4: Mount the Snapshot onto the Forensic Workstation

You can now mount the file share containing the **disksnapshot.dd** of the affected **Ubuntu18 VM** onto any forensic workstation to start the analysis (in this case, we used an on-premise **Windows forensic machine**)

Mounting the file share on the Windows Forensic workstation

1. Open **File Explorer** from the Start Menu or press Win+E key together
2. Select **This PC** from the left-side menu, click on **Computer,** and select **Map Network Drive**
3. Select the drive letter and provide the **UNC path** in file share (here, **\\chfisecurity.file.core.windows.net\\chfishare**)
4. If prompted, provide the **storage account name** as **username** and **storage key** as **password**




Map Network Drive

What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: X:

Folder: \\chfisecurity.file.core.windows.net\chfishare

Example: \\server\share

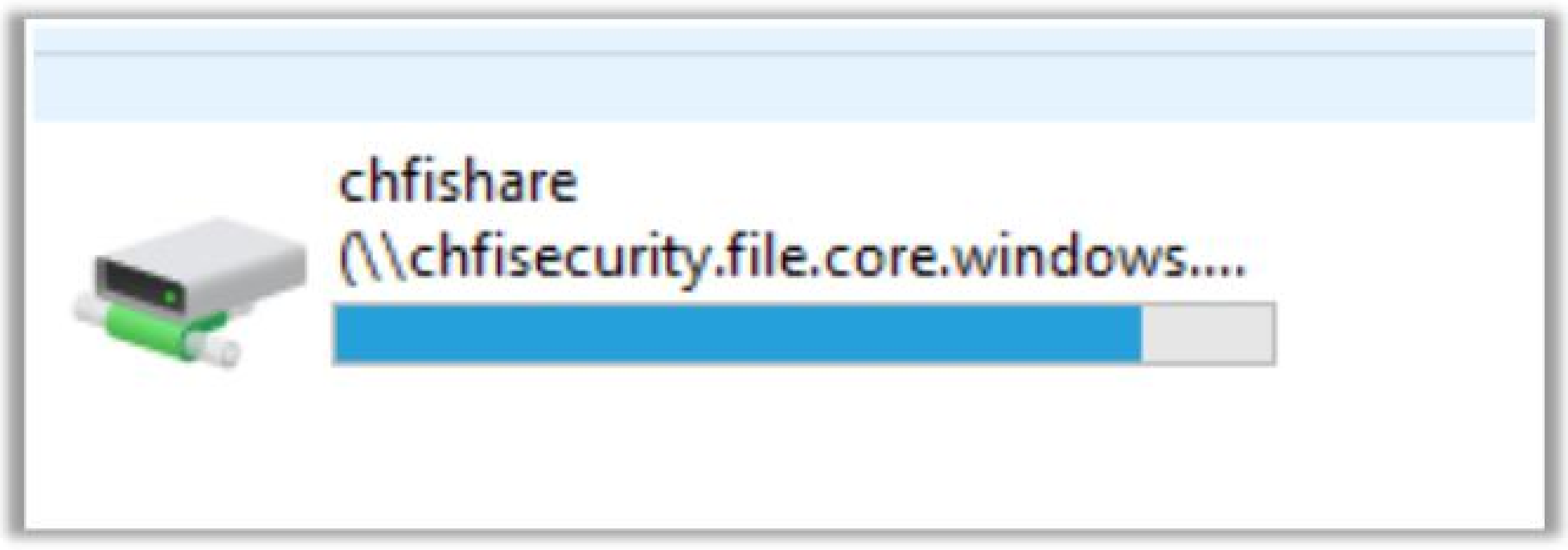
□ Reconnect at sign-in

□ Connect using different credentials

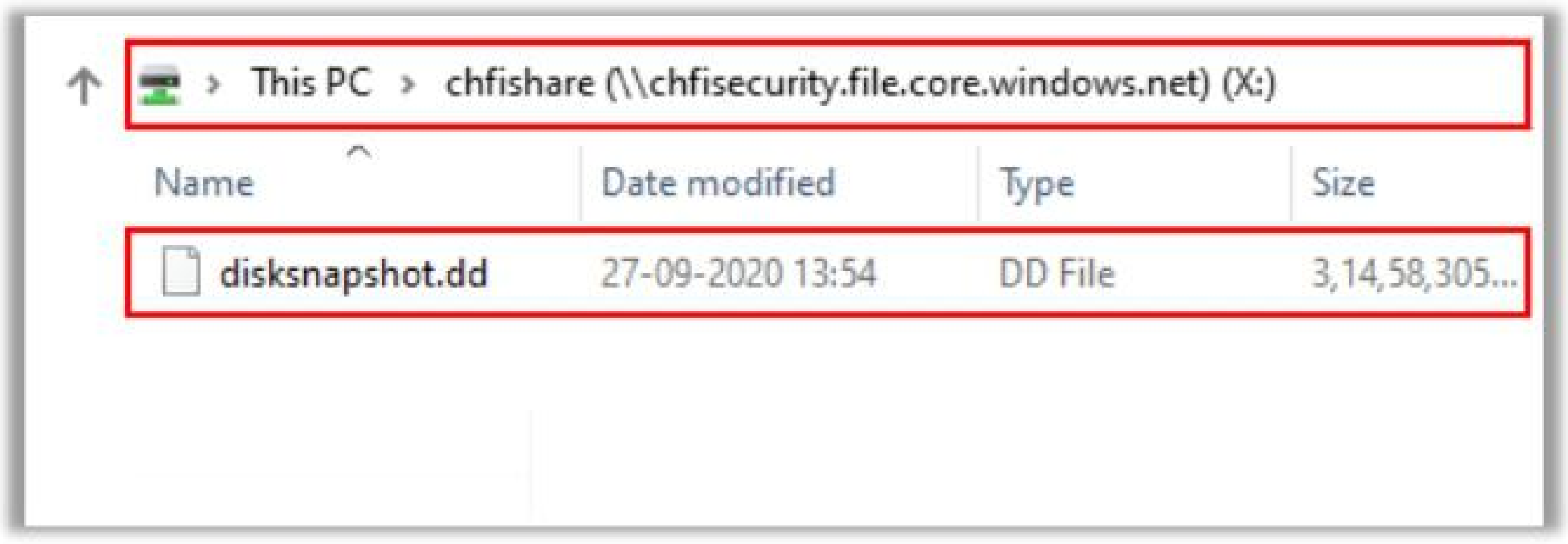
Connect to a Web site that you can use to store your documents and pictures.

Step 4: Mount the Snapshot onto the Forensic Workstation (Cont'd)

5. The file share **chfishare** is now mounted under Network Locations


chfishare
(\\chfisecurity.file.core.windows....

6. Double-click on the mounted share to view the dd file


This PC › chfishare (\\chfisecurity.file.core.windows.net) (X:)

Name	Date modified	Type	Size
disksnapshot.dd	27-09-2020 13:54	DD File	3,14,58,305...

Step 4: Mount the Snapshot onto the Forensic Workstation

You can now mount the file share containing the disksnapshot.dd of the affected Ubuntu18 VM onto any forensic workstation to start the analysis. In this case, we will be using an on-premise Windows forensic machine. Steps to mounting the file share on the Windows Forensic workstation are given below:

1. Open File Explorer from the Start Menu or press Win+E key together

2. Select This PC from the left-side menu, click on Computer, and select Map Network Drive

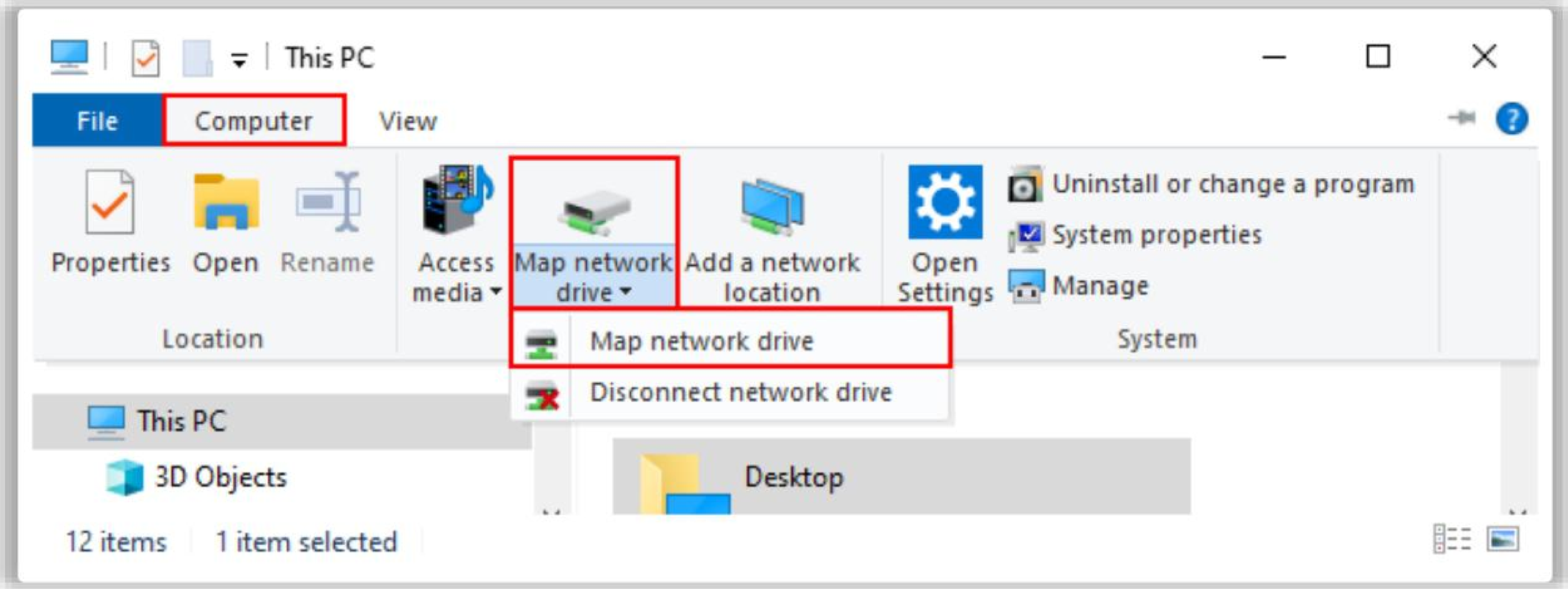
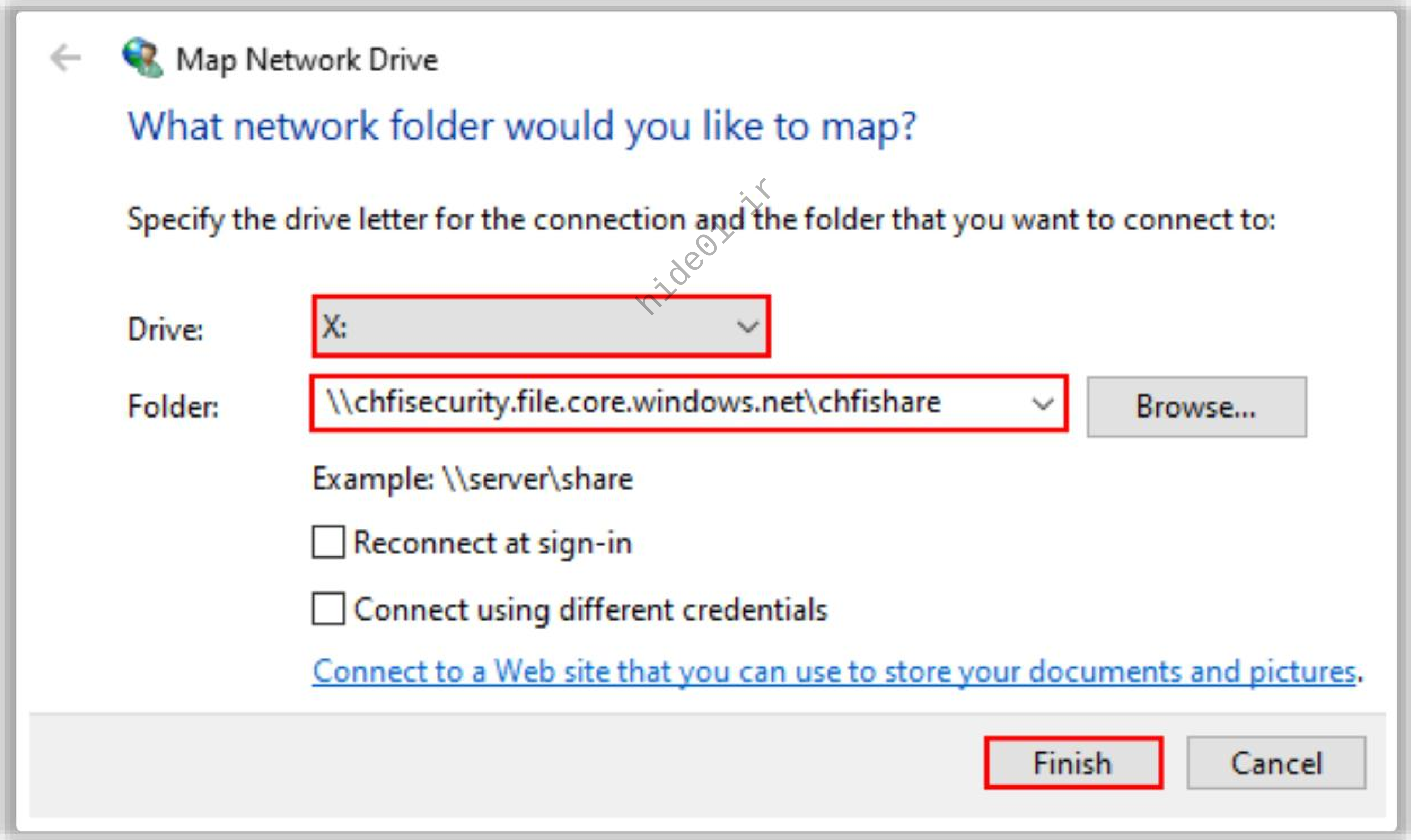


Figure 8.36: Selecting Map Network Drive option

3. Select a drive letter (here, we have selected X: as the drive letter) and provide the UNC path in file share. In this scenario, it is \\chfisecurity.file.core.windows.net\\chfishare

4. If prompted, provide the storage account name as username and storage key as password


Map Network Drive

What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: X:

Folder: \\chfisecurity.file.core.windows.net\chfishare

Example: \\server\share

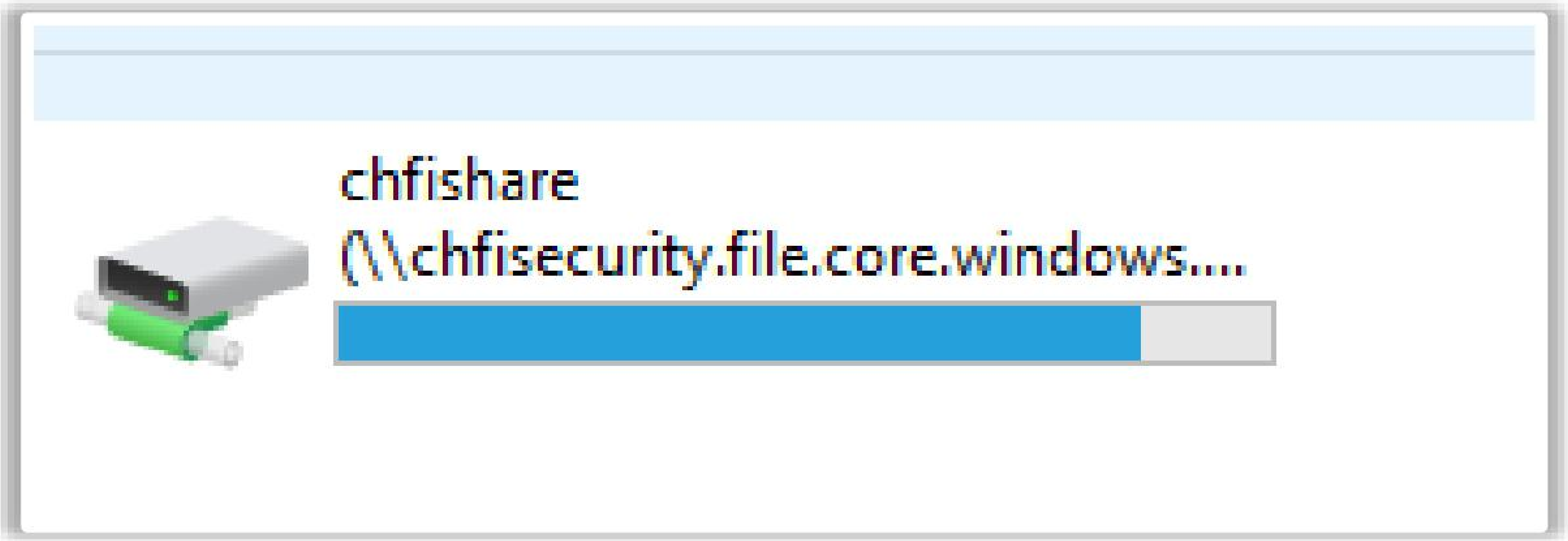
□ Reconnect at sign-in

□ Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures.

Figure 8.37: Selecting a Drive and UNC Path

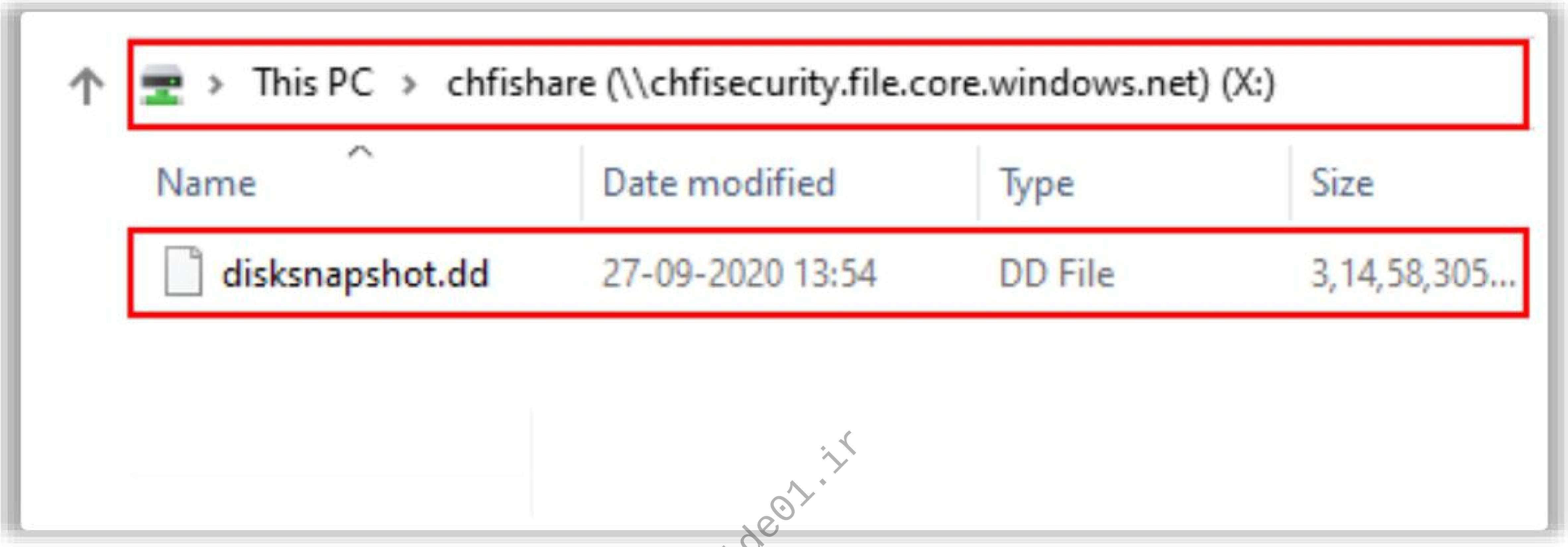
5. The file share chfishare is now mounted under Network Locations



chfishare
(\\chfisecurity.file.core.windows....

Figure 8.38: chfishare is Mounted

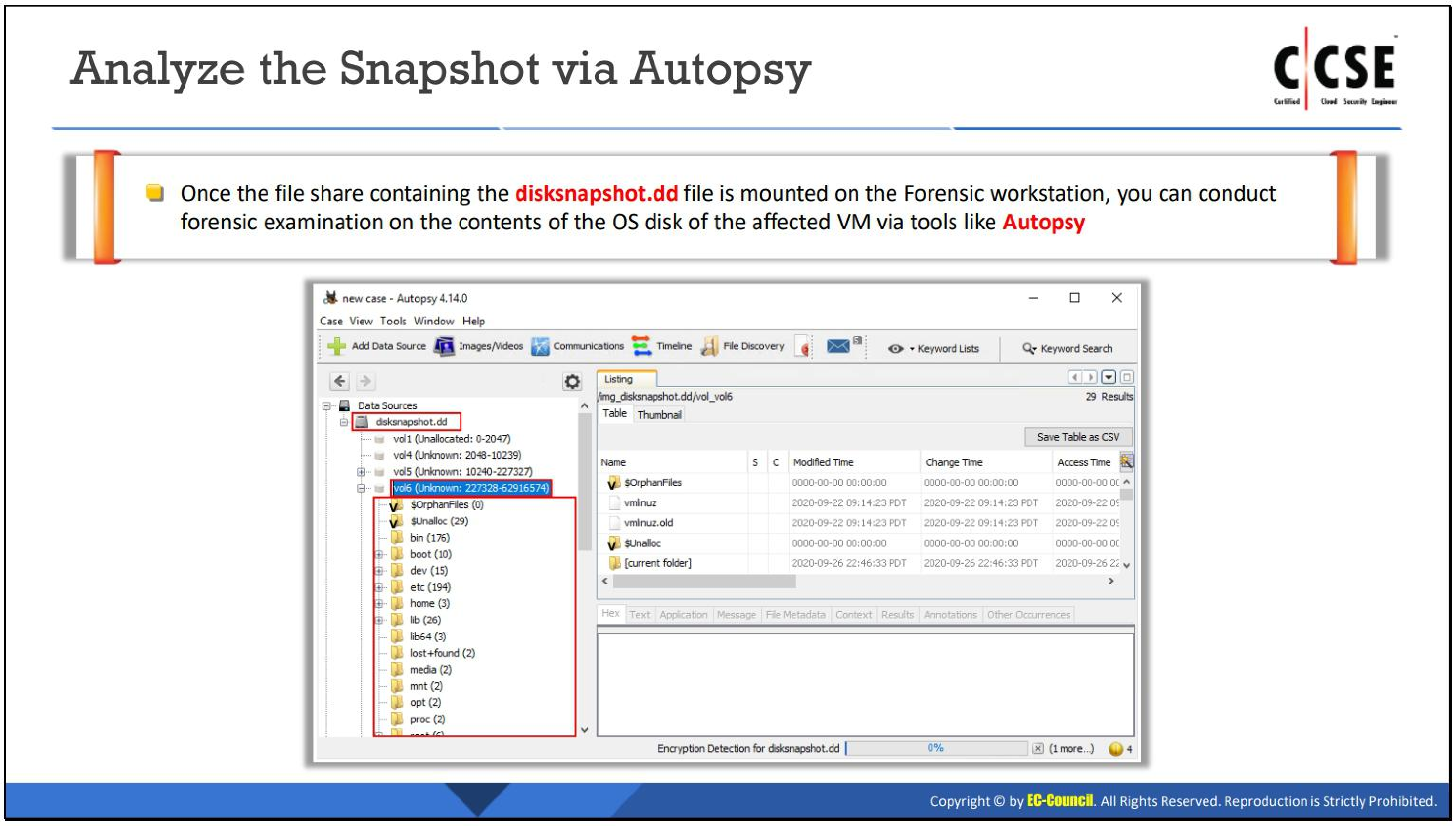
6. Double-click on the mounted share to view the dd file



This PC > chfishare (\\chfisecurity.file.core.windows.net) (X:)

Name	Date modified	Type	Size
disksnapshot.dd	27-09-2020 13:54	DD File	3,14,58,305...

Figure 8.39: Viewing the File in chfishare

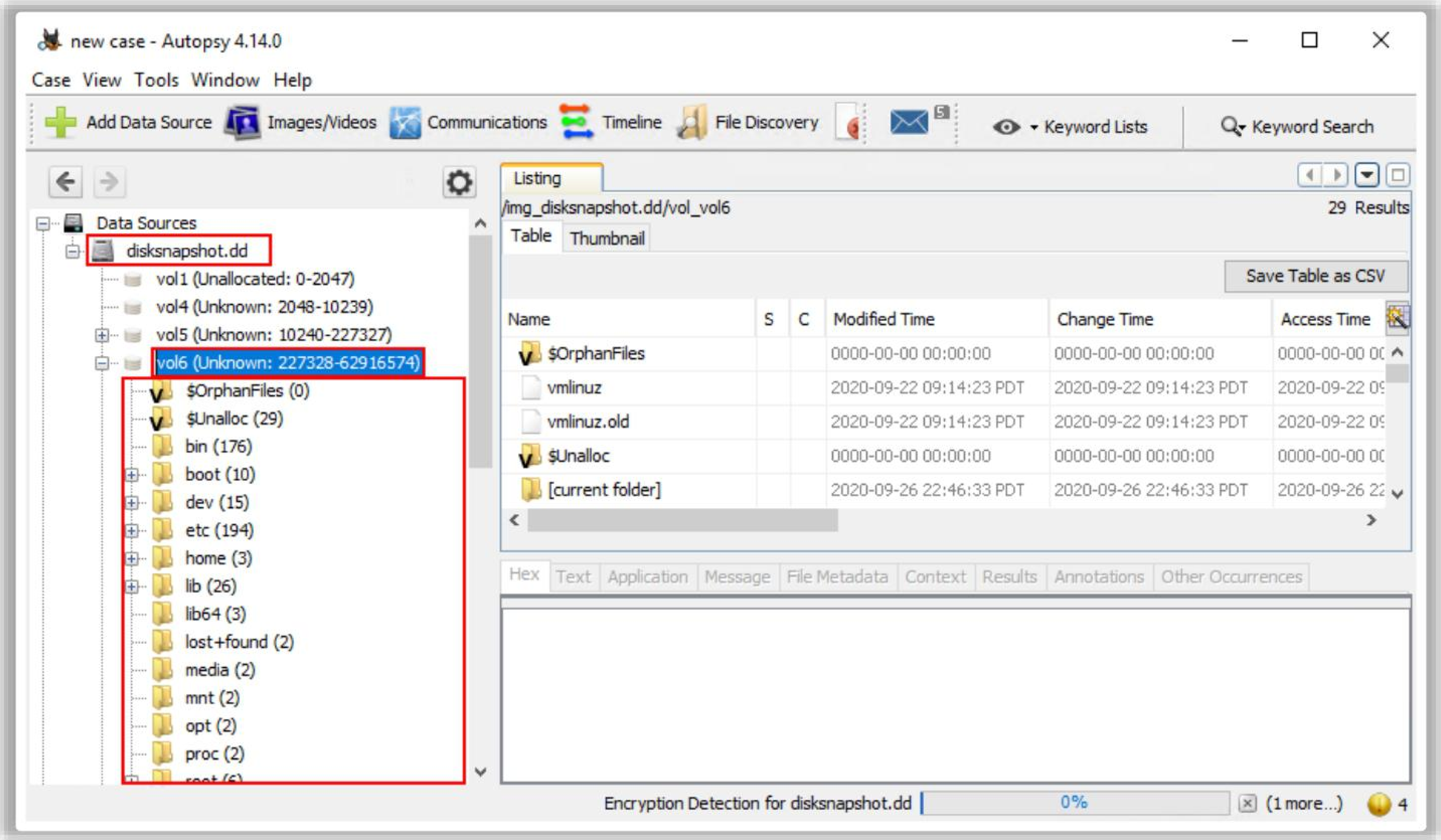


Analyze the Snapshot via Autopsy

- Once the file share containing the **disksnapshot.dd** file is mounted on the Forensic workstation, you can conduct forensic examination on the contents of the OS disk of the affected VM via tools like **Autopsy**

Analyze the Snapshot via Autopsy

Once the file share containing the disksnapshot.dd file is mounted on the Forensic workstation, you can conduct forensic examination on the contents of the OS disk of the affected VM via tools like Autopsy.



Name	S	C	Modified Time	Change Time	Access Time
$OrphanFiles			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00
vmlinuz			2020-09-22 09:14:23 PDT	2020-09-22 09:14:23 PDT	2020-09-22 09
vmlinuz.old			2020-09-22 09:14:23 PDT	2020-09-22 09:14:23 PDT	2020-09-22 09
$Unalloc			0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00
[current folder]			2020-09-26 22:46:33 PDT	2020-09-26 22:46:33 PDT	2020-09-26 22

Figure 8.40: Examining the disksnapshot.dd

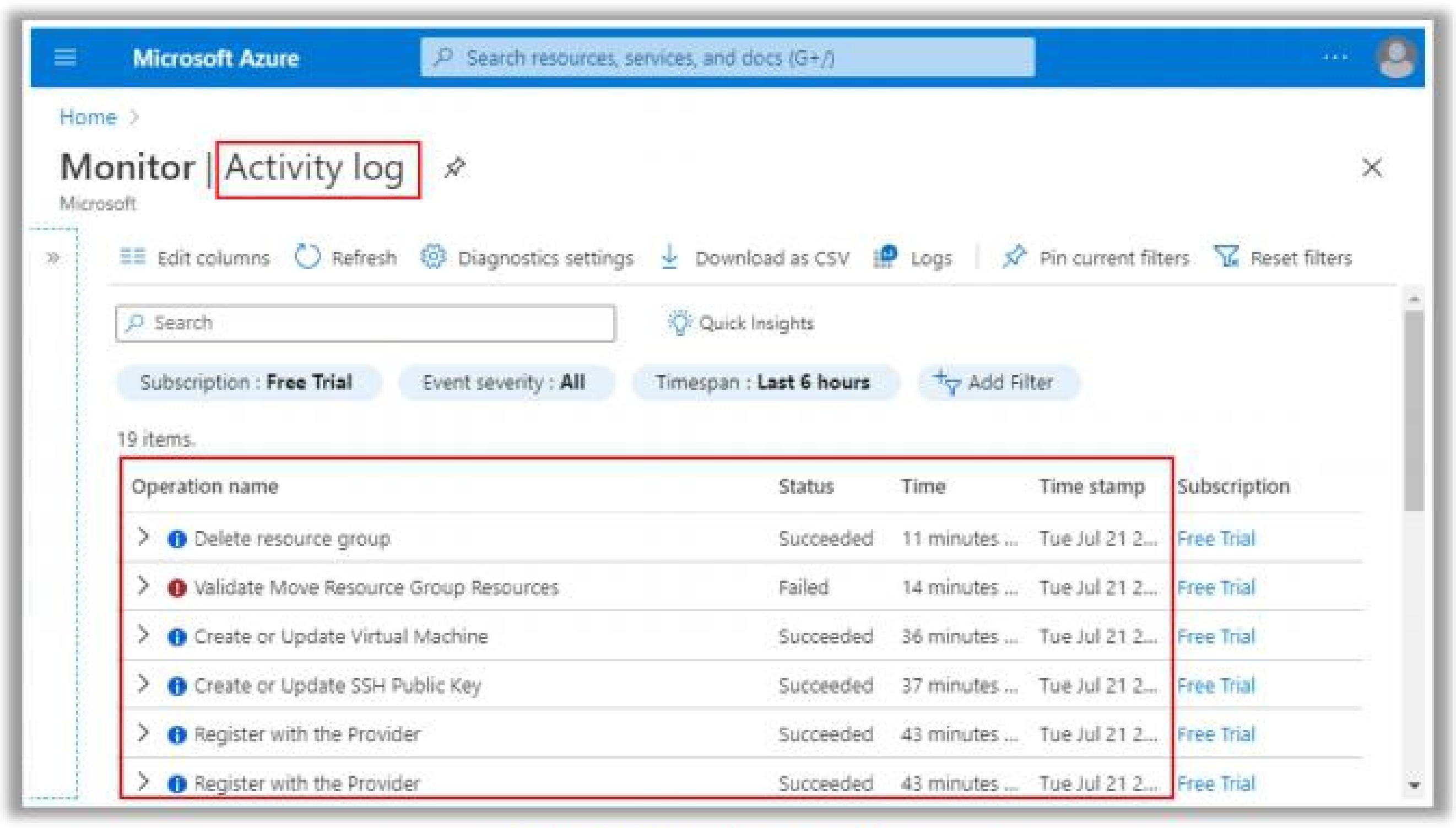
To summarize, we first created a snapshot of the Ubuntu18 VM by using Azure CLI. Next, we created a storage account named chfisecurity under Security-group and created a file share named chfishare in that storage account. Then, we copied the OS disk snapshot with the name disksnapshot.dd in chfishare so that it can be mounted on any forensic workstation. Once the snapshot has been copied, we deleted it from the Production-group for better security. We also created a blob container named chficontainer and stored the OS disk snapshot in it as a page blob with the name osdisk.vhd. Then, we mounted the file named disksnapshot.dd to an on-premises Windows forensic workstation and analyzed it via the Autopsy tool.

Logs in Azure

- There are multiple **security logging** and **auditing features** available in Microsoft Azure
- Different types of logs generated by the Azure resources and services can be **synced** with the **Azure monitor** or any other **SIEM solution** for better monitoring

1. Azure Activity Logs

- These logs record the write operations (**POST, UPDATE, PUT, DELETE**) performed on Azure resources within a particular subscription from outside
- **Log analysis** can reveal the operations that were performed on the resources, their timestamps, status of the operations, and the user who performed these operations
- Audit logs can be **viewed** on the **Monitor** option from the **Azure portal**



Subscription : **Free Trial** Event severity : **All** Timespan : **Last 6 hours**

19 items.

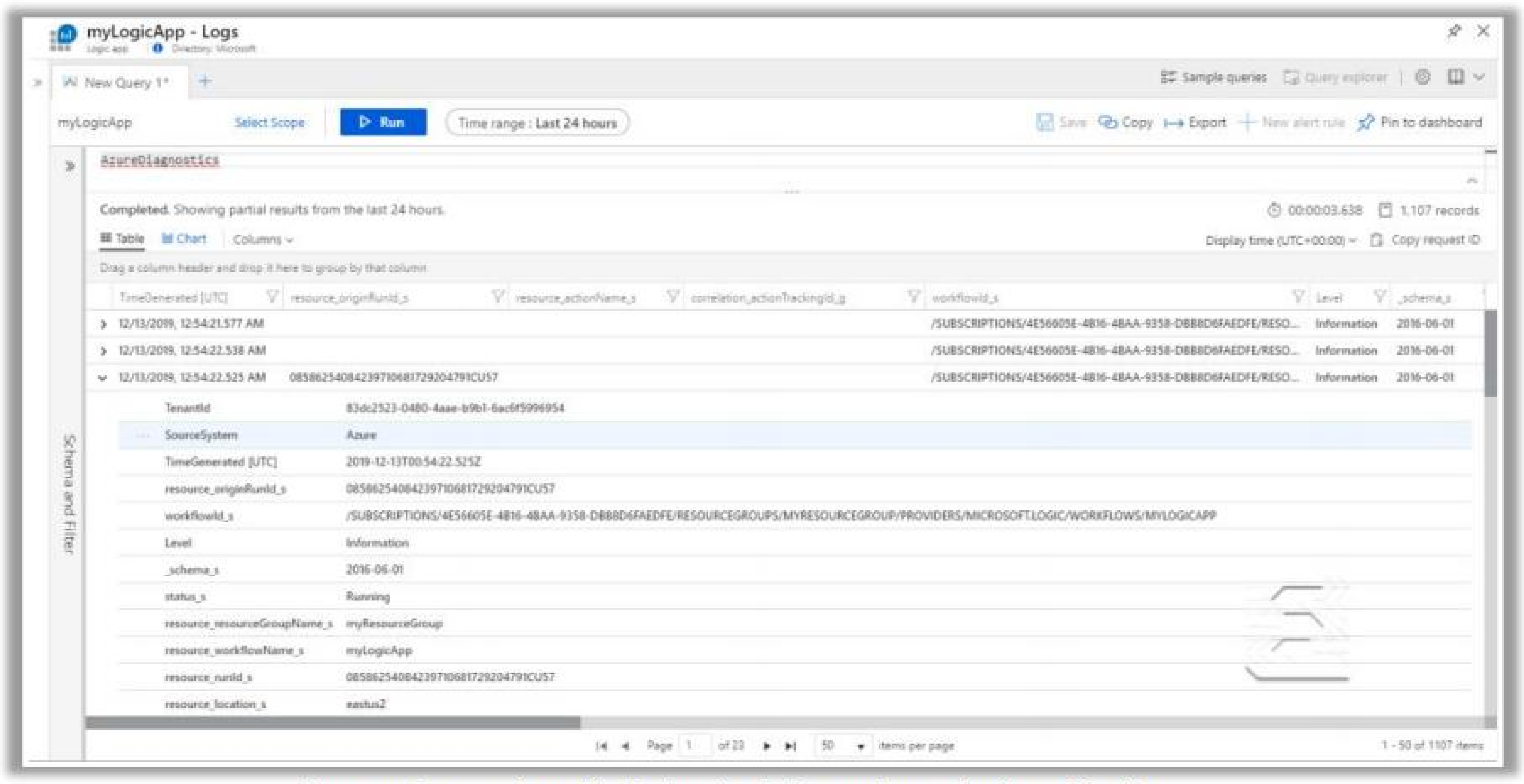
Operation name	Status	Time	Time stamp	Subscription
Delete resource group	Succeeded	11 minutes ...	Tue Jul 21 2...	Free Trial
Validate Move Resource Group Resources	Failed	14 minutes ...	Tue Jul 21 2...	Free Trial
Create or Update Virtual Machine	Succeeded	36 minutes ...	Tue Jul 21 2...	Free Trial
Create or Update SSH Public Key	Succeeded	37 minutes ...	Tue Jul 21 2...	Free Trial
Register with the Provider	Succeeded	43 minutes ...	Tue Jul 21 2...	Free Trial
Register with the Provider	Succeeded	43 minutes ...	Tue Jul 21 2...	Free Trial

Activity logs viewed in Azure portal

Logs in Azure (Cont'd)

2. Azure Resource Logs

- Previously known as diagnostic logs, these logs record the **operations** performed **within a resource**
- These logs might **vary** as per the **Azure resource/service type**
- Administrators must create a **diagnostic setting** for each Azure resource to collect their resource logs



```
AzureDiagnostics
```

Completed. Showing partial results from the last 24 hours. 00:00:03.638 1,107 records

TimeGenerated [UTC]	resource_originRunId_s	resource_actionName_s	correlation_actionTrackingId_g	workflowId_s	Level	_schema_s
12/13/2019, 12:54:21.577 AM				/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESO...	Information	2016-06-01
12/13/2019, 12:54:22.538 AM				/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESO...	Information	2016-06-01
12/13/2019, 12:54:22.525 AM	08586254084239710681729204791CU57			/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESO...	Information	2016-06-01

TenantId	83dc2523-0480-4aae-b9b1-6ac6f5996954
SourceSystem	Azure
TimeGenerated [UTC]	2019-12-13T00:54:22.525Z
resource_originRunId_s	08586254084239710681729204791CU57
workflowId_s	/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESOURCEGROUPS/MYRESOURCEGROUP/PROVIDERS/MICROSOFT.LOGIC/WORKFLOWS/MYLOGICAPP
Level	Information
_schema_s	2016-06-01
status_s	Running
resource_resourceGroupName_s	myResourceGroup
resource_workflowName_s	myLogicApp
resource_runId_s	08586254084239710681729204791CU57
resource_location_s	eastus2

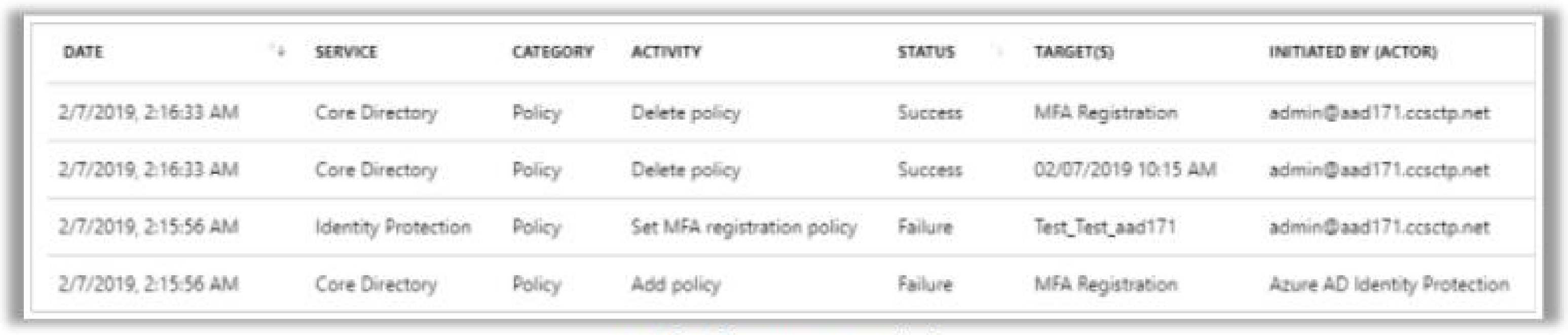
Resource logs as viewed in the Log Analytics workspace in Azure Monitor

https://docs.microsoft.com

Logs in Azure (Cont'd)

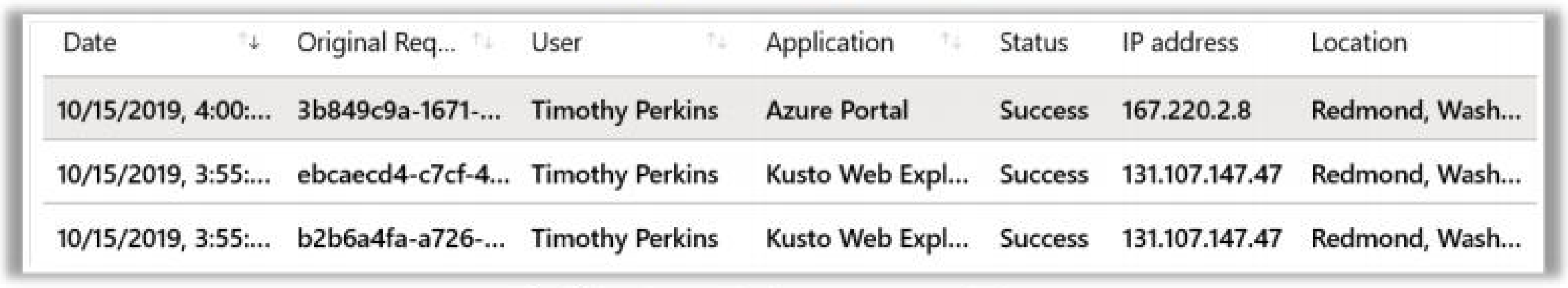
3. Azure Active Directory Reports

- **Security reports:** Two types of security reports are generated by Azure AD:
  - **Users flagged for risk:** It provides a comprehensive idea on user accounts that might have been **compromised**
  - **Risky Sign-ins:** It records **sign-in attempts** from unauthorized users (if any)
- **Activity reports:** Two types of activity reports are generated by Azure AD:
  - **Audit logs:** It records all **system activities** and tasks being performed within an organization. Audit activity logs might take an hour to get recorded.
  - **Sign-ins:** It records the **sign-in patterns** and **status** by users



DATE	SERVICE	CATEGORY	ACTIVITY	STATUS	TARGET(S)	INITIATED BY (ACTOR)
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	MFA Registration	admin@aad171.ccsctp.net
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	02/07/2019 10:15 AM	admin@aad171.ccsctp.net
2/7/2019, 2:15:56 AM	Identity Protection	Policy	Set MFA registration policy	Failure	Test_Test_aad171	admin@aad171.ccsctp.net
2/7/2019, 2:15:56 AM	Core Directory	Policy	Add policy	Failure	MFA Registration	Azure AD Identity Protection

Example of Azure AD audit logs



Date	Original Req...	User	Application	Status	IP address	Location
10/15/2019, 4:00:...	3b849c9a-1671-...	Timothy Perkins	Azure Portal	Success	167.220.2.8	Redmond, Wash...
10/15/2019, 3:55:...	ebcaecd4-c7cf-4...	Timothy Perkins	Kusto Web Expl...	Success	131.107.147.47	Redmond, Wash...
10/15/2019, 3:55:...	b2b6a4fa-a726-...	Timothy Perkins	Kusto Web Expl...	Success	131.107.147.47	Redmond, Wash...

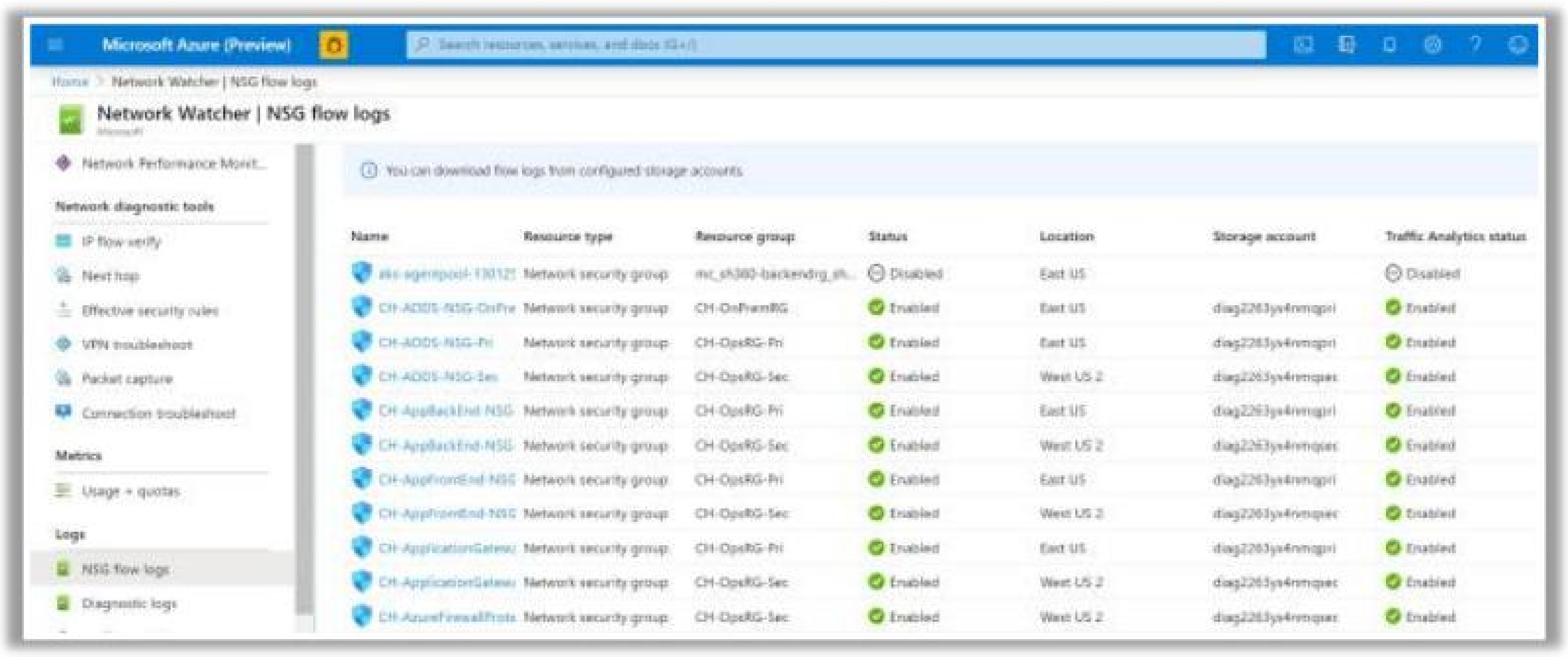
Example of sign-in entries in Azure AD activity report

https://docs.microsoft.com

Logs in Azure (Cont'd)

4. Network Security Group Flow Logs

- These logs record information related to the **inbound** and **outbound IP traffic** flowing through Azure resources within a Network Security Group (NSG)
- These logs are **recorded** in the **JSON format** on a per rule basis
- NSG flow logs can be **enabled** via **Azure Network Watcher** on the Azure portal, stored and **downloaded** from a configured **Azure storage account**, or exported to any **SIEM** or **IDS tool** for better visualization



You can download flow logs from configured storage accounts.

Name	Resource type	Resource group	Status	Location	Storage account	Traffic Analytics status
aks-agentpool-130121	Network security group	mc_sh360-backendrg_sh...	Disabled	East US		Disabled
CH-ADDS-NSG-OnPre	Network security group	CH-OnPremRG	Enabled	East US	diag2263yx4nmqpri	Enabled
CH-ADDS-NSG-Pri	Network security group	CH-OpsRG-Pri	Enabled	East US	diag2263yx4nmqpri	Enabled
CH-ADDS-NSG-Sec	Network security group	CH-OpsRG-Sec	Enabled	West US 2	diag2263yx4nmqsec	Enabled
CH-AppBackEnd-NSG	Network security group	CH-OpsRG-Pri	Enabled	East US	diag2263yx4nmqpri	Enabled
CH-AppBackEnd-NSG	Network security group	CH-OpsRG-Sec	Enabled	West US 2	diag2263yx4nmqsec	Enabled
CH-AppFrontEnd-NSG	Network security group	CH-OpsRG-Pri	Enabled	East US	diag2263yx4nmqpri	Enabled
CH-AppFrontEnd-NSG	Network security group	CH-OpsRG-Sec	Enabled	West US 2	diag2263yx4nmqsec	Enabled
CH-ApplicationGatew	Network security group	CH-OpsRG-Pri	Enabled	East US	diag2263yx4nmqpri	Enabled
CH-ApplicationGatew	Network security group	CH-OpsRG-Sec	Enabled	West US 2	diag2263yx4nmqsec	Enabled
CH-AzureFirewallProte	Network security group	CH-OpsRG-Sec	Enabled	West US 2	diag2263yx4nmqsec	Enabled

NSG flow log entries

https://docs.microsoft.com

Logs in Azure (Cont'd)

5. VM Log Data

- Data from **event logs** in **Windows VM** and **Syslog** in **Linux VM** can be collected and analyzed in a specific **Log Analytics Workspace** via Azure Monitor
- Enabling the **Log Analytics VM extension** option makes this log collection process easier and configures the log agent to send data to the specified Log Analytics Workspace automatically

6. Azure Storage Analytics Logs

- These log entries record information about the **authenticated** and **anonymous requests** made to specific **storage services**, such as Azure blob, queue, and table
- When enabled for a storage account, these logs are automatically placed in block blobs in a container called **$logs**

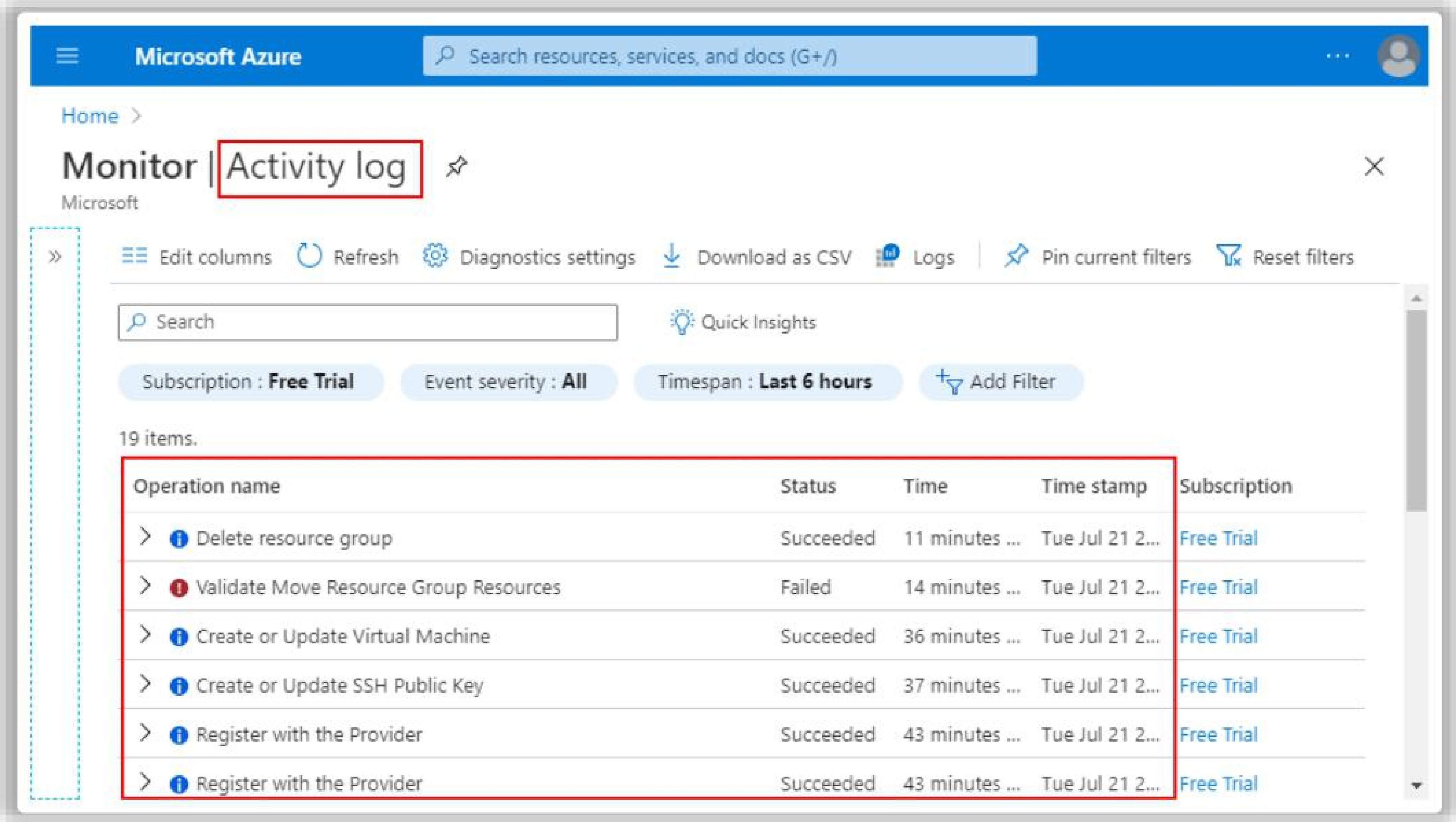
Logs in Azure

Source: *https://docs.microsoft.com*

Multiple security logging and auditing features are available in Microsoft Azure that can help in locating relevant evidence during the investigation of a security breach. Administrators can sync different types of logs generated by the Azure resources with services such as the Azure Monitor or any other SIEM solution for better monitoring.

Azure Activity Logs

Azure activity logs are a type of Azure platform logs that record information on the Azure subscription layer. These logs can be viewed via the Azure portal.



Operation name	Status	Time	Time stamp	Subscription
Delete resource group	Succeeded	11 minutes ...	Tue Jul 21 2...	Free Trial
Validate Move Resource Group Resources	Failed	14 minutes ...	Tue Jul 21 2...	Free Trial
Create or Update Virtual Machine	Succeeded	36 minutes ...	Tue Jul 21 2...	Free Trial
Create or Update SSH Public Key	Succeeded	37 minutes ...	Tue Jul 21 2...	Free Trial
Register with the Provider	Succeeded	43 minutes ...	Tue Jul 21 2...	Free Trial
Register with the Provider	Succeeded	43 minutes ...	Tue Jul 21 2...	Free Trial

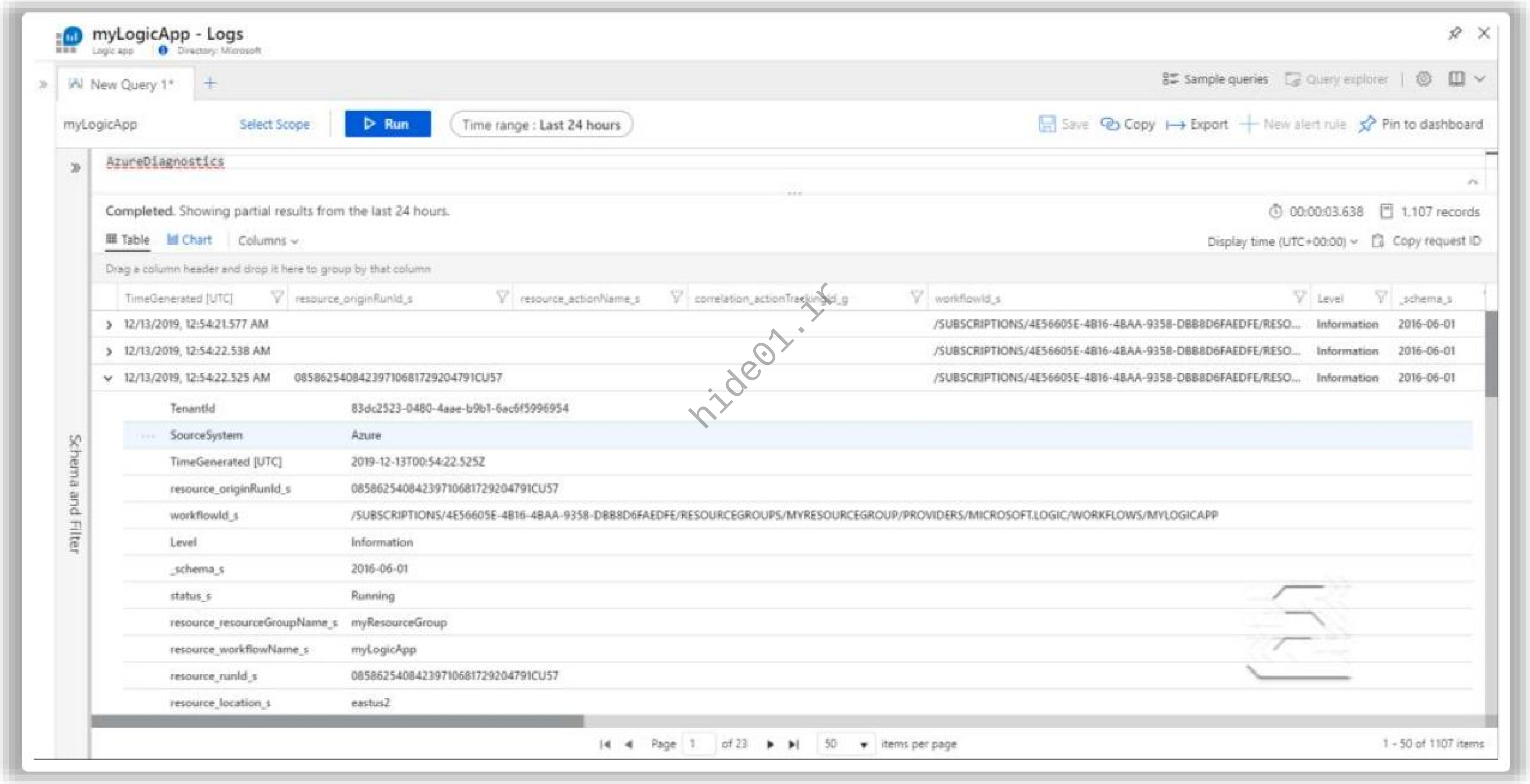
Figure 8.41: Activity Logs as Viewed from Azure Portal

These logs record the write operations (POST, UPDATE, PUT, DELETE) performed on Azure resources within a specific subscription from outside (or the management plane). Activity logs are generated for each Azure resource. By analyzing these logs, information related to the operations performed on a specific resource, timestamp, status of the operation, and the user responsible for it can be obtained.

Azure Resource Logs

Azure resource logs are another type of platform log that record information on the operations performed within an Azure resource (the data plane). The generation of resource logs might vary with the type of Azure resource/service.

Azure resources generate these logs automatically; however, they are not collected by default. To collect the resource logs, administrators must create a diagnostic setting for the resource they want to collect the logs from and send those logs to a destination such as Azure Log Analytics workspace. The resource logs can be also stored in Azure storage or forwarded to Azure Event Hubs.



```
AzureDiagnostics
```

Completed. Showing partial results from the last 24 hours. 00:00:03.638 1,107 records

TimeGenerated [UTC]	resource_originRunId_s	resource_actionName_s	correlation_actionTrackingId_g	workflowId_s	Level	_schema_s
12/13/2019, 12:54:21.577 AM				/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESO...	Information	2016-06-01
12/13/2019, 12:54:22.538 AM				/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESO...	Information	2016-06-01
12/13/2019, 12:54:22.525 AM	08586254084239710681729204791CU57			/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESO...	Information	2016-06-01

TenantId	83dc2523-0480-4aae-b9b1-6ac6f5996954
SourceSystem	Azure
TimeGenerated [UTC]	2019-12-13T00:54:22.525Z
resource_originRunId_s	08586254084239710681729204791CU57
workflowId_s	/SUBSCRIPTIONS/4E56605E-4B16-4BAA-9358-DBB8D6FAEDFE/RESOURCEGROUPS/MYRESOURCEGROUP/PROVIDERS/MICROSOFT.LOGIC/WORKFLOWS/MYLOGICAPP
Level	Information
_schema_s	2016-06-01
status_s	Running
resource_resourceGroupName_s	myResourceGroup
resource_workflowName_s	myLogicApp
resource_runId_s	08586254084239710681729204791CU57
resource_location_s	eastus2

1 - 50 of 1107 items

Figure 8.42: Resource Logs as Viewed in the Log Analytics Workspace in Azure Monitor

Azure Active Directory Reports

Azure Active Directory (AD) reports are a type of platform logs in Azure, which identify the risks within an Azure production environment, help administrators in understanding how Azure services and applications are executed within the cloud, and troubleshoot the potential issues hampering the organizational activities. Azure AD generates security and activity reports that can provide relevant clues to forensic investigators to determine the origin of a security incident. Two types of AD-generated reports are discussed below:

1. **Security Reports**

   There are two types of security reports are generated by Azure AD:

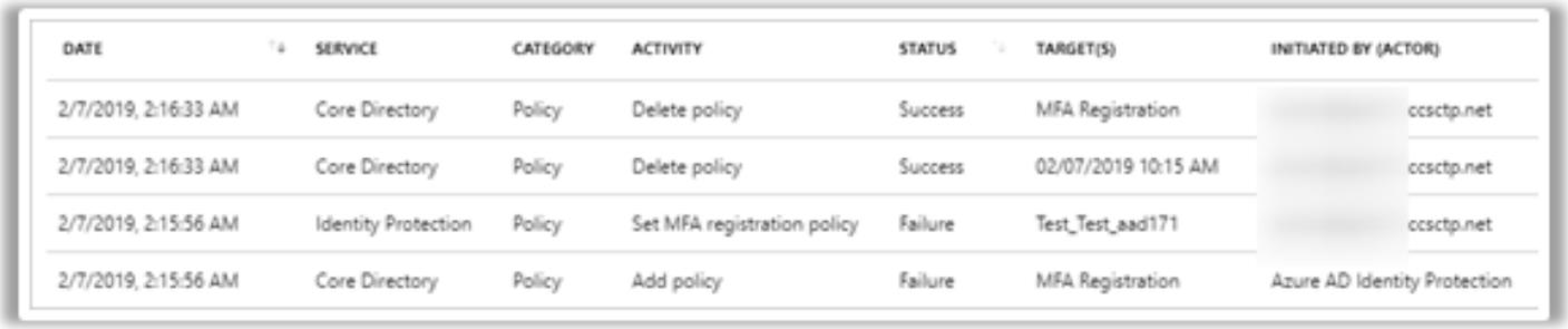
   - **Users Flagged for Risk:** It provides a comprehensive idea on user accounts that might have been compromised

- **Risky Sign-ins:** It records sign-in attempts from unauthorized users (if any)

2. **Activity Reports**

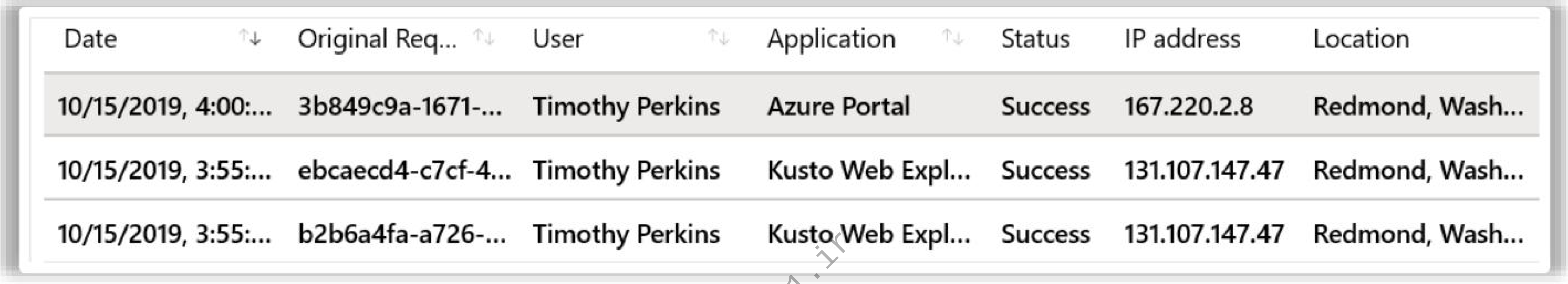
   There are two types of activity reports are generated by Azure AD:

   - **Audit Logs:** It records all system activities and tasks being performed within an organization. Audit activity logs might take an hour to get recorded.
   - **Sign-ins:** It records the sign-in patterns and status by users



DATE ↑↓	SERVICE	CATEGORY	ACTIVITY	STATUS ↑↓	TARGET(S)	INITIATED BY (ACTOR)
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	MFA Registration	[redacted]ccsctp.net
2/7/2019, 2:16:33 AM	Core Directory	Policy	Delete policy	Success	02/07/2019 10:15 AM	[redacted]ccsctp.net
2/7/2019, 2:15:56 AM	Identity Protection	Policy	Set MFA registration policy	Failure	Test_Test_aad171	[redacted]ccsctp.net
2/7/2019, 2:15:56 AM	Core Directory	Policy	Add policy	Failure	MFA Registration	Azure AD Identity Protection

Figure 8.43: Example of Azure AD Audit Logs

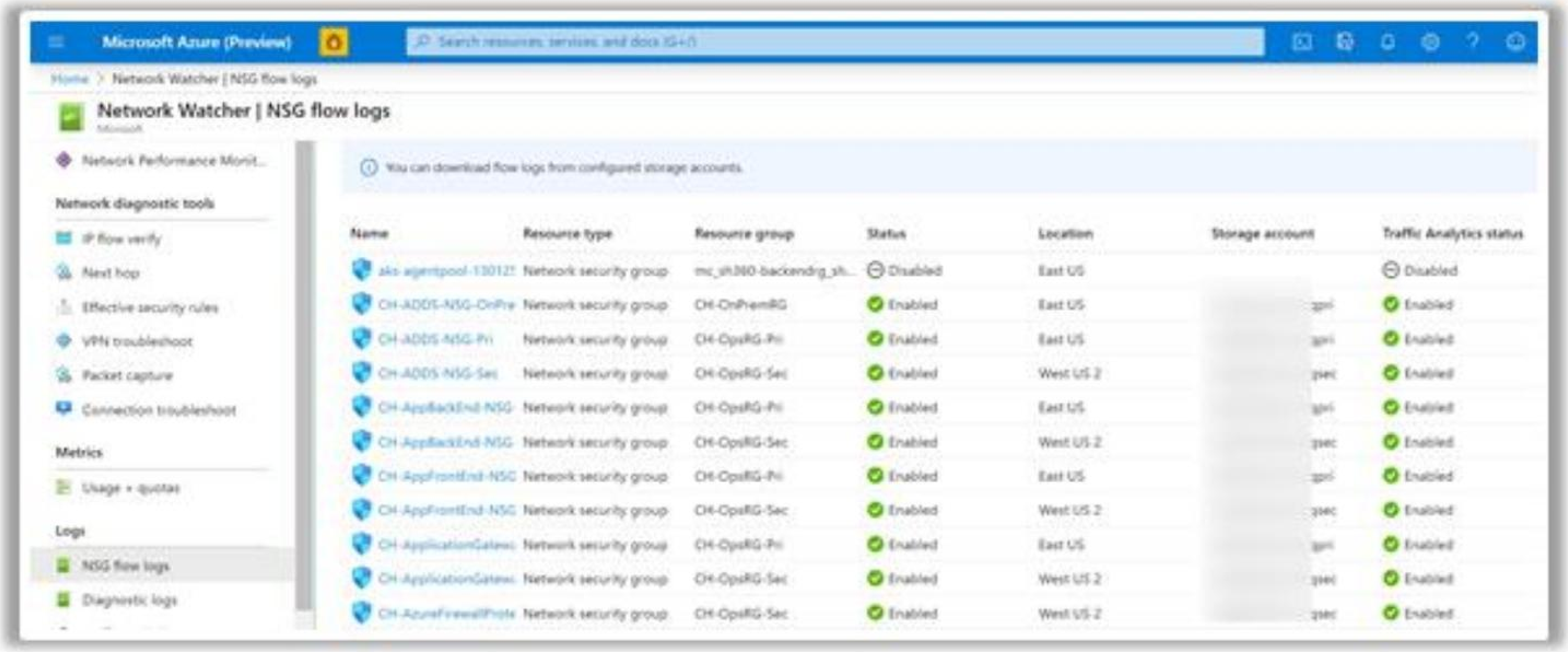


Date ↑↓	Original Req... ↑↓	User ↑↓	Application ↑↓	Status	IP address	Location
10/15/2019, 4:00:...	3b849c9a-1671-...	Timothy Perkins	Azure Portal	Success	167.220.2.8	Redmond, Wash...
10/15/2019, 3:55:...	ebcaecd4-c7cf-4...	Timothy Perkins	Kusto Web Expl...	Success	131.107.147.47	Redmond, Wash...
10/15/2019, 3:55:...	b2b6a4fa-a726-...	Timothy Perkins	Kusto Web Expl...	Success	131.107.147.47	Redmond, Wash...

Figure 8.44: Example of Sign-In Entries in Azure AD Activity Report

Network Security Group Flow Logs

Flow logs are added as a feature in Azure Network Watcher, which enables administrators/investigators to monitor the inbound and outbound network traffic flowing through Azure resources within a network security group (NSG).



You can download flow logs from configured storage accounts.

Name	Resource type	Resource group	Status	Location	Storage account	Traffic Analytics status
aks-agentpool-130125	Network security group	mc_sh360-backendrg_sh...	⊖ Disabled	East US		⊖ Disabled
CH-ADDS-NSG-OnPre	Network security group	CH-OnPremRG	✓ Enabled	East US	[redacted]pri	✓ Enabled
CH-ADDS-NSG-Pri	Network security group	CH-OpsRG-Pri	✓ Enabled	East US	[redacted]pri	✓ Enabled
CH-ADDS-NSG-Sec	Network security group	CH-OpsRG-Sec	✓ Enabled	West US 2	[redacted]sec	✓ Enabled
CH-AppBackEnd-NSG	Network security group	CH-OpsRG-Pri	✓ Enabled	East US	[redacted]pri	✓ Enabled
CH-AppBackEnd-NSG	Network security group	CH-OpsRG-Sec	✓ Enabled	West US 2	[redacted]sec	✓ Enabled
CH-AppFrontEnd-NSG	Network security group	CH-OpsRG-Pri	✓ Enabled	East US	[redacted]pri	✓ Enabled
CH-AppFrontEnd-NSG	Network security group	CH-OpsRG-Sec	✓ Enabled	West US 2	[redacted]sec	✓ Enabled
CH-ApplicationGatewa	Network security group	CH-OpsRG-Pri	✓ Enabled	East US	[redacted]pri	✓ Enabled
CH-ApplicationGatewa	Network security group	CH-OpsRG-Sec	✓ Enabled	West US 2	[redacted]sec	✓ Enabled
CH-AzureFirewallProte	Network security group	CH-OpsRG-Sec	✓ Enabled	West US 2	[redacted]sec	✓ Enabled

Figure 8.45: Example of NSG Flow Log Entries in Azure

This helps in identifying any suspicious traffic and comprehend the behavior of Azure resources and applications on the network. NSG flow logs are recorded in the JSON format and can be retained up to one year.

The flow logs can be enabled via Azure Network Watcher on the Azure portal. They can be exported to and downloaded from a storage account configured for this purpose or synchronized with any SIEM or IDS tool for better monitoring.

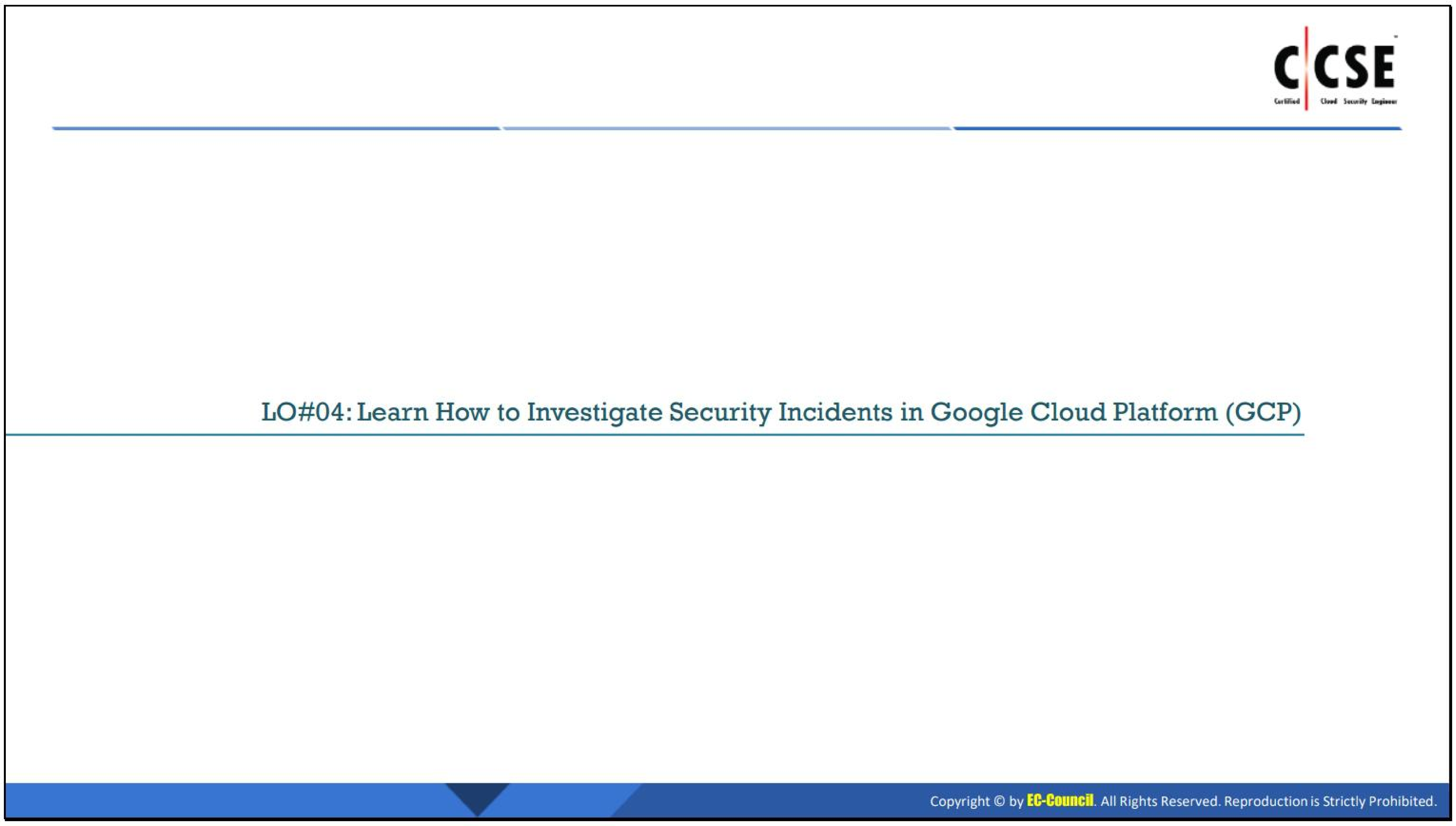
VM Log Data

Administrators can collect the system and other logging data from the VMs via the Azure Monitor. They can collect specific event log or performance-related data from Windows VM or Syslog from a Linux VM by creating a Log Analytics workspace and enabling the Log Analytics VM extension. This makes this log collection process easier and configures the log agent to send data to the specified Log Analytics Workspace automatically.

Azure Storage Analytics Logs

These logs record information related to all successful and failed requests made to Azure storage services such as Azure blobs, Azure queue, and Azure table. This logging feature can be enabled via the Azure portal.

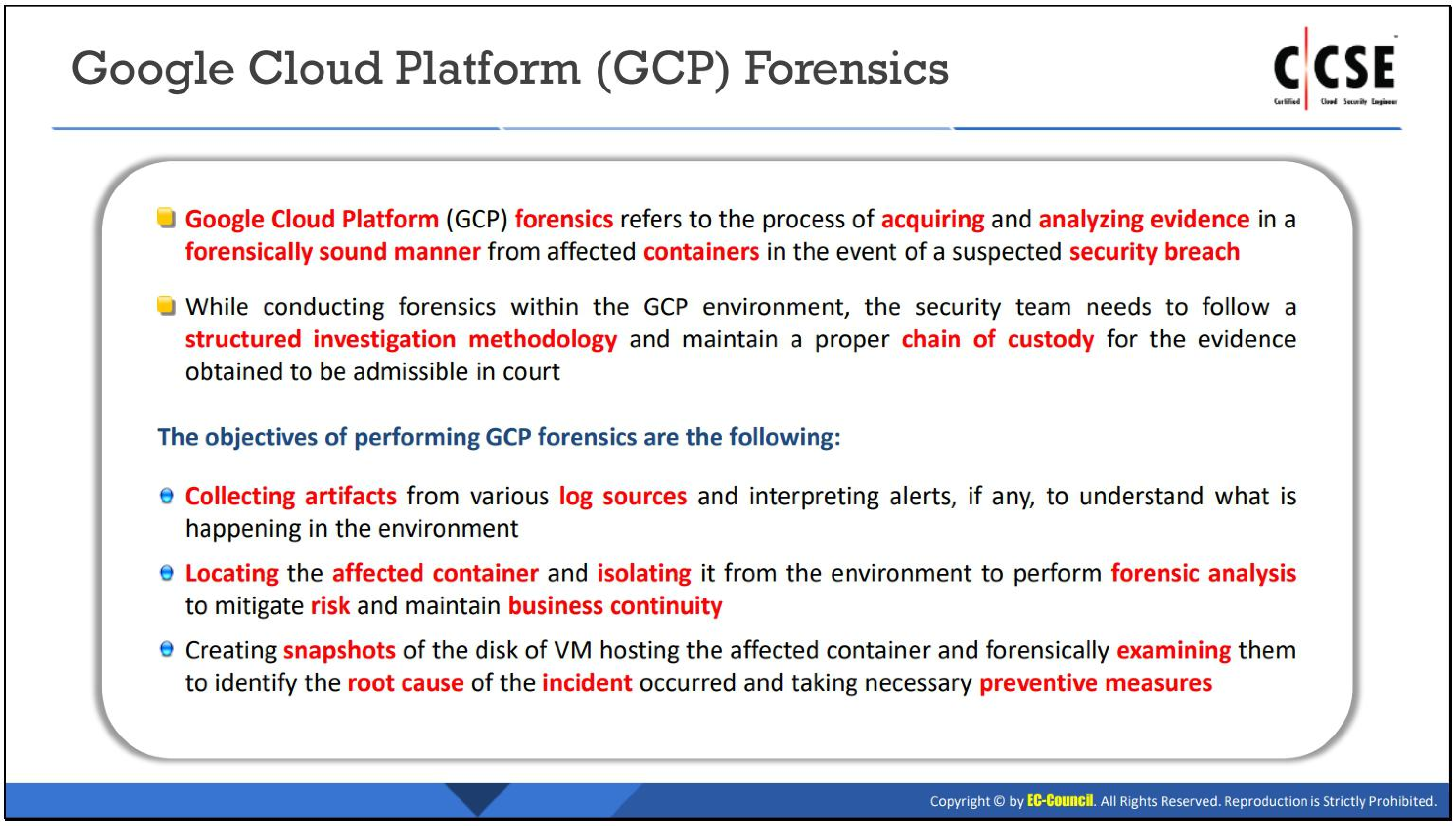
These logs record authenticated as well as anonymous requests. When enabled for a storage account, these logs are automatically placed in block blobs in a container called $logs. This container cannot be removed when the storage analytics are enabled.



LO#04: Learn How to Investigate Security Incidents in Google Cloud Platform (GCP)

LO#04: Learn How to Investigate Security Incidents in Google Cloud Platform (GCP)

This section discusses the forensic acquisition methodology of an GCP VM and discusses an assumed scenario to divide the whole process into multiple steps.



Google Cloud Platform (GCP) Forensics

- **Google Cloud Platform** (GCP) **forensics** refers to the process of **acquiring** and **analyzing evidence** in a **forensically sound manner** from affected **containers** in the event of a suspected **security breach**
- While conducting forensics within the GCP environment, the security team needs to follow a **structured investigation methodology** and maintain a proper **chain of custody** for the evidence obtained to be admissible in court

The objectives of performing GCP forensics are the following:

- **Collecting artifacts** from various **log sources** and interpreting alerts, if any, to understand what is happening in the environment
- **Locating** the **affected container** and **isolating** it from the environment to perform **forensic analysis** to mitigate **risk** and maintain **business continuity**
- Creating **snapshots** of the disk of VM hosting the affected container and forensically **examining** them to identify the **root cause** of the **incident** occurred and taking necessary **preventive measures**

Google Cloud Platform (GCP) Forensics

Google Cloud Platform (GCP) offers a host of cloud computing services such as computing, networking, machine learning, Big Data etc., that the customers can use with a monthly subscription. GCP comes with Google Kubernetes Engine (GKE) inbuilt within its environment that allows its customers to run containerized applications and multiple workloads at once. Customers can create their own managed GCP projects and manage all their Google Cloud resources by using Google Cloud Console or gcloud, the primary command-line tool of GCP that supports Google Cloud SDK.

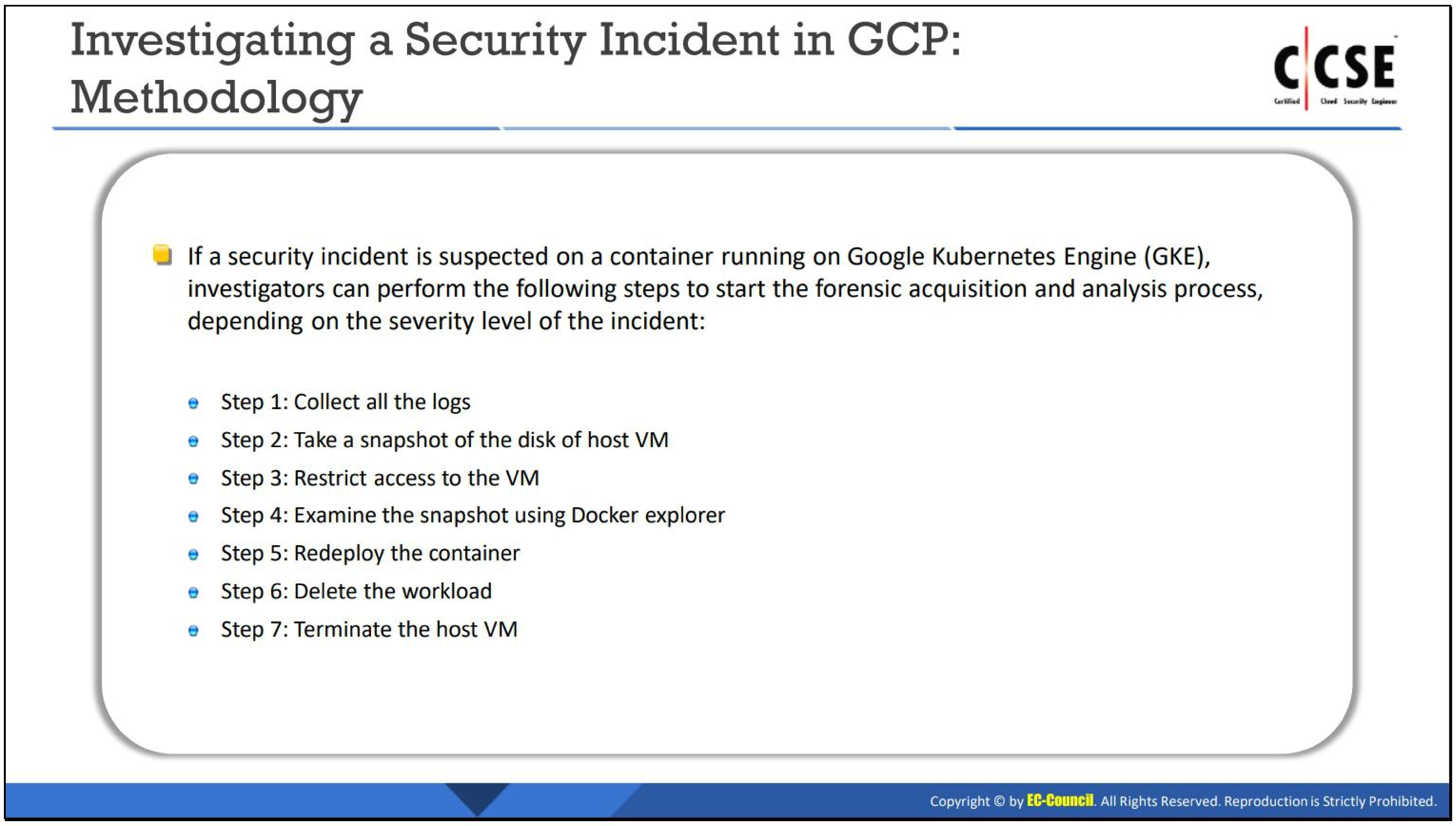
Organizations running business-critical applications and VMs in GKE environment need to draft an organized Incident Response and Management (IRM) plan that can enable them to promptly respond to an incident, maintain business continuity, and prevent any future attack. IRM planning within the GCP environment enables organizations to understand the nature of the incident, actions to be taken to stop the attack, and when to call the forensic experts.

GCP forensics should be an integral part of the IRM planning of an organization which refers to the process of retrieving and acquiring evidence from affected containers in a forensically sound manner and examine them to find the root cause of the incident. While conducting forensics within a GKE environment, it is also important for the forensics team to draft a chain of custody document for the evidence to be admissible in the court of law.

The Objectives of GCP Forensics Include the Following:

- Collecting log data from various sources and analyzing them to understand what is happening within the environment and determine its severity.
- Identifying the affected node and container and isolating the host VM from the other workloads so that business continuity can be maintained.

- Taking a snapshot of VM hosting the container and examining the disk image with the right forensic tools to identify the attack and its impact.
- Drafting a detailed report on the findings of the forensic investigation to help the concerned organization take preventive measures and implement necessary security controls.



Investigating a Security Incident in GCP: Methodology

- If a security incident is suspected on a container running on Google Kubernetes Engine (GKE), investigators can perform the following steps to start the forensic acquisition and analysis process, depending on the severity level of the incident:
  - Step 1: Collect all the logs
  - Step 2: Take a snapshot of the disk of host VM
  - Step 3: Restrict access to the VM
  - Step 4: Examine the snapshot using Docker explorer
  - Step 5: Redeploy the container
  - Step 6: Delete the workload
  - Step 7: Terminate the host VM

Investigating a Security Incident in GCP: Methodology

If a container running on a Google Kubernetes Engine (GKE) is suspected to be compromised, the security team should not immediately terminate the affected nodes/pods because such actions also limit the possibility of probing into the incident and retrieving evidentiary data. The concerned security team should rather follow a structured, step-by-step approach to initiate the forensic acquisition and analysis process as given below:

1. **Collect All the Logs:** Google Cloud generates multiple logs within a containerized environment such as Cloud Audit logs, GKE audit logs, etc. which the cloud administrators should start collecting when they create a GCP project with their own VMs and applications. The collection of these logs beforehand helps organizations ensure their availability when a security incident occurs.

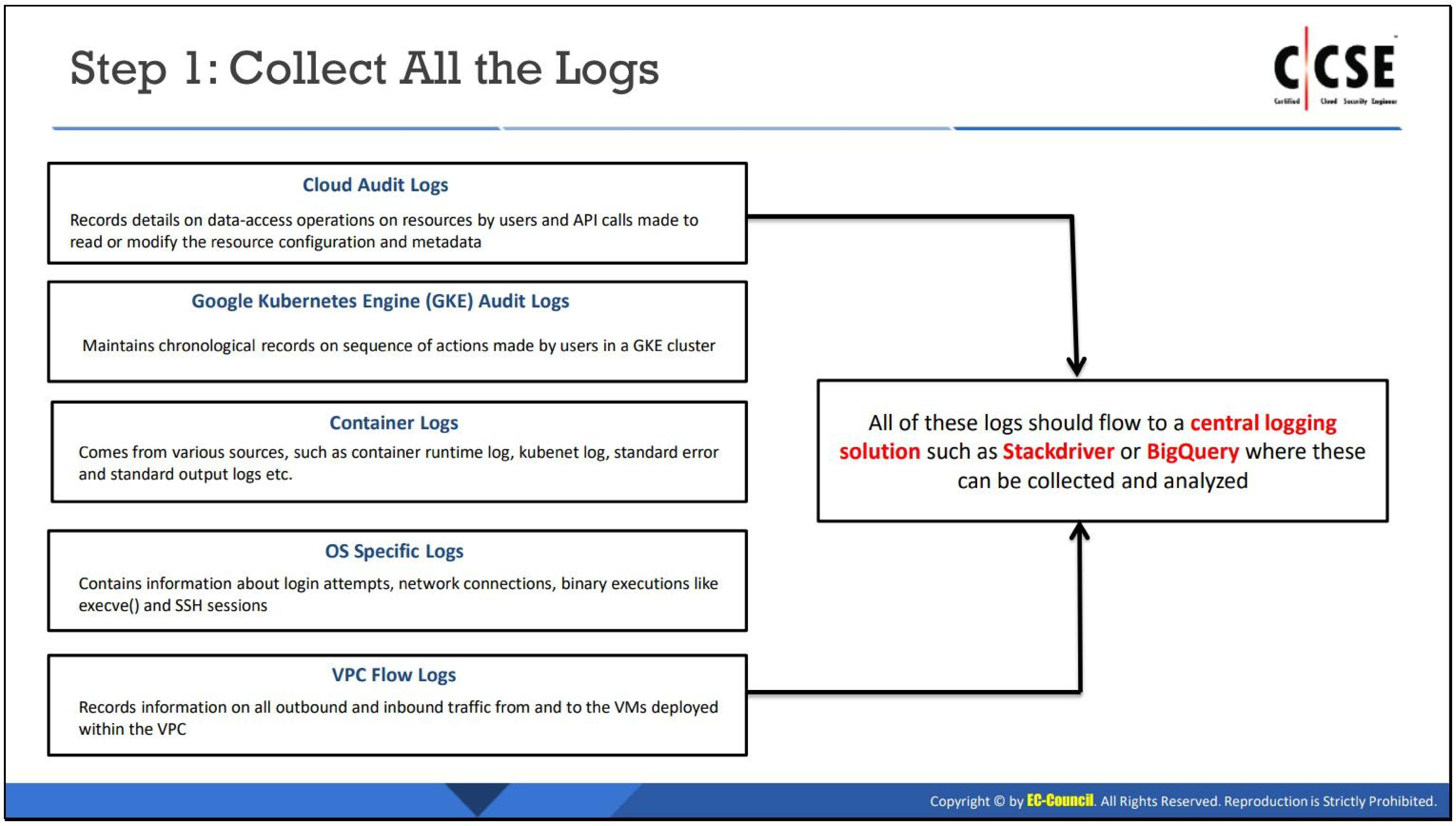
2. **Take a Snapshot of the Disk of the Host VM:** Taking snapshot of the disk of the VM hosting the affected container in GCP similar to acquiring an image of the hard drive of any affected computer in traditional forensics. This allows investigators to perform forensics on the disk snapshot using the right forensic tools.

3. **Restrict Access to the VM:** Limiting access to the VM hosting the affected container aids in isolating it from other workloads, which can play a key role in preventing any further spread of the attack and mitigating risk.

4. **Examine the Snapshot Using Docker Explorer:** Investigators can use open-source tools like Docker explorer to mount the disk snapshot and analyze it further.

5. **Redeploy the Container:** The process of redeploying a container involves the removal of the hosting Pod and creating a fresh copy of the container.

6. **Delete the Workload:** Deletion of workload should only be performed when there is an ongoing attack that needs to be stopped immediately.



Step 1: Collect All the Logs

Cloud Audit Logs

Records details on data-access operations on resources by users and API calls made to read or modify the resource configuration and metadata

Google Kubernetes Engine (GKE) Audit Logs

Maintains chronological records on sequence of actions made by users in a GKE cluster

Container Logs

Comes from various sources, such as container runtime log, kubenet log, standard error and standard output logs etc.

OS Specific Logs

Contains information about login attempts, network connections, binary executions like execve() and SSH sessions

VPC Flow Logs

Records information on all outbound and inbound traffic from and to the VMs deployed within the VPC

All of these logs should flow to a **central logging solution** such as **Stackdriver** or **BigQuery** where these can be collected and analyzed

Step 1: Collect All the Logs

Analysis of log is helpful during forensic investigation as they provide data pertaining to the activities that occurred in the google cloud environment and help investigators identify suspicious events. Some of the important log sources on GCP are discussed below:

Cloud Audit Logs

Most Google Cloud services generate four types of audit logs for each project that can help the security team understand what happened to their Google Cloud resources at a given point of time and who did that. These logs are given below:

- **Admin Activity Audit Logs:** These logs record details about administrative API calls made for modifying the metadata or configuration of deployed resources. These logs are free and enabled by default. Viewing these logs requires specific IAM roles such as Project/Viewer or Logging/Logs Viewer. The maximum retention period of these logs is 400 days.
- **Data Access Audit Logs:** These logs contain records of user-made API calls for reading, creating, or modifying data of Google Cloud resources that are not publicly shared. These logs need to be configured and enabled manually. Viewing these logs requires specific IAM roles such as Project/Owner. The maximum retention period of these logs is 30 days.
- **System Event Audit Logs:** These logs are generated by Google systems that contain activities related to configuration changes in Google Cloud resources. These logs are enabled by default. The maximum retention period of these logs is 400 days.
- **Policy Denied Audit Logs:** These logs are generated when a user is denied access to a particular Google Cloud service due to a violation of security policies. These logs are

chargeable but enabled by default. Google Cloud retains these logs for 30 days and then deletes them.

Google Kubernetes Engine (GKE) Audit Logs

Two types of audit logs are generated by GKE: GKE audit logs and Kubernetes audit logging. API calls made to the Kubernetes server are recorded in GKE Audit logs that are useful for looking into suspicious API calls and monitoring alerts and statistics during forensic investigations.

Kubernetes also record a series of activities occurring within a GKE cluster in a chronological order which is known as Kubernetes audit logging. These log entries are governed by a pre-set audit policy and stored at the backend. The audit policy determines which events and what kind of data should be recorded. Some audit-levels include **Metadata**, **Request,** and **RequestResponse**.

Container Logs

GKE also collects logs from running containers, clusters, nodes and pods that are accessed and stored by a logging agent. The primary sources of container logs in GKE include the following:

- Container runtime logs
- Standard output and standard error logs
- Log entries for system components

OS Specific Logs

All Compute Engines or VMs running on GKE produce their own logs which can be collected and sent to Cloud Audit logs. Additionally, administrators can configure their Linux machines with **auditd** daemon to pull the system-generated logs which can provide valuable information on login attempts, network connections, binary executions like **`execve()`**, and SSH sessions during forensic investigations.

VPC Flow Logs

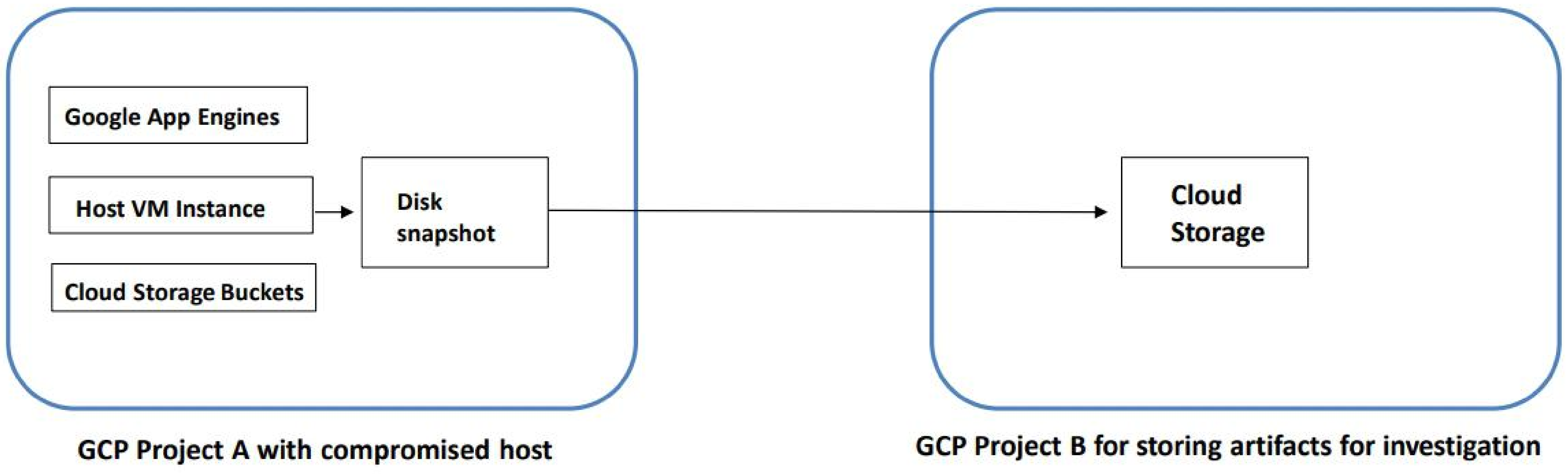
These logs record outbound and inbound network traffic sent from and to VMs instances within a subnet on GCP. VPC flow logs can be quite helpful in performing real-time network monitoring and analysis in the event of a security breach.

As a part of IRM planning, organizations should collect these logs periodically and choose any one of the following destinations for their long-term storage:

- **Cloud Storage:** Organizations can export these logs to a cloud storage bucket that retains them as JSON files.
- **Pub/Sub:** This Google Cloud storage service can be used by organizations if they want to collect and integrate the logs with some third-party tools such as Splunk.
- **BigQuery:** Organizations can also export their logs to BigQuery which stores the log data in a table format.

Step 2: Take a Snapshot of the Disk of Host VM

- As a part of the incident response plan, organizations must maintain a **separate GCP project** that is used collecting and analyzing artifacts.
- For example, if the analysis of the logs indicate toward a security breach within the GCP project A, the next step is to locate the affected nodes and the attached disks that might has been compromised
- The investigating team then needs to create a **snapshot** of the host VM's disk and send it to the cloud storage of GCP Project B meant of handling artifacts to perform forensic analysis.

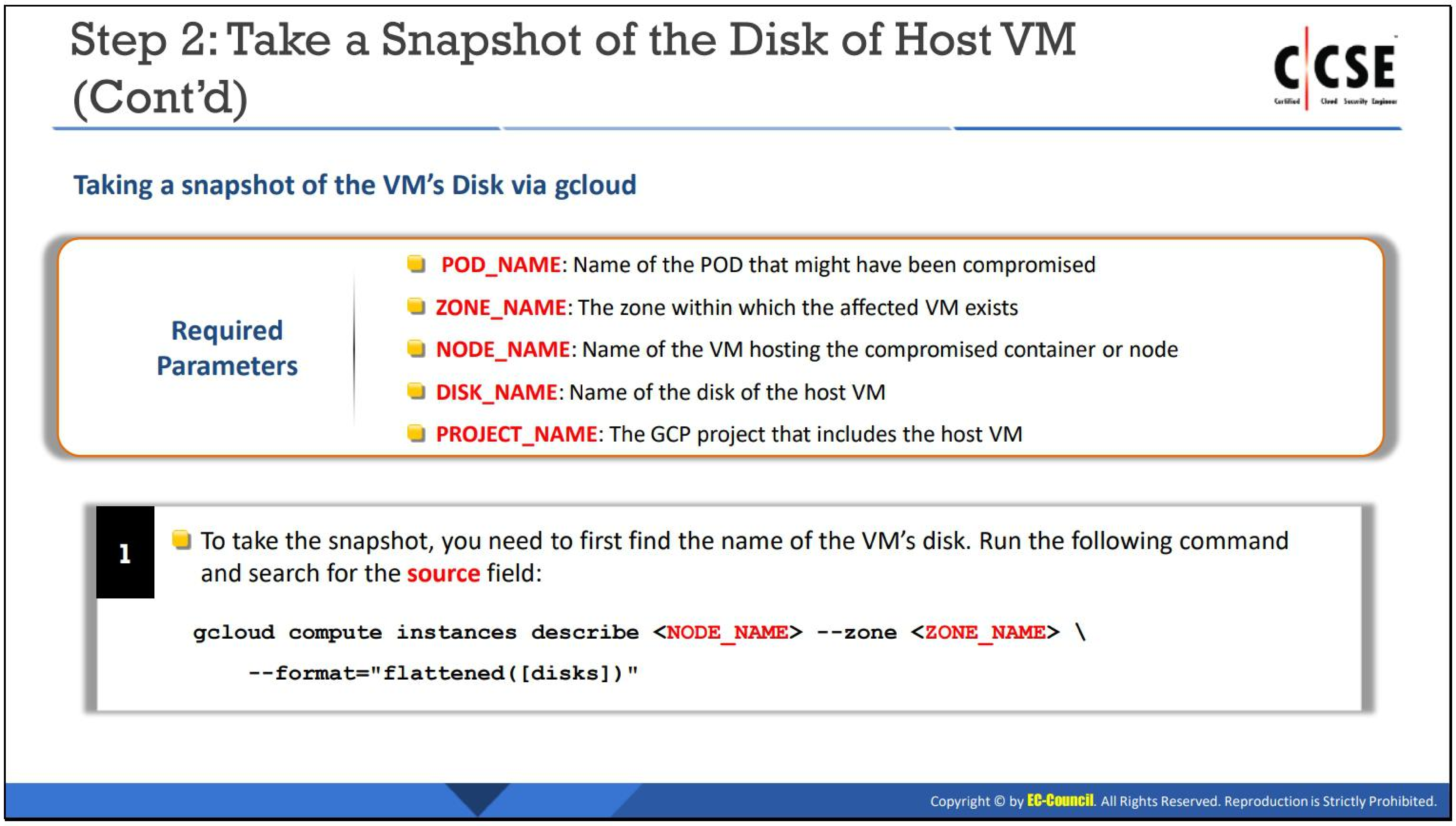

Google App Engines
Host VM Instance
Cloud Storage Buckets
Disk snapshot
Cloud Storage
GCP Project A with compromised host
GCP Project B for storing artifacts for investigation

Step 2: Take a Snapshot of the Disk of Host VM (Cont'd)

Taking a snapshot of the VM's Disk on Google Cloud Console

- Navigate to the **Create a Snapshot** page on the console
- When the page opens, provide a name for the snapshot and a description (optional).
- In the **Source Disk** section, select the disk name of the VM that you want to create a snapshot of. You might a **Regional** or **Multi-regional** location for the snapshot under the **Location** section.
- Once done, click on the **Create** button. The snapshot of the affected disk is now successfully created.

Note: The task of creating snapshots of VM's disks can be performed even when the instance is in a running state



Step 2: Take a Snapshot of the Disk of Host VM (Cont'd)

CCSE
Certified Cloud Security Engineer

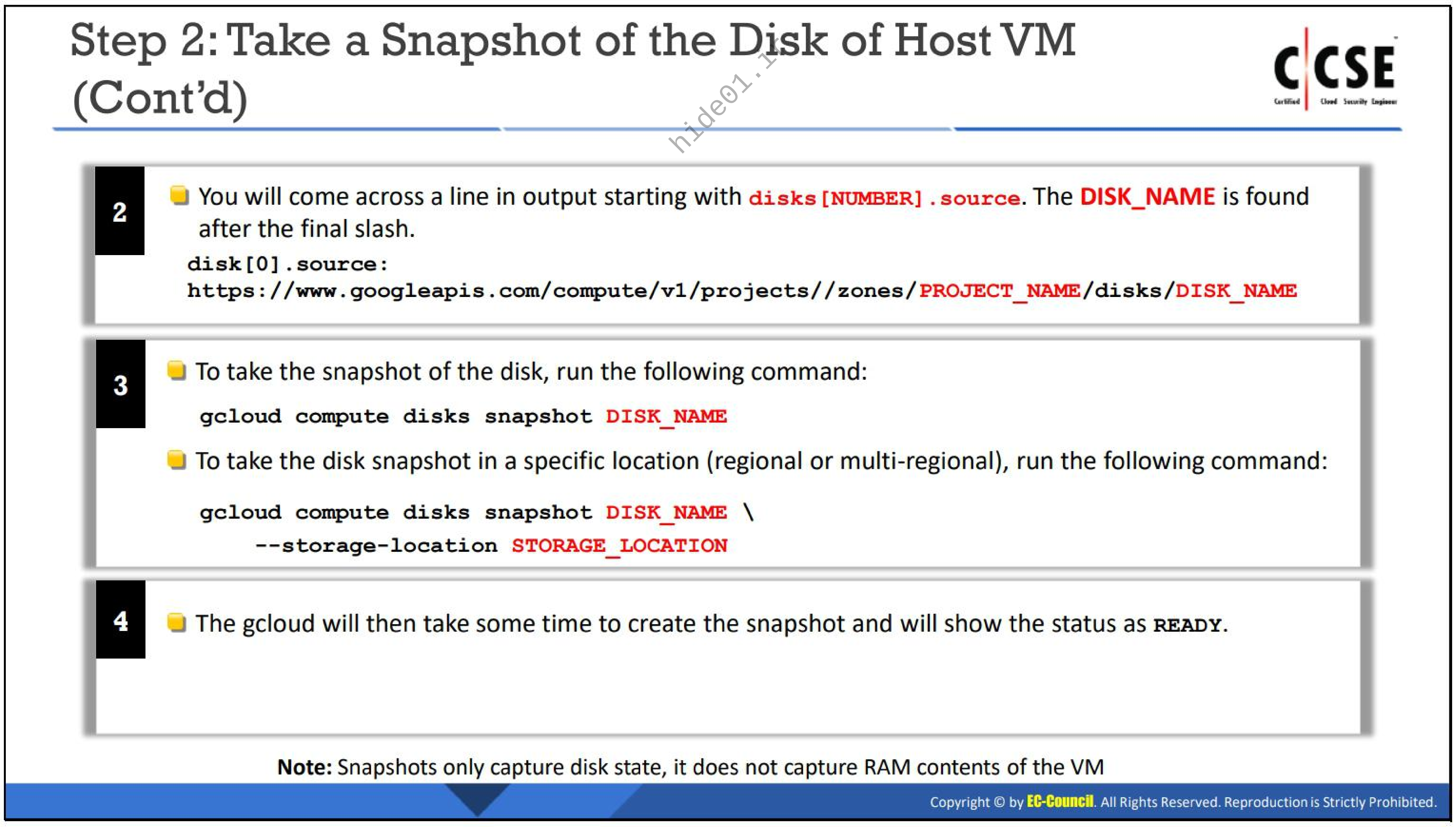
Taking a snapshot of the VM's Disk via gcloud

Required Parameters

- **POD_NAME**: Name of the POD that might have been compromised
- **ZONE_NAME**: The zone within which the affected VM exists
- **NODE_NAME**: Name of the VM hosting the compromised container or node
- **DISK_NAME**: Name of the disk of the host VM
- **PROJECT_NAME**: The GCP project that includes the host VM

1 To take the snapshot, you need to first find the name of the VM's disk. Run the following command and search for the **source** field:

```
gcloud compute instances describe <NODE_NAME> --zone <ZONE_NAME> \
    --format="flattened([disks])"
```



Step 2: Take a Snapshot of the Disk of Host VM (Cont'd)

CCSE
Certified Cloud Security Engineer

2 You will come across a line in output starting with `disks[NUMBER].source`. The **DISK_NAME** is found after the final slash.

```
disk[0].source:
https://www.googleapis.com/compute/v1/projects//zones/PROJECT_NAME/disks/DISK_NAME
```

3 To take the snapshot of the disk, run the following command:

```
gcloud compute disks snapshot DISK_NAME
```

To take the disk snapshot in a specific location (regional or multi-regional), run the following command:

```
gcloud compute disks snapshot DISK_NAME \
    --storage-location STORAGE_LOCATION
```

4 The gcloud will then take some time to create the snapshot and will show the status as `READY`.

Note: Snapshots only capture disk state, it does not capture RAM contents of the VM

Step 2: Take a Snapshot of the Disk of Host VM

If the analysis of collected log data indicates a security incident involving a running container, the next step should be to identify the affected node and disks attached. It is recommended that an organization maintains a separate GCP project that is solely used for handling and storing forensic artifacts. Therefore, once the affected node is identified, the forensic team should create a snapshot of the host VM which can be examined later. Disk snapshots can be created even when the VM is in running state. After taking the disk snapshot, it should be transferred to the Cloud Storage of the GCP Project maintained for handling forensic artifacts. Investigators can use both the Google Cloud Console and gcloud command-line tool to create the snapshot.

You, as a forensic investigator, need to perform the following steps to create a snapshot of the VM's Disk on Google Cloud Console:

- Navigate to the Create a Snapshot page on the Console
- When the page opens, provide a name for the snapshot and a description (optional)
- In the Source Disk section, select the disk name of the VM you want to create a snapshot of. You may also select a Regional or Multi-regional location for the snapshot under the Location section.
- Once done, click on the Create button. The snapshot of the affected disk is now successfully created.

If you need to create the disk snapshot via gcloud, it is important to know some required parameters as given below:

- **POD_NAME:** Name of the pod that might have been compromised
- **ZONE_NAME:** The zone within which the host VM exists
- **NODE_NAME:** Name of the VM hosting the compromised pod/container
- **DISK_NAME:** Name of the disk of the host VM
- **PROJECT_NAME:** The GCP project that includes the host VM

While using gcloud, the first task is to identify the name of the disk that is attached to the host VM. To locate it, execute the command given below:

```
gcloud compute instances describe NODE_NAME --zone COMPUTE_ZONE \
--format="flattened([disks])"
```

This command would generate an output with a line starting with **`disks[NUMBER].source`**. The disk name can be found after the final slash:

```
disks[0].source:
https://www.googleapis.com/compute/v1/projects/PROJECT_NAME/zones/COMP
UTE_ZONE/disks/DISK_NAME
```

Copy the DISK_NAME and run the following command to create the snapshot:

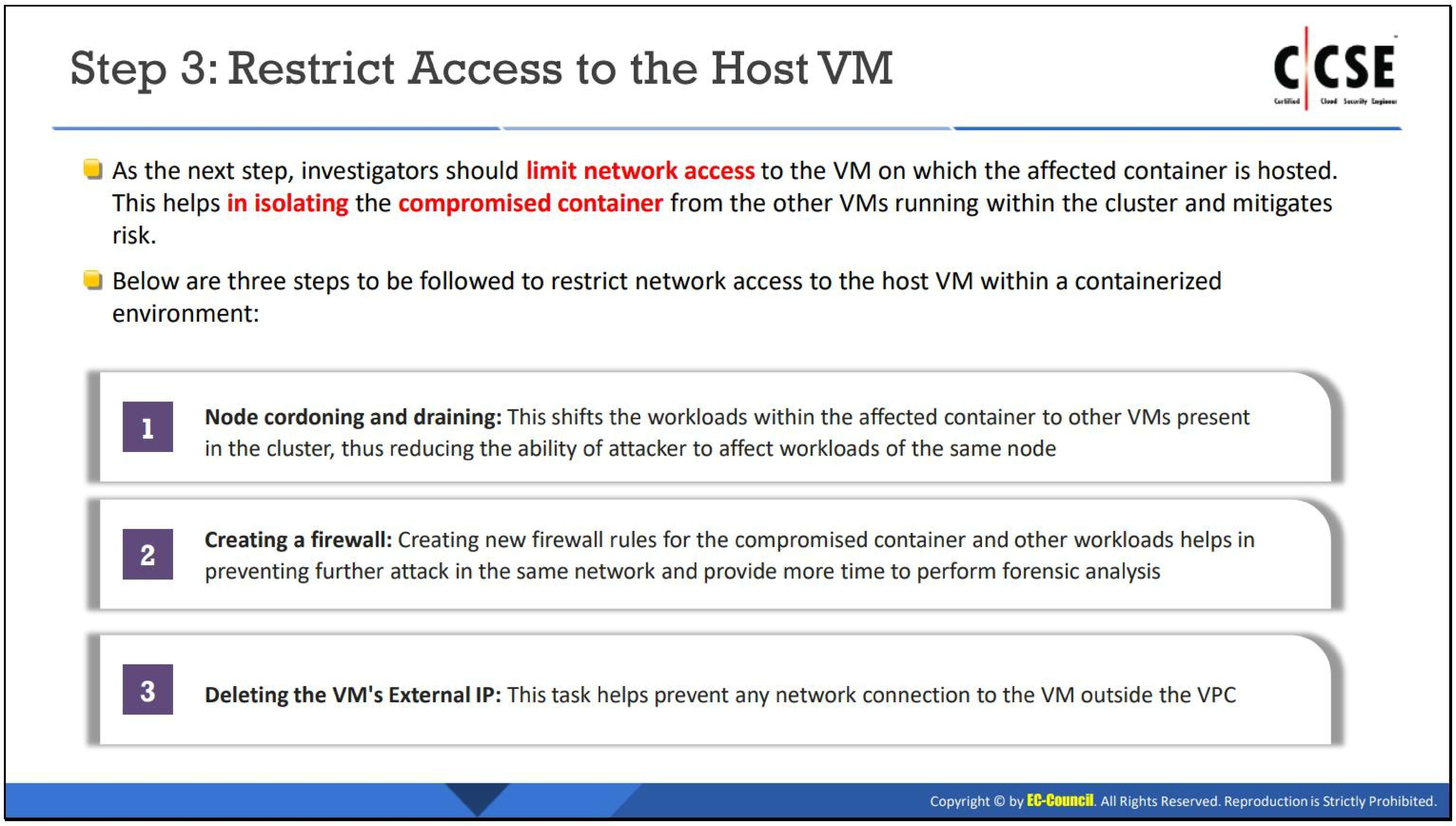
```
gcloud compute disks snapshot DISK_NAME
```

If you want to store the snapshot in a specific location, add the **`--storage-location`** flag in the following manner:

```
gcloud compute disks snapshot DISK_NAME \
--storage-location STORAGE_LOCATION
```

To complete this task of snapshot creation via Google Cloud Console or gcloud, you would require specific permissions such as **`compute.disks.createSnapshot`** on the disk, **`compute.instances.useReadOnly`** on the host VM and **`compute.snapshots.create`** on the project.

Note: Snapshots only capture disk state, they do not capture RAM contents of the VM



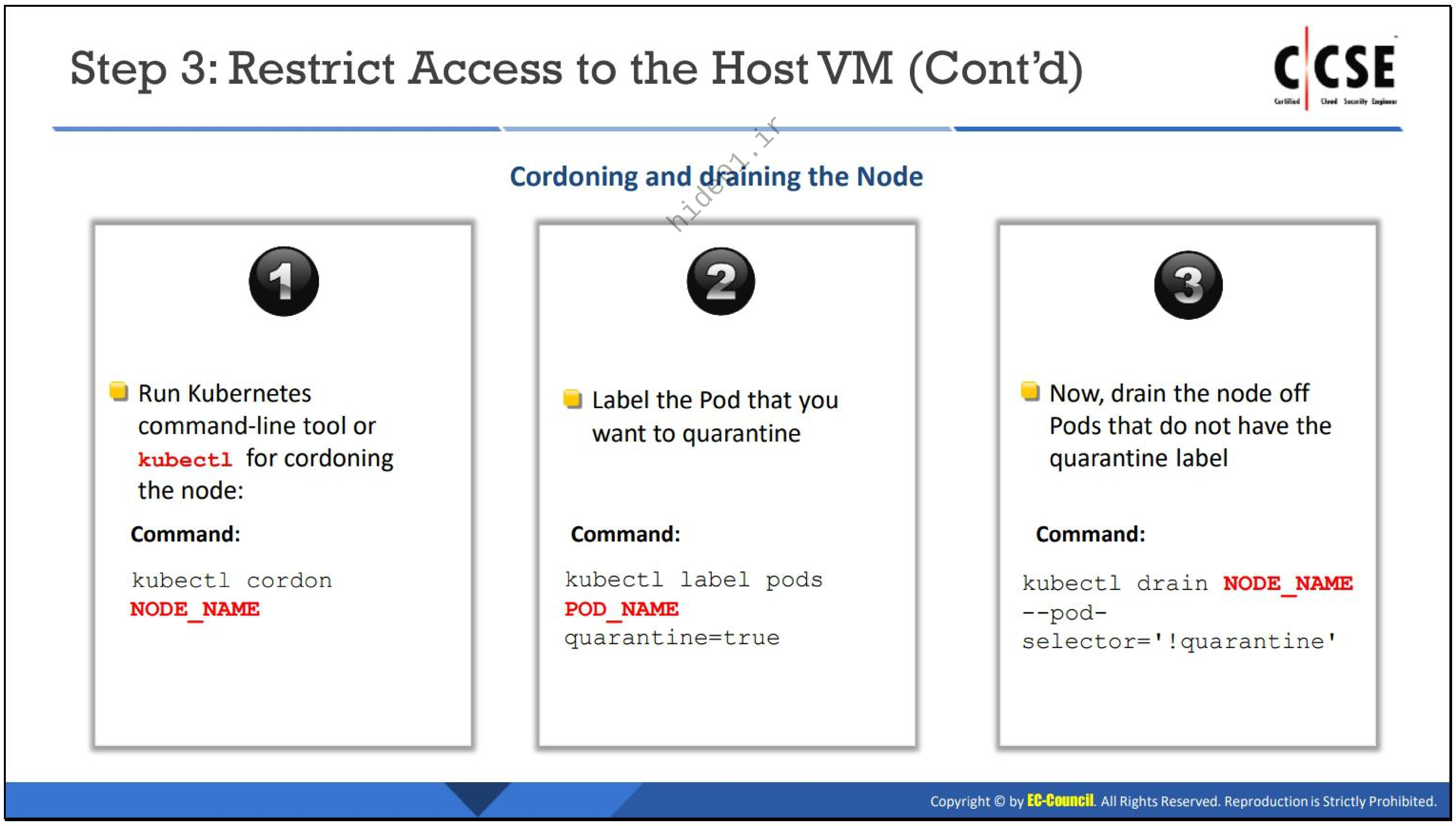
Step 3: Restrict Access to the Host VM

- As the next step, investigators should **limit network access** to the VM on which the affected container is hosted. This helps **in isolating** the **compromised container** from the other VMs running within the cluster and mitigates risk.
- Below are three steps to be followed to restrict network access to the host VM within a containerized environment:

1 **Node cordoning and draining:** This shifts the workloads within the affected container to other VMs present in the cluster, thus reducing the ability of attacker to affect workloads of the same node

2 **Creating a firewall:** Creating new firewall rules for the compromised container and other workloads helps in preventing further attack in the same network and provide more time to perform forensic analysis

3 **Deleting the VM's External IP:** This task helps prevent any network connection to the VM outside the VPC



Step 3: Restrict Access to the Host VM (Cont'd)

Cordoning and draining the Node

1

- Run Kubernetes command-line tool or **kubectl** for cordoning the node:

Command:

```
kubectl cordon
NODE_NAME
```

2

- Label the Pod that you want to quarantine

Command:

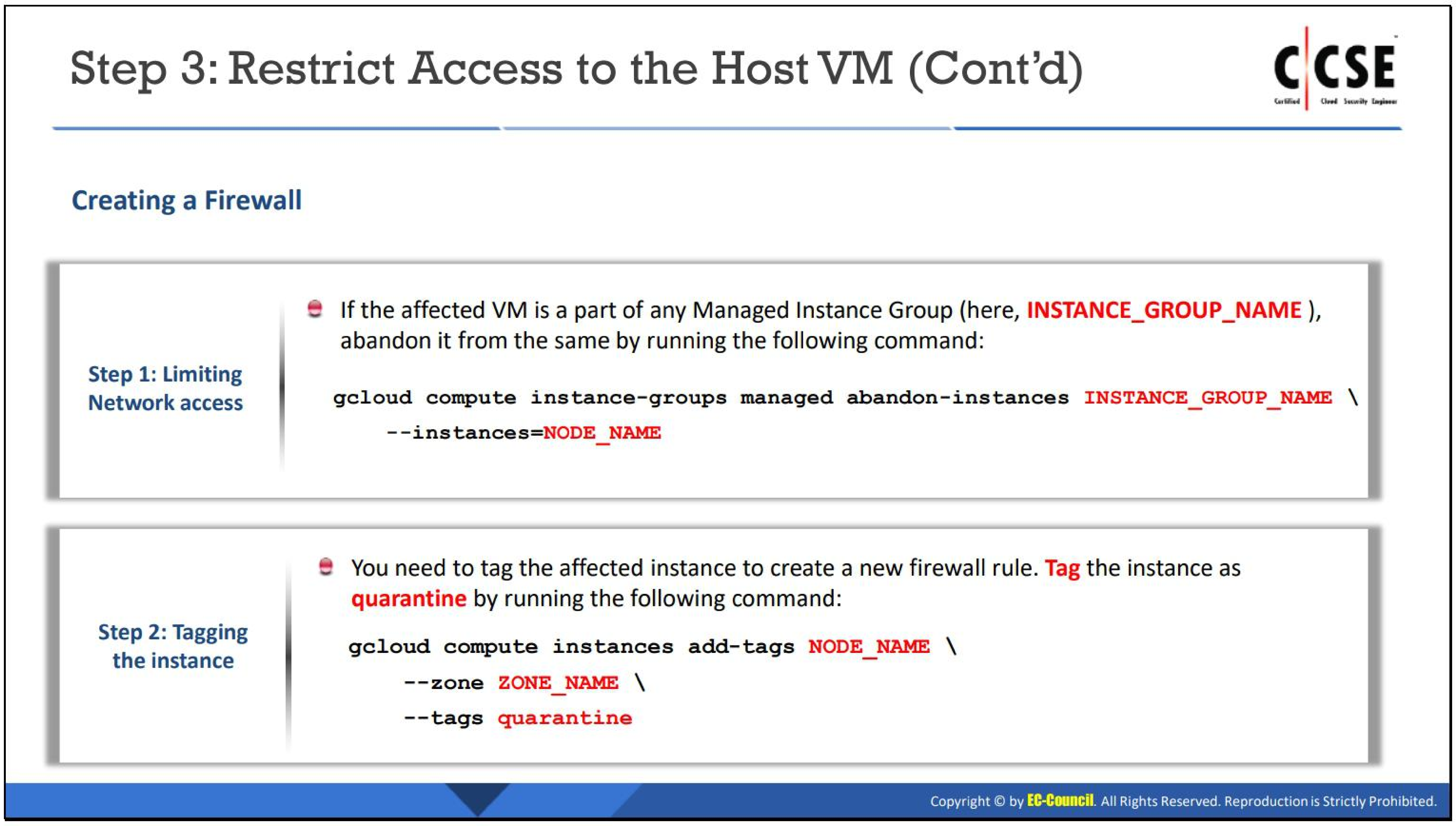
```
kubectl label pods
POD_NAME
quarantine=true
```

3

- Now, drain the node off Pods that do not have the quarantine label

Command:

```
kubectl drain NODE_NAME
--pod-
selector='!quarantine'
```



Step 3: Restrict Access to the Host VM (Cont'd)

Creating a Firewall

Step 1: Limiting Network access

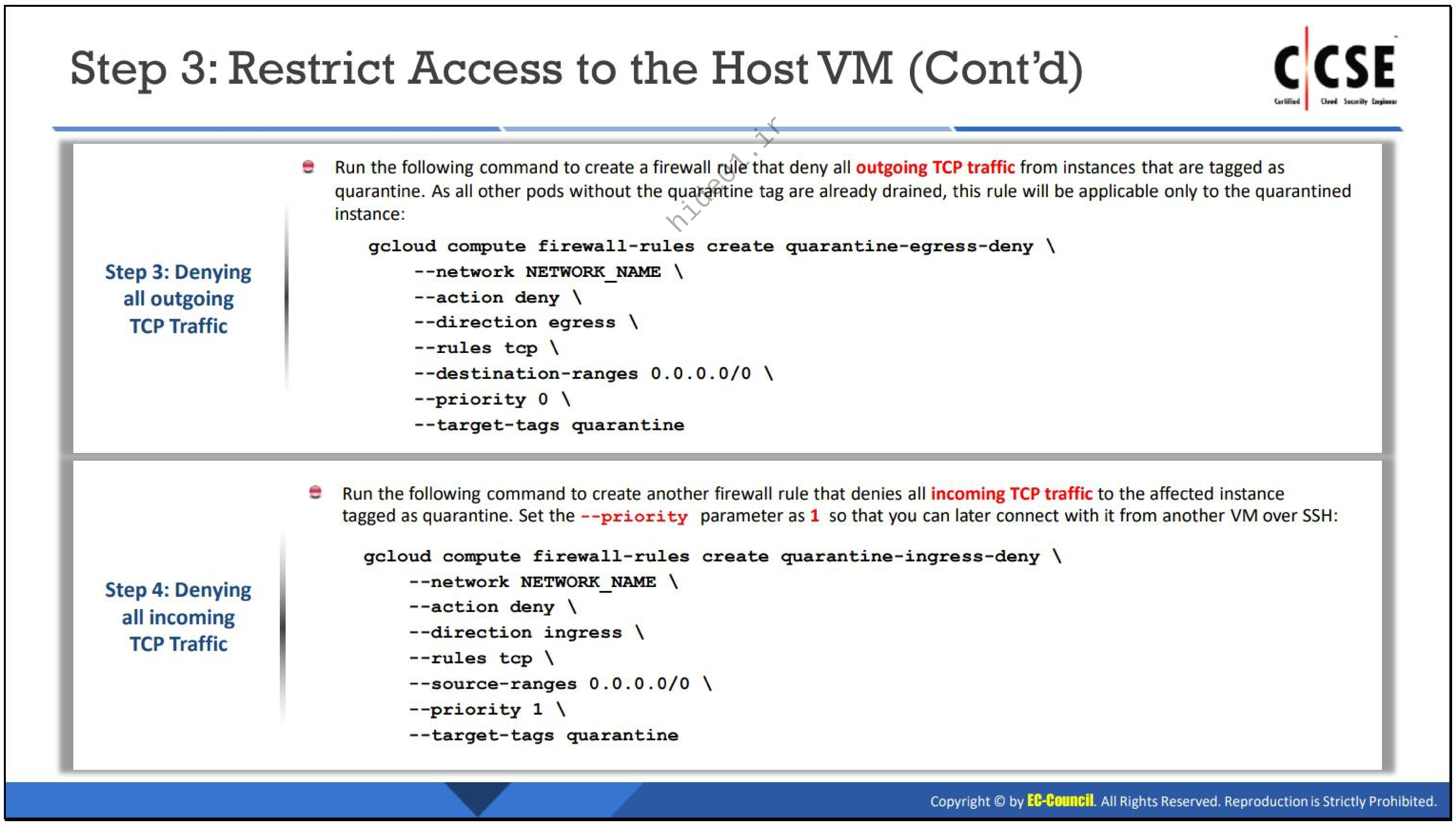
- If the affected VM is a part of any Managed Instance Group (here, **INSTANCE_GROUP_NAME**), abandon it from the same by running the following command:

```
gcloud compute instance-groups managed abandon-instances INSTANCE_GROUP_NAME \
    --instances=NODE_NAME
```

Step 2: Tagging the instance

- You need to tag the affected instance to create a new firewall rule. **Tag** the instance as **quarantine** by running the following command:

```
gcloud compute instances add-tags NODE_NAME \
    --zone ZONE_NAME \
    --tags quarantine
```



Step 3: Restrict Access to the Host VM (Cont'd)

Step 3: Denying all outgoing TCP Traffic

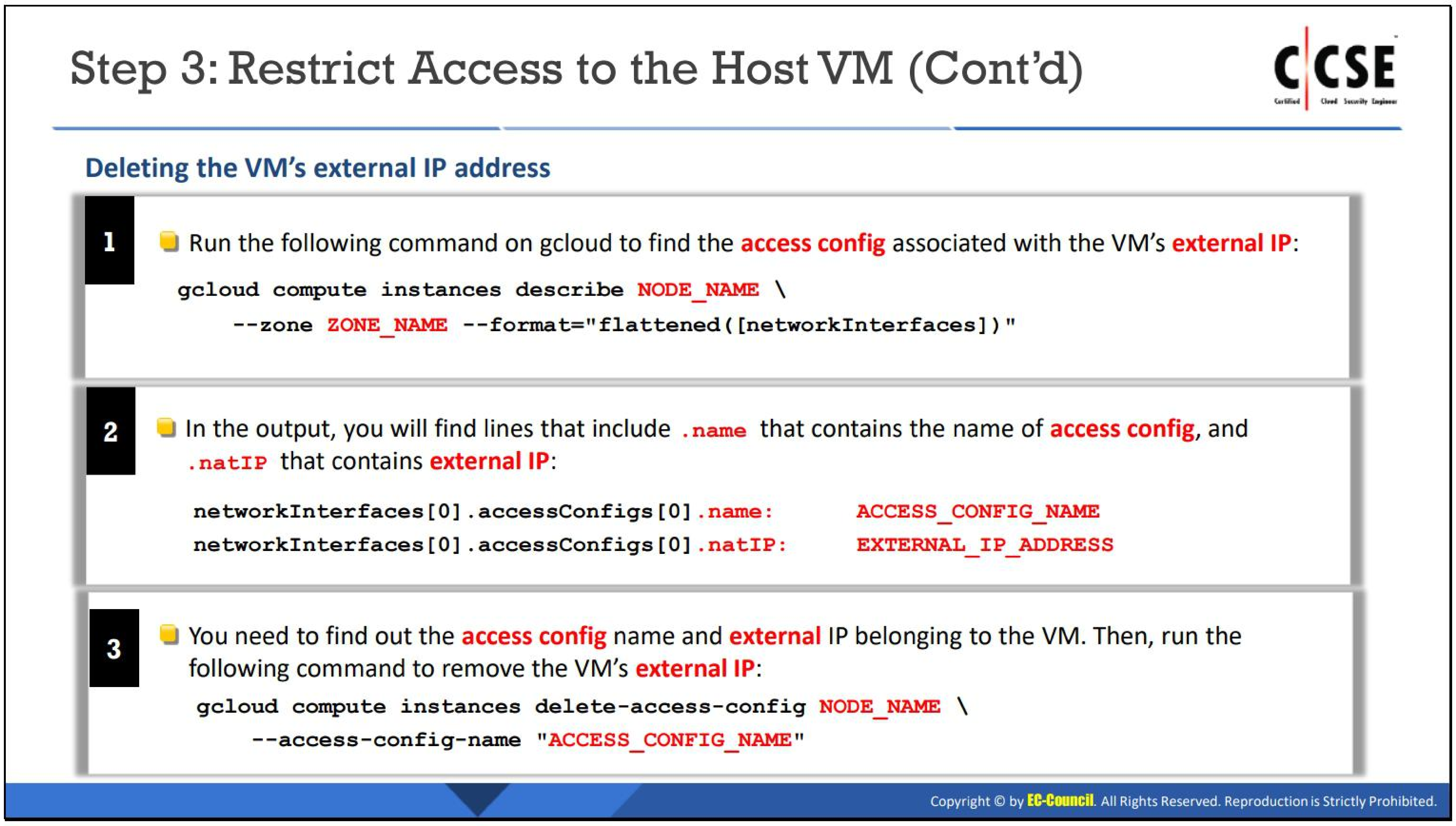
- Run the following command to create a firewall rule that deny all **outgoing TCP traffic** from instances that are tagged as quarantine. As all other pods without the quarantine tag are already drained, this rule will be applicable only to the quarantined instance:

```
gcloud compute firewall-rules create quarantine-egress-deny \
    --network NETWORK_NAME \
    --action deny \
    --direction egress \
    --rules tcp \
    --destination-ranges 0.0.0.0/0 \
    --priority 0 \
    --target-tags quarantine
```

Step 4: Denying all incoming TCP Traffic

- Run the following command to create another firewall rule that denies all **incoming TCP traffic** to the affected instance tagged as quarantine. Set the `--priority` parameter as **1** so that you can later connect with it from another VM over SSH:

```
gcloud compute firewall-rules create quarantine-ingress-deny \
    --network NETWORK_NAME \
    --action deny \
    --direction ingress \
    --rules tcp \
    --source-ranges 0.0.0.0/0 \
    --priority 1 \
    --target-tags quarantine
```



Step 3: Restrict Access to the Host VM (Cont'd)

Deleting the VM's external IP address

1. Run the following command on gcloud to find the **access config** associated with the VM's **external IP**:

```
gcloud compute instances describe NODE_NAME \
    --zone ZONE_NAME --format="flattened([networkInterfaces])"
```

2. In the output, you will find lines that include `.name` that contains the name of **access config**, and `.natIP` that contains **external IP**:

```
networkInterfaces[0].accessConfigs[0].name:      ACCESS_CONFIG_NAME
networkInterfaces[0].accessConfigs[0].natIP:     EXTERNAL_IP_ADDRESS
```

3. You need to find out the **access config** name and **external** IP belonging to the VM. Then, run the following command to remove the VM's **external IP**:

```
gcloud compute instances delete-access-config NODE_NAME \
    --access-config-name "ACCESS_CONFIG_NAME"
```

Step 3: Restrict Access to the Host VM

In Google Cloud environment, restricting access to a VM hosting the affected container helps in partially quarantining it from the VMs running in the same cluster. This mitigates risk to an extent but does not completely stop an attack as the attacker can still exploit the existing vulnerability to move laterally within the network.

There are three stages to restricting access to a VM hosting a compromised container:

1. **Node Cordoning and Draining:**

   This process enables the forensic team to shift workloads within the affected container to other VMs present in the cluster. This can reduce the ability of the attacker to affect workloads present in the same node.

   - You can use **`kubectl`**, the command-line tool of Kubernetes to perform node cordoning and draining. Following is the command for cordoning a node:

   ```
   kubectl cordon NODE_NAME
   ```

   The output would show if any pods are scheduled within the specific node. In such a case, you need to drain all other unaffected pods scheduled on it. Running the above command would make the specified node unschedulable and detach other workloads from it which might result in downtime.

   - To drain the other pods from the node, tag the pod you want to isolate by running the command below:

   ```
   kubectl label pods POD_NAME quarantine=true
   ```

   This would provide a quarantine label to the specified pod.

- Now, run the following command to drain all other pods that do not have any quarantine label:

```
kubectl drain NODE_NAME --pod-selector='!quarantine'
```

2. **Creating a Firewall:**

This is the second stage in which the forensic team needs to block all incoming and outgoing traffic coming to and going from the host VM. This helps in quarantining the affected VM from other workloads present in the same network.

- If the host VM is part of any Managed Instance Group, the first task is to abandon it which allows more time to conduct forensic analysis. execute the command as given below:

```
gcloud compute instance-groups managed abandon-instances
INSTANCE_GROUP_NAME \
   --instances=NODE_NAME
```

- To create new firewall rules, you need to tag the VM instance:

```
gcloud compute instances add-tags NODE_NAME \
    --zone COMPUTE_ZONE \
    --tags quarantine
```

This command will provide a quarantine tag to the VM instance. Check that no other VMs are tagged the same before running this command to avoid conflicts.

- Run the following command to create a firewall rule that denies all outgoing TCP traffic from instances that are tagged as quarantine. As all other pods without the quarantine tag are already drained, this rule will apply only to the quarantined VM instance:

```
gcloud compute firewall-rules create quarantine-egress-deny
\
    --network NETWORK_NAME \
    --action deny \
    --direction egress \
    --rules tcp \
    --destination-ranges 0.0.0.0/0 \
    --priority 0 \
    --target-tags quarantine
```

- Run the following command to create another firewall rule that denies all incoming TCP traffic to the affected instance tagged as quarantine:

```
gcloud compute firewall-rules create quarantine-ingress-deny
\
    --network NETWORK_NAME \
            --action deny \
            --direction ingress \
            --rules tcp \
            --source-ranges 0.0.0.0/0 \
            --priority 1 \
            --target-tags quarantine
```

In the above command, the `--priority` flag is set to 1 which allows connecting to the quarantine instance from another instance over SSH connection in case any live inspection of the VM is required.

3. **Deleting VM’s External IP Address:**

To delete the external IP address of the VM, you need to first locate the access config name linked with it:

```
gcloud compute instances describe NODE_NAME \
--zone ZONE_NAME --format="flattened([networkInterfaces])"
```

In the output, you will see lines with `.name`, which contains the name of access config, and `.natIP` which contains external IP:

```
networkInterfaces[0].accessConfigs[0].name:
ACCESS_CONFIG_NAME

networkInterfaces[0].accessConfigs[0].natIP:
EXTERNAL_IP_ADDRESS
```

Find out the access config name and external IP of the host VM and run the following command to remove the external IP:

```
gcloud compute instances delete-access-config NODE_NAME \
    --access-config-name "ACCESS_CONFIG_NAME"
```

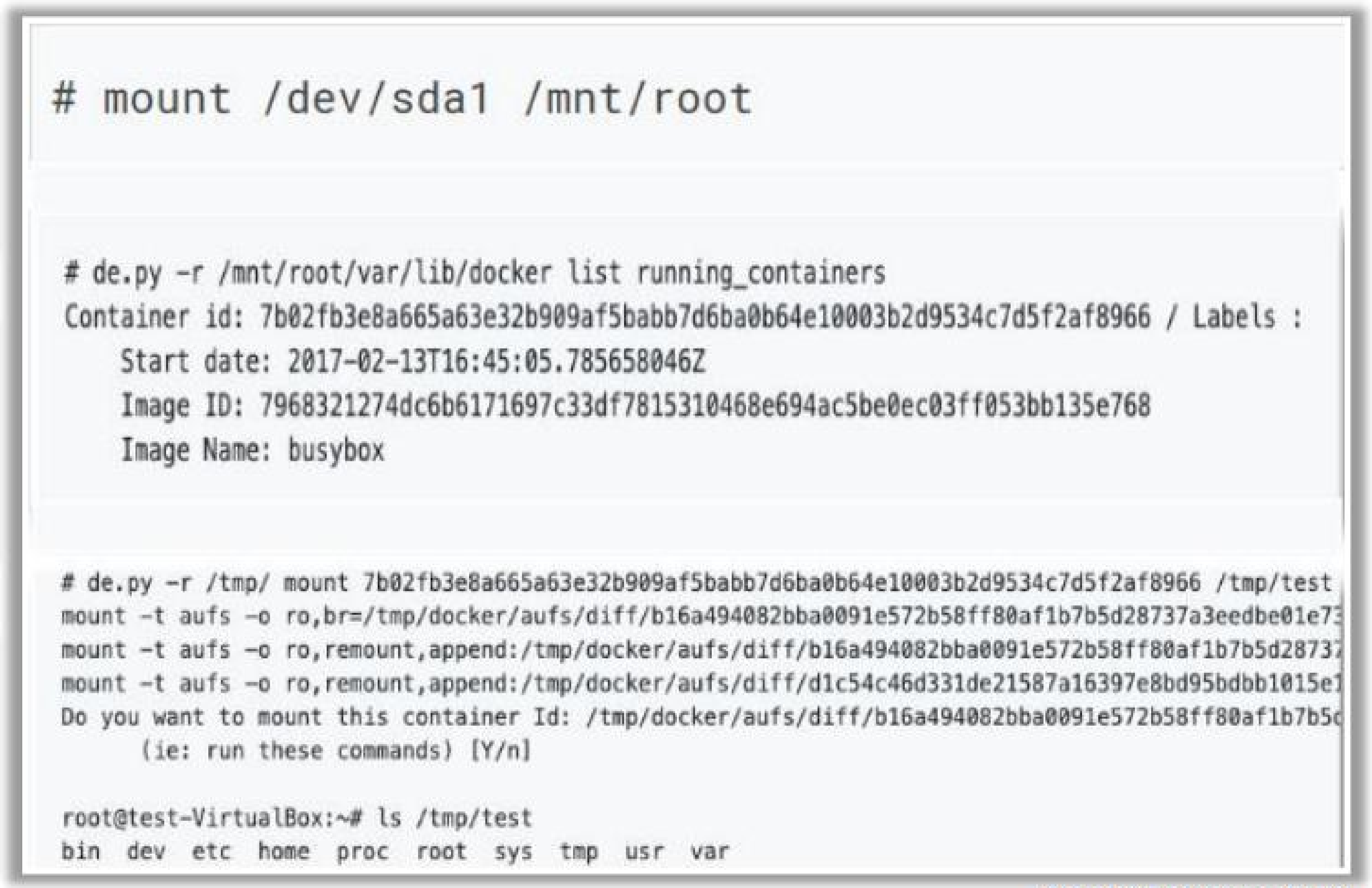
Step 4: Examine the Snapshot using Docker-Explorer

- **Docker explorer** is an open-source project supported by Google that helps forensic investigators examine **docker filesystems** offline from via a **disk snapshot**

1 Using docker explorer, **attach** and **mount** the disk snapshot

2 when mounted, list the **container IDs** that were running when the disk snapshot was created

3 Once the suspect container ID is found, you can mount that particular **container filesystem** to examine it further



```
# mount /dev/sda1 /mnt/root
```

```
# de.py -r /mnt/root/var/lib/docker list running_containers
Container id: 7b02fb3e8a665a63e32b909af5babb7d6ba0b64e10003b2d9534c7d5f2af8966 / Labels :
    Start date: 2017-02-13T16:45:05.785658046Z
    Image ID: 7968321274dc6b6171697c33df7815310468e694ac5be0ec03ff053bb135e768
    Image Name: busybox
```

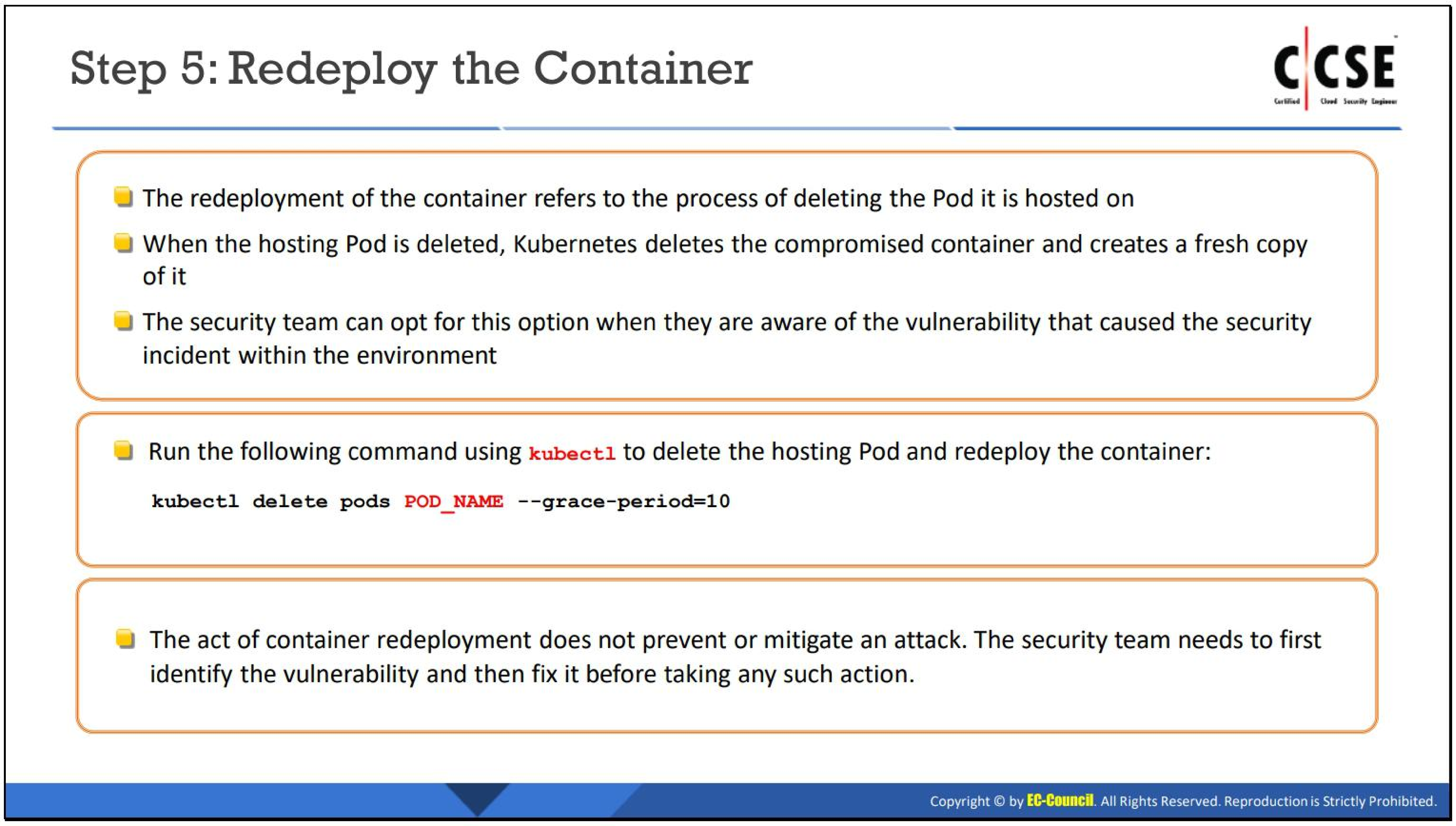
```
# de.py -r /tmp/ mount 7b02fb3e8a665a63e32b909af5babb7d6ba0b64e10003b2d9534c7d5f2af8966 /tmp/test
mount -t aufs -o ro,br=/tmp/docker/aufs/diff/b16a494082bba0091e572b58ff80af1b7b5d28737a3eedbe01e73
mount -t aufs -o ro,remount,append:/tmp/docker/aufs/diff/b16a494082bba0091e572b58ff80af1b7b5d28737
mount -t aufs -o ro,remount,append:/tmp/docker/aufs/diff/d1c54c46d331de21587a16397e8bd95bdbb1015e1
Do you want to mount this container Id: /tmp/docker/aufs/diff/b16a494082bba0091e572b58ff80af1b7b5c
      (ie: run these commands) [Y/n]

root@test-VirtualBox:~# ls /tmp/test
bin  dev  etc  home  proc  root  sys  tmp  usr  var
```

https://static.sched.com

Step 4: Examine the Snapshot using Docker Explorer

Docker explorer is an open-source project supported by Google that helps forensic investigators examine docker filesystems offline via a disk snapshot.

After installing the tool, investigators need to first mount the disk snapshot using the mount command. Then, they can list the container IDs that were running when the snapshot was taken. This process helps in identifying the suspect container ID. Once found, they can mount the compromised container's filesystem and perform further forensic analysis.



Step 5: Redeploy the Container

- The redeployment of the container refers to the process of deleting the Pod it is hosted on
- When the hosting Pod is deleted, Kubernetes deletes the compromised container and creates a fresh copy of it
- The security team can opt for this option when they are aware of the vulnerability that caused the security incident within the environment

- Run the following command using `kubectl` to delete the hosting Pod and redeploy the container:

```
kubectl delete pods POD_NAME --grace-period=10
```

- The act of container redeployment does not prevent or mitigate an attack. The security team needs to first identify the vulnerability and then fix it before taking any such action.

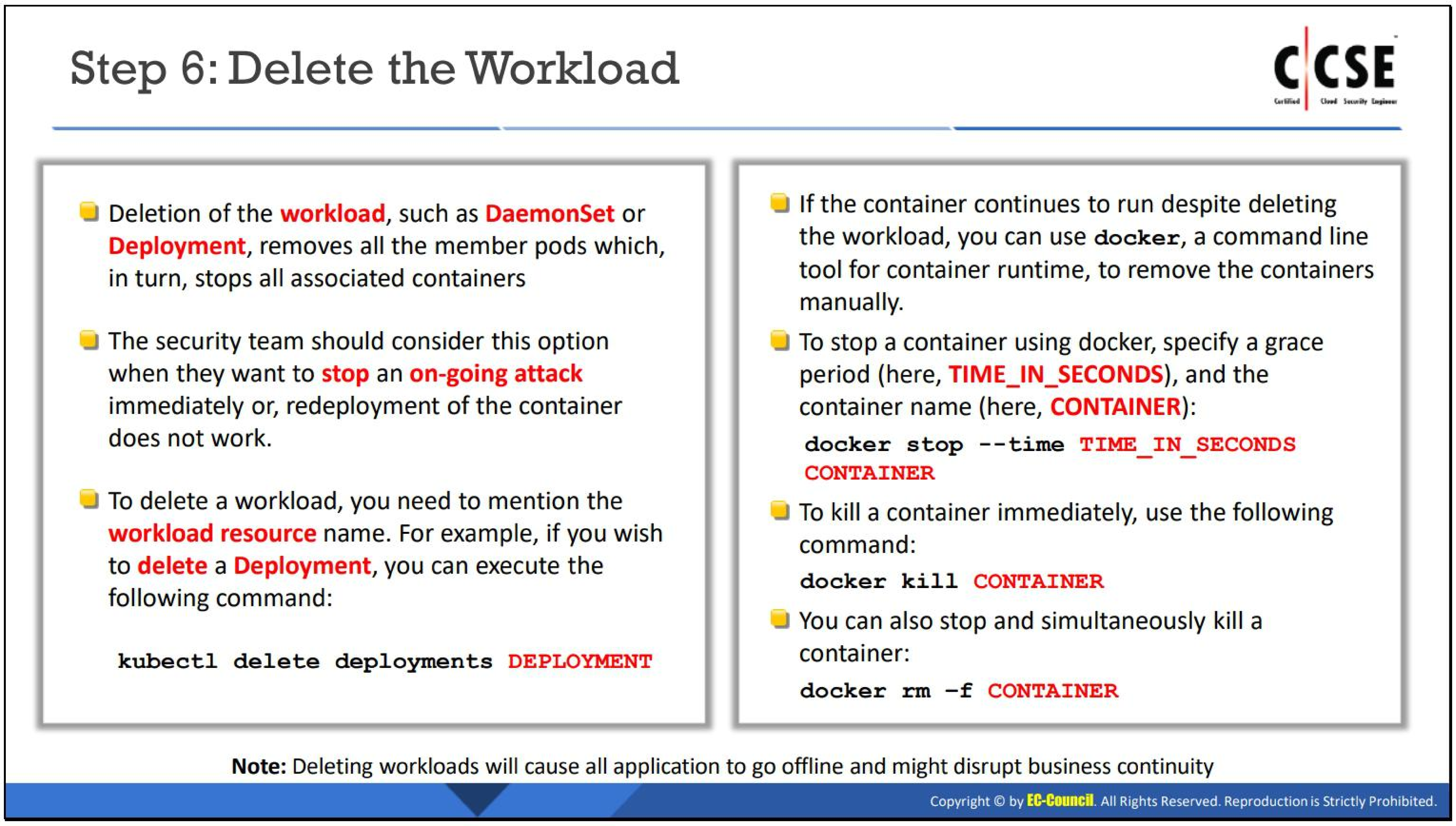
Step 5: Redeploy the Container

Redeployment of the container in GKE environment refers to the process of deleting the Pods that it is hosted on. With the hosting pod getting deleted, Kubernetes removes the compromised container and creates another fresh copy of it which starts running.

The redeployment of the container can be considered when the investigating team is already aware of the security issue and has also found a fix for it. In case the Pod to be deleted is managed by a DaemonSet or Deployment, Kubernetes schedules a new pod with the act of deletion, which contains fresh containers.

You can use **`kubectl`** to delete a Pod hosting a compromised container:

```
kubectl delete pods POD_NAME --grace-period=10
```



Step 6: Delete the Workload

- Deletion of the **workload**, such as **DaemonSet** or **Deployment**, removes all the member pods which, in turn, stops all associated containers
- The security team should consider this option when they want to **stop** an **on-going attack** immediately or, redeployment of the container does not work.
- To delete a workload, you need to mention the **workload resource** name. For example, if you wish to **delete** a **Deployment**, you can execute the following command:

```
kubectl delete deployments DEPLOYMENT
```

- If the container continues to run despite deleting the workload, you can use `docker`, a command line tool for container runtime, to remove the containers manually.
- To stop a container using docker, specify a grace period (here, **TIME_IN_SECONDS**), and the container name (here, **CONTAINER**):

```
docker stop --time TIME_IN_SECONDS
CONTAINER
```

- To kill a container immediately, use the following command:

```
docker kill CONTAINER
```

- You can also stop and simultaneously kill a container:

```
docker rm –f CONTAINER
```

Note: Deleting workloads will cause all application to go offline and might disrupt business continuity

Step 6: Delete the Workload

Deletion of workloads, such as DaemonSet or Deployment, removes all the member pods which, in turn, stops all associated containers. This option should only be considered when the investigating team is aware that there is an ongoing attack and wants to stop it immediately. Deletion of the workload takes all the applications offline which might disrupt business continuity.

To delete the workload using **`kubectl`**, the investigator needs to mention the workload resource name. For example, if the investigator has to delete a Deployment, they should run the following command:

```
kubectl delete deployments DEPLOYMENT
```

It is, however, possible that the associated containers and pods continue to run even after a workload has been deleted. In such cases, the investigating team needs to remove them manually. They can use **`docker`**, a command-line tool for container runtime, for this purpose.

To stop a container using **`docker`** command line, the following command can be used:

```
docker stop --time TIME_IN_SECONDS CONTAINER
```

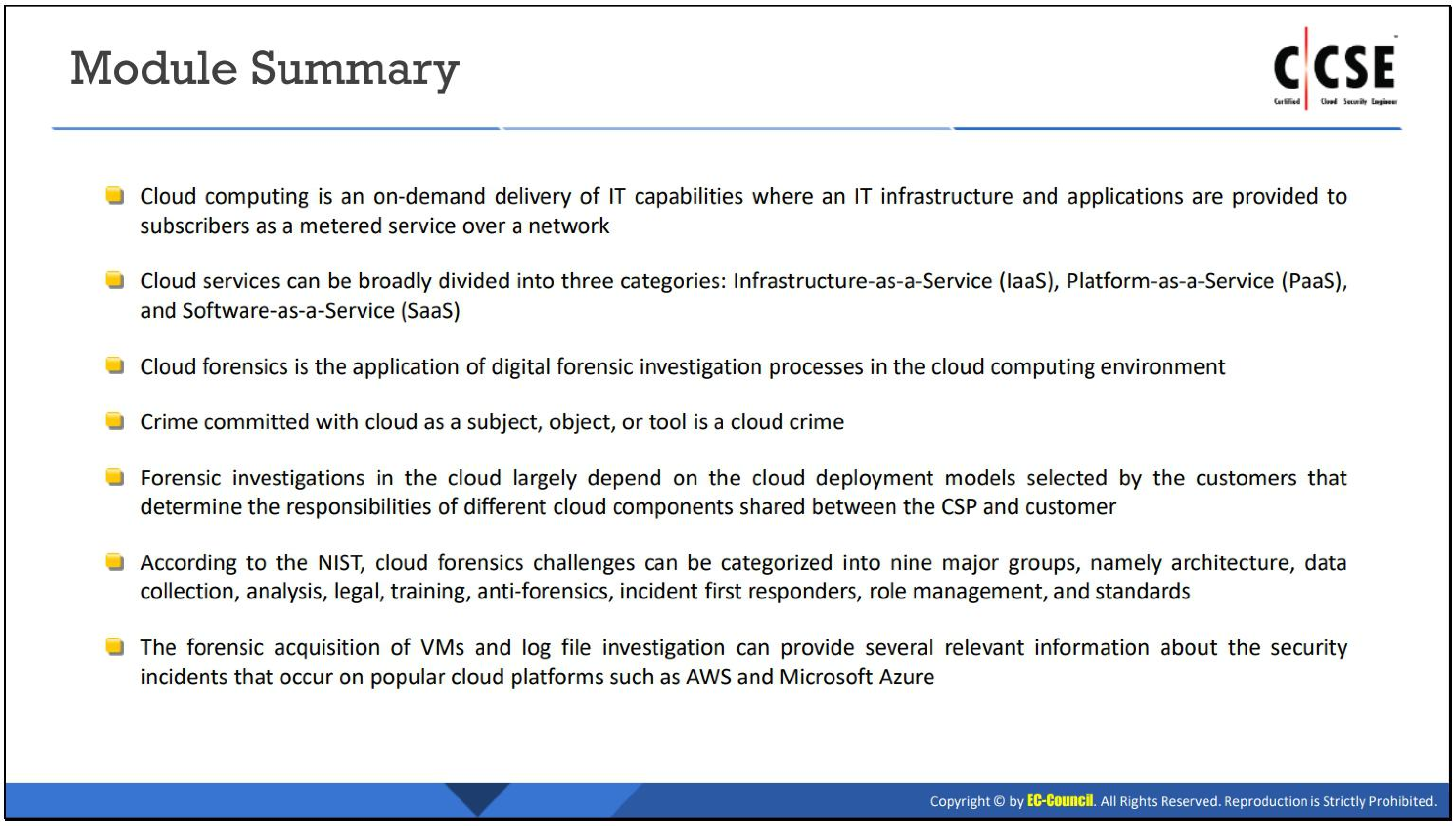
In the command above, the **`docker stop`** command will send a **`SIGTERM`** signal for the root process to stop and exit within the grace time period mentioned in the **`--time`** flag. If it does not exit by that time, the command will send a **`SIGKILL`** signal for stopping the container.

The investigating team can also use the **`docker kill`** command to stop a container, as this command sends the **`SIGKILL`** signal immediately:

```
docker kill CONTAINER
```

The following **`docker`** command can be used to stop and kill a container at once:

```
docker rm –f CONTAINER
```



Module Summary

- Cloud computing is an on-demand delivery of IT capabilities where an IT infrastructure and applications are provided to subscribers as a metered service over a network
- Cloud services can be broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS)
- Cloud forensics is the application of digital forensic investigation processes in the cloud computing environment
- Crime committed with cloud as a subject, object, or tool is a cloud crime
- Forensic investigations in the cloud largely depend on the cloud deployment models selected by the customers that determine the responsibilities of different cloud components shared between the CSP and customer
- According to the NIST, cloud forensics challenges can be categorized into nine major groups, namely architecture, data collection, analysis, legal, training, anti-forensics, incident first responders, role management, and standards
- The forensic acquisition of VMs and log file investigation can provide several relevant information about the security incidents that occur on popular cloud platforms such as AWS and Microsoft Azure

Module Summary

This module discussed the basic concepts related to cloud computing along with the major threats and attacks associated with the cloud environment. This module also discussed how investigators can conduct investigations on a cloud and examine the evidence data.

This module particularly focused on two major cloud computing platforms, namely AWS and Microsoft Azure. It discussed various cloud-based services provided by these CSPs, the sharing of responsibilities between the CSP and customer according to different cloud service models, and customer data storage by the CSPs.

It also highlighted different log files generated by different resources and services of AWS and Azure and how these files can be utilized while investigating security incidents. This module also elucidated the process of forensic acquisition of the affected VMs that run on these cloud platforms.

The key highlights of this module have been discussed below:

- Cloud computing is an on-demand delivery of IT capabilities where an IT infrastructure and applications are provided to subscribers as a metered service over a network
- Cloud services can be broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS)
- Cloud forensics is the application of digital forensic investigation processes in the cloud computing environment
- Crime committed with cloud as a subject, object, or tool is a cloud crime

- Forensic investigations in the cloud largely depend on the cloud deployment models selected by the customers that determine the responsibilities of different cloud components shared between the CSP and customer
- According to the NIST, cloud forensics challenges can be categorized into nine major groups, namely architecture, data collection, analysis, legal, training, anti-forensics, incident first responders, role management, and standards.
- The forensic acquisition of VMs and log file investigation can provide several relevant information about the security incidents that occur on popular cloud platforms such as AWS and Microsoft Azure

This page is intentionally left blank.