

CompTIA Security+

Lecture 08

SY0-701

Security Architecture

By
Dr. Khaled ELMenshawy

Lecture

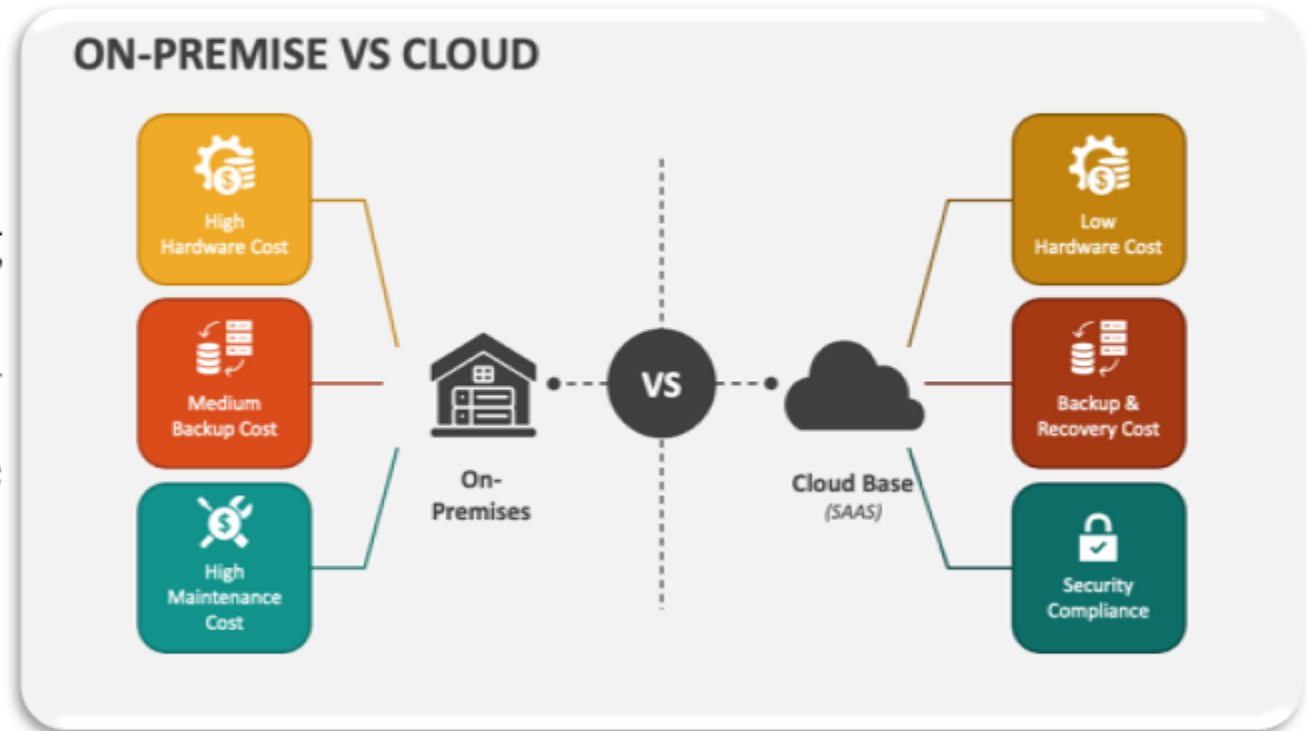
08

LEARNING OUTCOMES

- Differential between On-Premise and cloud computing.
- Understanding the cloud computing.
- Understanding Virtualization.
- Differential between Type I and Type II hypervisor.
- Understanding Containerization.
- Understanding cloud security.
- Understanding Serverless and Microservices.
- Understanding ports and protocols.
- Recognize different port numbers.

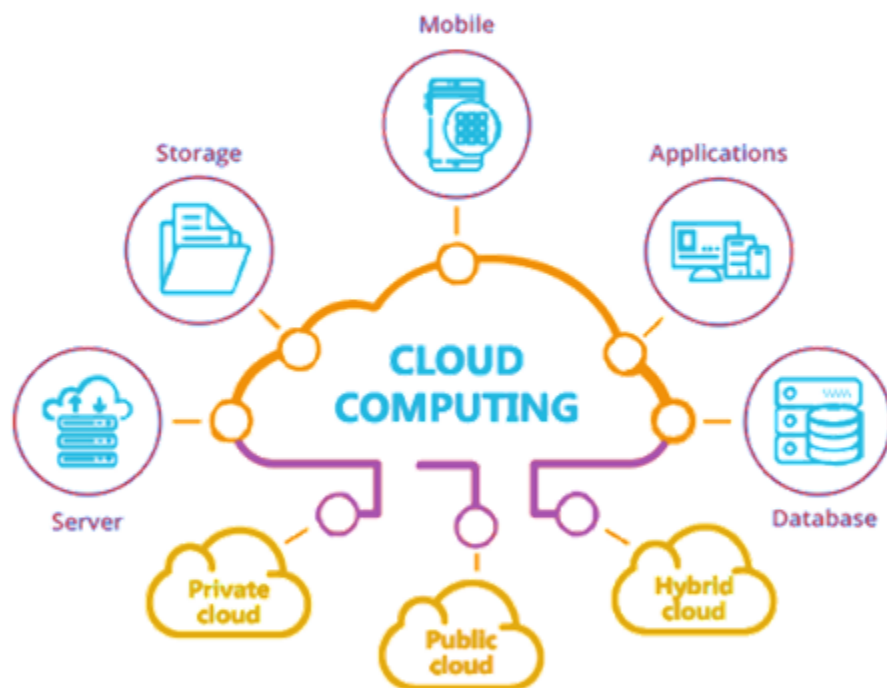
What is On-Premise Solutions?

On-Premise Solutions; is the computing infrastructure that installed and runs on computers that is physically located on site at a business (organization).



What is Cloud Computing?

Cloud Computing; is the *delivery of computing services including servers, storage, databases, networking, software, analytics, and intelligence over the Internet (the cloud)* to offer faster innovation, flexible resources, and economies of scale.



- **Responsibility matrix;** outlines the *division of responsibilities between the cloud service provider and the customer*. **For example;** in an infrastructure as a service (IaaS) model, the provider is responsible for managing the infrastructure while the customer handles everything else including the operating system, middleware, data and applications.
- **Third party vendors;** provide specialized services that *enhance the functionality, security, and efficiency of cloud solutions*. These services range from cloud management tools, security solutions to data analytic tools.

➤ **Hybrid solutions;** *combine on-premise infrastructure, private cloud services, and public cloud services.* They allow businesses to move workloads between environments depending on their needs.

When considering a hybrid solution, *businesses must consider data security, compliance, compatibility and cost.*

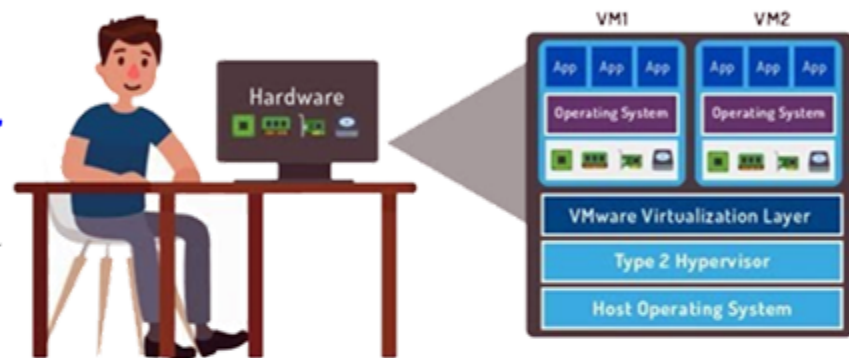
There are some keys that should be consider in cloud computing:

- *Availability*; is the system *ability to be accessed when needed*.
- *Resilience*; is the system *ability to recover from failures and continue to function* due to their distributed and redundant infrastructure.
- *Cost*; is *consider both the immediate and the long-term costs* of cloud adoption.
- *Responsiveness*; is *the speed at which the system can adapt to changes in demand*.
- *Scalability*; is the *system ability to handle increased workloads*.
- *Power*; The cloud service provider takes care of all the infrastructure, including the power supply.

- *Ease of deployment*; cloud services are generally *easier to deploy* than on-premise solutions. They don't require physical installation.
- *Risk transference*; some *risks are transferred to the provider*. However, the customer is still responsible for securing their data and applications.
- *Ease of recovery*; Cloud services often offer *easy data recovery and backup solutions*, protecting businesses from data loss.
- *Patch availability*; Cloud service providers *regularly release patches to fix vulnerabilities*.
- *Compute*; refers to *the number of computational resources that a customer can use*. This includes the number of CPUs, the amount of memory, and the type and size of storage.

What is Virtualization?

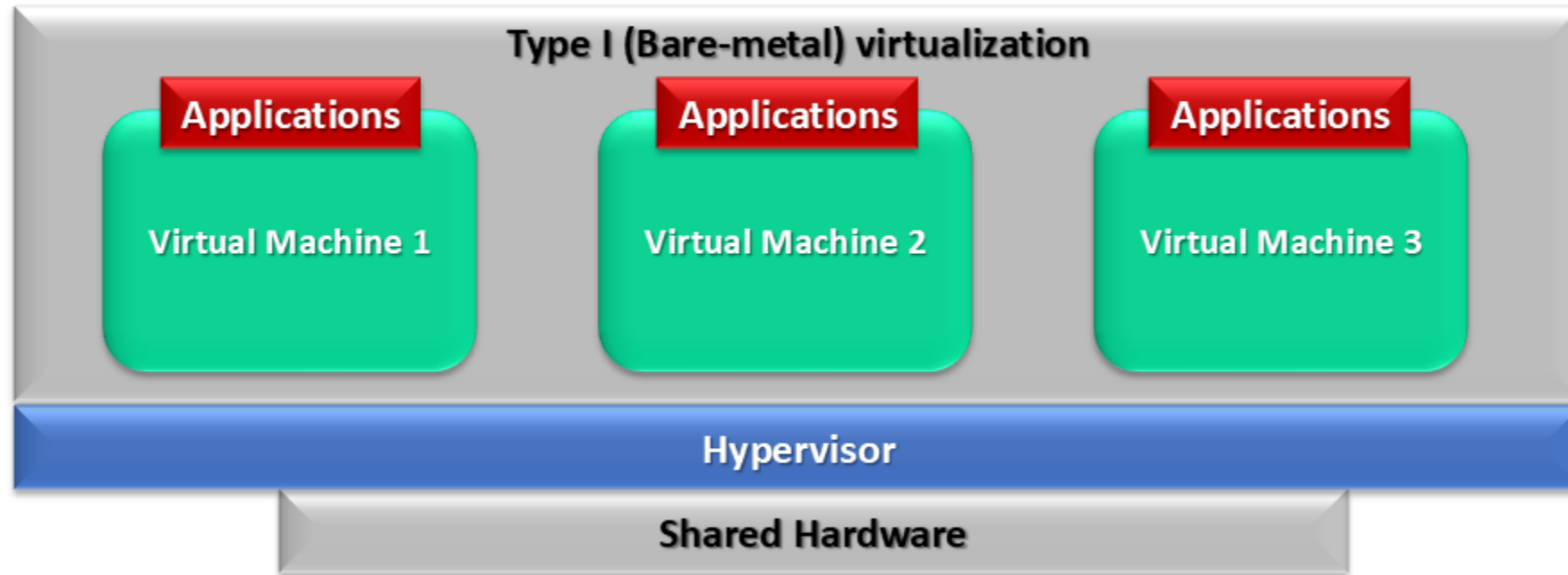
Virtualization; is the *simulation of the software and/or hardware* upon which other software runs. This simulated environment is called virtual machine (VM).



Running *multiple operating systems on the same hardware at the same time* and *Each VM can run its own operating systems and applications* as if it were in a physical machine.

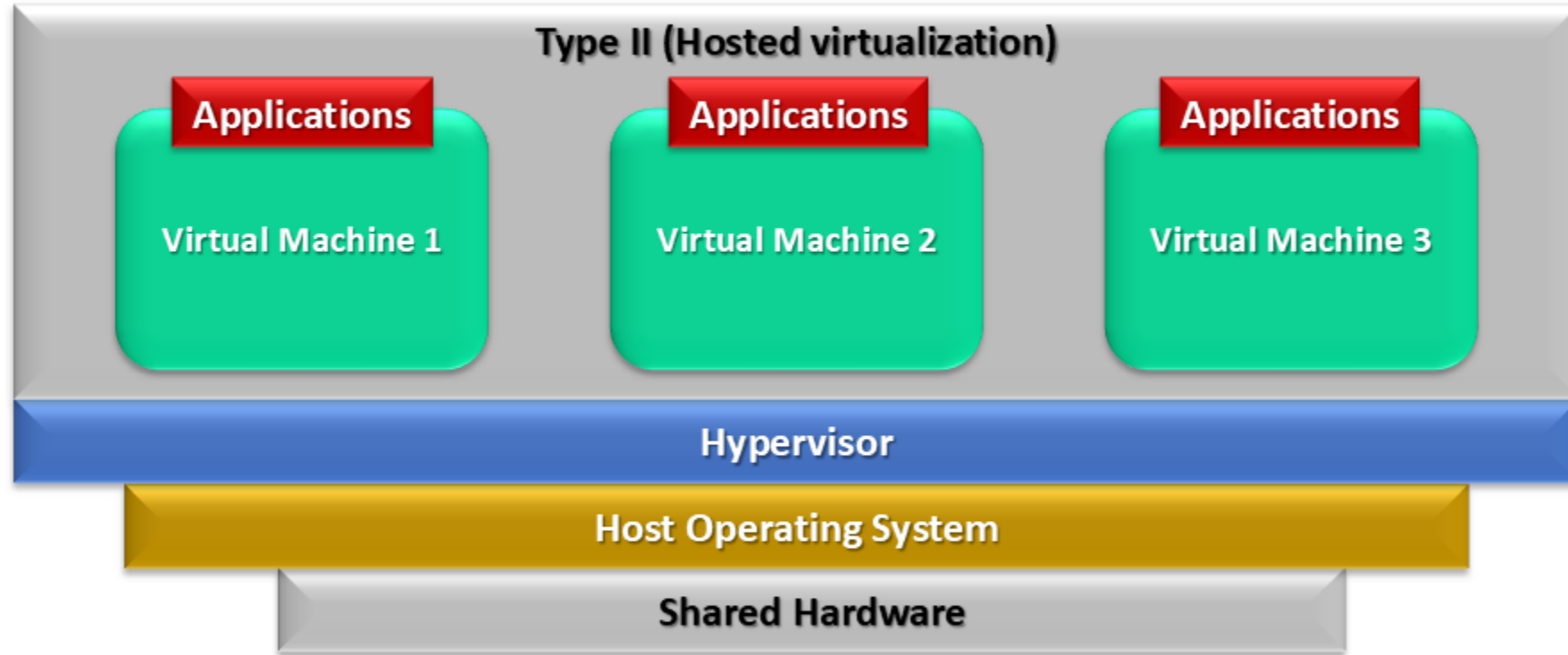
A piece of software called a **hypervisor** is installed to which *manages the distribution of the physical servers' resources* such as the processor, memory, and hard disk space among the virtual machines

Type I Hypervisor



Also known as a ***Bare-Metal*** or a *native hypervisor*, runs directly on the host hardware and functions similarly to an operating system. **Examples of** type 1 hypervisors include Microsoft Hyper-V, Citrix's XenServer and VMware ESXi and vSphere.

Type II Hypervisor

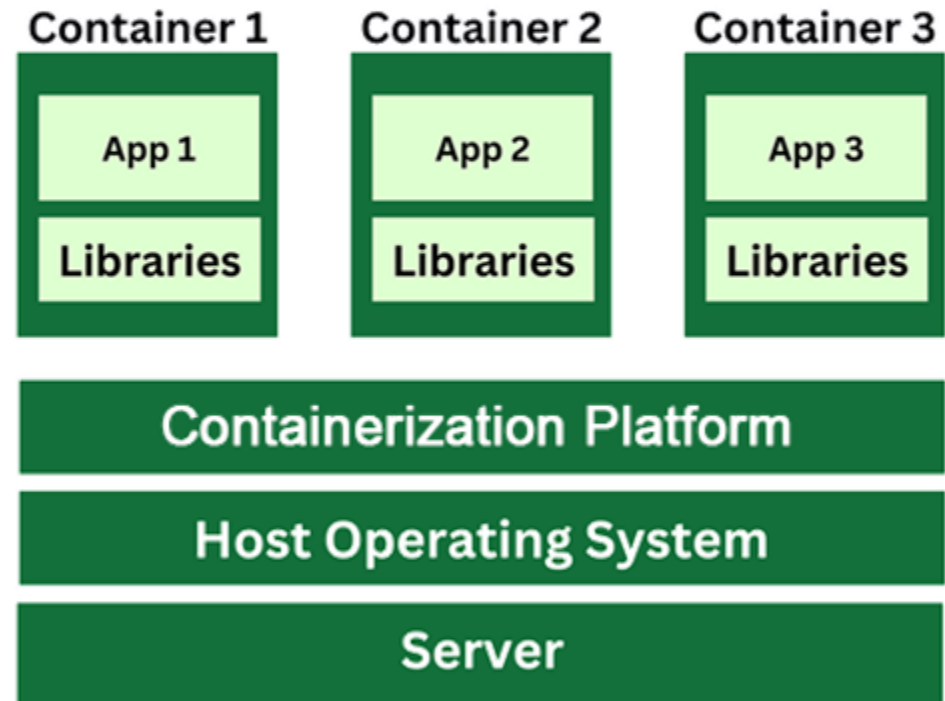


A *Hosted Architecture* which *operates within a standard operating system* such as Windows, Mac, or Linux. *Examples of* type 2 hypervisors include VirtualBox and VMware

Containerization

Containerization; is a software deployment process that *bundles an application's code with all the files and libraries it needs to run* on any infrastructure.

This approach offers several advantages, including *efficiency* and *speed*, *portability*, *scalability*, *isolation*, and *consistency*. Popular containerization technologies include Docker and Red Hat OpenShift.



There are some vulnerabilities that come to virtualization:

➤ Virtual machine escape; is breaking out one of isolated virtual machines and *interacting directly with the underlying hypervisor*, then attacker could *migrate themselves out and into another virtual machine* being hosted on the same physical server.

To mitigate VM escape:

- Keep virtual machine software patched.
- Install only the resource-sharing features that you really need.
- Keep software installations to a minimum because each program brings its own vulnerabilities.

- **Privilege elevation**; is occurring when a user is able to *grant themselves the ability to run functions as a higher-level user* such as the root or the administrator.

To mitigate Privilege elevation: Regularly patching and updating software, operating systems, and firmware.

- **Live migration of virtual machines**; is when a *virtual machine needs to move from one physical host to another*. If an attacker can place themselves between these two physical machines, they can implement a form of an *adversary-in-the-middle (AiTM)* attack where they can *capture the data being sent between two physical servers*.

To mitigate AiTM: use strong and unique passwords, enable MFA, and Use conditional access policies.

- **Resource reuse**; is a concept in computing where system resources, like memory, storage, or processing power can be reused for different tasks or processes. A resource reuse vulnerability occurs when *a system resource such as memory is not properly cleared or reset before being assigned to another task*. This can potentially *allow sensitive information from the previous task to be exposed to the new task*.

There are some vulnerabilities that come to Cloud Computing:

- *Shared physical server vulnerabilities*; multiple users share the same physical server. This *shared environment can lead to vulnerabilities* if one user's data is compromised, potentially affecting other users on the same server.

To mitigate Shared physical server vulnerabilities:

- Cloud providers should *implement strong isolation mechanisms* such as hypervisor protection.
- Regular vulnerability scanning and patching.
- Network segmentation can help isolate VMs from each other.

➤ *Inadequate virtual environment security*: Virtualization is a cornerstone of cloud computing. However, *inadequate security in the virtual environment* can lead to unauthorized access, data breaches, and other security incidents.

To mitigate Inadequate virtual environment security:

- Secure VM templates.
- Updating and patching VMs.
- Monitoring VMs for unusual activities.

➤ *User access management*; is *unauthorized access to sensitive data and systems*. This is often due to weak passwords, excessive permissions, or lack of user activity monitoring.

To mitigate User access management:

- Implementing strong access control policies.
- Using multi-factor authentication.
- Limiting user permissions based on the principle of least privilege.
- Monitoring user activities for any suspicious behavior.

- *Lack of up-to-date security measures*; Cloud environments are dynamic and constantly evolving making it crucial to *keep security measures up to date*.

To mitigate Lack of up-to-date security measures:

- Regularly updating security measures.
- Regularly reviewing and updating security policies.
- Staying informed about the latest threats and security best practices.

➤ *Single points of failure*: Cloud services often rely on specific resources or processes. *If these fail, it can lead to a complete system outage affecting all users.*

To mitigate Single points of failure:

- Implementing redundancy and failover procedures.
- Regularly testing failover procedures to ensure they work as expected.

➤ *Weak authentication and encryption*; can *allow unauthorized users to gain access to cloud systems*. While weak encryption can *leave data exposed during transmission or storage*.

To mitigate weak authentication and encryption:

- Using multi-factor authentication.
- Strong encryption algorithms.

➤ Unclear policies; refer to the *lack of clear guidelines or procedures for various security aspects* such as data handling, access control, incident response.

To mitigate unclear policies:

- Develop clear, comprehensive security policies that cover all aspects of cloud security.
- Regularly reviewed and updated security policies.
- Regularly training and awareness programs can also help ensure that everyone understands and follows the policies.

- **Data remnants**; refer to the *residual data left behind after deletion or erasure processes*. These data remnants can be a potential security risk as they can be recovered and exploited by malicious actors.

To mitigate Data remnants:

- Implement secure data deletion procedures.
- Managing backups securely.
- verifying that data has been completely removed after deletion.

What is Serverless?

Serverless; is a model where the *responsibility of managing servers, databases, and some application logic is shifted away from the developers.* In a serverless architecture these tasks are handled by cloud service providers.

One of the most popular examples of serverless computing is AWS Lambda, a service offered by Amazon Web Services. With Lambda, developers can run their code without provisioning or managing servers.



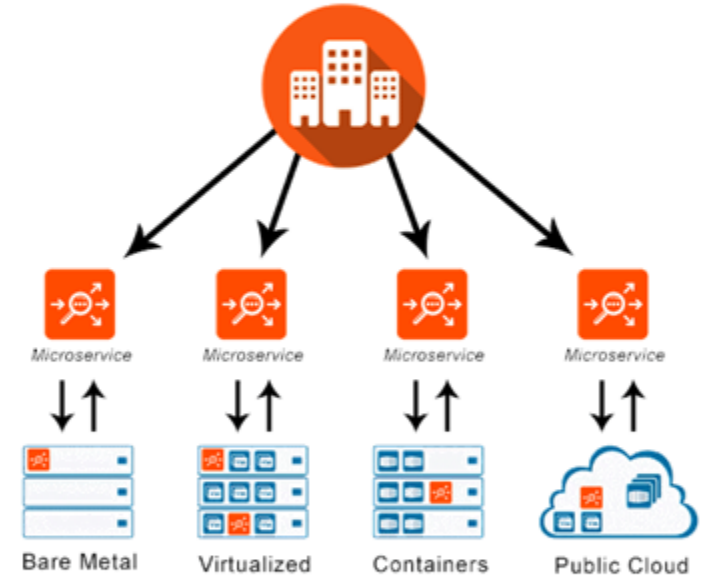
What is Microservices?

Microservices; are a type of software architecture where *large applications are broken down into smaller, independent services*. Each of these services runs a unique process and communicates through a well-defined lightweight mechanism to serve a business goal.

Monolithic
Architecture



Microservices Architecture



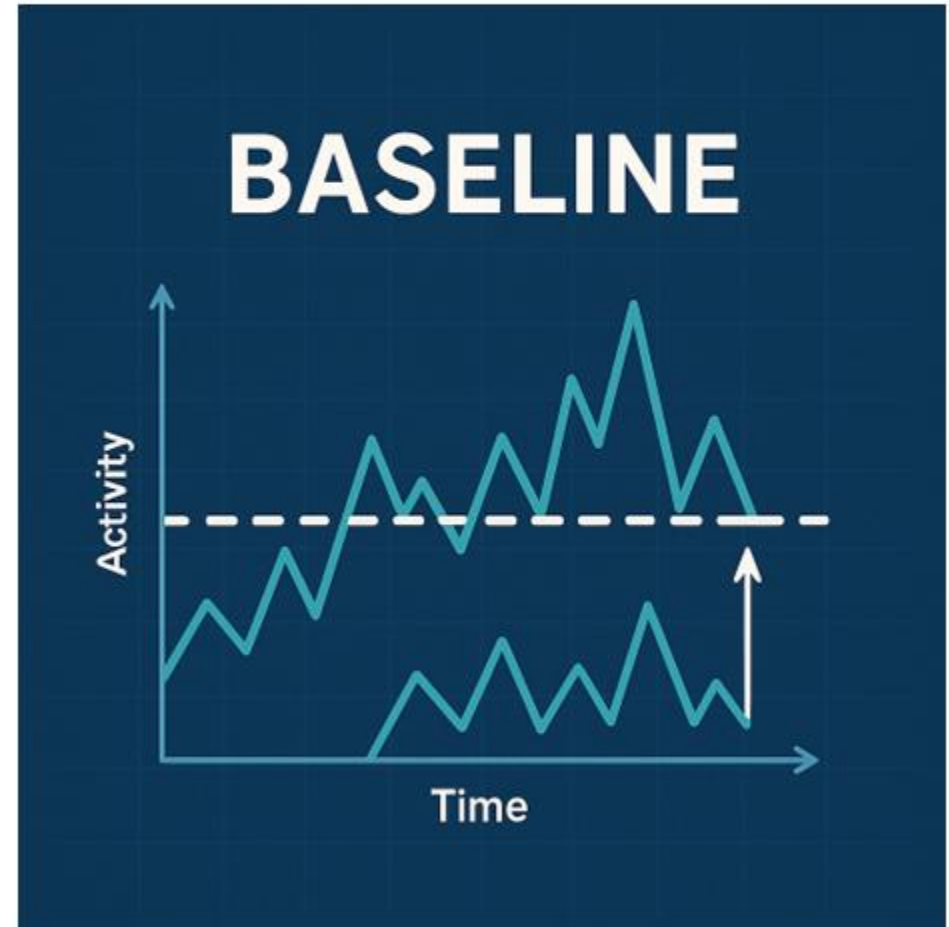
- **Monitoring;** involves the *continuous observation of a system or network* to detect anomalies, intrusions, and other potential security threats.
- **Automated monitoring;** involves *using software tools* to automatically scan and analyze network logs, system logs, network traffic, and other data for signs of malicious activity.
 - **Manual monitoring;** use *human personnel* to review and analyze the system data to detect potential security issues.

- **System monitoring**; involves tracking our system resources like CPU utilization, memory consumption, disc utilization and network performance.
- **Application monitoring**; involves managing and monitoring the performance and availability of the software applications.
- **Infrastructure monitoring**; involves observing the performance and availability of an organization's physical or virtual infrastructure.

➤ What is a Baseline?

A **baseline** refers to *a set of performance metrics and data points under normal operating conditions* of a computer system, network or application.

It's established by *observing and recording system performance and security metrics over time*, such as network traffic, user login frequency, CPU utilization, and memory usage.



➤ **How to Build a Baseline?**

- *Collect data*; Monitor metrics such as CPU load, traffic volume, authentication logs, Memory usage, etc.
- *Analyze normal behavior*; Observe patterns across days, weeks, or months.
- *Establish thresholds*; Define acceptable variance (e.g., $\pm 20\%$).
- *Implement in tools*; Configure alerts when deviations exceed defined limits.
- *Review and update regularly*.

- **Alerting;** refers to *the process of notifying* relevant personnel when a potential security incident occurs.
- **Log aggregation;** is used to *collect and consolidate log data from various sources into a centralized location.*
 - **Alerting;** is going to be used to *set up notifications to inform relevant stakeholders when specific events or conditions are occurring.*
 - **Scanning;** involves *conducting regular examinations of the systems, networks, or applications* to identify vulnerabilities, configuration issues, or other potential problems.

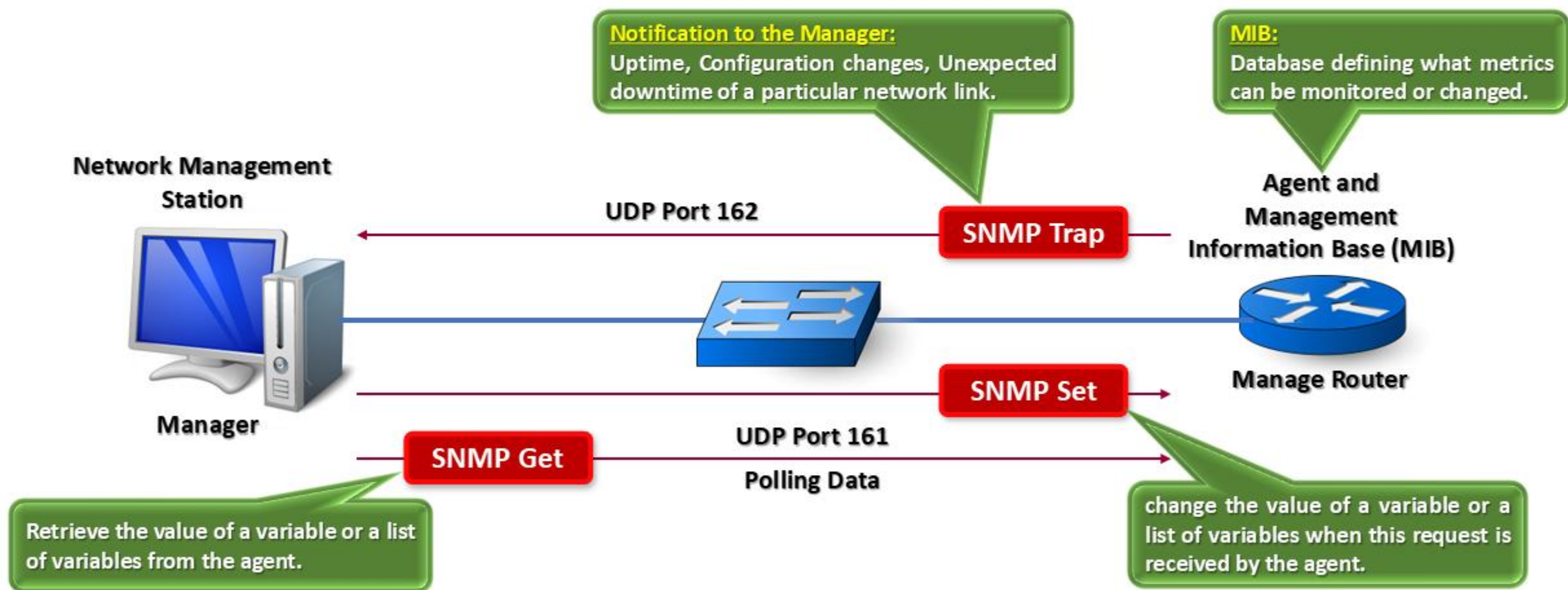
- Reporting; is going to be used to *generate summaries or detailed reports based on the collected and analyzed data.*
- Archiving; is used to *store data for long-term retention periods* and for future reference.
- Alert response ; involves *taking the appropriate actions* in response to received alerts.
- Remediation and validation; is going to be used to *take the steps necessary to resolve identified issues or vulnerabilities* and verify that your remediation efforts were successful.

➤ **What is a Simple Network Management Protocol (SNMP)?**

SNMP is a standard protocol *used for monitoring, managing, and controlling devices on IP networks*. It enables network administrators to gather information about network performance, detect problems, and configure devices remotely. It provides a real-time or near real-time analysis and operates at the application layer (Layer 7) of the OSI model

- **Monitoring**; Collecting real-time data (e.g., CPU usage, bandwidth, uptime).
- **Management**; Adjusting device settings or configurations remotely.
- **Alerting**; Sending automatic notifications (called traps) when problems occur.

Common SNMP Tools; SolarWinds Network Performance Monitor (NPM),
ManageEngine OpManager.



SNMP Architecture

➤ What is a Security Information and Events Management (SIEM)?

SIEM is a centralized cybersecurity system that *collects, aggregates, and analyzes log data from across an organization's IT infrastructure* to detect, investigate, and respond to potential security threats in real time.

Common SIEM Tools; IBM Qradar, Microsoft Sentinel, Splunk Enterprise Security.



✓ Collect

✓ Categorizing

✓ Correlation

- **Automation**; refers to the *process of automatically executing tasks without manual intervention*, typically achieved using specialized software or scripts.
- **Orchestration**; is a coordinated and sequenced *execution of multiple automated tasks* to achieve a specific outcome or workflow.
- **Security, Orchestration, Automation, and Response (SOAR)**; is a comprehensive cybersecurity framework and class of tools *designed to streamline, automate, and coordinate security operations across multiple systems and teams*. **SOAR** is primarily *used for incident response and threat hunting*.

➤ **Example of SOAR**; if *somebody clicked on a link in a phishing email*, **SOAR** framework performs the following steps:

- Go to the machine, isolate it from the network,
- Do a virus scan to make sure they haven't infected themselves or anyone.
- Check the registry to make sure there's nothing in there for persistence.
- Back up all the user data, refill them with a new computer.
- Reinstall the image and put the data back on it.

- Case scenario of automation; **Tool**; Splunk SOAR, **Case**; automatically isolate a compromised endpoint after an alert.

Automation Workflow (Playbook):

- Receive alert from SIEM.
- Query endpoint details via EDR API.
- Isolate host.
- Notify analyst via email.

```
# Cortex XSOAR automation snippet
incident = demisto.incidents()[0]
host_ip = incident['CustomFields']['sourceip']

# Call EDR API to isolate
response = demisto.executeCommand("crowdstrike-isolate-endpoint", {"ip": host_ip})

if response[0]['Type'] == 1:
    demisto.results(f"✅ Host {host_ip} successfully isolated.")
else:
    demisto.results(f"⚠️ Failed to isolate {host_ip}. Check EDR connectivity.")
```

Python Automation Script

Feature	SIEM	SOAR
Primary Purpose	Detect threats using log analysis	Respond to threats using automation
Main Function	Collect & correlate logs → generate alerts	Execute automated playbooks → remediate incidents
Data Source	Logs from servers, firewalls, apps, endpoints	Alerts from SIEM + context from external systems
Focus Area	Detection & visibility	Response & orchestration
Human Involvement	Higher — analysts review alerts	Lower — automated actions take place
Examples of Use	Find brute-force attacks, anomalies	Isolate host, delete malicious emails, block IP
Integrations	Ingests logs	Communicates with tools: SIEM, EDR, DNS, email, firewalls

