Applied Virtual Networks

COMP 4912

Instructor: **Dawood Sajjadi**

PhD, SMIEEE, CISSP

ssajjaditorshizi@bcit.ca

Winter-Spring 2025

Week #8

BCIT™

RECAP

The **Ansible** project is an open source community sponsored by Red Hat. It uses YAML to perfectly describe IT application environments in Ansible **Playbooks**. **Ansible Engine** is a supported product built from the **Ansible** community project.

IBM

Red Hat

What Ansible Does?

- ✓ Overview of Configuration Management and Automation
- ✓ Implementing an agentless architecture to manage resources

Key Features

- ✓ Simple, YAML-based Playbooks
- ✓ Agentless Approach
- ✓ Scalability and Performance
- ✓ Extensibility with Modules

ANSIBLE

BCIT™

How Ansible Works

RECAP

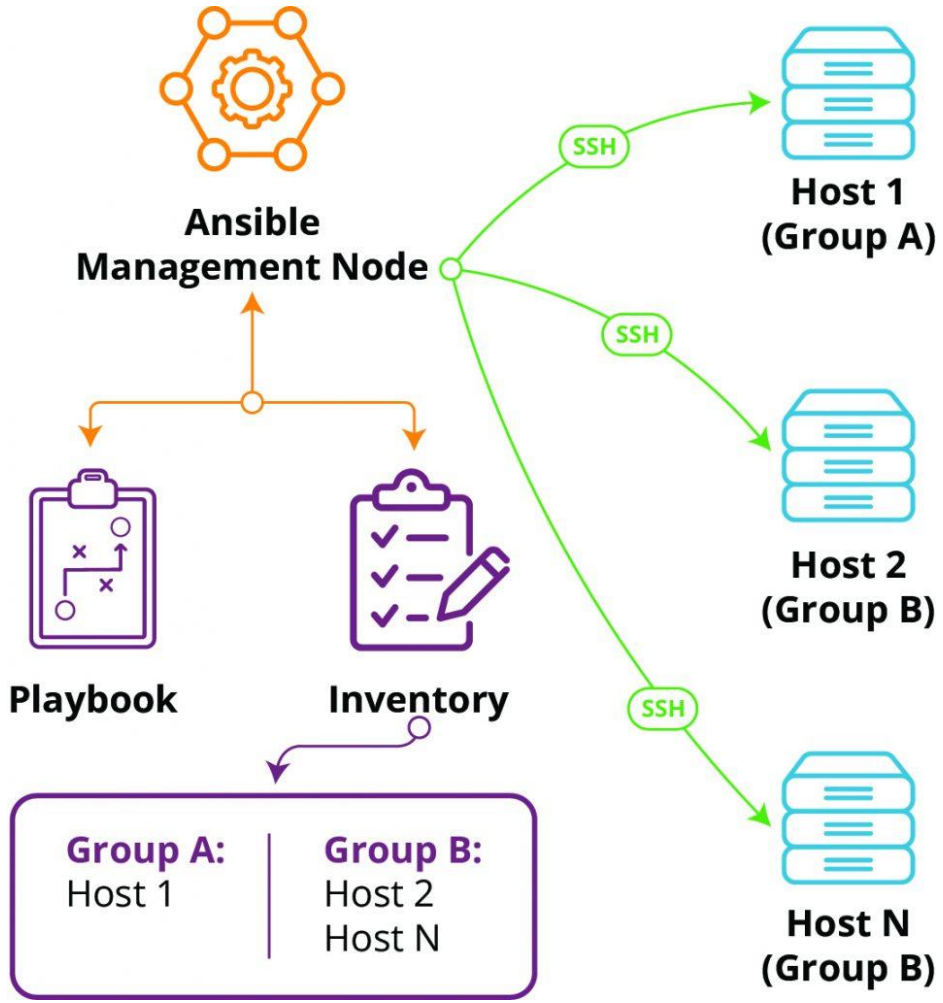
Execution Flow of an Ansible Playbook

→ Communication via SSH
→ Mostly written in Python
→ It is Agentless (no service running)

Common Use Cases

→ Configuration Management
→ Application Deployment
→ Orchestration of Complex Workflows

Architecture Overview



BCIT™

Terraform, the de facto standard for IaC

RECAP

Terraform is used to create, manage, and update infrastructure resources such as virtual networks, VMs, security rules, containers, domains and more. Almost any infrastructure type can be represented as a resource in Terraform.

- Declarative
- Written in Go
- Multiplatform
- Free Software & Open Source
- Freemium (Premium options: GUI, support, …)
- HashiCorp Product (https://developer.hashicorp.com/terraform/intro)

It is important to note that Terraform is primarily an IaC tool, designed to provision and manage infrastructure such as servers, networks, and storage, but **not to install or configure software on those servers**.

Terraform does NOT replace configuration management tools like Ansible.

BCIT™

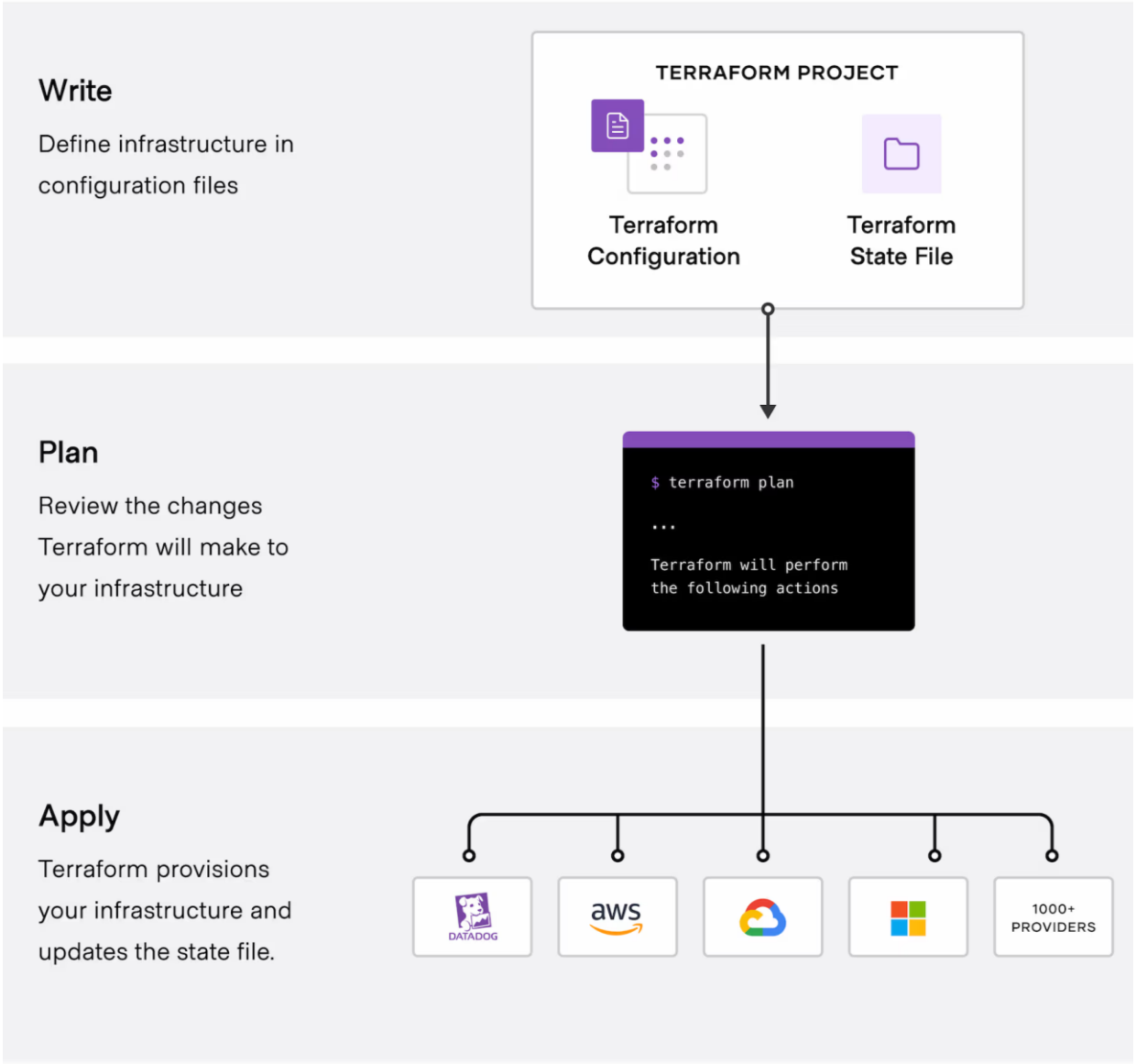
HashiCorp Terraform

How Terraform Works?

RECAP

- ✓ **Terraform init** initializes the (local) Terraform environment. Usually executed only once per session.
- ✓ **Terraform plan** compares the Terraform state with the as-is state in the cloud, build and display an execution plan. This does not change the deployment (read-only).
- ✓ **Terraform apply** executes the plan. This potentially changes the deployment.
- ✓ **Terraform destroy** deletes all resources that are governed by this specific terraform environment.

The core Terraform workflow consists of three stages:



BCIT™

Ansible + Terraform

Check This Video (Red Hat)

https://www.youtube.com/watch?v=vQSDWa8MlN8

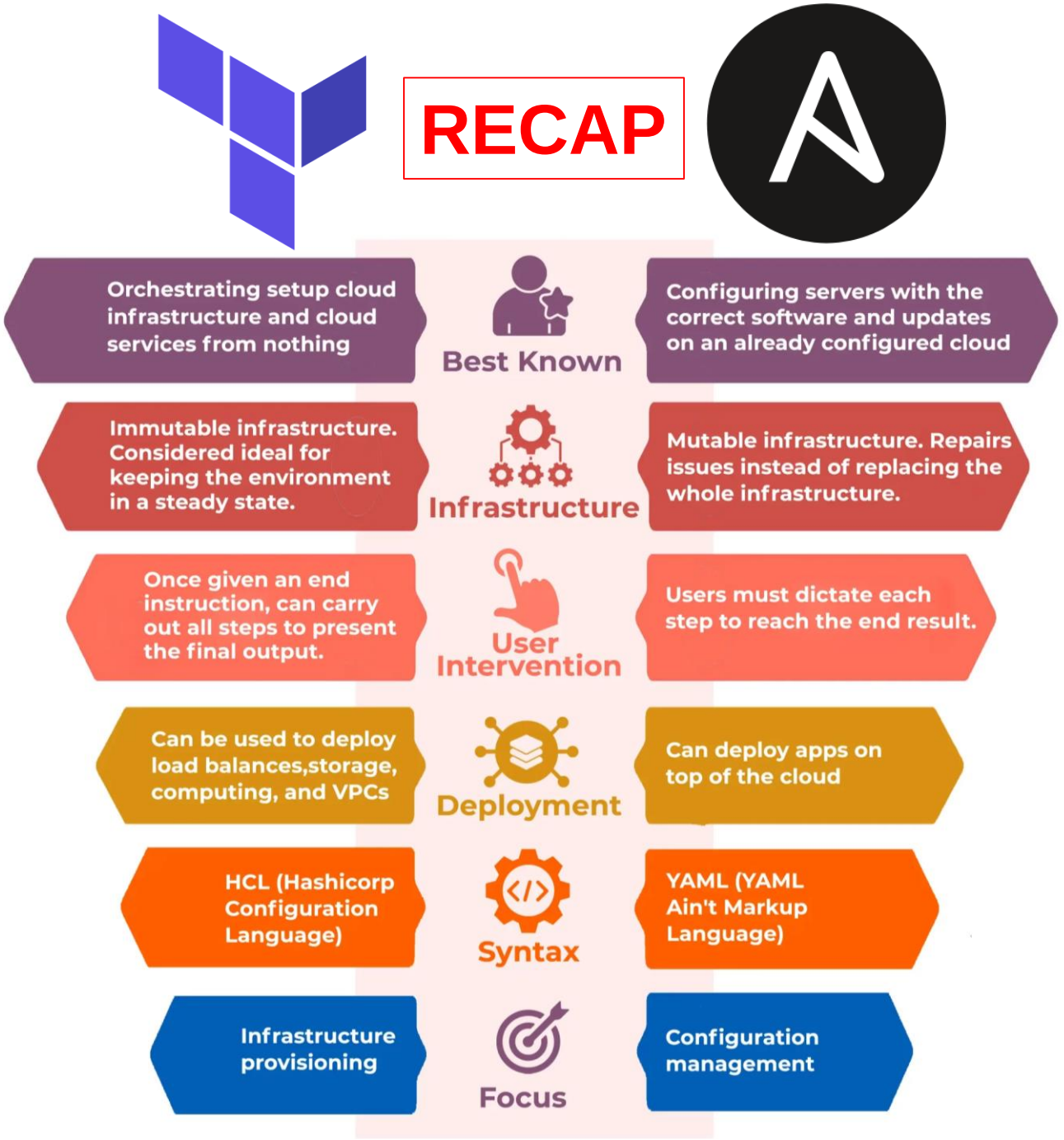
NetworkChuck (Ansible)

https://www.youtube.com/watch?v=5hycyr-8EKs

John Hammond (Terraform)

https://www.youtube.com/watch?v=a3vVUiLzm8w

BCIT™



Learning Outcomes of Week #8

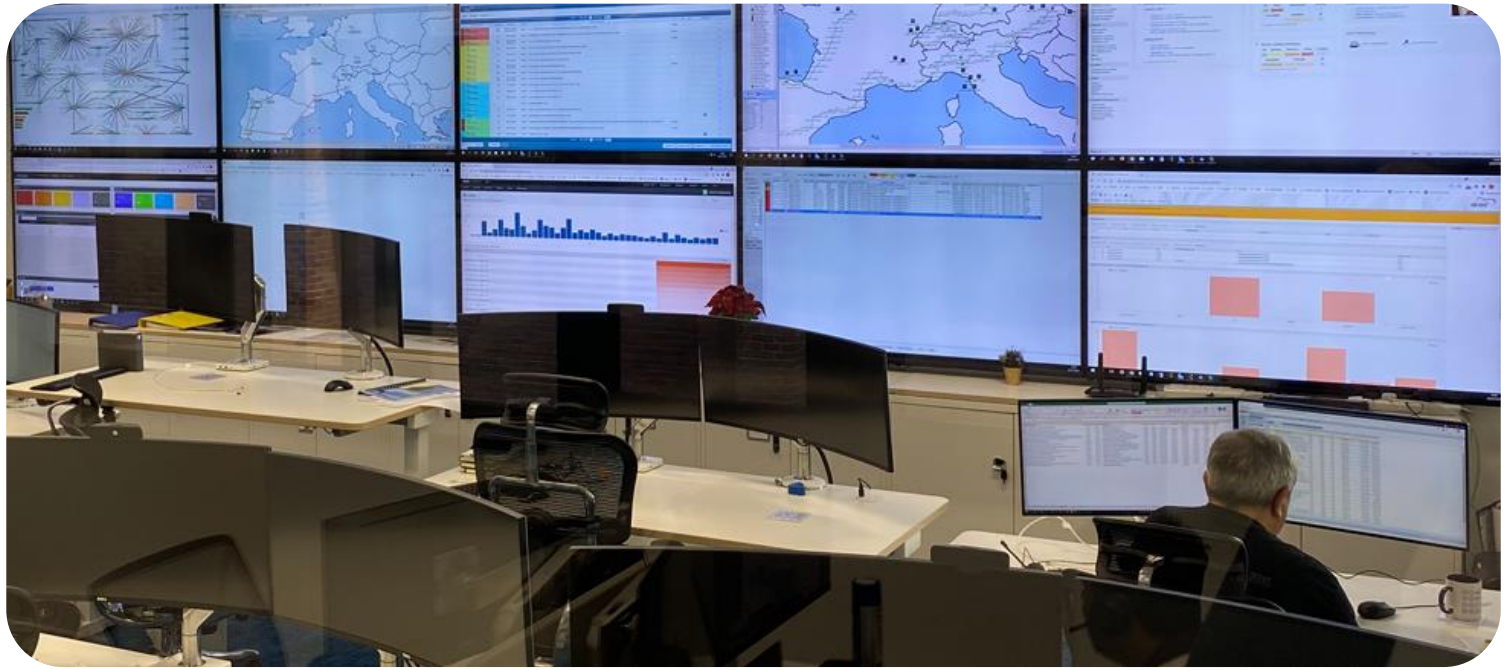
1. Understanding the key concepts of Service Monitoring.
2. Get familiar with Zabbix and How it can monitor different types of systems and service.
3. Explore Zabbix web-interface and monitor a Linux host using zabbix-agent.
4. Understanding SNMP protocol and its applications.
5. Gain hands-on experience on setting up an SNMP agent on a Linux host.
6. Describe Syslog as a system logging protocol and its components.
7. Learn how to setup an Rsyslog agent to send/receive logs from a Linux host.

BCIT™

Monitoring an IT Infrastructure

- **Ensure System Reliability** – Detect and prevent failures before they impact operations.
- **Performance Optimization** – Identify bottlenecks and optimize resource utilization.
- **Early Threat Detection** – Spot security vulnerabilities and respond to anomalies in real time.
- **Cost Efficiency** – Reduce downtime and optimize IT spending by identifying inefficiencies.
- **Minimize Downtime** – Proactive monitoring helps avoid costly service disruptions.
- **Compliance & Auditing** – Maintain logs and reports to meet regulatory requirements.
- **Capacity Planning** – Predict future growth and scale infrastructure accordingly.
- **Root Cause Analysis** – Quickly identify and troubleshoot system issues.

Network
Operation
Center



Security
Operation
Center

BCIT™

Metrics for Service Monitoring

There are several known metrics that help to define service **Reliability, Availability, and Disaster Recovery strategies.**

SLO (Service Level Objective)

- ✓ A **targeted performance goal** used internally by service teams.
- ✓ Defines desired service levels (Latency < 100ms, or Uptime > 99.95%).

SLA (Service Level Agreement)

- ✓ A **formal contract** between a service provider and a customer.
- ✓ Defines **minimum service guarantees** (e.g., 99.99% uptime).
- ✓ Includes **penalties** if the provider fails to meet commitments.

BCIT™

Metrics for Service Monitoring

There are several known metrics that help to define service **Reliability, Availability, and Disaster Recovery strategies.**

SLO (Service Level Objective)

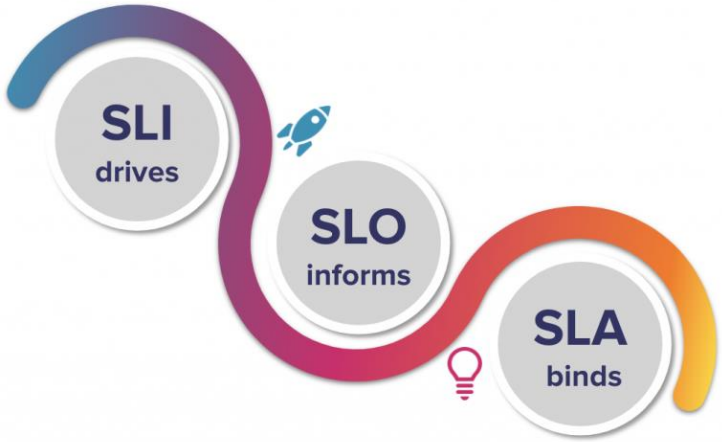
- ✓ A **targeted performance goal** used internally by service teams.
- ✓ Defines desired service levels (Latency < 100ms, or Uptime > 99.95%).

SLA (Service Level Agreement)

- ✓ A **formal contract** between a service provider and a customer.
- ✓ Defines **minimum service guarantees** (e.g., 99.99% uptime).
- ✓ Includes **penalties** if the provider fails to meet commitments.

SLI	SLO	SLA
service level **indicator**: a well-defined measure of 'successful enough'	service level **objective**: a top-line target for fraction of successful interactions	service level **agreement**: consequences
• *used to specify SLO/SLA*	• *specifies goals (SLI + goal)*	• *SLA = (SLO + margin) + consequences = SLI + goal + consequences*
• *Func(metric) < threshold*		

- ✓ **SLI:** current measured API latency is **160ms**
- ✓ **SLO:** **internal target latency < 150ms**
- ✓ **SLA:** guaranteed max latency to customer is **< 170ms**
  There will be a penalty & legal action for breach of SLA



BCIT™

Metrics for Service Monitoring

Four Golden Signals

Latency

Job processing time
Response generation time

Traffic

Length of message queue
Requests per second

Errors

Request failures
Response correctness

Saturation

CPU & memory utilization
Network bandwidth

Server Logs
Response duration
No network time

Synthetic Testing
External services
Pingdom/StatusCake

JavaScript Monitoring
Network load time
Browser rendering

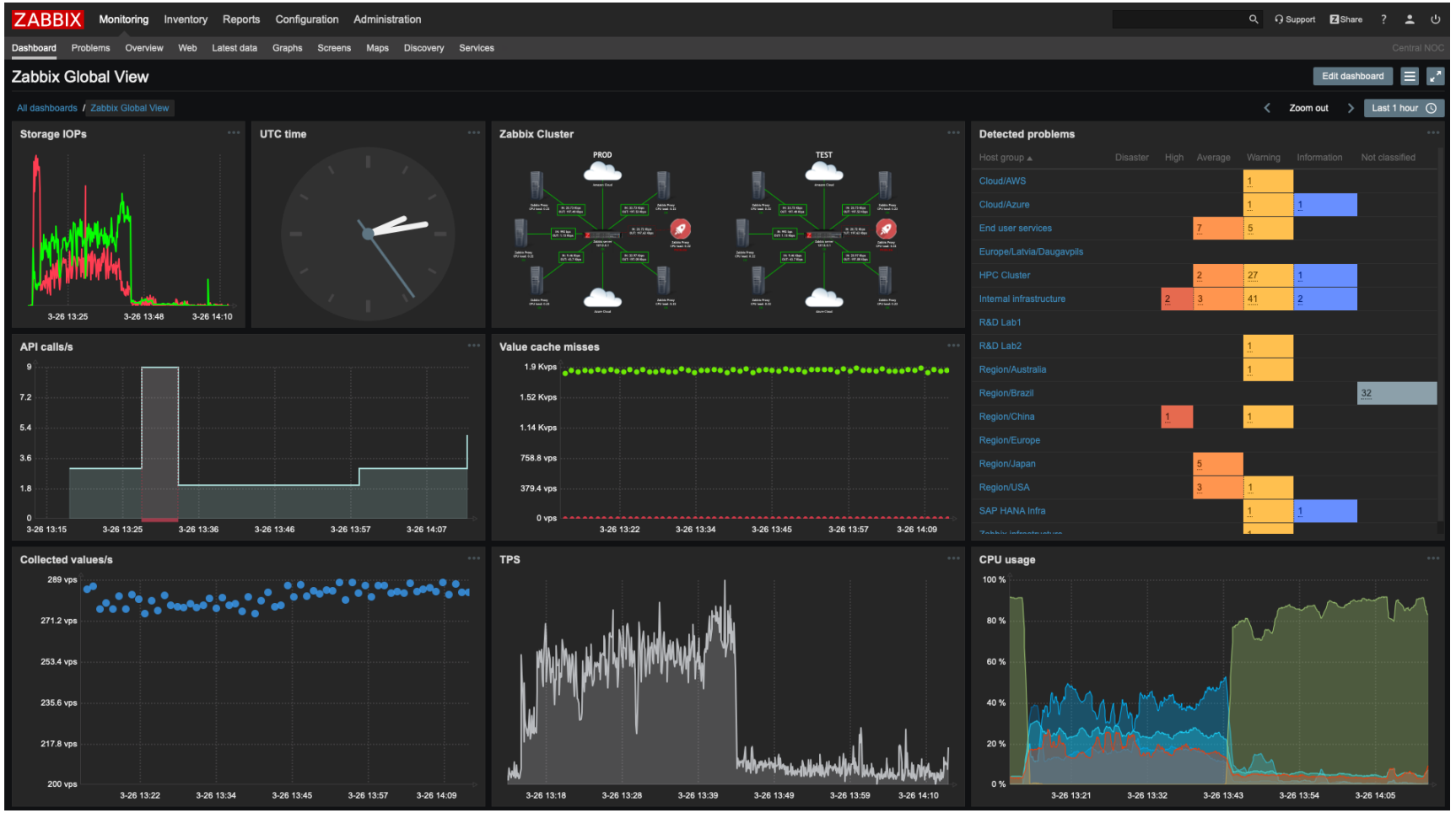
BCIT

Monitoring an IT Infrastructure

Zabbix is an open-source, enterprise-level monitoring solution that provides real-time insights into networks, servers, and applications.

https://www.zabbix.com

Let's visit this link to get a better understanding about Zabbix features ➔ https://www.zabbix.com/features



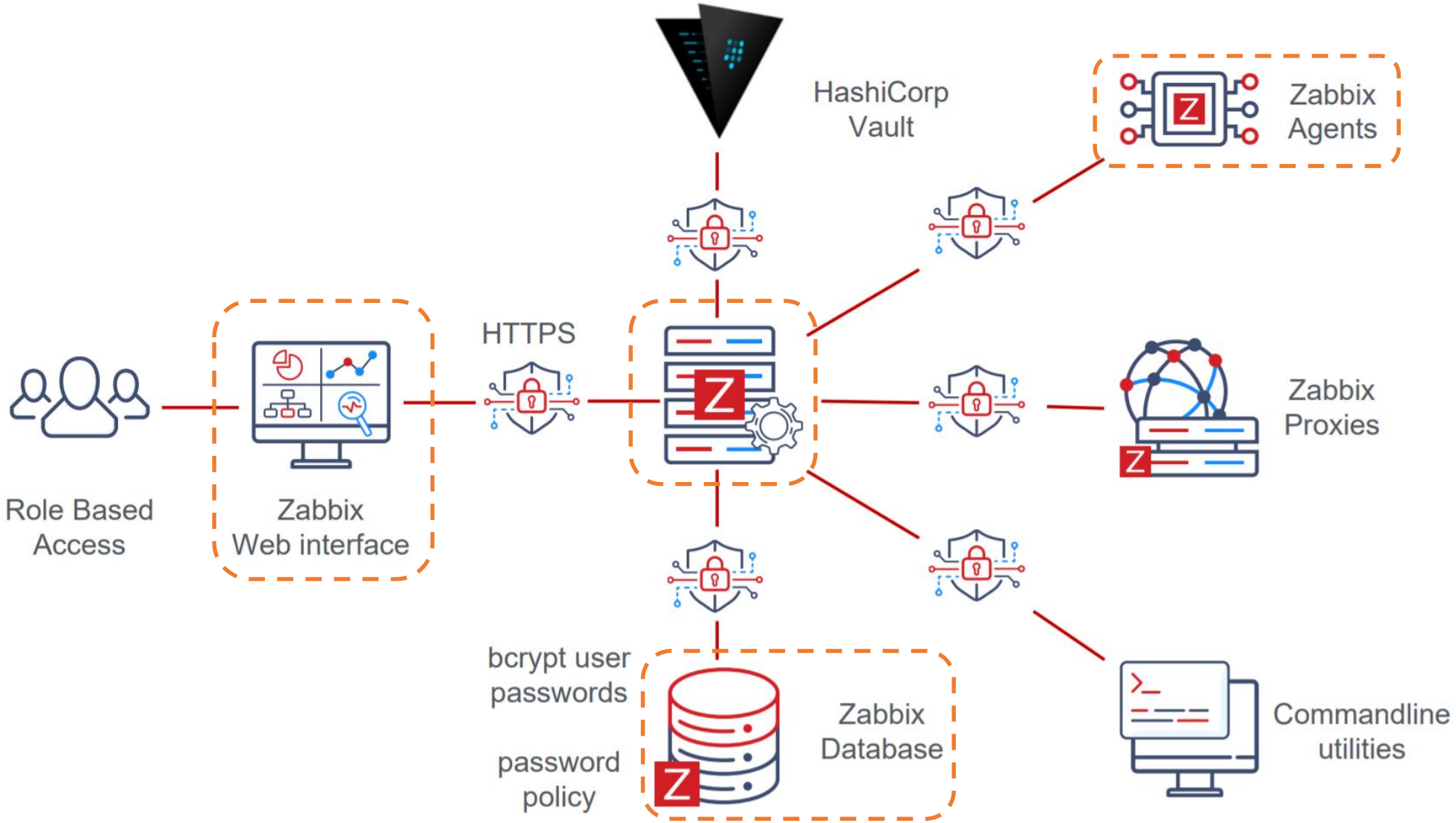
BCIT™

Monitoring an IT Infrastructure

ZABBIX

https://www.zabbix.com

Zabbix Structure



BCIT™

Create a Zabbix Container

1. Install **docker** and **docker compose** on your Linux VM.
   https://github.com/5tuxnet/courses/blob/main/comp4912/lab5/zabbix-mariadb/docker_install.sh

2. Download **docker-compose.yaml** file from GitHub and copy that under a folder named ~/zabbix in your home directory.
   https://github.com/5tuxnet/courses/blob/main/comp4912/lab5/zabbix-mariadb/docker-compose.yaml

3. Run '**docker-compose up –d**' inside ~/zabbix directory.

4. Verify docker containers are running via '**docker ps**' and ensure port 8080 TCP is in LISTENING state using '**netstat –nltup**'.

5. Run '**hostname –I**' command on your Linux VM to find the IP address which is accessible.

6. Open your browser and try access http://<IP address>:8080. To login the page, use **admin/zabbix** as credential.

7. Create an EC2 instance in AWS and install a Zabbix agent on it. Then, add this host into your Zabbix server to be monitored.

BCIT™

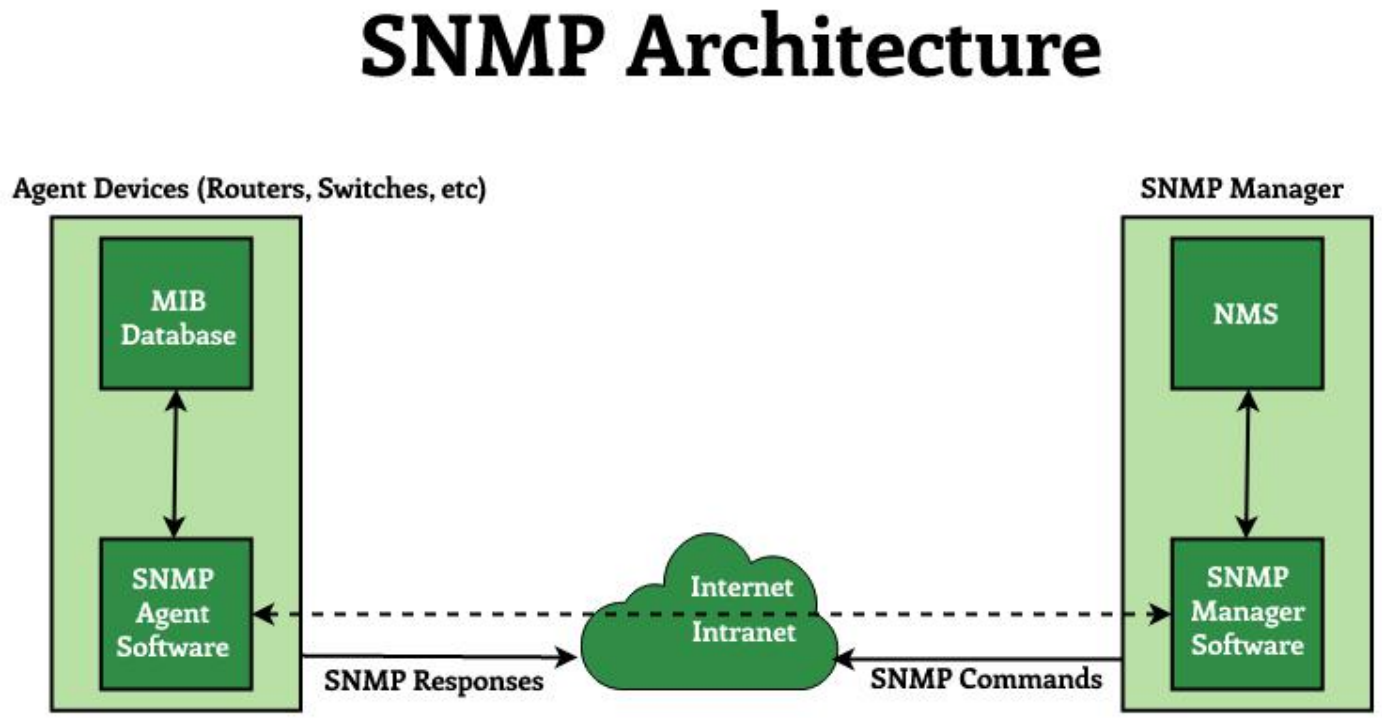
SNMP Protocol

SNMP (**S**imple **N**etwork **M**anagement **P**rotocol) is a standard that helps network devices communicate and share management data. It is commonly used to monitor and control network components.

- ✓ SNMP sends messages to network devices using SNMP agents.
- ✓ These messages are called SNMP Get-Requests.
- ✓ SNMP uses the User Datagram Protocol (UDP) to send messages.
- ✓ SNMP Management Information Bases (MIBs) define what can be collected from a device.

Why SNMP is important?

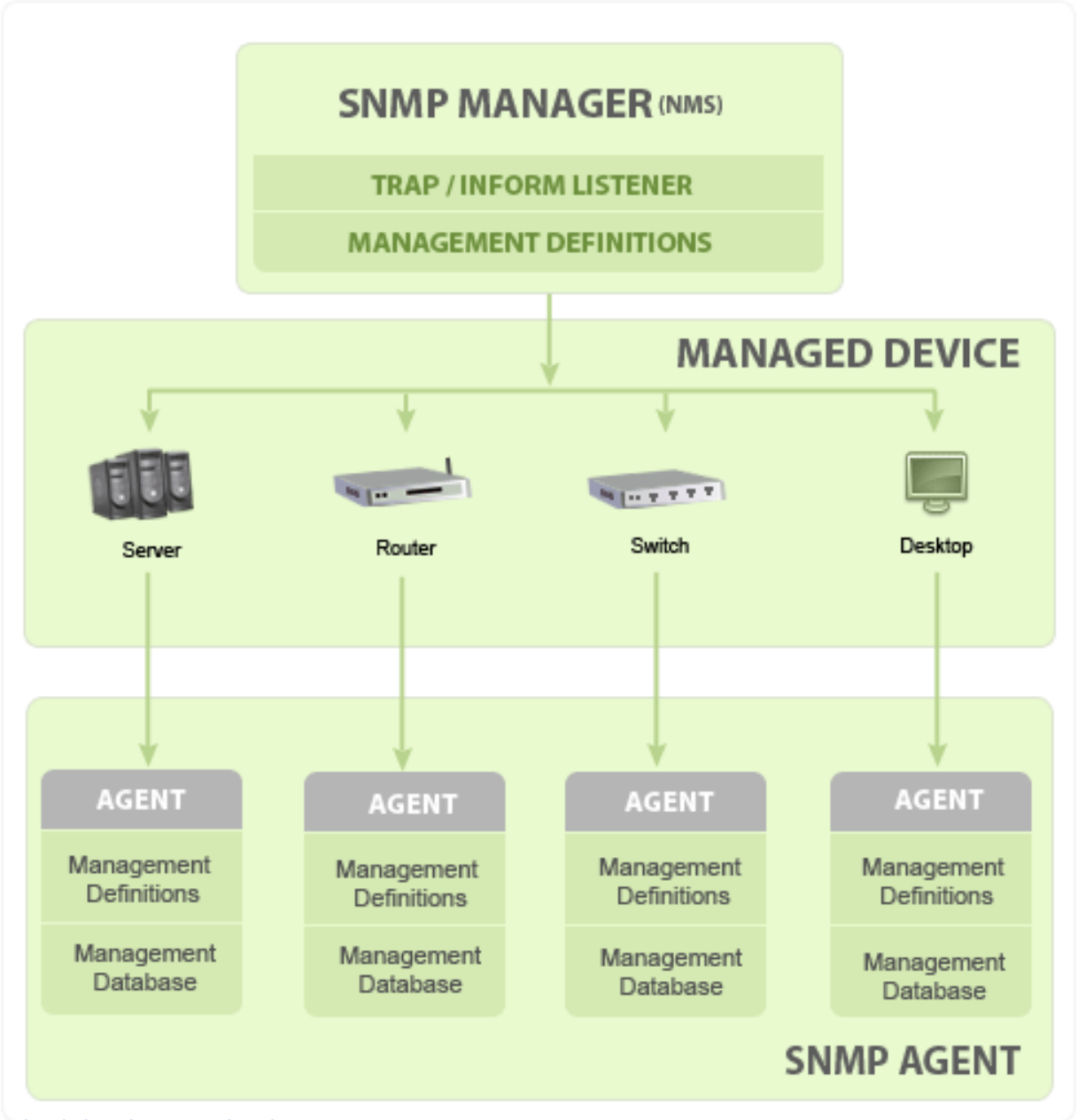
- ➢ SNMP helps to ensure a network is running with no issue.
- ➢ It allows network engineers to gather information about network equipment.
- ➢ It's compatible with most network devices, including routers, switches, servers, and firewalls.



- SNMP Agent
- SNMP Manager
- Managed Devices
- Management Information Base (MIB)

BCIT™

Basic SNMP Communication Diagram



https://www.manageengine.com/network-monitoring/what-is-snmp.html

SNMP Hierarchy

To provide flexibility and extensibility, SNMP doesn't require network devices to exchange data in a rigid format of fixed size. Instead, it uses a Tree-like format, under which data is always available for managers to collect

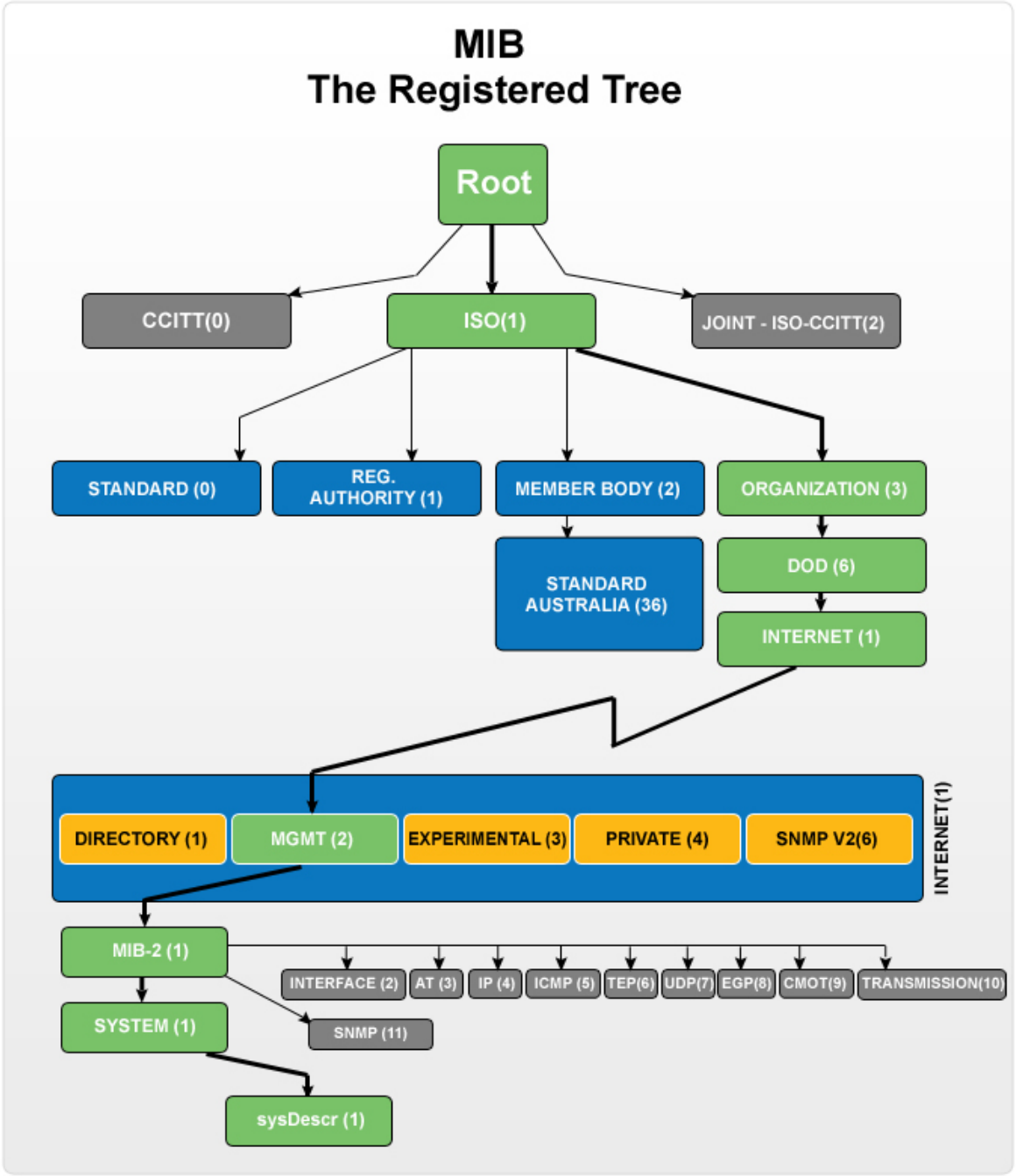
Multiple tables, referred to as **Management Information Bases** or **MIBs**, make up the data tree (or branches, if we stick with the tree metaphor). Each **MIB** groups together specific types of devices or device components. They have a unique identifying number and string, which can be used interchangeably, similar to how IP and hostnames used.

Each MIB consists of one or more nodes, which represent individual devices or device components on the network. In turn, each node has a unique **Object Identifier (OID).** The OID for a given node is determined by identifier of the MIB on which it exists combined with node's identifier within its MIB.

MIBs are a Logical Grouping for **OIDs**.

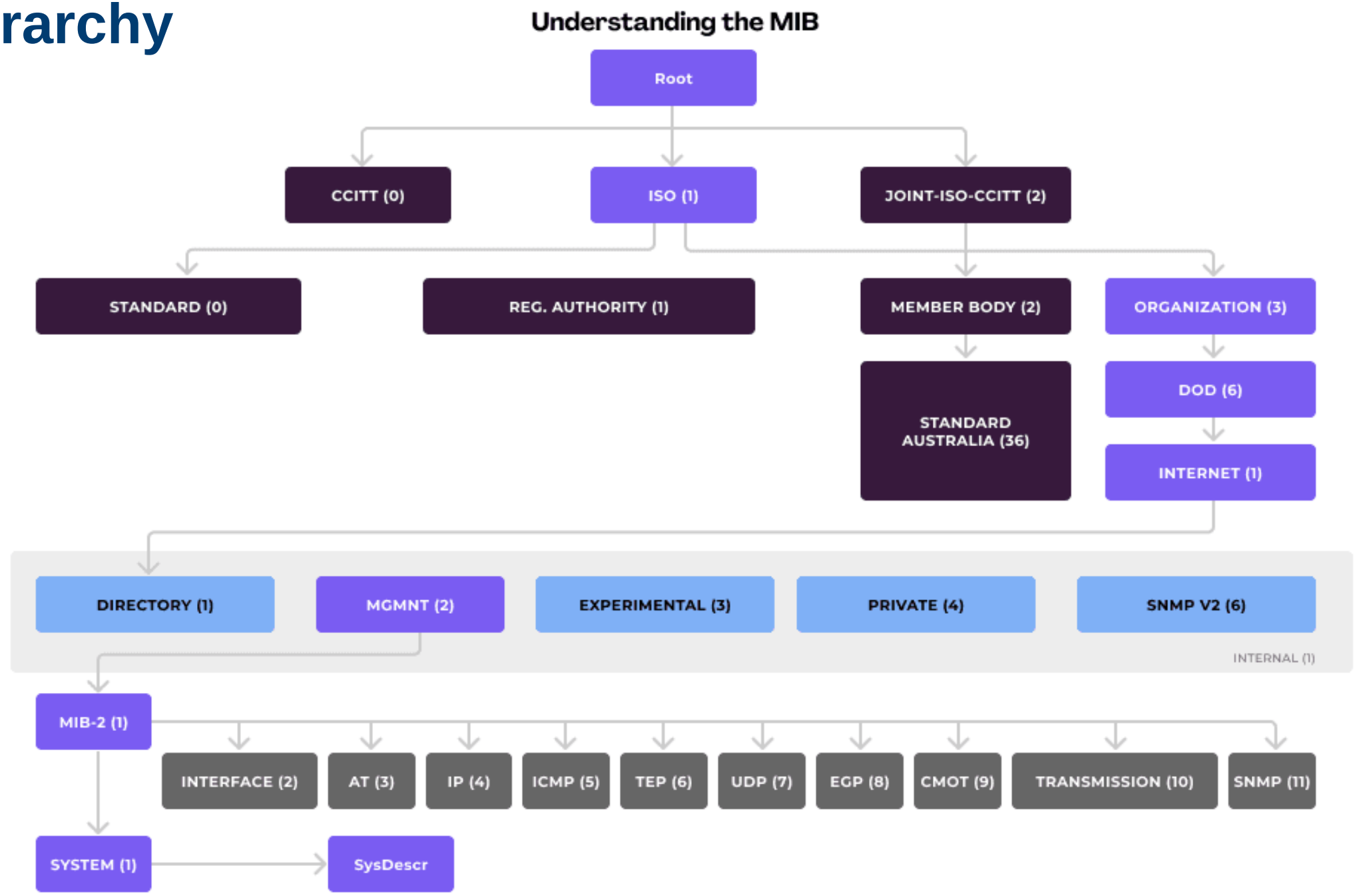
MIB: Hierarchical database of network objects
OID: Unique identifier for each managed object

1.3.6.1.2.1.1.1.0 → OID for System description



https://www.manageengine.com/network-monitoring/what-is-snmp.html

SNMP Hierarchy



https://www.auvik.com/franklyit/blog/network-basics-what-is-snmp/

SNMP Hierarchy

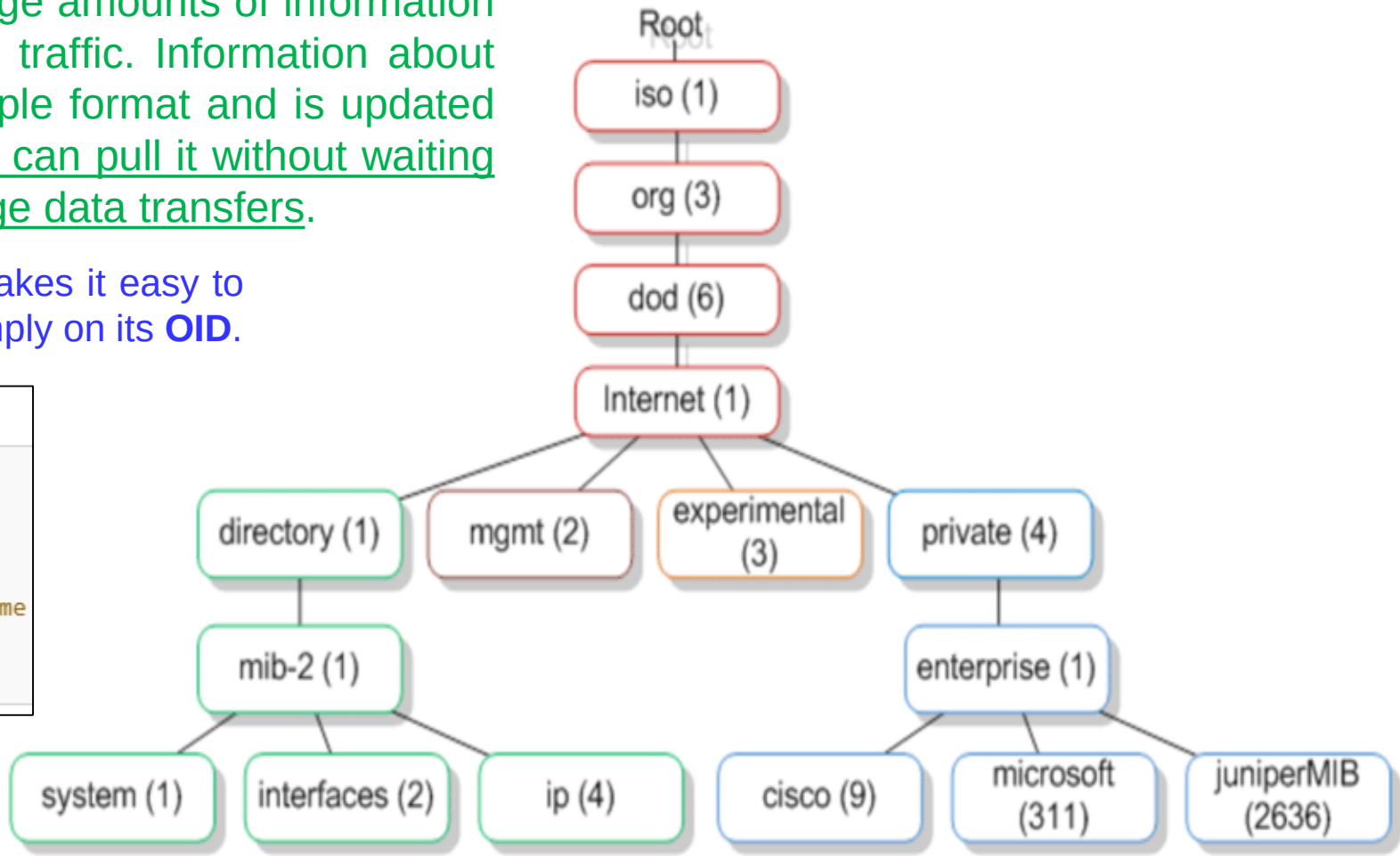
This hierarchy also provides an easy, flexible way to organize many devices across a network. It works no matter how large or small the network is, or what kind of devices are on it.

SNMP also makes it possible to collect large amounts of information quickly without clogging the network with traffic. Information about device status is always available in a simple format and is updated in real-time. This means SNMP managers can pull it without waiting for the data to be collected or requiring large data transfers.

Some OID values are vendor-specific, which makes it easy to gain some information about a device based simply on its **OID**.

CPU Usage Percentage from UCD-SNMP-MIB

```sql
.1.3.6.1.4.1.2021.11.9.0  (ssCpuUser) - User CPU time
.1.3.6.1.4.1.2021.11.10.0 (ssCpuSystem) - System CPU time
.1.3.6.1.4.1.2021.11.11.0 (ssCpuIdle) - Idle CPU time
```



OID Tree Example

BCIT™

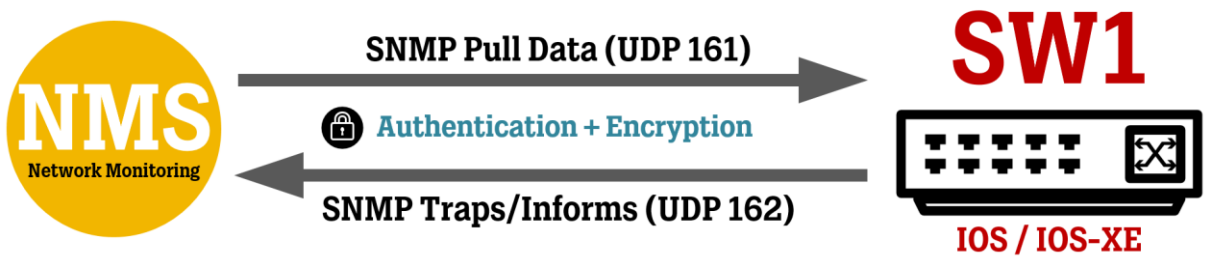
SNMP Versions

- SNMP comes in three versions: SNMPv1, SNMPv2c, and SNMPv3.
- SNMPv3 is the most Secure version.

SNMPv1: Basic functionality, no security.

SNMPv2c: Improved performance, still uses community strings.

SNMPv3: Security enhancements (Authentication & Encryption).

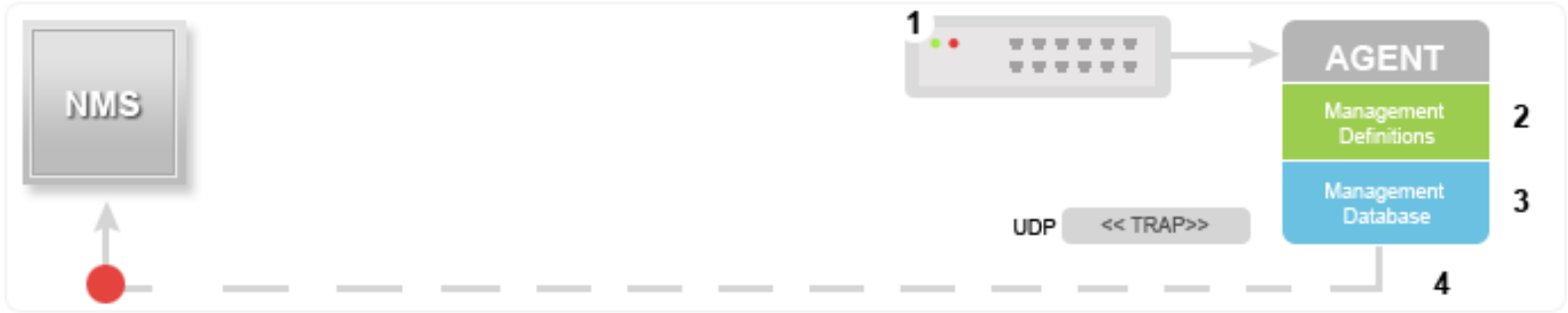


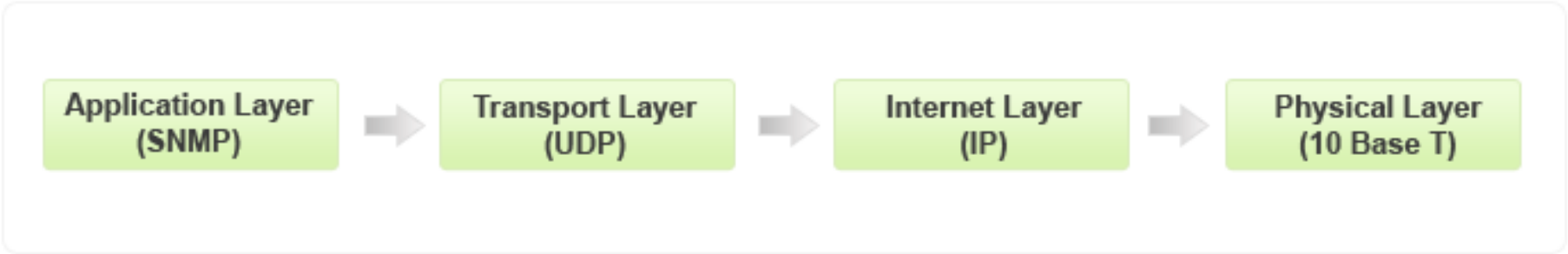
SNMPv1	SNMPv2c	SNMPv3
Easy to set up	Easy to set up	Hard to set up
Less Efficient	Less Efficient	More Efficient
Supports 32 bits counters	Supports 64 bits counters	Supports 64 bits counters with security
Plain-text community string	Improved error handlingand SET commands	Adds encryption and authentication
Packet Types: • Get-Request • Get-Next-Request • Set Request • Get Response	Packet Types: • Get-Request • Get-Bulk-Request • Get-Next-Request • Set Request • Inform-Response • SNMP v2 Trap	Basic functions are like v1 & v2 New packet types for SNMPv3

BCIT™

SNMP Trap

Event-driven Communication through SNMP Traps, providing real-time updates about network events to management systems. SNMP Traps are sent from Agent to Manager when an event occurs. Indeed, they are unsolicited messages and will be generated at events like **Link Failure**, **High CPU Usage**, **Interface Goes Down**.





BCIT™

Create a Zabbix Container

1. Create a new EC2 instance (using Ubuntu image) and install SNMP Agent on the host.

   $ sudo apt update
   $ sudo apt install snmpd libsnmp-dev
   $ sudo apt install net-tools
   $ netstat –nltup | grep 161 → Check 161 UDP is Listening
   $ sudo systemctl stop snmpd
   $ sudo net-snmp-create-v3-user -ro -A 'mypassword!' -X 'mypassword#' -a SHA -x AES snmpv3user
   $ sudo vim /etc/snmp/snmpd.conf
   => set value for agentAddress udp:161,udp6:[::1]:161
   $ sudo systemctl start snmpd
   $ sudo systemctl status snmpd

2. On your Linux VM (WSL), run '**curl ifconfig.me**' to get its Public IP address to access Internet.
3. Add this IP into the **Security Group** of your EC2 instance and **Allow port 161 UDP**.
4. From your Linux VM, run the following **snmpwalk** command.
   => $ snmpwalk -v3 -u snmpv3user -l authPriv -a SHA -A 'mypassword!' -x AES -X 'mypassword#' <EC2 instance public IP>
5. In your Zabbix server (running on Linux VM as a container), add the EC2 host using '**Linux by SNMP**' template.
6. Configure SNMPv3 for this host based on the user you created in Step 1.

BCIT™

When adding the **SNMPv3** details, don't set any value for **Context Name** (just **Security Name** is enough).
Check the items (OIDs) created from the 'Linux by SNMP' template in your Zabbix.

Syslog

- ✓ **Syslog** is a standard protocol for **Logging System Messages**.
- ✓ It allows **Applications** and **Systems** to send logs to a **Centralized Location**.
- ✓ Used for **Monitoring**, **Troubleshooting**, and **Security** analysis.

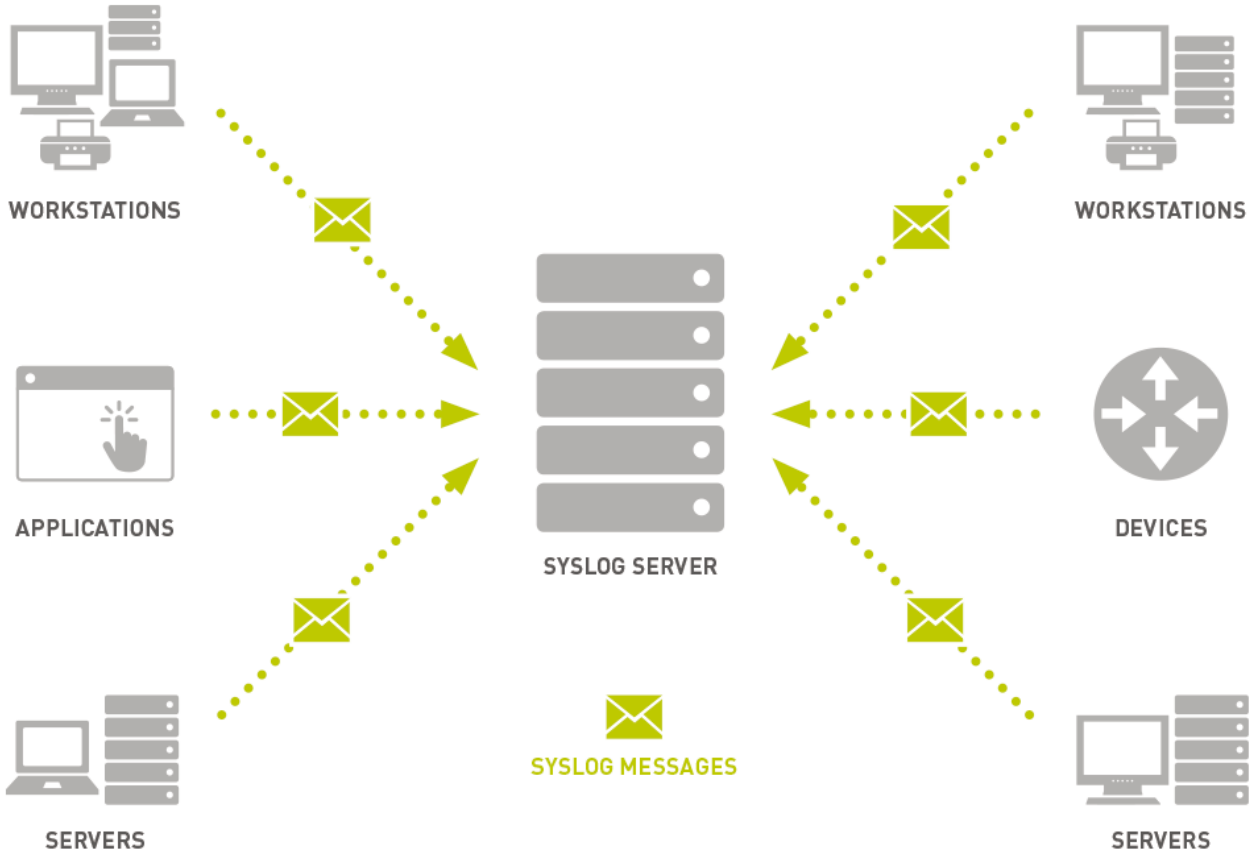
Components of Syslog

Log Generators – Applications/System processes that create logs.
Syslog Daemon - Manages log processing and forwarding .
Log Storage/Server - Collects and stores logs centrally for analysis.

Syslog Message Format

Priority (PRI) - Severity and facility level.
Timestamp - When the log was generated.
Hostname - The system that sent the log.
Application name - Source of the message.
Message Content - The actual log details.

<34> Mar 8 10:15:22 myserver sshd[1234]: Failed login attempt



BCIT™

Syslog

Syslog Severity Levels

Level	Code	Description
Emergency	0	System is unusable
Alert	1	Immediate action required
Critical	2	Critical conditions
Error	3	Error messages
Warning	4	Warning conditions
Notice	5	Normal but significant events
Informational	6	General information
Debug	7	Debugging messages

Benefits of using Syslog

- **Security Monitoring** - Detecting unauthorized access attempts.
- **Performance Monitoring** - Tracking system performance logs.
- **Network Management** - Monitoring router and firewall logs.
- **Compliance & Auditing** - Ensuring compliance with security policies.

BCIT™

https://www.rsyslog.com/doc/index.html
https://github.com/rsyslog/rsyslog

Syslog Facilities (indicates which process created the message)

Facility	Code	Description
kern	0	Kernel messages (hardware, drivers, etc.)
user	1	General user-level messages
mail	2	Mail system logs (Postfix, Sendmail, etc.)
daemon	3	System daemons (background services)
auth	4	Authentication logs (login attempts, SSH, etc.)
syslog	5	Messages from the syslog daemon itself
lpr	6	Printer-related logs
news	7	Usenet news system logs
uucp	8	Unix-to-Unix Copy protocol logs
cron	9	Scheduled job logs (crontab tasks)
authpriv	10	Private authentication messages
ftp	11	FTP server logs
ntp	12	Network Time Protocol (NTP) logs
audit	13	Security audit logs
alert	14	Logs related to critical alerts
clock	15	Clock daemon logs
local0 - local7	16-23	Custom facilities for applications

Create a simple Syslog server

1. In your last EC2 instance, check **rsyslog** is installed and running. If it is not then, run:

   $ sudo apt update && sudo apt install rsyslog

2. Open **/etc/rsyslog.conf** file and uncomment the following lines:

   ```
   module(load="imudp")
   input(type="imudp" port="514")
   ```

3. Next, you need to specify where the logs should be stored. In our example, it will be in a text file (/var/log/remote_logs.log).

   Create a file named **/etc/rsyslog.d/remote.conf** and add the following line into it.

   ```
   if $fromhost-ip != '127.0.0.1' then /var/log/remote_logs.log
   ```

4. Restart the rsyslog service by running '**systemctl restart rsyslog**'.
5. Ensure inbound traffic for **Port 514/UDP** is allowed for your EC2 instance (by adding it into the Security Group).
6. In your syslog client (Linux VM), open '**/etc/rsyslog.conf**' file and append the following line to the end of the file.

   . @<Server IP>:514 **=>** replace the <Server IP> with the Public IP address of your instance.

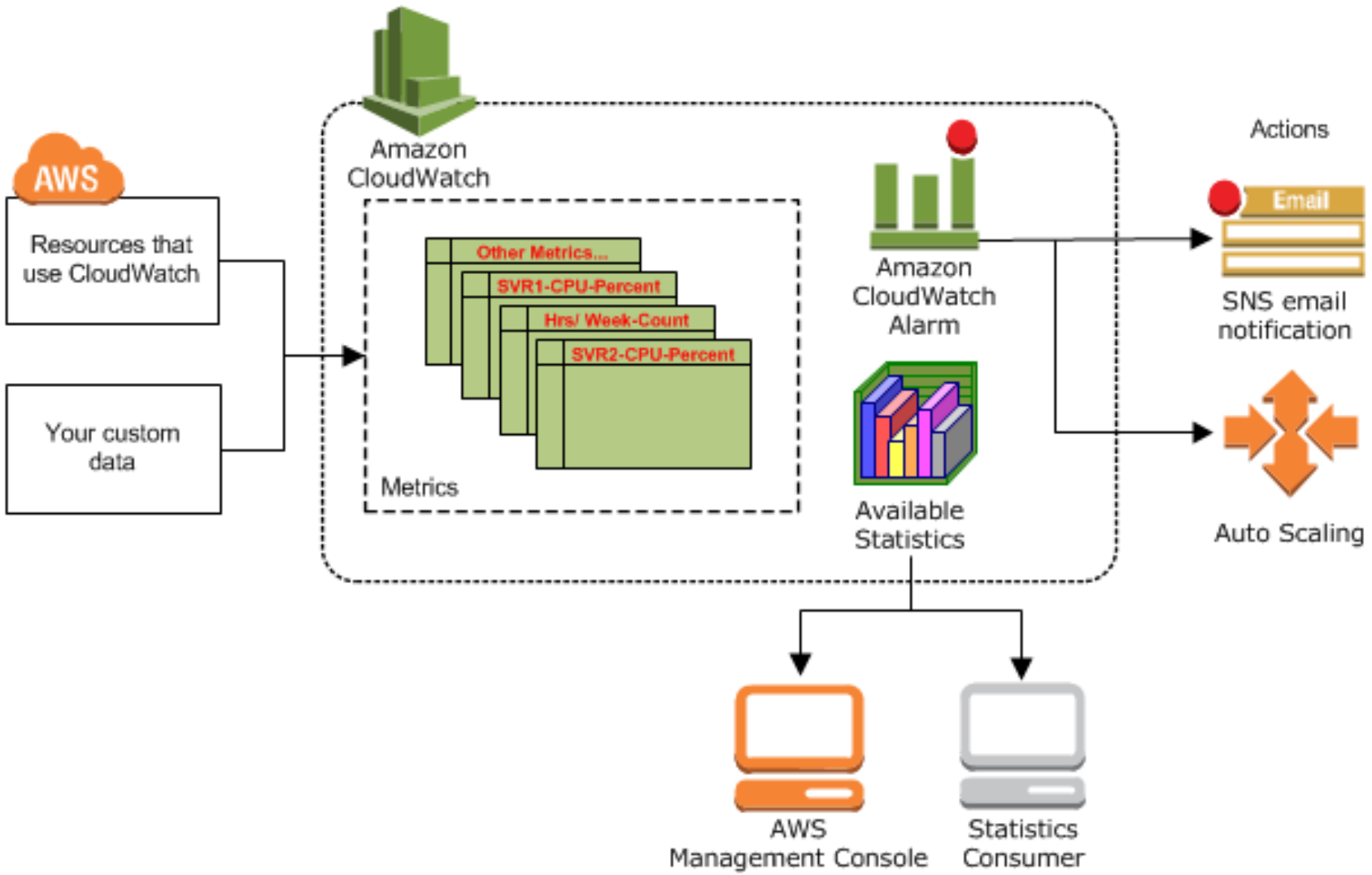
7. Restart rsyslog daemon on your syslog client and sent a test message by running following commands:

   $ sudo systemctl restart rsyslog && logger "Test message from Client…"

8. On your EC2 instance as rsyslog server, run '**tail –f /var/log/remote_logs.log**' to see incoming logs.
9. Try to SSH to your syslog client (Linux VM) from your own computer and enter wrong passwords!
10. You should be able to see reported '**Authentication Failures**' logs in **remote_logs.log** file at your syslog server.

BCIT™

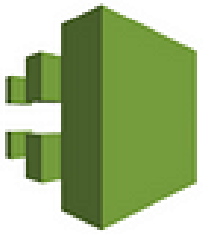
Amazon CloudWatch



https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

AWS CloudWatch/CloudTrail/Config

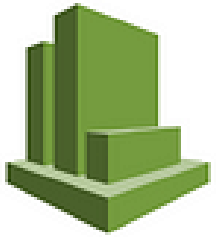
AWS CloudTrail



Auditing

Who made the change?
What changes were made?
Which location?

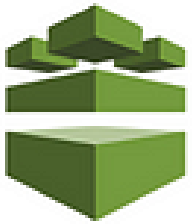
AWS CloudWatch



Monitoring

Collects and tracks metrics
Collects and Monitors the logs
Alarms and Notifcations

AWS Config



Compliance

What was changed?
Is the resource still compliant?
Maintain the change log

BCIT™

End of Lecture #8

THANK

Y☺U

- Dawood Sajjadi

BCIT™