MONITORING CLOUD SYSTEMS

INSIGHTS REQUIRED FOR INFRASTRUCTURE AS A SERVICE CLOUD ASSURANCE

© MOHAMED ELMESSEIRY

2020

Author Background

My Name is Mohamed ELMesseiry, Born in Alexandria, Egypt in 1982. I spent the last 15+ years of my career working with multiple information technology domains.

I was graduated from university in 2003 from an economics major. But, I had to follow my passion for technology.

I got my master degree in E-Commerce Applications Security, in 2007 from the University of Sunderland, United Kingdom. That was on the academic level.

On the commercial side, I gained many skills in multiple areas of technology, including Cloud Computing with OpenStack, Apache Cloud Stack and VMWare. Development using Python, Perl, NodeJS, C#, ASP.Net, PHP, Objective-C and Swift. Networking, Software Defined Wan (SD-Wan), Network Function Virtualization (NFV), Infrastructure Automation and Provisioning using Puppet, Ansible, Vagrant and lately artificial intelligence and machine learning.

Among all these technologies and fields of studies, there has been one constant that kept coming again and again; that is "service assurance". I worked as a monitoring consultant for almost 12 years, many times it was my full-time job, but also it was a business as usual in many cases.

I Started my career as a software developer, I was in love with developing web applications, back in the 2000's where Facebook, YouTube, twitter was not as famous as it is now.

Web applications shaped how businesses operates, there was also many challenges in developing highly available web applications. The tools, languages and even training was not as mature as of now. Through that Era, I witnessed the transformation from Structure Oriented Programming, Object Oriented Programming to Agile Development. And every time there is a change on how applications are built there are also requirements from the infrastructure to cope and adapt. During that time, I developed and designed more than 100 web applications, content management systems and web plugins.

In 2007, I started to ask so many questions about how other systems works. The "hacker" inside raised again, and decided to be a system integrator. I said to myself "there are so many unknowns" and in fact, if I could integrate multiple systems I would get introduced to many technologies, and I really wanted to do that.

I started by integrating security solutions, telecom applications, OSS, BSS, Charging systems, Core Banking Applications, Online Banking Applications, ATM Systems, and more. That was my golden age; where had to explore a new system every single day. I did my fair share of travels as well. Along the line Monitoring became a full-time engagement. Every customer had an SLA and all of them wanted to reach the maximum.

While implementing monitoring solutions I discovered a lot but also learned a lot. In Monitoring there are no best practices, and there is no single solution that will offer for your end to end monitoring, and that's why I had a job for all those years.

In 2014, I started to focus on designing and implementing cloud solutions for customers around the globe, as part of my role I worked with solutions based on VMWare, Redhat OpenStack, Mirantis OpenStack, Apache Cloud Stack.

In 2017, I worked as a Senior Advisory Architect/Consultant for Dell EMC, where I lead the consultancy OpenStack Practice, prior to that I used to lead the consultancy telemetry practice. I lead Dell EMC Service Assurance Tiger Team and designed many of Dell EMC Solutions for monitoring specially for Service Providers and Telecom.

Currently, I live in Canada, I'm a consultant product manager for Dell Technologies, where am help in bring new solutions to many customers around the globe.

Disclaimer

The ideas and information presented in this document is my own interpretation and opinion, it doesn't represent my employers view on the market or how to do things.

Special Thanks & Dedication

As God said in the **Quran**

Chapter (9) sūrat l-tawbah (The Repentance)

وَقُلِ ٱعۡمَلُواْ فَسَيَرَى ٱللَّهُ عَمَلَكُمۡ وَرَسُولُهُۥ وَٱلۡمُؤۡمِنُونَۖ وَسَتُرَدُّونَ إِلَىٰ عَٰلِمِ ٱلۡغَيۡبِ وَٱلشَّهَٰدَةِ فَيُنَبِّئُكُم بِمَا كُنتُمۡ تَعۡمَلُونَ ١٠٥

[9.105] And say: Work; so Allah will see your work and (so will) His Apostle and the believers; and you shall be brought back to the Knower of the unseen and the seen, then He will inform you of what you did.

I pray that this piece for work would reach who ever need it. giving back to the community whom I have learned a lot from all those years.

I would like to dedicate this effort to my wife; who has always supported me and pushed me to reach new heights. Also, to my son Malek, and my daughters Maira and Maria, to whom I foresee the bright future through their eyes.

I would like to dedicate it to my family, and to everyone who mentored or helped me along my interesting journey as a software engineer.

Thank You,
Mohamed ELMesseiry

1 Background and strategic fit

So, what is Monitoring?

Let's start by the definition of the term. In the English language; monitor is a verb that was derived from the Latin word (monit = warned) it means "*observe and check the progress or quality of something, and keep under systematic review*". Other synonyms are Observer, Watch, keep eye on. Track or supervise.

Each industry would then apply the same definition to its line of business. In the IT & Network operations industry; engineers would need to monitor IT Systems, Cloud Environments, and even application code. each monitoring domain will have a set of measurements and characteristics of what to monitor, how, and set expectation of the result of those measurements.

In Summary we can define monitoring IT systems as:

"Observing operation systems behaviors, workflow and processes overtime, to assure quality and enhance end services"

If the monitoring solution is not helping to sustain the business, then it has failed to achieve its goal.

While monitoring solutions include a great deal of reporting. Reporting on its own is only one step along the way to assurance. It's a view on the data you have collected from various places, it provides an easy way to navigate operations data. The real value is in assurance, which can be achieved in many ways, which we will discuss in this book.

1.1 Why this Guide is Important?

Am going to list here few reasons on why you would need to read this book:

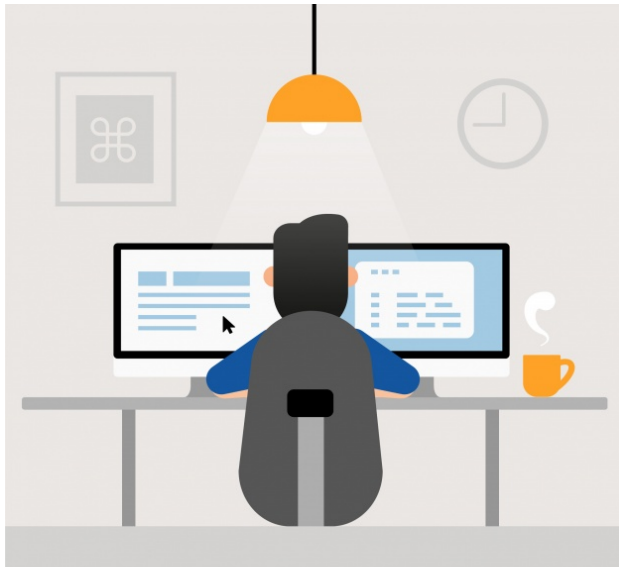
- System Admins who has implemented or used monitoring solutions as part of his/her daily duties, and probably faced challenges getting the right data that can explain performance of IT & Network resources.
- cloud solution architect must include cloud assurance as a core component of cloud architecture.
- Organizations managing IT system used monitoring who need a guide on what to watch to guarantee IT & Network service assurance.
- Despite the variety of tools and solutions out there, still some systems are failing to meet the SLAs, this guide will make it easy to understand those SLAs and KPIs required for effective measurement.
- The efficiency of the information systems depends on the efficiency of the monitoring systems that is integrated.
- Major companies like British Airways, BT, and many more have suffered from millions of loses to recover from computer systems crashes. [1]
- While using monitoring solutions, some flaws in the monitoring coverage may affect in great deal the profitability and revenues of such companies.

---

[1] https://www.networkworld.com/article/3200105/british-airways-outage-like-most-data-center-outages-was-caused-by-humans.html

- Having an insufficient monitoring solution is much worse than not having a monitoring system at all.
- The book will explore some of the latest service assurance use cases.
- Assist data scientists with explanation of IT & Network service assurance datasets for infrastructure as a service cloud.

1.2 The promise of monitoring?



As traditional enterprises shift investments to enable their Digital Transformation; they are in need for tools and solutions that speed up that transformation process.

Investments will help service providers build agile operations capabilities. The goal is to bring operations cost as low as possible, save time spent on repetitive tasks, and boost productivity to sell more services, reduce OPEX and make revenues.

The transformation journey is long and requires adaptation of various technologies, by time technologies are becoming very easy to use, but harder to operate. Think about how easy it is to check your calendar, book a taxi, find restaurants, Etc.

It's becoming so easy to consume, and it's changing so fast. But these large-scale application deployments, requires reliable operations. your customers have many options out there, there is nothing stopping them from trying other vendors. You may have the best application service, but if you cannot keep it running all time 24/7 then probably your business may be at facing more risk.

Redundancy was the ultimate solution for applications availability. Redundancy is basically replicating your workload so if one goes down the other instance will take place. Think about setting your alarm twice, so you won't miss your appointments.

That concept worked for some time, but as deployments became more complex. Software and hardware components had to be decoupled, each single component would have its own redundancy. Disaggregation enabled new technologies like cloud, containers, API driven applications, and microservices.

Modern IT & Network services demand agile infrastructure, and application deployment and operation models. We cannot hardcode procedures anymore. If you read this far in this book then you would probably know that onboarding a new server would have taken 6-12 months to go from a request to actual deployment. In 2020 the whole industry might change while you are still ordering your server to deploy new or existing services. It's not acceptable anymore to have such delay, instead of months we need to bring this down to hours if not minutes.

IT operations is like driving a car, there is absolutely no way to drive a car without visibility. Its what shows you the road ahead, you may collect information about all kind of stuff, but if its not offering you visibility, then it's useless. Visibility answers the question, what is happening around you, so you would respond, to speed up or hit the brakes or do other actions to guarantee safety till you reach destination on time. That's why to get a license to drive, your eye sight should be suitable for operating the vehicle.

Unlike the transportation industry, that put standards for car manufactures to provide at least the bare minimum of monitoring and operating cars, planes, trains and others; IT & network operations are still on best effort bases, with no enforced standards. But look at what we can do now; we have autonomous driving cars that is more accurate and more safe than human operated ones. in the future there will be no one sitting at the network operation center watching hundreds of screens, trying to fix problems, it will be all autonomous operations. and this is the nirvana stage where all service providers are aspiring for.

Technologies has evolved, more and more AI applications are created every day, that should help the next transformation phase to autonomous operations.

For that to happen we need couple of things:

1. Standards for telemetry data exchange.
2. Multiple levels of automation.
3. Monitoring and data management software/hardware.
4. Skills set.

There is a lack of standards in defining what a good monitoring solution is?, some industries started to include it as a critical part of its delivery model; for example, consider NFV (Network Function Virtualization) standards of ETSI (The European Telecommunications Standards Institute is an independent, not-for-profit, standardization organization in the telecommunications industry in Europe); Monitoring is considered one of the building blocks of the ITSI standard architecture, in fact, a vendor won't comply with ETSI standards unless monitoring is delivered as part of its reference architecture. In the ETSI standards, monitoring is integrated with virtual infrastructure management (VIM), as well as the NFV Orchestrator and VNF Manager.

It's considered a critical part, as the monitoring solution will monitor the infrastructure, and application and report it's health to the NFV orchestration framework which will then consult the VNF manager on the proper action to take to keep the application up and running, that may include moving the application from one node to another, scaling the application in or out, or even destroying the application, and initiating a new instance of it.

If you noticed what had just happened, the application state has changed multiple times throughout its life cycle. Hence, the monitoring solution need to be aware of these changes and keep tracking of all changes in the behavior of the operated devices. And cope with the dynamic topology, otherwise; wrong information might be reported, which is something we don't want in our operations, as it may lead to calculating service levels wrongly.

In my opinion; standards should guarantee the availability of telemetry data, regardless of it's:

1. Data structure.
2. Location.
3. Volume.
4. Frequency.
5. Device or application type.

Monitoring is a crucial part for the transformation journey as it's the only way to sustain delivery of your enterprise services and keep performance levels at higher rates. It's a reflection on how efficient IT and network operations are.

The promise of monitoring is to provide decision maker with the right information, in the right time to guarantee service delivery.

This guide shall demonstrate the different components of cloud models, and key performance indicators (KPIs) contributing to the performance and availability of these components.

2 Introduction

Any system that we use in our everyday life needs monitoring to measure its effectiveness, continue its development and get notified when it breaks. Since the first computer invented in 1946 till cloud solutions and big-data. The infrastructure architecture has changed, side by side application architecture evolved too, from monolithic applications to client-server, to three tiers, to service oriented and now micro service architectures.

Developing the systems that assure the proper operation of any computer system is, by its very nature, a game of constant catch-up.

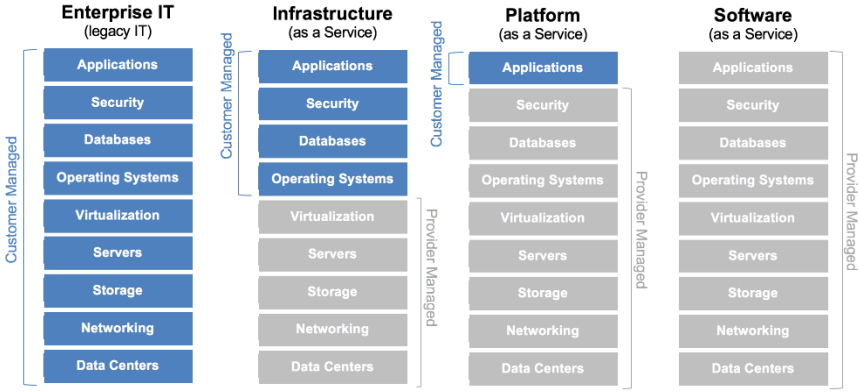
In this guide, I will try to explain what would a monitoring solution cover in terms of KPI's and Events, regardless of its implementation aspect.

There might be hundreds of monitoring solutions and platforms out there, and you can use this guide to qualify and explore its capabilities and coverage. In addition, to be productive, this writeup will focus on Infrastructure as a Service use cases, which is one of the cloud deployments offering models, we will not cover any specific application KPI's that is built on top of the cloud. Covering infrastructure as a service is the most critical part of any cloud offering, as it presents the backbone of the cloud offering.

The term **"cloud"** has recently become a part of our modern IT, there are many definitions and distinctions of different cloud deployment and service models.

There are essentially five characteristics that defines a cloud system, also there are different models for deployment and offering services on top.

Cloud System Characteristics	**cloud deployment model**	**cloud service models**
On-demand self-service Broad network access Resource pooling Rapid elasticity Measurable service	Public Private Hybrid	Software as a Service (SaaS) Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)



The previous diagram shows the different components of each cloud service models known today. The components of concern depend on the view of the user providing or using the cloud system.

For example, in a Software as a Service model; Service providers must be concerned on all components of the stack starting from the data center, networking, servers all the way up to applications as application as a service is the final product offered to end users.

On the contrary a service provider offering Infrastructure as a service (IaaS) will be only concerned into components that deliver only infrastructure frameworks that includes datacenter, networking, virtualization, storage, servers. The SLA of that type of service provided stops at the Operating system layer, the rest of the components are considered customer managed.

Infrastructure as a Service (IaaS) is a standardized, highly automated offering, where compute resources, complemented by storage and networking capabilities are owned and hosted by a service provider and offered to customer's on-demand. It also introduced service catalogs where users can request infrastructure resources (Virtual Machines, Storage, and Network Configurations) on the fly usually through a web portal.

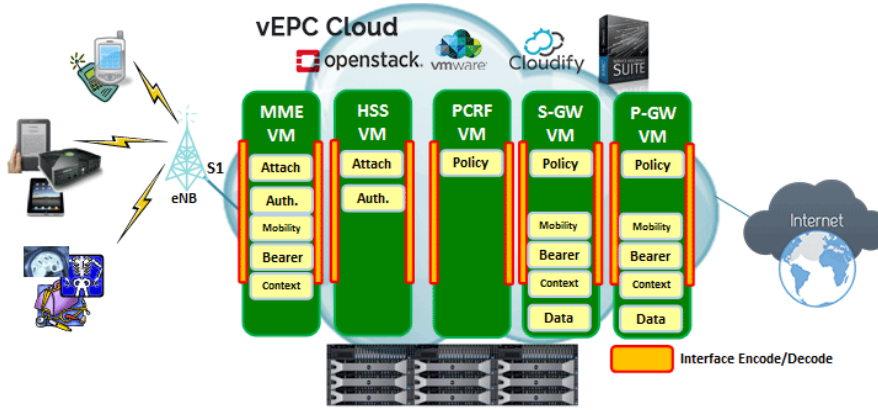
It offers great agility in providing services that enable end users to quickly deploy applications but also reduce the cost dramatically while keeping a constant supply of resource for end users in a highly available system.
The expectation from enterprises toward cloud offerings require the cloud provider to deliver higher Service-level agreements (SLA) as well as shorter remediation time provided an SLA occur. In addition to expected performance KPIs.

Cloud computing is attractive to consumers and businesses because of its efficiencies, convenience and pay-per-use pricing models. Telecom operators and service provider have always tried to maximize the value of their assets as cloud computing offers great advantages for service providers over their current enterprise IT Models. Telecom services are getting transformed from a traditional rigid IT system into flexible and agile cloud environment.

The Virtual Evolved Packet Core Example (vEPC) use case

Consider a telecommunication use case like **vEPC (virtual Packet Core)**. Virtual Evolved Packet Core (vEPC) is a framework for virtualizing the functions required to converge voice and data on 4G Long-Term Evolution (LTE) networks. vEPC moves the core network's individual components that traditionally run on dedicated hardware to software that operates on low-cost commercial off-the-shelf (COTS) servers. By virtualizing evolved packet core (EPC) functionality, mobile providers can theoretically customize networks to meet the unique requirements of individual customers, mixing and matching individual network components as needed at fraction of the time and in a more efficient way. Each node in a traditional EPC is virtualized in the cloud as one or more virtual machine or instance.

Service chaining is configured between different nodes performing a service. Networking between different nodes can be automated and changed easily to reflect tenants. The following diagram shows the high-level architecture of vEPC.



Service provider can use unified method for managing the infrastructure of their applications by adopting cloud technologies, by that the hardware can be from any vendor as long it provides the required throughput and performance removing vendor lock from investment decision.
Cloud operating systems like (VMware or OpenStack) acts as a controller for compute, storage, networking and Security resources.

An orchestrator software can create, provision and utilize hardware & software components, create new deployments of services and applications or changing exiting blueprints to accommodate the required scale.

A monitoring solution can be used to measure the availability, fault & performance of infrastructure components & applications in the cloud. It can also be integrated with the orchestrator using DevOps methodologies to change the deployed environment based on specific key performance indicators.

3 IaaS Cloud Components Monitoring

Any cloud offering will incorporate the following components, these components availability and performance are the keys to the operation success of the cloud. Operation success means higher availability and performance.

Service Level Agreements (SLA) is the contractual agreement between the cloud users and service provider that defines the KPIs that measure service levels and availability. SLAs establish customer expectations with regard to the service provider's performance and quality in a number of ways. Some metrics that SLAs may specify include:

- Availability and uptime - the percentage of the time services will be available.
- Specific performance benchmarks to which actual performance will be periodically compared.
- Application response time.
- The schedule for notification in advance of network changes that may affect users.
- Help desk response time for various classes of problems & Mean Time to Recover (MTTR)

SLA may specify availability, performance and other parameters for different types of customer infrastructure -- internal networks, servers and infrastructure components such as uninterruptable power supplies.

For example; While SLA Agreements are diverse from one provider to another. The components involved in providing these SLA are mostly comparable. The KPI's of those components are usually similar regardless of the component vendor.

> *For example: Network Interface Utilization (ifInOctets, ifOutOctets) within a host; is a valid KPI for measuring Ethernet interfaces utilization and performance, the KPI itself doesn't change if the vendor is Intel, Cisco or Edimax. However, the collection method of the metrics may be different. It can be through SNMP, CLI, API, or rest api etc.*

A breakdown of the IaaS cloud system components is mentioned in the following sections. Highlighting the required metrics and KPIs required to measure the availability and performance of each component. A monitoring solution need to be able to monitor all these components and relevant KPIs and report it efficiently.

In the coming section I will split our scope into groups and each group I will break it into components and sub-components, and then I will list the most important events to watch in addition to the most vital performance KPI's. this can be used as a guide line to evaluate the monitoring solution you are using whether it's an opensource, enterprise licensed product, or even a custom developed solution.

3.1 Datacentre

Datacenters should be covered by sensors to monitor the operation of the datacenter environment.

Weather conditions and unexpected disasters like leaks, floods, fires and malfunctioning air conditioners.

So, you can be alerted immediately when an event occurs so that you can take fast, corrective action. Recognizing these events is the critical first step.

Thresholds for each sensor need to be agreed on according to many design factors. The below are the main component that need to be watched.

In the table below the breakdown of each essential component in a datacenter

KPI Group	Description	Monitored Elements	KPIs & Alarms	Collection Method
Temperature and Airflow	A rise in temperature indicates an issue, Many CRAC/CRAH systems have built-in alarm fault panels that can be connected to your monitoring system to alert you the moment your cooling system detects an internal problem.	Datacenter room Racks/Servers Datacenter Floor	Rack Cooling Index (RCI), which is a measure of the cooling effectiveness of the IT equipment racks to ASHRAE levels Return Temperature Index (RTI), which is a measure of CRAH/CRAC air flow versus Rack air flow by measuring temperature differential across computer room air handler (CRAH)/CRAC and racks Alarm: Temperature reaching threshold.	Most Sensors Manufactures use the following methods to export metrics to enterprise monitoring systems. Send SNMP Traps SNMP Polling SOAP API REST API CSV file Collections Syslog Email SMS
Humidity	wireless sensor system enabled the data center operators to monitor Relative humidity in real-time.	Datacenter room Racks Datacenter Floor	Humidity Value	

KPI Group	Description	Monitored Elements	KPIs & Alarms	Collection Method
Leak Detection	Detects the presence of water leaks and other conductive fluids.	Datacenter ceiling & floor	Alarm: Leak detected	
Power & Equipment failure	Monitoring power equipment's including PDU, Panel Boards, and Busway for power consumption, availability and redundancy	PDU	current (amps) current voltage Power (kVA, kW) Energy Consumption (kWh) Power Uptime Alarm: Increased Power Usage Alarm: Power Loss Alarm: Power Redundancy Activated	
Smoke, Gas & Fire	Detect smoke gas and fire with a datacenter, rack	Datacenter Rack	Alarm: Smoke detected Alarm: gas leak detected	
Physical Security	Physical security and intrusion detection may indicate risk for data due to un-predictable access.	Datacenter doors Rack doors Movement with datacenter	Alarm: Glass Break Detected Alarm: motion detected Alarm: vibration detected. Alarm Door Opened detected. Alarm: Access Denied to door x	

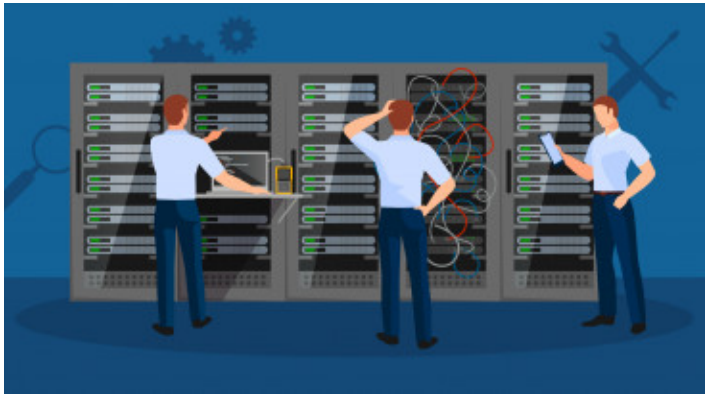
The sensors and various components listed above might be coming from different vendors, the manuals and technical specification should list the alarms methodology and if can also push some performance KPI's.

usually, these sensors will dump the monitoring data into a proprietary system or a third-party monitoring solution standard protocols or output like SNMP, XML, or Rest.

The list above can be used as a compliance Metrix when evaluating technology vendors.

You need also to make sure that those KPI's reach the operators of the datacenter, this can prevent humongous damage and avoid failure of service.

3.2 Servers, hosts, and compute resources



A server is a computer that provides data or computing power to other computers. That definition deviated into multiple to reflect the changes in technology.

A Hypervisor server is a server that can virtualize its resources and offer it to Virtual Servers created within its space, that formed the foundation of cloud computing.
A Bare-metal Server is a server that all its resources are used for one purpose and is not been virtualized in a cloud system or non-cloud system. The metrics that describe the performance of server are usually the same despite some changes due to the nature and use of the type of servers we are monitoring.

In that case we will need to distinguish between a host and guest virtual machine (VM), as the host will be doing more functions by definition.

The following table lists the most important KPI's to be collected from a hosts and virtual machines.

Category	Domain	KPI (Avg, Max, Min)	Description
System	**Host**	**Power On**	Indicates if a system is powered on or off
		System Uptime	Number of seconds since the last system startup.
	Host	**Availability**	The percentage of the availability of the system from the system uptime
	Host	**Reachability**	Indicates if the host is reachable from the monitoring solution.
	Host	**Is in Maintenance**	Indicates if the system is in maintenance mode.
	Fan	**Fan speed**	Indicates the fan's speeds
	Fan	**Fan Health State**	
	Temperature	**Temperature Current Value**	Om
	Power	**Power Use**	Host Power Use in Watts
	Power	**Power Capacity**	Host Power Capacity in Watts
	Power	**Power Supply Available**	Indicate the availability of the power supply
	Host	**Total Number of VM**	Total No of Virtual Machines hosted in a compute node
	VM	**Running Processes Count**	
	VM	**Sleeping Processes Count**	
	VM	**Zombie Processes (Linux/Unix)**	
CPU	**Host/VM**	**CPU Utilization**	Utilization of CPU

Category	Domain	KPI (Avg, Max, Min)	Description
			Utilization = 100% - (Percentage of time that is spent in idle task)
	Host/VM	**CPU Total Capacity**	Total CPU capacity in megahertz.
	Host/VM	**CPU Wait**	CPU time spent in idle state.
	Host/VM	**IO Wait**	IO wait time in milliseconds.
	Host/VM	**CPU Usage**	CPU use in megahertz
	Host/VM	**CPU Core Utilization**	Percent core utilization.
	Host/VM	**Total CPU mhz**	Total amount of CPU resources of all Hosts in the cluster. The maximum value is equal to the frequency of the processors multiplied by the number of cores. totalmhz = CPU frequency × number of cores
	Host	**Number running VCPUs**	Number of virtual CPUs on powered-on virtual machines.
	VM	**Reserved Capacity**	Total CPU capacity reserved by the virtual machines within a Host.
	Host/VM	**CPU Swap Wait**	Amount of time waiting for swap space.
	Host/VM	**Number of Cores**	Indicates no of Core in a system
	Host/VM	**Number of CPU**	Indicates no of CPU in a system
	VM	**CPU Latency**	Percentage of time the VM is unable to run because it is contending for access to the physical CPUs.
	Host	**CPU Reserved Capacity**	The sum of the reservation properties of the (immediate) children of the Host's root resource pool.

Category	Domain	KPI (Avg, Max, Min)	Description
Memory	**Host/VM**	**Total Memory**	Total available memory in a Host or virtual machine
	Host/VM	**Free Memory**	Total Free Memory
	Host/VM	**Used Memory**	Total memory used in kilobytes in Host or virtual machine
	Host/VM	**Free Swap Percent**	Indicates Free Swap Percent
	Host/VM	**Page Major Faults**	Indicates Page Major Faults
	Host/VM	**Page faults**	Indicates Page faults
	Host	**Reserved capacity**	Show the reserved capacity of memory for a hypervisor, it can be broke down for each instance
	Host/VM	**Memory heap**	Amount of memory allocated for heap.
	Host/VM	**Memory Heap free**	Amount of free space in the heap.
	Host/VM	**Memory Swap Total**	Total Swap Memory
	Host/VM	**Memory Swap in**	Amount of memory swapped in.
	Host/VM	**Memory Swap Out**	Amount of memory swapped out.
	Host/VM	**Memory Swap Used**	Amount of memory used for swapped space in kilobytes.
	Host/VM	**Memory sys Usage**	Amount of memory used by the kernel.
	Host	**Memory unreserved**	Amount of unreserved memory in kilobytes.
	Host/VM	**Memory Zero**	Amount of memory that is all zero.
	Host/VM	**Memory Swap in**	Rate at which memory is swapped from disk into active memory during the interval in kilobyte per second.

Category	Domain	KPI (Avg, Max, Min)	Description
	Host/VM	**Memory Swap Out**	Rate at which memory is being swapped from active memory to disk during the current interval in kilobytes per second.
	Host/VM	**Memory active write**	Average active writes in kilobytes.
	Host/VM	**Memory compressed**	Average memory compression in kilobytes.
	Host/VM	**Memory compression Rate**	Average compression rate in kilobytes per second.
	Host/VM	**Memory decompressio n Rate**	Decompression rate in kilobytes per second.
	Host/VM	**Memory total Capacity**	Total capacity in kilobytes.
	Host/VM	**Memory latency average**	Percentage of time the VM is waiting to access swapped or compressed memory.
	Host/VM	**Memory Swap In Rate**	Rate at which memory is being swapped from Host cache into active memory.
	Host/VM	**Memory Swap Out Rate**	Rate at which memory is being swapped to Host cache from active memory.
Network	**Interface/Ho st/VM**	**Packets Rx Per Sec**	Number of packets received in the performance interval.
	Interface/Ho st/VM	**Packets Tx Per Sec**	Number of packets transmitted in the performance interval.
	Interface/Ho st/VM	**Packets Per Sec**	Number of packets transmitted and received per second.
	Host/VM	**Network usage average**	The sum of the data transmitted and received for all the NIC instances of the Host or virtual machine.

Category	Domain	KPI (Avg, Max, Min)	Description
	Host/VM	**Usage capacity**	I/O Usage Capacity.
	Interface/Host/VM	**Max Observed Tx KBps**	Max observed transmitted rate of network throughput.
	Interface/Host/VM	**Max Observed Rx KBps**	Max observed received rate of network throughput.
	Interface/Host/VM	**Dropped Packets Rx**	Number of received packets dropped
	Interface/Host/VM	**Dropped Packets Tx**	Number of transmitted packets
	Interface/Host/VM	**Dropped Packets**	Number of dropped packets
	Interface/Host/VM	**Broadcast Rx**	Number of broadcast packets received
	Interface/Host/VM	**Broadcast Tx**	Number of broadcast packets transmitted
	Interface/Host/VM	**Errors Rx**	Number of packets with errors received.
	Interface/Host/VM	**Errors Tx**	Number of packets with errors transmitted.
Disk	**Disk/Host/VM**	**Disk usage (Avg)**	Average of the sum of the data read and written for all of the disk instances of the Host or virtual machine (Kbps)
	Disk/Host/VM	**Disk usage capacity**	This metric is a function of storage|usage_average and disk|workload. storage|usage_average is an average over all storage devices. This means that disk|usage_capacity is not specific to the selected VM or the Host of the VM.

Category	Domain	KPI (Avg, Max, Min)	Description
	Disk/Host/VM	**Disk total latency**	The average amount of time taken for a command from the perspective of a Guest OS. This is the sum of Kernel Command Latency and Physical Device Command Latency.
	Disk/Host/VM	**Disk reads**	Average number of read commands issued per second during the collection interval.
	Disk/Host/VM	**Disk writes**	Average number of write commands issued per second during the collection interval.
	Disk/Host/VM	**Disk commands issued**	The number of disk commands issued in the performance interval.
	Disk/Host/VM	**Disk commands aborted**	The number of disk commands aborted in the performance interval.
	Disk/Host/VM	**Disk read latency**	The average time taken to complete a read from the physical device.
	Disk/Host/VM	**Disk kernel read latency**	The average time spent in ESX Server VMKernel per read.
	Disk/Host/VM	**Disk guest read latency**	The average amount of time taken for a read from the perspective of a Guest OS. This is the sum of Kernel Read Latency and Physical Device Read Latency.
	Disk/Host/VM	**Disk queue read latency**	The average time spent in the ESX Server VMKernel queue per read.
	Disk/Host/VM	**Disk write latency**	The average time taken to complete a write from the physical device.

Category	Domain	KPI (Avg, Max, Min)	Description
	Disk/Host/VM	**Disk kernel write latency**	The average time spent in ESX Server VMKernel per write.
	Disk/Host/VM	**Disk guest write latency**	The average amount of time taken for a write from the perspective of a Guest OS. This is the sum of Kernel Write Latency and Physical Device Write Latency.
	Disk/Host/VM	**Disk queue write latency**	The average time spent in the ESX Server VMKernel queue per write.
	Disk/Host/VM	**Disk device request latency**	The average time taken to complete a command from the physical device.
	Disk/Host/VM	**Disk kernel request latency**	The average time spent in ESX Server VMKernel per command.
	Disk/Host/VM	**Disk queue reqest latency**	The average time spent in the ESX Server VMKernel queue per command.
	Disk/Host/VM	**Disk io pending**	Number of Outstanding IO Operations.
	Disk/Host/VM	**Disk io queued**	Queued Operations.
	Disk/Host/VM	**Disk demand**	Demand.
	Disk/Host/VM	**Disk queue requests**	Sum of Queued Operation and Outstanding Operations.
	Disk/Host/VM	**Disk max io observed**	Max Observed IO for a disk.
	Disk/Host/VM	**Disk highest latency**	Highest Latency.
	Disk/Host/VM	**Disk max queue depth**	Maximum queue depth during the collection interval.
	Disk/Host/VM	**Disk scsi reservation conflicts**	SCSI Reservation Conflicts.

Category	Domain	KPI (Avg, Max, Min)	Description
	Disk/Host/VM	**Disk commands issued**	The number of disk commands issued in the performance interval.
	Disk/Host/VM	**Disk commands aborted**	The number of disk commands aborted in the performance interval.
	Disk/Host/VM	**Disk read latency**	The average time taken to complete a read from the physical device.
File System	**File System**	**FS Capacity**	Total capacity on guest file system in megabytes.
	File System	**FS Free space**	Total free space on guest file system in megabytes.
	File System	**FS Percentage**	Percent guest file system.
	File System	**FS Usage**	Total usage of guest file system.
	File System	**FS Free space total**	Total free space on guest file system.
	File System	**FS Capacity total**	Total capacity on guest file system.
	File System	**FS Percentage total**	Guest file system space utilization.
	File System	**FS Usage total**	Total usage of guest file system.
	File System	**File System Read/Write Percentage**	File System Read/Write Percentage / Minute

The KPI's listed above can be collected at many levels. In case of bare-metal hosts/servers, some hardware vendors may offer an interface to monitor some of those KPI's even without an operating system installed. Dell offers IDRAC interface, HP offers ILOM. There are some how similar in the offerings however, the challenge is usually in the integration of those KPI's with the monitoring solution in an efficient way.

Also, these KPI's are reported to the Operating System, whether it's a virtualization system or typical Linux, windows, AIX, etc.. the monitoring solution may deploy local agents, which is a small piece of software that will run on the system and continuously monitor those components and report the output to a management platform. Other monitoring solution might adapt the agentless approach, and it can monitor the same KPI's by using a protocol like SNMP to pull the information from source.

The choice between agent based or agent less options simple depends on how efficient the monitoring platform and the overhead it creates (if any). It's really hard to determine; as every monitoring system works use a different approach and coded differently. A proof of technology is required to measure the reliability of the system and any overhead it creates. But as a rule of thumb; all these KPIs are already there in the operating system, the agent will be just pulling those measurements and sending them over the wire. So, it's assumed if there is any overhead; then its minimal.

> ***Note:*** *Virtual Machines (vm) may also be referred to as instances. Hypervisors may be referred to as Host.*

> ***Note:*** *File System is mainly monitored in virtual machines or hosts that are not hypervisors. As usually hypervisors are connected to external storage and virtualizing this storage for virtual machines, file systems lay over the virtualized storage and is the only visible storage from application side.*

Remember that in infrastructure as a service you only offer the virtual machines as a service the end customer would be responsible of monitoring, but in most cases, you would need to provide the customer with a platform to monitor the operation of his environment. Even if your customers are using their own monitoring plugins, that won't affect the service offering, while you need to monitor the host KPI's yourself, offering the platform to monitor the rest of the KPIs can be useful and profitable from a business perspective.

Think of what Amazon is doing, it's offers AWS EC2 instances (virtual machines), and it offers also AWS CloudWatch. Amazon CloudWatch is a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers. You can use CloudWatch to set high resolution alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to optimize your applications, and ensure they are running smoothly.

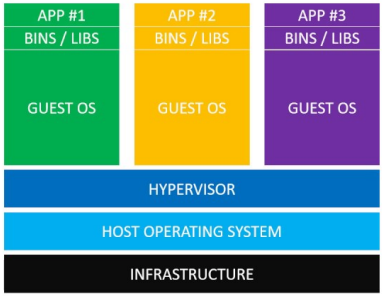
3.2.1 Containers monitoring

Containers are the new virtual machines of the next 10 years. While containers are so different than a virtual machine. A container main purpose is to separate the user space for each application component. Containers can enable customers to pack a lot more applications into a single physical server than a virtual machine (VM) can, simply because its more efficient in using the host resources. Container technologies, such as Docker, beat VMs at this part of the cloud or data-center game.

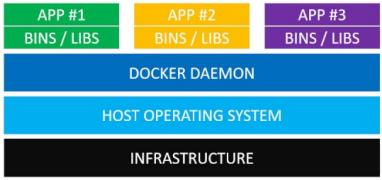
VMs take up a lot of system resources. Each VM runs not just a full copy of an operating system, but a virtual copy of all the hardware that the operating system needs to run. This quickly adds up to a lot of RAM and CPU cycles. In contrast, all that a container requires is enough of an operating system, supporting programs and libraries, and system resources to run a specific program. What this means in practice is you can put two to three times as many as applications on a single server with containers than you can with a VM.

Having said that, 99% of the containers deployments nowadays are implemented inside virtual machine and not bare metal, the reason for that is ease of management, and scalability.

The following diagram shows the difference between VMs and Containers.



Virtual Machines



Docker Containers

As displayed the containers minimize the need to deploy multiple operating systems and simplify the deployment of applications. However, The KPIs that applies to a virtual machine will also apply to a container. The container itself is not aware that it's a container, it thinks that it's a physical server, same like virtual machines.

However, the complexity in monitoring containers is not in monitoring individual docker containers, it's the monitoring of services and availability of the whole system.

> **Note:** Container management and monitoring shall need more investigation and I will bring it as a separate topic in future books. But the same concepts that you learn here should be applied in the container's world.

One of the latest implementations for monitoring containers is what Redhat has implemented to monitoring containers deployed in Redhat OpenShift Containers Framework using opensource tools including but not limited to Collectd and Prometheus.

3.3 Networking

Network devices are the core components of any system in the datacenter, by monitoring those devices you will be able to react on many issues before users or clients even notice a problem.

When looking into network monitoring we should consider not only physical network devices but also virtual network function as within a cloud deployment both flavors would exist and may be co-working together.

Physical devices may include: Routers, Switches, Firewall, Load balancers while virtual devices may include the same but adding virtual as a prefix.
For Example, virtual router or virtual switch. The availability of those devices and functions define the applications connectivity and availability, performance KPI also provide an indication on the activities happening on those devices, capacity, throughput, bandwidth, and usage. All that contribute to the health of network operations and promised SLA to cloud customers.

The following KPI represent the most important KPIs to focus on when monitoring network functions:

3.3.1 Network Topology

Topology is the core of any monitoring systems. Without topology the system won't be able to understand the dependency between components, some systems use the topology collected to calculate root cause of issues. This is very effective to understand the dependencies between hardware layers but also software and logical layers in your infrastructure.

Category	Domain	KPI (Avg, Max, Min)	Description
Network Topology	**All network elements**	**Network element type count**	While network topology is not a KPI, some Counters may indicate the size or changes within the network
		Topology Changed	Indicate the change in the network topology. For example: adding new devices or removing existing.

3.3.2 Physical and Virtual Devices (Routers/Switches)

Category	Domain	KPI (Avg, Max, Min)	Description
Device	All Network Elements [2]	Availability	Indicates the availability of the devices, it's derived from the system up-time. Usually collected using SNMP. And for Virtual Devices usually collected using SOAP or REST API.
		Reachability	Indicates if the system components is reachable from the monitoring system or not.
Device (Switch, Router)	Interfaces	ifAdminStatus	The state of this interface: 100 (up), 0 (down), 0 (testing)
		ifOperStatus	The current operational state of the interface: 100 (up), 0 (down), NaN (testing), 100 (dormant), 0 (notPresent), 0 (lowerLayerDown)
		CurrentUtilization	Current utilization of the interface calculated from ifInOctets/ifOutOctets and ifSpeed

[2] All network elements refers to all physical and virtual network elements, including Router, vRouter, Switch, vSwitch, Load balancer, Firwall, etc…

Category	Domain	KPI (Avg, Max, Min)	Description
		Availability	Availability of interface in percentage computed from ifAdminStatus and ifOperStatus [(Availability = 'ifOperStatus' if 'ifAdminStatus'=100) else (NaN)]
		ifInDiscards	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space
		ifInErrors	For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character- oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol

Category	Domain	KPI (Avg, Max, Min)	Description
		ifInUnknownProtos	For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter will always be 0
		ifOutDiscards	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space

Category	Domain	KPI (Avg, Max, Min)	Description
		ifOutErrors	For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors

Category	Domain	KPI (Avg, Max, Min)	Description
		ifSpeed (stdIfT3)	An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object should be zero
		ifInOctets (stdIfT3)	The total number of octets received on the interface, including framing characters
		ifOutOctets (stdIfT3)	The total number of octets transmitted out of the interface, including framing characters

Category	Domain	KPI (Avg, Max, Min)	Description
		ifInUcastPkts (stdIfT3)	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer
		ifInNUcastPkts	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast or broadcast address at this sub-layer
		ifOutUcastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent
		ifOutNUcastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent

Category	Domain	KPI (Avg, Max, Min)	Description
		ifSpeed (stdIfXT3)	An estimate of the interface's current bandwidth in units of 1,000,000 bits per second
		ifInMulticastPkts	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses
		ifInBroadcastPkts	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer
		ifOutMulticastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses

Category	Domain	KPI (Avg, Max, Min)	Description
		ifOutBroadcastPkts	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent
		ipForwDatagrams	The number of input datagrams for which this switch was not their final IP destination
		ipForwarding	The indication of whether this switch is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to it
		Uptime	The time (in hundredths of a second) since the network management portion of the system was last re-initialized
All Network Devices (Physical)	Power Supply	Availability	Operational status of Power supply: 100%(on) or 0% (offEnvOther,offAdmin, offDenied,offEnvPower, offEnvTemp,offEnvFan, failed,onButFanFail,offCooling,offConnectorRating,onButInlinePowerFail)

Category	Domain	KPI (Avg, Max, Min)	Description
	Processor	Current Utilization	The average utilization of CPU on the active supervisor
	Memory	Current Utilization	The average utilization of memory on the active supervisor
	Temperature	Temperature	Sensor reading
	Disk/Flash	StorageSize	Total size of the storage
	Disk/Flash	StorageFree	Free space in storage
Virtual Switch[3]	vSwitch[4]	MTU	Number of attached physical NICs.
	vSwitch Port	rx_bytes	Total ingress traffic (KBps).
		tx_bytes	Total egress traffic (KBps).
		ucast_tx_pkts	Egress unicast packets per second.
		mcast_tx_pkts	Egress multicast packets per second.

[3] The KPIs for virtual switch is quite similar to physical switches. As what important here is to focus on the traffic quality and utilization and making sure there is no dropped traffic or traffic with errors.

[4] The KPIs may be collected using different methods for example in the case of vmware, the API from vCenter provide such KPIs in the case of OpenStack OpenvSwitch the command ovs-ofctl dump-ports portname provide this information

OFPST_PORT reply (xid=0x2): 1 ports

port LOCAL: rx pkts=23, bytes=1278, drop=0, errs=0, frame=0, over=0, crc=0

tx pkts=369369, bytes=62820789, drop=0, errs=0, coll=0

Category	Domain	KPI (Avg, Max, Min)	Description
		bcast_tx_pkts	Egress broadcast packets per second.
		ucast_rx_pkts	Ingress unicast packets per second.
		mcast_rx_pkts	Ingress multicast packets per second.
		bcast_rx_pkts	Ingress broadcast packets per second.
		dropped_tx_pkts	Egress dropped packets per second.
		dropped_rx_pkts	Ingress dropped packets per second.
		rx_pkts	Total ingress packets per second.
		tx_pkts	Total egress packets per second.
		utilization	Utilization in (KBps).
		dropped_pkts	Total dropped packets per second.
		dropped_pkts_pct	Percentage of dropped packets.
		maxObserved_rx_bytes	Max observed ingress traffic (KBps).
		maxObserved_tx_bytes	Max observed egress traffic (KBps).
		maxObserved_utilization	Max observed utilization (KBps).

3.3.3 Firewalls

The firewall implements packet filtering and thereby provides security functions that are used to manage data flow to, from and through the router. Along with the Network Address Translation it serves as a tool for preventing unauthorized access to directly attached networks and the router itself as well as a filter for outgoing traffic.

When looking into firewall performance there might be few indicators to determine the efficiency of the firewall. While that firewall could be a software firewall or physical appliance, they all share the same characteristics by definition. The following are the main KPIs to look into when evaluating a firewall function:

KPI [5] [6] **(Avg, Max, Min)**	**Description**
Availability	Firewall Availability is the main KPI to make sure security functions are up and running. Monitoring solutions should be aware of high availability modes of firewalls
Memory Usage	Shows the percentage of Memory of a firewall device or virtual instance
CPU Usage	Show the percentage of CPU usage of firewall device or Virtual Instance
Interface Utilization	Show received and sent traffic per firewall interface.

[5] **http://soa.sys-con.com/node/2266270**

[6] **https://tools.ietf.org/html/rfc3511**

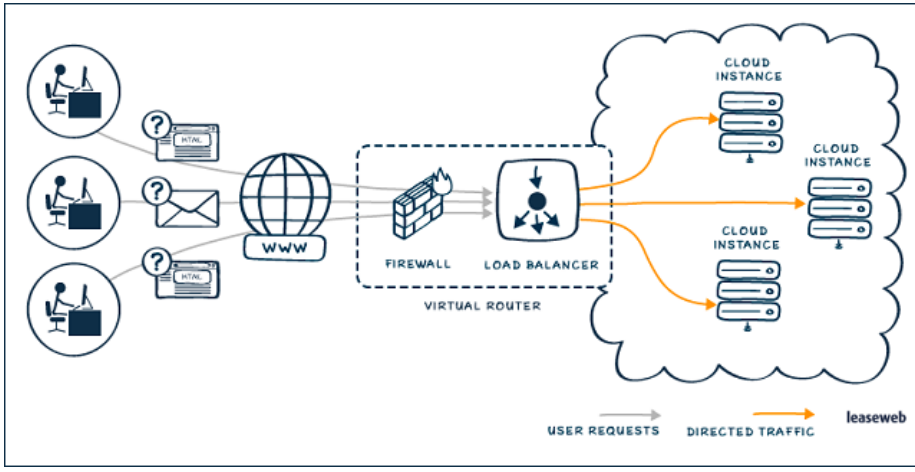
Number of shadowed or redundant rules	Shadowed rules refer to rules that are masked, completely or partially. A rule base filled with shadowed rules is not only inefficient, it puts a much greater strain on the firewall then is necessary, which can lead to performance issues.
Number of unused rules	Unused rules can lead to serious exposures, such as allowing access to a server that is no longer being used and, as a result, exposing a service likely not properly patched
Number of unused objects	An object is a component of a rule, and a single field of a rule (i.e., source, destination or service) can have multiple objects - such as a business unit having access to multiple destinations and/or services. Not only do unused objects appear much more frequently than unused rules, they are that much harder to find manually. Cleaning up unused objects can significantly tighten up a rule base and often lead to improved performance.
Number of rules with permissive services	The most common examples of this are rules with "ANY " in the service field, but in general, permissive services give more access then is needed to the destination by allowing additional services (which are often applications), which can lead to unauthorized use, allow the service to be a springboard to other parts of the network, or leave it exposed to malicious activity.

Number of rules with risky services (such as telnet, ftp, snmp, pop, etc.) in general or between zones (i.e., between Internal, DMZ, External, or between development and production networks)	Risky services are deemed risky because they usually allow credentials to be passed in plain text, often contain sensitive info or enable access to sensitive systems. Any service that exposes sensitive data or allows for shell access should be tightly monitored and controlled.
Number of expired rules	Any rule that was created on a temporary basis and has clearly expired is just taking up space and does not need to remain in the rule base. If there is no documentation as to when or why the rule expired, check the firewall logs for its "hit count" (or usage, in firewall management-speak).
Number of unauthorized changes	These are rules that are not associated with a specific change ticket. In order to ensure all requests are properly handled, all requests, from initial request to final implementation, and should be managed via a ticketing system
Number of rules with no documentation	While the comments section of a firewall rule has text limits that inhibit proper documentation, all change tickets have a comments section, which can be used to provide a business justification for the rule.
Number of rules with no logging	Proper firewall management is impossible without leveraging the data found in firewall logs. Similar to other areas of IT, there was a resistance to turning on logging because it would cause performance issues.
Number of Packets allowed	Number of Packets allowed

Number of Active Sessions	Number of Active Sessions
Number of Rejected Sessions	Number of Rejected Sessions
Number of Dropped Packets	Number of Dropped Packets
Number of Attacks / Type	Number of Attacks / Type
Number of Sessions	Including Active, Accept and Failed Sessions
Session Failures	The device sessions load is calculated as a percentage of Failed sessions divided by the Current Active sessions.

3.3.4 Load balancers

A load balancing setup includes a load-balancing virtual server and multiple load-balanced application servers. The virtual server receives incoming client requests, uses the load balancing algorithm to select an application server, and forwards the requests to the selected application server. The following conceptual drawing illustrates a typical load balancing deployment. Another variation involves assigning a global HTTP port.



KPI (Avg, Max, Min)	Description
Availability	Load balancer Availability is the main KPI to make sure security functions are up and running. Monitoring solutions should be aware of high availability modes of load balancers
Failover Pair Status	Shows Status of load balancer failover pairs
Memory Usage	Shows the percentage of Memory Used
CPU Usage	Show the percentage of CPU Used

Interface Utilization	Show received and sent traffic per load balancer interface/Trunk
Interface Availability	Show Availability of Interfaces, Ports, Trunks also VLANs
Environmental Factors	Environmental factors, such as power supply status, CPU temperature, chassis

3.3.5 Network Protocols (BGP, MPLS)

The Border Gateway Protocol (BGP) is the protocol used throughout the Internet to exchange routing information between networks. It is the language spoken by routers on the Internet to determine how packets can be sent from one router to another to reach their final destination. BGP has worked extremely well and continues to the be protocol that makes the Internet work. BGP is implemented in and between datacenters, it fairly important to get some KPIs for the health of the BGP setup. as networks are transforming now and Software Defined Networking (SDN) is becoming the new network standard. BGP is still implemented in many architectures that includes as well SD-WAN or Software defined wan. As BGP is used to define routes that runes inside internet IP-Sec Tunnels for SD-WAN Tenants and Branches. Multi-protocol label switching (MPLS) is a way to insure reliable connections for real-time applications, but it's expensive, leading enterprises to consider SD-WAN as a way to limit its use. However, in today's world we see MPLS and SD-WAN deployed together or integrated somehow.
For that as much as we need modern monitoring solution that can talk to SDN-Controller, and understand the dynamic nature of today's network; also we need to have visibility into the legacy stuff.

Category	Domain	KPI (Avg, Max, Min)	Description
	BGP (BGP Peer)	**bgpPeerState**	The BGP peer connection state
		bgpPeerRemoteAs	The remote autonomous system number

Network Protocol Configure on Network Device/Interface	**bgpPeerInUpdates**	The number of BGP UPDATE messages received on this connection. This object should be initialized to zero (0) when the connection is established
	bgpPeerOutUpdates	The number of BGP UPDATE messages transmitted on this connection. This object should be initialized to zero (0) when the connection is established
	bgpPeerInTotalMessages	The total number of messages received from the remote peer on this connection. This object should be initialized to zero when the connection is established
	bgpPeerOutTotalMessages	The total number of messages transmitted to the remote peer on this connection. This object should be initialized to zero when the connection is established
	bgpPeerLastError	The last error code and subcode seen by this peer on this connection. If no error has occurred, this field is zero. Otherwise, the first byte of this two byte OCTET STRING contains the error code, and the second byte contains the subcode

bgpPeerFsmEstablishedTransitions	The total number of times the BGP FSM transitioned into the established state
bgpPeerFsmEstablishedTime	This timer indicates how long (in seconds) this peer has been in the Established state or how long since this peer was last in the Established state. It is set to zero when a new peer is configured or the router is booted
bgpPeerInUpdateElapsedTime	Elapsed time in seconds since the last BGP UPDATE message was received from the peer. Each time bgpPeerInUpdates is incremented the value of this object is set to zero (0)
REPORT> BGP Traffic (msg/sec)	Shows the overall BGP traffic (peaks may indicates abnormal activity (eg. high route changes)
REPORT> Exception: Flapping (or disconnected) BGP Neighbors	List all disconnect BGP sessions or flapping sessions (a session less than 1-hour duration average is considered flapping
REPORT> View per BGP Neighbors	Shows all properties and statistics per BGP Neighbors
REPORT> View per BGP Routers	Shows all properties and statistics per BGP Router

	MPLS Interface	**mplsInterfaceInLabelsUsed**	This value indicates the specific number of labels that are in use at this point in time on this interface in the incoming direction.
		mplsInterfaceInPackets	This variable reflects the number of labeled packets that have been received on this interface.
		mplsInterfaceInDiscards	The number of inbound labeled packets, which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a labeled packet could be to free up buffer space.
		mplsInterfaceFailedLabelLookup	This value indicates the number of labeled packets that have been received on this interface and were discarded because there were no matching entries found for them in mplsInSegmentTable
		mplsInterfaceOutLabelsUsed	Indicates the number of top-most labels in the outgoing label stacks that are in use at this point in time on this interface.

		mplsInterfaceOutPackets	This variable contains the number of labeled packets that have been transmitted on this interface.
		mplsInterfaceOutDiscards	The number of outbound labeled packets, which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a labeled packet could be to free up buffer space.
		mplsInterfaceOutFragments	This variable indicates the number of outgoing MPLS packets that required fragmentation before transmission on this interface.
	VRF	**mplsVpnVrfPerfRoutesAdded**	Indicates the number of routes added to this VPN/VRF over the course of its lifetime.
		mplsVpnVrfPerfRoutesDeleted	Indicates the number of routes removed from this VPN/VRF.
		mplsVpnVrfPerfCurrNumRoutes	Indicates the number of routes currently used by this VRF.

3.3.6 Traffic Monitoring (NetFlow, sFlow, IPFIX)

Netflow, sFlow, IPFIX are standard protocols for monitoring traffic in the network, its configured-on routers to export information about the traffic, senders and receivers. While it's a technology that is usually deployed in router, modern software defined switches like OVS, and modern SDN Controllers may export traffic flow data to monitoring solution in the same way.
These protocols will provide the following:

- **Identifying Top Talkers and Conversations in the network:** Determine which users and what applications are using maximum bandwidth, and drill down for conversational details.
- **Monitoring and projecting Traffic Trends and Usage Patterns:** View trends in network traffic, and determine top applications and peak usage times.
- **Defining Applications to Monitor Specific Traffic**: Use a combination of ports and protocols to define unlimited applications, and recognize this traffic exclusively in traffic reports. You can also mention a particular IP address to map an application.
- **Department based Bandwidth monitoring per Department**: Define departments based on IP addresses, and identify bandwidth usage and application usage for each department.
- **Managing Devices Exclusively**: Categorize devices and group them data into logical groups, and monitor traffic reports exclusively, for the groups.
- **Site to site traffic monitoring**: lets you monitor traffic between two specific sites, which are created based on IP Address or IP Network. This feature helps you understand the network traffic behavior between any two user's defined sites.

Traffic Monitoring Protocols can be enabled on Routers, Switches. But also virtual Switches like (VMWare Distributed vSwitch, NSX Router or OpenVSwitch [7]). Then traffic information will be sent to a monitoring host for storage, aggregation and reporting.

[7] http://docs.openvswitch.org/en/latest/howto/sflow/

3.3.7 SDN Controllers

SDN Controllers are used to control a software function, it could be a switch, router, firewall or even SD-WAN. The SDN Controller usually stand as a virtual machine/instance. So to monitor it from physical & virtual perspectives it can share these same KPIs of a traditional virtual machine or hots.

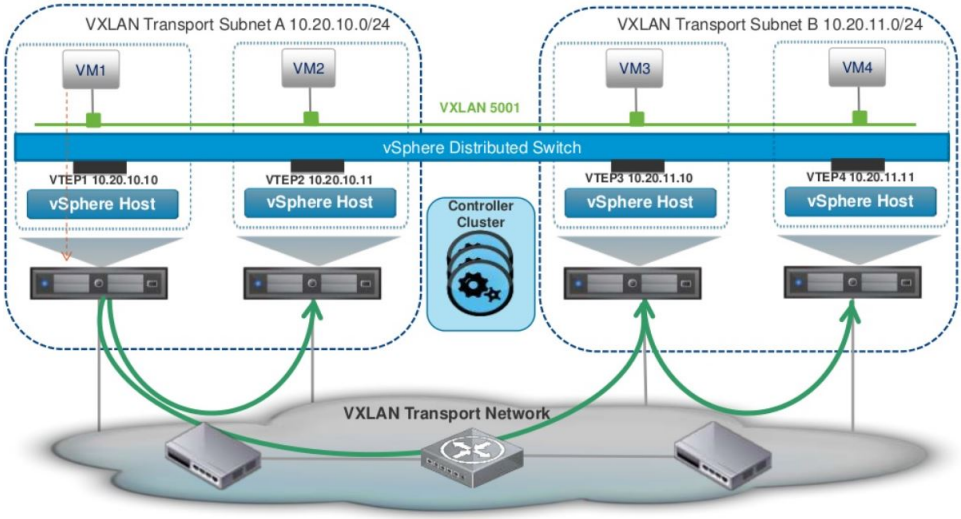
Also SDN controller usually export an API to export the underlying managed application performance indicators. So it could be possible to extend the KPIs collected for SDN application through the SDN Controller to the monitoring solution. In that manner customized and aggregated reports could be created for various domains.

SDN controller usually collect traffic and performance information about the network tenants deployed. This information can be pulled out using the available APIs and integrated with your monitoring solution. That is available using API Interface or using any of the standard traffic flow protocols like sFlow, IPFIX, Netflow.

3.3.8 Overlay Network Monitoring

Overlay networking is a method of using software to create layers of network abstraction that can be used to run multiple separate, discrete virtualized network layers on top of the physical network, often providing new applications or security benefits.

Created by taking two endpoints and creating a virtual connection between them, multiple secure overlays can be built using software over existing networking hardware infrastructure. These endpoints could be actual physical locations, such as a network port, or they could be logical locations designated by a software address in the networking cloud.



It some sort of encapsulation to be able to separate logical tunnels which leads to easier ways of creating network tenants. Tenants used be created using VLAN segmentation configured on a switch however It's limited to the number of tenants to create. Also VXLAN comes with a lot of scalability and security benefits.
Overlay networks are usually created on specific network bridge ports, virtual switch port. The availability of these ports need to be monitored in addition to the traffic send and received. Also tenants are created with specific tagging to separate tenants traffic passing from same interface.

The monitoring solution should be able to pull data about the interfaces and separate the monitored KPIs per tenants.
The following table show most important KPIs.

Category	**Domain**	**KPI (Avg, Max, Min)**
Availability	VXLAN Bridge, Physical Interface, VNIC, VTEP	Link Availability
Utilization	VXLAN Bridge, Physical Interface, VNIC, VTEP	Packets
		RX/TX Bytes
		RX/TX Packets
		RX/TX Errors
		RX/TX Errors
		RX/TX Dropped

Note: in a case of SDN (Software Define Networking), SDN Controller may give KPI's about the components of the SDN functions that cover performance, utilization, latency.

3.4 Virtualization & Cloud Services Monitoring

Monitoring virtualization of the cloud is not only depending on the physical and virtual applications providing the infrastructure. The cloud operating system (OpenStack, or vSphere) is composed of a list of applications & processes working together to provide the cloud system. While every cloud system have different process and application model they all share the same categories of core components. These components/services can be monitored to provide additional information on how the cloud system is running. Also some of these component may be highly available by design. The monitoring system need to be aware of the availability of these components.

The components can be categorized into services:

1. **Hypervisor Services:** the management components for providing compute, storage, network & security to the virtual layer.
2. **Communication Services**: indicates those services that allow the communication between the cloud system components and to external components as well.
3. **API Services:** each cloud system should expose API for third parties to interact with the system. The API may provide different functionalities including configurations, monitoring, management and orchestration.

4 **Orchestration Services:** the cloud system need application the can be used to automate tasks and operation based on specific criteria.
5 **Data Services:** the cloud system may include databases to store data. These database are usually in high available mode or cluster.
6 **Authentication Services:** the cloud system provide authentication for users with different privileges, the authentication may be separated into tenants as well. It can also be integrated with external authentication services like LDAP.

The previous components can be monitored for availability to make the sure that the cloud OS functions are up and running as per the design. Possible way of monitoring include (Process monitoring, API monitoring, HTTP Request Check, Ping, TCP Port Monitoring, Authentication Test Monitoring)

3.5 Storage

Storage metrics cover a lot of ground. Administrators can get information on everything ranging from storage performance to bandwidth and cost. Whether you're working with object, block or file storage. The metrics that matter most depend on how your storage is being used. The metrics can be divided into 4 categories:

- **Availability**
- **Performance**: including IOPS, bandwidth, and response time or latency
- **Capacity:** including percent utilization as well as reduction ratios. Storage capacity metrics tell users more than just the amount of storage there is to work with. They can detail what's been allocated, the amount of raw capacity, whether the storage is thick or thin provisioned, and more.
- **Additional performance metrics:** including reads, writes, random, sequential or IO size

- **Economics:** Storage metrics also help with the accounting side of storage management.

3.5.1 Block Storage

Block storage is data storage typically used in storage-area network (SAN) environments where data is stored in volumes, also referred to as blocks. Each block is assigned an arbitrary identifier by which it can be stored and retrieved, but no metadata providing further context. Database storage is a common use for block storage.

Each block acts as an individual hard drive and is configured by the storage administrator. These blocks are controlled by the server-based operating system, and are generally accessed by Fibre Channel, iSCSI or Fibre Channel over Ethernet protocols.

Because the volumes are treated as individual hard disks, block storage works well for storing a variety of applications. File systems and databases are common uses for block storage because they require consistently high performance. Email servers such as Microsoft Exchange use block storage in lieu of file- or network-based storage systems.

RAID arrays are a prime use case for block storage as well. With RAID, multiple independent disks are combined for data protection and performance. The ability of block storage to create individually controlled storage volumes makes it a good fit for RAID.

Virtual machine file systems are another common use for block-level storage. Virtualization vendors such as VMware support block storage protocols, which can improve migration performance and improve scalability. Using a SAN for block storage also aids virtual machine (VM) management, allowing for non-standard SCSI commands to be written.

While there are benefits to using block storage, there are also alternatives that may be better suited to certain organizations or uses. Two options stand out when it comes to facing off with block-level storage: file storage and object storage.

Category	Component/Part type	KPI (Avg, Max, Min)	Description
Block Storage	Array Disk LUN Storage Pool Port Application Host	**Availability**	Determine if the monitored component is up and running and available.
	LUN Storage Pool	**IORate**	The number of host IO operations performed each second by all Symmetrix volumes, including writes and random and sequential reads.
	Disk LUN Storage Tier	**CurrentUtilization**	The current utilization in percent.

	LUN Disk	**Cache(Read/Write)Hits**	
	LUN Disk	**CacheMisses (Read/Write)**	The amount of read/write misses on the FAST cache.
	LUN Disk	**Hard/Soft Errors (Read/Write)**	The total amount of Hard Errors in data.
	Disk	**DiskIdle**	The number of disk idle ticks.
	Port	**LinkStatus**	The percent indication of the link status on all Ports.
	LUN	**ReadCacheEnabled**	The status of LUN read cache state.
	LUN Disk Storage Pool RAID Group	**Read/Write Requests**	The amount of read/write requests in data per second.
	LUN Disk Storage Pool RAID Group	**Read/writeThroughput**	The read/write throughput in KB per second
	LUN Disk Storage Pool RAID Group	**ResponseTime**	The average time it takes to satisfy IO requests
	Disk LUN	**ServiceTime**	The average time it takes to deliver a service on the system including reading and writing requests.

	Port	**SFPState**	The SFP state in percent.
	LUN Disk Storage Pool RAID Group	**Used Capacity**	Amount of Used Capacity
	LUN Disk Storage Pool RAID Group	**Capacity**	Amount of total Capacity
	LUN Disk Storage Pool RAID Group	**Free Capacity**	
	Storage Pool	**OverSubscribe Capacity**	
	LUN Disk Storage Pool RAID Group	**Bandwidth**	Amount of data read/write per sec.

> **Note** There are more metrics that can be collected by these are the main ones to reflect the storage system availability and performance.

3.5.2 File Storage

File storage, also called file-level or file-based storage, stores data in a hierarchical structure. The data is saved in files and folders, and presented to both the system storing it and the system retrieving it in the same format. Data can be accessed using the Network File System (NFS) protocol for Unix or Linux, or the Server Message Block (SMB) protocol for Microsoft Windows.

NFS, originally developed by Sun Microsystems, allows a client to store and view files on a server as if they were on the client computer. All or part of the file system can be mounted on a server, where it is accessible by clients with assigned privileges to a file. SMB uses data packets sent by a client to a server, which responds to the request. Most network-attached storage (NAS) systems support NFS and SMB, which was formally known as the Common Internet File System.

Category	Component/Part type	KPI (Avg, Max, Min)	Description
File Storage	Memory Processor File System	**Current Utilization**	Current Utilization of the system components
	Checkpoint Disk FileSystem Snapshot Storage Pool	**Capacity**	capacity available for the component
	Checkpoint Disk FileSystem Snapshot Storage Pool	**UsedCapacity**	Used capacity for the component
	Checkpoint Disk FileSystem Snapshot Storage Pool	**FreeCapacity**	Free capacity available for the component

	FileSystem	**Read/write Requests**	The amount of read / write requests in data per second
		Read/write Throughput	The amount of read/write throughput in MB/s.
	Data Movers	**DataMoversCalls**	Returns the total number of calls (including bad calls, duplicates, etc.)
	Data Movers	**fsstat**	Returns the number of NFS V2/V3 fsstat calls received by the server

3.5.3 Object Storage

Object storage, also called object-based storage, is an approach to addressing and manipulating data storage as discrete units, called objects. Objects are kept inside a single repository, and are not nested as files inside a folder inside other folders.

Category	Component/Part type	KPI (Avg, Max, Min)	Description
Object Storage Node	Node, Virtual Datacenter	**Availability**	Monitor the SDS system components and Services.
	Node, Virtual Datacenter	**Transaction error rate per node.**	

Performance	Storage Pool Disk Array	**Current Utilization**	Current Utilization of the system components
	Storage Pool Disk Volume	**Capacity**	capacity available for the component
	Storage Pool Disk Volume	**UsedCapacity**	Used capacity for the component
	Storage Pool Disk	**FreeCapacity**	Free capacity available for the component
	Object	**Object Count**	Show the no of Objects used
	Buckets	**Buckets Quota usage**	

3.5.4 Software Defined Storage (SDS)

Software-defined storage (SDS) is a computer program that manages data storage resources and functionality and has no dependencies on the underlying physical storage hardware. It can also be referred to as software products designed to run on commodity server hardware with Intel x86 processors and to enable cost savings over traditional storage area network (SAN) and network-attached storage (NAS) systems that tightly couple software and hardware. Examples of software defined storage includes Dell EMC vFlex (ScaleIO), and Redhat Ceph Storage. Most of the software defined storage can present storage points as object or block, but also it can present file system with some products. For that the metrices that we have list for block, file and object may be also be available in SDS in addition to that; below are the metrics that can be collected from software defined storage systems.

Category	**Component/ Part type**	**KPI (Avg, Max, Min)**	**Description**
SDS	SDS Components and Services	**Availability**	Monitor the SDS system components and Services.
Performance	Storage Pool Drive Volume Array	**Current Utilization**	Current Utilization of the system components
	Storage Pool Drive Volume	**Capacity**	capacity available for the component
	Storage Pool Drive Volume	**UsedCapacity**	Used capacity for the component
	Storage Pool Drive Volume	**FreeCapacity**	Free capacity available for the component

	Storage Pool Drive Volume	**Read/write Requests**	The amount of read / write requests in data per second
	Storage Pool Drive Volume	**Read/write Throughput**	The amount of read/write throughput in MB/s.
	Storage Pool Drive Volume	**Un-Configured capacity**	
	Storage Pool Drive Volume	**Un-Usable Capacity**	RawCapacity - (UsedCapacity+HotSpareCapacity+(AvailableCapacity*2)), Where UsedCapacity is System level statistics field(capacityInUseInKb) converted in GB and AvailableCapacity is System level statistics field(capacityAvailableForVolumeAllocationInKb) converted in GB and HotSpareCapacity is System level "SpareCapacityInKb" converted to GB from ScaleIO REST feed
	Storage Pool Drive Volume	**ThinUsedCapacity & ThickUsedCapacity**	

3.6 Cloud Essential Services

All cloud providers are using software systems for the management and operations of their daily activities, these systems availability can affect the ability of a system admin to act on issues or even get notified.
The following table shows some of the essential cloud services required for monitoring.

Domain	Check	Description
Service Assurance[8]	**Availability**	Monitor the Service Assurance Application Service for Availability. That include Monitoring the Monitoring and Management Solution for the following. • Software Services Availability. • Databases Availability. • Portals and Consoles Availability. • Communication between Internal components.

[8] Service Assurance is the application of policies and processes by a Communications Service Provider (CSP) to ensure that services offered over networks meet a pre-defined service quality level for an optimal subscriber experience

	Connectivity	Within the monitoring solution, components are exchanging information. If the communication between these components breaks that means the monitoring solution won't be able to deliver it's goal. The connectivity between the components need to be monitored. Usually all monitoring solution include a self-monitoring mechanism for that purpose.
	Discovery Monitoring	Discovery of devices and system components need to be monitoring for any change of behavior. For example if a network device or database lost connectivity with the monitoring solution it could mean that the device or system is no longer operating or may indicate a network communication issue. Most monitoring solutions include a self-monitoring of discovered components and shall notify the admin when an anomaly occur. You may also configure the monitoring solution for auto-discovery when there is a new devices or applications added to the topology.

Enterprise Applications	**Notification Systems** Email SMS Ticketing System	Notification systems like Email and SMS. Are critical; most admins relay on them for automating the notification and make it easier to get notified on issues. Monitoring those systems and the integration points between them and the monitoring solution is very important. This can be done by: • Monitoring SMS Gateway: using SNMP, SOAP/REST Request, Script • Monitoring SMTP Email: by using test emails and TCP/IP Check or Process Monitoring of the email Service. • Ticketing System: by using ping, TCP/IP Check or SOAP/Rest check.
Management Systems	**Configuration management databases (CMDB)**	The CMDB holds information about services and systems, it could be in many cases a good source for automatic the discovery of new systems. It should be monitored for availability and connectivity with Service Assurance.
	Orchestrators	The automation jobs are handled by the orchestrator software's, the availability of those software is critical for the overall operation.

	Others	Each organization is different and use different system components, these management components need to be monitored to insure a healthy operation of the cloud system. That may include other systems like CRM & internal portals, HR systems, Billing, Access Systems, inventory management, …

4 Latest Use Cases and Monitoring Challenges

The latest few years showed an evolution in the way applications are developed, applications are now running our lives, and every day there is a new startup that aims to bring new services and applications to the market. For that to happen they need to adapt new technologies like cloud computing, containers, machine learning, NFV and DevOps. In the below section, am going to list some of the recent use cases in the market, where monitoring is required to be looked at in a different way.

4.1 Monitoring Network Function Virtualization (NFV)

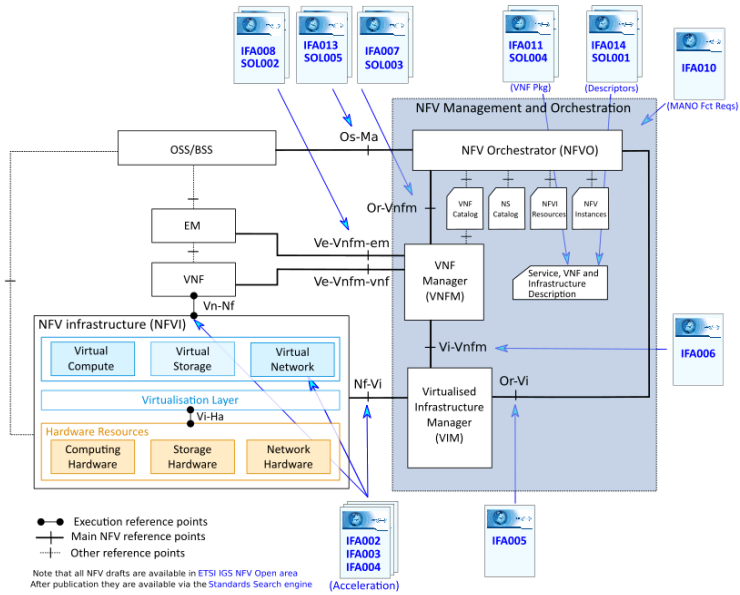
Network functions virtualization (NFV) offers a new way to design, deploy and manage networking services. NFV decouples the network functions, such as network address translation (NAT), firewalling, intrusion detection, domain name service (DNS), and caching, to name a few, from proprietary hardware appliances so they can run in software.

It's designed to consolidate and deliver the networking components needed to support a fully virtualized infrastructure – including virtual servers, storage, and even other networks. It utilizes standard IT virtualization technologies that run on high-volume service, switch and storage hardware to virtualize network functions. It is applicable to any data plane processing or control plane function in both wired and wireless network infrastructures. [9]

While NFV is not only a telecom use case, it used heavy in telecom use cases. And as Telecom loves standards due to the criticality of the systems. ETSI[10] provides the most welcomed and adopted standards for NFV. The standards displayed in the diagram below demonstrate the building blocks of a reliable NFV Solution, from the ground up. It defines what makes a NFV setup, and it introduces standards when it comes to the integration between the different components.

[9] **https://www.sdxcentral.com/networking/nfv/definitions/whats-network-functions-virtualization-nfv/**

[10] The European Telecommunications Standards Institute



According to ETSI, the common ground of the NFV environment is the hardware resources, which is composed on 3 main categories: compute, storage and network. It doesn't define any technology as all these resources will be virtualized in the layer above.

Network Function Virtualization Infrastructure (NFVi)
A Network Function Virtualization Infrastructure (NFVi) is composed of the virtualized components for compute, storage and network. These can be offered from any vendor, and can use any technology that achieve the required performance KPI's.

Example: Redhat openstack platform offers virtualized platform of compute that is using KVM. Virtualized storage that is using Ceph as a software defined storage (SDS), and virtualized network that uses by default open vSwitch (OVS) as software defined network (SDN). Within the platform various combinations of plugins and platforms can be used for SDN, and SDS. For example; you might use Ceph but also you can use Dell EMC XtreamIO as a backend storage, DELL ECS as object storage or HPAR as backend storage.

The same goes for the software defined network; it can be as simple as deploying open vswitch but you might be also using an SDN controller from Juniper or Cisco that manages the openstack NFV cloud cluster networking.

That was one example but vendor offering are a lot; some of them are opinionated and some are open and offers lots of customizations.

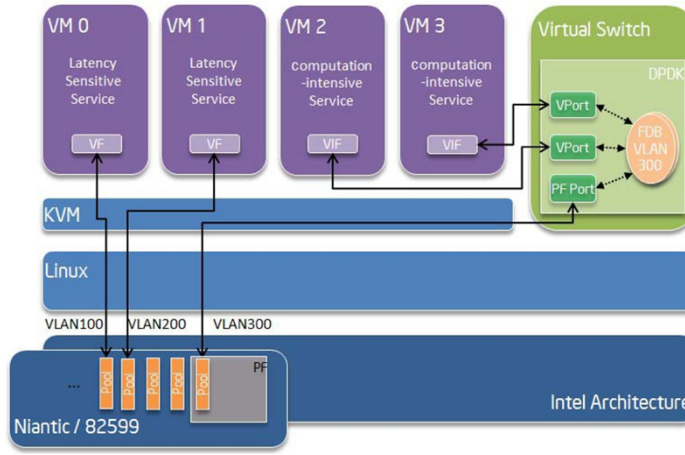
4.1.1 Virtual Infrastructure Manager

The virtual infrastructure manager is the component that manages the lifecycle of the virtual resources. It must offer an open API than can call the different functions to create, updated and delete, and move virtual resources. As per the previous example; that would be the openstack API. For more details on what this API offers please visit the following link. **https://developer.openstack.org/api-guide/quick-start/** the flexibility of the API is what differentiate one product form another.

Virtual network functions are deployed as applications on virtual machines, the virtual machine will be using the virtualized resources but, in many occasions, telecom workloads will need the most accelerated or dedicated resources they can get.

4.1.2 SR-IOV

Some concepts of acceleration are used in that case. As most network applications are demanding for resources and need to eliminate any bottle necks or latency. Concepts like PCI Pass-through might be used to dedicate network ports to virtual machines. And since PCI pass-through will limit the number of virtual machines to be deployed on a single host according to the number of network cards available; SR-IOV came up. SR-IOV stands for Single Root IO Virtualization. It's an implementation of the PCIe specification in a network card. And what that means is you would have a special network card that you can split the physical ports on it into multiple virtual functions; these virtual functions are the ones that will be dedicated to the virtual machine. That will offer the virtual machines dedicated and allows many more virtual machines to be deployed on the same host. The following illustration shows the implementation of SR-IOV in a virtual environment.



4.1.3 DPDK

DPDK is another concept of acceleration is also applied, but it's a software acceleration instead. DPDK is the Data Plane Development Kit that consists of libraries to accelerate packet processing workloads running on a wide variety of CPU architectures. DPDK uses huge pages and processing power to accelerate the move of data, for that we have to consider dedicating some core to the DPDK driver that will do that to keep the separation between the management and workload planes. DPDK can implemented on the hypervisor layer or inside the virtual machine; in the user space.

4.1.4 Smart Nics

SmartNIC goes beyond simple connectivity, and implements network traffic processing on the NIC that would necessarily be performed by the CPU in the case of a foundational NIC. Some vendors' definition of a SmartNIC is focused entirely on the implementation. But this is problematic as different vendors have different architectures and thus a SmartNICs can be ASIC, FPGA, and System-on-a-Chip (SOC) based. Naturally vendors who make just one kind of NIC seem to insist that only the type of NIC they make should qualify as a SmartNIC. [11]

All are only some methods of acceleration, they work in different ways. while acceleration methods provides great deal of flexibility, the performance of one infrastructure may differ from one design to another. Consider DPDK deployed, it needs some CPU Cores to be dedicated to DPDK Poll-mode driver, while SR-IOV limit the capability of moving virtual machines from one node to another unless certain criteria is met. For that when it comes to NFV, awareness of the NFVi design is important to have a successful monitoring platform.

4.1.5 VNF Manager and Orchestrator

The VNFM is a key component of the NFV-MANO that helps standardize the functions of virtual networking and increases interoperability of software-defined networking elements. The VNFM is responsible for the lifecycle management of VNFs under the control of the NFVO, which it achieves by instructing the VIM. VNFM operations include:

- Instantiation of VNFs
- Scaling of VNFs
- Updating and/or upgrading VNFs
- Termination of VNFs

[11] **http://www.mellanox.com/blog/2018/08/defining-smartnic/**

All VNF instances are assumed to have an associated VNF manager. A VNFM may be assigned the management of a single VNF instance or multiple VNF instances. The managed VNFs can be of the same or different types. VNF manager functions are assumed to be generic and can be applied to any VNF. VNFMs are critical for scaling, changing operations, adding new resources, and communicating the states of VNFs to other functional blocks in the ETSI NFV-MANO architecture.

> An example of the importance of a VNFM is key performance indicator (KPI) monitoring. During the lifecycle of a VNF, the VNF management functions may monitor defined KPIs of a VNF. The management functions can use this information for scaling operations. The monitoring plugins can pull data from element management systems or from the virtual infrastructure manager (AKA; OpenStack)/.

Ultimately, the VNFM maintains the virtualized resources that support the VNF functionality without interfering with the logical functions performed by the VNFs. The services provided by the VNFM can be employed by authenticated and properly authorized NFV management and orchestration functions (e.g., functions that manage network services). The orchestrator in this case may manager multiple VNF manager, create service chaining and deploy new services. It may also use the monitoring API available from the virtual infrastructure manager to take decisions to re-deploy an application, scale, move or even destroy.

The main concept behind NFV is to have a self-healing environment where resources and components are monitored and orchestration components are taking decisions to keep the applications and business services up and running.

It's changing the way to monitoring application are deployed and integrated, monitoring platforms need to be aware of the various configurations of the infrastructure specially around networking, it needs to have an API that makes the integration easier with other functional components.

4.2 Monitoring Containers

Monitoring containers is not a mystery, containers expose all metrics through the container's management control plane, there are many software's out there that has packaged a solution for monitoring containers. Containers imposes a lot of challenges that operators are not used to. Traditionally compute resources used to sit in an appliance in the data center, later years after; compute resources become virtualized, containers are the next evolution of compute. Containers offers many features, including easy deployment, cross platform compatibility, and portability. However, in production environments, every group of containers runs as a single service, there are multiple replicas of the same containers running simultaneously. For that; the location of each container is not as important as the service availability.

Monitoring solutions need to keep track on all the changes in the container's environment, and reflect that to operators, so they can make better design decisions.

Everything is a tradeoff in engineering and choosing your monitoring approach is no different. You must consider what aspects are most important, such as real-time log stream, time-series data visualization, off-the-shelf integrations, or flexibility for custom integrations. Ultimately your solution should provide, at a minimum:

- Key container metrics for CPU, Memory, I/O.
- Container orchestration integration such as Kubernetes, DC/OS, or Docker Swarm.
- Off-the-shelf integrations with other infrastructure components.
- Ability to monitor the containers network.
- Offers recommendations on replicas design and capacity planning.

4.3 Artificial Intelligence for operations & autonomous operations (AIOps)

Data analytics refers to the qualitative and quantitative techniques and processes used to enhance productivity and business gain. Data is extracted and categorized to identify and analyze behavioral data and patterns, and techniques vary according to organizational requirements.[12]
For many years since the invention of the computer, data analytics has been used in many ways; mostly prescriptive; to support IT operations.
Gartner introduced the concept of AIOps (originally called Algorithmic IT Operations, now Artificial Intelligence for IT Operations) to describe growing interest and investment in a set of technologies. The definition from Gartner is as follow:

> AIOps platforms utilize big data, modern machine learning and other advanced analytics technologies to directly and indirectly enhance IT operations (monitoring, automation and service desk) functions with proactive, personal and dynamic insight. AIOps platforms enable the concurrent use of multiple data sources, data collection methods, analytical (real-time and deep) technologies, and presentation technologies."

[12] **https://www.techopedia.com/definition/26418/data-analytics**

4.3.1 What are the main types of AI?

Let's break the meaning of the words so we can be specific. AI means the collection of technologies that rely on algorithms and programmatic responses to simulate intelligence, generally with a focus on a specific task.

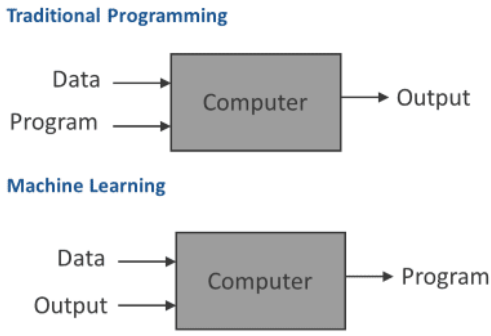
There are 2 types of that; first; "**Narrow AI**", which is the implementation of those algorithms without the computer actually thinking on its own. The other type is "**True AI**"; which is designed to be cognitive, to be aware of context and nuance, and to make decisions that are not programmatic in nature but rather the result of a reasoned analysis.

But in general AI is been in defined in many ways and for many reasons. You will find dozens of definitions online about it.

4.3.2 What is Machine Learning

AI refers to the capability of a machine to imitate human behavior. Machine learning, which evolved from the study of pattern recognition and computational learning theory in AI, explores construction of algorithms that can learn from and make informed decisions.

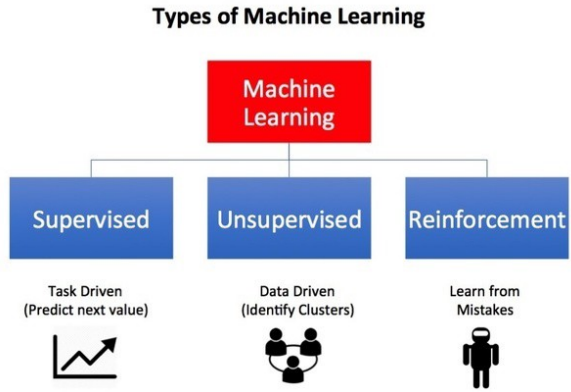
Machine learning needs data. In fact, you could say data is the 'currency' of machine learning.



In traditional programming, a human writes a computer program and provides the data, which the computer processes to create the output. In machine learning, humans provide the data along with the desired output, rules and constraints, and the computer writes the program to deliver this.

There are mainly 3 types of machine learning algorithms:

- Supervised machine learning.
- Unsupervised ml.
- Reinforcement learning.



https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

4.3.2.1 Supervised learning

In a supervised learning model, the algorithm learns on a labeled dataset, providing an answer key that the algorithm can use to evaluate its accuracy on training data.

But, what Netflix[13] does to determine if you will like a certain movie or not? Netflix categorize movies by many tags, there is quite a high chance that a user will be picking up movies of the same nature, which depends on every personal preference.

The business need behind such implementation is to keep subscribers for longer time by recommending movies they might like. save them time thinking or researching what to watch.

IT Ops and NOC teams may use such algorithms to label operations events, classify them based on any criteria, this will help to reduce the number of events that the NOC team receive dramatically. The algorithms need to be selected carefully and decent amount of data cleaning and preparation should be done before we expect meaningful outputs from machine learning.

4.3.2.2 Unsupervised learning

An unsupervised model, in contrast, provides unlabeled data that the algorithm tries to make sense of by extracting features and patterns on its own. a good example to that which can be used in IT operations.

---

[13] https://becominghuman.ai/how-netflix-uses-ai-and-machine-learning-a087614630fe

Operation data is mostly time series, we can use un supervised machine learning algorithms to do anomaly detection and Correlation, while the program doesn't know anything about data itself. it will be able to detect changes in data behavior, which is referred to as anomaly detection. or do correlation and try to discover hidden relationships between data points. this will help with events and metrics correlation, and aids root cause analysis, with an autonomous correlation model.

4.3.2.3 Semi supervised learning

Semi-supervised learning takes a middle ground. It uses a small amount of labeled data bolstering a larger set of unlabeled data.

4.3.2.4 Reinforcement learning

Reinforcement Learning (RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

As compared to unsupervised learning, reinforcement learning is different in terms of goals. While the goal in unsupervised learning is to find similarities and differences between data points, in reinforcement learning the goal is to find a suitable action model that would maximize the total cumulative reward of the agent.

CSP can use these algorithms, to train AI agents to work like IT & network admins, with more API driven apps this will soon become a possibility, this will save enormous time working on repetitive tasks, and divert the focus from operations to service management.

4.3.2.5 Deep learning

Deep learning is actually a subset of machine learning. It technically is machine learning and functions in the same way but it has different capabilities.

The main difference between deep and machine learning is, machine learning models become better progressively but the model still needs some guidance. If a machine learning model returns an inaccurate prediction then the programmer needs to fix that problem explicitly but in the case of deep learning, the model does it by himself. Self-driving car systems are good example of deep learning.[14]

[14] **https://www.geeksforgeeks.org/artificial-intelligence-vs-machine-learning-vs-deep-learning/**

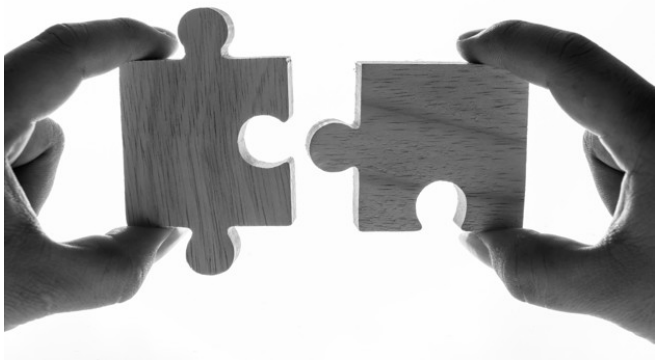
5 General Requirements of any monitoring solution

Consider the following 12 features before buying or designing a monitoring platform for your cloud.

1. Easy to deploy.
2. Scalable, preferably running on containers.
3. Highly available.
4. Capable of correlating events cross IT and network domains.
5. Able to collect all basic KPI's in addition to custom created KPI's from all IT and Network resources (compute, network, storage, and *CUSTOM*)
6. Able to create formulas and equation on collected KPI's and present it to the dashboard.
7. Multi-tenant support.
8. Multi-cloud platform support.

9. On premise and/or public cloud deployment model.
10. Provide deep fault management and insights into the network performance (LAN, WAN, SD-WAN, Optical, etc.)
11. Extensible through a REST API for pushing or receiving metrics and events.
12. Utilize AI/ML when possible to understand the behavior of devices and applications or make automated decisions.

6 Conclusion



In this book, I have explained the main differences between multiple cloud offerings. Discussed some real use cases. And defined the scope of monitoring infrastructure as a service. By breaking down a cloud system into multiple components and exploring the possibilities of monitoring each part.

Note that every system component might be coming from a different vendor and may need a different model to monitor and reports events and performance. The monitoring solution need to be aware of the different possibilities and be should cover a wide variety of technologies and systems without spending too much time into the implementation.

I hope that the information presented was helpful and sufficient to start evaluating a proposed monitoring solution or even design your own based on specific requirements, based on your own IT operations model, and the required visibility to operate successfully toward business goals.

The scope of this book was to define what it means to operate an infrastructure as a service cloud model, other components like application monitoring was not covered here, as according to this model cloud users are responsible for this. However, cloud service providers may offer additional add-on services to help customer manage their applications.

Monitoring is a practice, it needs to be included as a process when designing, deploying or operating IT, network and application systems.

IT & network systems need an effective monitoring solution, that is capable of providing the right information at the right time to multiple decision makers to help in maintainig service assurance, enhance information and data services.

What this book is useful for ?

1. Explore the requirements for monitoring solutions to guarantee IT & network service assurance for infrastructure as a service (IaaS) cloud systems

2. if you need to understand how a cloud solution work, what are the main components, and how these components report performance.

3. Explain how an IT & network operations data set may contain; if you are a data scientist who wants to explore AI/ML in IT operations domain.

4. if you are administrating or managing an IT cloud and need to know what performance KPIs you should consider or collect.

check linkedin profile

About the author

Mohamed ELMesseiry, data scientist, 17 years of solution consultancy experience. subject domain expert in IT & Network operations, system integration, public/private cloud systems, IT automation, & CI/CD, software development, SD-WAN, artificial intelligence operations (AIOPS).

"this book is for you, we will dig deep into IT/Network cloud operations & service assurance without having to read 1000 pages, its short, it definitive, its comprehensive and right to the point"