Comparing Infrastructure Monitoring with CloudStack Compute Services for Cloud Computing Systems

Aparna Datt[1], Anita Goel[2], and Suresh Chand Gupta[3]

[1] Department of Computer Science, University of Delhi, India
[2] Department of Computer Science, Dyal Singh College, University of Delhi, India
[3] Visiting Faculty, Department of Computer Science and Engineering, IIT Delhi, India
{aparna.datt,goel.anita,gupta.drsc}@gmail.com

Abstract. CloudStack is an open source IaaS cloud that provides compute, network and storage services to the users. Efficient management of available resources in the cloud is required in order to improve resource utilization and offer predictable performance to the customers. To facilitate providing of better quality of service, high availability and good performance; a comprehensive, reliable, centralized and accurate monitoring system is required. For this, the data needs to be collected from the components of CloudStack and analyzed in an efficient manner. In this paper, we present a detailed list of attributes required for monitoring the infrastructure associated with CloudStack. We identify the processes related with the compute services and its associated parameters that need to be monitored. We categorize the infrastructure monitoring, and list the parameters for monitoring parameters. Further, the proposed list is applied to three monitoring software that are commonly used for monitoring resources and processes associated with CloudStack. Developers and system administrators can benefit from this list while selecting the monitoring software for their system. The list is useful during the development of new monitoring software for CloudStack, as the functionality to be monitored can be selected from the list.

1 Introduction

CloudStack is an open source, open standards, multi-tenant cloud orchestration platform. It is scalable, flexible, secure and hypervisor agnostic. A service offering is a set of virtual resources, like, CPU, memory and disk. Compute and storage resources are provisioned to the users in form of service offerings. The system administrator can define different type of offerings as per the user requirements. A compute service offering includes guest CPU, guest RAM, guest networking type (virtual or direct) and tags on the root disk [14]. Along with the resources, a service offering also involves features related to resource metering, usage and charges for usage. A user can select from the available offerings while creating a virtual machine (VM). Once provisioned, CloudStack enables users with virtual machine management services that include starting, stopping, restarting and destroying virtual machines. Management of associated resources, like, CPU, network and storage and their utilization is also a part of VM management.

W. Chu et al. (Eds.): DNIS 2015, LNCS 8999, pp. 195–212, 2015.

Using virtual machines in an organization is a cost effective mechanism for management of fluctuating system workloads. With increase in demand for the compute services, size of the resources for enabling these services is also increasing. An efficient and accurate monitoring system is required to monitor resource utilization and status of the system. Monitoring of compute services basically includes monitoring the available physical infrastructure, like, CPU, memory and disk and the associated processes. The performance of infrastructure resources and allocation of these resources in form of the virtual machines needs to be measured for complete system performance.

The process of infrastructure monitoring involves collecting information about the system, and analyzing it to understand performance levels of the systems. Data required for monitoring the system resources and available services is collected at multiple points in the CloudStack software. The monitoring software collect the data, analyzes it and presents it in an understandable format, like, graphs, charts etc. for the users. Some of the popular open-source monitoring software used with CloudStack are- CollectD, Nagios and Zenoss Zenpack. These software target specific aspects of infrastructure monitoring, like, resource usage, network statistics etc. None of the available monitoring software target at the comprehensive monitoring of CloudStack.

Also, in the requirement analysis phase there is a need for designers of the system to have a detailed knowledge of the information to be monitored. While the system is running, the system administrator needs to monitor the resources and performance of the system. Instead of identifying the monitoring information at different levels by different users of the system, it would be of use if a comprehensive list of resources and processes to be monitored is available to the users. It would make it convenient to the users to select the functionality they desire to monitor from the given list.

In this paper, the aim is to identify essential features of CloudStack compute that need to be measured for infrastructure monitoring. The infrastructure functionality that needs to be monitored for CloudStack compute can be divided under four main criteria. We present a categorically distributed list of attributes relevant to compute infrastructure in CloudStack that need to be monitored for efficient functioning of the system. We have listed attributes under the following headings-

- Compute infrastructure
- Compute usage data
- OS process utilization data
- Background processes

A detailed study of CloudStack, its physical and logical components and the associated API's lead to the creation of this list [15]. This list will provide the system administrators and system developers an insight into the data that needs to be collected from the system for providing exhaustive system monitoring information. The users can use this list and make selection of required monitoring functionality. Also, third party monitoring software developers can refer to the list while updating the existing monitoring software and also during the development of new monitoring solutions for CloudStack. The attribute list presented here is a comprehensive list inclusive of all the main areas of compute monitoring in the current system and can be updated as and when new compute functionality is added to the system. Instead of generating their individual monitoring list, various users associated with the system can select the attributes from the given list.

A case study based on three popular third party monitoring tools used with CloudStack is performed. We have applied our list to CollectD [17], Nagios [18] and Zenoss Zenpack [16] to find the extent and kind of monitoring functionality provided by these infrastructure monitoring software for CloudStack.

In this paper, Section 2 provides an overview of CloudStack. Section 3 describes the structure of infrastructure monitoring along with the functionality lists. In section 4, we describe the case study for identifying functionality in existing infrastructure monitoring software. Section 5 discuses advantages of functionality list. The related work is discussed in Section 6. Section 7 concludes the paper.

2 An Overview of CloudStack

CloudStack is an open source IaaS cloud platform governed by the Apache Software Foundation. It is used for building public, private and hybrid IaaS clouds. Cloud infrastructure basically comprises of resources required for providing compute, storage and network services. On-demand, elastic cloud computing services can be provisioned using CloudStack. Service providers can sell self-service virtual machine instances, storage volumes, and networking configurations over the Internet [13]. CloudStack is a refined platform that has in-built features, like, massive scalability, high availability and centralized management, which allows single point management of vast number of physically distributed resources. CloudStack software works with a variety of hypervisors, like, VMware, Oracle VM, KVM and XenServer.

CloudStack manages its vast number of geographically distributed servers effectively. A centralized management server takes care of the infrastructure and the processes. The system is designed in a fault tolerant manner and failure of a single component does not cause the system to crash. Management server can be serviced periodically without causing any affect on the virtual machines running in the cloud. Customizable graphical user interface is available for administrators for provisioning and managing the cloud and for users for running VMs and managing VM templates. APIs are available that provide access to the management features in the GUI.

The process of VM provisioning in CloudStack is a well-defined and step-wise process. The steps for provisioning of a VM are as follows-

- Request for an instance is placed by the user.
- Operating System (Windows, Linux), Compute Offering (CPU & RAM), Disk Offering (Volume Size), Network Offering are selected.
- Instance template is copied from secondary storage to primary storage on the cluster in which the machine is to be provisioned .
- Data volume is created on primary storage for the cluster.
- An instance is created.

Once created, a VM can be started, stopped, restarted and destroyed. Service offerings of a VM can also be changed. The status of a VM for the resources it is utilizing can also be monitored within a CloudStack system.

2.1 Architectural View of CloudStack

The architectural design of CloudStack comprises of two main components, namely, the physical component which includes the cloud infrastructure and the logical component that comprises of the management Server.

The physical infrastructure of a CloudStack implementation consists of the following [14]-

- **Host:** A compute node in a cluster is called a host. The guest virtual machines created by the users are provisioned and run on a host. It is a single computer and contains resources, like, CPU, memory, storage and networking required to run a VM. With every host a hypervisor is associated for management of VMs. Multiple hosts can be interconnected via TCP/IP network.
- **Cluster:** A combination of a set of hosts and primary storage is a cluster. The hosts in a cluster are of the same kind, i.e. all hosts in a cluster have identical hardware, run the same hypervisor, are on the same subnet, and access the same shared primary storage. Virtual machine instances can be live-migrated from one host to another within the same cluster, without interrupting service to the user.
- **Pod:** A set of one or more clusters along with a switch form a pod. These are the second largest unit in the cloud infrastructure. They are invisible to users.
- **Zone:** A single or multiple pods and secondary storage combine to form a zone. Logically a zone can be compared to a single datacenter. It is the largest infrastructural unit within the CloudStack setup and aims at providing physical isolation. An end user can see the available zones in a cloud setup and has to select a zone at the time of provisioning a VM. There can be public and private zones in a cloud. Zones specific to a particular domain are private and the zones where any user can create a guest VM are public.
- **Primary storage:** It is associated with every cluster. The disk volumes for all the VMs running on hosts in a cluster are stored in the primary storage.
- **Secondary storage:** It is the storage associated with a zone. It is used for storing templates, ISO images and disk volume snapshots.

When a cloud is set up using CloudStack, storage, IP addresses and hosts are to be provisioned. For managing the resources, zones and hosts; a software component is required. Management Server is the software component of CloudStack that is used for management of the cloud resources. It has a user interface and a set of APIs that are used to configure and manage the hosted infrastructure. The management server is stateless and can be run on a server or a VM. The management server is used for managing VM provisioning, associated storage and networking on hosts. Management of snapshots, templates, ISO images and the storage is also handled by the management server. There can be a single or multiple management servers in a cloud environment. Single management server can manage multiple zones having large number of hosts. A multiple management server deployment in form of a cluster is beneficial in cases where scalability and redundancy is required.

3 Related Work

Cloud computing with its increasing complexity and vast set of resources requires monitoring as an important component. Monitoring of infrastructure and services associated with the cloud provide an in depth information on the health and performance of the system under consideration. Performance monitoring in the clouds is a focal area of research these days. Multiple monitoring software exist for infrastructure monitoring of the cloud.

There are not many monitoring systems that satisfy all the cloud administrator requirements which imposes that clouds need to be monitored by a set of monitoring systems [3].Cloud monitoring systems need to be advanced and customized to the diversity, scalability, and high dynamic cloud environment [1].

Montes et al. [10] propose a unified cloud monitoring taxonomy, based on which they define a layered cloud monitoring architecture. They implement GMonE, a general-purpose cloud monitoring tool which covers all aspects of cloud monitoring by specifically addressing the needs of modern cloud infrastructures. Meng et al. [7] propose a MaaS framework that achieves significant lower monitoring cost, higher scalability, and better multi tenancy performance. Kai et al.[5] present SCM monitoring system that collects accurate metrics from both physical and virtual resources, including the main components of Apache CloudStack (system virtual machines, Secondary Storage, Primary Storage and management servers), and makes these data easily accessible and human readable, which is quite friendly to the CloudStack users. Bellavista [2] present a novel framework for Easy Monitoring and Management of IaaS (EMMI) solutions. McGilvary et al. [6] propose Cloudlet Control and Management System (C2MS); a system for monitoring and controlling dynamic groups of physical or virtual servers within cloud infrastructures. This system allows administrators to monitor group and individual server metrics on large-scale dynamic cloud infrastructures where roles of servers may change frequently.

Monitoring solutions based on specific characteristics are also being developed. Moldovan [8], in their paper introduce MELA, a customizable framework, that enables service providers and developers to analyze cross-layered, multi-level elasticity of cloud services, from the whole cloud service to service units, based on service structure dependencies. Morariu [11] presents the design of a monitoring solution that integrates several open source tools and can assure QoS for private clouds. He et al. [4] in their paper present a novel security monitoring framework for intrusion detection in IaaS cloud infrastructures. The framework uses statistical anomaly detection techniques over data monitored both inside and outside each Virtual Machine instance. Zhang [12], presents a SLA-driven state monitoring framework for cloud service based on the matrix factorization model that contributes to improving the monitoring accuracy, and also takes less overhead during the communication.

Distributed Architecture for Resource manaGement and mOnitoring in cloudS (DARGOS), a completely distributed and highly efficient Cloud monitoring architecture to disseminate resource monitoring information has been proposed by Molina[9]. DARGOS is flexible and adaptable in nature and ensures an accurate measurement of physical and virtual resources in the cloud keeping at the same time a low overhead and enabling cloud administrators to design better cloud provisioning strategies and to avoid SLA violations.

Our study and research did not reveal any academic research study on infrastructure monitoring list for compute services in CloudStack. With the availability of this list, the designers of the system, the system administrators and the third party monitoring tool developers, all will gain insight about the features that require monitoring in CloudStack compute. The list will help them in choosing the functionality they need to monitor in the system.

4 Structure of Infrastructure Monitoring

The physical and logical infrastructure of the CloudStack setup needs to be monitored by the system administrator. Infrastructure monitoring of the system enables the system administrator to keep an updated record of the systems performance and resource utilization. This information is beneficial to the administrator as it helps to keep a check on the health and status of the system. Features, like, live migration, high availability, failure detection and scalability can be provided efficiently if complete information about total resources, resources currently under utilization and the quantity of free resources is available. The monitoring process can also enable the administrator to keep a check on resources that are running low or are underutilized. Usage trends can also be predicted by this information. System administrators can also determine information about the most preferred compute offerings by different kinds of users, size of storage users want to associate with specific offerings etc.

In this paper, we define various aspects of infrastructure monitoring keeping in mind the system administrator's perspective. We have defined the infrastructure monitoring list after studying the CloudStack architecture, infrastructure and API's [9]. We provide the infrastructure monitoring list under following four main headers:

- *Compute infrastructure list* for monitoring all attributes associated with compute offerings.
- *Compute infrastructure usage list* for assessing resource usage.
- *Background processes list* to keep track of processes running at the backend.
- *OS Process usage list* to track usage of CPU, network and memory.

4.1 Compute Infrastructure List

Compute infrastructure of CloudStack consists of the physical resources that can be virtualized. Based on the varying demands of the users, the infrastructure can auto scale. Virtual machine provisioning is the main service provided in compute offerings. Infrastructure monitoring of compute resources facilitates the system administrator to keep a track of resources being consumed and the change in workloads of the system. This information enables the system administrator in load balancing the system. Also in case, a host fails within or outside a cluster, when VM migration is required, the knowledge of current status of the system is very important. Features like high availability and scalability also require system information to be executed efficiently.

At every level of CloudStack infrastructure, there are a specific set of attributes that provide detailed information about that particular level. Attributes at Zone level are identified as id, name and description of zone. For pods and clusters also similar kind of information is maintained. The attribute list for the hosts contains host details and supported hypervisor's details. For projects and VMs, the status, resources allocated, start/ stop dates and other information is maintained.

Table 1 provides the attribute list for compute infrastructure.

Table 1. Compute infrastructure list

Levels	Parameters	Description
Zones	Id, Name, Network type, Zone token, Description, Display text	Attributes required to identify a zone.
	Zone dedicated / not dedicated	Zone is dedicated for a particular task or not.
	Domain, Domain id, Domain name	Details of domain related to a zone.
	Local storage enabled	true if local storage offering enabled, false otherwise
	Security groups enabled	true if security groups support is enabled, false otherwise
	Pod id, Pod name	Details of the pods associated with the zone
	Cluster id, Cluster name	Details of the clusters associated with the zone.
	Resources (Account, Domain, Customer, key, project) , resource id , resource type, resource details	Meta data associated with the zone (key/value pairs)
Pods	Id, Name	Attributes defining the information required to identify a Pod.
	IP(Start/ End), Gateway, Net mask	Network information with a pod.
	Zone id, Zone name	Details of the zone related to a pod.
	Cluster Id, Cluster name	Details of the cluster related to a pod.
Cluster	Id, Name, Cluster type	Details of attributes required to identify a Cluster.
	Hypervisor type	Type of hypervisor with a cluster.
	Managed state	Whether this cluster is managed by CloudStack.
	Memory over commit ratio	Memory over commit ratio of cluster
	Pod id, Pod name	Pod related to a cluster.
	Zone id, Zone name	Zone related to a cluster.
Domain	Id, name, level	Details of attributes required to identify a domain.

Table 1. (*continued*)

	Has child	Whether the domain has one or more sub-domains
	Network domain	Network information with a domain.
	Parent domain (id, name)	Details of the parent domain, in case the domain under consideration is a child domain.
	Path	Path of the domain
Host	Id, name, State (connected/ disconnected), Removed, HA, type, version, created, capabilities	Attributes defining the information required to identify a host.
	Cluster (Id, name, type)	Cluster information related to a host.
	Hypervisor, Hypervisor version	The host hypervisor
	Management server id	Management server ID of the host
	Network kbs read, Network kbs write	Network information of host.
	Os category id, Os category name	The OS category ID of host, the OS category name of host
	Pod id, Pod name	Pod related to the host.
	Zone id, zone name	Details of zone related to the host.
	Job id, Job status	ID and status of latest asynchronous job acting on this object.
Account	Account (Id, details, type, Name, State)	Attributes defining the information required to identify an account.
	Default zone id	Default zone of the account
	Domain(name, id)	Domain related to account.
	Groups	List of groups that account belongs to.
	Project (available, limit, total)	Projects associated with an account.
	Received bytes, Sent bytes	Network traffic information of the account.
	Snapshot (available/ limit/ total)	The total number of snapshots available, can be stored and are currently stored for this account. stored by this account
	Template (available/ limit/ total)	The total number of templates available to be created, available for creation, have been created by this account.
	Vm (available, limit, total, running, stopped)	The total number of virtual machines available, can be deployed, currently running and stopped for this account.
	user(id, account, account id, account type, created, domain, domain id, user details)	List of users associated with account.
Project	Id, Display text, Name, State	Attributes defining the information required to identify a project.
	Account	Account name of the project's owner
	Domain (name, id)	Domain information of a project.

Table 1. (*continued*)

	Snapshot (available, limit, total)	The total number of snapshots available, stored and can be stored for this project.
	Template (available, limit, total)	Total number of templates available to be created, can be created and have been created by this project.
	VM (available, limit, running, stopped, total)	Total number of virtual machines available, can be deployed, currently running and stopped for this account.
	Resource (id, type)	Id and type of resource.
VM	Id, Account, Created, Details, display name, Display vm, state	Details of attributes required to identify a virtual machine.
	Domain, Domain id	The domain information with a VM.
	Group (name, id)	The group information with a VM.
	Guest os id	Os type ID of the virtual machine
	High availability enable, scalable	True if high-availability is enabled, false otherwise
	Hosted, Hostname	Host information with a VM.
	Hypervisor	Hypervisor on which template runs
	Memory, root volume	Memory allocated for a VM
	Project (name, id)	Project information with a VM.
	Service offering (id, name)	Details of service offering with a VM.
	Resource id, resource type	Resources associated with a VM.
	Template display text	Alternate display text of the template for VM
	Template (id, name)	Template information with a VM.
	Zone (id, name)	Zone information with a VM.

4.2 Compute Infrastructure Usage List

The information regarding the usage of compute infrastructure requires consistent monitoring. The current state of zones, pods, clusters, status of resources (CPU, memory and network) currently in use on hosts, number of VMs on the host etc. need to be monitored. Monitoring of the data usage helps to identify the total resource usage, quantity of available resources and current load on the system at all different levels. This information provides a detail on the usage aspect of the system.

Compute usage list contains attributes required by the administrator for monitoring the usage of the system. Information, like, host uptime, number of network read/writes, available services on each level, list of compute offerings etc. is maintained. This information provides the administrator with exact state of a system at a given point of time. Administrator can assess the utilization levels of the system, trending resources and need for more resources in the system. Table 2 displays the list for compute usage attributes.

Table 2. Compute infrastructure usage list

Level	Parameter	Description
Zones	Capacity (total, used, percent)	Total, in use and used percentage capacity of zone.
	Allocation state	Allocation state of the cluster.
Pods	Capacity (total, used, percent, type)	Total, in use, used percentage and type of capacity of Pod.
	Allocation state	Allocation state of Pod.
Cluster	Allocation state	Allocation state of cluster.
	CPU over commit ratio	CPU over commit ratio of cluster.
	Capacity (total, used, percent, type)	Total, in use, used percentage and type of capacity of the Pod.
Host	Average load	Average CPU load on the host.
	Events	Events available for host.
	Enough capacity	True if this host has enough CPU and RAM capacity to migrate a VM to it, false otherwise.
	local storage active	True if local storage is active, false otherwise.
	Last pinged	The date and time the host was last pinged.
	Resource state	Resource state of the host.
	Suitable for migration	True if this host is suitable (has enough capacity and satisfies all conditions) to migrate a VM to it, false otherwise.
Account	IP(available, limit, total)	The network information associated with the account.
	Primary storage (available, limit, total)	Total primary storage space (in GiB) available to be used, can be owned and is owned for this account.
	Secondary storage (available, limit, total)	Total secondary storage space (in GiB) available, can be owned and is owned for this account.
	Volume (available, limit, total)	Total volume available, being used and can be used for this account.
Project	Primary storage (available, limit, total)	Total primary storage space (in GiB) available to be used, can be owned, and owned for this project.
	Secondary storage (available, limit, total)	Total secondary storage space (in GiB) available to be used, can be owned, and owned for this project.
	Volume (available, limit, total)	Total volume available for this project, can be used and currently being used by this project.

4.3 OS Process Utilization Data List

Within each level of CloudStack compute, there are three standard OS processes that run to facilitate the smooth functioning of compute services. OS processes utilization data list provides the parameters that need to be monitored for measuring utilization for each of these. Utilization of CPU, memory and network is measured at host, account, project and VM level. At all levels, CPU, network and memory utilization data can be monitored for total, allocated and used options.
Table 3 presents the OS processes utilization list.

Table 3. OS process utilization list

Level	Process	Details
Host	CPU utilization	CPU (allocated, number, sockets, speed, used)
	Memory utilization	Memory (allocated, total, used)
	Network utilization	Network kbs (read, write)
Account	CPU utilization	CPU (available, limit, total)
	Memory utilization	Memory (available, limit, total)
	Network utilization	Network (available, domain, limit, total)
Project	CPU utilization	CPU (available, limit, total)
	Memory utilization	Memory (available, limit, total)
	Network utilization	Network (available, limit, total)
VM	CPU utilization	CPU(number, speed, used)
	Memory utilization	Disk – I/O (read/write), kbs (read/write), offering (id, name)
	Network utilization	Network kbs(read, write)

4.4 Background Process Functionality List

For listing the functionality and the parameters to be monitored for background processes in CloudStack compute, we have defined the background processes functionality list. This list can be referred by the system administrator for monitoring background processes and can also be utilized at the time of developing monitoring component for CloudStack.

Background processes are carried out by the management server, console proxy VM and secondary storage VM. Information required for monitoring running processes, like, resource provisioning, snapshot management, template management, zone management, orchestration, high availability, live migration, connection management and secondary storage management can be collected from the system. Parameters to be monitored within each process are identified and listed.

Management of provisioning of resources, snapshot, zone and template management are the basic processes that run in the management server to facilitate the IaaS services provided by CloudStack. Orchestration is an important process that runs in the management server. High availability and live migration are two processes that make CloudStack services more efficient and reliable.

Connection management of a VM to display VMs console on the management server web interface for remote management is managed by console proxy VM. Secondary storage VM runs secondary storage management process to facilitate snapshots, templates, volumes and ISO files management in and out of the cloud environment in a secure manner. Table 4 provides a list of background processes that need to be monitored.

Table 4. Background process list

Process	Details	Component
Resource provisioning	VM allocation to host, storage devices, IP addresses.	Management Server
Snapshot management	Full snapshots, incremental snapshots.	Management Server
Template management	Volume templates.	Management Server
Zone management	Management of zones.	Management Server
Orchestration	VM starts, snapshot copies, template propagation.	Management Server
High Availability	VMs, hosts, primary storage, secondary storage.	Management Server
Live Migration	Migration of a running VM. Stopped VM’s can’t be migrated.	Management Server
Connection management	Facilitates the secure connection to individual VM's through their respective hypervisor hosts to display the VMs console on the management server web interface for remote management.	Console Proxy VM
Secondary storage management	Facilitates moving snapshots, templates, volumes and ISO files in and out of the cloud environment in a secure manner. Connects to individual hypervisors to instruct them on how to mount secondary storage as well as how to store and cleanup snapshots and templates.	Secondary storage VM

5 Case Study

As a case study, we have applied these lists on three popular open source monitoring software generally used with CloudStack, namely, Zenoss Zenpack, CollectD and Nagios. This case study will enable us to identifying the kind of infrastructure monitoring provided by these tools for CloudStack. Comparison of our lists with the information monitored by these software will also help to identify the extent of monitoring depth provided by the tools. Also the case study can be used to identify the areas of CloudStack that are not being monitored effectively.

Zenoss ZenPack [16] is an open source cloud monitoring software that provides monitoring support for CloudStack. Zenoss allows system administrators to monitor some of the important aspects of CloudStack like availability, inventory, configuration, performance, and events related to the system. The key metrics provided by zenoss are for CPU and memory tracking. These metrics are provided within Zenoss at all the levels of granularity. Alerts regarding low and high thresholds of the resources are also available.

CollectD [17] is a daemon which collects information about zones, pods, clusters, storage and hosts. Information about the available and resources being used currently by the system are collected by CollectD. System performance statistics are also gathered periodically and can be stored in a variety of ways. Information to be monitored is collected in form of system metrics relating to memory, CPU, secondary storage etc. and can be used to measure the current status of system.

Nagios[18] is a popular open source cloud monitoring software. It monitors usage of available resources like CPU, memory, etc. Cloud capacity including memory, storage, private and public ip's is also monitored. Information regarding state and status of VM's is monitored. Checks for usage of various resources like memory, CPU, network and disk usage within a VM are also carried out.

Table 5 presents the list detailing comparison of monitoring of background processes by the three monitoring software under consideration. Table 6 combines the list of comparison of three selected monitoring software for compute infrastructure attributes and compute usage. OS processes Utilization data of infrastructure monitoring is displayed in Table 7. In the tables, notation '√' denotes that parameter is supported by the monitoring software; 'x' denotes parameter not supported.

Table 5. Background process case study

Process	Zenoss	CollectD	Nagios
Resource provisioning	√	√	√
Snapshot management	×	×	×
Template management	×	×	×
Zone management	×	×	×
Orchestration	×	×	×
High Availability	×	×	×
Live Migration	×	×	×
Connection management	×	×	×
Secondary storage management	×	×	×

Table 6. Compute infrastructure and usage case study list

Levels	**Infrastructure Parameters**	**Usage Parameter**	Zenoss	Zenoss	CollectD	CollectD	Nagios	Nagios
Zones	Details	Capacity (total, used)	×	×	√	√	√	√
	Resources , resource id , resource type, resource details	Percentage used	×	×	√	√	√	√
	List of Domains	Allocation state	×	×	×	×	×	×
	List of Pods		×	×	×	×	×	×
	List of Clusters		×	×	×	×	×	×
Pods	Details	Capacity (total, used)	×	×	√	√	√	√
	Network Details	Percentage used	×	×	√	×	√	√
	Zone Details	Allocation state	×	×	√	×	×	×
	List of Clusters		×	×	√	×	×	×
Cluster	Details	Type	×	×	√	×	×	×
	Hypervisor type	Capacity (total, used)	×	×	×	√	×	√
	Managed state	Percentage used	×	×	×	×	×	×
	Pod details	Allocation state	×	×	√	×	×	×
	Zone details		×	×	√	×	×	×
Domain	Details		×	×	√	×	√	×
Host	Id, name, State (connected/ disconnected), Removed, HA, type, version, created, capabilities	Average load	×	×	√	×	√	×
	Cluster (Id, name, type)	CPU(allocated, number, used, speed)	×	×	√	√	√	√
	Hypervisor, Hypervisor version	Capacity	×	×	×	√	×	×
	Management server id	Last pinged	√	×	√	×	×	×

Table 6. (*continued*)

	Network details	Resource state	×	×	√	√	√	√
	Pod details	Suitable for migration	×	×	√	√	√	×
	Zone details		×	×	√	×	×	×
Account	Account details	IP(available, limit, total)	×	×	√	√	×	√
	Domain details	Memory (available, limit, total)	×	×	√	√	×	√
	Network details	Network (available, domain, limit, total)	×	×	×	√	×	√
	Project details	Primary storage (available, limit, total)	×	×	×	√	×	√
	Snapshot details	Secondary storage (available, limit, total)	×	×	×	√	×	√
	Template information	Volume (available, limit, total)	×	×	×	√	×	√
	VM (available, limit, total, running, stopped)		×	×	×		√	
	User information		×	×	×	×	×	×
Project	Id, Display text, Name, State	CPU (available, limit, total)	×	×	√	√	√	√
	Domain (name, id)	Memory (available, limit, total)	√	√	√	√	√	√
	Snapshot (available, limit, total)	Network (available, limit, total)	×	√	×	√	×	√

Table 6. (*continued*)

	Vm (available, limit, running, stopped, total)	Secondary storage (available, limit, total)	√	√	×	√	×	√
	Resource (id, type)	Volume (available, limit, total)	×	√	×	√	×	√
		Vpc (available, limit, total)	×	√	×		×	√
VM	Details	Status	√	√	√	√	√	√
	Domain, Domain id	Memory, root volume	√	√	×	√	×	×
	Guest os id	Network (kbsread, kbswrite)	×	√	×	√	×	×
	High availabilty enable, scalable	Resource usage	×	√	√	√	×	×
	Hosted, Hostname		√	×	×	×	×	×
	Hypervisor			×	×	×	×	×
	Project (name, id)		√	×	×	×	×	×
	Service offering (id, name)		√	×	√	×	√	×
	Resource id, resource type		√	×	×	×	√	×
	Template (id, name)		×	×	×	×	×	×
	Zone (id, name)		√	×	×	×	×	×

The case study carried out above indicates that the major aspect of monitoring covered by all the three monitoring software is of compute infrastructure and its usage. Except for resource provisioning, monitoring support for other background processes is not there in any of the software. Our study reveals that the maximum monitoring support for OS processes is provided by Zenoss Zenpack. The common parameters monitored by all the monitoring software are the status of total resources, resources currently in use, available resources and metrics about resources at different levels of granularity.

Table 7. OS processes case study list

Level	Process	Zenoss	CollectD	Nagios
Host	Cpu utilization	√	√	×
	Memory utilization	√	√	×
	Network utilization	√	√	×
Account	Cpu utilization	√	×	×
	Memory utilization	√	×	×
	Network utilization	√	×	×
Project	Cpu utilization	√	×	×
	Memory utilization	√	×	×
	Network utilization	√	×	×
VM	Cpu utilization	√	√	√
	Memory utilization	√	√	√
	Network utilization	√	√	√

6 Benefits of Functionality List

The functionality list presented in our paper is exhaustive for listing monitoring functionality in CloudStack. We have arrived at the functionality list after performing a study of CloudStack architecture and API's. The functionality list is beneficial to the system administrators and the designers of the system. The list makes the task of monitoring easier for the system administrator as the desired features to be monitored can be selected form the defined list. Monitoring tools specific to monitoring of CloudStack compute services can also be developed by referring to this list Additions to the existing monitoring software can also be made to increase their monitoring support towards CloudStack compute. On the basis of case study presented above, it is evident that none of the existing monitoring tools available provide a complete monitoring coverage of all features of CloudStack compute. All the basic set of infrastructure, associated parameters and the processes that need to be monitored consistently for maintaining compute system efficiently are provided in this functionality list.

7 Conclusion

In our paper, we have presented a list defining the attributes and processes and their parameters of infrastructure that need to be monitored in a CloudStack compute system. It facilitates designers in specifying monitoring requirements during requirement specification phase. Also, the system administrator can use the list for selecting the functionality that needs to be monitored in the software. user with a list of all attributes that require monitoring. The process of selecting from the functionality list is easier than creation of a new list by different users of the system who require monitoring functionality. Software designers, system administrators and developers of monitoring systems for CloudStack can view and select the details of all the functionality, processes and attributes that can be monitored and features that can be integrated in any new monitoring software being developed. This list can be extended and updated when new compute services are added to the system or the current ones are updated.

References

1. Alhamazani, K., Ranjan, R., Mitra, K., Rabhi, F., Khan, S., Guabtni, A., Bhatnagar, V.: An Overview of the Commercial Cloud Monitoring Tools: Research Dimensions, Design Issues, and State-of-the-Art, `http://www.nist.gov/itl/cloud/`
2. Bellavista, P., Giannelli, C., Mattetti, M.: A practical approach to easily monitoring and managing IaaS environments. In: IEEE Symposium on Computers and Communications (ISCC) Symposium (2013)
3. Carvahlo, M., Esteves, R., Rodrigues, G., Granville, L.: A cloud monitoring framework for self-configured monitoring slices based on multiple tools. In: 9th International Conference on Network and Service Management (CNSM), Zurich, pp. 180–184 (2013)
4. He, S., Ghanem, M., Guo, L., Guo, Y.: Cloud Resource Monitoring for Intrusion Detection. In: IEEE International Conference on Cloud Computing Technology and Science (2013)
5. Kai, L., Weiqin, T., Liping, Z., Chao, H.: SCM: A Design and Implementation of Monitoring System for CloudStack. School of Computer Engineering and Science, Shanghai University. IEEE (2013)
6. McGilvary, A., Rius, J., Goiri, N., Solsona, F., Barker, A., Atkinson, M.: C2MS: Dynamic Monitoring and Management of Cloud Infrastructures. In: IEEE International Conference on Cloud Computing Technology and Science (2013)
7. Meng, S., Liu, L.: Enhanced Monitoring-as-a-Service for Effective Cloud Management. IEEE Transactions on Computers 62(9) (September 2013)
8. Moldovan, D., Copil, G., Truong, H., Dustdar, S.: MELA: Monitoring and Analyzing Elasticity of Cloud Services. In: IEEE International Conference on Cloud Computing Technology and Science (2013)
9. Molina, J., Vega, J., Soler, J., Corradi, A., Foschini, L.: DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds. Future Generation Computer Systems 29, 2041–2056 (2013)
10. Montesa, J., Sánchez, A., Memishic, B., Pérez, M., Antoniu, G.: GMonE: A complete approach to cloud monitoring. Future Generation Computer SystemsFuture Generation Computer Systems 29, 2026–2040 (2013)
11. Morariu, O., Borangiu, T., Morariu, C.: Multi Layer QoS Monitoring in Private Clouds. In: 24th International Workshop on Database and Expert Systems Applications (2013)
12. Zhang, Y., Liu, H., Lu, Y., Deng, B.: SLA-Driven State Monitoring for Cloud Services. In: IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (2013)
13. `http://cloudstack.apache.org/`
14. `http://docs.cloudstack.apache.org/projects/cloudstack-administration/en/4.4/`
15. `http://cloudstack.apache.org/docs/api/`
16. `http://wiki.zenoss.org/ZenPack:CloudStack`
17. `http://collectd.org/`
18. `http://exchange.nagios.org/directory/Plugins/Cloud/nagios-2Dcloudstack/details`