Paper 209-2019

Should You Move Your SAS Into the Cloud?

Paul Janssen, Janssen Consulting Inc.

ABSTRACT

Some people should not move their SAS 9.4 into the cloud, and those who do, may run into difficulties. IT departments and consulting firms are learning cloud technology on the fly. Improper installations may cause SAS to run at one third of the speed it could run compared to on-premises.[1] Furthermore, cloud technology can be expensive, and may cost hundreds of thousands of dollars per year more for a very large install, compared to keeping your SAS on-premises.

This paper was written for SAS programmers and IT departments alike. You want to move your SAS 9.4 installation into the cloud, and do it right… the first time! Choosing the most appropriate cloud compute instance type and size, and designing a proper cloud storage configuration can be challenging; done incorrectly, you will have a sub-standard or poorly performing SAS installation, resulting in a competitive disadvantage for your organization and frustrating your SAS users.

This paper defines terms, explains the bread-and-butter components of cloud technology, talks about SAS and Amazon Web Services architecture, covers SAS resource requirements, and presents a methodology for putting your SAS into the cloud, when it makes sense to do so, while being mindful of performance caveats. SAS programmers will gain a better understanding of the IT technologies required to run SAS, and IT professionals will gain better insights regarding SAS performance requirements and how to handle the cloud situation.

SAS has always lived and died on disk I/O throughput[2], and moving it into the cloud will not change that. If you do not get SAS I/O right, nothing else is going to work properly. This paper will emphasize ways to solve the I/O dilemma, so that a successful SAS cloud deployment is in the cards.

INTRODUCTION

Cloud is the future, and public cloud adoption is growing at staggering double-digit rates. Gartner calls it the *cloud gold rush*[3], and organizations that do not join the "cloud first" frenzy risk falling behind. However, cloud adaptation is difficult due to the complexities of cloud technology and it can be more costly than an on-prem deployment (contrary to popular belief). A lift-and-shift approach to migrating existing applications may backfire due to significant differences between on-premises *systems engineering* and public cloud *software engineering* (hint: you may need to *refactor* – redesign – your on-premises application and underlying infrastructure to function properly in the public cloud). Adopting the public cloud, if done incorrectly, can be costly and cause the organization to fall behind as well. If it is done right, however, your organization will benefit from the most innovative technology available.

Where does SAS fit into this picture? SAS 9.4 has resource requirements that can be difficult to meet in any environment, be it on-prem or in the cloud, especially disk I/O throughput. So how do you properly allocate, manage and monitor CPU, RAM and disk I/O resources in your SAS installation? And how does all that change when you put your SAS installation into the cloud? This paper will address those questions and can help you avoid costly mistakes by covering the following topics:

---

[1] On-premises or "on-prem" refers to computer systems that are located on the premises of the organization that uses and manages the computer systems, typically in a server room or data center.
[2] Disk input/output (I/O) is simply the act of reading data from disk into computer memory (where the data is processed by the Central Processing Unit or CPU), and writing modified data from computer memory back to disk. Due to the large volume of data that is typical of analytics processing, disk I/O must be sufficiently fast to allow SAS to process jobs that access that data within an acceptable amount of time.
[3] https://www.gartner.com/smarterwithgartner/7-hidden-cloud-growth-opportunities-for-technology-service-providers/

- **SAS System Requirements:** An overview of the resources (CPU, RAM, and disk I/O throughput) that are needed by SAS.
- **Cloud Technology Explained:** An introduction to cloud technology that covers compute instances and storage options, and defines common cloud terms in the process.
- **SAS Architecture:** An explanation of the essence of SAS 9.4 server topologies in preparation of deploying in the cloud.
- **Putting SAS in the Cloud:** A methodology to select the most appropriate cloud components to run your SAS installation, plus a review of common risks and how to mitigate them.
- **Conclusion:** A summary of what has been discussed in this paper, with final advice.

SAS SYSTEM REQUIREMENTS – *HOW* MUCH CPU DID YOU SAY YOU NEED??

Resource requirements for SAS, when compared to traditional IT systems, might seem unfathomable to IT administrators who are used to configuring two to four CPU cores[4] to power a medium to large-size file server, or perhaps eight CPU cores to power a small to medium-size database for tens to hundreds of users. By contrast, a single SAS user's server session can fully consume two server CPU cores or more, introducing the potential of having only four power SAS users using as many CPU resources as a few hundred database users!

If that doesn't seem outlandish enough, consider SAS disk input/output (I/O). The current SAS 9.4 best practice is to provision systems for a minimum I/O throughput of 100 to 150 megabytes[5] per second per CPU core (100 - 150MB/s/core). SAS typically reads from and writes to disk in blocks of 64 kilobytes. Since SAS typically processes large datasets sequentially (frequently barreling through every observation or record in these datasets), SAS qualifies their I/O as "sustained large block sequential I/O." Traditional storage subsystems are often optimized for much smaller 4 kilobyte or 8 kilobyte blocks that are written to more random storage locations, such as updates to a SQL[6] database. A storage system optimized for small random block Input/Output Operations Per Second (IOPS; pronounced "eye-ops") may perform poorly with SAS I/O.

Besides CPU and disk I/O, there is RAM (memory) to consider. At a general requirement of 8 - 16GB[7] per CPU core, it is normally not difficult to configure a system that has sufficient RAM for SAS. However, when in-memory data processing is required (for example, with SAS LASR or SAS Viya servers), RAM does become a critical resource with demands that can reach or exceed 1 terabyte[8] of RAM per server.

CLOUD TECHNOLOGY EXPLAINED

DEFINITION

A public cloud can be defined as a data center[9] that is managed by a cloud services provider and that makes computing power and data storage available remotely to users over the internet. Larger public clouds typically consist of multiple interconnected data centers, sometimes all over the world. Cloud environments are scalable, meaning that they can be sized up and down manually and automatically to accommodate fluctuating changes in demand, and to accommodate long-term growth (hence the use of the term "elastic" in conjunction with many cloud technology components). Although a public cloud is,

---

[4] A modern multi-core processor (CPU) may have anywhere from 2 to 48 CPU cores, or more; each CPU core acts like an independent CPU and runs processes simultaneously and independently.
[5] A megabyte equals one million bytes when used in reference to storage.
[6] Structured Query Language, a database language to create, query and update databases.
[7] A gigabyte (GB) of RAM is about 1.07 billion bytes.
[8] A terabyte (TB) of RAM is 1,024 gigabytes (GB) or almost 1.1 trillion bytes.
[9] A data center is an air-conditioned building or room that contains computer servers, storage devices, security appliances, and telecommunications equipment, often installed in 19-inch wide racks.

well, public, customers create virtual private clouds (VPCs) inside these vast public clouds, which are private and secure. Since the cloud provider takes care of the physical equipment that runs the cloud infrastructure software, customers do not have to worry about hardware maintenance contracts or periodic hardware refresh (replacement) cycles. Through the sharing of resources in a cloud environment, cloud providers can reduce cost due to economies of scale, especially for applications designed from the ground up for cloud. Note, however, that for high-performance applications such as SAS 9.4, a cloud deployment may be significantly more expensive than running a comparable SAS deployment on-prem.

WHAT ARE MY OPTIONS FOR SAS IN THE CLOUD?

Generally speaking, you have three main options to consider when looking at adopting cloud technology:

- **Moving your SAS to a public cloud:** You can opt to replicate your on-prem architecture in the cloud as-is (a *lift-and-shift* approach), or you can opt to rearchitect and customize your SAS design for the cloud (a process known as *refactoring*). Your level of success will depend on the size and complexity of your SAS environment, performance requirements, the size of your data, the number of users, geography, and several other factors. With larger SAS installations, refactoring is generally required to mitigate performance limitations in the cloud, especially in the area of I/O throughput, and it may make your cloud deployment more cost-effective. For small installations, including single-server SAS, a lift-and-shift with minor or no refactoring may just work, but there are still several pitfalls that you need to be aware of (several have been covered below).
- **Building a hybrid cloud:** You may leave part of your SAS installation on-premises and move another part of it to the public cloud, or you can augment your on-prem installation with a cloud deployment that leverages new technologies such as SAS Viya.[10] For very large SAS installations, augmenting in the cloud may be most beneficial and cost effective, since you could leave heavy ETL[11], linkage processing and summarization on-premises (where you enjoy maximum performance at a reasonably low cost), while modernizing your SAS installation by building agile, dynamic, exploratory solutions in the cloud. Over time, as cloud technology becomes faster and more affordable, you could move more on-premises components, data and processes into the cloud.
- **Staying On-Premises:** Some very large installations may not be a good fit for cloud today, due to cloud performance limitations and high costs associated with a large volume of cloud storage and cloud egress. For example, insurance companies and manufacturers that process terabytes of data daily and have very large SAS datasets that contain hundreds of millions of records, may find that it is more cost effective and easier to meet SLAs[12] by staying on-premises (for now).

CLOUD SERVICES

There are many cloud-native services which provide significant value, especially when designing new cloud applications from scratch. Examples include a variety of databases such as Amazon Relational Database Service (RDS) or RedShift, web server load balancers, data transfer, search, queues, directory services, developer and automation tools, monitoring tools, and so forth. These services are outside the scope of this document, which focuses on the two most important, bread-and-butter components that you need to move your SAS 9.4 into the cloud: elastic cloud compute instances and elastic cloud storage.

When your SAS 9.4 is in the cloud, the elastic cloud compute instances run SAS software and your SAS code; the elastic cloud storage contains your SAS datasets.

---

[10] SAS Viya is SAS Institute Inc.'s new cloud-ready analytics and data management platform that is highly scalable, cloud-ready, and built for the future.
[11] ETL stands for Extract, Transform and Load and is the process through which data is extracted out of enterprise database systems, transformed (converted, cleansed and augmented) and loaded into SAS.
[12] SLA stands for Service Level Agreement, which is an agreement between an IT organization (a service provider) and its internal or external customers that defines services and service standards. For example, an SLA may include performance guarantees and support request response time guarantees.

Computing Environments

Over a decade ago, IT departments would still specify and procure a physical server machine, install it in a computer rack in their data center, install an operating system such as Linux or Windows[13], install applications, and make the applications available to their users, a process that would take weeks at best, and sometimes months. Today, with cloud technology, IT departments can connect to a cloud management portal on the web, select a virtual server machine (called a *cloud instance*) from a menu, provide a few additional details, click a launch button, and within minutes, have a virtual server on which they can install applications to make available to their users. How did we get from there to here? There are many parts to that story, but in this paper, we will focus only on technology and service innovations.

To explain cloud compute instances, let's look at the evolution of computer hardware and software, from physical machines to virtual machines to cloud instances to containers.

- **Physical Machine:** This is *actual* server, workstation or desktop *hardware* that is purchased through a value-added partner, reseller, or directly from the manufacturer. Physical machines may be installed on-premises, and are also found in cloud provider data centers. Today, physical machines serve as the foundation of virtual machines and cloud instances but may also be used directly for applications.

  Figure 1 shows a physical machine with a bare metal operating system installation and applications without the use of modern virtualization or cloud technology.

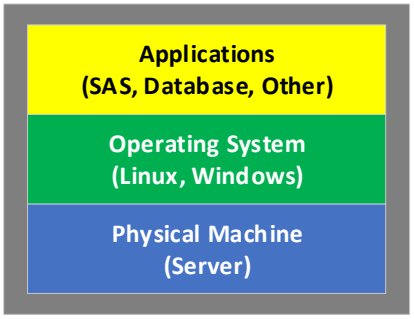


Figure 1. A Physical Machine

- **Virtual Machine:** This type of configuration starts with a physical machine (called *host*), typically located on-premises, on which IT staff installs *hypervisor* software such as VMware vSphere or Microsoft Hyper-V. Hypervisors virtualize and manage resources in the physical machine, such as CPU, RAM, device adapters, the DVD drive, and so on. IT staff then creates virtual machines (VMs) that are run by that hypervisor. Each virtual machine functions similarly to a physical machine, but is implemented in software rather than hardware based.
- **Cloud Compute Instance:** A public cloud compute instance is similar to a virtual machine, except it typically exists in a data center managed by a third-party cloud services provider rather than on-premises and managed by your own IT department. Examples of public clouds include Amazon Web Services, Microsoft Azure, and Google Cloud. Like virtual machines, cloud instances are managed by a (cloud) hypervisor and ultimately still run on physical machines (hosts). Unlike virtual machines, cloud instances have less design flexibility since they come in pre-defined combinations of CPU, memory, maximum I/O speed, and (for some instance types) local storage.

  Figure 2 on the left displays three virtual machines being managed by a hypervisor, which in turn runs on a physical machine. The combination of a virtual machine, operating system and application may be called a *virtual server*. On the right, it shows three cloud instances (which may also be called virtual machines) being managed by a (cloud) hypervisor, which in turn runs on physical infrastructure

[13] An operating system (OS) controls a computer's basic functions such as running applications, scheduling tasks, allocating memory, and controlling peripherals such as displays, printers, and storage.

located at a cloud services provider.

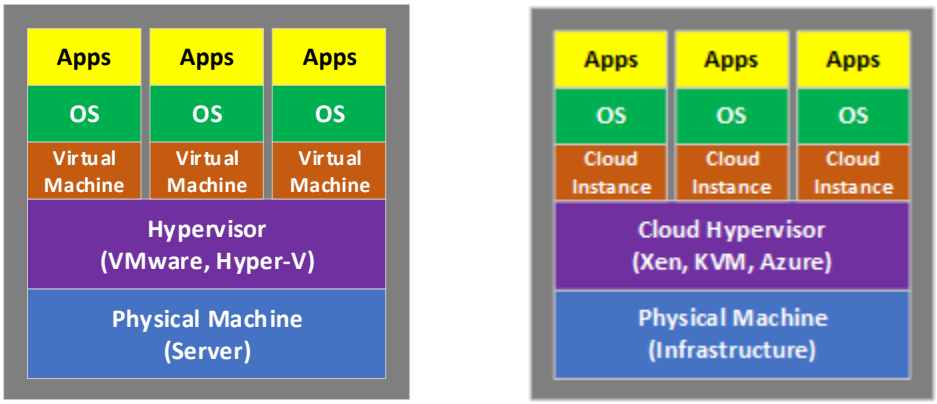


Figure 2. Virtual Machines and Cloud Instances

- **Container:** A container is a self-contained software package that includes an application and all dependent software components. A container is more lightweight and granular than a virtual machine or cloud instance. Unlike virtual machines or cloud instances, containers do not include an operating system. Containers run the same regardless of the underlying infrastructure, and they are portable, meaning that they can be moved from one computing environment to another with relative ease.[14]

  Figure 3 illustrates three containers that are being managed by a container engine which runs on an operating system, which runs on a physical machine or infrastructure located on-prem or in a cloud.

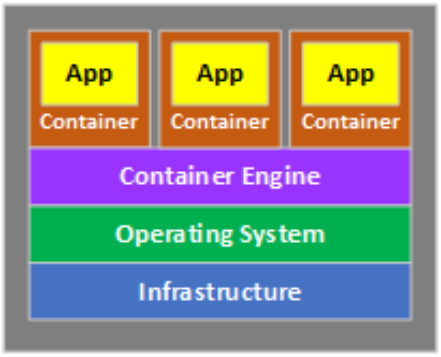


Figure 3. Containers

In public cloud environments, the general term *infrastructure* is commonly used as an abstract term for physical hosts, data communications hardware and other hardware and software components, since cloud customers generally do not know exactly what underlying components are used in the cloud.

Let's take a more detailed look at physical machines, virtual machines, cloud instances, and containers.

Physical Machines

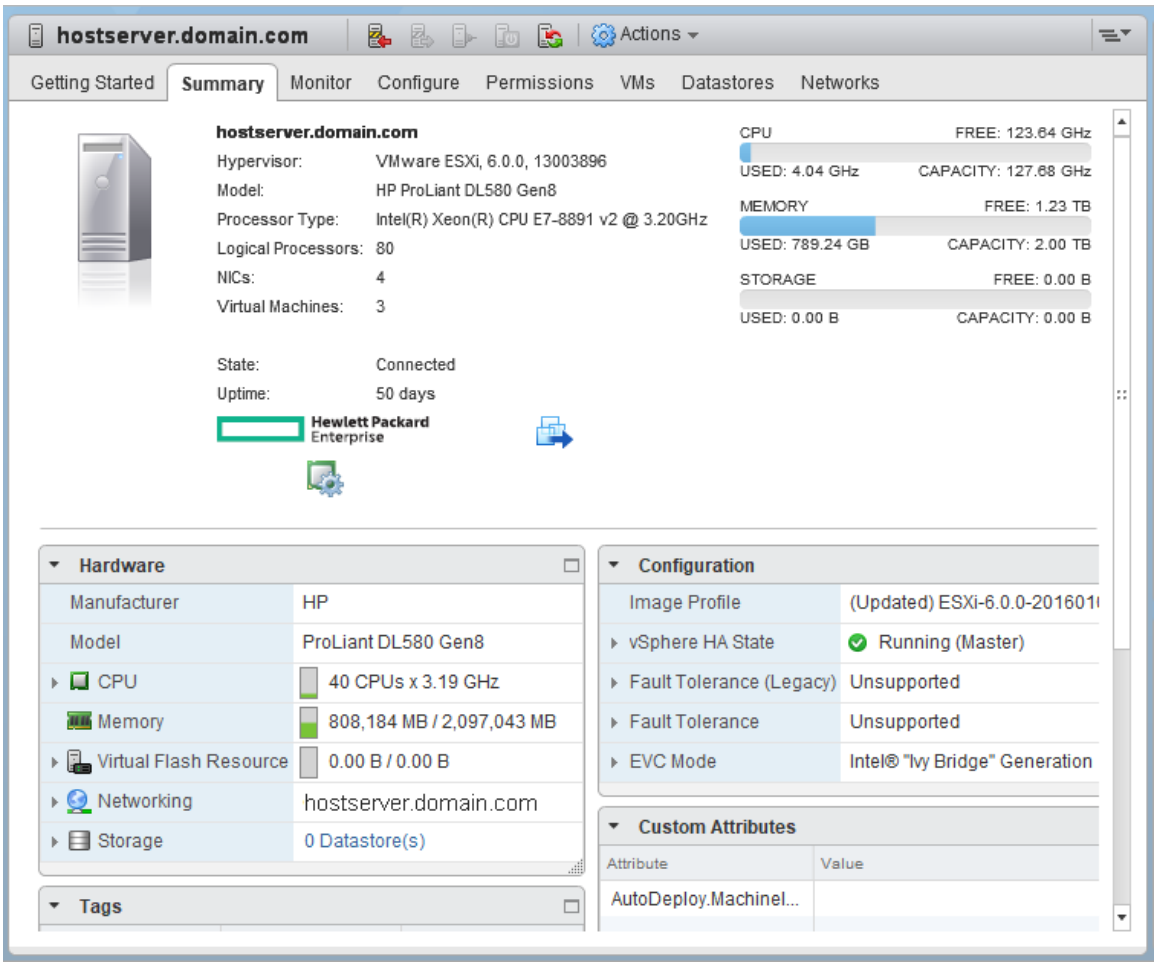
Physical server machines may be customized extensively to match a particular purpose or application. For example, the HPE ProLiant DL580 Gen10 server is available in six pre-configured models and can be highly customized by an authorized reseller or partner through HPE's One Config Simple (OCS) website.[15] Available options includes a range of CPUs, various memory configurations, local storage options, various network and storage controller cards, graphics accelerator cards, and so on. If the physical machine will be dedicated to a particular application or set of applications, IT staff may first install

[14] Containers can be moved within a variety of Linux environments. Moving containers between Linux and Windows environments is generally difficult or not supported.

[15] See https://sce-public.houston.hp.com/SimplifiedConfig/Welcome. Please note that this author is not affiliated with, or receiving compensation from, Hewlett Packard Enterprise (HPE).

an operating system such as Linux or Windows directly on the machine (this is called a “bare metal” installation), and will then install applications as needed. This method of deploying dedicated physical servers has been around for decades, but became nearly extinct after virtualization became the predominant way of running servers. Today, physical machines (servers) are mostly used as the foundation of virtualization and cloud environments (you will always need CPUs and memory to run VMs).

Display 1 shows the Summary screen of a physical server machine named *hostserver.domain.com* that has a hypervisor installed (VMware vSphere 6.0). The hypervisor is running three virtual machines; in the Hardware section, note that this hypervisor controls a total of 40 CPU cores and 2 terabytes of RAM.



Display 1. A Physical Machine Running a Hypervisor Hosting Three Virtual Machines

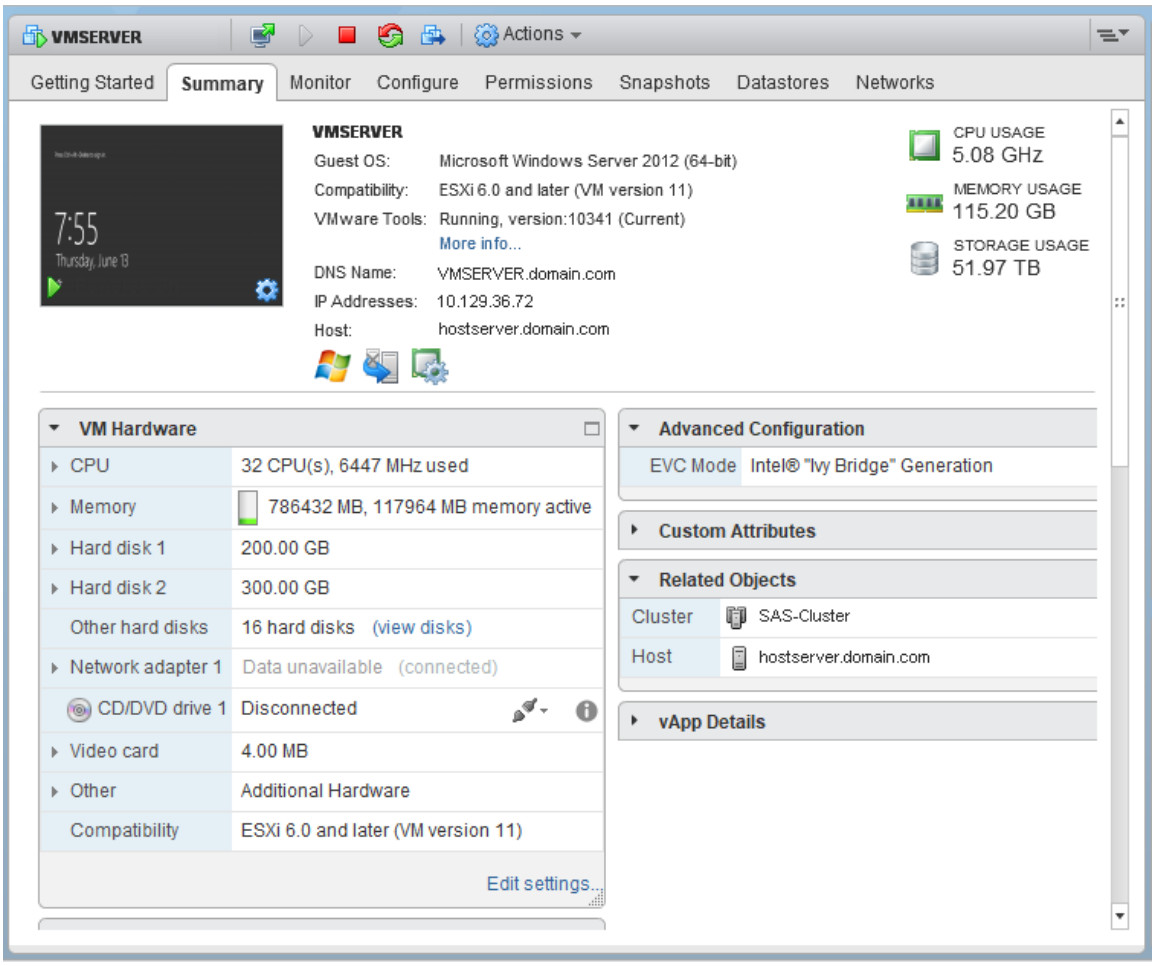
Virtual Machines

Just like physical machines, virtual machines contain a number of (virtual) CPUs, an amount of (virtual) RAM, (virtual) peripheral devices and adapters, etc., all configured by a system administrator. *The main difference is that virtual machines are software-based, and physical machines are hardware-based.* Inside each virtual machine, IT staff will install an operating system and needed applications.[16] To the operating system, the virtual machine looks and feels the same as a physical machine, and the operating system does not have to be aware that it is running inside a virtual machine, rather than a physical machine; however, in performance-optimized virtual machines, the operating system drivers may be

[16] An operating system such as Windows or Linux can be thought of as a *supervisor*, since it supervises the applications’ use of the physical machine hardware. A program that supervises supervisors (operating systems) is therefore called a *hypervisor*.

aware of (and communicating with) the hypervisor for optimal performance.[17] IT administrators may oversubscribe virtual resources (e.g., create several virtual machines that together have more virtual CPUs and virtual RAM than the underlying physical host machine). This may be okay, since not all servers will use all of their allocated resources to the maximum extent at the same time.

Display 2 shows virtual machine *VMSERVER* running Microsoft Windows Server 2012 R2 as the guest operating system.[18] This virtual machine is one of the three virtual machines hosted on the physical machine shown in Display 1 above. Out of the 40 CPUs and 2TB RAM available through the hypervisor, this virtual machine has 32 CPUs allocated and 768GB of RAM (memory). This is a SAS compute server.



Display 2. A Virtual Machine Running the Microsoft Windows Server Operating System

Cloud Compute Instances

Cloud compute instances are very similar to virtual machines (some cloud providers actually call their cloud compute instances virtual machines for simplicity). One significant difference between an *on-premises* virtual machine and a *cloud* compute instance or virtual machine is that the latter type only comes in specific, limited pre-defined configurations based on the underlying type of physical host and its storage network connection. AWS EC2 instance types, for example, are grouped by category, such as General Purpose, Compute Optimized, Memory Optimized, and Storage Optimized. Each category has

[17] Hypervisor-aware operating system drivers are called *paravirtual* or *paravirtualized* drivers.
[18] A *guest operating system* may be any operating system supported by the hypervisor software, such as Unix (many flavors), Microsoft Windows, Apple OS X, and others. It is called a guest operating system in this case because it is hosted by a host operating system (the hypervisor).

between 3 and 7 instance types, and each instance type comes in 2 to 14 sizes or models.

Table 1 shows a sampling of select AWS instance models for popular instance types m5, r5 and i3:[19]

Instance Category	Instance Type/Size	vCPUs (Cores)	RAM (GiB)[20]	Storage	Network Perf.
General Purpose	m5.xlarge	4 (2)	16	EBS-Only[21]	Up to 10Gb
General Purpose	m5.4xlarge	16 (8)	64	EBS-Only	Up to 10Gb
General Purpose	m5d.4xlarge	16 (8)	64	2 x 900GB NVMe SSD[22]	Up to 10Gb
General Purpose	m5d.24xlarge	96 (48)	384	4 x 900GB NVMe SSD	25Gb
Memory Optimized	r5.xlarge	4 (2)	32	EBS-Only	Up to 10Gb
Memory Optimized	r5.4xlarge	16 (8)	128	EBS-Only	Up to 10Gb
Memory Optimized	r5d.4xlarge	16 (8)	128	2 x 300 NVMe SSD	Up to 10Gb
Memory Optimized	r5.24xlarge	96 (48)	768	EBS-Only	25Gb
Storage Optimized	i3.xlarge	4 (2)	30.5	1 x 950GB NVMe SSD	Up to 10Gb
Storage Optimized	i3.4xlarge	16 (**8**)	122	2 x 1.9TB NVMe SSD	Up to 10Gb
Storage Optimized	i3.8xlarge	32 (**16**)	244	4 x 1.9TB NVMe SSD	10Gb
Storage Optimized	i3.16xlarge	64 (**32**)	488	8 x 1.9TB NVMe SSD	25Gb
Storage Optimized	i3.metal	72 (**36**)	512	8 x 1.9TB NVMe SSD	25Gb

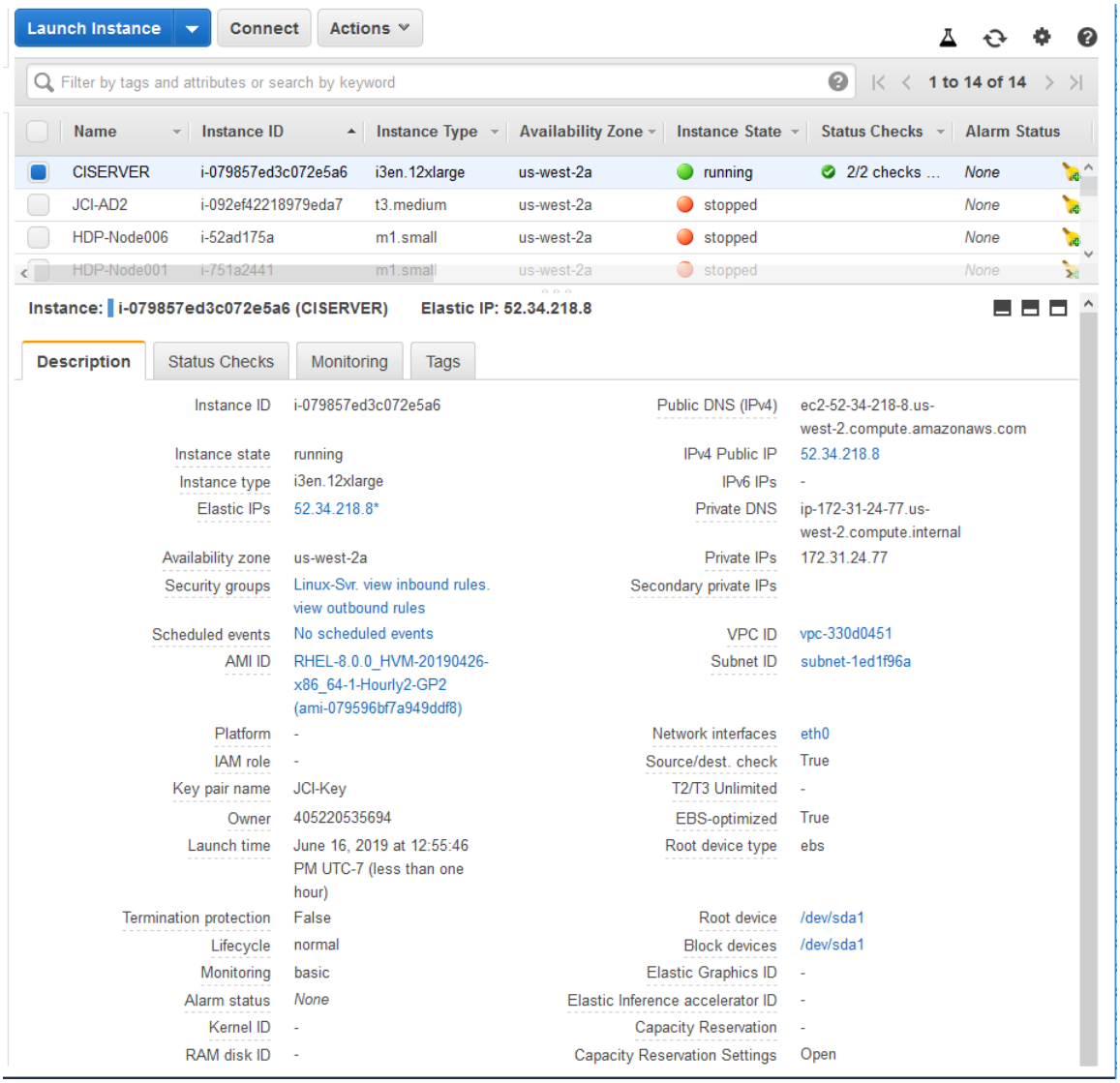
Table 1. Examples of Amazon Web Services Instance Types and Models

In Table 1 above, you can see that instances consist of specific combinations of CPU, RAM, local storage devices (for some), and I/O throughput capability. For example, for a physical i3 host (machine) that can make up to 72 virtual CPUs and 512 GB RAM available to cloud instances through its hypervisor, you may only be able to select instances with 2, 4, 8, 16, 32 or 64 cores, but you cannot select your own preferred number of cores (for example, 12 cores or 24 cores) and you cannot freely pick any RAM size that you like. This is a major difference between on-premises virtualization and cloud computing: *with cloud, you cannot mix and match resources to meet a specific configuration.* A key reason for this is cost-savings: the cloud services provider's automation software may put two i3.8xlarge or four i3.4xlarge instances on a single i3 host. The result, however, is that instances that you select sometimes have excess resources that you do not use, but that are billed by the cloud services provider regardless.

Display 3 below shows Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instance CISERVER running Red Hat Enterprise Linux v8 as the guest operating system. It is seen here from the perspective of the EC2 Dashboard, which is part of the AWS Management Console that is accessed via the web. It is not possible to see the hypervisor on which this instance is running, since that information is not available to AWS customers. The attributes of the virtual machine are not shown here, but are defined by the

[19] For a list of all Amazon EC2 Instance Types, see https://aws.amazon.com/ec2/instance-types/
[20] GiB stands for *gibibyte* which equals 1,073,741,824 bytes (a power of 2: 2^30). Today, GB (*gigabyte*) equals 1,000,000,000 bytes (a power of 10: 10^9), but may also be used as a synonym of GiB.
[21] Elastic Block Store (EBS) storage is explained in the Cloud Storage section below.
[22] NVMe (Non-Volatile Memory Express) is an interface protocol (language) built especially for Solid State Drives (SSDs). NVMe SSDs are at least 4 – 10 times faster than traditional hard drives.

instance type, which is the i3en.12xlarge with 24 CPU cores (48 virtual CPUs) and 384GB RAM.



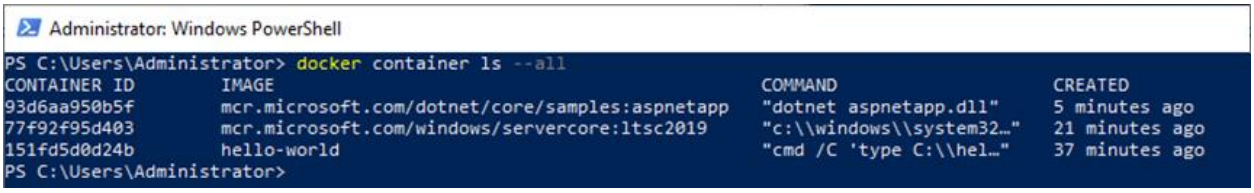
Display 3. Amazon Elastic Compute Cloud Instance running in Amazon Web Services

Containers

A container is a standardized, self-contained software package that includes an application and dependent software components such as binaries (programs) and libraries (interfaces) that support the application. Containers do not include an operating system, but instead rely on an underlying operating system to run. Because of this, containers are more lightweight than virtual machines or cloud compute instances (which do include an operating system), and IT administrators generally can run more containers on a host than virtual machines because they are smaller and use less resources. Generally, applications have to be “containerized” to run inside containers. Because containers are literally self-contained, each container has its own virtual IP address separate from the host IP address. Pre-made containers with popular software applications can be downloaded from online repositories.

Display 4 shows a list of Docker containers running on a host, including a web server (“aspnetapp”) and an application server core (“ltsc2019”). The third “hello-world” container was a test container that ran

briefly and did not remain active in memory.



Display 4. Sample Docker Container List

Docker containers can be managed by Docker Universal Control Plane (UCP), the cluster management solution from Docker (not depicted), which is included with Docker Enterprise.

Machine Type Pros and Cons

Dedicated physical machines running operating systems on bare metal may be the best performing option, but they are also expensive to buy and operate, and the most inflexible. Today, dedicated physical server machines are only used for a few remaining applications that require it. Virtual machines run slower than physical machines due to hypervisor overhead and due to competing demands for physical resources from other virtual machines, but tend to be less expensive per virtual machine and offer very flexible configuration options. Public cloud instances have less configuration flexibility than virtual machines, but can be cost efficient depending on the use case. Finally, containers, which are relatively new, have a small footprint, are very flexible and portable, are very cost-efficient and have tremendous potential for today and the near future.

CLOUD STORAGE

Storage is needed to hold all kinds of files, such as SAS datasets, spreadsheets, text files, Word and Excel documents, PowerPoint presentations, and so forth. On a desktop PC, storage can be as simple as a storage controller, a cable, an internal drive, and a particular protocol[23] that is used by the computer to communicate with the drive. Enterprise storage architectures just scale up from this basic concept. Let's start by looking at traditional storage and then at cloud storage.

Traditional Storage

Generally, we can divide traditional storage into several categories:

- **Direct Attached Storage (DAS):** This typically refers to disks that are installed in an enclosure that is directly attached to a computer via a storage interface cable and a host bus adapter (HBA).[24] Local drives installed inside a computer are also considered Direct Attached Storage. Communications between the computer and storage device(s) is done through an optimized storage protocol such as SCSI. A storage protocol is used to transfer data between computer and the storage disks.
- **Network Attached Storage (NAS):** This would include disks that are installed in an enclosure that connects to a computer host via a network interface card (NIC). A network protocol is normally used to transfer files between computer and the NAS device. *Files stored on a NAS device can be shared by multiple virtual machines or cloud instances, if needed.*
- **Storage Area Network (SAN) Storage:** This is a more elaborate storage setup, with storage enclosures that connect to computer host bus adapters via storage communications devices (intermediaries). While multiple physical machines may be connected to the same SAN storage (providing storage to the virtual machines or cloud instances that they host), generally, each physical

[23] A protocol is a common language or interface between two hardware or software components; for example, SCSI or NVMe.
[24] A host bus adapter is simply an electronic controller card that is plugged into a computer's bus. It connects to either internal hard drives or an external hard drive enclosure via a storage cable.

machine has its own dedicated storage area in the SAN. There are exceptions.

Table 2 compares the most common traditional storage types.

Storage Type	Typical Controller	Typical Protocols	Storage
Direct-Attached Storage (DAS)	Host Bus Adapter (HBA)	SCSI, NVMe, SATA	Block
Network-Attached Storage (NAS)	Network Interface Card (NIC)	NFS, CIFS	File
Storage Area Network (SAN)	Host Bus Adapter (HBA)	Fibre Channel, SCSI	Block

Table 2. Traditional Storage Types

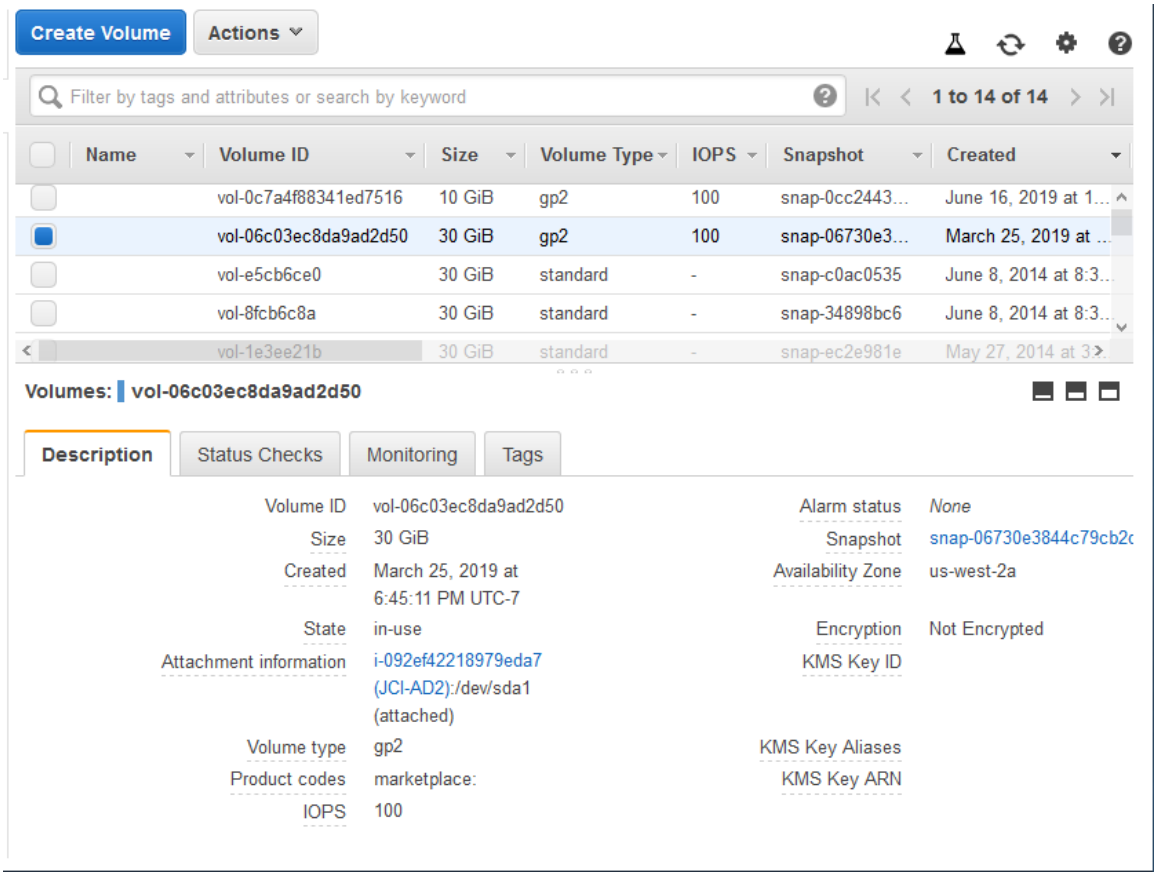
Elastic Cloud Storage

How is storage categorized in the cloud? At Amazon Web Services, for example, the following primary storage service types can be distinguished:

- **Amazon EC2 Instance Store:** This is direct-attached storage (directly attached to a host server that runs EC2 instances). Conceptually, it is similar to the hard drives or SSDs inside your desktop computer. With some EC2 instance types, the instance store is ephemeral, meaning, when the instance is stopped or restarted (rebooted), you lose all of the files previously stored in the instance store. With other EC2 instance types, the instance store can live as long as the instance itself.
- **Amazon Elastic Block Store (Amazon EBS):** This storage behaves similarly to SAN block storage in the way that it is provisioned and presented to EC2 instances, although behind the scenes and invisible to cloud users, cloud providers may use NAS devices instead for the flexibility it offers. Block storage is used for many purposes including file storage and database storage. The following storage types are available for EBS:
    - **General Purpose SSD (gp2):** SSD storage that balances price and performance for a wide variety of workloads. Used by system boot volumes as well. This is a good storage option for SAS metadata servers and middle tier servers.
    - **Throughput-Optimized HDD (st1):** Magnetic disk storage designed for frequently accessed, throughput-intensive workloads. *Generally, the best option for SAS compute servers.*
    - **Provisioned IOPS (io1)**: Highest performing SSD storage for mission-critical low-latency or high-throughput database workloads. This storage type can work quite well for SAS compute servers, but *Provisioned IOPS are subject to a site-wide maximum which might be too low for large SAS installations.* Io1 storage is also much more expensive than st1 storage.
    - **Cold HDD (sc1):** Lowest cost magnetic disk storage designed for less frequently accessed workloads. This is not a good option for live SAS data.
- **Amazon Simple Storage Service (Amazon S3):** This is a bottom-less object store that can be used for storing and retrieving any amount of data from within EC2 instances or from anywhere on the web. For example, web servers may store static content and other resources in S3 storage. Parallel processing clusters may store data in S3. Decentralized data collection applications may deposit data in S3 directly from the internet. S3 is also commonly used as a backup location for EBS storage.
- **Amazon Elastic File System (Amazon EFS):** This is network-attached storage that can be shared by multiple EC2 instances.

Cloud architectures still use physical storage devices under the hood (the data has to be physically stored somewhere!), but the cloud interface often obscures whether the cloud provider uses NAS or SAN devices in the backend.

Display 5 shows several Amazon Elastic Block Store (EBS) volumes, with detailed properties shown for the second volume from the top.



Display 5. Amazon Elastic Block Store Properties

SAS 9.4 ARCHITECTURE

SAS architecture can range from a modest single desktop machine to multimillion-dollar enterprise installations with many servers and enormous storage capacity. Let's start from the beginning and work our way up.

ARCHITECTURAL TIERS

The SAS 9.4 platform is split into different functional parts, known as tiers.

- **Client Tier:** SAS client application and web browser access to the SAS Intelligence Platform. SAS Enterprise Guide and SAS Enterprise Miner are examples of client tier applications.
- **Middle Tier:** SAS web servers and web applications.
- **Metadata Tier:** A central repository for security and configuration information (users, user groups, user roles, security templates, server and data definitions, application objects, and so forth).
- **Compute Tier:** SAS servers that actually process data. There are general and more specialized SAS compute tier servers. This tier includes SAS Grid servers.
- **Data Tier:** Enterprise data sources accessed through SAS, likely in various locations.

Basic SAS architectures may not include all tiers. The diagrams below will clarify tier use.

SINGLE MACHINE SAS

Figure 4 below shows two single machine SAS installations. The legacy desktop machine on the left does not include SAS Enterprise Guide; SAS users would write and run code through the Display Manager System, which is embedded in the SAS executable program. The desktop machine on the right lets users

generate and submit code through the SAS Enterprise Guide client, which submits SAS programs to the SAS executable program, which functions as a local SAS server that performs all of the computations.

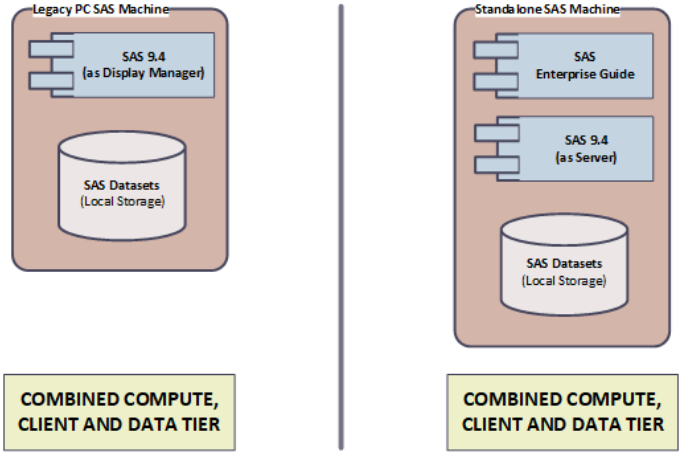


Figure 4. Two Examples of Single Machine SAS

The advantages of running SAS on a single desktop machine include simplicity (relatively easy to configure) and affordability (no need to buy a SAS server). There are many disadvantages, including significant performance limitations, issues with security (data can more easily be compromised) and issues with continuity (data may not be adequately backed up and the system may crash with larger datasets).

SIMPLE SAS TOPOLOGY

Figure 5 shows two SAS clients: SAS Enterprise Guide to write and submit SAS code, and SAS Management Console to manage users and configurations on the SAS server. There may be multiple client machines that connect to the server machine. In the center is a SAS server machine that contains a SAS Metadata server, and various other servers that execute SAS code. To the right in the figure is a collection of possible data sources for the SAS server machine (there may be others as well). *A server, in SAS parlance, is a process that runs on a machine; therefore, you can have multiple servers running on the same machine.*

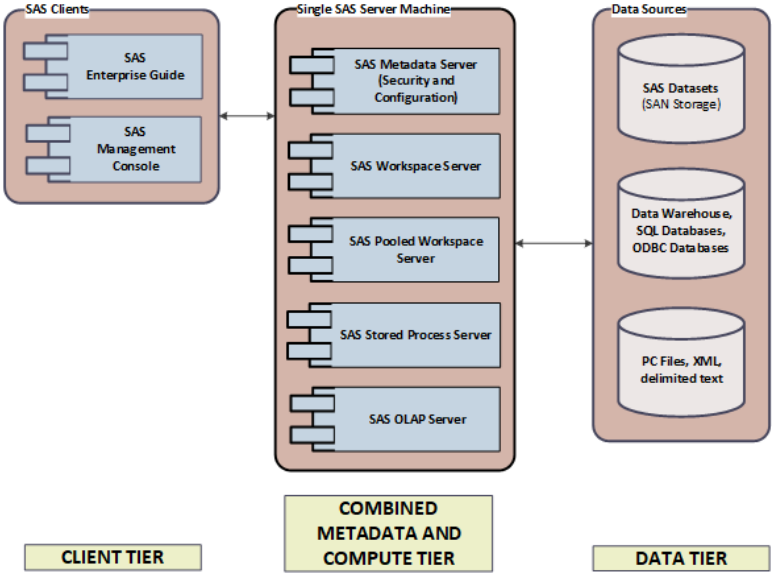


Figure 5. Single Server Machine SAS Topology

The advantages of running SAS on a server generally include improved performance, security and continuity (compared to the single desktop machine model), plus the ability to serve multiple client machines (users). Disadvantages include increased cost (hardware or cloud hosting fees, and SAS license fees) plus a lack of scalability (limited ability to size-up the system as you add more clients).

BASIC DISTRIBUTED SAS TOPOLOGY

Figure 6 shows a basic distributed SAS topology, with SAS Metadata Server, SAS Compute Server, and SAS Middle Tier Server running on separate machines.

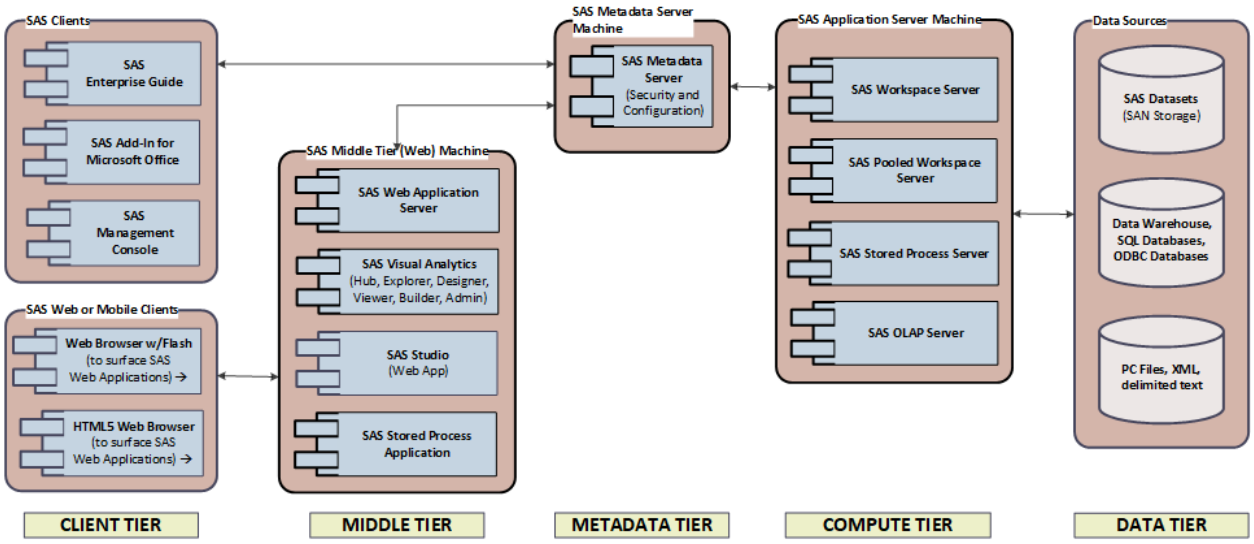


Figure 6. Basic Distributed SAS Topology

The advantages of running SAS on multiple server machines include improved performance, security and scalability (compared to the single server topology). Disadvantages include increased cost (hardware or cloud hosting fees, and increased SAS license fees), plus continuity (reliability) can still be improved.

ADVANCED DISTRIBUTED SAS TOPOLOGY

Figure 7 shows how you can achieve additional performance and continuity by running *horizontal* SAS Metadata Server and Middle Tier *clusters*.[25]

[25] A horizontal cluster is made up of multiple machines (called *cluster nodes*) that each carry some of the compute workload. Horizontal clusters can often withstand the loss of a single cluster node (*redundancy*).

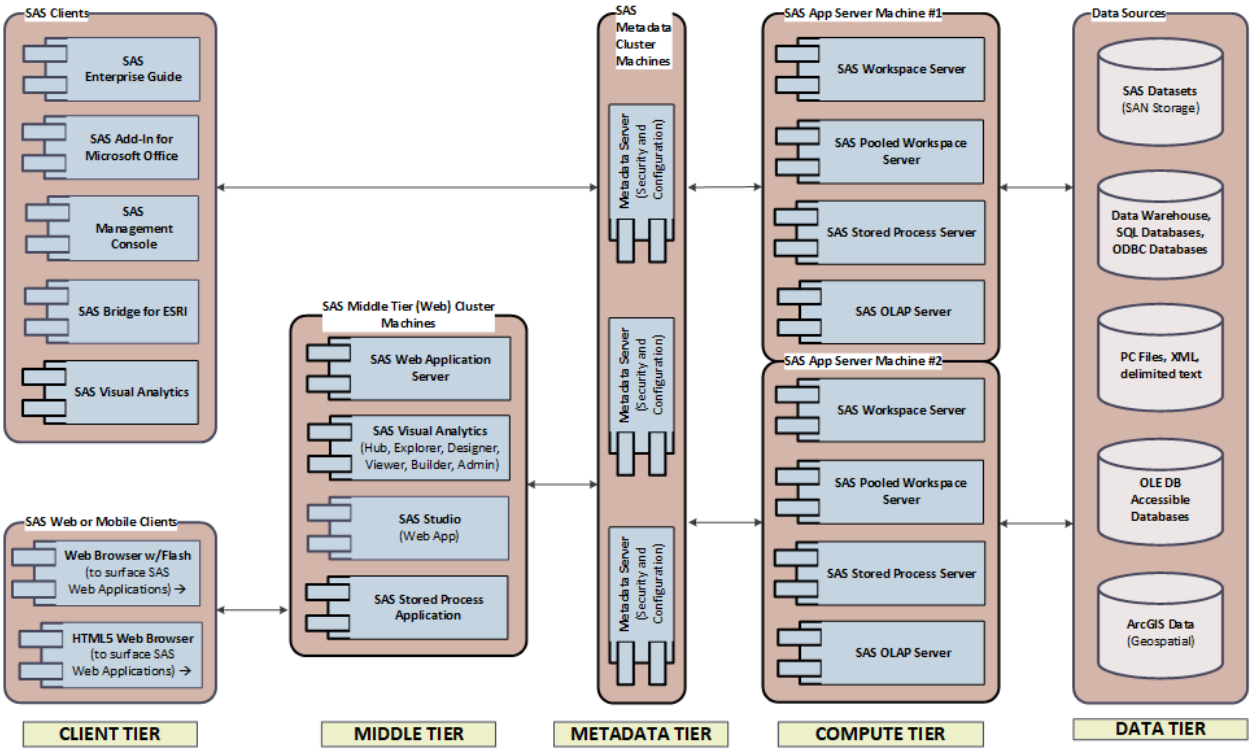


Figure 7. Advanced Distributed SAS Topology

The advantages of running SAS on multiple redundant server machines generally include improved performance, scalability and continuity (compared to the multiple server machine model above). Disadvantages include increased cost (hardware or cloud hosting fees, and SAS license fees).

SAS VIYA ARCHITECTURE

SAS Viya is SAS Institute Inc.'s new analytics and data management platform that is highly scalable, cloud-ready, and built for the future. SAS has a number of products available for SAS Viya, including SAS Visual Analytics, SAS Visual Statistics, SAS Model Manager, and many others. SAS Viya is outside of the scope of the current version of this SAS 9.4 cloud paper. For a great introduction to SAS Viya, see https://video.sas.com/category/videos/introducing-sas-viya.

PUTTING SAS INTO THE CLOUD

This is the most important section of this paper, and is based on Amazon Web Services. Most concepts presented in this section should apply to other cloud services platform providers as well, such as Microsoft Azure or Google Cloud.

SELECTING THE MOST SUITABLE CLOUD INSTANCES

1. Start by reading the latest version[26] of Margaret Crevar's excellent paper called "Important Performance Considerations When Moving SAS® to a Public Cloud" (see References below).

   An enhanced version of the *i3* instance type, which is recommended for SAS compute servers in the paper, is available as of May 2019 as the *i3en*, which significantly improves the CPU core-to-I/O ratio.

2. Collect the SAS system specifications or site records that apply to your SAS installation and make a list of your SAS server machines, to determine the cloud compute instances that you will need.

[26] Currently dated March 2019

3. Open the following two pages in a web browser simultaneously, on different tabs:

   - https://aws.amazon.com/ec2/instance-types/
     for a complete list of instance types, *by optimization,* with features and benefits.

   - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html
     for important supplemental instance type information about *I/O capabilities.*

4. To select a compute instance type for a server, SAS or otherwise, determine whether your instance(s) should first prioritize CPU, RAM, or storage (I/O throughput). *The reason is that you cannot freely combine your own ideal CPU, RAM and storage options, as shown above in this paper. Instead, you have to select an instance from a limited list of pre-configured instance types, each with specific CPU, RAM, and storage options.* This is why you may need to prioritize by resource type.

   Using the two AWS web pages opened above, browse through the AWS instance types using the instance type page, and shortlist some instance candidates for your SAS machines, with their key specifications, while keeping SAS' recommendations in the Crevar cloud paper and your SAS server topology in mind.

   - If the most important resource for the application is CPU, look for compute optimized instance types in the instance type list, such as the c5.
   - If the resource priority is RAM, look for memory optimized instance types in the instance type list, such as the r5.
   - If the top priority is storage (and I/O throughput), look for storage optimized instance types in the instance type list, such as the I3 or the i3en. Since SAS is typically I/O bound, a storage optimized instance is generally preferred for SAS compute servers because it offers direct attached high-speed NVMe SSD drives, that can be used for temporary storage such as the SAS WORK volume. Temporary storage has the highest I/O throughput requirement and does not have to be shared between servers.
   - Be careful with instance types that specify network performance or EBS bandwidth of "*up to* …"; this could mean that your instance would be running on a physical machine shared with other cloud tenants. If you have a "noisy neighbor," your instance may suffer from performance degradation and cause issues in your SAS installation. In this case, and especially for SAS compute servers, select a higher instance model that lists actual expected performance and bandwidth without the use of the "*up to*…" prefix.
   - Always look for instance types that offer unexpected benefits; for example, a memory optimized instance type that happens to have great CPUs, or a compute optimized instance type that has better than expected I/O throughput. Find these benefits by comparing instance type features.
   - For compute servers, select instance models that have twice the number of vCPUs than the cores allowed by your SAS 9.4 license (remember that there are two logical processors or virtual CPUs per core, and that SAS is licensed by the core). For example, if you have a SAS server license for 24 cores, and you are looking at i3en instances for your compute server machines, initially select the i3en.12xlarge with 48 vCPUs to match the license.

     Note that when you move your SAS license to the cloud, your local SAS sales office may double your licensed processor count to match the number of vCPUs in your cloud instance. This is because SAS 9.4 will only use one out of every two vCPUs (dedicated CPU cores).
   - For in-memory servers such as LASR or Viya, there may be cases where you might want to prioritize RAM over I/O throughput and examine different instance types to find the best possible fit. The i3.16xlarge and i3.metal instances, for example, have sufficient RAM at 488GB and 512GB, respectively, for 16 core SAS 9.4 compute servers; however, heavy use of SAS LASR and other in-memory SAS products such as SAS Viya may require more RAM than the i3 can offer. The newer i3en.24xlarge at 768GiB of RAM improves on the i3, but if even more memory is required, it may be necessary to step away from i3 and the i3en instance types and consider a

memory optimized instance type such as the X1 or X1e. SAS LASR and SAS Viya with the Cloud Analytics Services (CAS) engine are beyond the scope of this document.

5. Go through your instance type shortlist and for each SAS compute server instance, determine the maximum I/O throughput for that instance, using the second AWS web page opened above (the EBS-optimized instance list), as follows:
   - Divide the number of virtual CPUs by two to get the number of cores.[27]
   - Find the instance in the second AWS web page (the EBS Optimized instances list) and determine the maximum available I/O throughput for that instance.
   - Divide the cores by the maximum available I/O throughput to calculate MB/s per core.
   - Add the calculated throughput per second per core to your shortlist.

For most if not all AWS EC2 instances types and models, *AWS Elastic Block Store (EBS) I/O throughput is capped at a 1,750MB/s maximum per instance* (as of July, 2019). For most combinations of I/O throughput and CPU core counts, the maximum I/O throughput per instance therefore peaks at 50 - 55MB per second per core, on paper (the exception is the new i3en, which has a better ratio). *Actual I/O throughput is also related to the EBS configuration, which is covered below.*

CORRECTING OVERALL CPU COUNT OR THE I/O TO CPU CORE COUNT RATIO

If, after selecting the best possible instance type and model for each SAS compute server machine, the best available instance type and model provides significantly less I/O throughput than the SAS minimum recommendation of 100 – 150MB/s/core, *select the next higher instance model with (typically) twice the I/O throughput.* Since this will also double your number of vCPUs, make a note to disable excess vCPUs when launching the instance, so your SAS 9.4 license will not shut your compute server down due to it having more CPUs than the license permits. For example, an i3.16xlarge instance can provide up to 1,750MB/s I/O throughput (satisfying 16 cores at up to 109.4MB/s/core or 12 cores at up to 145.8MB/s/core) but actually contains 32 cores (64 vCPUs). The lesser i3.8xlarge instance contains a better matching number of cores if we aim for 16 cores but can only provide up to 875MB/s throughput (satisfying only 6 to 8 cores). This is why disabling CPUs may be necessary in some cases.

Another possibility is that you selected the best possible instance type and model for your compute servers, but you were unable to find a vCPU count that matches your SAS license. For example, you have a SAS license for 16 cores, selected the i3en instance type as the best option, and noticed that the i3en does not have an instance type with 32 vCPUs (the equivalent of 16 cores). In this case, you would select the next higher available instance model in the i3en instance type (for example, the i3en.12xlarge with 48 vCPUs) and make a note to disable the excess 16 vCPUs when launching the instance.

Regardless of the situation, if you disable CPUs, you'll still pay for that unused CPU power. With the new i3en instance type, it may not be necessary to disable as many CPUs since this instance type has an improved network/storage throughput to vCPU ratio compared to the i3, depending on the model.

How to Disable excess vCPUs When Launching AWS Instances

In the AWS Management Console (https://aws.amazon.com/console/), after logging in and selecting the EC2 service, click Launch Instance (which creates a new instance), select your operating system and instance type, and click on *Next: Configure Instance Details* at the bottom of the window. In the *Configure Instance Details* screen, select the check box labeled *Specify CPU Options* and change *Threads per core* from 2 to 1, and adjust the *Core count* appropriately to match your SAS license.

Figure 8 shows how to reduce the core count on an AWS EC2 instance during instance launch. *Once launched, the core count and threads per core can no longer be changed on an EC2 instance.*

[27] A CPU core generally contains two hyper-threads; these hyper-threads allow the CPU core to act more like two separate processors (with some limitations). This is why each hyper-thread is called a virtual CPU or vCPU.

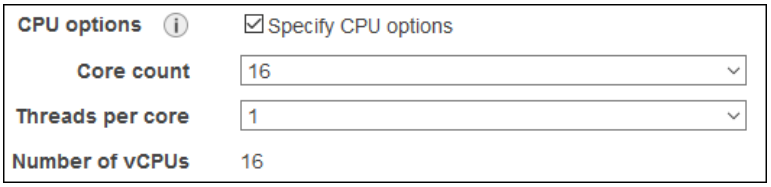


Figure 8. Reducing core count on i3en.12xlarge AWS EC2 instance from 24 to 16.

Next, follow the remaining steps in the Launch Instance wizard as needed and click Review and Launch to launch (create) the instance. *Note, however, that you will still pay the same rate for the instance, whether you have disabled any CPUs or not.*

LOCAL INSTANCE STORE CONFIGURATION

Local instance stores can provide high internal I/O bandwidth for the SAS WORK file system. The AWS i3.16xlarge and i3.metal EC2 instance sizes each have eight 1.9TB NVMe local instance storage SSDs which when striped (aggregated) together provide nearly 13TB of usable storage. A single 1.9TB NVMe SSD drive in an i3 instance type can perform up to 1,000MB/s (1GB/s). All eight i3 1.9TB NVMe SSD drives in a striped configuration can hold 12.8TB and can perform up to 6,800MB/s (6.8GB/s). This satisfies the requirements for a file system that will hold SAS WORK.

The AWS i3en.24xlarge has eight 7.5TB NVMe local instance storage SSDs, which when striped together should provide approximately 54TB of usable storage for SAS WORK. At the time of writing this paper, i3 and i3en are also the only *storage-optimized* instance type that have large-size instance stores with NVMe disks, rather than HDD's. i3 has been the SAS recommended instance type for AWS, but now that the i3en was announced in May 2019, it is possible the i3en will take the place of the i3.

Since the local instance stores on the i3 and i3en instance types are ephemeral[28], you will need a provisioning script that runs on startup to mount and format the drives for use with SAS WORK. The Technical Paper *Performance and Tuning Considerations on Amazon Web Services with SAS® 9.4 Using IBM Spectrum Scale* contains a Linux script to provision ephemeral storage on an AWS EC2 instance (see References below). If you need a PowerShell script, please contact the author of this paper.

CONFIGURING EBS STORAGE

Most modern instance types are EBS optimized, but that is no guarantee for good I/O throughput performance. Rather, it depends on the instance type and model, as explained above, as well as the attached EBS storage configuration.

Out of the available volume types for EBS volumes, *Throughput-Optimized HDD* (st1) is the most appropriate choice for SAS and its sustained large block sequential I/O. The EBS *Provisioned IOPS* volume type (io1) works very well for SAS and is actually faster than st1, but with very large compute servers you may run into the provisioned IOPS site limit, which limits the number of compute servers and the number and size of EBS volumes installed at an AWS site. This is one key reason why st1 has been the SAS and IBM-recommended compute server or file system storage type for several years. AWS customer support can increase the provisioned IOPS site limit for you but not indefinitely. Last but not least, io1 is at least twice as expensive as st1 (which is the other key reason why st1 is recommended).

EBS st1 volumes provide up to approximately 40MB per second I/O throughput *per terabyte of provisioned storage*. Provisioning 12,800GB or greater will enable a single EBS st1 volume to perform up to its full 512MB I/O throughput per second potential while not being dependent on I/O credits.[29] *512MB/s*

---

[28] Ephemeral storage is not persistent, meaning that when the instance is stopped and restarted, all volumes, folders and data on the ephemeral storage will be gone.

[29] I/O credits accumulate in buckets while I/O is relatively low, and are consumed when I/O exceeds a base rate that is set for the EBS volume at the provisioned size. The effect on SAS can be inconsistent I/O performance which may cause performance and stability issues in SAS.

(500MiB/s) is the maximum throughput for any EBS volume regardless of its volume type (st1, io1, etc.).

A single EBS volume at 512MB/s is not sufficient to meet the SAS minimum throughput requirement of 100MB – 150MB per second per core if the server contains more than 4 cores. Therefore, multiple EBS volumes may need to be striped (aggregated) in sufficient numbers to reach the required throughput *or* the overall per-instance maximum I/O throughput of 1,750MB/s (which would take a minimum of four EBS st1 volumes, striped together so they work in parallel).

There is a possibility that more disk space will need to be provisioned than is actually needed for storage purposes, just to satisfy I/O throughput performance. *AWS charges for provisioned disk space, regardless of how much space you actually use, and costs can be significant.* Also, the striping per SAS would need to be done without redundancy built in at the OS-level, relying instead on redundancy supplied by the AWS EBS storage array back-end. If you apply a redundant storage scheme to EBS volumes, performance drops significantly. The disk performance test software *fio* (Flexible I/O Tester) written by Jens Axboe can be used to test the I/O throughput of AWS volumes of any type, in various configurations.

CLOUD COSTS

The AWS Simple Monthly Calculator can be used to plug in information about your EC2 instances, EBS storage, and other products and services inside AWS: https://calculator.s3.amazonaws.com/index.html. This will produce an estimated monthly cost for AWS services. After inputting all of the data, click on the next tab (Estimate of your Monthly Bill) where you can save the information to a unique URL or export it as CSV. Then, if you are using a value-added provider, if you know the cost of your SAS licensing, you may be able to make an educated guess about how much your provider makes on your managed AWS SAS arrangement (if applicable). *There are no guarantees for the accuracy of the calculator.*

CLOUD RISKS

Any change in data processing environments carries risk; cloud is no exception. Here is a partial list of risks that you need to look out for when you are considering to move your SAS into the cloud:

- Vendor lock-in may be an issue, since it may not be easy to migrate fully deployed cloud infrastructures from one cloud vendor to another. Moreover, you will be susceptible to any outages, delays and other interruptions of a cloud environment that are beyond your control.

- As explained in this paper, cloud I/O throughput may be inadequate for your use case. This is caused by performance limitations in the AWS Elastic Block Store (EBS), and the need to use software RAID when aggregating individual EBS volumes for performance. There are various tools available to measure disk throughput, including iotest.sh or rhel_iotest.sh for Unix environments, Sasiotest for Windows environments, or the fio (flexible I/O tester) which is available for both environments.

- It may take significant time to upload large datasets to the cloud. Can you meet your internal customer's SLA requirements for having data available in the cloud in time? Are you able to update the data in the cloud in time for each update cycle to finish before the next cycle starts? An assessment of data movement and an analysis of throughput speeds may be necessary to determine if your data logistics will continue to meet customer requirements as you move to the cloud.

- In a hybrid cloud environment, *off-cloud data consumption* may cause performance issues or timeout errors due to network throughput limitations, latency and contention. One solution is to put SAS clients in the cloud as well, together with the SAS servers, but having SAS client software away from on-premises desktop software may cause different issues.

- If your SAS is in the cloud but your SAS clients are on-premises, there may also be minor to significant delays while users access data or wait for results to be returned to their desktops.

- Significant cloud data egress (data leaving the cloud) may become expensive, since cloud service providers typically charge a tiered per-gigabyte fee. Also, your storage daily change rate may be high

with SAS and result in higher than usual S3 storage charges (where you store your EBS backups). This is an important topic to discuss with your cloud provider account team.

CONCLUSION

Moving your SAS into the cloud is not an easy task. First, you need to understand cloud technology and its benefits and limitations. Moreover, you need to know how to approach a cloud design, while keeping an eye on the most important and often neglected IT resource required by SAS: I/O throughput. When adopted correctly, cloud computing can allow your organization to take advantage of the latest, most innovative technologies available. However, if cloud cannot meet your SLAs or is cost-prohibitive, do not be afraid to postpone cloud adoptation. Either way, the information presented in this paper should help you make a compelling case for moving your SAS into the cloud, or keeping it on-premises (for now).

REFERENCES

Crevar, M. 2019. “Important Performance Considerations When Moving SAS® to a Public Cloud.” *Proceedings of the SAS Global Forum 2019*, Dallas, TX SAS3633–2019 : SAS Institute Inc. Available at https://www.sas.com/en_us/events/sas-global-forum/program/proceedings.html.

Crevar, M. “Does it Matter Where the Various Components of Your SAS Infrastructure are Installed?” SAS Support Communities, Administration, Admin & Deploy. Accessed July 14, 2019. Available at https://communities.sas.com/t5/Architecture/Does-It-Matter-Where-the-Various-Components-of-Your-SAS/m-p/483426

Smith, B., Kuell, J., and Porter, B. 2018. “Performance and Tuning Considerations on Amazon Web Services with SAS® 9.4 Using IBM Spectrum Scale™.” *SAS Technical Paper*, Cary, NC : SAS Institute Inc. Available at http://support.sas.com/resources/papers/performance-tuning-considerations-amazon-web-services-sas-9-4-ibm-spectrum-scale.pdf.

ACKNOWLEDGMENTS

I would like to thank Russell Lavery for his valuable insights that contributed to this paper.

RECOMMENDED READING

- *Amazon Elastic Compute Cloud User Guide for Linux Instances*
- *Amazon Elastic Compute Cloud User Guide for Windows Instances*
- *Amazon Web Services General Reference Version 1.0*
- *Architecting for the Cloud – AWS Best Practices*
- *SAS® 9.4 Companion for UNIX Environments, Sixth Edition*
- *SAS® 9.4 Companion for Windows, Fifth Edition*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Paul Janssen
Janssen Consulting Inc.
+1-916-716-2326
paul.janssen@janssenconsulting.com
https://www.janssenconsulting.com