GEEKGUIDE

SUSE

Modernizing with SQL Server on Linux for a Cloud Native World

Microsoft SQL Server on SUSE Linux

Abstract

Microsoft SQL Server enables developers and organizations to successfully harness the power of data in their preferred language and environment, including Linux for on-premises, cloud and edge scenarios as well as container deployments. With SUSE, you have access to all the same enterprise-grade security features found in SQL Server on Windows with no compromises, available with cloud options like Pay-As-You-Go and Bring-Your-Own-Subscription and container-type deployments leveraging the SUSE Containers-as-a-Service offering.

LINUX™ JOURNAL

Table Of Contents

Geek Guide

Mission-critical information for the most technical people on the planet.

About the sponsor

SUSE, the world's largest independent open source company, powers digital transformation with agile, enterprise-grade open source solutions, edge to core to cloud. Collaborating with partners, communities and customers, SUSE delivers and supports enterprise-grade Linux, software-defined infrastructure and application delivery solutions to create, deploy and manage workloads anywhere — on premises, hybrid and multi-cloud — with exceptional service, value and flexibility.

SUSE is a digital bridge to customers' cloud future. SUSE and Microsoft help customers to build, enrich and extend into cloud so they can realize those futures. Together, SUSE and Microsoft aim to empower customers to Simplify, Modernize and Accelerate digital transformations by connecting their modern Linux platforms and Open Source tools to the power and scale of Microsoft Azure. They want to empower customers to share, invent and accelerate growth through community-driven products that are fully supported by SUSE.

Introduction

Data has saturated all segments of our lives in numerous form factors. it plays a key role whether the data is hosted in a private data center or a public cloud or even at the Edge. A person might currently be exchanging data, for instance, between a cell phone and a Nest thermostat or a tablet and the Fitbit they're wearing. Beyond that, organizations across just about any industry are using big data in a variety of scenarios, including customer relationship management, risk assessment, regulatory compliance tracking, research and development efforts, and much more.

For technologists, this proliferation of data requires us to be able to converge disparate data sets from numerous sources – relational, non-relational or even unstructured – into a single dashboard to gain meaningful insights. Databases and the platforms they run on are the key to secure, scalable success for businesses and their partners and, ultimately, consumers.

The National Institute of Standards and Technology has rated it as the least vulnerable database platform for the last nine years in a row. At the same time, many organizations remain committed to open source platforms and technology, including Linux. This Geek Guide offers background, an introduction to SQL Server and Linux, plus resources for database administrators and their organizations considering SQL Server on a Linux platform, such as a SUSE. We'll cover important topics, including security, high availability scenarios, containers and container management, performance, and Big Data Clusters. And we'll look ahead to how the cloud and high-performance computing will bring innovative solutions to a variety of scenarios.

About SQL Server on Linux

A rich and deep history exists for Linux as a platform, starting with its inception in 1991 by Linus Torvalds. Early on, most database administrators (DBAs) avoided running databases on Linux because it lacked the maturity of Unix distributions.

As with other open-source Unix distributions, Linux development is done on the GNU Compiler Collection or the Intel C Compiler. Torvalds hoped to limit commercial activity, but Linux has become the most widely used OS distribution used on enterprise servers. Numerous Linux distributions exist, including SUSE, openSUSE, Debian, Ubuntu, CentOS, RedHat, and Fedora. In fact, Microsoft recently announced there are now more virtual machines (VMs) running Linux than Windows in Azure, its cloud computing service.

...Microsoft recently announced there are now more virtual machines (VMs) running Linux than Windows in Azure, its cloud computing service.

Microsoft has been supporting Linux for SQL Server since September 2017, offering customers more choices of platform and embracing the open source community. SUSE was quick to acknowledge that the Microsoft-Linux combination would be a great benefit to customers.

Since then, Microsoft has been dedicating a lot of engineering effort to ensure feature parity between Windows and Linux. With the most recent release of SQL Server on November 4, 2019, this has come to nearly 99% equality.

Why move databases to SQL Server on Linux?

Agile development practices and DevOps are becoming the norm, and organizations are expected to move faster and with fewer environment silos. As teams begin to form across multiple technical areas, often including business stakeholders, SQL Server can also broaden to support all these areas.

The skills required to maintain a Windows server are different than those required to support Linux. Although Linux is a “flavor” of Unix distribution, other direct Unix distributions, such as HP-UX and AIX, have mostly been abandoned for a unified Linux offering. Linux has become the preferred OS for this revolution.

Microsoft has embraced Linux for its enterprise database platform, Microsoft SQL Server, which actually required little change to the underlying SQL Server code. Linux support has increased from a simple installation to advanced configurations, containers, and advanced multi-distribution support. Today, incredible features like PolyBase allows you to query any data source, including Hadoop clusters, NoSQL, Oracle, and Teradata. And data scientists can access Machine Learning Services through SQL Server as well.

Today, incredible features like PolyBase allows you to query any data source, including Hadoop clusters, NoSQL, Oracle, and Teradata.

The SQL Server product group worked with the Microsoft research team to create a component called SQL PAL (Platform Abstraction Layer) which allows us to run the same SQL Server binary on Windows and Linux platforms. So, when you run a query against SQL Server on Windows or Linux it will analyze, optimize and execute the exact same data. Managing or administering SQL Server is virtually the same, on either platform.

Many open source technologies, such as big data solutions, are native to Linux, and Microsoft made a strategic decision not to port its Big Data Clusters feature to Windows. They worked with the Linux and open source community instead, bringing the workload engine (MS SQL) to the platform and to the data, where that data resides.

Why SUSE and how they partner with Microsoft

If you're new to the idea of Microsoft and SUSE Linux, here's the story. The relationship began almost 15 years ago, back in 2006. The companies entered into a partner agreement to collaborate on interoperability – a unique move among Linux distributors at the time. This alliance delivered greater choice to customers who were developing comprehensive environments that included a wide variety of technologies and they've been building together ever since.

SUSE Linux, as an operating system for data analytics and IoT workloads, has been providing a stable, secure and reliable platform for customers and partners for a long time. They've been partnering with leading data platform providers for decades and have thousands of certifications, including benchmarks with Microsoft SQL Server, DB2, and SAP.

...SUSE was SAP's own in-house implementation platform and was first to market with SAP on Azure solutions.

When talking about big data, if you think your organization is ready to embrace microservices and an analytics solution like Apache Spark running on Hadoop, SQL Server on SUSE is a great option. This is especially true if your organization relies on SAP, because SUSE was SAP's own in-house implementation platform and was first to market with SAP on Azure solutions. SUSE powers the world's largest SAP deployments, including customers running SAP on Azure, and integrates with Azure monitoring.

And, if you are ready for a journey to the cloud you should know that SUSE and Microsoft co-engineered a SUSE Linux Enterprise kernel that was tuned for Azure to optimize SQL Server throughput and reduce latency.

To deploy SQL Server in the Azure Cloud you can choose Microsoft's Pay-As-You- Go (PAYG) pre-configured Marketplace offer, which includes the tuned SUSE kernel or you can also leverage Bring-Your-Own-Subscription (BYOS) and consume your SUSE software on Azure.

Managing security with SQL Server and Linux

Some organizations have especially intense security needs. For instance, securing sensitive data and staying compliant with industry regulations, such as PCI-DSS (Payment Card Industry Data Security Standard) and GDPR (General Data Protection Regulation), can be particularly important. A compromised database system can lead to a loss of revenue, regulatory fines and a negative impact on the reputation of your business.

Tracking compliance and maintaining database security requires significant admin resources. SQL Server has tools, such as Data Discovery and Classification, and SQL Vulnerability Assessment tools that allow you to identify compliance issues and help tag and classify specific datasets to ensure compliance. SQL Server 2019 on Linux and Windows has obtained Common Criteria certification as of August 2020.

SQL Server offers many security features that address these challenges, such as TDE (Transparent Data Encryption), SSL (Secure Sockets Layer) encryption, Always Encrypted, Auditing, Dynamic Data Masking, and Row-Level Security. These enterprise-grade security features are available to SUSE Linux Enterprise Server customers, not just to Windows customers.

In addition, you can enable AD authentication. This allows domain-joined clients on either Windows or Linux to authenticate to SQL Server using their domain credentials and the Kerberos protocol. If you are a Linux user who is new to SQL Server, review this walkthrough of the security features of SQL Server on Linux.

[AD authentication] allows domain-joined clients on either Windows or Linux to authenticate to SQL Server using their domain credentials and the Kerberos protocol.

Meeting high availability needs

You'll want to safeguard data to ensure that it is available during an outage – planned or unplanned – also known as providing business continuity. The ability to respond to local incidents and get back up and running quickly is known as High Availability (HA). For instance, if the storage on a physical server fails, you need to switch traffic to another server quickly. That sort of recovery should be possible within the same data center and cause minimal impact.

A more catastrophic event, such as the loss of a data center, triggers what is commonly referred to as Disaster Recovery (DR). DR generally involves more than just ensuring that a database is ready for use elsewhere. Before bringing a database or instance online, core aspects of the infrastructure must be functioning as well.

Organizations that need to plan for DR for mission-critical workloads can use SUSE Linux, but DBAs and architects still must learn how best to architect a highly available platform with the combination of SQL Server and SUSE.

Organizations that need to plan for DR for mission-critical workloads can use SUSE Linux, but DBAs and architects still must learn how best to architect a highly available platform with the combination of SQL Server and SUSE.

You can use the following components of SQL Server HA on Linux for your HA systems:

- **Pacemaker.** Pacemaker is the core clustering control component, which coordinates across the clustered machines.
- **CoroSync.** CoroSync is the preferred cluster messaging layer and works in conjunction with Pacemaker.
- **Fence method.** Fencing is the process of isolating a node of a computer cluster or protecting shared resources. Several fencing methods can be employed.
- **libQB.** This library provides high performance logging, tracing, inter-process communication (IPC), and polling.

- **SQL Server HA resource agent.** This agent offers specific functionality to integrate with Pacemaker and monitor availability groups.
- **External cluster type.** This new clustering mechanism has been introduced to communicate with Linux distributions.

Want to learn more? Find out how to configure a cluster SUSE Linux Enterprise Server for a SQL Server availability group and discover business continuity options from SQL Server.

Solving big data problems

The first magnetic storage devices were large, bulky, and expensive and required massive amounts of power and cooling. They also failed quite often. But, as time has progressed, the sizes of computing components – especially storage – have shrunk and costs have decreased. However, the capabilities, reliability and power of those components have increased. This allows us to store much more data, and because we've had those technologies longer, we've been able to store more data over time.

And what can we do with it all — why keep it? Given a system that processes and delivers the data quickly enough, there are quite a few examples of specific use cases for large sets of data. The most basic use of big data is to simply query it, usually in the context of gaining insights to help make business decisions and solve problems. But with the vast amount of data coming from so many sources, scaling up or out is necessary.

Scaling physically can be done, but it becomes expensive. So, virtual machines (VMs) are a better option. For instance, you can use a single physical computer to host multiple virtual computers running Hadoop and quickly and easily turn them off and on. But you can only slice up the physical computer's components so

...you can use a single physical computer to host multiple virtual computers running Hadoop and quickly and easily turn them off and on.

many times before performance suffers. And then there is the work of installing the operating system multiple times, with all the drivers, patches, libraries and so on. This is where containers come in.

Containers for SQL Server on Linux

The first obvious benefit of containers is that they are isolated, meaning that processes or dependencies do not interact with others. They also behave the same way as specified in the build file and they are portable, such that dependencies are always shipped with the container no matter the environment, on premises or cloud. Containers are also typically easy to use and start/stop and there is no installation required. Instead, applications like SQL Server are ready to run immediately. Learn more about how to deploy and connect to SQL Server docker containers.

[Containers] also behave the same way as specified in the build file and they are portable, such that dependencies are always shipped with the container no matter the environment, on premises or cloud.

Simplifying scaling with containers

Deploying SQL Server in containers simplifies and speeds up deployments, with easier application development and better database DevOps. Most SQL Server deployments are on physical servers or virtual machines (on premises or in the cloud). Cloud-based offerings, such as Azure SQL Database or Azure SQL Database Managed Instance, are considered Platform as a Service (PaaS) offerings. These offer the ability to deploy databases and instances without the need to manage the underlying infrastructure. But now there's another option: containers.

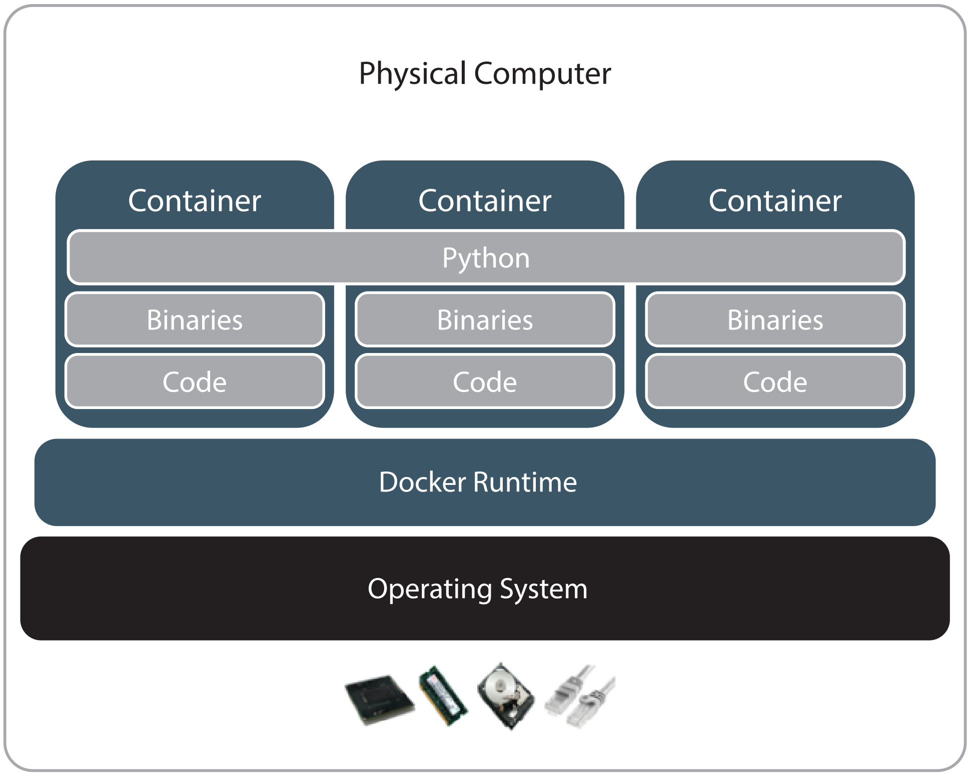
An introduction to containers

If containers are new to you, here's an example of how they work. Containers offer a better solution than a scale-out architecture. A container system does not represent the CPU, I/O, memory, and networking subsystems of a computer, ready for you to install a full operating system. Instead, a container uses a runtime engine on the current operating system. Layers of binary files run in this engine, isolated from other containers and the host operating system.

The advantages of a container system can be explained well through an example. Let's say you want to run a Python program. To do that, you need a computer, an operating system, the Python program binary files, any code you want to run, and any assets (graphics files, database, and so on) that you create and use in the program that is installed for every version of the application you want to run.

Using a container runtime engine, in this case Docker, the architecture is illustrated in the following graphic.

Container Architecture Example:



The benefit of this architecture is that the system only carries the assets it needs to run the Python program, and that environment is completely encapsulated. If the engine determines it's the same Python throughout, it only loads that version once. You could redeploy a copy of that program with a different version of Python or your application, and it would operate independently.

You could redeploy a copy of that program with a different version of Python or your application, and it would operate independently.

The other major advantage associated with this system is how a container is created and deployed. The general process is declarative:

1. Create a **text** file that describes what binaries and files you want in your container.

2. Utilizing this file, build a container image, which is a binary

file containing all of those assets. You can store this image in a public or private repository and then run a **pull** command to bring the binary image to your system.

3. Use a **run** command to load the binary image into the container runtime engine (Docker, in this case) and start up. This environment is now a container.

4. Depending on what your application needs to accomplish, you can allow network access in and out of the container.

While container technology is revolutionary, it leads to a significant issue: how to manage multiple containers. You want a scale-out architecture for your big data processing system, and that can mean dozens, hundreds, or even thousands of containers running the processing. This can become overwhelming to deploy and manage. This is why an orchestration solution is needed. You can deploy SQL Server on Kubernetes. Learn more about [SQL server deployment on AKS and deploying and connecting to SQL Server Docker containers]().

Scaling with containers

There are several container orchestration tools, but the most popular right now is Kubernetes. It also provides enhanced networking and storage systems that allow for a robust cluster environment. Kubernetes encompasses four main components, plus the container itself:

[Kubernetes] also provides enhanced networking and storage systems that allow for a robust cluster environment.

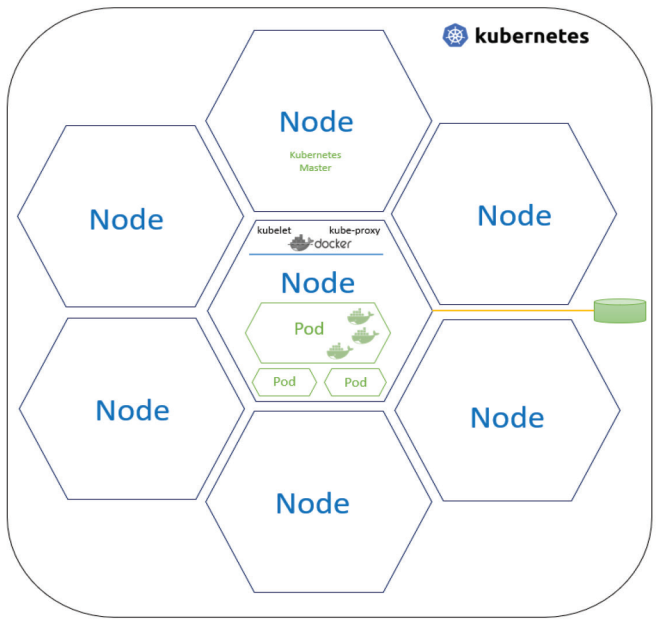
- **Pod.** This is the logical boundary around one or more containers.
- **Node.** Pods run on nodes, which are computers (virtual or bare metal). Each node will run, at a minimum, the container runtime, the kubelet, and kube-proxy processes, which handle the nod and pod operations and in-cluster networking.
- **Cluster.** One or more nodes make a cluster. One node in each cluster is designated as the Kubernetes Master, which controls the node.

- **Volume.** A volume exists as a separate entity for storage. Because the pods and nodes can move around in the cluster, storage is accessed through a software claim.

Here's how all of these elements work together.

Kubernetes components

- Container(s) live in *Pods*
- Pod(s) are abstractions within *Nodes*
- Node(s) are PCs or VMs
- Cluster(s) are groups of *Nodes*
- Storage is by means of Volume(s) mounted through a *Claim*



Class container management platform

Containers-as-a-Service (CaaS) is a cloud service model that allows you to manage and deploy containers, Linux applications and clusters through container-based virtualization. Container engines, orchestration, and the underlying compute resources are delivered to users as a service. In a CaaS model, containers and clusters are provided as a service that can be deployed in onsite data centers or in the cloud.

In a CaaS model, containers and clusters are provided as a service that can be deployed in onsite data centers or in the cloud.

CaaS lets you start, stop, scale and manage containers using a provider's API calls or GUI interface. Like most cloud services, you'll pay only for the CaaS resources — such as compute instances, load balancing and scheduling capabilities — that you use.

SUSE offers a CaaS platform that includes Kubernetes to automate lifecycle management of modern applications and surrounding technologies that enrich Kubernetes and make the platform itself easy to operate. With the SUSE Containers-as-a-Service offering you can accelerate modern application delivery with Kubernetes, simplify Kubernetes administration and maximize return on investment with a flexible, no lock-in solution.

With the SUSE Containers-as-a-Service offering you can accelerate modern application delivery with Kubernetes, simplify Kubernetes administration and maximize return on investment with a flexible, no lock-in solution.

Closing data gaps with SQL Server Big Data Clusters and PolyBase

As the variety of data and the volume of that data has risen, the number of types of databases has also grown dramatically. SQL Server initially met those needs by adding support for XML, JSON, in-memory, and graph data in the database. It became a flexible database engine that companies count on for performance, high availability and security. But SQL Server was not designed to be a database engine for analytics on the scale of petabytes or exabytes. In the past it could be challenging to join your SQL Server data with external data.

Now, SQL Server Big Data Clusters and PolyBase provide flexibility in how you interact with all that data. Big Data Clusters encompass a set of capabilities that allow you to deploy scalable clusters of SQL Server, Spark and HDFS containers running on Kubernetes. These components run side by side to enable you to read, write and process big data from Transact-SQL or Spark. This, in turn, makes it possible for you to easily combine and analyze high-value relational data with high-volume big data.

SQL Server PolyBase is the feature you can use to query external data by using the same T-SQL syntax used to query a database table. It also allows you to query data on Hadoop Distributed File System (HDFS) directly. In addition, the Big Data Cluster's HDFS system can mount other storage (such as S3 and Azure sources), allowing yet another way to virtualize data.

In-depth view of big data cluster architecture

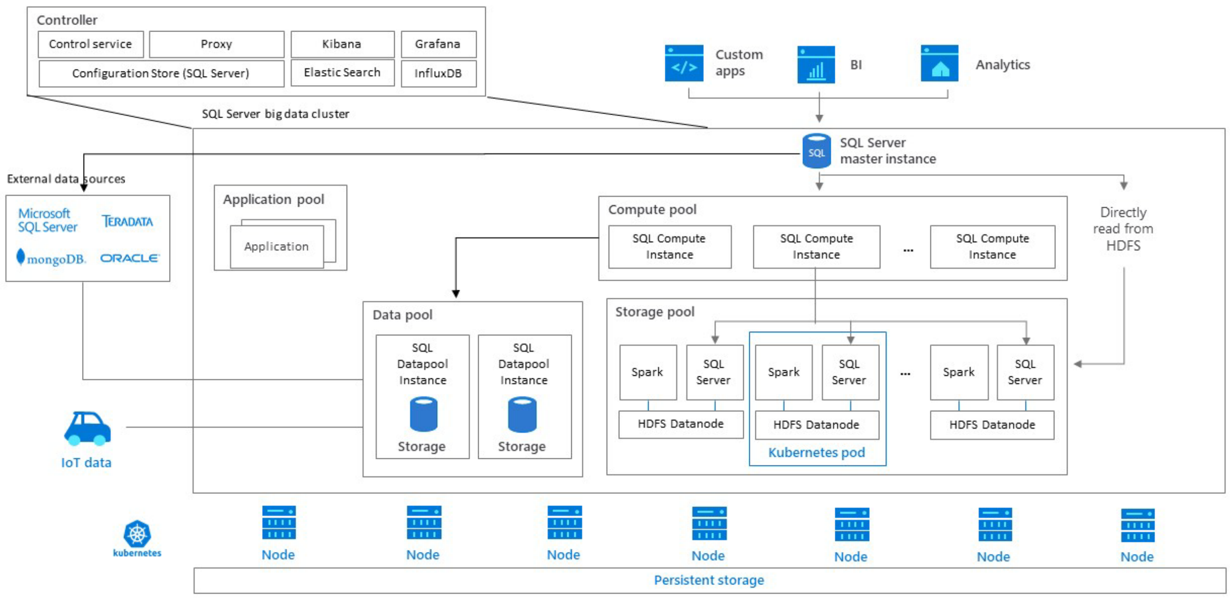
Microsoft is fully embracing Apache Spark (a data streaming/ data crunching solution) such that, in the same Kubernetes pod, you can be running SQL Server Engine and Apache Spark runtime talking to an underlying HDFS store. Learn more about the Spark job submission feature in this how-to guide.

Big Data Clusters are based on Kubernetes pods, which are single units of isolation. A pod can access a HDFS data node, which is based upon persistent storage. While people often think of containers as dealing with ephemeral data only, keep in mind that when running SQL Server on containers, you can use persistent storage. If you use persistent storage, data will remain after the lifetime of that container. You can leverage SUSE Enterprise Storage (SES) – powered by Ceph technology – in that scenario. Kubernetes can use several ceph storage types like RBD, CephFS and SES is one of the default CSI plugins for Kubernetes.

Kubernetes can use several ceph storage types like RBD, CephFS and SES is one of the default CSI plugins for Kubernetes.

The following diagram illustrates how the components a SQL Server Big Data Cluster architecture work together.

SQL Server Big Data Cluster architecture:



Maximizing performance

SQL Server can be used in the most demanding computing environments required today. Using a variety of features and techniques, including in-memory database operations, can make dramatic increases in your transaction processing rate while still allowing near-real-time analysis without having to move your transaction data to another data warehouse for reporting and analysis. SQL Server has also recently expanded the number of opportunities to tune database operations automatically, along with tools and reports to allow monitoring and optimization of queries and workloads. Comprehensive diagnostic features like Query Store allow SQL Server to identify performance issues quickly.

Learn more by checking out this performance guide and best practices for SQL on Linux and also for configuration guidance for SQL Server Big Data Clusters. If you're a Linux user who is new to

SQL Server, have a look at this walkthrough for the performance features of SQL Server on Linux. You can view results posted by the Transaction Processing Performance Council (TPC) for SQL 2017 on SLES:

- http://www.tpc.org/tpch/results/tpch_result_detail5.asp?id=117111701
- http://www.tpc.org/tpch/results/tpch_result_detail5.asp?id=119040101

Journey to the cloud: running SQL Server on Azure Virtual Machines

Azure is not only about services or acting as a Platform as a Service. Azure's SQL data platform offerings are helpful when you want to make the transition to the cloud but may have critical requirements that cause hesitation. SQL Server on Azure Virtual Machines (VMs) lets you deploy SQL Server in the cloud without having to manage any physical hardware. SQL Server VMs also simplify licensing costs when you Pay-As-You-Go.

SQL Server on Azure Virtual Machines (VMs) lets you deploy SQL Server in the cloud without having to manage any physical hardware.

Currently, you can build SQL Server High Availability deployments with pacemaker clusters on SLES VMs on the Azure platform. You can also build SQL Server-specific HA based on Pacemaker clusters with the Azure Fencing agent (SUSE HA extension), available for Bring-Your-Own subscription-based SLES releases on Azure.

IT Infrastructure Insights

A Linux-based infrastructure can be vital to supporting the analytics applications of tomorrow. As noted on its roadmap, SUSE prioritizes helping customers gain actionable insights about the efficiency of their resources by using Prometheus and Grafana. Prometheus is a monitoring tool that is used to record real-time metrics in a time-series database. It is an open-source software project, written in Go. Metrics are collected using HTTP pulls, allowing for higher performance and scalability. Grafana is used for data visualization, monitoring and analysis.

SUSE Manager, which provides supported Prometheus and Grafana, greatly simplifies the management of Linux deployments helping drive compliance across cloud and on-premises instances. Database administrators could use SUSE Manager and leverage opensource SQL Exporters to combine with Prometheus and Grafana for dashboard and insights. In the future, SUSE would like to leverage SQL Server monitoring to help customers act on usage of CPU, memory, OS layers, and so on.

SUSE Manager, which provides supported Prometheus and Grafana, greatly simplifies the management of Linux deployments helping drive compliance across cloud and on-premises instances.

Conclusion

SQL Server enables DBAs and organizations to harness the power of data using their preferred language and environment, which includes Linux since SQL Server 2017. SUSE continues to work with Microsoft on new offerings including cloud options such as Pay-As-You-Go (including our tuned SLES kernel) and Bring-Your-Own-Subscription and container-type deployments leveraging the SUSE Containers-as-a-Service offering.

Learn more and get started

- The Power of SQL Server on Linux
- Microsoft SQL Server Product Page
- Microsoft Azure Pay-As-You-Go Offer