

19:00 - Begrüßung und PASS Community News 19:15 - SQL Server on Docker 21:00 - Q&A, Diskussionen und Networking

Docker ist ein Ooken

Docker ist ein Open-Souroe-Lösung, die es ermöglicht
Anwendungen in einer geschlossenen Umgebung (Container) zu
betreiben. Dies bietet für Administratoren vielfältige
Möglichkeiten für die Bereitstellung und den Betrieb von SOL
Server. In dem Vortrag möchte ich Euch zeigen, wie man mit SOL
Server in einer Container-Umgebung auf Basis von Windows,
Linux und in Azure arbeitet – Voraussetzungen, Bereitstellung,
Betrieb und Speziallösungen. Lasst Euch überraschen und von
den neuen Möglichkeiten von SOL Server in einer solchen
Umgebung begeistern.

Sylvio Hellmann is working as a Principal Consultant and Team Leader of the data group at URANO Informationssysteme GmbH. His main focus is SQL Server, Microsoft Business Intelligence/PowerBl as well as Nachine Learning technologies on premise and in the cloud. He supports oustomers around the world to solve complex challenges in projects and in operational environments. Sylvio has also a deep knowledge of development skills and he loves to work with PowerShell.





#### Terms of Use

Diese Präsentation und die Demo-Skripte sind nur für die Ausbildung bestimmt und wird bereitgestellt "wie besehen" ohne Gewährleistung jeglicher Art, ob ausdrücklich oder konkludent, einschließlich, jedoch nicht auf konkludente Garantien der Handelsüblichkeit, Eignung für einen bestimmten Zweck und Nichtverletzung beschränkt.

Ich übernehme keine Garantie für die Richtigkeit der Informationen zum Zeitpunkt der Veröffentlichung. Sofern nicht anders angegeben, sind Firmen, Organisationen, Produkten, Domänennamen, E-mail-Adressen, Logos, Personen, Orte und Ereignisse in den Beispielen frei erfunden. Einige Videos, Bilder, Schriftarten und Logos wurden aus dem Internet in die Präsentation eingefügt. Hierbei wurde jeweils die Quelle genannt - direkt in der Präsentation oder in den Notizen. Eine Weiterverbreitung oder gewerbliche Nutzung dieser Präsentationselemente sind vorher mit dem Copyright-Besitzer zu klären. Präsentationselement, die ich erstellt habe,  $\ d\"{u}rfen\ nur\ mit\ meiner\ ausdr\"{u}cklicher\ Genehmigung\ durch\ andere\ Personen\ kommerziell$ verwendet werden.

Als Präsentationsvorlage wurde das <u>Vega-Template</u> von Jun Akizaki "The Power of PowerPoint" verwendet. Für eine weitere Verwendung ist das Copyright mit dem Designer zu klären.

© Sylvio Hellmann

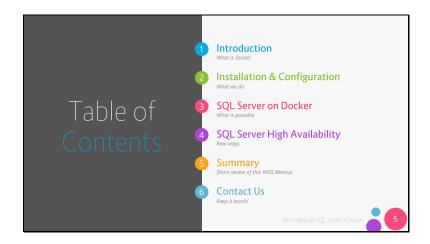


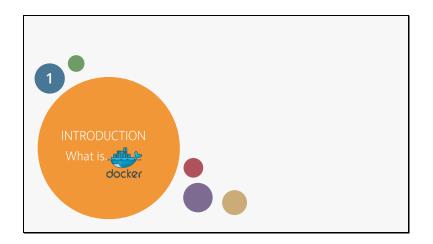


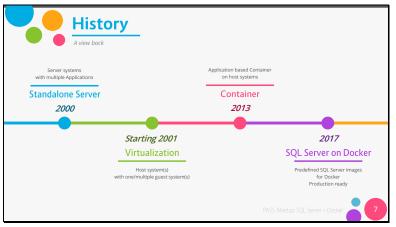














#### Source:

https://en.wikipedia.org/wiki/Timeline\_of\_virtualizati on development

- From 1965: start virtualization on mainframes
- October 26, 1998, <u>VMware</u> filed for a patent on their techniques, which is granted as U.S. Patent 6,397,242
- July 2001, VMware created the first x86 server virtualization product.
- July 12, 2006 VMware releases <u>VMware</u> <u>Server</u>, a free machine-level virtualization product for the server market.
- Docker wurde im März 2013 von dotCloud veröffentlicht. Der initiale Commit auf dem <u>Git-Repository stammt allerdings vom 19. Januar 2013 und umfasste 1146 Zeilen Code</u>

## Picture Source:

http://opensourceforu.com/2017/02/docker-favourite-devops-world/

Containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries — anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment.



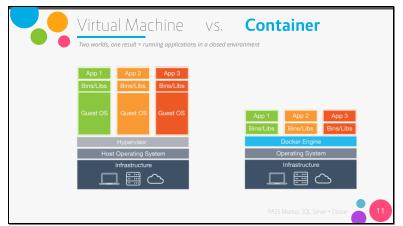
# Advantages of Docker

- 1. Rapid application deployment: Docker requires minimal runtime for any application, reducing the size and allowing quick deployment.
- 2. Portability across machines: An application with all its dependencies can be bundled into a single container, which is independent from the host version of the Linux kernel, deployment model or platform distribution. This container can be simply transferred to another machine that runs Docker, and executed there without any compatibility issues.
- 3. Version control and component reuse: We can track the successive versions of a Docker container, inspect the differences, or even roll back to previous versions. Docker reuses components from the preceding layers, which makes them lightweight.
- 4. Sharing: We can use a remote repository to share our Docker container with others. Red Hat even provides a registry for this purpose, and it is also possible to configure our own private repository.
- 5. Lightweight footprint and minimal overhead:
  Docker images are usually very small, which helps in rapid delivery and reduces the time to deploy any new application containers.
- 6. Simplified maintenance: Docker also reduces the effort and risk of problems caused due to application

dependencies.

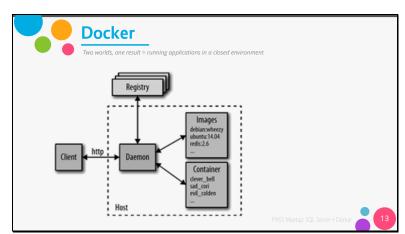
7. Security: From a security point of view, Docker ensures that applications that are running on containers are completely segregated from each other, granting us complete control over the traffic flow and management. No Docker container can look into processes that are running inside another container. From an architectural standpoint, each container actually gets its own set of resources, ranging from processing to network stacks.





https://www.sdxcentral.com/cloud/containers/definitions/containers-vs-vms/





**Container Host:** Physical or Virtual computer system configured with the Windows Container feature. **Container Image:** A container image contains the base operating system, application, and all the application dependencies that are needed to quickly

**Container OS Image**: The container OS image is the operating system environment.

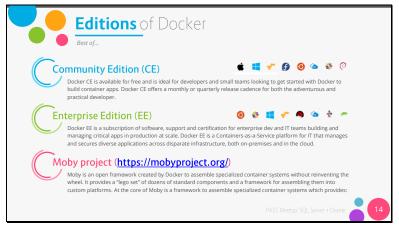
deploy a container.

Container Registry: Container images are stored in a container registry, and can be downloaded on demand. It is a place where container images are published. A registry can be remote or on-premises. Docker Engine: It is the core of the Docker platform. It is a lightweight container runtime that builds and runs your container.

**Docker file**: Docker files are used by developers to build and automate the creation of container images. With a Docker file, the Docker daemon can automatically build a container image.

Im Zentrum steht der *Docker Daemon*, der für das Erstellen, Ausführen und Überwachen der Container zuständig ist. Aber auch das Bauen und Speichern von Images fallen in seinen Zuständigkeitsbereich. Sowohl Container als auch Images sehen Sie in der Abbildung rechts. Der Docker Daemon wird durch docker daemon gestartet, was normalerweise schon durch das HostBetriebssystem geschieht.

- Der Docker Client auf der linken Seite wird dazu genutzt, per HTTP REST mit dem Docker Daemon zu kommunizieren. Standardmäßig geschieht das über ein Unix Domain Socket, aber es ist auch ein TCP-Socket möglich, um Remote Clients zu nutzen, oder ein Dateideskriptor für durch systemd verwaltete Sockets. Da die gesamte Kommunikation über HTTP abläuft, ist es einfach, sich mit entfernten Docker Daemons zu verbinden und Bindings an Programmiersprachen zu entwickeln. Es hat aber auch Auswirkungen darauf, wie Features implementiert werden – so ist zum Beispiel ein Build Context für Dockerfiles notwendig (siehe Abschnitt 4.2.1). Die für die Kommunikation mit dem Daemon verwendete API ist wohldefiniert und dokumentiert, so dass Entwickler Programme schreiben können, die direkt mit dem Daemon arbeiten, statt auf den Docker Client zurückgreifen zu müssen. Docker Client und Docker Daemon werden als ein Binary bereitgestellt.
- In Docker Registries werden Images abgelegt und verteilt. Die StandardRegistry ist der Docker Hub, auf dem tausende öffentlich verfügbarer Images zur Verfügung stehen, aber auch kuratierte »offizielle« Images. Viele Organisationen und Firmen nutzen eigene Registries, um kommerzielle oder »private« Images zu hosten, aber auch um den Overhead zu vermeiden, der mit dem Herunterladen von Images über das Internet einhergeht. In Abschnitt 7.4.1 finden Sie weitere Informationen über das Betreiben einer eigenen Registry. Der Docker Daemon lädt Images aus Registries herunter, wenn er docker pull-Anforderungen bekommt. Aber auch, wenn docker run ausgeführt wird oder Images in der FROM-Anweisung von Dockerfiles aufgeführt sind, die lokal nicht vorhanden sind, greift er auf die Registry zu.



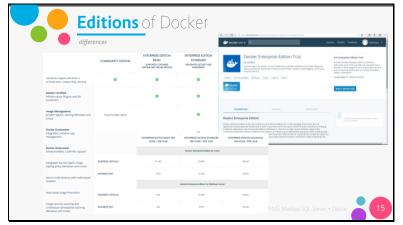
Moby IS RECOMMENDED for anyone who wants to assemble a container-based system.

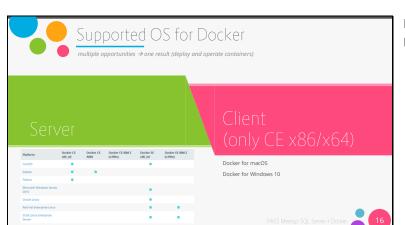
- Hackers who want to customize or patch their Docker build
- System engineers or integrators building a container system
- Infrastructure providers looking to adapt existing container systems to their environment
- Container enthusiasts who want to experiment with the latest container tech
- Open-source developers looking to test their project in a variety of different systems
- Anyone curious about Docker internals and how it's built

Moby is NOT recommended for the following use cases:

- Application developers looking for an easy way to run their applications in containers. We recommend Docker CE instead.
- Enterprise IT and development teams looking for a ready-to-use, commercially supported container platform. We recommend Docker EE instead.
- Anyone curious about containers and looking for an easy way to learn. We recommend the docker.com website instead

https://docs.docker.com/engine/installation/

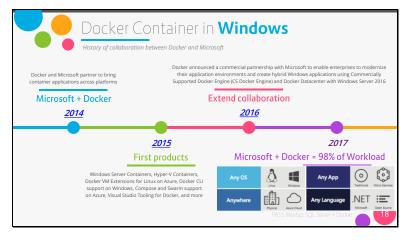




https://docs.docker.com/engine/installation/#deskto



https://hosting.1und1.de/digitalguide/server/knowhow/docker-alternativen-im-ueberblick/

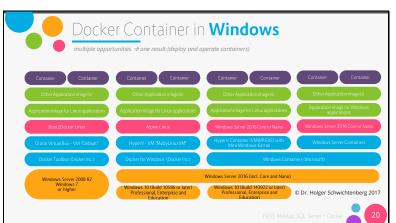


On October 15, 2014, <u>Microsoft</u> announced integration of the Docker engine into the next (2016) <u>Windows Server</u> release, and native support for the Docker client role in Windows.

On June 22, 2015, Docker and several other companies announced that they are working on a new vendor and operating-system-independent standard for software containers.

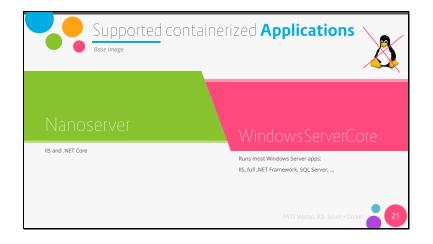
On June 8, 2016, Microsoft announced that Docker now could be used natively on Windows 10 with Hyper-V Containers, to build, ship and run containers utilizing the Windows Server 2016 Technical Preview 5 Nano Server container OS image.

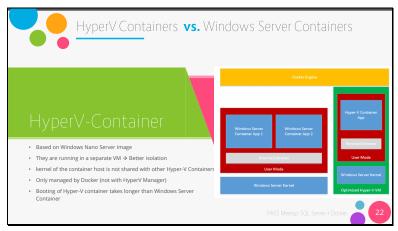




# https://heise-

de.imgix.net/developer/imgs/06/2/2/1/5/7/0/3/1\_D ocker\_Varianten-7fc897bd4250e6d9.png
If your system does not meet the requirements to run Docker for Windows, you can install <u>Docker Toolbox</u>, which uses Oracle Virtual Box instead of Hyper-V.
The current version of Docker for Windows runs on 64bit Windows 10 Pro, Enterprise and Education (1511 November update, Build 10586 or later). In the future we will support more versions of Windows 10. Make sure you have 14393.222 or later before continuing





http://www.techsupportpk.com/2016/10/how-to-create-hyper-v-containers-in-windows-server-2016.html

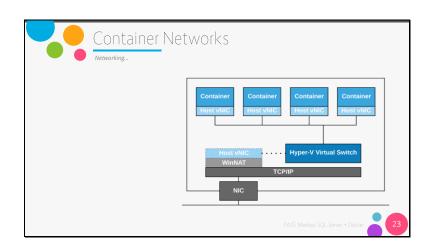
**Windows Server Containers** 

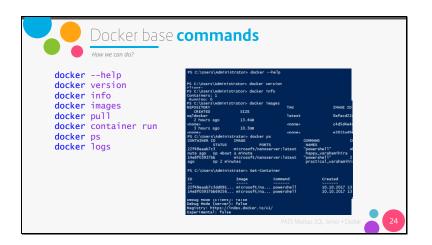
Isolation provided through namespace, resource & process isolation

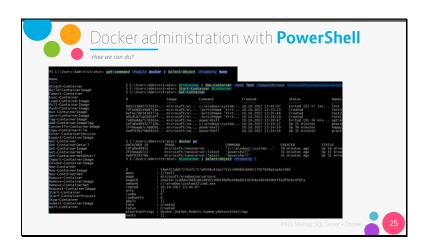
Share kernel with host and other containers

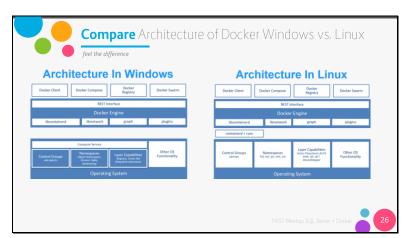
Hyper-V containers

Expand on isolation from windows server containers Run inside of a specialised virtual machine Does not share a kernel with host Specified at runtime with --isolation=hyperv switch

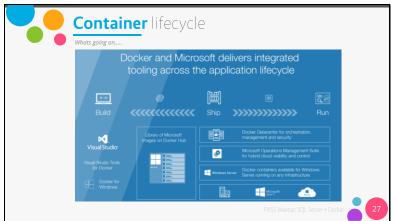




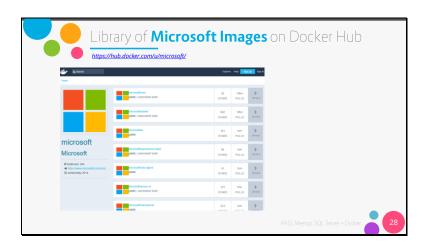


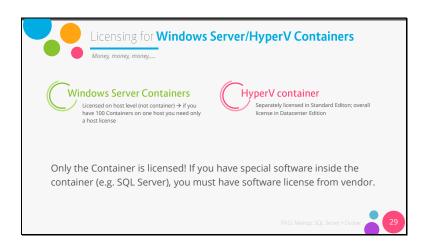


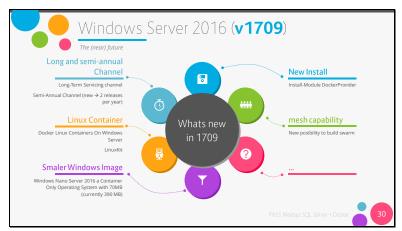
http://storage.googleapis.com/xebia-blog/1/2017/03/windows-vs-linux-architecture1-1024x342.png



https://i2.wp.com/blog.docker.com/wp-content/uploads/6a3fe5f7-1f87-447e-ae08-70f09c585b42.jpg?fit=1600%2C896&ssl=1



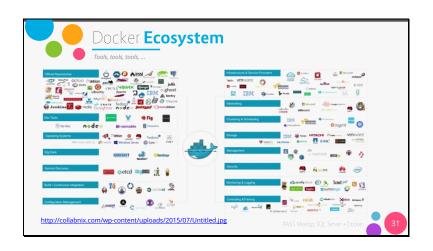


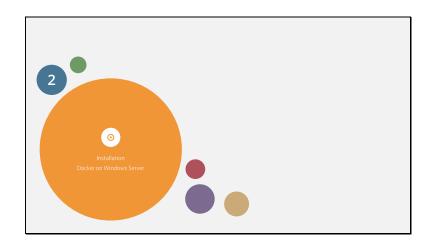


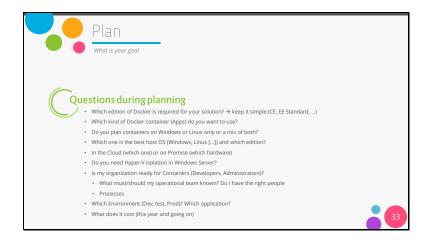
Long-Term Servicing channel – This Is the current and most familiar module where a new operating system version is released every 2-3 years with 5 years support cycle.

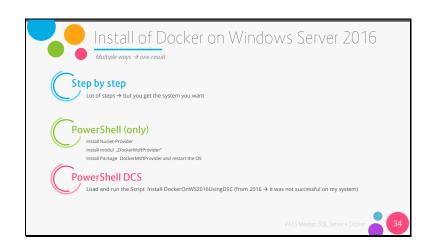
Semi-Annual Channel – This new module will offer two new releases every year with new features and roles, support will be available for 18 months.

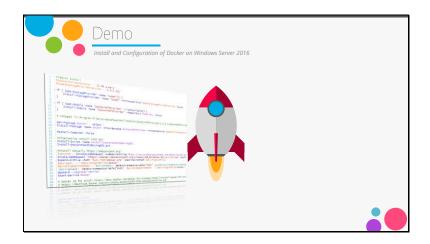
Docker Linux Containers On Windows Server – this feature will ship with version 1709 as a preview option only and probably will go GA 6 months later. Linux containers are available using the new LinuxKit released a few weeks ago.

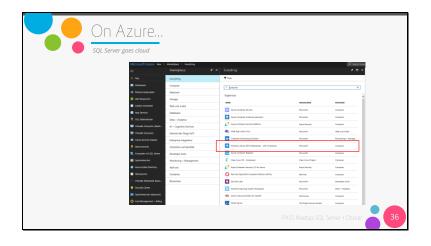


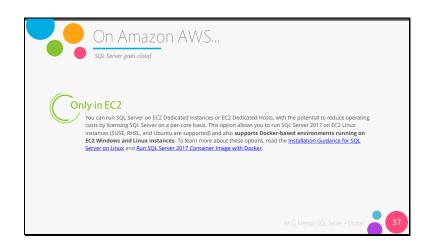


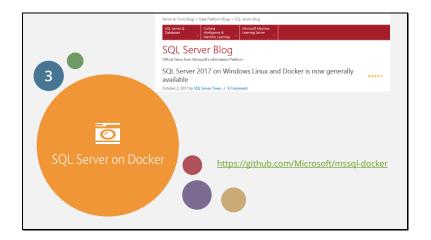


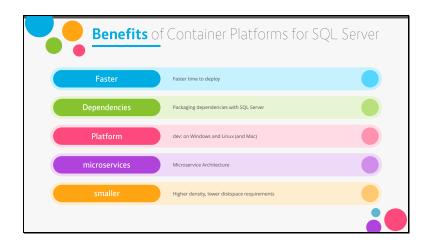














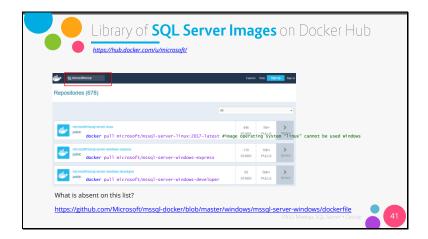
SQL Server 2017 is being released simultaneously for Windows and various flavours of Linux: Red Hat Enterprise Linux 7.3, SUSE Linux Enterprise Server 12, Ubuntu and Docker. 3.25GB or more of RAM is required. The official Docker image is based on Ubuntu 16.04.

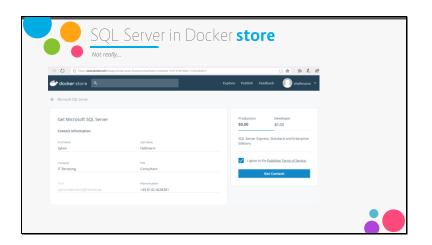
SQL Server supports Docker Enterprise Edition, Kubernetes and OpenShift container platforms. There are no support for Reporting Services or Analysis Services, nor Machine Learning Services (formerly R Services).

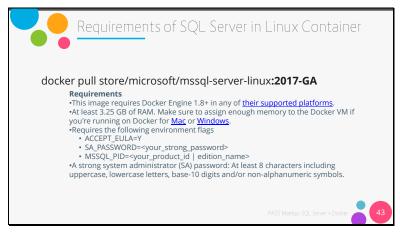
Replication is not supported. File Table, which exposes a SQL Server table as if it were part of the file system, does not work on Linux.

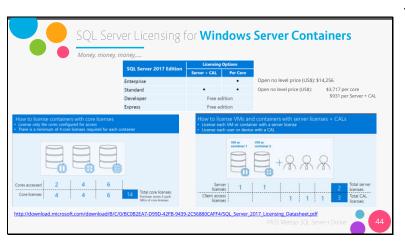
Management tools remain for the most part Windows only, though command-line tools work.













# How to use this image Start a mssql-server instance

docker run -e 'ACCEPT\_EULA=Y' -e
'SA\_PASSWORD=yourStrong(!)Password' -e
'MSSQL\_PID=Standard' -p 1433:1433 -d
store/microsoft/mssql-server-linux:2017-GA

# Start a mssql-server instance using the latest update

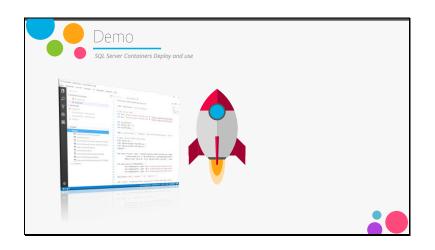
docker run -e 'ACCEPT\_EULA=Y' -e 'SA\_PASSWORD=yourStrong(!)Password' -e

'MSSQL\_PID=Standard' -p 1433:1433 -d store/microsoft/mssql-server-linux:2017-latest

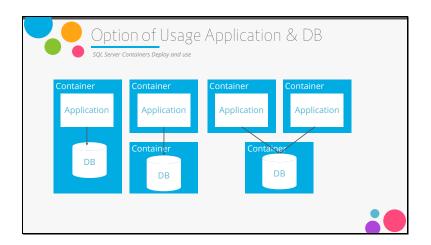
### **Connect to Microsoft SQL Server**

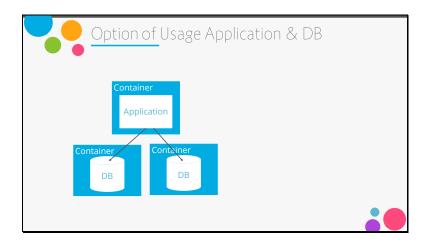
You can connect to the SQL Server using the sqlcmd tool inside of the container by using the following command on the host:

docker exec -it <container\_id|container\_name>
/opt/mssql-tools/bin/sqlcmd -S localhost -U sa -P
<your\_password>









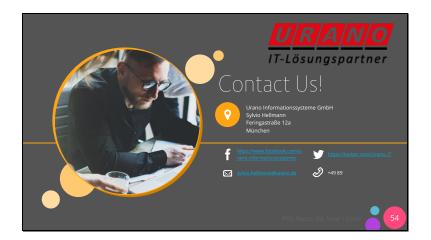


https://tr2.cbsistatic.com/hub/i/r/2016/06/06/c8427 cad-e219-4c7c-9f8f-5a8e7de11c86/resize/770x/dd1d47a9046c5e47f9572 fdb27c7e409/redservers.jpg









# Thank You for Watching! Any Questions?

Sylvio Hellmann - http://sylvioh.wordpress.com