Agenda

- Linux Containers
- Docker
- Demo
Why Containers?
Challenges to Address

Developers
• Frequent release vs. staged production schedule
• “It works on my machine”

Operations
• Managing growing services, from virtual to cloud
• Reliability and uptime when adding new codes
• Time to market, agility and efficiency

New features; faster please!
Linux Containers

• Lightweight virtualization
  - Faster provisioning, less downtime
  - Higher virtualization density

• Flexibility and agility
  - Containerized apps can be deployed anywhere
  - Normal I/O, no congestion

• Near native performance
  - IBM research: http://ibm.com/Search/?q=rc25482
Linux Containers
Traditional virtualization

Virtual Machine

- App A
- Bins/Libs
- Guest OS

- App A'
- Bins/Libs
- Guest OS

- App B
- Bins/Libs
- Guest OS

- App B'
- Bins/Libs
- Guest OS

Hypervisor (Type 2)

Host OS

Server
What is a Linux Container?

Container 1
- Kernel namespaces
  - Apps
- cgroups

Container 2
- Kernel namespaces
  - Apps
- cgroups

Kernel

Server
Advantages of Linux Containers

- Lightweight virtualization solution
  - Isolated from the other processes
  - 1 kernel to rule them all
  - Normal I/O
  - Dynamic changes possible without reboot
  - Nested virtualization is not a problem
  - No boot time or very short one
- Isolate services (e.g. web server, ftp, ...)
- Provide root read-only access
  - Mount host / as read-only
  - Add only needed resources read-write
Linux Containers Use Cases

• Deploy everywhere quickly
  – Deploy application and their dependencies together.

• Enterprise Data Center
  – Limit applications which have a tendency to grab all resources on a system:
    – Memory (databases)
    – CPU cycles/scheduling (compute intensive applications)

• Outsourcing business
  – Guarantee a specific amount of resources (SLAs!) to a set of applications for a specific customer without more heavy virtualization technologies
Linux Containers – Limitations

• They cannot run a different OS/architecture
  – Cannot run Windows containers on Linux

• Risk of escaping from containers
  – Solution: user namespaces

• Shared kernel with the host
  – Syscall exploits can be exploited from within the container
  – Solution: seccomp2 (in Linux kernel since 3.5)
Linux Containers – Security

• Do not give root privileges unless needed
• Apply security patches both on the host and on inside of the container
• Drop Kernel capabilities that are not used
• Secure containers with SELinux, AppArmor
  – SELinux policy applies to complete container
  – Support for SELinux with LXC on a case by case basis
  – AppArmor support is ready upstream
• Paranoid? Run the containers inside of a VM
Docker
Why Docker?

- Shipping applications everywhere
- Repository of images
  - https://registry.hub.docker.com/
  - Private repository possible
- Workflow for containers like git
  - Commits; push / pull
  - DevOps oriented
- Better disk usage: changes layers
- Easy to build new images
- Allows for image versioning
Docker

Containers
- App A
- App A'
- App B
- App B'
- YaST
- Docker Daemon

Host OS

Server
Speak Like Docker

• Registry
  On-line storage for docker images

• Repository
  Bag containing several versions of an image

• Image
  Prepared system to run in a container

• Container
  Linux container running a docker image
Docker at SUSE
Official images

• Pre-built images ready to be download
• Built from trusted sources
• Actively maintained by SUSE
• Available for different architectures
• Can be audited and inspected with tooling made by SUSE
YaST module

- Simple solution to get started with Docker
- Manage the available Docker images
- Run Docker images
- Control of running containers
Portus

• Authentication: control access to your images
• Easy of use: navigate and search your catalog of images
• Collaboration: organize your users with teams
• Auditing: keep everything under control
What's Next – SLES 12 SP1

- Portus fully supported
- Patch and update of images/containers
- OpenStack integration
- Support for IBM Power and System z
- Docker Security
Outlook

- Patch and update UI and integration
- Minimal OS (JeOS)
- Orchestration
- Physical to Docker migration
Docker from SUSE, Fully Supported

Enterprise-ready
- Images from trusted source (repository)
- Full control over your data: on-premise registry, authentication
- Pre-built Docker images

Operational Efficiency
- Complementary virtualization of Xen/KVM
- Btrfs support
- Higher virtualization density

Easy-to-use tools
- YaST interface
- sle2docker, zypper-docker
- Portus
Learn More

• We listen! Join our Docker beta program:

• Docker mini-course videos
  • https://www.suse.com/promo/sle/docker.html

• Try SUSE Linux Enterprise Server 12
  • https://www.suse.com/products/server/download/

• SUSE Docker QuickStart
  • https://www.suse.com/documentation/sles-12/singlehtml/dockerclick/dockerquick/dockerquick.html

• More information in SUSE Linux Enterprise 12
  • https://www.suse.com/promo/sle12.html
It's Demo Time!

Thank you.
Docker at SUSECon 2015

**TUT19930 - Docker & Portus : A Winning Duo for Your Infrastructure**

- Tue, Nov 3rd, 3:15 PM – 4:15 PM
  5 Roland Holst kamer

**HO19929 - Hands on session on Docker**

- Wednesday, Nov 4th, 2:15 PM - 4:15 PM
  B-Keurzaal
- Thursday, Nov 5th, 9:00 AM - 11:00 AM
  B-Keurzaal
BACKUP
Unpublished Work of SUSE LLC. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.