SUSE

How to use a purpose built, lightweight and immutable OS with IBM z/VM

Meet the presenter

**Mike
Friesenegger**

IBM Alliance
Solutions Architect

mikef@suse.com
+1 303 249 0817

Agenda

1. Immutable OS

2. SLE Micro today

3. The future

"The operating system is a commodity."

What does an OS need to provide in order to really become a commodity?

Core Operating System Requirements

Commodity is not a given, it needs to be earned.

Easy mass deployment

- Pre-configured images with easy configuration during deployment
- Reliable automated installation framework

Low maintenance efforts

- Automatic health-checks
- Automatic system tuning
- Automated recovery in case of error

Stability

- Built from a reliable codebase
- Reliable lifecycle with best-in—class support
- Bugfixes and security updates
- Secure Software Supply Chain

Security

- Pre-hardened
- Confinement framework plus attestation
- Means to apply (security) updates
- Security Certifications

What is SLE Micro

Lightweight and Secure OS Platform for Containers and VMs

Small Footprint, Enterprise-Grade

Lightweight immutable OS that's also optimized for embedded devices, edge use cases and industrial IoT.

Mission-critical

Immutable design and security features ensure reliable run time.

Prepared to support very long product lifecycles.

Low Maintenance

Our goal is to deliver **zero** maintenance infrastructure.

Developers love SLE Micro - 100% open source, modular architecture and easy to customize.

Perfect for Containers & Kubernetes

Built from ground up to support containers and microservices.
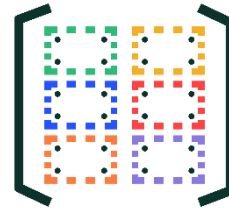
SLE Micro helps you run your workloads optimally

Container workloads with or without Kubernetes | Workloads in virtualized environment

Single-Node Container Host
Run container workloads in a non-Kubernetes environment.

Single-Node KVM Virtualization Host
Run workloads on Immutable OS

Kubernetes Cluster Node
Run Kubernetes workloads on Immutable OS

Public Cloud
Use Immutable OS for containerized or virtualized workloads in public cloud infrastructure

SLE Micro enables you to compute anywhere.   Edge | Embedded | Industrial IoT

SLE Micro - Key Technology & Business Benefits

Solid foundation for modernization - enabling containerization and virtualization

Gain competitive advantage
Quicker time to market

Low Maintenance

Reduce operational expenses
Improve productivity

Open standards

Keep control of technology roadmap
Avoid locked-down stacks
Unlock full TCO savings
- Hardware – use h/w of choice
- Software – use open-source tools of choice. Build your own open stack.

SLE Micro: Core Features

— **Immutability**: Read-only root filesystem with transactional updates

— **Reliability:** rollback to old snapshot, health check daemon

— **Hardening:** fully supported SELinux (framework + policy), minimal OS footprint

— **Manageability:** Cockpit web-based management (optional install)

— **Realtime:** Realtime kernel included on Intel & AMD 64 (optional install)

Architectures / Installation Scenarios

Architectures

- Intel & AMD 64, including RT-Kernel

- Arm 64bit

- IBM Z and LinuxONE

Installation Methods

- Manual installation using YaST2
- Automated installation AutoYaST / YOMI
- Image based deployment

Installation Modes

- Bare metal installation (manual or image based)

- KVM Host

- Virtual Machine

SLE Micro is either a container host or a KVM VM host. Workloads are to be run within containers or VMs.

SUSE is providing images for public cloud marketplaces.

Image Based Deployment

Manual or automated installation from media using YaST / AutoYaST

Deploy pre-configured image to storage device and boot using Ignition config file for OS configuration

Boot system from self-deploying image and configure OS using Ignition config file

Ignition and Combustion for initial system configuration

- Ignition does the initial system configuration for the OS (password, hostname, network etc.)
- Requires configuration file: https://documentation.suse.com/sle-micro/5.4/single-html/SLE-Micro-deployment/#cha-images-ignition
- Web based Ignition configuration:

https://opensuse.github.io/fuel-ignition/

Innovating your IBM z/VM infrastructure

- On-going collaboration on rapid deployment of SLE Micro and Rancher on IBM zSystems and LinuxONE systems running z/VM.

DEMO: Deploying a SLE Micro image to an IBM z/VM guest

SLE Micro image deployment host

IBM z/VM guest

Boot disk – DASD | FBA

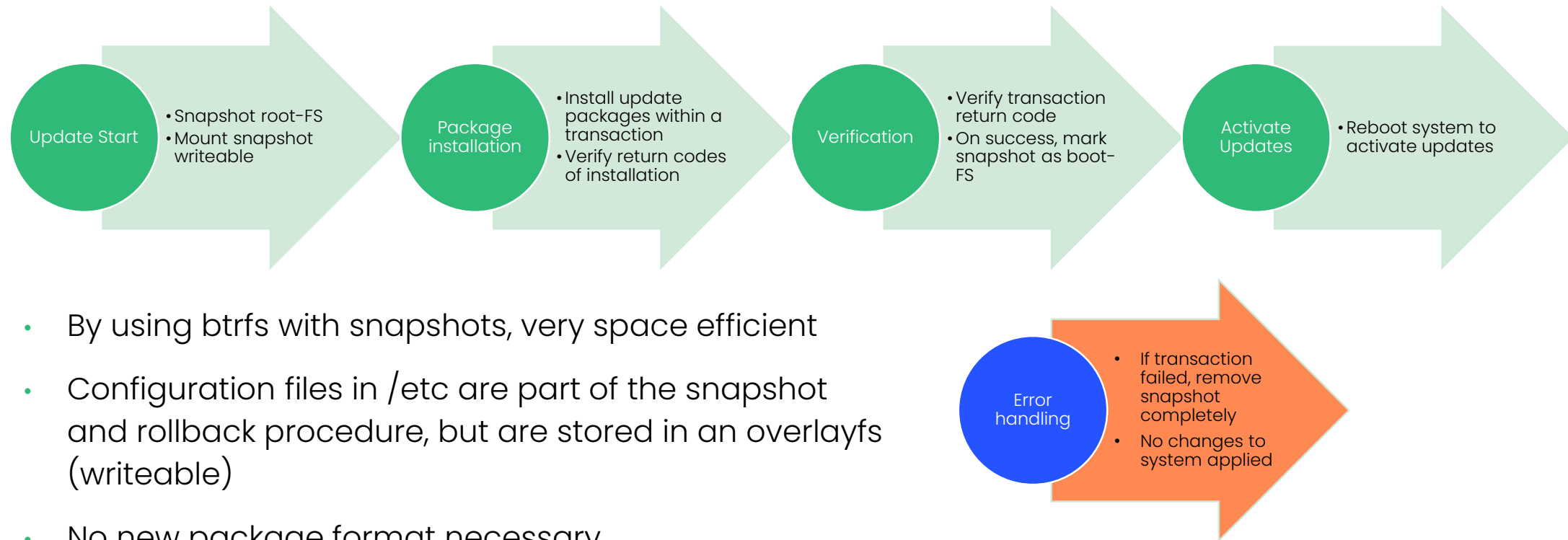Ignition / Combustion device – DASD | FBA | VDISK

Deploy pre-configured image to storage device and boot using Ignition config file for OS configuration

- Setup image deployment host

- Prepare Ignition/Combustion files

- Define z/VM guest

- Configure Ignition drive of z/VM guest

- Write SLE Micro image to z/VM guest

Transactional Updates

Applying updates to an immutable OS in an atomic way.

Update Start
- Snapshot root-FS
- Mount snapshot writeable

Package installation
- Install update packages within a transaction
- Verify return codes of installation

Verification
- Verify transaction return code
- On success, mark snapshot as boot-FS

Activate Updates
- Reboot system to activate updates

Error handling
- If transaction failed, remove snapshot completely
- No changes to system applied

- By using btrfs with snapshots, very space efficient

- Configuration files in /etc are part of the snapshot and rollback procedure, but are stored in an overlayfs (writeable)

- No new package format necessary

- No size limitation for partitions/OS

- Easily enhanceable

Core features in SLE Micro for s390x

More Secure, Faster to Deploy, Integrated with SUSE Rancher

Kubernetes compatible network stack
- NetworkManager integration enables management via k8s (nmstate.io)

Kernel Live Patching
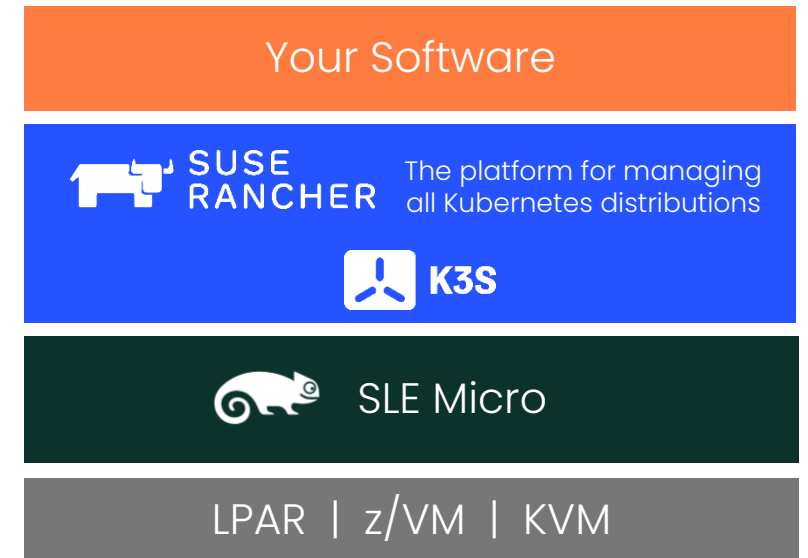- Enables continuous operation of an Immutable OS in a secure fashion

Enhanced web-based management
- 1:1 web-based system management through integrated Cockpit stack

Support for HA deployments
- keepalived and haproxy offer HA (High Availability) functionality in edge deployments

Example Stack Deployment

Your Software
SUSE RANCHER — The platform for managing all Kubernetes distributions / K3S
SLE Micro
LPAR | z/VM | KVM

Core features in SLE Micro for s390x

More Secure, Faster to Deploy, Integrated with SUSE Rancher

Integrated SELinux
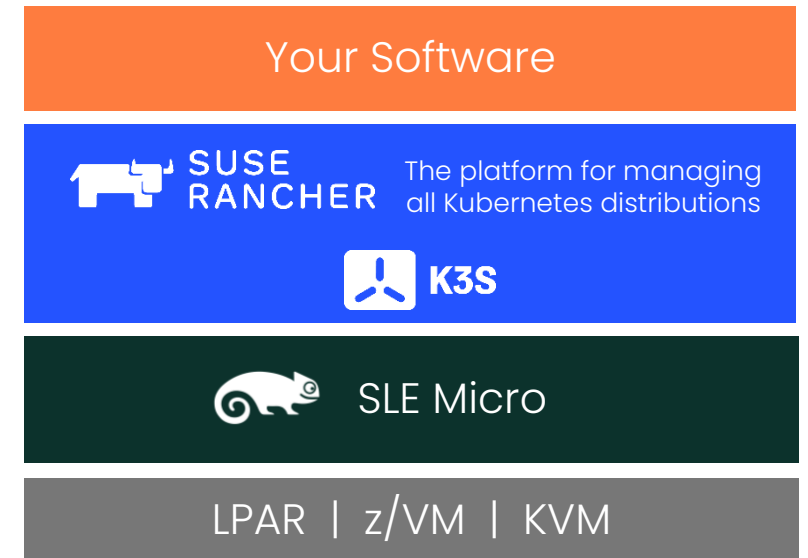- Fully secure your infrastructure already during installation

Remote attestation
- Compliant operation of your infrastructure

Security Hardening
- SLE Security Modules are listed at NIAP for under evaluation for FIPS 140-3 certification. SLE Micro 5.3 is currently being evaluated for Common Criteria NIAP and Common Criteria EAL4+ certification. This will enable most secure organizations to incorporate SLE Micro.

Example Stack Deployment

Your Software
SUSE RANCHER — The platform for managing all Kubernetes distributions / K3S
SLE Micro
LPAR	z/VM	KVM

DEMO: Deploying a SLE Micro image to an IBM z/VM guest

SLE Micro image deployment host

IBM z/VM guest

Boot disk – DASD | FBA

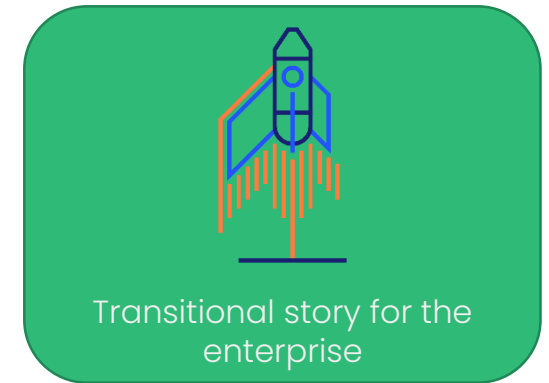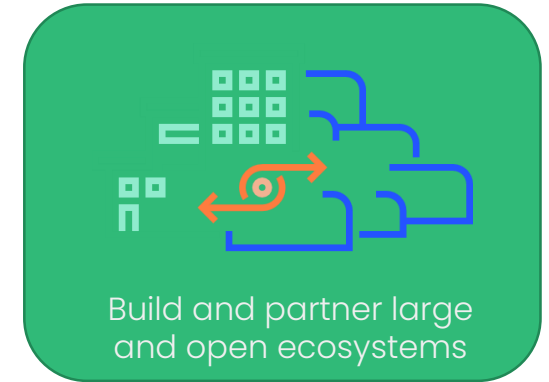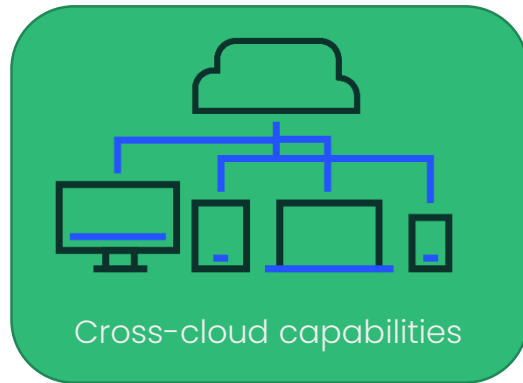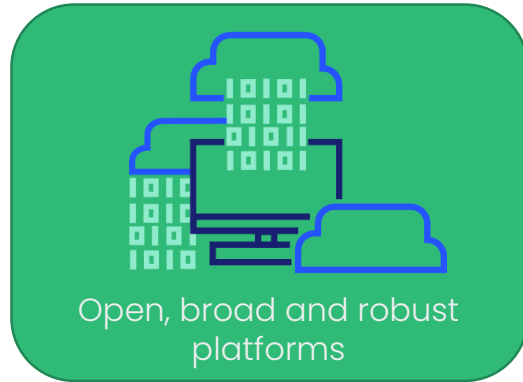Ignition / Combustion device – DASD | FBA | VDISK

Deploy pre-configured image to storage device and boot using Ignition config file for OS configuration

- Setup image deployment host

- Prepare Ignition/Combustion files

- Define z/VM guest

- Configure Ignition drive of z/VM guest

- Write SLE Micro image to z/VM guest

- Boot the z/VM guest

The Future of the Immutable OS

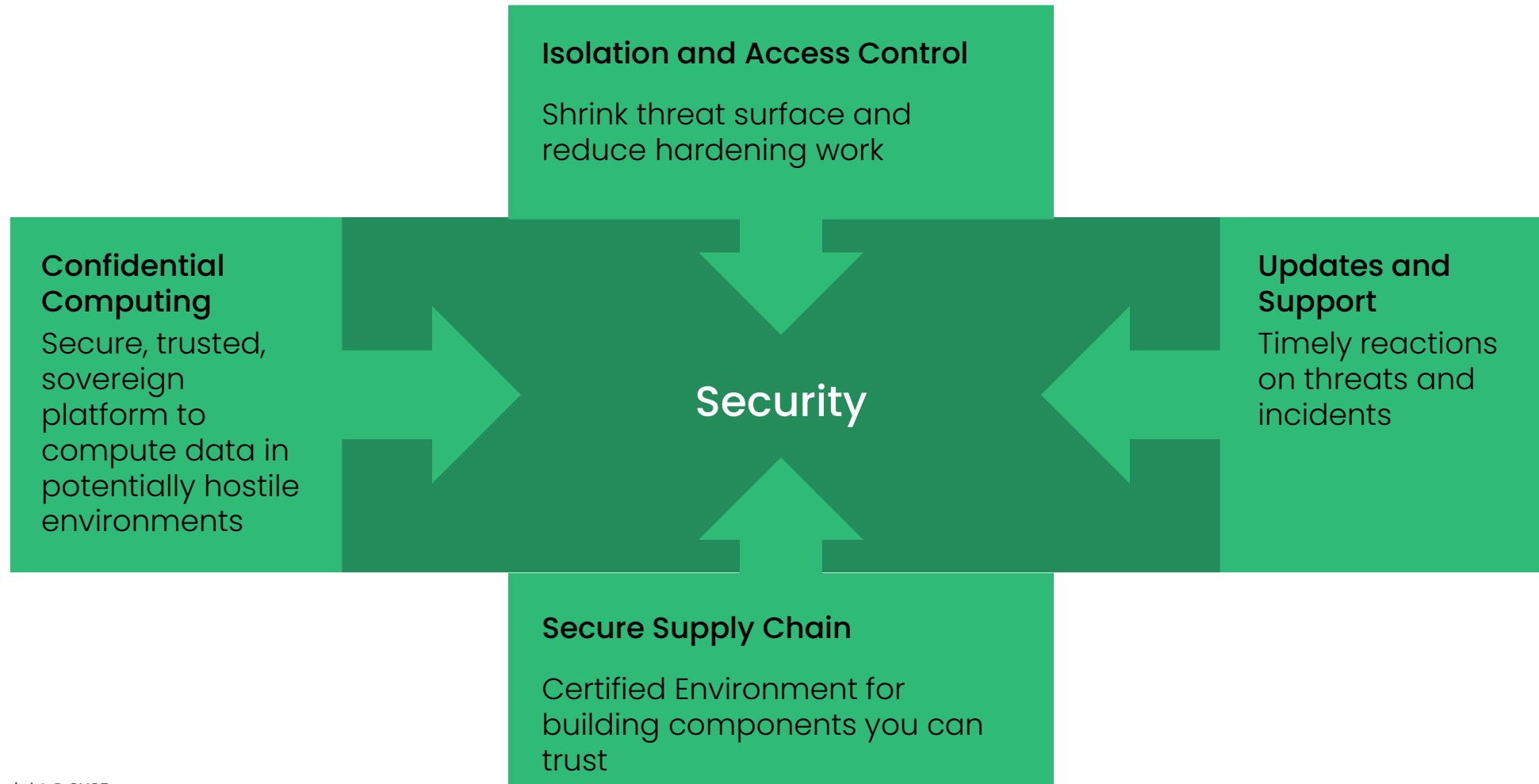# Digital Transformation

Open, broad and robust platforms

Cross-cloud capabilities

Customers

Community

Employees

Partners

Transformational Assistance

Build and partner large and open ecosystems

Transitional story for the enterprise

Bringing the pieces together

Security

Isolation and Access Control

Shrink threat surface and reduce hardening work

Confidential Computing

Secure, trusted, sovereign platform to compute data in potentially hostile environments

Security

Updates and Support

Timely reactions on threats and incidents

Secure Supply Chain

Certified Environment for building components you can trust

Self-everything - Zero Touch Automation

Reduce "Overhead", concentrate on essentials

Self-Healing
The system recovers from any damaging changes to the OS, including pathing and configuration changes

Self-Management
Automatic updates and adaption ensure a higher uptime of workloads

Zero-Touch
Reducing manual interactions frees the admins to concentrate on improvements rather than 'keep-the-lights-on'.

Self-Tuning / Self-Optimized
Self-optimization dependent on workloads increase performance and lower TCOs.

Self-Explaining
Explanatory help-texts at the place where they are needed help to avoid wasting time searching and make changes easier for the admins.

Life Cycles

Flexibility for the parts

— **Updates and Life Cycles: Granularity**
Compartmentalization allows for more flexibility when it comes to updates of applications, run-time environments and development tools without breaking other parts of the systems.

— **System Life Cycles: Duration**
Separation of Applications and Host as well as the compartmentalization enables customers to get longer support for a specific component.

— **Hardware changes made easy**
Update your hardware and change it without impacting the run-time environment needed for your workload and your ISV applications.

Workload-centric

"Get The Job Done!"

— **Environment agnostic**
Running workloads unchanged on-premises, off-premises, bare-metal or with a cloud provider. Don't worry where you will be in a few years, your workload will be there with you.

— **Footprint reduction**
Install only what you need. And be able to remove it again without the fear of breaking something else.

— **Reduce 'background-noise'**
Disentangling the Application layer from the Host system makes it easy for the admins to concentrate on the layer important for them - where their workloads run.

— **Enable the 'right' stacks**
Compartmentalization helps to have several stacks run in parallel, without affecting unintentionally parts of the system.

When done right, being commodity is an award.

DEMO: Deploying a SLE Micro image to an IBM z/VM guest

SLE Micro image deployment host

IBM z/VM guest

Boot disk – DASD | FBA

Ignition / Combustion device – DASD | FBA | VDISK

Deploy pre-configured image to storage device and boot using Ignition config file for OS configuration

- Setup image deployment host

- Prepare Ignition/Combustion files

- Define z/VM guest

- Configure Ignition drive of z/VM guest

- Write SLE Micro image to z/VM guest

- Boot the z/VM guest

- Access the SLE Micro Web console

- Verify the workload (Kubernetes – Rancher K3s)

Enter for a chance to win 3 Bluetooth speakers...

- Go to **suse.com**

- Click on **Free Downloads**

- Find **SUSE Linux Enterprise Micro** download tile

- Download **SLE-Micro.s390x-5.4.0-Default-dasd-GM.raw.xz**

- Take a screenshot **showing the download in progress**

- Send the screenshot to **mikef@suse.com** to enter the drawing

By June 27, 2023 11:59pm ET

SLE-Micro.s390x-5.4.0-Default-dasd-GM.raw.xz

File Size
747468276

Checksum
SLE-Micro.s390x-5.4.0-Default-dasd-GM.raw.xz.sha256

Signature
SLE-Micro.s390x-5.4.0-Default-dasd-GM.raw.xz.sha256.asc

SBOM in SPDX 2.0 format
SLE-Micro.s390x-5.4.0-Default-dasd-GM.raw.xz.spdx.json

Download

Filename
SLE-Micro.s390x-5.4.0-Default-kv

SLE-Micro.s390x-5.4.0-Default-fba-GM.raw.xz

File Size
792023044

Checksum
SLE-Micro.s390x-5.4.0-Default-fba-GM.raw.xz.sha256

Signature
SLE-Micro.s390x-5.4.0-Default-fba-GM.raw.xz.sha256.asc

SBOM in SPDX 2.0 format
SLE-Micro.s390x-5.4.0-Default-fba-GM.raw.xz.spdx.json

SBOM in CycloneDX format
SLE-Micro.s390x-5.4.0-Default-fba-GM.raw.xz.cdx.json

Instructions
How do I do this?

Download

Products   Solutions   Support   Partners   Communities   About   **Free Downloads**

Google Chrome   1 min remaining
Downloading
SLE-Micro.s390x-5.4.0-Default-dasd-GM.raw.xz to
Downloads

Thank you

SUSE

For more information, contact SUSE at:

+1 800 796 3700 (U.S./Canada)

Frankenstrasse 146

90461 Nürnberg

www.suse.com