

Cloud in a box

Fully automated installation of
SUSE® Openstack Cloud 5 on Dell VRTX

Lars Everbrand

Software Developer

lars.everbrand@ericsson.com



From



To



Introduction

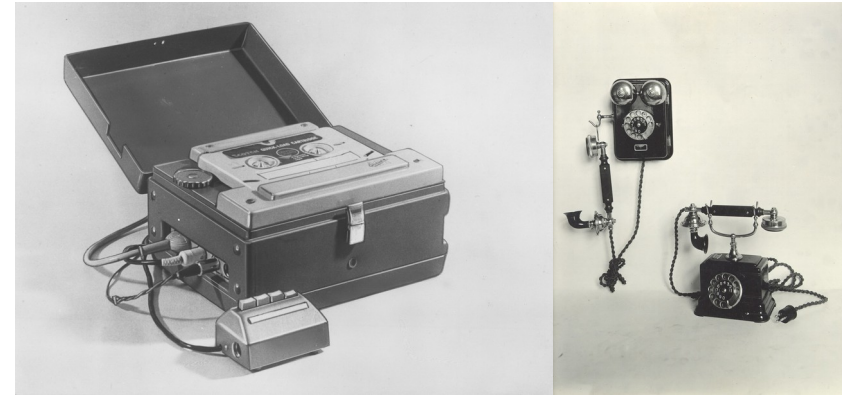
- **ERICSSON** 

- Disclaimer

- All views expressed are my own
- SUSE Openstack Cloud is **not** Ericsson Cloud

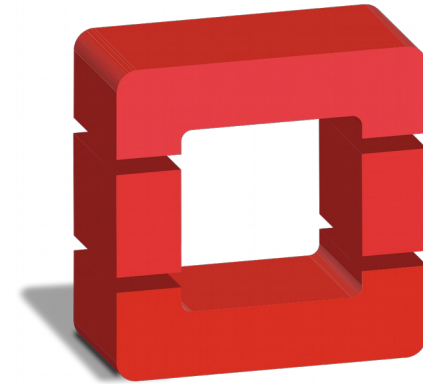
- Background

- SUSECON 2014
 - USB with SUSE Openstack Cloud
- “SUSE Rules the Stack” (9 minute Open Stack installation)
 - <https://www.suse.com/company/press/2015/suse-rules-the-stack-yet-again-at-openstack-summit.html>



Our need

- Test of Linux based products
 - Node bring-up
 - Cloud readiness
- Cost
- Simple installation
- Deterministic and **reproducible** environment
 - Stable OEM reference platform
 - “Return to factory defaults”
- Performance and capacity not a factor
- Installation from USB for “offline” use case



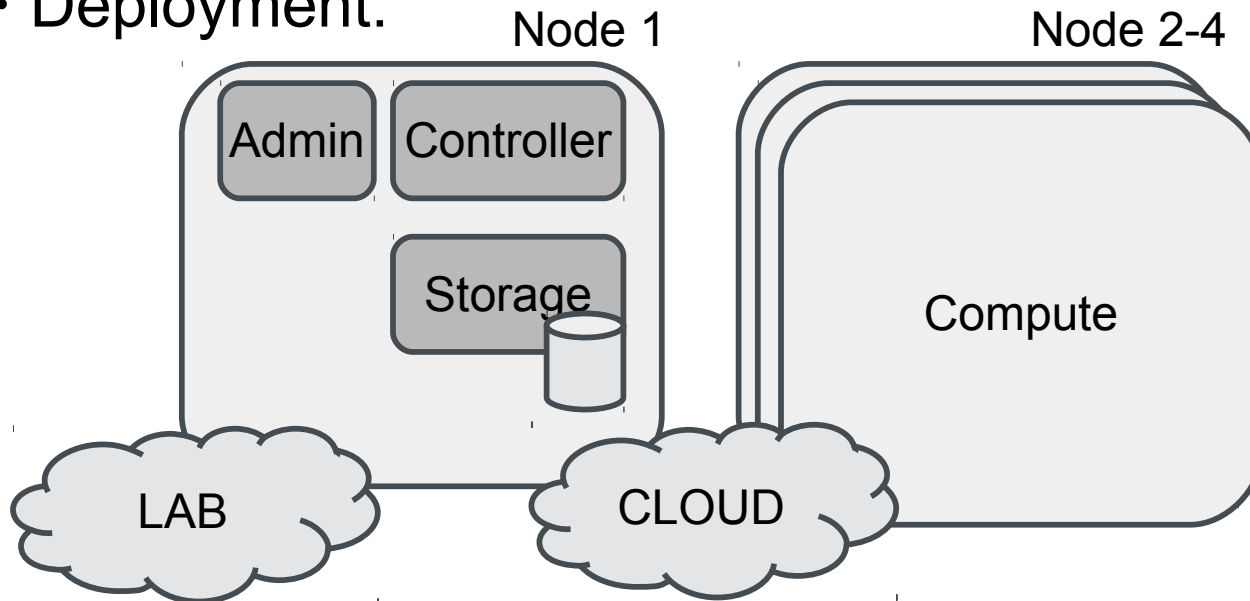
Proof of concept hardware

- Hardware kept symmetric
 - Flexibility
 - Less configurations
- Dell PowerEdge VRTX
 - 4 M630 blades
 - 2 x Intel Xeon
 - 64 Gb RAM
 - Local disk (300 GB)
 - Chassi has 11 “shared”/flexible disk

Planned deployment

- Prioritize compute
 - Hardware deployment of admin, controller steals valuable compute resources.
 - Virtualize all “infrastructure” on one blade

- Deployment:



First SUSE Cloud experience

- Documentation
- Repositories and ISOs
- Network configuration
- Fail and retry, **from scratch**

Automation

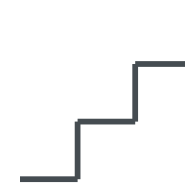


- Originally planned tools
 - autoyast
 - scripting
- Main procedures needed:
 - Power and boot control for blades (idrac/IPMI)
 - Admin node installation and configuration
 - Sequencing of nodes in SUSE Cloud
 - Configuration of SUSE Cloud
- Quite early it was apparent that some automation framework was needed, especially for prototyping

9pm

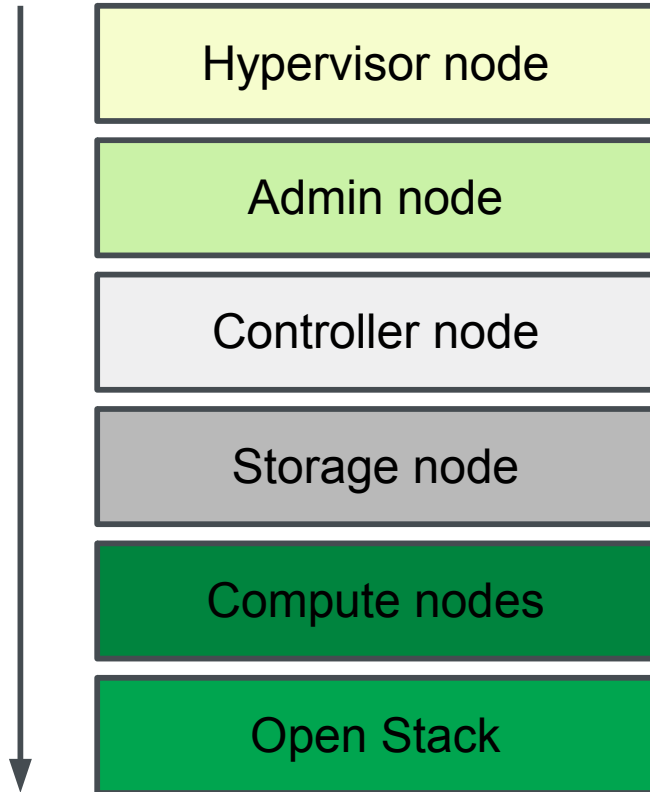
- 9pm (<https://github.com/rical/9pm>)
 - Simple and quick, just enough functionality

Automation stages



- During testing installation was network based for convenience
- Stage I
 - Manual
- Stage II
 - Semi-automated with 9pm and appliance
 - Network based
- Stage III
 - Fully-automated, custom images
 - Network or USB based

High level view



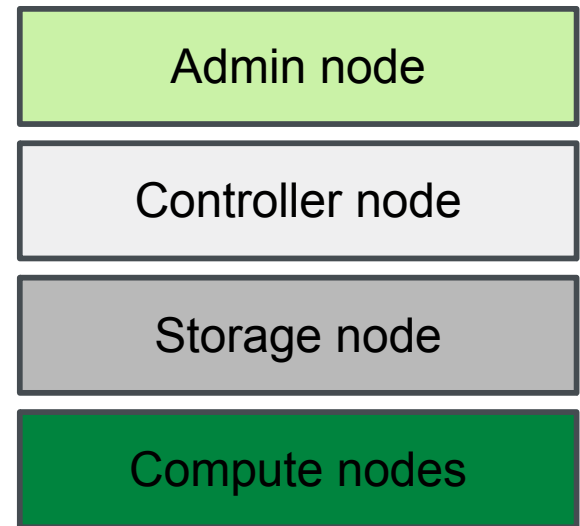
Configuration Hypervisor

Hypervisor node

- Standard SLES 12 installation
 - Minimized package selection
 - Minimal pattern
 - libvirt
 - openvswitch
 - openssh
- Local disk sliced with LVM for virtual machines
- Shared block device available for storage VM only
- Runs 3 VMs
 - Admin
 - Controller
 - Storage
- Available on lab network

Configuration SUSE Cloud

- Admin VM
 - SLES 11 SP3
 - Crowbar
 - Reachable via hypervisor (admin net)
- Storage VM
 - SLES 11 SP3
 - Glance
 - Cinder
- Controller
 - SLES 11 SP3
 - All other Open Stack services
- Compute
 - SLES 12
 - Running on hardware (node 2, 3 and 4)



Configuration Open Stack

Open Stack

- Configured via Crowbar on Admin node
- Configuration kept standard
 - Purpose of environment is to be able to run tests in different setup, e.g. openvswitch vs bridging
 - System reconfiguration is simple (installation from scratch)

DEMO

Hardware preparation

- Hardware installation
 - idrac configured
 - Shared disk configured (attached to slot-1)
- Switch configuration
 - Stage I: manual via web UI
 - Stage II, III: Scripted via 9pm and telnet connection to switch CLI

Hypervisor

Hypervisor node

- Autoyast install packages and enabled services
 - Small local disk (40GB), space for LVM
 - Minimal + openssh + libvirt + openvswitch
- Configuration (9pm)
 - openvswitch
 - Network configuration (wicked + openvswitch)
 - Add libvirt networks
 - Add logical volumes for Vms
- Image

Hypervisor, steps

Hypervisor node

- Stage I
 - autoyast, existing lab provisioning
- Stage II
 - autoyast, power and boot device automated via 9pm
- Stage III
 - Appliance built via kiwi which can be installed either via network or USB.
 - 9pm automation starts from hypervisor node
 - Scripts needs to be able to run over network or locally

Admin node

Admin node

- Transfer image over SCP (network installation) if needed
- Start image (virt-install for simplicity)
- Iptables for NAT
- Prompt for manual intervention
 - Needed due to SUSE OpenStack Admin Appliance (SUSE Studio) usage
 - Not needed for kiwi image
- Wait until crowbar state “ready”

Admin node, steps

Admin node

- Stage I
 - Installed manually (painful)
- Stage II
 - SUSE Studio Appliance started
 - NAT configured on hypervisor
 - First boot configuration over VNC
- Stage III
 - kiwi image
 - Repositories on hypervisor (NFS)

Controller & Storage

Controller node

Storage node

- Provisioned via crowbar/SUSE Cloud
 - crowbar/knife CLI commands
 - network boot

Controller & Storage, steps

Controller node

Storage node

- Stage I
 - Manual provision of VMs on hypervisor
- Stage II, III
 - Logical Volume for VM disk created
 - virt-install started via 9pm

Compute

Compute nodes

- Provisioned via crowbar/SUSE OpenStack Cloud
 - crowbar/knife CLI commands
 - network boot
- Hardware does **not** run on SLES 11 SP3 (without additional drivers)
 - sleshammer image in SUSE Cloud is SLES 11 SP3 based
 - DUD (Driver Update Disk) needed

Compute, steps

Compute nodes

- Stage I
 - Manual installation (autoyast) SLES 12
 - Join SUSE Cloud via crowbar_join script
- Stage II, III
 - Power/boot control via idrac
 - Add updated module to /updates and discovering-pre hook to load to enable SLES 11 SP3 to find the disk

OpenStack

OpenStack

- Stage I
 - Point and click in crowbar UI
- Stage II, III
 - Set node definitions according to configuration
 - Import barclamp configuration into crowbar using cli

OpenStack, steps

Open Stack

- Wait until all nodes are “discovered”
- Configure nodes via 'crowbar' CLI
 - Set node aliases (controller, compute1, ...)
 - Set roles
 - Set target OS (knife, json-edit)
 - Allocate nodes
- Wait until nodes “ready”
- Import barclamp configuration
- Wait until nodes “ready”

Cloud – build small or big?

- Big

- High availability
 - Open Stack services
 - Hardware
 - Facilities (power, cooling)
- Dimensioning
 - Network vs Compute

- Small

- Known building block unit (for scale out)
- HA not as important
- Local, distributed

Bugs/issues

- Some bugs found
 - Crowbar automation
 - Crowbar rename racy
 - kiwi bug, /run symlink
 - localhost = ::1
 - Dell iDRAC stability

Improvements

- Repositories
 - Point towards SMT and it should be done
 - Why is multiple versions RPMs of non-used software 'needed'
 - Firefox, X11, etc



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.

