

Journey to Containerized Applications: Uyuni Experiences

Don Vosburg Systems Management Enthusiast don@vosburgs.org Product Manager – SUSE Manager dvosburg@suse.com



Uyuni: upstream project for SUSE® Manager

- Community-supported
- Monthly meeting: Uyuni Community Hours (last Friday of the month, 4pm CET)
- Runs on openSUSE Leap
- Monthly releases
- Rolling-release
- Product import using CLI tool (vs wizard in SUSE Manager)
- All translations enabled by default (even community ones)

Join Us at uyuni-project.org



What is Uyuni?



- Scalable, multi-distribution systems management solution server and proxy
- Deploy and manage all kinds of workloads, wherever they are, from a single place
- Hardware and software inventories
- Configuration management: automatically maintain standard configurations
- Granular control over content delivery
- Large codebase that has grown over a long time

Where did Uyuni come from?

Origins: Spacewalk project



- Free & Open Source Systems Management
- Started in 2008
- Upstream of RH Satellite 5, SUSE Manager before 3.2
- Project shut down in May 2020



Uyuni – a fork of Spacewalk



Updated features from SUMA (downstream to upstream)

Things we have added/augmented

- Salt
- Scalability
- Usability
- React Web UI
- Python 3 and JDK11 codebase



Uyuni Under the Covers Key open source projects

- Apache, tomcat, openjdk, squid
- Postgresql
- Salt master, broker
- Prometheus/grafana



Uyuni Architecture





Why go to containers?

the state of

Why containers?



- Potential to deploy on other distros
- Modernize the offering
- Align with DevOps strategies
- Simplify deployment

Where do we start?



Where do we start - the right audience

- "Edge" deployments an easier win for containers
- Users looking for repeat, consistent installation
- Places where deploying traditional proxy servers (vm's) may prove difficult

Where do we start – the right subset

- Far too difficult and lengthy to containerize entire codebase
- Something that requires minimal deployment customization
- Addresses the "edge" use case
- Already has "stateless" elements

Where do we start? Proxy!



- Far too difficult and lengthy to containerize entire codebase
- Something that requires minimal deployment customization
- Addresses the "edge" use case
- Already has "stateless" elements



Lowering risk – retaining existing equivalent

- Keeping the previous VM form of proxy
- Allows users to add containerized proxy with new environments
- Lifecycle controls allow time to solidify offering without • sacrificing functionality

Building it out





Building it out – Base Container Image

- Started with SUSE Base Container Images
 - <u>https://www.suse.com/products/base-container-images/</u>
- Open, flexible, and secure container images
- Based on SUSE Linux Enterprise
- Familiar and aligned with existing offering

Building it out - Divide and Conquer



- Looked at the proxy delivery and how to intelligently split the load
- Five pods
 - salt-broker, httpd, tftpd, ssh, squid
- Each has its own function, so processes can be isolated
- Common configuration "form"



Bring it together part 1 - podman

- Chose podman as initial deployment tool
 - Included in our distro and direction
 - Integrated with systemd
- A single binary package rpm to get started
 - Podman as dependency
 - Installs and sets the systemd services



Bring it together part 2 - kubernetes

- Created a helm chart to deploy on k8s
- Additional instructions around clustered server •
 - Network via metallb •
 - Storage via local storage provider common mount point ٠

Other important decisions



- Choosing image repositories
- Build processes
- Storage
- Documentation

Hindsight is 20:20

What we learned later...



- QA and testing was a donut big holes
- Container image updates not automatic
- Release processes not CI/CD, timing
- New build processes caused some angst

Uyuni – Getting Started



Installation guide:

https://www.uyuni-project.org/uyuni-docs/en/uyuni/quickstart-uyuni/install-uyuni-

server.html

Runs on openSUSE Leap (currently 15.3)

alternative and the second of The same of the same to the same the sa Demo, 14 M R 8 5 8 AA P



Thank you!

(uyuni-project) /uyuni-project **/**UyuniProject

10

Join Us at uyuni-project.org

Ö

License

This slide deck is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. It can be shared and adapted for any purpose (even commercially) as long as Attribution is given and any derivative work is distributed under the same license.

Details can be found at https://creativecommons.org/licenses/by-sa/4.0/

General Disclaimer

This document is not to be construed as a promise by any participating organisation 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. openSUSE 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 openSUSE products remains at the sole discretion of openSUSE. Further, openSUSE 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 openSUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC, in the United States and other countries. All third-party trademarks are the property of their respective owners.

Credits

Template & Design Pau Garcia Quiles

Inspiration openSUSE Design Team http://opensuse.github.io/brandingguidelines/

Heterogeneous Linux environments

Interoperability unmatched!



Uyuni delivers support for these clients:



Recently added support for:

- Amazon Linux 2
- Alma Linux 8
- Rocky Linux 8





Uyuni features you can use!



- Transparent integration with Salt
- Manage on-prem, cloud, hybrid cloud or multi-cloud systems
- Content Lifecycle Management: define stages (DEV, TEST, PROD) for your software channels and apply filters to add/remove contents and create new channels
- Recurring actions
- Build OS (kiwi) and container images (docker)
- Compliance: CVE audit, SCAP, subscription matching (SUSE clients)
- Libvirt virtualization management
- Monitoring (Prometheus & Grafana stack), including federation
- Formulas with Forms: create YAML automation templates, no programming skills required!