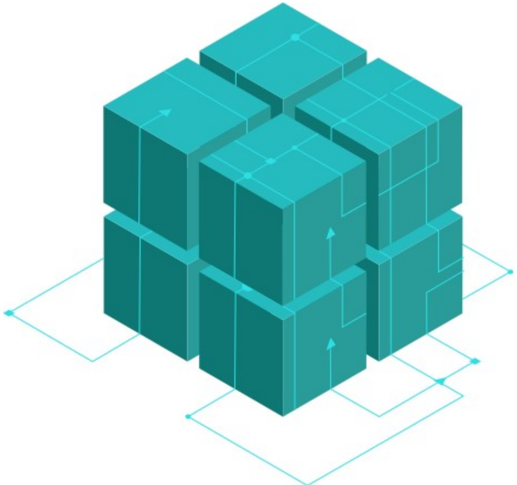
UYUNI

How to transform a server application to a “macroservice”

How we did it in Uyuni



Michele Bussolotto
Software Engineer
mbussolotto@suse.com

About me

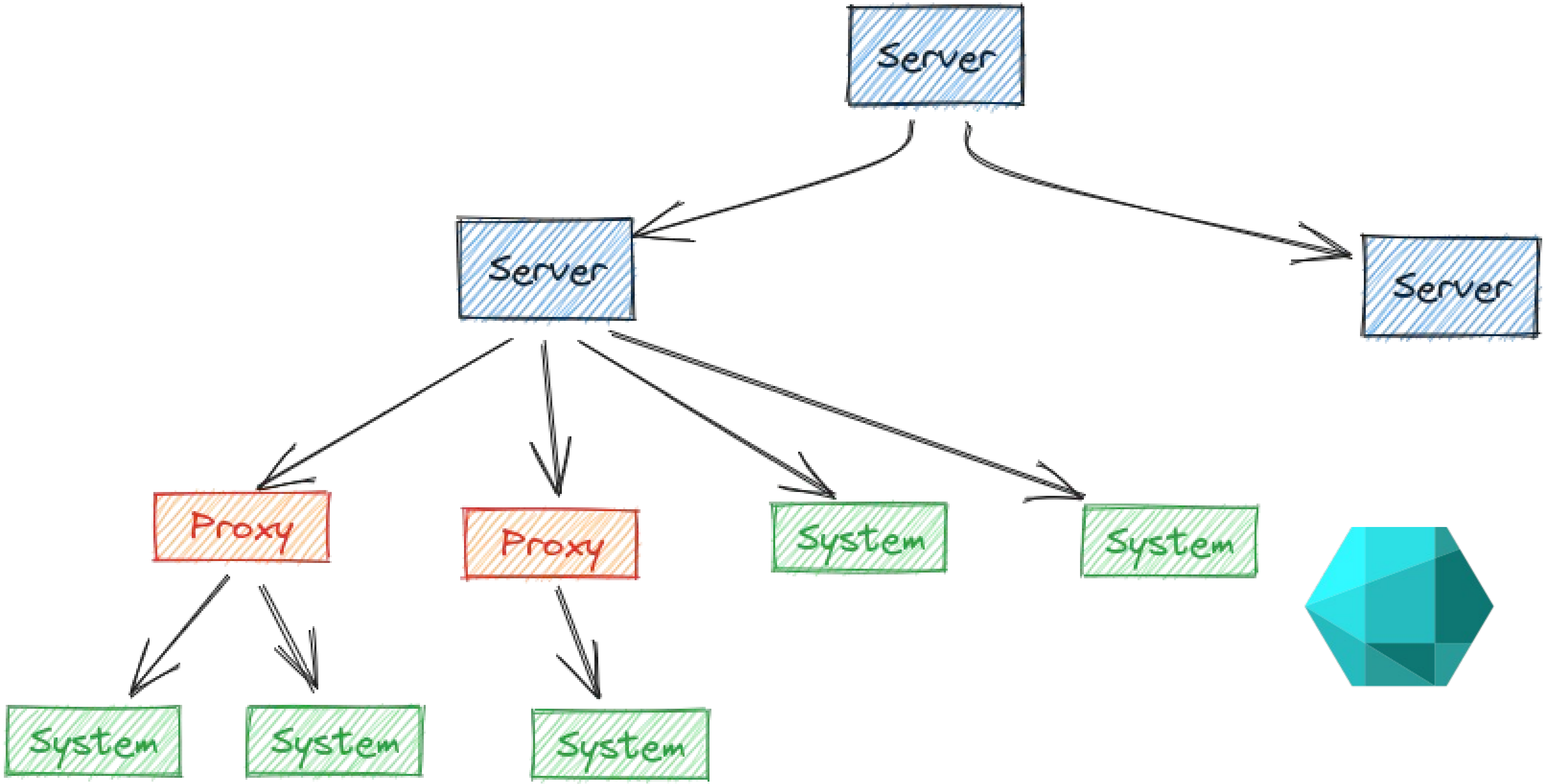
Michele Bussolotto

Software Engineer @SUSE

Uyuni / SUSE Manager contributor
Tomcat Maintainer

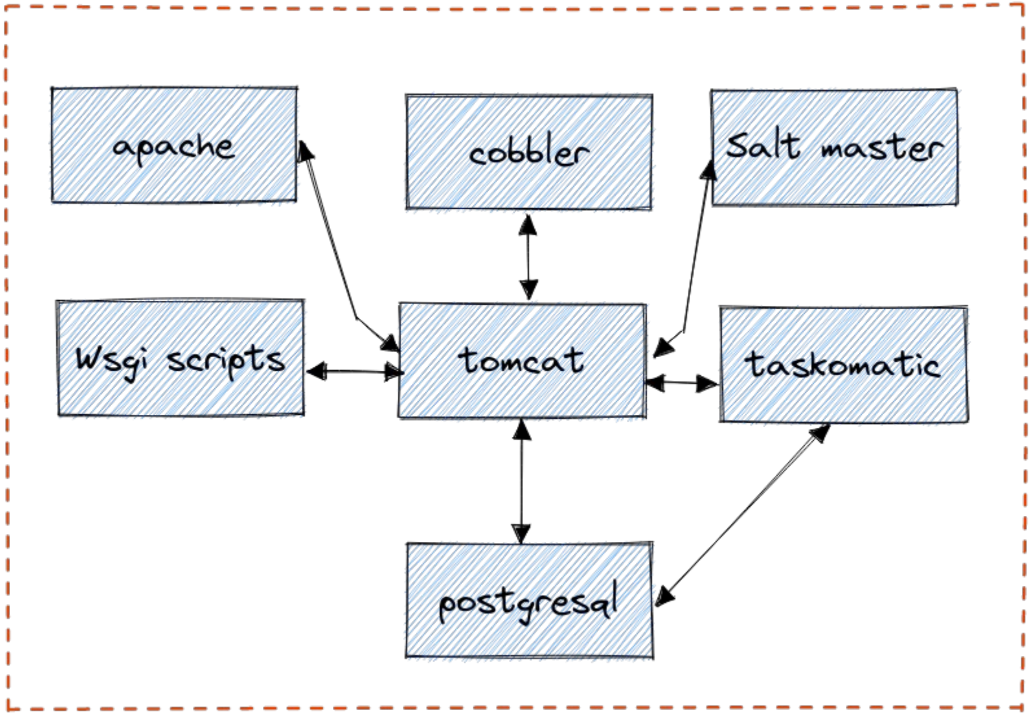
Gitter: mbussolotto
E-mail: mbussolotto@suse.com

Architecture



UYUNI

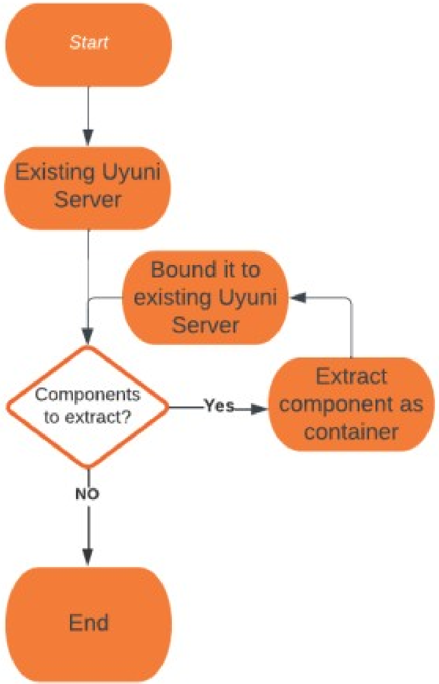
10000 ft view



Possible approach for containerize a Server Application

How containerized Uyuni Server

- Option 1



Start

Existing Uyuni Server

Bound it to existing Uyuni Server

Components to extract?

Yes

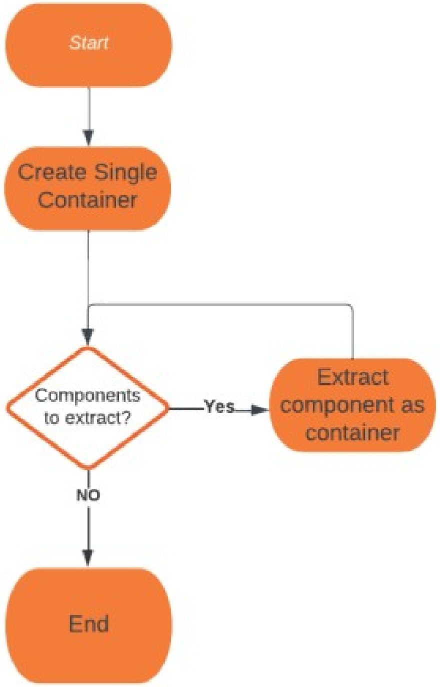
Extract component as container

NO

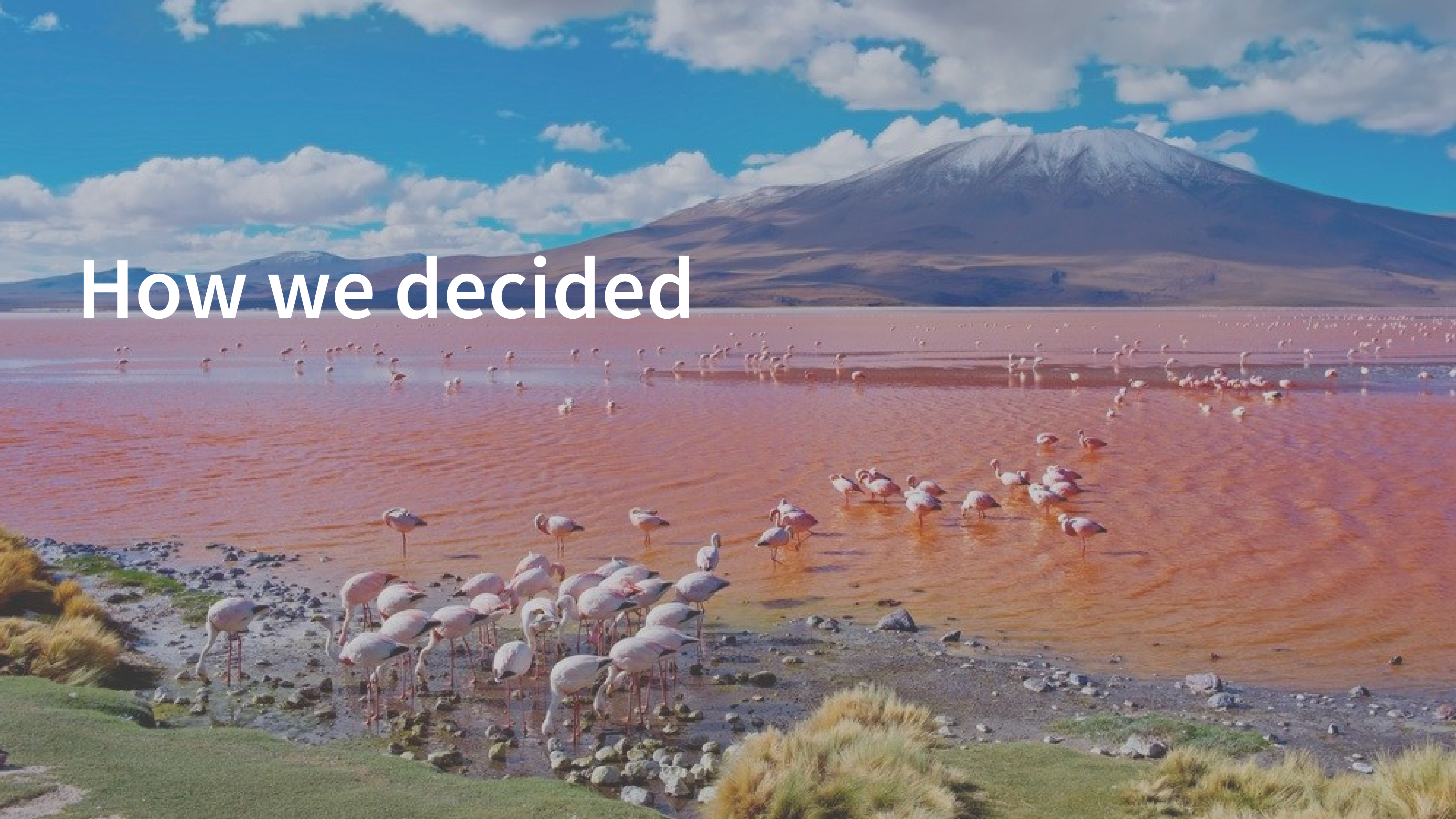
End

How containerized Uyuni Server

- Option 2



How we decided



Which are our goals in containerization?

- Indipendence from the host operating system
- Easier dependency management
- More modular
- Easy for non-container ready users
- Easier to maintain and secure
- Faster innovation
- Modernize the offering
- Align with Devops Strategies

What are our weaknesses?

- Container unfriendly code
- Calls to external tools
- Hostname / FQDN Dependency
- Timezone

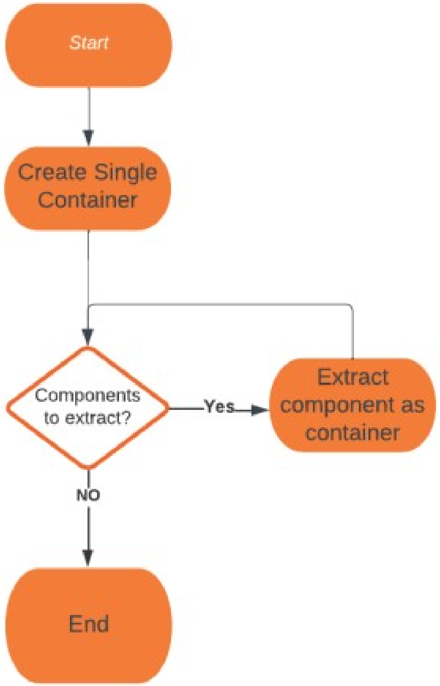
Tools like Windup https://windup.github.io helped for this analysys

Cutting Piecing

- Is it easy?
  - Container already ready?
  - Only configuration?
  - Requires code changes? How much?
- Does it bring value?
  - Can I scale it?
  - Better modularization?
  - Would it be easier to maintain?

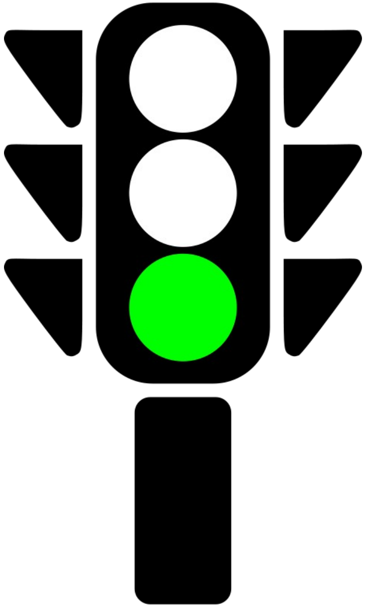
And the winner is

Option 2


Start
Create Single Container
Components to extract?
Yes
Extract component as container
NO
End

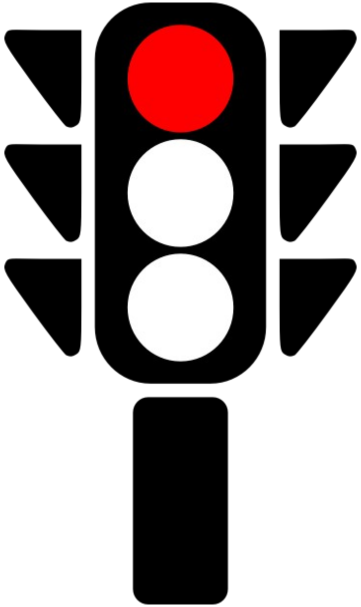
Which goals can be achieved with a single container?

- Indipendence from the host operating system
- Easier dependency management



Which goals can’t be achieved with a single container?

- More modular
  - Not without refactoring
  - To be started with selected containers
- Scalability (but this wasn’t one of our goal)



How we are proceeding

Single Container

- Prerequisite: TEST
- Create a Podman container with systemd
  - Start from an opeSUSE Leap Version (bci-init)
  - Install Uyuni as in a normal system
  - Run Unit Test, Acceptance Test, Manual tests, Reboot and test
- Perform the same tests in K8s
  - RKE2
  - K3s

In this way, we would avoid from the beginning choice that make hard to port Uyuni in K8s.

Challenge: Volumes

- Configuration files should be store in volumes
  - In this way they are persisted, otherwise they would be lost during upgrade
- Files owned by the image should not be store in volumes
  - Otherwise they would not be upgraded

Challenge: FQDN

- In K8s, FQDN cannot be defined
  - Configuration file
  - Tools (e.g. Postfix)

Challenge: Timezone

- Which timezone should be used in a container?
  - Provides methods for settings timezone

Challenge: SSL

- SSL
  - Certificates generated by cert-manager
  - Terminated at Ingress
- Ingress routes
- No fixed hostname

What about “Easy for non-container ready users” ?

How to install a Uyuni Server Container

Uyuniadm is a tool for helping user administer Uyuni servers on K8s and Podman. It provides command for:

- Install a new server from scratch
- Uninstall a server
- Migrate a remote server to container

Uyuniadm is available for

- AlmaLinux 8
- Debian 12
- EL 9
- Fedora 38
- SLE 12 / SLE 15
- Ubuntu 20.04 / Ubuntu 22.04
- openSUSE Leap 15.5
- openSUSE Micro 5.5
- openSUSE Tumbleweed

- https://github.com/uyuni-project/uyuni-tools
- https://build.opensuse.org/package/show/systemsmanagement:Uyuni:Master:ContainerUtils/uyuni-tools

Uyuniadm install

```
Install a new server from scratch

Usage:
  uyuniadm install [command]

Available Commands:
  kubernetes  install a new server on a kubernetes cluster from scratch
  podman      install a new server on podman from scratch
```

Example

```
uyuniadm install podman [fqdn] [flags]
uyuniadm install kubernetes [fqdn] [flags]
```

Uyuniadm migrate podman

This migration command assumes a few things:

- the SSH configuration for the source server is complete, including user andall needed options to connect to the machine,
- an SSH agent is started and the key to use to connect to the server is added to it,
- podman is installed locally

NOTE: for now installing on a remote podman is not supported yet!

```
uyuniadm migrate podman [source server FQDN] [flags]
```

Uyuniadm migrate kubernetes

This migration command assumes a few things:

- the SSH configuration for the source server is complete, including user and all needed options to connect to the machine,
- an SSH agent is started and the key to use to connect to the server is added to it,
- kubectl is installed locally
- A working kubeconfig should be set to connect to the cluster to deploy to

When migrating a server with a automatically generate SSL Root CA certificate, the private key password will be required to convert it to RSA in order to be converted into a kubernetes secret. This is not needed if the source server does not have a generated SSL CA certificate.

```
uyuniadm migrate kubernetes [source server FQDN] [flags]
```

Uyuniadm uninstall

Uninstall a server and optionally the corresponding volumes.

Note that removing the volumes could also be handled automatically depending on the StorageClass used when installed on a kubernetes cluster.

For instance on a default K3s install, the local-path-provider storage volumes will be automatically removed when deleting the deployment even if --purge-volumes argument is not used.

```
Usage:
  uyuniadm uninstall [flags]

Flags:
  -n, --dry-run         Only show what would be done
  -h, --help            help for uninstall
      --purge-volumes   Also remove the volume
```

Using Uyuni in containers: uyunictl

- Commands e.g.:

  ```
  uyunictl exec tail /var/log/rhn/rhn_web_ui.log
  ```

- shell in the pod run :

  ```
  uyunictl exec -ti bash
  ```

- To copy files to the server:

  ```
  uyunictl cp <local_path> server:<remote_path>
  ```

- to copy files from the server :

  ```
  uyunictl cp server:<remote_path> <local_path>
  ```

Uyunictl is available for

- AlmaLinux 8
- Debian 12
- EL 9
- Fedora 38
- SLE 12 / SLE 15
- Ubuntu 20.04 / Ubuntu 22.04
- openSUSE Leap 15.5
- openSUSE Micro 5.5
- openSUSE Tumbleweed

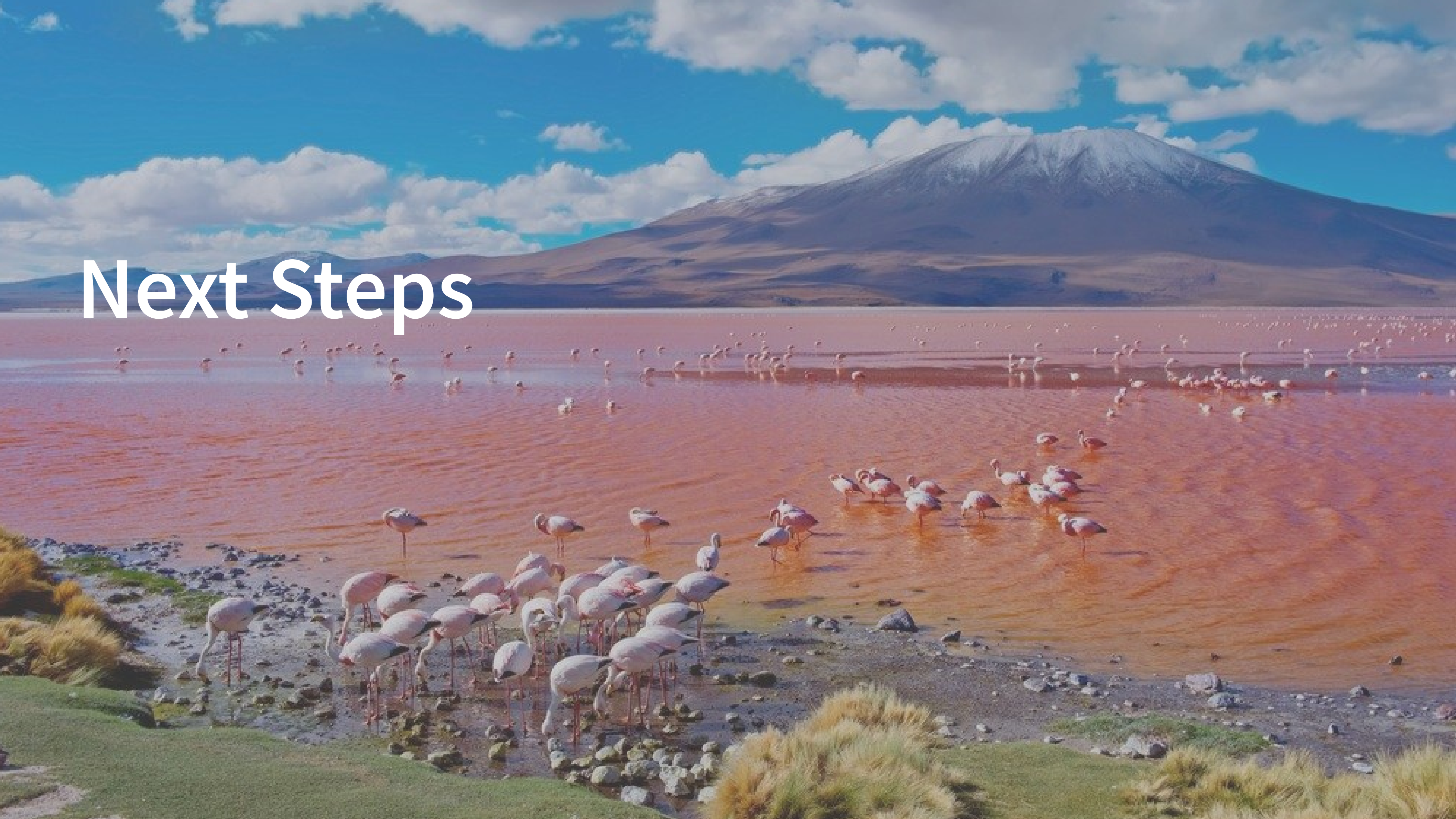
- https://github.com/uyuni-project/uyuni-tools
- https://build.opensuse.org/package/show/systemsmanagement:Uyuni:Master:ContainerUtils/uyuni-tools

Deploying code

To deploy java code on the pod change to the java directory and run:

```
ant -f manager-build.xml refresh-branding-jar deploy-restart-container
```

Next Steps



Next Steps

- Provide tool for migrating AND upgrading a Uyuni Server
- Starting to separate containers

Q&A

/uyuni-project

/uyuni-project

/UyuniProject

Join Us at uyuni-project.org

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/branding-guidelines/