



## **Appliance Management**

StratusLab Tutorial (Orsay, France)

28 November 2012



## What is an appliance?

- Generic machine image with an OS
- Optionally contains pre-installed and pre-configured services

## Why is appliance management a challenge?

- Usually large (1-10 GB) files
- Opaque, difficult to “see” state of machine in file
- Provenance is important for trusting an image
- Removing private information from images is hard



## Machine image creation is a barrier to cloud adoption

- Creating virtual machine images is time-consuming
- Ensuring that machines are secure and correct is difficult
- Sharing existing machines lowers this barrier

## Marketplace facilitates sharing of images

- Registry of metadata for machine & disk images
- Image contents are kept in cloud, grid, or web storage

## Benefits

- **End-users:** browse and use existing images for their analyses
- **Creators:** publicize their work and attract larger user base
- **Cloud Admins.:** Use metadata to evaluate trustworthiness of images

## REST interface

- Exposes a simple HTTP-based REST interface
- Easy to program against in all languages

## Web interface

- REST interface also allows browsing via a web browser
- Signed entries can also be uploaded via the browser

## Global Service

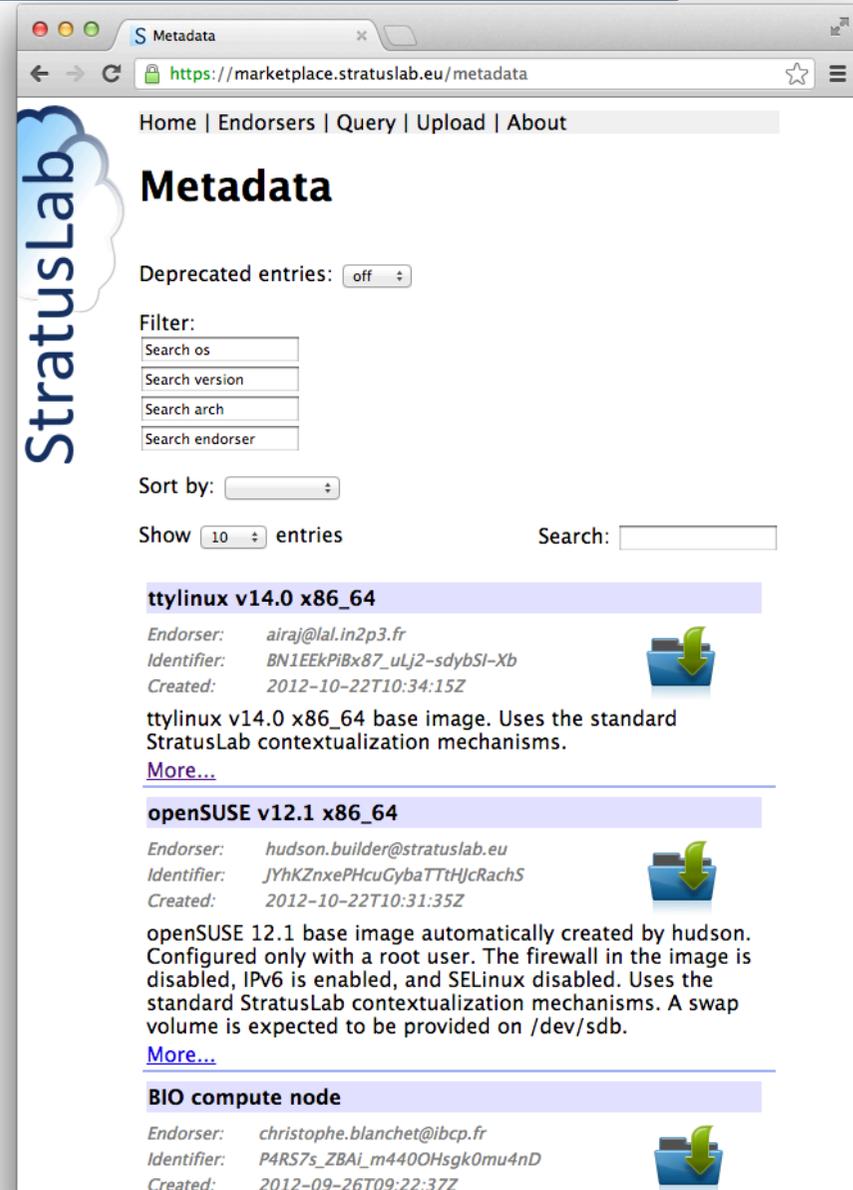
- Global URL: <https://marketplace.stratuslab.eu/>
- Private or local Marketplace instances are possible

# Appliance Summary

*Identifier is the fingerprint of the image. It is SHA-1 checksum in base64 encoding.*

Search Criteria

Appliance Summary



The screenshot shows the StratusLab Metadata page. The page title is "Metadata" and it includes navigation links for Home, Endorsers, Query, Upload, and About. The page features a search filter section with input fields for "Search os", "Search version", "Search arch", and "Search endorser". Below the filter is a "Sort by" dropdown and a "Show 10 entries" indicator. The main content area displays a list of appliances:

- ttylinux v14.0 x86\_64**  
Endorser: airaj@lal.in2p3.fr  
Identifier: BN1EEkPI8x87\_uLj2-sdybSI-Xb  
Created: 2012-10-22T10:34:15Z  
Description: ttylinux v14.0 x86\_64 base image. Uses the standard StratusLab contextualization mechanisms.  
[More...](#)
- openSUSE v12.1 x86\_64**  
Endorser: hudson.builder@stratuslab.eu  
Identifier: JYhKZnxPHcuGybaTTtHjcRachS  
Created: 2012-10-22T10:31:35Z  
Description: openSUSE 12.1 base image automatically created by hudson. Configured only with a root user. The firewall in the image is disabled, IPV6 is enabled, and SELinux disabled. Uses the standard StratusLab contextualization mechanisms. A swap volume is expected to be provided on /dev/sdb.  
[More...](#)
- BIO compute node**  
Endorser: christophe.blanchet@ibcp.fr  
Identifier: P4RS7s\_ZBAI\_m440OHsgk0mu4nD  
Created: 2012-09-26T09:22:37Z

# Appliance Details

Identifier

Description

Detailed Info.

Location(s)

Other formats

Home | Endorsers | Query | Upload | About

## Metadata

### BN1EEkPiBx87\_uLj2-sdybSI-Xb

ttylinux v14.0 x86\_64 base image. Uses the standard StratusLab contextualization mechanisms.

**type:** base  
**kind:** machine  
**format:** raw  
**endorser:** airaj@lal.in2p3.fr  
**os:** ttylinux v14.0 x86\_64  
**version:** 1.0  
**endorsed:** 2012-10-22T10:34:15Z  
**created:** 2012-10-22T10:32:51Z  
**valid:** 2013-04-18T10:32:51Z  
**hypervisor:** kvm  
**publisher:** StratusLab  
**bytes:** 102400000

<b>MD5</b>	12cdc99c87300d1b86b34656ab60c079
<b>SHA-1</b>	137510490f881c7ceffb8b8f6fac7726d223e5db
<b>SHA-256</b>	31ac9167278234e90cb928f966b365035f7f2fe8fc9b73475d4f3591c15912d1
<b>SHA-512</b>	cf824d3e5232b636915985c1e498dd4b7312bc132b0c4422f4cdebad899dae87cbe4eae1622559459685de2e3b9a7d679b3181f6e97ba3c73917947255ac3598

**checksum:**

**location:** [http://appliances.stratuslab.eu/images/base/ttylinux-14.0-x86\\_64-base/1.0/ttylinux-14.0-x86\\_64-base-1.0.img.gz](http://appliances.stratuslab.eu/images/base/ttylinux-14.0-x86_64-base/1.0/ttylinux-14.0-x86_64-base-1.0.img.gz)



## Image metadata

- Must conform to a defined schema
- Uses the RDF-XML format
- Must be cryptographically signed with a (grid) certificate
- Must contain image ID and checksums to make connection to image
- May contain location elements with image content URL(s)

## Image Content

- Separated from metadata
- Can be stored in web, cloud, or other storage
- Multiple locations of the image can be provided
- Cached by the cloud to provide low-latency starts



## Typical Marketplace workflow:

- Create image from scratch or based on existing image
- Upload the image to cloud, grid, or web storage area
- Create the metadata for the image
- Sign the metadata with your (grid) certificate
- Upload the signed metadata to the Marketplace

# Creating an Image



## Creating an image is difficult, long, tedious...

- Don't do it; reuse an existing image instead!
- Images for popular operating systems are provided by StratusLab

## Adapt an existing image:

- StratusLab provides tools to create new images from existing ones
- Provides standard contextualization and good security practices

## Create an image from scratch:

- Only do this if you really must and contact us for help!
- Must provide contextualization for image to work on cloud
- Ensure no private information is embedded in image
- Lock down services to avoid security holes

# Creating and Uploading Image



## Cheat (!)

- Normally, use one of the previous methods to create image
- *Copy the ttylinux image locally to try out metadata utilities*

## Uploading of image

- Normally, image would be transferred to cloud, grid, or web storage
- Public images must be accessible via http(s) at the moment
- Private images can be accessible via a pdisk URL
- Location URL(s) must be part of metadata for image to be used
- *Just note the URL of the image that you have downloaded*

# Create Metadata Description



## Use stratus-build-metadata for creating metadata:

```
$ stratus-build-metadata \  
  --author='your name' \  
  --os=ttylinux \  
  --os-version=9.7 \  
  --os-arch=i486 \  
  --image-version=1.3 \  
  --location=http://example.org/ttylinux.gz \  
  --compression=gz \  
  ttylinux-9.7-i486-base-1.3.img  
$
```

- Wait for the unknown state, then kill (remove) the instance:

## Look at the contents of the file:

- Identifier is a base64 encoded SHA-1 checksum
- Checksums ensure that downloaded images match the metadata;  
*these must be the checksums of the uncompressed image!*
- Empty endorser element and no signature element

# Create Metadata Description



## Try to validate the unsigned metadata file:

- There is no signature so the file should not be valid

```
$ stratus-validate-metadata ttylinux-9.7-i486-base-1.3.xml
Invalid: ttylinux-9.7-i486-base-1.3.xml
no signature
```

## Sign the contents of the file with a grid certificate:

- Signed file contains endorser and signature elements
- Certificate location and password can be specified in config. file
- A signed file can be re-signed by same or different person

```
$ stratus-sign-metadata \
  --p12-cert grid.p12 \
  --p12-password xxxxxx \
  ttylinux-9.7-i486-base-1.3.xml
  Manifest file successfully signed: ttylinux-9.7-i486-base-1.3.xml

$ stratus-validate-metadata ttylinux-9.7-i486-base-1.3.xml
Valid: ttylinux-9.7-i486-base-1.3.xml
```

# Upload Metadata Description

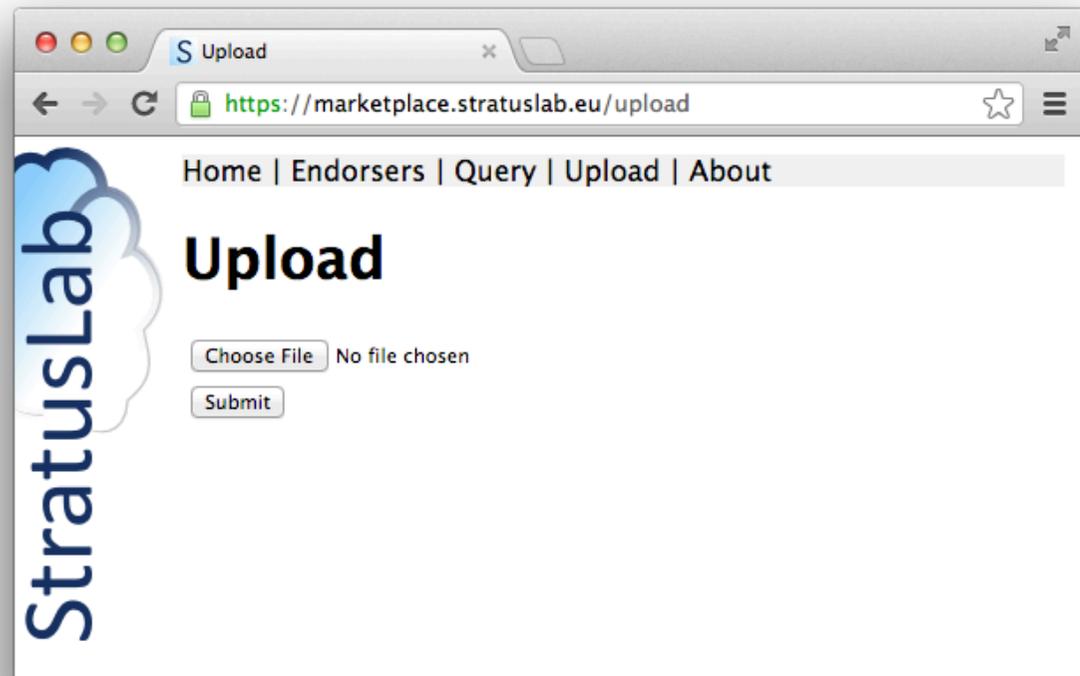
File can be uploaded via the command line:

- Use: `stratus-upload-metadata`

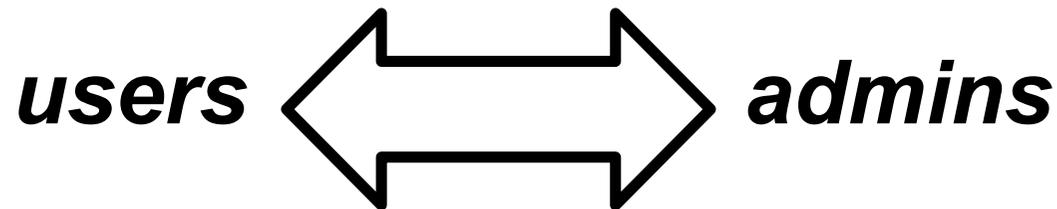
```
$ stratus-upload-metadata ttylinux-9.7-i486-base-1.3.xml  
http://cloud.lal.stratuslab.eu:8081/metadata/LwCRbwCalYSysY1wftQdAj6Bwoi/  
loomis@lal.in2p3.fr/2011-09-13T09:58:54Z
```

Or via a browser →

Note: The server may validate the email address in the metadata.



# Using a Marketplace Image



## Pass the identifier for metadata entry to start instance

- `stratus-run-instance`  
`LwcRbwCa1YSysY1wftQdAj6`  
`Bwoi`
- Use normal machine lifecycle to control machine

## StratusLab will validate image before running it:

- `stratus-policy-image` enforces policy defined by administrator
- Policies can include endorser white lists, checksum black lists, etc.

# Image Deprecation



## Invalidating (remove endorsement of) an image:

- Use: `stratus-deprecate-metadata`
- The command deprecates an image and gives a reason
- If there are no other endorsers, the image won't be run

```
$ stratus-deprecate-metadata \  
  --reason="JUST FOR FUN" \  
  --p12-cert=/Users/loomis/.globus/cert.p12 \  
  --p12-password=XXXXXX \  
  $TTYLINUX_ID  
http://cloud.lal.stratuslab.eu:8081/metadata/LwcRbwCalYSysY1wftQdAj6Bwoi/  
loomis@lal.in2p3.fr/2011-09-21T14:52:43Z
```

---

# Questions and Discussion

# Exercises: Marketplace Metadata



## Marketplace

- Search Marketplace to see what types of machines are available
- What metadata is available for existing virtual machines?
- What metadata would you like to have?

## Image Metadata Lifecycle

- ***For the exercises do not actually upload the image metadata.***
- Run through entire lifecycle (except uploading) for image metadata
- What information is required in the metadata?
- What additional information can be provided?
- Can there be multiple metadata entries for an image?
- How would you use the Marketplace as end-user, administrator, developer?



<http://www.stratuslab.eu>

Copyright © 2012, Members of the StratusLab collaboration.

This work is licensed under the Creative Commons Attribution 3.0 Unported License (<http://creativecommons.org/licenses/by/3.0/>).

