Getting to Know AutoYaST

An introduction to automated installation with AutoYaST & Friends

Imobach González Sosa
igonzalezsosa@suse.com
Knut Anderssen González
kanderssen@suse.com

What is AutoYaST?

Unattended Installation/Upgrade

- Tool to perform unattended installation/upgrade of openSUSE/SUSE systems
- Allow configuration of already installed systems
- It takes a description (known as a *profile*)...
- … and it *drives* YaST to setup the system

AutoYaST Profiles

```
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://www.suse.com/1.0/configns">
  <partitioning config:type="list">
    <!-- Partitioning schema -->
  </partitioning>
  <software>
    <!-- Software selection -->
  </software>
  <networking>
    <!-- Network configuration -->
  </networking>
  <scripts>
    <!-- Scripts to be executed pre/during/after installation-->
  </scripts>
</profile>
```

A minimal profile

```
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://www.suse.com/1.0/configns">
  <users config:type="list">
    <user>
      <encrypted config:type="boolean">false</encrypted>
      <user_password>nots3cr3t</user_password>
      <username>root</username>
    </user>
  </users>
</profile>
```

AutoYaST XML: Simple Values

```
<element config:type="TYPE">VALUE</element>
```

- XML using an special attribute to specify the type
- Supported types are *integer*, *boolean*, *symbol* and *list*
- Check the documentation to see which type applies

AutoYaST XML: Lists

```
<packages config:type="list">
  <listentry>ruby2.5</listentry>
  <listentry>git</listentry>
</packages>
```

- In some cases, *<listentry>* can be replaced with some meaningful name (like *<user>*)

Writing a profile

- From scratch
- Clone an installed system (and tune it if needed)
  - AutoYaST UI
  - Cloning tool

Playing with the AutoYaST UI

- Install the *autoyast2* package
- Start YaST and open the *Autoinstallation Configuration* module
- Just play around

How AutoYaST Works?

Overview

1) Boot the installer with the option autoyast=<URL>

2) AutoYaST imports the profile and installs the system (1st stage)

3) Reboot

4) Additional configuration (2nd Stage)

AutoYaST URL

- *autoyast* is used to specify the profile URL
- Several URL schemas are supported (file, device, http(s), ftp, nfs, usb, label, etc.)
- When network is required, do not forget to add the *ifcfg* option

```
autoyast=http://192.168.122.1/autoinst.xml ifcfg=eth0=dhcp
```

1st Stage: System Installation

- Usually, AutoYaST retrieves the profile
- AutoYaST reads settings from the profile...
- … and proposes default values for all missing settings
- Configure several basic settings: language, bootloader, partitioning, etc.
- Software installation

2nd Stage: Additional Configuration

- It happens after the reboot
- Additional services configuration
- Optional
- Slowly moving stuff to 1st stage

Building Our Own Profile

Let’s Create Our Own Profile

- Setting Country Configuration
- Adding some software
- Setting up our user account
- Adjusting the partitions layout
- Configuring the network
- Opening the SSH service
- Locking out the bad guys

Before we start

- Check out the AutoYaST draft documentation
- Get this slides and the base profile from https://bit.ly/2LsuMyt

Setting Country Configuration

- The <*language*>, <*keyboard*> and <*timezone*> sections allow to set country related configuration
- Language uses the ISO code (*en_US*, *cs_CZ*, etc.)
- Keyboard uses values like *english*, *german*, etc.
- Time zone uses the typical name (*Atlantic/Canary*)
- See Country Settings documentation

Exercise 1: Setting Country Configuration

- Add the country configuration which fits you

Adding Some Software

- <*software*> defines which software should be installed
- The user can specify *patterns* and *packages*
- When a package is included in a pattern, it is not listed
- See Software documentation

Exercise 2: Adding Some Software

- Check your base system patterns
- Select your preferred shell and desktop environment
- Hint: *zypper se -t pattern*

Setting Up Our User Account

- <*users*> and <*groups*> allow to define users and groups
- Only username and password are mandatory
- Watch out for duplicated IDs
- See Users and Groups

Exercise 3: Setting Up Our User Account

- Add your own user account
- Do not forget to adjust the shell to the one you installed

Adjusting the Partitions Layout

- The partitioning support has been reimplemented for openSUSE Leap 15.0 (and SUSE Linux Enterprise 15)
- It (re)uses the same approach than the new storage layer
- When it is not defined, it uses the guided proposal

Adjusting the Partitions Layout

- <*partitioning*> defines a <*drive*> section for each device
- Each <*drive*> contains a set of partitions
- A <*drive*> can be a physical drive or a logical one (like an LVM volume group)
- See Automated Partitioning

Exercise 4: Adjusting the Partitions Layout

- Let’s try to define the following partitioning layout
  - 10GiB *Btrfs* filesystem partition for root (/)
  - 512MiB for swap
  - The remaining space should be assigned to an *ext4* filesystem to be used as */home*

Network Configuration

- For fetching remote profiles, network configuration is needed

```
autoyast=http://192.168.122.1/autoinst.xml ifcfg=eth0=dhcp
```

- By default, linuxrc network configuration is merged or copied at the end of the 1st stage (since Leap 42.3)
- The <*networking*> resource is used to store the whole network configuration
- See Network Configuration

Exercise 5: Configuring the Network

- Set your hostname
- Set your nameservers as ‘8.8.8.8’ and ‘8.8.4.4’

Managing Services

- *<services-manager>* allows to enable/disable services
- The default target can be specified too
- No services will be started during the 1st stage
- See Services and Targets

Exercise 6: Opening the SSH service

- Open the SSH service
  - Install the required package
  - Enable the service

Locking out the bud guys

- SuSEFirewall2 has been replaced with *firewalld*
- Includes a predefined set of zones and services
- See Firewall Configuration

Exercise 7: Locking out the bad guys

- Configuring the firewall
  - Set your default zone as ‘block’
  - Assign your interface card to the ‘public’ zone
  - Block all services except ‘ssh’

Let’s do it interactive

Asking Questions

- Mechanism to gather information from users at runtime
- Offers basic widgets and simple workflow control
- It can use the answers to:
  - Run scripts
  - Modify the profile
  - Store values in some file

Exercise 7: Ask Some Questions

- Ask for this information and update the profile accordingly:
  - Username and password
  - Preferred desktop environment

What else?

Running Scripts

- Able to run user scripts at different points of the installation
- They offer a way to extend AutoYaST
- Scripts can be defined inline or downloaded
- See Custom User Scripts

Deploying Files

- Full configuration files can be written by AutoYaST
- The content can be downloaded or embedded in the profile
- It happens during the 2nd stage
- See Adding Complete Configurations

Error Reporting

- AutoYaST is meant to be unattended
- But some problem may happen
- Reporting level can be controled
- See Reporting

Configuring an Installed System

- AutoYaST is able to configure an already installed system
- Not all sections are applied
- See Running AutoYaST in an Installed System

```
yast ayast_setup setup filename=/path/to/autoinst.xml
```

Rules and Classes

- A *class* can be used to define the common parts of different profiles
- A *rule* allows to select a given profile depending on systems properties
- See Rules and Classes

Salt/Puppet Integration

- Part of the work can be delegated to a *Configuration Management System*
- AutoYaST does the initial installation: partitioning, network configuration, software installation, etc.
- Salt/Puppet performs additional configuration: more software installation, services configuration, etc.
- See Configuration Management

Troubleshooting

Validate Your Profile

- Install the *yast2-schema* and *jing* packages
- Run *jing* against your profile

```
jing \
/usr/share/YaST2/schema/autoyast/rng/profile.rng \
/path/to/your/profile.xml
```

Check Logs (optional)

- Check YaST logs (/var/log/YaST2)
- Have a look at */var/adm/autoinstall/*
  - /var/adm/autoinstall/cache/installedSystem.xml
  - Scripts logs are located there too

Ask for Help

- Ask for help on the mailing lists, or the forums or IRC
- Open a bug report if you think it is an AutoYaST problem
- Please, attach logs (using *save_y2logs*)

Thanks!

Thanks!

- My beloved YaST team for developing and maintaining AutoYaST…
- … especially to Stefan Schubert
- The openSUSE Project!

References

- YaST Homepage

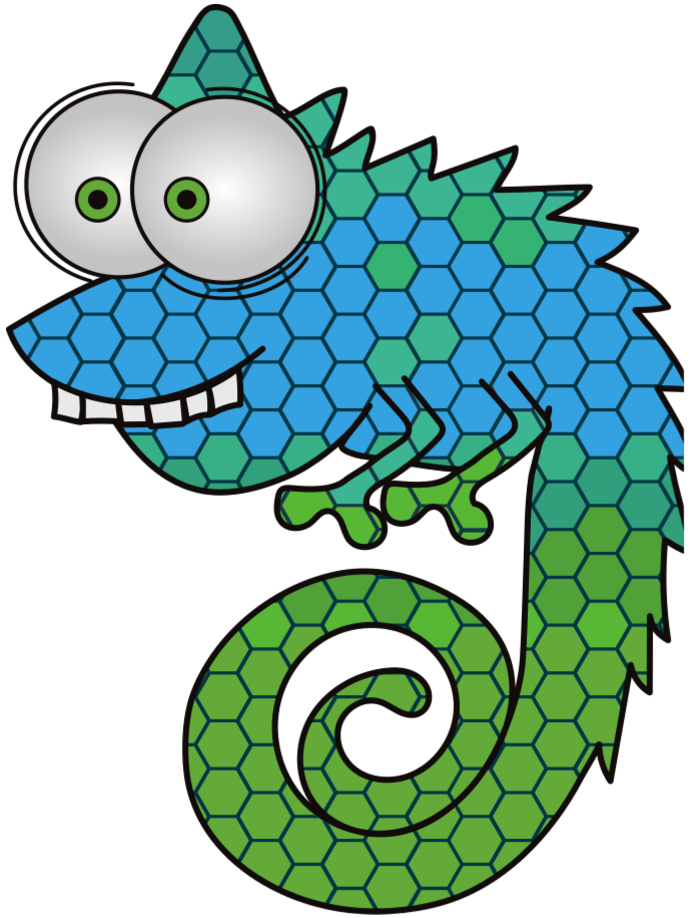
  http://yast.opensuse.org
- AutoYaST Documentation for openSUSE

  http://doc.opensuse.org/projects/autoyast/
- Documentation Drafts

  http://susedoc.github.io
- Linuxrc Reference

  https://en.opensuse.org/SDB:Linuxrc

Questions?



Join Us at www.opensuse.org

rbrown@opensuse.org

http://opensuse.github.io/branding-guidelines/