ypid’s Ansible roles Documentation

Release master

Robin ‘ypid‘ Schneider

May 23, 2017

Contents

Welcome to the documentation of ypid's Ansible roles based on the structure of debops/docs.

CHAPTER 1

Ansible role: ypid.dhclient

Introduction

Configure the hostname to be send by **dhclient** in case it should differ from the system hostname.

Installation

This role requires at least Ansible `v1.9.0`. To install it, run:

```
ansible-galaxy install ypid.dhclient
```

Getting started

Example inventory

To configure dhclient on a given host it should be included in the `ypid_service_dhclient` Ansible inventory group:

```
[ypid_service_dhclient]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.dhclient` role:

```
---

- name: Configure dhclient
  hosts: [ 'ypid_service_dhclient' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.dhclient
      tags: [ 'role::dhclient' ]
```

This playbook is shipped with this role under `docs/playbooks/opsi.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after host is first configured to speed up playbook execution, when you are sure that most of the configuration has not been changed.

Available role tags:

`role::dhclient` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::dhclient:pkgs` Tasks related to system package management like installing or removing packages.

ypid.dhclient default variables

Sections

Required packages

`dhclient__base_packages`

List of base packages to install.

```
dhclient__base_packages:
  - 'isc-dhcp-client'
```

dhclient configuration

dhclient__send_hostname

Hostname to send by **dhclient**. When an `gethostname()` is defined, the hostname will not be explicitly set in `/etc/dhcp/dhclient.conf` which should result in the system hostname being used.

```
dhclient__send_hostname: 'gethostname()'
```

Copyright

```
ypid.dhclient - Configure dhclient

Copyright (C) 2016 Robin Schneider <ypid@riseup.net>

DebOps is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License version 3, as
published by the Free Software Foundation.

DebOps is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with DebOps. If not, see https://www.gnu.org/licenses/.
```

Changelog

ypid.dhclient

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.dhclient v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

CHAPTER 2

Ansible role: ypid.dm

Introduction

Configure a display manage. Supported options are:

- GNOME Display Manager.
- LightDM.

Installation

This role requires at least Ansible `v2.1.4`. To install it, run:

```
ansible-galaxy install ypid.dm
```

This role uses features recently introduced in Jinja2, namely the `equalto` filter which was released with Jinja 2.8. Jinja 2.8 is available in Debian Jessie Backports.

Getting started

Example inventory

To manage the display manager on a given host it should be included in the `debops_service_dm` Ansible inventory group:

```
[debops_service_dm]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.dm` role:

```
---

- name: Configure a display manager
  hosts: [ 'ypid_service_dm' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.dm
      tags: [ 'role::dm' ]
```

This playbooks is shipped with this role under `./docs/playbooks/dm.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::dm` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::dm:pkg` Tasks related to system package management like installing or removing packages.

ypid.dm default variables

Sections

Selected display manager

dm__name

Which display manager should be used?

Supported display managers:

- `gdm`
- `lightdm`

Note that the different display managers support a different set of features so not all configuration options that this role provides will have an effect depending on this choice.

```
dm__name: 'lightdm'
```

Required packages

dm__package_map

dm__name to base core package name mapping.

```
dm__package_map:
  lightdm: 'lightdm'
  gdm: 'gdm3'
```

dm__packages

List of packages to install regardless of the selected display manager.

```
dm__packages: []
```

dm__lightdm_packages

Additional list of packages to install when LightDM is selected.

```
dm__lightdm_packages: []
```

dm__gdm_packages

Additional list of packages to install when GDM is selected.

```
dm__gdm_packages:
  - 'libpam-gnome-keyring'
```

User automatic login

dm__automatic_login_enabled

Should the user configured by *dm__automatic_login_user_name* be automatically logged in?

```
dm__automatic_login_enabled: False
```

dm__automatic_login_user_name

Name of the user which should be automatically logged in.

```
dm__automatic_login_user_name: 'user'
```

dm__automatic_login_time_delay

How long should what before *dm__automatic_login_user_name* will be logged in to give other users the chance to login as their user?

```
dm__automatic_login_time_delay: 0
```

Greeter

> **Warning:** Can not be configured for GDM in Debian jessie.

dm__greeter_include_all_users

Only include selected logins in the greeter.

```
dm__greeter_include_all_users: False
```

dm__greeter_include_users

Only include selected logins in the greeter.

```
dm__greeter_include_users: []
```

dm__greeter_time_show

Should the greeter show the time?

```
dm__greeter_time_show: True
```

dm__greeter_time_format

strftime(3) format string.

```
dm__greeter_time_format: '%F %T %a'
```

dm__greeter_lightdm_indicators

List of indicators to show on the panel.

```
dm__greeter_lightdm_indicators: '~host;~spacer;~clock;~spacer;~language;~session;~
↪a11y;~power'
```

dm__greeter_show_user_list

Should a list of users be shown?

```
dm__greeter_show_user_list: False
```

Debugging

`dm__debugging_enabled`

More verbose logs. Additionally lets the X server dump core if it crashes.

```
dm__debugging_enabled: False
```

Display manager restart

`dm__restart_on_configuration_change`

Should the display manager be restarted when the configuration is changed? Useful for testing.

> **Warning:** This will kill active user sessions!

```
dm__restart_on_configuration_change: False
```

Session manager

`dm__session_manager`

File to the executable session manager file. If given, it will be configured as the default session manager.

You can check the available session managers using **`update-alternatives --list x-session-manager`**.

If this variable is set to the special value `{{ omit }}`, the configuration is not altered.

```
dm__session_manager: '{{ omit }}'
```

Paths

`dm__binary_path`

Path to the executable display manager file. This is used to set the default display manager for the X server.

```
dm__binary_path: '/usr/sbin/{{ dm__package_map[dm__name] }}'
```

Copyright

Changelog

ypid.dm

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.dm v0.1.0 - unreleased

Added

- Initial coding and design with GDM. [ypid]
- Support LightDM. [ypid]
- Support to switch between different display managers. [ypid]
- Support to set the default session manager using *dm__session_manager*. [ypid]

Chaneged

- Rename role from `ypid.gdm` to `ypid.dm` after adding support for LightDM. [ypid]

CHAPTER 3

Ansible role: ypid.epoptes

Introduction

The `ypid.epoptes` role allows you to manage and configure Epoptes. Epoptes is a Free and open-source software computer lab management and monitoring tool.

The role has originally been written to deploy Epoptes in a linuxmuster.net environment using the Postsync feature of LINBO.

In the process of writing and testing this role, direct deployment against clients has also being implemented.

Installation

This role requires at least Ansible `v2.1.5`. To install it, run:

```
ansible-galaxy install ypid.epoptes
```

Getting started

Preparing clients

Epoptes is packaged in Debian which is used for installation. Some of the Epoptes configuration needs to be changed/managed by this role. In order for dpkg to not ask about changed files managed by this role during package updates, the role diverts those files by default.

This step is done when `prepare_client` is present in the *epoptes__deploy_modes* list. Preparation is the first stage of the role and followed by the configuration stage. Prepared clients are the foundation for both supported configuration modes described in the following sections.

Direct configuration modes

The role can be run directly against all teacher and student computers and setup Epoptes in classrooms this way. This mode does not require a server other than the Ansible controller for client configuration.

This mode is active when one of `teacher` or `student` is present in the *epoptes__deploy_modes* list.

linuxmuster.net postsync configuration modes

The postsync mode was based on the Epoptes integration in linuxmuster.net but handles the access/distribution of the private key of Epoptes differently to mitigate exposure of this key via unauthenticated rsync access which is possible when following the Epoptes integration in linuxmuster.net documentation. A skilled adversary (never underestimate students ;-) could use this key to setup their own Epoptes server after also spoofing the IP address on which clients try to reach the server. This attack has been mitigated by only copying the private key to the teacher computer when a teacher has logged in.

This mode is active when `postsync` is present in the *epoptes__deploy_modes* list.

The directory structure on the server will look similar to this:

```
/var/linbo/linuxmuster-client/image_name
- r23_student
|   - etc
|   |   - default
|   |   |   - epoptes-client
|   |   - epoptes
|   |   |   - server.crt
|   |   - init.d
|   |   |   - epoptes-client
|   |   - xdg
|   |       - autostart
|   |           - epoptes-client.desktop
|   - usr
|       - local
|           - bin
|               - epoptes-client-loop.sh
- r23_teacher
|   - etc
|   |   - default
|   |   |   - epoptes
|   |   - epoptes
|   |   - init.d
|   |   |   - epoptes
|   |   - sudoers.d
|   |   |   - ansible-teacher-epoptes-restart
|   |   - xdg
|   |       - autostart
|   |           - epoptes-copy-key.desktop
|   - usr
|       - local
|           - bin
|           |   - epoptes-copy-key.sh
|           - share
|               - applications
|                   - epoptes.desktop
```

Example host inventory

The inventory configuration will be different depending on which mode of operation you choose for your environment.

To manage Epoptes on a given host or set of hosts, they need to be added to the [ypid_service_epoptes] Ansible group in the inventory:

```
[ypid_service_epoptes]
hostname
```

Common inventory for both modes:

```
[sint.example.org_clients_r23_teachers]
r23-pc01.sint.example.org

[teachers:children]
sint.example.org_clients_r23_teachers

[sint.example.org_clients_r23_students]
r23-pc02.sint.example.org
r23-pc03.sint.example.org

[students:children]
sint.example.org_clients_r23_students

[sint.example.org_clients_r23:children]
sint.example.org_clients_r23_teachers
sint.example.org_clients_r23_students

[sint.example.org_clients:children]
sint.example.org_clients_r23
```

For direct configuration mode, all client hosts would be member of the [ypid_service_epoptes] Ansible group so that the role is run against all of them directly. This could look as follows, where [sint.example.org_clients_r23] is an Ansible group itself as defined in the example above:

```
[ypid_service_epoptes:children]
sint.example.org_clients_r23
```

For linuxmuster.net postsync configuration mode, the server needs to be member of the [ypid_service_epoptes] Ansible group. Additionally, all clients should be part of the group as well so that they can be prepared. Note that you would typically only run Ansible against one of those clients, then make an image of this one client and distribute the image to all other clients from which they will sync their root filesystem.

```
[ypid_service_epoptes]
server.sint.example.org

[ypid_service_epoptes:children]
sint.example.org_clients
```

Example Ansible inventory variables

To choose the mode in which the role runs against each remote host, the `epoptes__deploy_modes` variable has to be set for each host explicitly. This should be done again using Ansible groups.

For direct configuration mode:

`ansible/inventory/group_vars/teachers`:

```
epoptes__deploy_modes:
  - 'prepare_client'
  - 'teacher'
```

`ansible/inventory/group_vars/students`:

```
epoptes__deploy_modes:
  - 'prepare_client'
  - 'student'
```

For linuxmuster.net postsync configuration mode:

`ansible/inventory/group_vars/sint.example.org_clients`:

```
epoptes__deploy_modes:
  - 'prepare_client'
```

`ansible/inventory/host_vars/server.sint.example.org`:

```
epoptes__deploy_modes:
  - 'postsync'
```

Example playbook

Here's an example playbook that uses the `ypid.epoptes` role:

```
---

- name: Setup and manage Epoptes
  hosts: [ 'ypid_service_epoptes' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:
    - role: ypid.epoptes
      tags: [ 'role::epoptes' ]
```

This playbooks is shipped with this role under `./docs/playbooks/epoptes.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using ypid-ansible-common where the symlinking is done for you :-)

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::epoptes` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::epoptes:pkgs` Tasks related to system package management like installing or removing packages.

`role::epoptes:prepare_client` Tasks related to client preparation.

`role::epoptes:keys` Tasks related to key management like creating the private key and the X.509 certificate.

ypid.epoptes default variables

Mode of operation

`epoptes__deploy_modes`

In which mode(s) should the role operate when running against a specific host. Possible options:

`prepare_client` Prepare a client computer for later configuration deployment.

`teacher` Directly configure a teacher computer.

`student` Directly configure a student computer.

`postsync` Write the configuration into a directory structure and configure LINBO using it's postsync feature to sync the appropriate configuration to clients. The role is expected to be run against a linuxmuster.net server in this mode.

Please choose one or more modes. Note that the sets `{ "postsync" }` and `{ "teacher", "student" }` can not be used together. The role asserts for this to ensure valid configuration.

```
epoptes__deploy_modes: []
```

epoptes__prepare_mode

How should the role handle client preparation? Possible options:

divert Default. Uses **dpkg-divert** to rename the files. This operation is package management aware and thus preferred.

remove Can be used for debugging when you used *epoptes__deploy_modes* set to `student` or `teacher` and later want a clean client again.

```
epoptes__prepare_mode: 'divert'
```

epoptes__deploy_state

What is the desired state which this role should achieve? Possible options:

present Default. Ensure that Epoptes is installed and configured as requested.

absent Ensure that Epoptes is uninstalled and it's configuration is removed. This mode is not fully tested and might not remove all "traces".

```
epoptes__deploy_state: 'present'
```

System packages

epoptes__student_packages

List of packages to install on student client systems.

```
epoptes__student_packages:
  - 'epoptes-client'
```

epoptes__teacher_packages

List of packages to install on teacher client systems.

```
epoptes__teacher_packages:
  # Master component (server).
  - 'epoptes'
```

epoptes__optional_client_packages

List of optional packages to install on all client systems with Epoptes enabled.

```
epoptes__optional_client_packages: []
```

Epoptes X.509 certificate

Those settings only apply when the private key and X.509 certificate are generated on the initial role run. You can remove those files on the Ansible controller to generate new once. They will be pushed to remote hosts on the next role run. Note that you will need to update/redeploy all clients which use Epoptes for it to work correctly.

epoptes__cert_key_type

Epoptes public key algorithm to use. Refer to *req(1ssl)* for details.

```
epoptes__cert_key_type: 'rsa'
```

epoptes__cert_key_size

Epoptes public key size to use. This depends on the setting of *epoptes__cert_key_type*.

```
epoptes__cert_key_size: 4096
```

epoptes__cert_days_valid

Days until the certificate expires.

```
epoptes__cert_days_valid: '{{ 366 * 10 }}'
```

linuxmuster.net postsync configuration

epoptes__postsync_image_name

Name of the Linbo image for which to generate the postsync script.

```
epoptes__postsync_image_name: 'linux_mint'
```

epoptes__postsync_script

File path of the postsync script.

```
epoptes__postsync_script: '/var/linbo/{{ epoptes__postsync_image_name }}.cloop.
↪postsync'
```

Epoptes configuration

epoptes__port

The port where the server will be listening on and where clients will try to connect to. For security reasons it defaults to a well-known port which requires elevated permissions to listen on.

```
epoptes__port: 789
```

epoptes__wol_enabled

Should teachers be able to start computers using Wake on LAN?

```
epoptes__wol_enabled: True
```

epoptes__wol_mode

Wake On LAN mode.

```
epoptes__wol_mode: 'g'
```

epoptes__socket_group

System group the Epoptes server will use for the communications socket. That means that any user in that group will be able to launch the Epoptes UI and control the clients.

```
epoptes__socket_group: 'teachers'
```

epoptes__secret_path

Path on the Ansible controller where the public-private-keypair is stored.

```
epoptes__secret_path: '{{ secret + "/epoptes/by-group/all" }}'
```

epoptes__client_key_dir

Directory where the private key will be cached in on the client when a teacher logs in.

```
epoptes__client_key_dir: '/dev/shm/.k'
```

epoptes__teacher_ui_name

Name of the Epoptes (G)UI which can be used by teachers to control computers of students.

```
epoptes__teacher_ui_name: 'Epoptes'
```

epoptes__room_prefix

Room prefix string to extract the room from the hostname.

```
epoptes__room_prefix: ''
```

epoptes__room_suffix

Room prefix string to extract the room from the hostname.

```
epoptes__room_suffix: '-*'
```

epoptes__rooms

Epoptes room definitions.

Refer to the *documentation of all options* for more details.

```
epoptes__rooms: {}
```

epoptes__enforce

Enforce mode aka sweet revenge mode :) Restart the display server if a student tries to kill the **epoptes-client**. Work in progress and probably not needed/handled differently in Epoptes.

```
epoptes__enforce: False
```

epoptes__enforce_dm

The display server to restart in *epoptes__enforce* mode.

```
epoptes__enforce_dm: 'mdm'
```

Default variable details

Some of ypid.epoptes default variables have more extensive configuration than simple strings or lists, here you can find documentation and examples for them.

epoptes__rooms

The *epoptes__rooms* dictionary allows you to configure the (class)rooms for Epoptes. The key of a given room is the room name (for example `r23` or `r42`).

Each item is a dictionary itself with the following supported keys:

`room_prefix` Optional, string. Defaults to *epoptes__room_prefix*.

`room_suffix` Optional, string. Defaults to *epoptes__room_suffix*.

`teachers` Required, list of strings. List of teacher hosts.

`teacher_hosts` Optional, list of strings. The teacher hostnames to which clients will connect. Note that Epoptes on a student computer can only be configured to connect to one teacher computer. The first host `teachers` or `teacher_hosts` will be picked and additional once are ignored. This was done so that the role can provide a future proof interface.

`students` Required, list of strings. List of student hosts.

Examples

Configure Epoptes in room `r23` with `r23-pc01` being the teacher host and `r23-pc02` and `r23-pc03` being student hosts:

```
epoptes__rooms:
  'r23':
    teacher_hosts:
      - 'r23-pc01.sint.example.org'
    teachers:
      - 'r23-pc01'
    students:
      - 'r23-pc02'
      - 'r23-pc03'
```

Copyright

Changelog

ypid.epoptes

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.epoptes master - unreleased

ypid.epoptes v0.2.0 - 2017-05-23

Added

- Role documentation based on the DebOps documentation standard for Ansible roles. [ypid]

Changed

- Rename from `ypid.linuxmuster_net-server-epoptes_via_postsync` to `ypid.epoptes`. [ypid]
- Merge `ypid.linuxmuster_net-client-epoptes_via_postsync` into `ypid.epoptes` to have everything in one role. The role has been fully reworked and all inventory variables have been renamed. Refer to the role documentation and *Upgrade from v0.1.X to v0.2.X*. [ypid]

ypid.epoptes v0.1.0 - 2015-11-13

Added

- Initial coding and design. [ypid]

Upgrade notes

The upgrade notes only describe necessary changes that you might need to make to your setup in order to use a new role release. Refer to the *Changelog* for more details about what has changed.

Upgrade from v0.1.X to v0.2.X

All inventory variables have been renamed so you might need to update your inventory. This script can come in handy to update a subset of variable names:

```
#!/bin/bash
## Upgrade inventory variables for migration from ypid.epoptes v0.1.X to v0.2.X.
## The script is idempotent.

git ls-files -z "$(git rev-parse --show-toplevel)" | xargs --null -I '{}' find '{}' -type f -print0 \
 | xargs --null sed --in-place --regexp-extended '
     s/linuxmuster_net_server_epoptes_/epoptes__/g;
     s/epoptes__room_definition/epoptes__rooms/g;
     s/linuxmuster_net_display_manager/epoptes__enforce_dm/g;
     s/epoptes__via_postsync_image_name/epoptes__postsync_image_name/g;
   '
```

The script is bundled with this role under `./docs/scripts/upgrade-from-v0.1.X-to-v0.2.X` and can be invoked from there.

Note that you will need to update your inventory manually according to the documentation. The *epoptes__rooms* variable has been changed from a list to a dictionary which you might need to update manually in your inventory.

CHAPTER 4

Ansible role: ypid.linuxmuster_net_client

Introduction

The `ypid.linuxmuster_net_client` role allows you to configure a linuxmuster.net client.

Installation

This role requires at least Ansible `v2.1.3`. To install it, run:

```
ansible-galaxy install ypid.linuxmuster_net_client
```

Getting started

Example inventory

Add the linuxmuster.net client which should be configured to the `ypid_service_linuxmuster_net_client` Ansible inventory host group:

```
[ypid_service_linuxmuster_net_client]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.linuxmuster_net_client` role:

```yaml
---

- name: Configure a linuxmuster.net client
  hosts: [ 'ypid_service_linuxmuster_net_client' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.linuxmuster_net_client
      tags: [ 'role::linuxmuster_net_client' ]
```

The playbooks is shipped with this role under `docs/playbooks/linuxmuster_net_client.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using the ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::linuxmuster_net_client` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::linuxmuster_net_client:pkgs` Tasks related to system package management like installing or removing packages.

ypid.linuxmuster_net_client default variables

Sections

Client software packages

linuxmuster_net_client__packages_preset

List of preset "global" packages provided by this role to install. If you want to start of with an empty preset, you can set:

```
linuxmuster_net_client__packages_preset: []
```

in your inventory. Refer to the README and the ./vars/ directory of the role for details.

```
linuxmuster_net_client__packages_preset: |
  {{
    (linuxmuster_net_client__packages_preset_ypid|d([]) | list) +
    []
  }}
```

linuxmuster_net_client__packages

List of additional "global" packages to install.

```
linuxmuster_net_client__packages: []
```

linuxmuster_net_client__group_packages

List of packages for a group of hosts (only one group supported).

```
linuxmuster_net_client__group_packages: []
```

linuxmuster_net_client__packages_preset_absent

List of preset "global" packages provided by this role to ensure that they are absent/removed. If you want to start of with an empty preset, you can set:

```
linuxmuster_net_client__packages_preset_absent: []
```

in your inventory.

```
linuxmuster_net_client__packages_preset_absent: |
  {{
    (linuxmuster_net_client__packages_preset_ypid_absent|d([]) | list) +
    []
  }}
```

linuxmuster_net_client__packages_absent

List of additional "global" packages to remove.

```
linuxmuster_net_client__packages_absent: []
```

linuxmuster_net_client__group_packages_absent

List of packages for a group of hosts (only one group supported) remove.

```
linuxmuster_net_client__group_packages_absent: []
```

Linuxmuster.net user settings

linuxmuster_net_client__template_user

User name of the user which acts as template for teachers and students.

```
linuxmuster_net_client__template_user: 'linuxadmin'
```

Manage local user for offline usage

Work in progress: TODO: Test on fresh system. Create user.

Testing: Don’t allow any internet access for local user: iptables -A OUTPUT -m owner –uid-owner localuser -j REJECT

linuxmuster_net_client__local_user_enabled

Should a local user for students and teachers be created on the clients? This user can be used if the client has been booted without network where normal login does not work.

```
linuxmuster_net_client__local_user_enabled: False
```

linuxmuster_net_client__local_user_name

Unix user name to configure.

```
linuxmuster_net_client__local_user_name: 'localuser'
```

linuxmuster_net_client__local_user_script_filepath

File path of the sync_localuser script which resets the settings of the local user with those of the template user. If management of the local user is enabled by *linuxmuster_net_client__local_user_enabled*, this script will be invoked on each reboot.

```
linuxmuster_net_client__local_user_script_filepath: '/usr/local/bin/sync_localuser'
```

linuxmuster_net_client__local_user_directories

List of directories to create in the home directory of the local user.

```
linuxmuster_net_client__local_user_directories:
  - 'Dokumente_tmp'
```

Linuxmuster.net client configuration

FIXME: More documentation/rework of this section needed.

linuxmuster_net_client__copy_custom_config_etc

Configure the client by copying role provided configuration files to `/etc`.

```
linuxmuster_net_client__copy_custom_config_etc: False
```

linuxmuster_net_client__etc_delete_sync

FIXME: Is this needed? Directory paths on the client which should by configured by configuration provided by this role.

> **Warning:** Note that this overwrite or delete all files below the given directory paths.

To enable this for certain directory paths, take a look at this example:

```
linuxmuster_net_client__etc_delete_sync:
  - '/etc/linuxmuster-client/pre-mount.d'
  - '/etc/linuxmuster-client/post-mount.d'
```

By default, this option is disabled.

```
linuxmuster_net_client__etc_delete_sync: []
```

Client auto shutdown

Allows to shutdown clients at a specific time when no user is logged in. Logged in users are given the choice if they are ok with the scheduled shutdown or if they want to continue. If no answer is given, the client is going to shutdown after 2 minutes.

linuxmuster_net_client__auto_shutdown_enabled

Should client auto shutdown be enabled for the times configured by *linuxmuster_net_client__auto_shutdown_time*?

```
linuxmuster_net_client__auto_shutdown_enabled: True
```

linuxmuster_net_client__auto_shutdown_time

Cron entries to perform the auto shutdown.

```
linuxmuster_net_client__auto_shutdown_time:
  - { hour: '17', minute: '15'}
  - { hour: '20' }
  - { hour: '00' }
```

Additional scripts

linuxmuster_net_client__additional_scripts_state

Default state for additional scripts provided by this role.

```
linuxmuster_net_client__additional_scripts_state: 'present'
```

Copyright

```
ypid.linuxmuster_net_client - Configure a linuxmuster.net client

Copyright (C) 2016 Robin Schneider <ypid@riseup.net>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, version 3 of the
License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
```

Changelog

ypid.linuxmuster_net_client

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.linuxmuster_net_client v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

Changed

- Rename Ansible role tag `role::linuxmuster_net_client:software` to `role::linuxmuster_net_client:pkgs` for consistency reasons with DebOps. [ypid]

Fixed

- Installation of the `make_clean_for_image` script. [ypid]

CHAPTER 5

Ansible role: ypid.opsi

Introduction

The `ypid.opsi` role is meant for managing the software distribution and management system Opsi (open PC server integration). Opsi primarily targets Microsoft Windows clients but can also install and manage GNU/Linux hosts.

While this role is designed to be flexible enough to be used with a variety of different Opsi server deployments, the original authors primary use case of this role was to automate paedML Linux school environment deployment and support. In case you have the same intention, chances are you might be interested in the ypid.paedml_linux role.

Because of this, the role currently does not supports setting up the Opsi server because this is already done when deploying the paedML Linux VM templates.

The main focus of the role is to do the finishing touches on the Opsi server like installing additional Opsi products and completing Opsi products containing proprietary software, which can not be shipped in a ready to use form.

Installation

This role requires at least Ansible `v2.1.3`. To install it, run:

```
ansible-galaxy install ypid.opsi
```

Getting started

Example inventory

Add the hosts on which Opsi should be manage to the `ypid_service_opsi` Ansible inventory host group:

```
[ypid_service_opsi]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.opsi` role:

```
---

- name: 'Manage Opsi using Ansible :)'
  hosts: [ 'ypid_service_opsi' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.opsi
      tags: [ 'role::opsi' ]
```

This playbook is shipped with this role under `docs/playbooks/opsi.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::opsi` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::opsi:pkgs` Tasks related to system package management like installing, upgrading or removing packages.

`role::opsi:prod_update` Tasks related to OPSI product updating.

`role::opsi:installfiles` Tasks related to OPSI product installation files.

ypid.opsi default variables

Sections

Packages and installation

opsi__base_packages

List of base packages to install.

```
opsi__base_packages:
  - 'p7zip-full'
```

opsi__deploy_state

What is the desired state which this role should achieve? Possible options:

present Default. Ensure that Opsi is installed and configured as requested.

absent Ensure that Opsi is uninstalled and it's configuration is removed.

Warning: Currently, only `present` is fully supported!

```
opsi__deploy_state: 'present'
```

Opsi product updates

Opsi products are usually installed/updated thought a central repository. This repository can hold a variety of Opsi products which can be updated (and installed if not already present) using the **opsi-product-updater**.

opsi__product_updater_run

Whether or not to run the **opsi-product-updater** during role execution.

Note: This might take a long time, especially on the first run.

```
opsi__product_updater_run: True
```

opsi__product_updater_options

Additional command line options for **opsi-product-updater**.

Refer to the *paedML Linux 6.0* section for details on the specific flags set as default here.

```
opsi__product_updater_options:
  - '-i'
  - '-v'
  - '-v'
```

Install and remove Opsi products

This section allows you to manually install provided Opsi products on the server.

Refer to the *documentation of all options* for more details.

opsi__products

This variable is intended to be used in Ansible's global inventory. More specific variables can overrule less specific variables.

```
opsi__products: {}
```

opsi__products_host_group

This variable is intended to be used in a host inventory group of Ansible (only one host group is supported).

```
opsi__products_host_group: {}
```

opsi__products_host

This variable is intended to be used in the inventory of hosts.

```
opsi__products_host: {}
```

Opsi product installation files

Some Opsi products are not usable out of the box. This section describes how to make those Opsi products usable by providing missing installation files. The reason why they are not usable is because of the restrictive nature of proprietary software (e.g. Microsoft).

Refer to the *documentation of all options* for more details.

opsi__installfiles

This variable is intended to be used in Ansible's global inventory. More specific variables can overrule less specific variables.

```
opsi__installfiles: {}
```

opsi__installfiles_host_group

This variable is intended to be used in a host inventory group of Ansible (only one host group is supported).

```
opsi__installfiles_host_group: {}
```

opsi__installfiles_host

This variable is intended to be used in the inventory of hosts.

```
opsi__installfiles_host: {}
```

opsi__installfiles_achrive_hash_algo_preference

List of hash algorithms, in order of preference starting with the preferred one. The first hash algorithm will be used when it was specified using *archive_checksum*.

Currently, only `sha1` is supported but the role default will use the strongest hash function available in the Ansible `copy` module. The `stat` module could be used to use stronger hash function right now (as of Ansible 2.0) but the `copy` module calculates a checksum anyway so it was decided to wait for proper support in the `copy` module. Checksum calculation and compararchiven should be backwards compatible.

```
opsi__installfiles_achrive_hash_algo_preference: [ 'sha1' ]
```

Internals

opsi__depoy_base_path

Base directory path on the Opsi server where the Opsi products are installed.

```
opsi__depoy_base_path: '/var/lib/opsi/depot'
```

opsi__depoy_installfiles_dir_name

Name of the directory where the install files should be placed.

```
opsi__depoy_installfiles_dir_name: 'installfiles'
```

opsi__products_files_dir_path

Base directory path on the Opsi server where the Opsi products defined in the *Install and remove Opsi products* section will be copied to.

```
opsi__products_files_dir_path: '/home/fixme/ansible_managed'
```

opsi__products_files_owner

System user who owns the *`opsi__products_files_dir_path`* and all files and directories below it.

```
opsi__products_files_owner: 'root'
```

opsi__products_files_group

System group of the *`opsi__products_files_dir_path`* and all files and directories below it.

```
opsi__products_files_group: 'root'
```

opsi__products_files_permissions

Unix permissions of the *`opsi__products_files_dir_path`* and all files and directories below it. Refer to *`chmod(1)`* for details.

```
opsi__products_files_permissions: 'u=rwX,g=rwX,o=rX'
```

opsi__installfiles_archive_dir_path

Base directory path on the Opsi server where the Opsi products are installed.

```
opsi__installfiles_archive_dir_path: '/var/local/archives'
```

opsi__installfiles_archive_tmp_mount_dir_path

Base directory path on the Opsi server where the archive files will be mounted in order to get to the files contained in the archive.

```
opsi__installfiles_archive_tmp_mount_dir_path: '/media/archive'
```

opsi__testing_mode

If set to True, don't run Opsi specific commands in case they are not installed. This allows for role development against a system which does not have Opsi installed.

```
opsi__testing_mode: False
```

Default variable details

Some of ypid.opsi default variables have more extensive configuration than simple strings or lists, here you can find documentation and examples for them.

Install and remove Opsi products

src_opsi_file Optional, string. File path to the Opsi product file on the Ansible controller.

state Optional, string. State of the Opsi product. Defaults to present.

Choices:

present Opsi product will be installed if needed.

copied Opsi product file(s) will be copied to the Opsi server. The installation state of the product will not be altertet.

absent. Opsi product will be removed if needed.

OPSI product installation files

controller_archive_file_path Required, string. File path to the archive file on the Ansible controller to use as the OPSI product's installation files.

archive_checksum Optional, dict. The key is the hashing algorithm, the value is the expected checksum value of the archive file.

Example:

```
archive_checksum:
  sha1: 'da39a3ee5e6b4b0d3255bfef95601890afd80709'
```

Refer to *opsi__installfiles_achrive_hash_algo_preference* for more details.

paedML Linux 6.0

The main documentation on which this role builds on is:

- Installationsanleitung-20160419.pdf (language: German)

Refer to the LMZ homepage for more documentation (language: German).

If you are following this documentation more then once then you found the right tool in this role because it can help you out in automating the repetitive parts of this documentation away.

Refer to ypid.paedml_linux which is the base Ansible role for setting up paedML Linux. The role also contains additional documentation regarding paedML Linux.

Opsi product updates

By default, product updates will be done using the `-i` option to install a set of additional products which would not be installed otherwise.

```
-i Install all downloadable packages from configured repositories (ignores excludes).
```

If you want to influence this behavior, checkout the *`opsi__product_updater_options`* variable.

As of 2016-06-01, `-i` causes the following products to be installed additionally:

- `acroread`
- `moviemaker`
- `ms-encoder`
- `mshotfix-vista-win2008-x64-glb`
- `mshotfix-vista-win2008-x86-glb`
- `mshotfix-win10-win2016-x64-glb`
- `mshotfix-win10-x86-glb`
- `mshotfix-win2003-winxp-x64-deu`
- `mshotfix-win2003-x86-deu`
- `mshotfix-win81-x86-glb`
- `mshotfix-win8-win2012-x64-glb`
- `mshotfix-win8-x86-glb`
- `mshotfix-winxp-x86-deu`
- `msservicepack`
- `nfs`
- `office_2003_hotfix`
- `office-2013-servicepack-x86-deu`
- `office_2016_hotfix`
- `ooffice3`
- `opsi-clonezilla`
- `opsi-local-image-opensuse13-2`
- `opsi-local-image-ubuntu`
- `opsi-local-image-win10-x64`

- `opsi-local-image-win8`
- `opsi-local-image-win81`
- `opsi-local-image-win8-x64`
- `opsi-local-image-winxp`
- `opsi-set-wim-imagenames`
- `paedml-pcsperre`
- `paedml-school-client`
- `paint-net`

If you only need a few of those products, you can also consider to use something like:

```
opsi__product_updater_options:
  - '-p'
  - 'paint-net,opsi-local-image-ubuntu'
```

Experiences with different computer models

In this section the author documents his experiences with various computer models he has worked with and how to best configure them for a school environment.

System firmware setup

TODO

- Is it possible to import system firmware settings using the GUI software thingy that the OEMs provide?

Reasons for changing certain settings

Build-in cameras

The author of this documentation and teachers he has worked with believe that in a school environment the number of occurrences where a build in camera would be actually useful to be very limited. Thus cameras are disabled in the system firmware settings. Additionally, consider convening the cameras with electrical tape just to be sure.

Build-in microphone

Most system firmware settings don't give you control over disabling the build in microphone independently from the audio subsystem.

System firmware for which this has been verified to work:

- HP ProBook 6560b

Without hardware modifications, there seems to be no easy and reliable fix for that.

I/O MMU virtualization

Fujitsu seems to disable this by default. Reason is unclear to the author and it might come in handy when deploying desktop virtualization and as security feature (currently unused by common proprietary OSes). So it was decided to enable it.

Fujitsu

ESPRIMO P556

Infos

- WOL only gets enabled after the system has been put into ACPI G3.

System firmware setup

- Enter setup via F2
- "Security" → "Administrator Password" → set password
- "Advanced" → "Trusted Computing" → "HashPolicy" → "SHA-2"
- "Power" → "Power Failure Recovery" → "Always Off"
- "Power" → "Low Power Soft Off" → "Disabled"
- "Boot" → "Boot Menu" → "Disabled"
- "Boot" → "Boot Removable Media" → "Disabled"
- "Boot" → "Boot Option #1" → NIC

- "Boot" → "Boot Option #2" → data storage device

Fujitsu Lifebook E556

Infos

- Opsi boot image needs a more recent Linux Kernel to recognize the ethernet NIC. Works successfully with Linux 4.4. Check: http://download.opensuse.org/repositories/home:/uibmz:/opsi:/opsi40-experimental/Univention_3.2/amd64/
- `opsi-linux-bootimage` as of 20160608-1 fails somehow to boot/install the OS when the lid is closed. And yes, I am a 100% positive on this. I installed 28 of those machines so I guess you could call this representative. I did not yet verify if the problem also occurs with other models with the same version of `opsi-linux-bootimage`. Needs testing.
- Can be ordered without build in camera :)
- WOL does not seem to work. Even in AC mode. NIC link is shut down in S5. Only works right after leaving the firmware setup with save and power off. Might be related: http://support.fujitsupc.com/CS/Portal/supportsearch.do?srch=FAQ&Series=E%20Series&Model=E556&ProductType=Notebook%20PC#wakeonlan

System firmware setup

- Enter setup via F2
- "Advanced" → "CPU Features" → "Intel(R) VT-d" → "Enabled"
- "Advanced" → "Miscellaneous Configurations" → "Wake up on LAN" → "Enabled"
- "Advanced" → "Miscellaneous Configurations" → "Wake up on LAN" → "On Battery" → "Disabled"
- "Security" → "Set Supervisor Password" → set password
- "Security" → "Boot from Removable Media" → "Supervisor Only"
- "Security" → "Boot Menu" → "Disabled"
- "Boot" → "Boot Priority Order" → Network, HDD, all other sources disabled

Fujitsu Stylistic R726

Infos

- WOL does not seem to work although NIC link is up in S5.

System firmware setup

- Enter setup via F2
- "Security" → "Secure Boot Configuration" → "Secure Boot Option" → "Disabled" Needed to enable CSM.
- "Security" → "Set Supervisor Password" → set password
- "Security" → "Boot from Removable Media" → "Supervisor Only"
- "Security" → "Boot Menu" → "Disabled" (In UEFI mode, the Opsi boot image did not boot).

- "Advanced" → "CSM" → "Enabled" Needed to allow PXE boot of the Opsi boot image.
- "Advanced" → "Internal Device Configuration" → "Internal Cameras" → "Disabled"
- "Advanced" → "Wake up on LAN" → "Enabled"
- "Advanced" → "Intel(R) VT-d" → "Enabled"
- "Boot" → "Boot Priority Order" → Network, SSD, all other sources disabled

Copyright

Changelog

ypid.opsi

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.opsi v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

CHAPTER 6

Ansible role: ypid.paedml_linux

Introduction

The `ypid.paedml_linux` role allows you to automate the deployment of a paedML Linux environment.

The role targets the two servers which are running Univention Corporate Server (which is based on Debian GNU/Linux).

To manage Opsi, there is a separate Ansible role which you are encouraged to checkout called ypid.opsi.

Short overview of IT school solutions*

* That ypid has worked with or that are otherwise common in Germany.

`paedML Windows` Based on Microsoft Windows servers and clients.

`paedML Linux (>= 6.0)` Based on GNU/Linux, FreeBSD servers and Microsoft Windows clients. Although it has the word Linux in the name (where is the GNU btw ;)?), this solution does not really allow to run GNU/Linux on clients which is a shame and can be confusing. TODO: There is a way to install Debian 8 using Opsi, configure it with Ansible and then deploy it to clients using Opsi capture. This needs to be checked out :)

`linuxmuster.net` Was called `paedML Linux` up until 5.1. Then it was decided to let the project die and develop something new (refer to *paedML Linux*). Luckily, the project is quite alive and being lead by the original developer team and the community. If you want to support GNU/Linux and Microsoft Windows clients then checkout linuxmuster.net.

The author of this Ansible role has written several roles to manage linuxmuster.net environments too. See Erfahrungsbericht Epoptes Einrichtung/Ansible (language: German).

For more details, checkout: Vergleich der Musterlösungen: Linux - Windows - Novell (language: German).

Installation

This role requires at least Ansible `v2.1.3`. To install it, run:

```
ansible-galaxy install ypid.paedml_linux
```

Getting started

paedML Linux servers

main The "main" server has by default the hostname "server". It is responsible for the following services: DNS, DHCP, LDAP, print server and providing user profiles and network file shares. Targeted by this role.

opsi The "opsi" server has by default the hostname "backup". It is responsible for software distribution and management system using Opsi. Targeted by this role and by ypid.opsi.

adminvm Microsoft Windows 7 server for various management tasks. Not yet targeted by this role.

firewall Based on PfSense. Not yet targeted by this role.

Example inventory

To run the role on both servers, you can add them either directly to the `ypid_service_paedml_linux` Ansible inventory group:

```
[debops_service_paedml_linux_setup]
server.paedml-linux.lokal
backup.paedml-linux.lokal
```

Or use a slightly more flexible approach like this:

```
[paedml_linux:children]
ypid_service_paedml_linux

[paedml_linux_server_main]
server.paedml-linux.lokal

[paedml_linux_server_opsi]
backup.paedml-linux.lokal

[ypid_service_paedml_linux:children]
paedml_linux_server_main
paedml_linux_server_opsi
```

```
## In case you also want to run the `ypid.opsi` role.
# [ypid_service_opsi:children]
# paedml_linux_server_opsi
```

Note that you will need to use unique hostnames if you intent to manage more than one paedML Linux environment. ypid is wondering if the LMZ does support that :)

Example playbook

Here's an example playbook that uses the `ypid.paedml_linux` role:

```
---

- name: 'Manage paedML Linux using Ansible'
  hosts: [ 'ypid_service_paedml_linux' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.paedml_linux
      tags: [ 'role::paedml_linux' ]
```

The playbooks is shipped with this role under `docs/playbooks/paedml_linux.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using the ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::paedml_linux` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::paedml_linux:pkgs` Tasks related to system package management like installing, upgrading or removing packages.

ypid.paedml_linux default variables

Sections

Packages and installation

`paedml_linux__additional_packages`

List of additional packages to install on both servers.

```
paedml_linux__additional_packages:
  - 'git'
  # - 'ntfsprogs'
```

`paedml_linux__additional_packages_main_server`

List of additional packages to install on the main server.

```
paedml_linux__additional_packages_main_server: []
```

`paedml_linux__additional_packages_opsi_server`

List of additional packages to install on the Opsi server.

```
paedml_linux__additional_packages_opsi_server: []
```

`paedml_linux__packages_upgrade`

If `True`, ensure that all packages are installed in there latest version.

```
paedml_linux__packages_upgrade: True
```

Server roles

`paedml_linux__server_role`

Role of the server.

Refer to *paedML Linux servers* for a list of servers in paedML Linux.

The default is to auto detect the server role based on the hostname. This depends on the hostname not being changed (which is safe to assume because it is very likely not supported by the LMZ).

```
paedml_linux__server_role: 'autodetect'
```

Additonal automation

What more can be automated using other Ansible roles?

- Opsi management using ypid.opsi.
- Copying the documentation to the directory share of the teachers with debops.resources. Example configuration:

```
resources__group_files:

  - src: '/path/to/docs/dir/on/your/local/machine'
    dest: '/home/groups/klassen/lehrer-schule/'
    owner: 'Administrator'
    group: 'Domain Admins'
    mode: 'u=rwX,g=rwX,o=rX'
    # directory_mode: 'u=rwX,g=rwX,o=rwX'
```

Cheat cheat

Author's cheat cheat for setting up paedML Linux which you might find useful. This can also be considered a TODO list of things which can potentially be automated :)

Things to clarify with customer

- Teacher/student user import.
- M$ licenses.
- Should the use of Thumb Drives be allowed for students? If yes, disable link GPO `Musterloesung_Wechselmedienzugriff_W7`.

Things to remember

- Run in `screen`:

  ```
  univention-upgrade --ignoressh
  ```

- Import Root CA certificate into the dedicated browser profile you are using for the environment.
- Change `system_partition_size` to at least 70 GB to have space to install a few applications. (Side note: The applications to install are still nothing compared to the software list put together for a Ubuntu GNU/Linux client in a linuxmuster.net environment)
- In case there are computers in the mix with less than 200 GiB in capacity, consider adopting the `minimal_backup_parition_size` parameter appropriately. So it might not be possible to do a local backup for those computers.
- If there is a upstream Proxy server provided, use it by writing for example `cache_peer wwwproxy.belwue.de parent 8080 3130 default no-query` into `server:/etc/squid3/local.conf`.
- Setup a test client as VM.

Printing

Setting up printing can be confusing. Some hints.

- Optional: Do initial setup (before X.509 certificates have been regenerated and saved) on a peer-to-peer connection.
- Read printer vendor, model and details via SNMP: **`/usr/lib/cups/backend/snmp $printer_name`**.
- Printing protocol preference

1. IPP over SSL/TLS; printers for which I could not get this to work

   – Kyocera ECOSYS P6021cdn cups: 1.7.5-11+deb8u1 device system fw version: 2PT_3F00.003.021 device engine fw version: 2PS_1000.003.001

     With `encryption=required` e. g. TLS required, the device sent a TCP reset after Cups tried a connection upgrade to TLS via HTTP options.

2. AppSocket

- PPD files can be copied to: `/usr/share/ppd/eigene-treiber/` on the main server.
- Printers can be setup in the adminvm using **`printmanagement.msc`**.
- If the printing drivers for Microsoft Windows or Microsoft printing servers are somehow wired, try ignoring them and use "MS Publisher Color Printer". Cups should be able to do the conversion.
- Generic device security: Set language to English (translation can be confusing). Set admin password. Set hostname. Regen X.509 certificate. Enable HTTPS, IPP(s), and AppSocket. Disallow or change SNMP write community. Leave SNMP read as is as this might be useful for Monitoring and is also used by cups. Lock the operation panel. Disable every unneeded protocol or service like HTTP, SMTP, zeroconf, Google Cloud Print, AirPrint and so on.
- Device security hardening. Because the printer firmware is expected to have bugs and school networks can not be trusted, setup some kind of IP filter on the device to only allow access from the IP address of the main server. This also hides things like job history which some printers seem to expose via the web interface. The web interface can be accessed via an TCP tunnel over the main server via SSH and **`ncat`**. Other/additional possibilities (probably overkill): IEEE 802.1X, VLAN, Firewall.
- Setup two printers for one physical printer and limit one of them to only black and white (gray scale) printing. This should make it easier to manage color printing via the school console.
- Restrict access to Cups to not expose print job history and unlimited access to printers by editing `/etc/cups/cupsd.conf`. TODO: Check if user authentication makes sense.

> **Warning:** It seems paedML Linux does not configure Cups on the main server to enforce the printer to room mappings at all! Just add the printer on another Cups server via `https://server.paedml-linux.lokal:631/printers/$printername` and the main server happily printers your jobs unless you restrict it as suggested.

Default passwords

`adminvm` Administrator: nt123

`opsi-boot-image` root: linux123

Hint: You can also login to it via SSH :)

Remote access

You really want to go the OpenVPN route because the OpenSSH version installed is very old (OpenSSH_5.5p1 Debian-6.55.201601151048, OpenSSL 0.9.8o 01 Jun 2010) in terms of cyphers it supports.

OpenVPN in the default configuration of the paedML Linux does also not use the strongest cyphers out there.

Copyright

```
ypid.paedml_linux - Manage paedML Linux using Ansible

Copyright (C) 2016 Robin Schneider <ypid@riseup.net>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, version 3 of the
License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Changelog

ypid.paedml_linux

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.paedml_linux v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

CHAPTER 7

Ansible role: ypid.plymouth

Introduction

The ypid.plymouth role allows you configure a graphical boot and shutdown animation using Plymouth.

Installation

This role requires at least Ansible v2.1.4. To install it, run:

```
ansible-galaxy install ypid.plymouth
```

Getting started

Example inventory

To configure Plymouth on a host it should be included in the ypid_service_plymouth Ansible inventory group:

```
[ypid_service_plymouth]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.plymouth` role:

```
---

- name: Configure a graphical boot and shutdown animation using Plymouth
  hosts: [ 'ypid_service_plymouth' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: debops.grub
      grub_dependent_kernel_options: '{{ plymouth__grub__dependent_kernel_options }}'
      tags: [ 'role::grub' ]

    - role: ypid.plymouth
      tags: [ 'role::plymouth' ]
```

This playbooks is shipped with this role under `./docs/playbooks/plymouth.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::plymouth` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::plymouth:pkgs` Tasks related to system package management like installing or removing packages.

ypid.plymouth default variables

Sections

Packages and installation

`plymouth__base_packages`

List of base packages to install.

```
plymouth__base_packages:
  - 'plymouth'
```

plymouth__packages

List of additional packages to install.

```
plymouth__packages:
  - '{{ ["plymouth-themes"] if (ansible_distribution in ["Debian"]) else [] }}'
  - '{{ ["plymouth-theme-" + plymouth__theme] if (ansible_distribution in ["Ubuntu"])␣
↪else [] }}'
```

plymouth__deploy_state

What is the desired state which this role should achieve? Possible options:

`present` Default. Install and configure Plymouth.

`absent` Remove Plymouth.

```
plymouth__deploy_state: 'present'
```

Customization

plymouth__theme

The theme to enable. You can run **`plymouth-set-default-theme --list`** to get a list of installed themes.

```
plymouth__theme: 'solar'
```

plymouth__show_delay

On systems that boot quickly, you may only see a flicker of your splash theme before your DM or login prompt is ready. You can set this to an interval (in seconds) longer than your boot time to prevent this flicker and only show a blank screen.

If this variable is set to the special value `{{ omit }}`, the configuration is not altered.

```
plymouth__show_delay: '{{ omit }}'
```

Configuration for other Ansible roles

plymouth__grub__dependent_kernel_options

Configuration for the debops.grub role.

```
plymouth__grub__dependent_kernel_options:
  - 'splash'
```

Copyright

Changelog

ypid.plymouth

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.plymouth v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

CHAPTER 8

Ansible role: ypid.shutdown_if_battery_low

Introduction

The ypid.shutdown_if_battery_low role allows you to configure a system to shutdown if the build-in battery is low.

Installation

This role requires at least Ansible `v2.1.3`. To install it, run:

```
ansible-galaxy install ypid.shutdown_if_battery_low
```

Getting started

Example inventory

Add the hosts on which should shutdown if the build-in battery is low to the `ypid_shutdown_if_battery_low` Ansible inventory group:

```
[ypid_shutdown_if_battery_low]
hostname
```

Example playbook

Here's an example playbook that uses the ypid.shutdown_if_battery_low role:

```
---

- name: Configure shutdown if battery low
  hosts: [ 'ypid_shutdown_if_battery_low' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.shutdown_if_battery_low
      tags: [ 'role::shutdown_if_battery_low' ]
```

The playbooks is shipped with this role under `docs/playbooks/shutdown_if_battery_low.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using the ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::shutdown_if_battery_low` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

ypid.shutdown_if_battery_low default variables

shutdown_if_battery_low__limit_warn

When the battery capacity in percent is less then the following value, write a log entry check interval (refer to *shutdown_if_battery_low__sleep*).

```
shutdown_if_battery_low__limit_warn: 60
```

shutdown_if_battery_low__limit_shutdown

When the battery capacity in percent is less then the following value, the system is going to shutdown.

```
shutdown_if_battery_low__limit_shutdown: 20
```

shutdown_if_battery_low__shutdown_command

Command to execute when `battery capacity in percent` <= *shutdown_if_battery_low__limit_shutdown*.

```
shutdown_if_battery_low__shutdown_command: 'shutdown --poweroff +5'
```

shutdown_if_battery_low__sleep

Check interval/sleep time between check.

```
shutdown_if_battery_low__sleep: '6m'
```

shutdown_if_battery_low__script_filepath

File path where the main script will be copied to on the remote system.

```
shutdown_if_battery_low__script_filepath: '/usr/local/bin/shutdown_if_battery_low.sh'
```

shutdown_if_battery_low__environment_filepath

File path where the configuration variables are saved on the remote system.

```
shutdown_if_battery_low__environment_filepath: '/etc/default/shutdown_if_battery_low'
```

Copyright

```
ypid.shutdown_if_battery_low - Shutdown the system if the build-in battery is low

Copyright (C) 2015-2016 Robin Schneider <ypid@riseup.net>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, version 3 of the
License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Changelog

ypid.shutdown_if_battery_low

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.shutdown_if_battery_low master - unreleased

ypid.shutdown_if_battery_low v0.1.0 - 2015-11-10

Added

- Initial coding and design. [ypid]

Changed

- Changed namespace from `shutdown_if_battery_low_` to `shutdown_if_battery_low__`. `shutdown_if_battery_low_[^_]` variables are hereby deprecated and you might need to update your inventory. This oneliner might come in handy to do this.

  ```
  git ls-files -z | xargs --null -I '{}' find '{}' -type f -print0 | xargs --null␣
  ↪sed --in-place --regexp-extended 's/\<(shutdown_if_battery_low)_([^_])/\1__\2/g;
  ↪'
  ```

  [ypid]

CHAPTER 9

Ansible role: ypid.whereami

Introduction

This role setups up and configures whereami.

`whereami` allows to non-iteratively ascertain the location of the computer and reconfigure the system appropriately.

Installation

This role requires at least Ansible `v2.1.3`. To install it, run:

```
ansible-galaxy install ypid.whereami
```

Getting started

Example inventory

To configure **whereami** on hosts you can put them into the `ypid_service_whereami` Ansible inventory group:

```
[ypid_service_whereami]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.whereami` role:

```
---

- name: Setup and configure whereami
  hosts: [ 'ypid_service_whereami' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.whereami
      tags: [ 'role::whereami' ]
```

The playbooks is shipped with this role under `docs/playbooks/whereami.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using the ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after host is first configured to speed up playbook execution, when you are sure that most of the configuration has not been changed.

Available role tags:

`role::whereami` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::whereami:pkgs` Tasks related to system package management like installing or removing packages.

`role::whereami:install` Tasks related to the installation and patching of **`whereami`**.

`role::whereami:config` Tasks related to the configuration of **`whereami`**.

ypid.whereami default variables

Sections

Installation options

`whereami__base_packages`

List of base packages to install.

```
whereami__base_packages:
  - 'whereami'
```

whereami__install_additional_scripts

List of additional scripts and tests for whereami to install. Those scripts come with this role as they have not yet reached upstream or did not yet have enough time to flow downstream).

Choices:

testip Test if a certain network interface has been assigned a certain network address or other flags using regular expressions on the output of the **ip** command.

testnm-device Test against device network-manager can provide.

Check if wlan0 is currently connected to "connection_name":

```
./testnm-device "wlan0,GENERAL\.CONNECTION:connection_name"
```

testnm-connection Test against connection information network-manager can provide.

Check if given connection UUID is active:

```
./testnm-connection 'tun\d+,c910b0bc-5733-4093-812e-5f2a2d806314'
```

testneighbour Test for known hosts in the neighbour table of the kernel.

```
whereami__install_additional_scripts:
  - 'testip'
  - 'testnm-device'
  - 'testnm-connection'
  - 'testneighbour'
```

whereami__install_patches

Install a bug fixed version of **whereami** instead of the upstream version.

Fixed:

- Sort locations to not report a location change only because the order has changed.

```
whereami__install_patches: False
```

whereami__install_to_if_up_d

Install the hook script also to `/etc/network/if-up.d/`. This is needed when you use network manager because it does not run scripts on `if-pre-up.d`.

```
whereami__install_to_if_up_d: False
```

whereami__install_to_if_post_down_d

Install the hook script also to `/etc/network/if-post-down.d/` to run **whereami** also when a device goes down.

```
whereami__install_to_if_post_down_d: False
```

Configuration

whereami__configuration_mode

UNIX permissions for configuration files.

```
whereami__configuration_mode: '0644'
```

whereami__location_detection_rules

The detection configuration for **whereami**.

This file describes unique features of various environments you connect to and associates them with those locations.

When a test is successful, no further tests will be conducted until a subsequent 'if' or 'fi' statement, or if the line is preceded by the word 'always'.

You may also set environment variables for the scripts (test and action) subsequently called by whereami using "set <varname> <value>".

See *detect.conf(5)* for lots more information, especially for the details of the helper scripts used here.

```
whereami__location_detection_rules: |
  # It is a good idea to default to somewhere...
  default undocked

  always testip eth0,LOWER_UP  lan
  always testip wlan0,LOWER_UP wlan

  # Uncomment this for exhaustive debugging output
  # set DEBUGWHEREAMI 1

  # 'down' might be a location that is forced from the command line
  # like "whereami down"
  if down
    always at undocked
    always notat eth0,wlan0
  fi

  if lan
    testip 'eth0,inet 192\.0\.2\.\d+/24'    location1,location1-lan
    testip 'eth0,inet 198\.51\.100\.\d+/24' location1,location1-lan
  fi

  if wlan
    testnm-connection 'wlan0,GENERAL\.CONNECTION:location1-wlan' location1,location1-
↪wlan
    testnm-connection 'wlan0,GENERAL\.CONNECTION:location2-wlan' location2,location2-
↪wlan
  fi
```

whereami__action_rules

The action configuration for **whereami**. Check *whereami.conf(5)* for details.

```
whereami__action_rules: |
  # #
  # # Sample configuration file for 'whereami' copyright Andrew McMillan, 1999
  # #

  # # Regardless, we always rotate the netscape preferences file.
  # =any savelog -p /home/andrew/.netscape/preferences.js
  # =any cat /home/andrew/.netscape/preferences.js.0 >>/home/andrew/.netscape/
↪preferences.js
```

```
# # By default just send mail directly across the internet
# =any MAIL_RELAY="none"
# # By default send all mail via our main mailserver, since we do TLS+SASL there
# =any MAIL_RELAY="smtp.mydomain.com"
# By default we assume that the detection configured routing
=any GATEWAY=""
# By default we assume we have an unfiltered internet connection
=any PROXY="none"
=any NTPDAEMON=/etc/init.d/ntp-server
=any NTPSERVER=ntp


## You will need to install the "resolvconf" package for these to
## work, and edit the following couple of stanzas to the right
## interface names.

## This only happens if we are not at a WLAN
#!wlan ifconfig ath0 down
#!wlan resolvconf -d ath0

## This only happens if we are not at a LAN
#!lan ifconfig eth0 down
#!lan resolvconf -d eth0


# # Catalyst
# +catalyst bind-forwarders 192.168.3.10
# +catalyst setresolver search laptop.cat-it.co.nz wgtn.cat-it.co.nz catalyst.net.nz nameserver 127.0.0.1
# +catalyst mount /plato/general
# +catalyst mount /home/andrew/plato
# =catalyst export PROXY="192.168.3.10 8080"
# =catalyst export NTPSERVER="plato"
# =catalyst export GATEWAY="192.168.3.1"


# # At home
# =home export PROXY="172.16.85.3 3128"
# =home export NTPSERVER="adorno"
# =home setresolver search localhost mcmillan.net.nz nameserver 172.16.85.2
# +home smbmount //hegel/c /hegel/c -I 172.16.85.10 -N >>/dev/null
# -home umount /hegel/* >>/dev/null

# # AmTrust in Auckland
# +amtrust bind-forwarders 192.168.5.1
# +amtrust setresolver search laptop.cat-it.co.nz akld.capit.co.nz catalyst.net.nz nameserver 127.0.0.1

# # AGP in Sydney
# +agp bind-forwarders 10.3.0.10
# +agp setresolver search laptop.cat-it.co.nz agp.com.au catalyst.net.nz nameserver 127.0.0.1

# # When we're in Sydney we want to change to the correct timezone, and
# # then change it back when we leave.
# +agp rm /etc/localtime
# +agp ln -s /usr/share/zoneinfo/Australia/Sydney /etc/localtime
# -agp rm /etc/localtime
```

```
  # -agp ln -s /usr/share/zoneinfo/Pacific/Auckland /etc/localtime


  # # TTP In Auckland
  # =ttpak export MAIL_RELAY=10.2.0.59
  # =ttpak export PROXY="10.2.0.1 3128"

  # # First we mount the drives if we are in the docking station.  These always
  # # get unmounted cleanly because we have to shut down to remove the laptop.
  # =catalyst if ( grep "^docked" /var/lib/whereami/iam >/dev/null ) then
  # =catalyst   mount /dev/hde1 /dos/f -t vfat
  # =catalyst   mount /dev/hde2 /dock -t ext2
  # =catalyst   /root/bin/bkup2dock &
  # =catalyst fi

  # =docked mount /dos/f; mount /dock
  # =docked /root/bin/bkup2dock &

  # =undocked killall dhclient 2>/dev/null;  killall dhclient3 2>/dev/null
  # =undocked MAIL_RELAY=queue

  # # So we can override this for any configuration
  # =any setmailrelay ${MAIL_RELAY}
  # =any setoops ${PROXY}

  # # Set any gateway that got called for in passing
  # =any [ -n "${GATEWAY}" ] && route add default gw ${GATEWAY} 2>/dev/null

  # # Use any reasonable NTP server
  # =undocked if /bin/false ; then
  # =any ${NTPDAEMON} stop
  # =any ntpdate ${NTPSERVER}
  # =any ${NTPDAEMON} start
  # =undocked fi

  # =any /usr/local/sbin/local-firewalling start
  # =any sync
  # =any ( /etc/init.d/cupsys stop; sleep 10; /etc/init.d/cupsys start ) >/dev/null 2>
↪&1 &
```

Copyright

```
ypid.whereami - Setup and configure whereami

Copyright (C) 2016 Robin Schneider <ypid@riseup.net>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, version 3 of the
License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Affero General Public License for more details.
```

Changelog

ypid.whereami

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.whereami v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

Changed

- Changed namespace from `whereami_` to `whereami__`. `whereami_[^_]` variables are hereby deprecated and you might need to update your inventory. This oneliner might come in handy to do this.

  ```
  git ls-files -z | xargs --null -I '{}' find '{}' -type f -print0 | xargs --null␣
  ↪sed --in-place --regexp-extended 's/\<(whereami)_([^_])/\1__\2/g;'
  ```

  [ypid]

CHAPTER 10

Ansible role: ypid.wine

Introduction

The ypid.wine role allows you to setup and mange Wine. Wine is a free and open source compatibility layer software application that aims to allow applications designed for Microsoft Windows to run on Unix-like operating systems.

Installation

This role requires at least Ansible v2.1.3. To install it, run:

```
ansible-galaxy install ypid.wine
```

Getting started

Example inventory

Add the hosts on which Wine should be setup to the ypid_service_wine Ansible inventory group:

```
[ypid_service_wine]
hostname
```

Example playbook

Here's an example playbook that uses the `ypid.wine` role:

```
---

- name: Setup and manage Wine for running MS Windows programs
  hosts: [ 'ypid_service_wine' ]
  become: True

  environment: '{{ inventory__environment | d({})
                   | combine(inventory__group_environment | d({}))
                   | combine(inventory__host_environment  | d({})) }}'

  roles:

    - role: ypid.wine
      tags: [ 'role::wine' ]
```

The playbooks is shipped with this role under `docs/playbooks/wine.yml` from which you can symlink it to your playbook directory. In case you use multiple roles maintained by ypid, consider using the ypid-ansible-common.

Ansible tags

You can use Ansible `--tags` or `--skip-tags` parameters to limit what tasks are performed during Ansible run. This can be used after a host was first configured to speed up playbook execution, when you are sure that most of the configuration is already in the desired state.

Available role tags:

`role::wine` Main role tag, should be used in the playbook to execute all of the role tasks as well as role dependencies.

`role::wine:pkgs` Tasks related to system package management like installing or removing packages.

ypid.wine default variables

Sections

Packages and installation

`wine__base_packages`

List of base packages to install.

```
wine__base_packages:
  - '{{ [ "wine32" ] if (ansible_distribution == "Debian") else [] }}'
  - '{{ [ "libwine-print:i386" ] if (ansible_distribution == "Debian") else [] }}'
  - 'winetricks'
  ## Needed for wine. https://www.winehq.org/docs/wineusr-guide/misc-things-to-
↪configure#AEN942
  - 'cups-bsd'

  ## Needed for printing. https://forums.linuxmint.com/viewtopic.php?t=168952
  - 'libcups2:i386'

  ## Does not have wine-gecko?
  # - 'wine-development'
```

`wine__architecture_i386_enabled`

List of base packages to install.

```
wine__architecture_i386_enabled: True
```

`wine__deploy_state`

What is the desired state which this role should achieve? Possible options:

`present` Default. Ensure that Wine is installed and configured as requested.

`absent` Ensure that Wine is uninstalled and it’s configuration is removed.

```
wine__deploy_state: 'present'
```

Copyright

Changelog

ypid.wine

This project adheres to Semantic Versioning and human-readable changelog.

The current role maintainer is ypid.

ypid.wine v0.1.0 - unreleased

Added

- Initial coding and design. [ypid]

Index