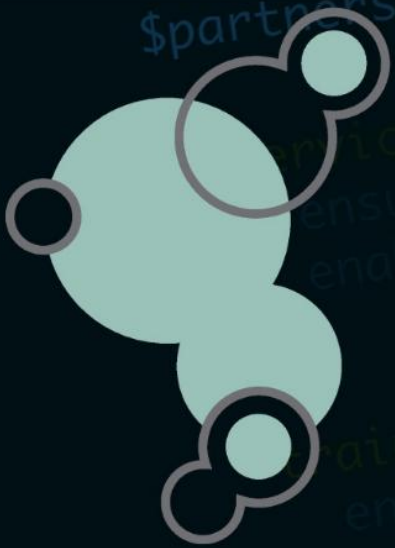


```
# Open-Future
#
# Installs, configures and keeps your open source
# infrastructure up and running.
```

```
class openfuture (
  $consultants = ['you?', 'johan', 'bert', 'patrik', 'patrick'],
  $sales        = ['ann'],
  $services     = ['infrastructure', 'consultancy', 'training'],
  $trainings    = ['zabbix', 'bacula', 'puppet', 'linux'],
  $partners     = ['nico', 'danny']
```

Zabbix automation

How to automate Zabbix with Puppet & the Zabbix API

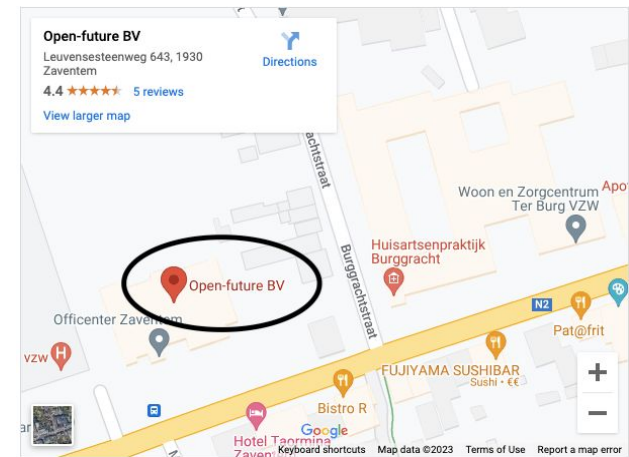


open-future

Author : Bert Deferme

Who is Open-Future

- Founded in 2009 by Danny and Nico
- Specialized in open-source solutions
- Focus on open-source partnerships with vendors
- Partnerships with Red Hat, Bacula, SEP, Zabbix, ...
- Provides Official Trainings for Bacula, Puppet and Zabbix
- One of the oldest Zabbix partners
- Provides trainings in our office, onsite and online



Who am I

- Bert Deferme
- An open-source consultant working for Open-Future
- 13+ years of experience in IT
- Started with Gentoo and RedHat 5
- Strong interest in
 - Config management
 - Monitoring
 - Troubleshooting

Why automate Zabbix?

- Enforce consistency

<input type="checkbox"/>	lac-apigw01c01	Applications 2	Items 42	Triggers 21	Graphs 10	Discovery 1	Web	lac-apigw01c01.onprvp.fgov.be: 10050
<input type="checkbox"/>	lac-apigw01c02	Applications 2	Items 42	Triggers 21	Graphs 10	Discovery 1	Web	10.200.4.32: 10050



Why automate Zabbix?

- Deliver rapidly at scale

Timestamp	Level	Message
16:18:54	notice	/Stage[main]/Zabbix::Resources::Web/Zabbix_host[llb-xlrelease01]/ensure created
16:18:55	notice	/Stage[main]/Zabbix::Resources::Web/Zabbix_host[llb-xldeploy01]/ensure created
16:18:55	notice	/Stage[main]/Zabbix::Resources::Web/Zabbix_host[llb-dem01]/ensure created
16:18:56	notice	Applied catalog in 32.24 seconds

Why automate Zabbix?

- Increase productivity and efficiency

```
server.pp > role::zabbix::server ●  
1 # @summary Set up a Zabbix server.  
2 #  
3 class role::zabbix::server {  
4     include profile::base  
5     include profile::zabbix::server  
6 }
```



Why automate Zabbix?

- Infrastructure as code also serves as documentation

```
67
68  $_zabbix_version = '6.0'
69
70  # Zabbix tuning params
71  $_zabbix_server_settings = {
72      startpingers           ⇒ 15,
73      startpollersunreachable ⇒ 15,
74      cachesize              ⇒ '192M',
75      trendcachesize         ⇒ '12M',
76      valuecachesize         ⇒ '64M',
77      timeout                ⇒ 15,
78      logslowqueries         ⇒ 3000,
79  }
80
```



Why automate Zabbix?

- Versioning

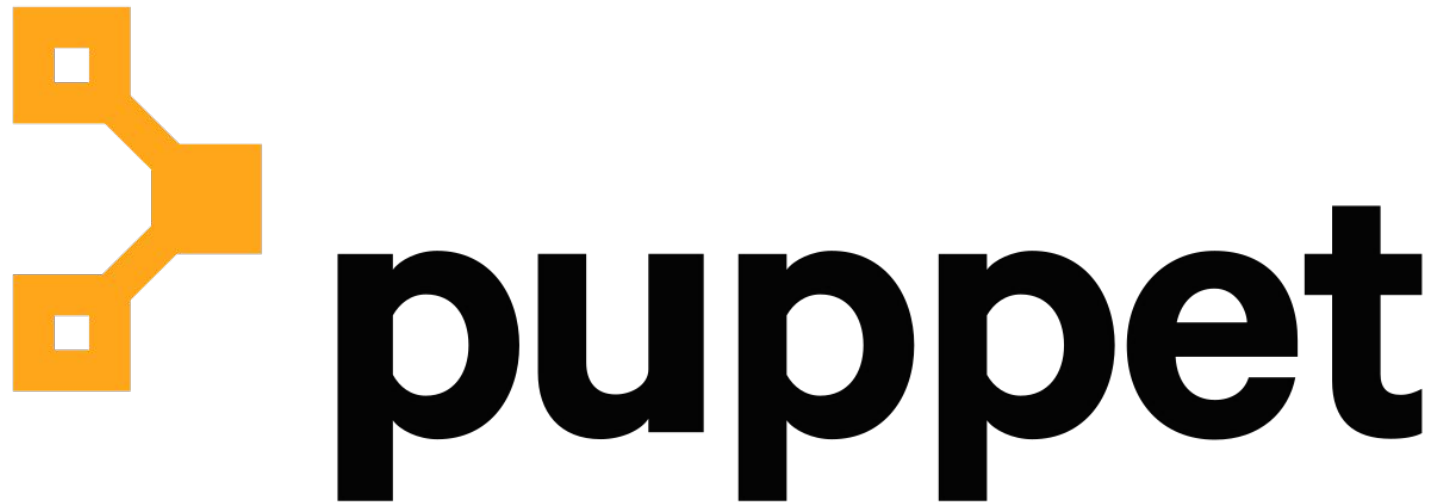
NeogitLogView

```
* 3487a8c [grafana] Use the Operating System dashboards
* 3b025ec [grafana] Add dashboards for Operating Systems
* 1d12a0a [grafana] Move profile::grafana::dashboards
* f83d5cb [Grafana/Alertmanager] Improve Operations dashboards
* dbb66ef Fix mediawiki authentication
* 8a9f0f5 Refactor graylog
* 3d2a99c Refactor nexus repository type
* 846a254 Remove redundant time unit from alerts
* 0ea836c Add key storage path to node resource file
* 1d81603 Refactor Elasticsearch
* 9c43f7b Refactor mongodb
* 6f780a1 Add Mediawiki admin auth group for cbi
* da223d6 Add require jboss class for selinux fuse
* 59c4981 Update fuse profile
* ce94b10 [prom2teams] Refactor `$teams_webhook` variables
* e7aba33 [prom2teams] Add route for mailsorting alerts (and only alert for production)
```



How to automate Zabbix?

- We chose Puppet in favor of Ansible
 - There was already Puppet knowledge in the team
 - Very good upstream puppet modules
- However, we also use Ansible
 - Provisioning tool
 - Orchestration



“Zabbix is difficult to automate”


- I hear this a lot
- I want to testament that it is not



“Zabbix is difficult to automate”

- Ansible integration with API



 Documentation ANSIBLEFEST PRODUCTS COMMUNITY WEBINARS & TRAINING BLOG

Namespace

Collections in the Senu Namespace

Collections in the Splunk Namespace

Collections in the T_systems_mms Namespace

Collections in the Theforeman Namespace

Collections in the Vmware Namespace

Collections in the Vultr Namespace

Collections in the Vynos Namespace

Collections in the Wti Namespace

Indexes of all modules and plugins

Playbook Keywords

Return Values

Ansible Configuration Settings

Controlling how Ansible behaves: precedence rules


YAML Syntax

Modules

- [zabbix_action module](#) – Create/Delete/Update Zabbix actions
- [zabbix_authentication module](#) – Update Zabbix authentication
- [zabbix_autoregister module](#) – Update Zabbix autoregistration
- [zabbix_discovery_rule module](#) – Create/delete /update Zabbix discovery rules
- [zabbix_globalmacro module](#) – Create/update /delete Zabbix Global macros
- [zabbix_group module](#) – Create/delete Zabbix host groups
- [zabbix_group_info module](#) – Gather information about Zabbix hostgroup
- [zabbix_host module](#) – Create/update/delete Zabbix hosts
- [zabbix_host_events_info module](#) – Get all triggers about a Zabbix host

“Zabbix is difficult to automate”

- Puppet Integration through modules



forge Modules ▾ Topics ▾ Resources ▾

Search Forge modules 🔍

Show search options ▾

Home > Modules > puppet > zabbix

zabbix

APPROVED

Installing and maintaining Zabbix. Will install server, proxy, java-gateway and agent on RedHat/Debian/Ubuntu (Incl. exported resources).

[Project URL](#) [RSS Feed](#) [Report issues](#)


Version information

9.2.0 (latest) ▾ released Jun 6th 2022

This version is compatible with:

- Puppet Enterprise 2023.0.x, 2021.7.x, 2021.6.x, 2021.5.x, 2021.4.x, 2021.3.x, 2021.2.x, 2021.1.x, 2021.0.x, 2019.8.x, 2019.7.x, 2019.5.x, 2019.4.x, 2019.3.x, 2019.2.x, 2019.1.x
- Puppet >= 6.1.0 < 8.0.0
- [RedHat](#), [Amazon](#), [OracleLinux](#), [Scientific](#), [CentOS](#), [CloudLinux](#), [XenServer](#), [Ubuntu](#), [Debian](#), [VirtuozzoLinux](#), [Archlinux](#), [Gentoo](#), [Raspbian](#), [windows](#), [AIX](#)

MODULE AUTHOR

 **Vox Pupuli**

[puppet](#)

MODULE STATS


📄 **292,280** downloads

📅 **3,858** latest version

🌟 **5.0** quality score ⓘ

“Zabbix is difficult to automate”

- Very good (and wel documented) API



ZABBIX Documentation 6.4 (current) English

Zabbix Manual

- 1. Introduction
- 2. Definitions
- 3. Zabbix processes
- 4. Installation
- 5. Quickstart
- 6. Zabbix appliance
- 7. Configuration
- 8. Service monitoring
- 9. Web monitoring
- 10. Virtual machine monitoring
- 11. Maintenance
- 12. Regular expressions
- 13. Problem acknowledgment
- 14. Configuration export/import
- 15. Discovery
- 16. Distributed monitoring
- 17. Encryption
- 18. Web interface
- 19. API
 - Method reference
 - Action
 - Alert
 - API info
 - Audit log
 - Authentication
 - Autoregistration

host.create

Description

`object host.create(object/array hosts)`

This method allows to create new hosts.

Note: This method is only available to *Admin* and *Super admin* user types. Permissions to call the method can be revoked in user role settings. See [User roles](#) for more information.

Parameters

(object/array) Hosts to create.

Additionally to the [standard host properties](#), the method accepts the following parameters.

Parameter	Type	Description
groups	object/array	Host groups to add the host to. The host groups must have the <code>groupid</code> property defined. Parameter behavior: - <i>required</i>
interfaces	object/array	Interfaces to be created for the host.
tags	object/array	Host tags .
templates	object/array	Templates to be linked to the host. The templates must have the <code>templateid</code> property defined.
macros	object/array	User macros to be created for the host.
inventory	object	Host inventory properties.

Return values

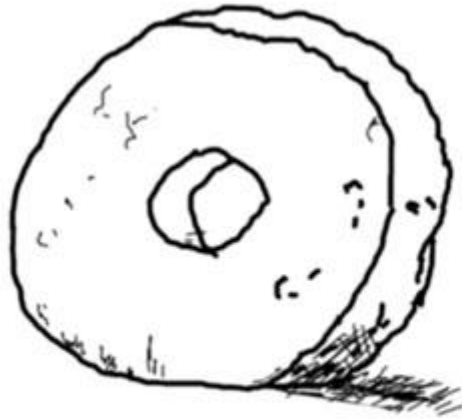
(object) Returns an object containing the IDs of the created hosts under the `hostids` property. The order of the returned IDs matches the order of the passed hosts.

Examples

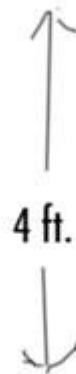
Creating a host

Don't reinvent the wheel

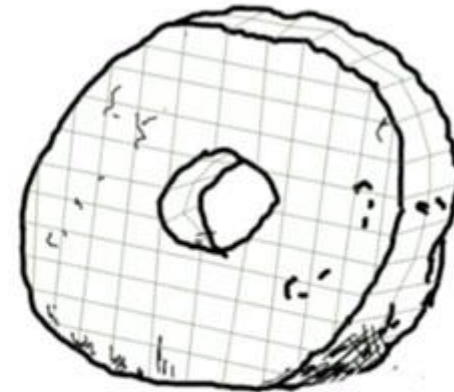
- Re-use excellent upstream modules
 - <https://github.com/voxpupuli/puppet-zabbix>



1. Prehistoric wheel

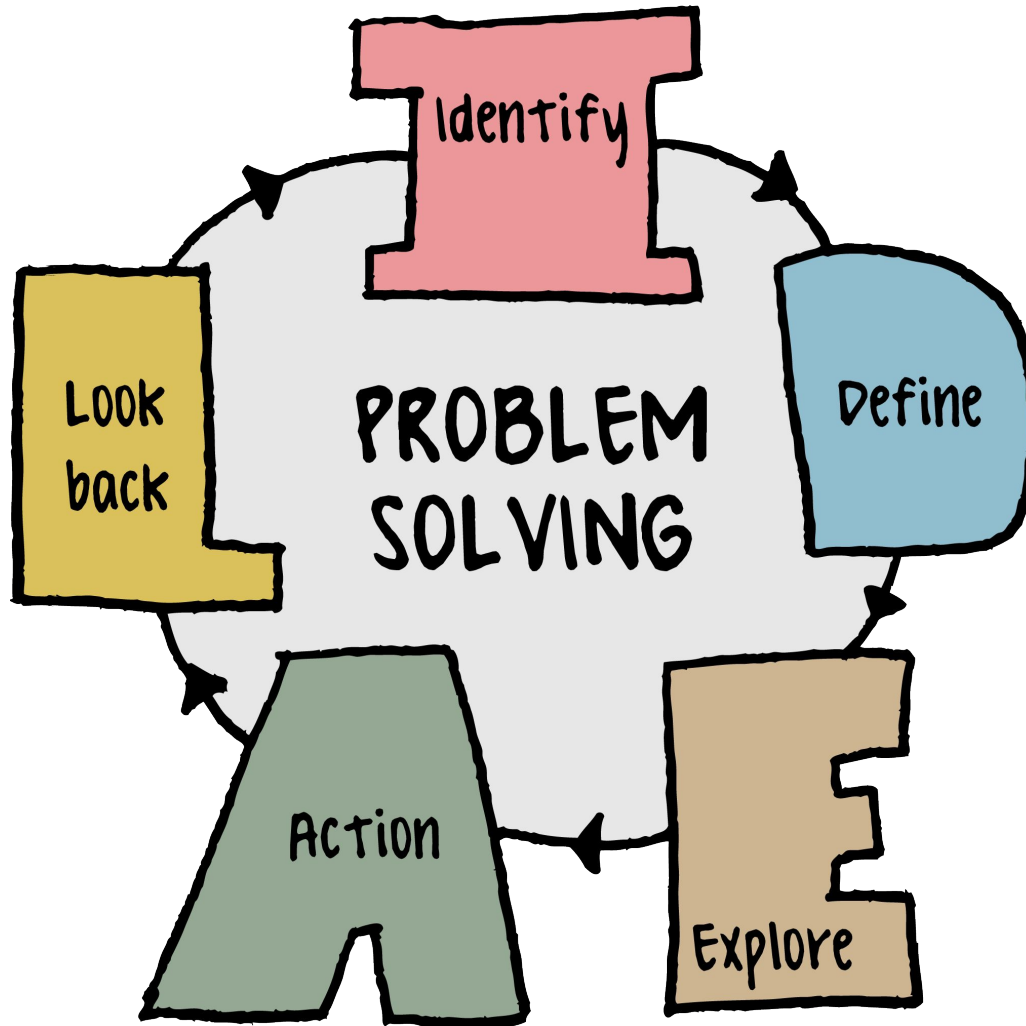


4 ft.



2. 3D Computer Model

The problems we encountered



Problem solving: Sensitive support

- No Sensitive support in the upstream module
 - <https://github.com/voxpupuli/puppet-zabbix/pull/828>



Problem solving: TimescaleDB support

- We use Zabbix on PostgreSQL with TimescaleDB
- No TimescaleDB support in the upstream module
 - <https://github.com/voxpupuli/puppet-zabbix/pull/827>



Problem solving: Setting the admin password

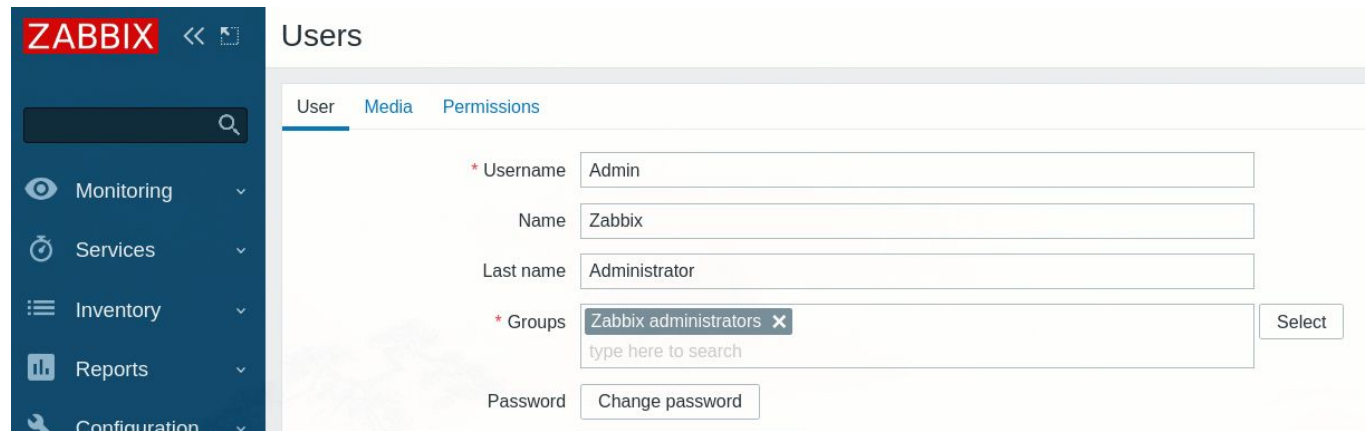
- No puppet resource for Zabbix Users (Before)

```
8  $zabbix_admin_pw_settings_query = [  
7    'UPDATE',  
6    'public.users',  
5    'SET',  
4    "passwd = md5('${admin_password.unwrap}'))",  
3    'WHERE',  
2    'userid = 1',  
1  ]  
215 $zabbix_admin_pw_check_query = [  
1    'SELECT 1 FROM',  
2    'public.users',  
3    'WHERE',  
4    'userid = 1',  
5    'AND',  
6    "passwd = md5('${admin_password.unwrap}'))",  
7  ]  
8  postgresql_psql { 'Update INTERNAL Admin Password':  
9    connect_settings => $connect_settings,  
10   db                => $_servername,  
11   command           => Sensitive($zabbix_admin_pw_settings_  
12   unless            => Sensitive($zabbix_admin_pw_check_que  
13   psql_user          => 'zabbix',  
14   psql_group         => 'zabbix',  
15   refreshonly       => true,  
16   require            => [  
17     Class['zabbix::database::postgresql'],  
18     Class['postgresql::client'],  
19   ],  
20 }
```



Problem solving: Setting the admin password

- Zabbix User provider (After)
 - <https://github.com/voxpupuli/puppet-zabbix/pull/829>

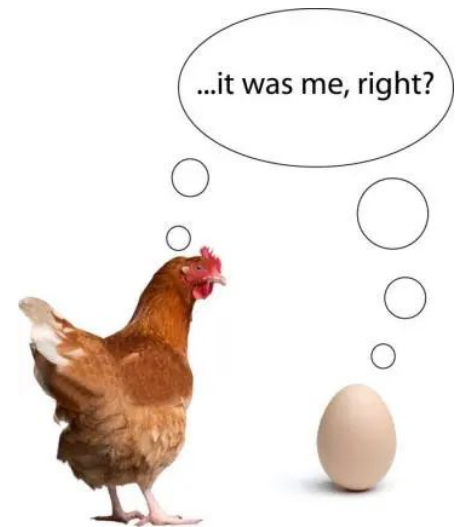


```
3
4  zabbix_user { 'Admin':
5      ensure      => present,
6      firstname   => 'Zabbix',
7      role        => 'Super admin role',
8      surname     => 'Administrator',
9      usrgrps     => ['Zabbix administrators'],
10     passwd      => $_admin_password,
11     require     => Zabbix_usergroup['Zabbix administrators'], # Don't lock ourselves out
12 }
13
```

Problem solving: Setting the admin password

- Zabbix User provider
 - Bonus: Chicken versus egg problem
 - Resolved using a custom fact that checks whether the current admin password is set to the default.

```
1 # Use default password unless the password was changed already
2 $_server_api_pass = $facts['zbx_admin_passwd_default'] ? {
3   true      ⇒ Sensitive('zabbix'),
4   default ⇒ $_admin_password,
5 }
6
7
```



Problem solving: Managing groups

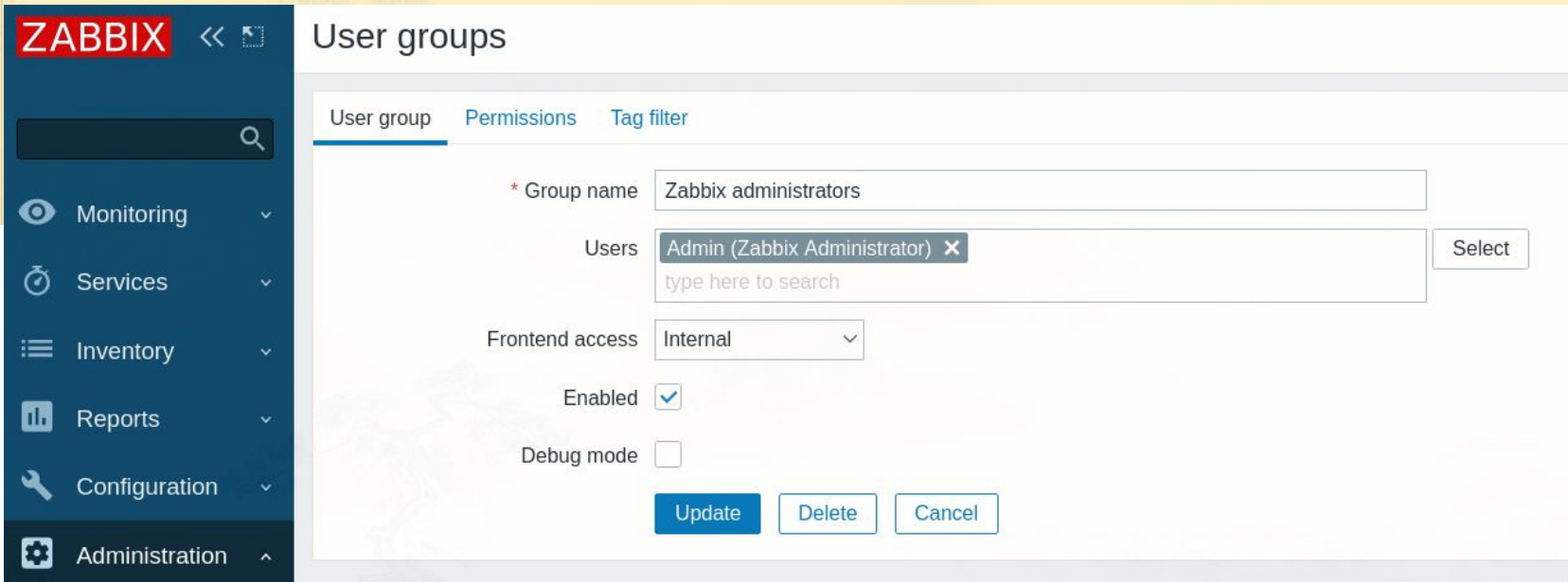
- No puppet resource for Zabbix Groups (Before)

```
16 $zabbix_admin_member_settings_query = [  
15   'UPDATE',  
14   'users_groups',  
13   'SET',  
12   "usrgrpid = (select usrgrpid FROM usrgrp WHERE name = 'Zabbix INTERNAL Administrators')",  
11   'WHERE',  
10   'userid=1',  
9   ]  
8 $zabbix_admin_member_check_query = [  
7   'SELECT 1 FROM',  
6   'users_groups',  
5   'WHERE',  
4   "usrgrpid = (select usrgrpid FROM usrgrp WHERE name = 'Zabbix INTERNAL Administrators')",  
3   'AND',  
2   'userid=1',  
1   ]  
252 postgresql['Zabbix INTERNAL Admin Group']:  
1   connect => connect_settings,  
2   db => servername,  
3   command => $zabbix_admin_member_settings_query.join(' '),  
4   query => $zabbix_admin_member_check_query.join(' '),  
5   type => 'zabbix',  
6   user => 'zabbix',  
7   ... => ['postgresql'],  
8   ... => ['Zabbix INTERNAL Admin Group'],
```

Problem solving: Managing groups

- Zabbix Group provider (After)
 - <https://github.com/voxpupuli/puppet-zabbix/pull/829>

```
1
2 # Update admin password
3 # Make sure 'Zabbix administrators' uses internal authentication
4 zabbix_usergroup { 'Zabbix administrators':
5   ensure      => present,
6   gui_access  => 'internal',
7 }
8
```

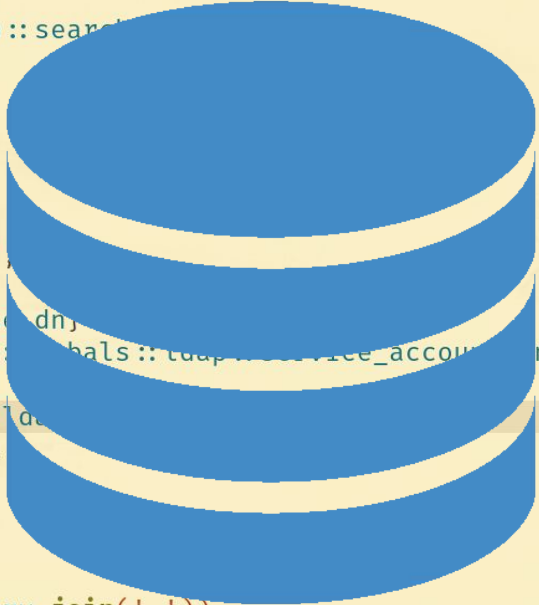


The screenshot displays the Zabbix web interface. On the left is a dark blue sidebar with the Zabbix logo and a navigation menu including Monitoring, Services, Inventory, Reports, Configuration, and Administration (which is highlighted). The main panel is titled 'User groups' and has three tabs: 'User group' (active), 'Permissions', and 'Tag filter'. Under the 'User group' tab, there are several configuration fields: 'Group name' is 'Zabbix administrators'; 'Users' is a dropdown menu showing 'Admin (Zabbix Administrator)' with a 'Select' button; 'Frontend access' is a dropdown menu set to 'Internal'; 'Enabled' is a checked checkbox; and 'Debug mode' is an unchecked checkbox. At the bottom of the form are three buttons: 'Update' (blue), 'Delete' (light blue), and 'Cancel' (light blue).

Problem solving: Managing LDAP/AD Authentication

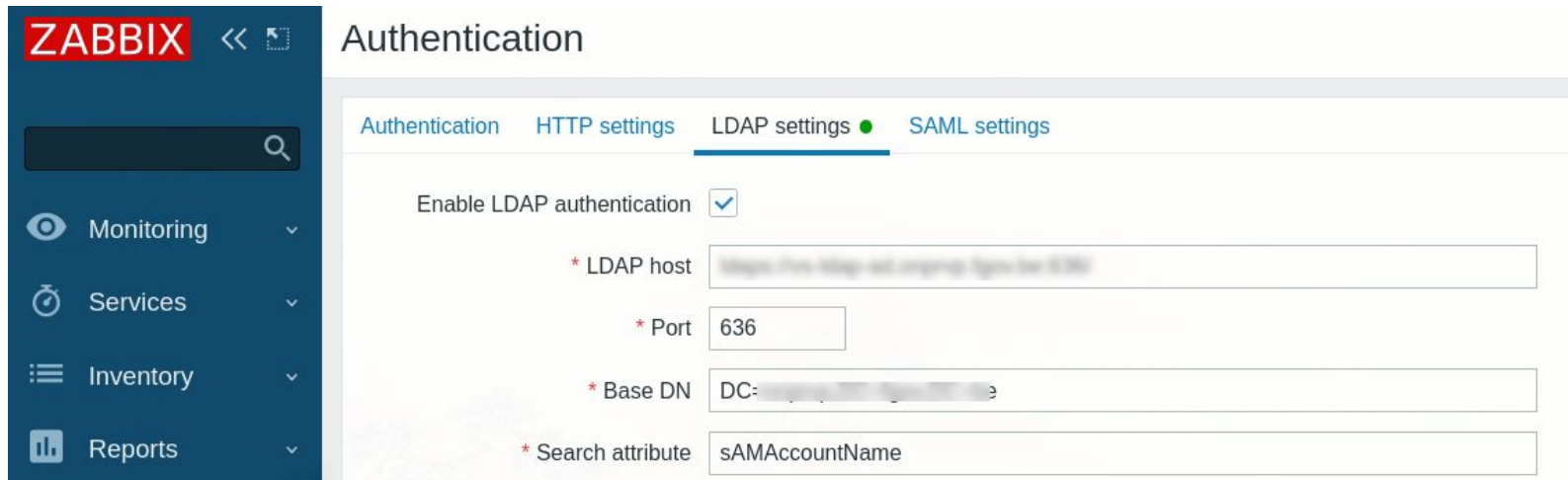
- No puppet resource for Authconfig (Before)

```
21
20 $ldap_settings_query = [
19   'UPDATE public.config',
18   'SET authentication_type = 1,',
17   "ldap_host='${profile::globals::ldap::uri}',",
16   "ldap_port=${profile::globals::ldap::port},",
15   "ldap_base_dn='${profile::globals::ldap::base_dn}',",
14   "ldap_bind_dn='CN=SRV_LDAP_Zabbix,${profile::globals::ldap::service_account_dn}',",
13   "ldap_bind_password='${ldap_password.unwrap}',",
12   "ldap_search_attribute='${profile::globals::ldap::search_attribute}',",
11   'ldap_configured=1,ldap_case_sensitive=1',
10   'WHERE configid = 1',
9   ]
8 $ldap_check_query = [
7   'SELECT 1 FROM public.config',
6   'WHERE authentication_type = 1',
5   "AND ldap_host='${profile::globals::ldap::uri}'",
4   "AND ldap_port=${profile::globals::ldap::port}",
3   "AND ldap_base_dn='${profile::globals::ldap::base_dn}'",
2   "AND ldap_bind_dn='CN=SRV_LDAP_Zabbix,${profile::globals::ldap::service_account_dn}',",
1   "AND ldap_bind_password='${ldap_password.unwrap}',",
288  "AND ldap_search_attribute='${profile::globals::ldap::search_attribute}',",
1   'AND ldap_configured=1 AND ldap_case_sensitive=1',
2   ]
3 postgresql_psql { 'Enable LDAP Auth':
4   connect_settings => $connect_settings,
5   db               => $_servername,
6   command          => Sensitive($ldap_settings_query.join(' ')),
7   unless           => Sensitive($ldap_check_query.join(' ')),
```



Problem solving: Managing LDAP/AD Authentication

- Zabbix authcfg provider (After)



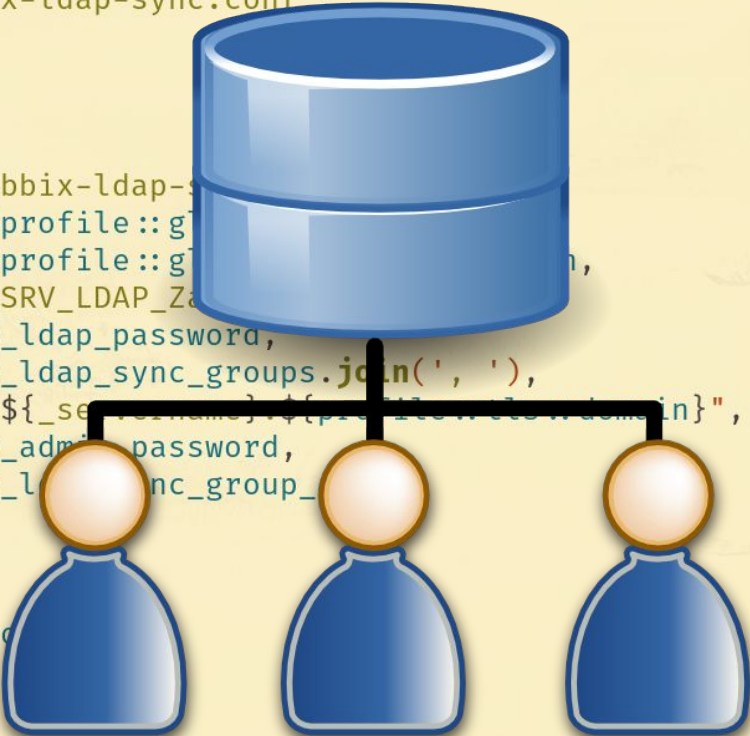
The screenshot shows the Zabbix web interface. On the left is a dark blue sidebar with the ZABBIX logo and navigation links: Monitoring, Services, Inventory, and Reports. The main content area is titled 'Authentication' and has four tabs: Authentication, HTTP settings, LDAP settings (which is active and highlighted with a green dot), and SAML settings. Under the 'LDAP settings' tab, there are several configuration options: 'Enable LDAP authentication' is checked; 'LDAP host' is a text input field; 'Port' is a text input field with the value '636'; 'Base DN' is a text input field with a value starting with 'DC='; and 'Search attribute' is a text input field with the value 'sAMAccountName'.

```
1  # Enable AD auth
2  zabbix_authcfg { '1':
3      ensure
4      authentication_type  => present,
5      ldap_host            => 'LDAP',
6      ldap_port            => $profile::globals::ldap::uri,
7      ldap_base_dn        => $profile::globals::ldap::port,
8      ldap_bind_dn        => $profile::globals::ldap::base_dn,
9      ldap_bind_password  => "CN=SRV_LDAP_Zabbix,${profile::globals::ldap::service_account_dn}",
10     ldap_bind_password  => $_ldap_password,
11     ldap_search_attribute => $profile::globals::ldap::search_attr,
12     ldap_case_sensitive => true,
13 }
14
```

Problem solving: Managing LDAP/AD Authentication

- Bonus: zabbix-ldap-sync
 - <https://github.com/zabbix-tooling/zabbix-ldap-sync>

```
1  file { '/etc/zabbix-ldap-sync/zabbix-ldap-sync.conf':
2
3      ensure    => file,
4      owner     => 'root',
5      group     => 'root',
6      mode      => '0600',
7      content   => epp('profile/zabbix/zabbix-ldap-s
8          'ldap_uri'           => $profile::g
9          'ldap_basedn'        => $profile::g
10         'ldap_binduser'      => 'SRV_LDAP_Za
11         'ldap_bindpassword'  => $_ldap_password,
12         'synced_groups'       => $_ldap_sync_groups.join(', '),
13         'zabbix_host'         => "${_se
14         'zabbix_admin_password' => $_admin_password,
15         'zabbix_ldap_sync_group' => $_l
16     }},
17     require => [
18         Package[$_ldap_sync_pkg],
19         Zabbix_usergroup[$_ldap_sync_gro
20     ],
21 }
22
```



The diagram shows a central blue database cylinder at the top, connected by a horizontal line to three blue server icons below it. This represents a central LDAP/AD server and multiple client servers.

Problem solving: Managing Roles

- Again, no provider existed
 - Before: Too long to show :-)
 - After:

```
3
4  zabbix_role { 'Production role':
5      ensure => present,
6      type   => 'Admin',
7      rules  => $_production_role_rules,
8  }
9
```

Thank you! Questions?

- bert@open-future.be
- <https://github.com/voxpupuli/puppet-zabbix>
 - Contributions are welcome ;-)
- <https://www.zabbix.com/documentation/current/en/manual/api>
- <https://rubygems.org/gems/zabbixapi/>