05/03/2019 • Ubuntu Bionic Beaver

Резервное копирование Mikrotik через Oxidized

Posted by ekzorchik

Прочитано: • 2 207

Итак сегодня речь пойдет об инструменте в задачу которого входит выполнение централизованного резервного копирования конфигурационных файлов (те которые выполняются с расширением `rsc`) оборудования `Mikrotik`. Плюс инструмент умеет отображать разницу в изменениях, а делает он это через складирование изменений посредством системы контроля версий `GIT`. Я если честно давно искал такой инструмент и вот я воплотил свою задумку разобрав от и до все шаги по разворачиванию сервиса `oxidized` в систему `Ubuntu 18.04 Server amd64`. Узнал я о нем это просматривая опубликованные презентации посвященные `Mikrotik`, к примеру если через поисковую систему `Google` ввести «Автоматизируем бэкап — MUM — MikroTik»

Предварительные действия: На всех **Mikrotik**(ах) для которых Вы хотите осуществлять резервное копирование нужно сделать учетную записи с минимальными правами, к примеру только на чтение:

```
ekzorchik@srv-bionic:~$ ssh -l admin 172.33.33.22

[admin@MikroTik] > user add name=test password=Aa1234567 group=read

[admin@MikroTik] > quit
```

```
Connection to 172.33.33.22 closed.
```

ekzorchik@srv-bionic:~$

Шаг №1: Настраиваю систему

```
ekzorchik@srv-bionic:~$ hg clone --insecure https://ekzorchik@172.35.35.8/hg/dproject

ekzorchik@srv-bionic:~$ sudo ./dproject/default
```

Полученная текущая система с учетом моего скрипта настройки который я применяю в своей локальной сети:

ekzorchik@srv-bionic:~$ uname -a && lsb_release -a

```
Linux srv-bionic 4.15.0-43-generic #46-Ubuntu SMP Thu Dec 6 14:45:28 UTC 2018
x86_64 x86_64 x86_64 GNU/Linux

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 18.04.2 LTS

Release: 18.04

Codename: bionic
```

ekzorchik@srv-bionic:~

Шаг №2: Устанавливаю пакет **Oxidized** в систему **Ubuntu 18.04 Server amd64:**

```
ekzorchik@srv-bionic:~$ sudo apt-get install ruby ruby-dev libsqlite3-dev libssl-dev pk

ekzorchik@srv-bionic:~$ gem install oxidized oxidized-script oxidized-web
```

```
Fetching: asetus-0.3.0.gem (100%)

ERROR: While executing gem ... (Gem::FilePermissionError)

You don't have write permissions for the /var/lib/gems/2.5.0 directory.
```

```
ekzorchik@srv-bionic:~$ sudo gem install oxidized oxidized-script oxidized-web
```

```
Done installing documentation for charlock_holmes, rack, tilt, rack-
protection, sinatra, emk-sinatra-url-for, temple, haml, htmlentities, puma,
rb-fsevent, ffi, rb-inotify, sass-listen, sass, backports, rack-test,
multi_json, sinatra-contrib, oxidized-web after 675 seconds

22 gems installed
```

```
ekzorchik@srv-bionic:~$
```

Шаг №3: Создаю пользователя **oxidized**, с правами которого будет запускаться система бэкапа:

```
ekzorchik@srv-bionic:~$ sudo useradd -m -p 712mbddr@ -s /bin/bash oxidized
```

На время даю права повышения привилегий, в конце заметки заберу.

```
ekzorchik@srv-bionic:~$ sudo usermod -aG sudo oxidized
```

Шаг №4: Создаю структуру запуска приложения **oxidized**

```
ekzorchik@srv-bionic:~$ sudo su - oxidized
```

```
oxidized@srv-bionic:~$ oxidized -v
```

```
0.26.1
```

```
oxidized@srv-bionic:~$
```

```
oxidized@srv-bionic:~$ whereis oxidized

oxidized: /usr/local/bin/oxidized

oxidized@srv-bionic:~$ oxidized

edit ~/.config/oxidized/config

oxidized@srv-bionic:~$ mkdir ~/.config/oxidized/crashes

oxidized@srv-bionic:~$ mkdir ~/.config/oxidized/mikrotik/

oxidized@srv-bionic:~$ sudo find / -name oxidized

/home/oxidized

/home/oxidized/.config/oxidized

/run/sudo/ts/oxidized

/usr/local/bin/oxidized

/var/lib/gems/2.5.0/gems/oxidized-web-0.13.1/lib/oxidized

/var/lib/gems/2.5.0/gems/oxidized-0.26.1/lib/oxidized

/var/lib/gems/2.5.0/gems/oxidized-0.26.1/bin/oxidized

/var/lib/gems/2.5.0/gems/oxidized-script-0.6.0/lib/oxidized

/var/lib/gems/2.5.0/doc/oxidized-web-0.13.1/ri/lib/oxidized

oxidized@srv-bionic:~$
```

Шаг №5: Преднастраиваем `GIT` для устройств `Mikrotik` бекапы которых будут хранится в репозитарии:

```
oxidized@srv-bionic:~$ sudo apt-get install git -y
```

```
oxidized@srv-bionic:~$ git config --global user.email "oxidized@ekzorchik.ru"

oxidized@srv-bionic:~$ git config --global user.name "oxidized"

oxidized@srv-bionic:~/.config/oxidized$ git init oxidized.git

Initialized empty Git repository in
/home/oxidized/.config/oxidized/oxidized.git/.git/

oxidized@srv-bionic:~/.config/oxidized$ git init devices.git

Initialized empty Git repository in
/home/oxidized/.config/oxidized/devices.git/.git/
```

Шаг №6: Настраиваем конфигурационный файл

```
1 oxidized@srv-bionic:~$ nano ~/.config/oxidized/config
```

```
1  ---
2
3  model: routeros
4
5  resolve_dns: true
6
7  #Как часто делать бекапы
8
9  interval: 3600
10
11 use_syslog: false
12
13 debug: false
14
15 threads: 30
16
17 timeout: 20
18
19 retries: 3
20
21 prompt: !ruby/regexp /^([\w.@-]+[#>]\s?)$/
22
23 rest: 0.0.0.0:8888
24
25 next_adds_job: false
26
27 vars: {}
28
29 groups: {}
30
31 models: {}
32
33 pid: "/home/oxidized/.config/oxidized/pid"
34
35 log: "/home/oxidized/.config/oxidized/log"
36
37 crash:
```

```
38
39  directory: "/home/oxidized/.config/oxidized/crashes"
40
41  hostnames: false
42
43  stats:
44
45  history_size: 10
46
47  input:
48
49  #Тип подключения к устройству
50
51  default: ssh
52
53  #Расширенное логирование отключено
54
55  debug: false
56
57  ssh:
58
59  secure: false
60
61  map:
62
63  verify_host_key: never
64
65  utf8_encoded: true
66
67  #где хранится конфигурации GIT
68
69  output:
70
71  default: git
72
73  git:
74
75  user: oxidized
76
77  email: oxidized@ekzorchik.ru
78
79  repo: "/home/oxidized/.config/oxidized/oxidized.git"
80
81  git:
82
83  user: oxidized
84
85  email: oxidized@ekzorchik.ru
86
87  repo: "/home/oxidized/.config/oxidized/devices/devices.git"
88
89  #Откуда берется информация об устройства которые бекапятся
90
91  source:
92
93  default: csv
94
95  csv:
96
97  file: "/home/oxidized/.config/oxidized/mikrotik/router.db"
98
99  delimiter: !ruby/regexp /:/
100
101 #Структура строки в router.db: name:model:ip:port:username:password
102
103 map:
104
105 name: 0
106
```

```
model: 1

ip: 2

port: 3

username: 4

password: 5

model_map:

cisco: ios

juniper: junos

mikrotik: routeros
```

На заметку: Если не настроена авторизация на **Mikrotik** через публичный ключ, т. е. Вход на **Mikrotik** с одним лишь указанием логина, то в конфигурационный файл **oxidized** следует добавить параметр: **verify_host_key: never**

Шаг №7: Копируем публичный **ssh**—ключ на **Mikrotik:**

```
oxidized@srv-bionic:~$ ssh-keygen
```

```
Your identification has been saved in /home/oxidized/.ssh/id_rsa.

Your public key has been saved in /home/oxidized/.ssh/id_rsa.pub.
```

```
oxidized@srv-bionic:~$ cp ~/.ssh/id_rsa.pub idrsa

oxidized@srv-bionic:~$ ftp 172.33.33.22
```

```
Connected to 172.33.33.22.

220 MikroTik FTP server (MikroTik 6.42.5) ready

Name (172.33.33.22:ekzorchik): admin

ftp> put idrsa

ftp> quit
```

```
oxidized@srv-bionic:~$

oxidized@srv-bionic:~$ ssh admin@172.33.33.22

[admin@MikroTik] > user ssh-keys import public-key-file=idrsa user=test

[admin@MikroTik] > quit
```

```
Connection to 172.33.33.22 closed.
```

```
oxidized@srv-bionic:~$
```

oxidized@srv-bionic:~$ ssh test@172.33.33.22 — подключение проходит без каких либо запросов на ввод пароля.

Шаг №8: Теперь создаю файл подключения к оборудованию **Mikrotik:**

```
1 oxidized@srv-bionic:~$ nano ~/.config/oxidized/mikrotik/router.db
2
3 mikrotik:routeros:172.33.33.22:22:test:Aa1234567
```

Шаг №9: Проверяю запуск и соответственно на предмет каких-либо ошибок препятствующих запуску:

```
oxidized@srv-bionic:~$ oxidized ~/.config/oxidized/config
```

```
Puma starting in single mode...
```

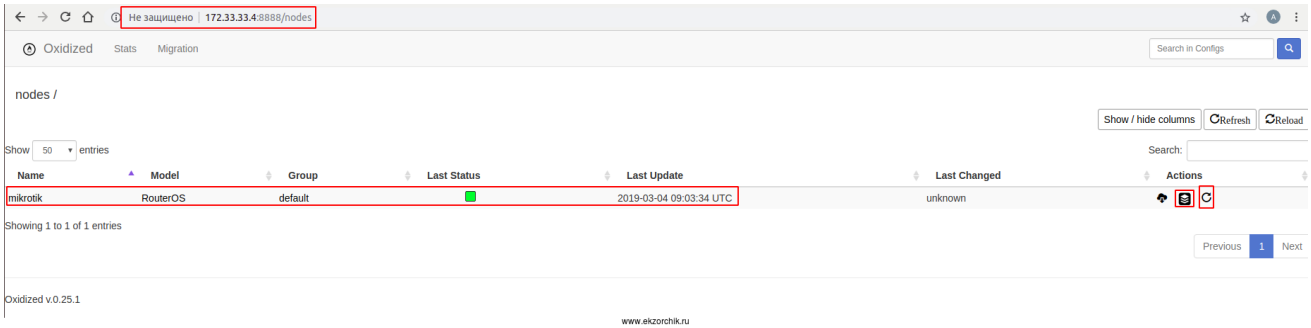
```
* Version 3.11.4 (ruby 2.5.1-p57), codename: Love Song
```

```
* Min threads: 0, max threads: 16
```

```
* Environment: development
```

```
* Listening on tcp://0.0.0.0:8888
```

```
Use Ctrl-C to stop
```



У каждой строчке есть колонка «**Actions**»

- Первая иконка: Посмотреть текущий конфигурационный файл

- Вторая иконка: Посмотреть историю изменений конфигурационного файла
- Третья иконка: Это запуск процедуры снятия изменения в репозитарий

Шаг №10: См.логи если что-то пошло не так

```
ekzorchik@srv-bionic:~$ tail -f /home/oxidized/.config/oxidized/log

W, [2019-03-04T11:34:49.491810 #28328] WARN -- : /mikrotik status
no_connection, retries exhausted, giving up

I, [2019-03-04T11:36:11.666786 #28380] INFO -- : Oxidized starting, running as
pid 28380

I, [2019-03-04T11:36:11.669229 #28380] INFO -- : lib/oxidized/nodes.rb:
Loading nodes

I, [2019-03-04T11:36:11.812262 #28380] INFO -- : lib/oxidized/nodes.rb: Loaded
1 nodes

I, [2019-03-04T11:36:14.351899 #28380] INFO -- : Configuration updated for
/mikrotik
```

Шаг №11: Для того, чтобы `oxidized` стартовал как служба, сделайте следующее:

```
oxidized@srv-bionic:~$ sudo find / -name oxidized.service

/var/lib/gems/2.5.0/gems/oxidized-0.26.1/extra/oxidized.service

oxidized@srv-bionic:~$ sudo cp /var/lib/gems/2.5.0/gems/oxidized-
0.26.1/extra/oxidized.service /lib/systemd/system/

oxidized@srv-bionic:~$ sudo nano /lib/systemd/system/oxidized.service
```

```
1  [Unit]
2
3  Description=Oxidized - Network Device Configuration Backup Tool
4
5  After=network-online.target multi-user.target
```

```
6
7  Wants=network-online.target
8
9  [Service]
10
11 ExecStart=/usr/local/bin/oxidized
12
13 User=oxidized
14
15 KillSignal=SIGKILL
16
17 Environment="OXIDIZED_HOME=/home/oxidized/.config/oxidized/"
18
19 [Install]
20
21 WantedBy=multi-user.target
```

oxidized@srv-bionic:~$ sudo systemctl enable oxidized.service

Created symlink /etc/systemd/system/multi-user.target.wants/oxidized.service → /lib/systemd/system/oxidized.service.

oxidized@srv-bionic:~$ sudo systemctl start oxidized

oxidized@srv-bionic:~$ sudo systemctl status oxidized | head -n5

● oxidized.service - Oxidized - Network Device Configuration Backup Tool

Loaded: loaded (/lib/systemd/system/oxidized.service; enabled; vendor preset: enabled)

Active: active (running) since Mon 2019-03-04 12:16:53 MSK; 9s ago

Main PID: 29039 (oxidized)

Tasks: 7 (limit: 2322)

Проверяю, что ожидаются соединения на порт **8888**

oxidized@srv-bionic:~$ sudo netstat -tulpn | grep :8888

```
1  tcp 0 0 127.0.0.1:8888 0.0.0.0:* LISTEN 29039/puma 3.11.4 (
```

oxidized@srv-bionic:~$

Шаг №12: По умолчанию у сервиса нет возможности указать авторизацию, а потому в конфиге `config` изменяю доступ на `172.0.0.1`

```
oxidized@srv-bionic:~$ nano ~/.config/oxidized/config
```

```
#rest: 0.0.0.0:8888

rest: 127.0.0.1:8888
```

```
oxidized@srv-bionic:~$ sudo apt-get install nginx -y
```

```
oxidized@srv-bionic:~$ sudo rm /var/www/html/index.nginx-debian.html
```

```
oxidized@srv-bionic:~$ sudo unlink /etc/nginx/sites-available/default
```

```
oxidized@srv-bionic:~$ sudo unlink /etc/nginx/sites-enabled/default
```

Создаю файл связки логина и пароля для аутентификации через `Web`—интерфейс:

```
oxidized@srv-bionic:~$ sudo sh -c "echo -n 'ekzorchik:' >>
/var/www/html/.htpasswd"
```

```
oxidized@srv-bionic:~$ openssl passwd -apr1
```

```
Password: 712mbddr@
```

```
Verifying - Password: 712mbddr@
```

```
$apr1$rwseclUD$ANBsXcpdvR.07mthC2PLO.
```

```
oxidized@srv-bionic:~$ sudo sh -c "openssl passwd -apr1 >>
/var/www/html/.htpasswd"
```

```
Password: 712mbddr@
```

```
Verifying - Password: 712mbddr@
```

```
oxidized@srv-bionic:~$ cat /var/www/html/.htpasswd
```

```
ekzorchik:$apr1$V6ZTHWcF$bdDbxqPJAQAzfdXw7Z9n/0
```

```
oxidized@srv-bionic:~$ sudo chown -R www-data:www-data /var/www/html/.htpasswd
```

```
oxidized@srv-bionic:~$ sudo nano /etc/nginx/sites-available/oxidized
```

```
server {

listen 172.33.33.4:80;

server_name _;

root /usr/share/nginx/html;

include /etc/nginx/default.d/*.conf;

location / {

auth_basic "UserName and Password Please";

auth_basic_user_file /var/www/html/.htpasswd;

proxy_pass http://127.0.0.1:8888/;

}

}
```

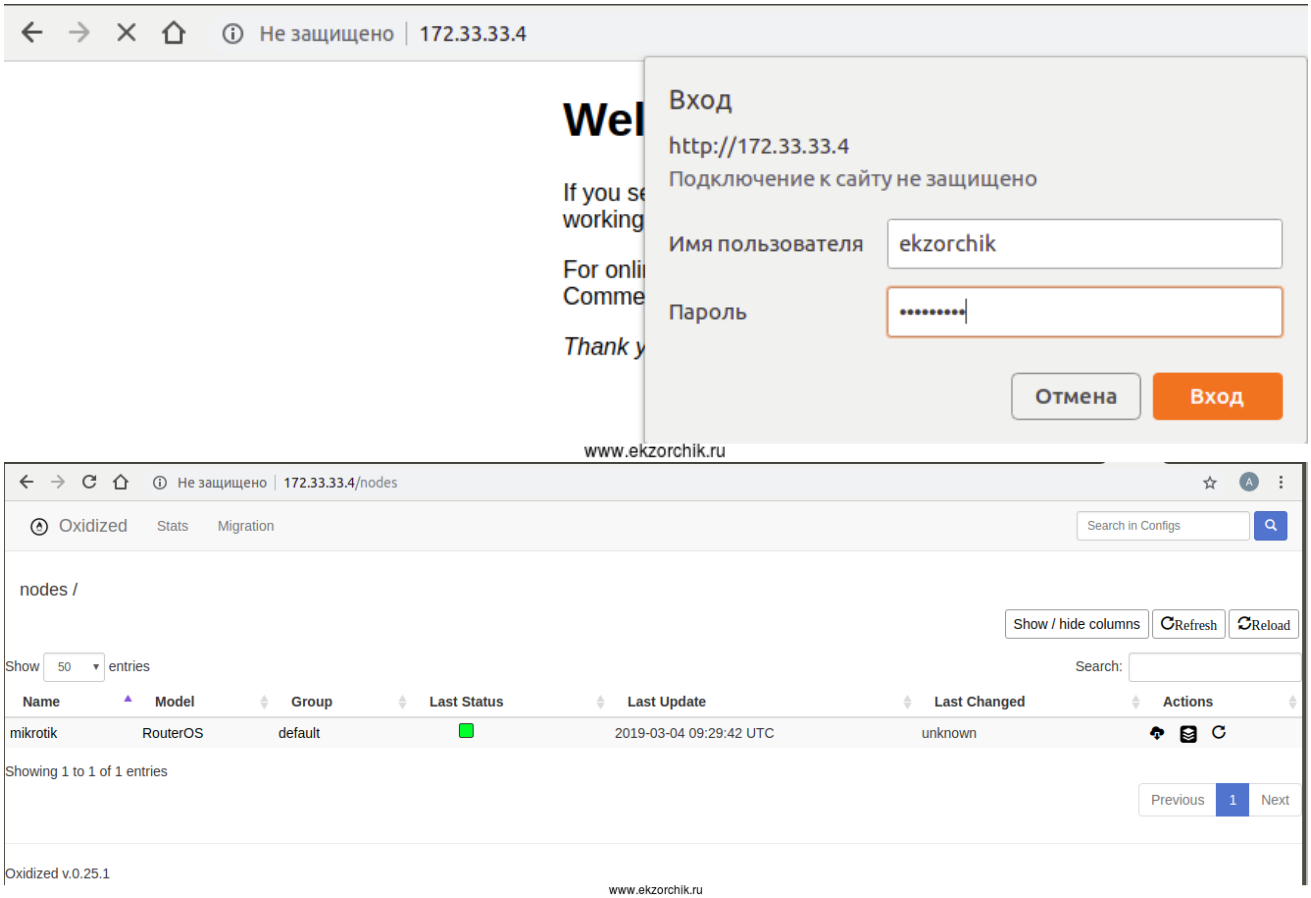
```
oxidized@srv-bionic:~$ sudo ln -s /etc/nginx/sites-available/oxidized /etc/nginx/sites-enabled/oxidized
```

```
oxidized@srv-bionic:~$ sudo nginx -t

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful
```

```
oxidized@srv-bionic:~$ sudo systemctl restart nginx && sudo systemctl status nginx | head -n5

● nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Mon 2019-03-04 12:19:27 MSK; 34ms ago

Docs: man:nginx(8)
```

```
Process: 29084 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5
--pidfile /run/nginx.pid (code=exited, status=0/SUCCESS)
```

```
oxidized@srv-bionic:~$
```



Работает! Итого я для себя разобрал, как использовать инструмент **oxidized** на **Ubuntu 18.04 Server amd64** для создания резервных копий с оборудования **Mikrotik** и просмотра внесенных изменений через **Web**-интерфейс с применением системы контроля версий на базе **GIT**. Жалко что нельзя использовать **Mercurial** — мне данная система контроля версий ближе, у меня все на ней построено: скрипты, конфигурационные файлы и т. д. А пока на этом у меня всё, с уважением автор блога **Олло Александр aka ekzorchik.**

BACKUP RSC MIKROTIK BACKUP RSC TO OXIDIZED OXIDIZED UBUNTU 18.04

« Изменение шаблона «История болезни» в МИС Инфоклиника

Обновленная установка Zabbix версии 4 на Ubuntu 18.04 Server »

2 059
liveinternet
1 354