

05/03/2019  
Ubuntu Bionic Beaver

## Резервное копирование Mikrotik через Oxidized

Posted by [ekzorchik](#)  
Прочтано: 4 700

Итак сегодня речь пойдет об инструменте в задачу которого входит **выполнение централизованного резервного копирования конфигурационных файлов** (те которые выполняются с расширением **rsync**) **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 шаг за шагом 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 запись с минимальными правами, к примеру только на чтение:

```
1 ekzorchik@srv-bionic:~$ ssh -l admin 172.33.33.22
2
3 [admin@Mikrotik] > user add none-test password=As1234567 group=read
4
5 [admin@Mikrotik] > quit
```

Connection to 172.33.33.22 closed.

### Шаг №1: Настраиваю систему

```
1 ekzorchik@srv-bionic:~$ hg clone --insecure https://ekzorchik@172.33.35.8/hg/dproject
2
3 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:

```
1 ekzorchik@srv-bionic:~$ sudo apt-get install ruby ruby-dev libsqlite3-dev libssl-dev pkg-config cmake libssl2-1-dev libicu-dev -y
2
3 ekzorchik@srv-bionic:~$ gem install oxidized oxidized-script oxidized-web
```

Fetching: astatus-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, hamlentities, puma, rb-fsevent, ffi, rb-inotify, sass-listen, sass, backports, rack-test, multi_json, sinatra-contrib, oxidized-web after 675 seconds

22 gems installed
```

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

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

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

```
ekzorchik@srv-bionic:~$ sudo usermod -oG 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/rdoc/lib/oxidized
```

### Шаг №5: Преднастраиваю GIT для устройств Mikrotik бэкапы которых будут храниться в репозитории:

```
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:~$ cd ~/.config/oxidized/ && git init oxidized.git
```

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

```
oxidized@srv-bionic:~$ cd ~/.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
2
3 ---
4
5 model: routers
6
7 resolve_dns: true
8
9 #Как часто забирать бэкапы
10 interval: 3600
11 use_syslog: false
12 debug: false
13 threads: 30
14 timeout: 20
15 retries: 3
16
17 prompt: !ruby/regexp /(?!(w.#-|e)#{1,7})$/
18
19 rest: 0.0.0.0:8888
20
21 next_addr_job: false
22
23 vars: {}
24
25 groups: {}
26
27 models: {}
28
29 pid: "/home/oxidized/.config/oxidized/pid"
30
31 log: "/home/oxidized/.config/oxidized/log"
32
33 crash:
34
35 directory: "/home/oxidized/.config/oxidized/crashes"
36
37 hostnames: false
38
39 stats:
40
41 history_size: 10
42
43 input:
44
45 #Путь подключения к устройству
46
47 default: ssh
48
49 #Расширенное логирование отключено
50
51 debug: false
52
53 ssh:
54
55 secure: false
56
57 map:
58
59 verify_host_key: never
60
61 etf8_encoded: true
62
63 #где хранятся конфигурации GIT
64
65 output:
66
67 default: git
68
69 git:
70
71 user: oxidized
72
73 email: oxidized@ekzorchik.ru
74
75 repo: "/home/oxidized/.config/oxidized/oxidized.git"
76
77 git:
78
79 user: oxidized
80
81 email: oxidized@ekzorchik.ru
82
83 repo: "/home/oxidized/.config/oxidized/devices/devices.git"
84
85 #Откуда берется информация об устройствах которые бэкапятся
86
87 source:
88
89 default: csv
90
91 csv:
92
93 file: "/home/oxidized/.config/oxidized/mikrotik/router.db"
94
95 delimiter: !ruby/regexp /-/
96
97 #Структура строки в router.db: name:model:ip:port:username:password
98
99 map:
100
101 name: 0
102
103 model: 1
104
105 ip: 2
106
107 port: 3
108
109 username: 4
110
111 password: 5
112
113 model_map:
114
115 elisco: tos
116
117 juniper: junos
118
119 mikrotik: routers
```

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:

```
1 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.

```
1 oxidized@srv-bionic:~$ cp ~/.ssh/id_rsa.pub idrsa
2
3 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
```

```
1 oxidized@srv-bionic:~$
2
3 oxidized@srv-bionic:~$ ssh admin@172.33.33.22
4
5 [admin@Mikrotik] > user ssh-keys import public-key-file=idrsa user=test
6
7 [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,routers,172.33.22,22:test:As1234567
```

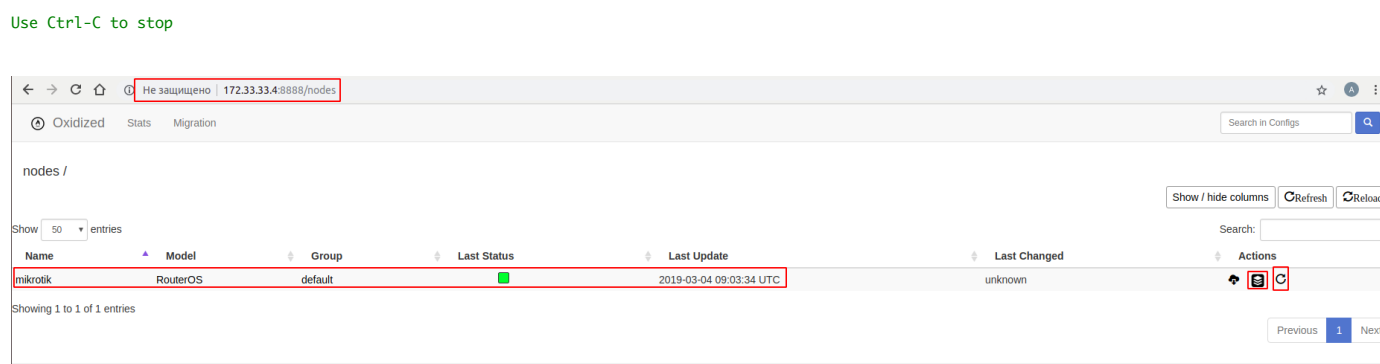
### Шаг №9: Проверяю запуск и соответственно на предмет каких-либо ошибок препятствующих запуску:

```
oxidized@srv-bionic:~$ oxidized -f ~/.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 строчки есть колонка «Actions»

Используйте прокси (заблокировано роскомнадзором, исп

Админский канал: <https://t.me/ekzorchik>

Мой канал: [https://t.me/aka\\_ekzorchik](https://t.me/aka_ekzorchik)

Поблагодари автора и новые статьи  
Будут появляться чаще :)

Карта МКБ: 4432-7300-2472-8059  
Yandex-деньги: 4100152005047

Большое спасибо тем кто благодарит автора за практиче

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

Шар №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
```

Шар №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
```

```
[Unit]
Description=Oxidized - Network Device Configuration Backup Tool
After=network-online.target multi-user.target
Wants=network-online.target
[Service]
ExecStart=/usr/local/bin/oxidized
User=oxidized
KillSignal=SIGKILL
Environment="OXIDIZED_HOME=/home/oxidized/.config/oxidized/"
[Install]
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 -tulnp | grep :8888
```

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

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

Шар №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
```

```
1 #rest: 0.0.0.0:8888
2 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:~$ sudo openssl passwd -apr1
```

```
Password: 712mbddr@
```

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

```
Sapr1$wseclUD$ANbXcpdvR.07mtHCZPLO.
```

```
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:Sapr1$wseclUD$ANbXcpdvR.07mtHCZPLO.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
```

```
1 server {
2     listen 172.33.33.4:80;
3
4     server_name _;
5
6     root /usr/share/nginx/html;
7
8     include /etc/nginx/default.d/*.conf;
9
10    location / {
11
12        auth_basic "Username and Password Please";
13        auth_basic_user_file /var/www/html/.htpasswd;
14
15        proxy_pass http://127.0.0.1:8888;
16    }
17 }
18 }
19 }
20 }
```

```
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
```

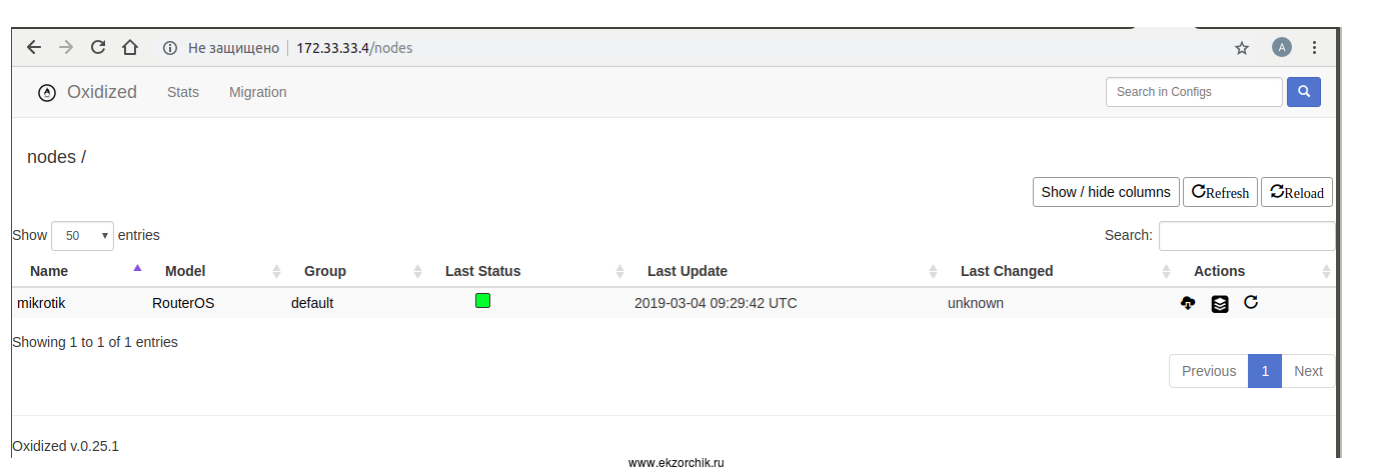
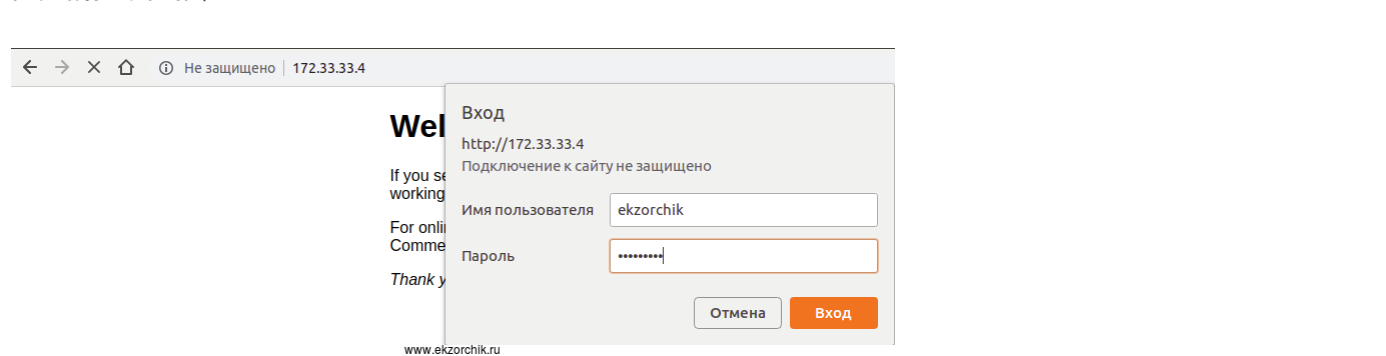
```
oxidized@srv-bionic:~$ sudo systemctl restart nginx && sudo systemctl status nginx | head -n5
```

```
● 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 версия на базе GIT. Жалко что нельзя использовать Mercurial — мне данная система контроля версий ближе, у меня все на ней построено: скритты, конфигурационные файлы и т. д. А пока на этом у меня всё, с уважением автор блога **Олло Александр aka ekzorchik**.