rcb Documentation

Release 0.1.0

Vladimir Botka

May 28, 2018

Contents:

CHAPTER 1

Introduction

RCB (Rsync-Crypto-Backup) is a set of scripts in Bash that:

- Create local backup with rsnapshot
- Encrypt local backup with rsyncrypto
- Sync encrypted backup to the remote server with rsync
- Restore the backup and
- Compare the restored data with the origin

Rsyncrypto ensures that doing rsync to synchronize the encrypted files to another machine will have only a small impact on rsync's efficiency. Whole RCB project is approximately 500 lines long.

Installation and configuration was tested with Ansible role https://galaxy.ansible.com/vbotka/rcb/ with FreeBSD and Ubuntu.

CHAPTER 2

Installation

2.1 Installation with Ansible

1.Install Ansible role

```
> ansible-galaxy install vbotka.rcb
```

2.Configure variables

Examples of playbooks and variables are available at RCB. Edit and change at least following variables.

- *rcb_BCK_HOST* and *rcb_BCK_DST* in *vars/rcb.yml*
- *rcb_BCK_DST* in *vars/rcb-backup-server.yml*
- *rcb_privatekey_passphrase* in *vars/rcb.yml*
- *rcb_cert_CN* in *vars/rcb.yml*
- *hosts* in *playbooks/rcb.yml*
- *hosts* in *playbooks/rcb-backup-server.yml*

3.Run Ansible playbooks

Following workflow was tested with Ubuntu(local Backup-Client) and FreeBSD (remote Backup-Server).

1. Create SSH keys at Backup-Clients and stores the public keys at the localhost

```
> ansible-playbook ~/.ansible/playbooks/rcb.yml -t phase1
```

2. Configure the ssh access of Backup-Clients to Backup-Server. Store the public keys of Backup-Clients, created in phase1, into the ~/rcb_BCK_USER/.ssh/authorized_keys.

```
> ansible-playbook ~/.ansible/playbooks/rcb-backup-server.yml
```

3. Configure the Backup-Clients.

```
> ansible-playbook ~/.ansible/playbooks/rcb.yml -t phase2
```

2.2 Test installation

Run tests and check /var/log/rcb.log for potential errors

```
> ansible-playbook ~/.ansible/playbooks/rcb.yml -t testall
```

CHAPTER 3

Configure rsnapshot

Add backup directories to rsnapshot.conf . Location of rsnapshot.conf is defined by RSNAPSHOT_PARAM in rcb.conf.

CHAPTER 4

Configure cron

1. Configuration with current Ansible role

Default value is *rcb_rsnapshot_cron=no* e.g. crontab is not configured by default. It is posible to configure *rcb_rsnapshot_cron=yes* in *vars/rcb.yml* and use Ansible to configure crontab.

```
rcb_rsnapshot_cron: "yes"
```

Default values are in the block below.

```
rcb_rsnapshot_cron_user: "root"
rcb_rsnapshot_cron_path: "{{ rcb_bin_dir }}:/usr/local/sbin:/usr/local/bin:/usr/sbin:/
↪usr/bin:/sbin:/bin"
rcb_rsnapshot_cron_mailto: "root"
rcb_rsnapshot_cron_hourly_hour: "*/4"
rcb_rsnapshot_cron_hourly_minute: "15"
rcb_rsnapshot_cron_hourly_command: "rcb-rsnapshot.sh -i=hourly"
rcb_rsnapshot_cron_daily_hour: "5"
rcb_rsnapshot_cron_daily_minute: "15"
rcb_rsnapshot_cron_daily_command: "rcb-daily-encrypt-rsync-decrypt-restore.sh"
rcb_rsnapshot_cron_weekly_hour: "1"
rcb_rsnapshot_cron_weekly_minute: "15"
rcb_rsnapshot_cron_weekly_day: "1"
rcb_rsnapshot_cron_weekly_command: "rcb-rsnapshot.sh -i=weekly"
rcb_rsnapshot_cron_monthly_hour: "2"
rcb_rsnapshot_cron_monthly_minute: "15"
rcb_rsnapshot_cron_monthly_day: "1"
rcb_rsnapshot_cron_monthly_command: "rcb-rsnapshot.sh -i=monthly"
```

2. Configuration with Ansible role linux-postinstall

Systemic way is to keep *rcb_rsnapshot_cron=no* and configure all crontab entries of the system with Ansible role linux-postinstall, if the system is Linux. To use this role, install it

```
> ansible-galaxy install vbotka.linux-postinstall
```

and configure the variables *lp_cron_var* and *lp_cron_tab*. Then configure the *linux-postinstall* playbook and run it.

```
> ansible-playbook playbooks/linux-postinstall.yml -t lp_cron
```

3. Manual configuration of cron

For manual configuration of cron RCB project provides crontab example .

CHAPTER 5

Scripts

5.1 rcb-rsnapshot.sh

Run rsnapshot.

```
> /root/bin/rcb-rsnapshot.sh
rcb-rsnapshot.sh [-h|--help] [-i|--increment=[hourly,daily,weekly,monthly]] --␣
↪rsnapshot backup
where:
    -h --help      show this help text
    -i --increment period [hourly,daily,weekly or monthly]
```

5.2 rcb-encrypt.sh

1. Delete files in directory $RCB_META. If directory $RCB_META doesn't exist, create it.
2. Create directory $RCB_META/$REMOTE if it doesn't exist. REMOTE are the directories defined in BACKUP POINTS (3rd parameter) of rsnapshot.conf
3. Store lists of empty directories, links, special files and sockets in the directory $RCB_META/$REMOTE. Create digests in $RCB_META/$REMOTE/$RCB_DIGESTS.
4. Encrypt $RCB_BCK_ROOT/$RCB_BCK_PREFIX to $RCB_ENC.

5.3 rcb-rsync.sh

Rsync encrypted data from SRC to DST, defined as follows

```
SRC="$RCB_ENC"
DST="$BCK_USER@$BCK_HOST:$BCK_DST"
```

5.4 rcb-rsync-back.sh

Restore encrypted data from the remote backup (SRC) to local (DST). SRC and DST are defined as follows.

```
SRC="$BCK_USER@$BCK_HOST:$BCK_DST"
DST="$RCB_ENCR"
```

Running the script with -l will symlink $RCB_ENC to $RCB_ENCR.

```
USAGE="$(basename "$0") [-h|--help] [-l|--link] [-d|--delete] -- rsync from backup
where:
  -h --help     show this help text
  -l --link     links the origin directory instead of rsync
  -d --delete   delete the destination defore rsync"
```

5.5 rcb-decrypt.sh

Dencrypt data from SRC to DST, defined as follows

```
SRC="$RCB_ENCR"
DST="$RCB_DEC"
```

Running the script with -d will delete data in DST before decryption.

```
USAGE="$(basename "$0") [-h|--help] [-d|--delete] -- decrypt backup
where:
  -h --help   show this help text
  -d --delete decrypted data, if exist"
```

5.6 rcb-restore.sh

Restore data to $RCB_RST_ROOT .

CHAPTER 6

Development

To facilitate the deployment and testing the rcb project provides the playbook rcb-devel to copy the current version of the scripts to the staging. By default the current version is locked (source_lock_set: "yes"). This means that the rcb role won't accidentally overwrite it.

The following sequence of commands copy, patch and install the scripts.

```
> ansible-playbook ~/.ansible/playbooks/rcb-devel.yml
> ansible-playbook ~/.ansible/playbooks/rcb.yml -t rcb_patch
> ansible-playbook ~/.ansible/playbooks/rcb.yml -t rcb_copy
```

CHAPTER 7

Testing

The scripts can be tested.

```
> ansible-playbook ~/.ansible/playbooks/rcb.yml -t testall
```

CHAPTER 8

Indices and tables

- genindex
- modindex
- search