Tài liệu backup volume

1. Backup volume CEPH bằng phần mềm Backy2

## Backup hàng ngày

* Yêu cầu:
  + Backup hàng ngày
  + Giữ lại 3 bản gần nhất
* Thư mục: **/root/script-backup-all/daily** - Chứa script và các file log trong quá trình backup Volume.
* File script Backup: **/root/script-backup-all/daily/backup-daily-all.sh**
* Thư mục log **/root/script-backup-all/daily/log**
  + **[Date].txt:** danh sách volume được backup theo ngày.
  + **backup\_[Date].log**: File dùng để lưu trữ log trong quá trình backup Volume theo ngày
  + **clean\_[Date].txt**: Danh sách Volume được xóa sau 28 ngày.
  + **clean\_backup\_[Date].log**: File dùng để lưu trữ log trong quá trình xóa các file backup đã quá 28 ngày.
* Crontab: **0 23 \* \* \* root bash /root/script-backup-all/daily/backup-daily-all.sh**
  + Thực hiện backup hằng ngày vào lúc 23:00
* Nội dung Script:

| # Get date  backup\_day=$(date +'%Y%m%d')  clean\_day=$(date --date="3 day ago" +'%Y%m%d')  apiToken='6073261344:AAEWZ83zFaVYEXW\_vAQB0jLiNyl2pmV-E5k'  chatId='-4101230458'  # Backup folder  DIR='/root/script-backup-all/daily/log/'  mkdir -p $DIR  start=$(date +%s.%N)  # Remove old backup  backy2 ls | grep day\_$clean\_day > $DIR/clean\_$clean\_day.txt  echo "Begin clean backup $clean\_day" 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  for BACKUP\_ID in $(cat $DIR/clean\_$clean\_day.txt | awk '{print $12}')  do  backy2 rm -f $BACKUP\_ID 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  done  #Get list all Volume of Ceph  rbd ls volumes > $DIR/$backup\_day.txt  # Backup  echo "Begin backup daily $backup\_day" > $DIR/backup\_$backup\_day.log  for VOLUME\_ID in $(cat $DIR/$backup\_day.txt)  do  if [ $VOLUME\_ID != 'ID' ] ; then  backy2 backup -t day\_$backup\_day rbd://volumes/$VOLUME\_ID $VOLUME\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  sleep 5s  #echo "Day la id volume:" $VOLUME\_ID  fi  done  echo "DONE backup $backup\_day" 2>&1 | tee -a $DIR/backup\_$backup\_day.log  date 2>&1 | tee -a $DIR/backup\_$backup\_day.log  #time\_all  end=$(date +%s.%N)  runtime=$(python3 -c "print(${end} - ${start})")  runtime\_M=$(echo "$runtime/60" | bc)  # Telegram notify  numberbackup=$(backy2 ls | grep day\_$backup\_day | wc -l)  numbervolume=$(cat $DIR/$backup\_day.txt | wc -l)  numberbackupvalid=$(backy2 ls | grep day\_$backup\_day | awk '{sum+=$14} END {print sum}')  numberbackupfailed=$(expr $numberbackup - $numberbackupvalid)  freedisk=$(df -h | grep "/backy2" | awk '{print $4 " / " $2}')  message="""\`[Backup weekly $backup\_day][Done in $runtime\_M min]  Count volume: $numbervolume  Count Weekly\_backup $backup\_day: $numberbackup  Count Weekly\_backup Error: $numberbackupfailed  Free Part Backup: $freedisk\`"""  #echo $message  curl -s \  -X POST \  https://api.telegram.org/bot$apiToken/sendMessage \  -d text="$message" \  -d chat\_id=$chatId \  -d parse\_mode=markdown |
| --- |

* Quy trình thực hiện:

1. Thực hiện script vào 23 giờ hằng ngày.
2. Kiểm tra và xóa các bản Backup đã quá 3 ngày.
3. Thực hiện list lại tất cả các Volume hiện tại và Backup toàn bộ volume.
4. Sau khi xóa backup cũ và Backup toàn bộ các volume gửi cảnh báo về telegram thông báo đã hoàn tất backup.

## Backup hàng tuần

* Yêu cầu:
  + Backup hàng tuần
  + Giữ lại 4 bản gần nhất
* Thư mục: **/root/script-backup-all/weekly** - Chứa script và các file log trong quá trình backup Volume.
* File script Backup: **/root/script-backup-all/weekly/backup-weekly.sh**
* Thư mục log **/root/script-backup-all/weekly/log**
  + **[Date].txt:** danh sách volume được backup theo tuần.
  + **backup\_[Date].log**: File dùng để lưu trữ log trong quá trình backup Volume theo tuần
  + **clean\_[Date].txt**: Danh sách Volume được xóa sau 4 tuần.
  + **clean\_backup\_[Date].log**: File dùng để lưu trữ log trong quá trình xóa các file backup đã quá 4 tuần.
* Crontab: **0 12 \* \* 0 root bash /root/script-backup-all/weekly/backup-weekly.sh**
  + Thực hiện backup hàng tuần vào chủ nhật vào lúc 12:00
* Nội dung Script:

| # Get date  backup\_day=$(date +'%Y%m%d')  clean\_day=$(date --date="21 day ago" +'%Y%m%d')  apiToken='6073261344:AAEWZ83zFaVYEXW\_vAQB0jLiNyl2pmV-E5k'  chatId='-4101230458'  # Backup folder  DIR='/root/script-backup-all/weekly/log/'  mkdir -p $DIR  start=$(date +%s.%N)  # # # Remove old backup  backy2 ls | grep weekly\_$clean\_day > $DIR/clean\_$clean\_day.txt  echo "Begin clean backup week $clean\_day" 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  for BACKUP\_ID in $(cat $DIR/clean\_$clean\_day.txt | awk '{print $12}')  do  backy2 rm -f $BACKUP\_ID 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  done  #Get list all Volume of Ceph -- OK  rbd ls volumes > $DIR/$backup\_day.txt  # # Backup --ok  echo "Begin backup week $backup\_day" > $DIR/backup\_$backup\_day.log  for VOLUME\_ID in $(cat $DIR/$backup\_day.txt)  do  if [ $VOLUME\_ID != 'ID' ] ; then  backy2 backup -t weekly\_$backup\_day rbd://volumes/$VOLUME\_ID $VOLUME\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  sleep 5s  #echo "Day la id volume:" $VOLUME\_ID  fi  done  echo "DONE backup weekly: $backup\_day" 2>&1 | tee -a $DIR/backup\_$backup\_day.log  date 2>&1 | tee -a $DIR/backup\_$backup\_day.log  # time\_all  end=$(date +%s.%N)  runtime=$(python3 -c "print(${end} - ${start})")  runtime\_M=$(echo "$runtime/60" | bc)  # Telegram notify  numberbackup=$(backy2 ls | grep weekly\_$backup\_day | wc -l)  numbervolume=$(cat $DIR/$backup\_day.txt | wc -l)  numberbackupvalid=$(backy2 ls | grep weekly\_$backup\_day | awk '{sum+=$14} END {print sum}')  numberbackupfailed=$(expr $numberbackup - $numberbackupvalid)  freedisk=$(df -h | grep "/backup" | awk '{print $4 " / " $2}')  message="""\`[Backup weekly $backup\_day][Done in $runtime\_M min]  Count volume: $numbervolume  Count Weekly\_backup $backup\_day: $numberbackup  Count Weekly\_backup Error: $numberbackupfailed  Free Part Backup: $freedisk\`"""  # #echo $message  curl -s \  -X POST \  https://api.telegram.org/bot$apiToken/sendMessage \  -d text="$message" \  -d chat\_id=$chatId \  -d parse\_mode=markdown |
| --- |

* Quy trình thực hiện:

1. Thực hiện script vào 12 giờ sáng chủ nhật hàng tuần.
2. Kiểm tra và xóa các bản Backup đã quá 4 tuần.
3. Thực hiện list lại tất cả các Volume hiện tại và Backup toàn bộ volume.
4. Sau khi xóa backup cũ và Backup toàn bộ các volume gửi cảnh báo về telegram thông báo đã hoàn tất backup.

## Cơ chế thực hiện restore máy ảo cho Volume ceph.

Xác định ID Instance và ID Volume:





Sau khi lấy được Volume ID thì vào Backup-Server.

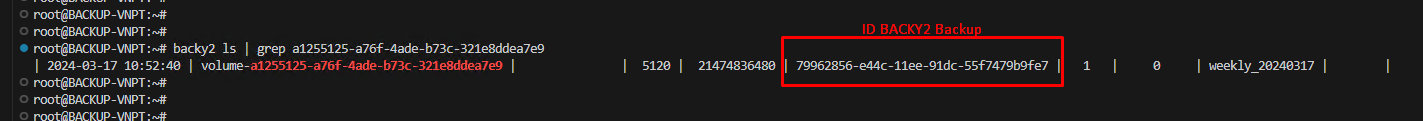
Volume ID:

| ce8ee18f-4318-4440-aacc-d4f91cc00ebd |
| --- |

Kiểm tra bản backup trên Backy2

| backy2 ls | grep volume-[Volume\_ID] |
| --- |

Kiểm tra thông tin dưới ceph về thông tin của volume, Xác định ID Backup:



Thực hiện Restore bằng lệnh sau:

| **backy2 restore ID\_Backy2\_Backup rbd://volumes/volume-[Volume\_ID]** |
| --- |

* **ID\_Backy2\_Backup:** Được lấy ở bước trên, xác định ID Backup
* **rbd://volumes/volume-[Volume\_ID]:** Đường dẫn của Volume.
  + **rbd://volumes/volume-5959ed71-a4b8-4b0c-8651-915fba865764**

2 Backup Local bằng cơ chế Snapshot Instance

# Backup hằng ngày

Yêu cầu:

* 1. Backup hàng ngày
  2. Giữ lại 3 bản gần nhất
* Thư mục: **/root/script-backup-all/daily** - Chứa script và các file log trong quá trình backup Volume.
* File script Backup: **/root/backup-local/daily/daily.sh**
* Thư mục log **/root/backup-local/daily/log**
  + **[Date].txt:** danh sách volume được backup theo ngày.
  + **backup\_[Date].log**: File dùng để lưu trữ log trong quá trình backup Volume theo ngày
  + **clean\_[Date].txt**: Danh sách Volume được xóa sau 28 ngày.
  + **clean\_backup\_[Date].log**: File dùng để lưu trữ log trong quá trình xóa các file backup đã quá 3 ngày.
* Crontab: **22 23 \* \* \* root bash /root/backup-local/daily/daily.sh**
  + Thực hiện backup hằng ngày vào lúc 23:22
* Các file Backup được lưu trữ vào thư mục: **/backup/Data\_Backup\_local/daily/** Trên Server backup. 10.10.100.253
* Nội dung Script:

| backup\_day=$(date +'%Y%m%d')  clean\_day=$(date --date=" day ago" +'%Y%m%d')  apiToken='6073261344:AAEWZ83zFaVYEXW\_vAQB0jLiNyl2pmV-E5k'  chatId='-4101230458'  source /root/backup-local/daily/admin-sources.sh  start=$(date +%s.%N)  # Backup folder  DIR='/root/backup-local/daily/log/'  mkdir -p $DIR  DIR\_BK='/backup/Data\_Backup\_local/daily/'  mkdir -p $DIR\_BK  # Remove file outdate old backup  ls /backup/Data\_Backup\_local/daily | grep $clean\_day > $DIR/clean\_$clean\_day.txt  echo "Begin clean backup $clean\_day" 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  for BACKUP\_ID in $(cat $DIR/clean\_$clean\_day.txt)  do  echo Remove backup ID: $BACKUP\_ID 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  rm /backup/Data\_Backup\_local/daily/$BACKUP\_ID  echo Done Remove backup ID: $BACKUP\_ID 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  echo "" 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  done  #Backup Disk Local  openstack server list --all-projects --host COMPU-VNPT-LOCAL-101 | grep '|' | grep '-' | awk '{print $2}' > $DIR/test\_$backup\_day.txt  for INSTANCE\_ID in $(cat $DIR/test\_$backup\_day.txt)  do  check\_attach=$(openstack server show $INSTANCE\_ID | grep volumes\_attached | awk '{print $5}')  echo $check\_attach  if [ "$check\_attach" = '' ] ; then  echo $INSTANCE\_ID 2>&1 | tee -a $DIR/ID\_$backup\_day.txt  fi  done  rm $DIR/test\_$backup\_day.txt  ## Snapshot  for INSTANCE\_ID in $(cat $DIR/ID\_$backup\_day.txt)  do  echo Backup Instance: $INSTANCE\_ID and Name\_Backup: $backup\_day\\_$INSTANCE\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  openstack server image create --name $backup\_day\\_$INSTANCE\_ID --wait $INSTANCE\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  sleep 5  name\_sn=$DIR\_BK$backup\_day\\_$INSTANCE\_ID.qcow2  echo $name\_sn  echo Save Image $backup\_day\\_$INSTANCE\_ID to $name\_sn 2>&1 | tee -a $DIR/backup\_$backup\_day.log  openstack image save $backup\_day\\_$INSTANCE\_ID --file $name\_sn  echo Done Save Image $backup\_day\\_$INSTANCE\_ID to $name\_sn 2>&1 | tee -a $DIR/backup\_$backup\_day.log  sleep 5  echo Delete image $backup\_day\\_$INSTANCE\_ID in CEPH 2>&1 | tee -a $DIR/backup\_$backup\_day.log  openstack image delete $backup\_day\\_$INSTANCE\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  echo Done delete image $backup\_day\\_$INSTANCE\_ID in CEPH 2>&1 | tee -a $DIR/backup\_$backup\_day.log  echo -e "----------------------------------\n\n" 2>&1 | tee -a $DIR/backup\_$backup\_day.log  done  echo Done Backup Instance Date: $backup\_day 2>&1 | tee -a $DIR/backup\_$backup\_day.log  echo ============Finish============== 2>&1 | tee -a $DIR/backup\_$backup\_day.log  #time\_all  end=$(date +%s.%N)  runtime=$(python3 -c "print(${end} - ${start})")  runtime\_M=$(echo "$runtime/60" | bc)  # Telegram notify  numberbackup=$(cat $DIR/ID\_$backup\_day.txt | wc -l)  numberbackupvalid=$(ls -1 $DIR\_BK | grep $backup\_day | wc -l )  freedisk=$(df -h | grep "/backup" | awk '{print $4 " / " $2}')  #Message  message="""\`[Backup daily for Local $backup\_day][Done in $runtime\_M min]  Count volume Local: $numberbackup  Count Daily\_backup $backup\_day: $numberbackupvalid  Free Part Backup: $freedisk\`"""  #echo $message  curl -s \  -X POST \  https://api.telegram.org/bot$apiToken/sendMessage \  -d text="$message" \  -d chat\_id=$chatId \  -d parse\_mode=markdown |
| --- |

# Backup hằng tuần

Yêu cầu:

* 1. Backup hàng tuần
  2. Giữ lại 4 bản gần nhất
* Thư mục: **/root/backup-local/weekly** - Chứa script và các file log trong quá trình backup Volume.
* File script Backup: **/root/backup-local/weekly/weekly.sh**
* Thư mục log **/root/backup-local/weekly/log**
  + **[Date].txt:** danh sách volume được backup theo ngày.
  + **backup\_[Date].log**: File dùng để lưu trữ log trong quá trình backup Volume theo ngày
  + **clean\_[Date].txt**: Danh sách Volume được xóa sau 28 ngày.
  + **clean\_backup\_[Date].log**: File dùng để lưu trữ log trong quá trình xóa các file backup đã quá 3 ngày.
* Crontab: **20 23 \* \* \* root bash /root/backup-local/daily/daily.sh**
  + Thực hiện backup hằng ngày vào lúc 23:20
* Các file Backup được lưu trữ vào thư mục: **/backup/Data\_Backup\_local/weekly/** Trên Server backup. 10.10.100.253
* Nội dung Script:

| backup\_day=$(date +'%Y%m%d')  clean\_day=$(date --date="28 day ago" +'%Y%m%d')  apiToken='6073261344:AAEWZ83zFaVYEXW\_vAQB0jLiNyl2pmV-E5k'  chatId='-4101230458'  source /root/backup-local/weekly/admin-sources.sh  start=$(date +%s.%N)  # Backup folder  DIR='/root/backup-local/weekly/log/'  mkdir -p $DIR  DIR\_BK='/backup/Data\_Backup\_local/weekly/'  mkdir -p $DIR\_BK  # Remove file outdate old backup  ls /backup/Data\_Backup\_local/weekly | grep $clean\_day > $DIR/clean\_$clean\_day.txt  echo "Begin clean backup $clean\_day" 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  for BACKUP\_ID in $(cat $DIR/clean\_$clean\_day.txt)  do  echo Remove backup ID: $BACKUP\_ID 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  rm /backup/Data\_Backup\_local/weekly/$BACKUP\_ID  echo Done Remove backup ID: $BACKUP\_ID 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  echo "" 2>&1 | tee -a $DIR/clean\_backup\_$clean\_day.log  done  #Backup Disk Local  openstack server list --all-projects --host COMPU-VNPT-LOCAL-101 | grep '|' | grep '-' | awk '{print $2}' > $DIR/test\_$backup\_day.txt  for INSTANCE\_ID in $(cat $DIR/test\_$backup\_day.txt)  do  check\_attach=$(openstack server show $INSTANCE\_ID | grep volumes\_attached | awk '{print $5}')  echo $check\_attach  if [ "$check\_attach" = '' ] ; then  echo $INSTANCE\_ID 2>&1 | tee -a $DIR/ID\_$backup\_day.txt  fi  done  rm $DIR/test\_$backup\_day.txt  ## Snapshot  for INSTANCE\_ID in $(cat $DIR/ID\_$backup\_day.txt)  do  echo Backup Instance: $INSTANCE\_ID and Name\_Backup: $backup\_day\\_$INSTANCE\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  openstack server image create --name $backup\_day\\_$INSTANCE\_ID --wait $INSTANCE\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  sleep 5  name\_sn=$DIR\_BK$backup\_day\\_$INSTANCE\_ID.qcow2  echo $name\_sn  echo Save Image $backup\_day\\_$INSTANCE\_ID to $name\_sn 2>&1 | tee -a $DIR/backup\_$backup\_day.log  openstack image save $backup\_day\\_$INSTANCE\_ID --file $name\_sn  echo Done Save Image $backup\_day\\_$INSTANCE\_ID to $name\_sn 2>&1 | tee -a $DIR/backup\_$backup\_day.log  sleep 5  echo Delete image $backup\_day\\_$INSTANCE\_ID in CEPH 2>&1 | tee -a $DIR/backup\_$backup\_day.log  openstack image delete $backup\_day\\_$INSTANCE\_ID 2>&1 | tee -a $DIR/backup\_$backup\_day.log  echo Done delete image $backup\_day\\_$INSTANCE\_ID in CEPH 2>&1 | tee -a $DIR/backup\_$backup\_day.log  echo -e "----------------------------------\n\n" 2>&1 | tee -a $DIR/backup\_$backup\_day.log  done  echo Done Backup Instance weekly: $backup\_day 2>&1 | tee -a $DIR/backup\_$backup\_day.log  echo ============Finish============== 2>&1 | tee -a $DIR/backup\_$backup\_day.log  #time\_all  end=$(date +%s.%N)  runtime=$(python3 -c "print(${end} - ${start})")  runtime\_M=$(echo "$runtime/60" | bc)  # Telegram notify  numberbackup=$(cat $DIR/ID\_$backup\_day.txt | wc -l)  numberbackupvalid=$(ls -1 $DIR\_BK | grep $backup\_day | wc -l )  freedisk=$(df -h | grep "/backup" | awk '{print $4 " / " $2}')  #Message  message="""\`[Backup weekly for Local $backup\_day][Done in $runtime\_M min]  Count volume Local: $numberbackup  Count Weekly\_backup $backup\_day: $numberbackupvalid  Free Part Backup: $freedisk\`"""  #echo $message  curl -s \  -X POST \  https://api.telegram.org/bot$apiToken/sendMessage \  -d text="$message" \  -d chat\_id=$chatId \  -d parse\_mode=markdown |
| --- |

# Thực hiện restore máy ảo boot Local

Tìm kiếm trong thư mục **/backup/Data\_Backup\_local**

* **weekly**
* **daily**

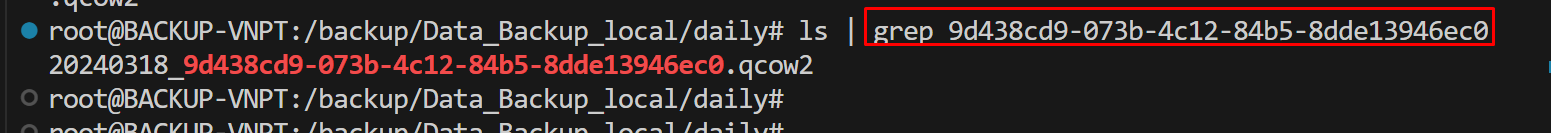
Xác định máy ảo local:



Lưu ý máy ảo local là máy ảo không được boot từ Ceph



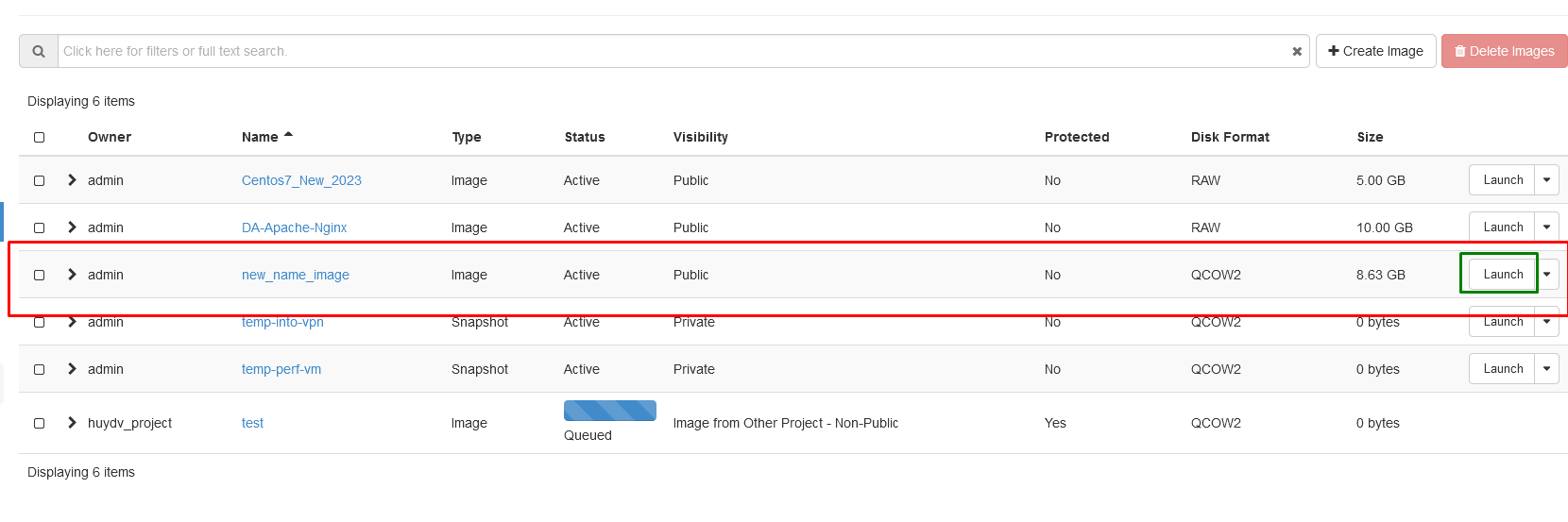
Tìm name file backup.



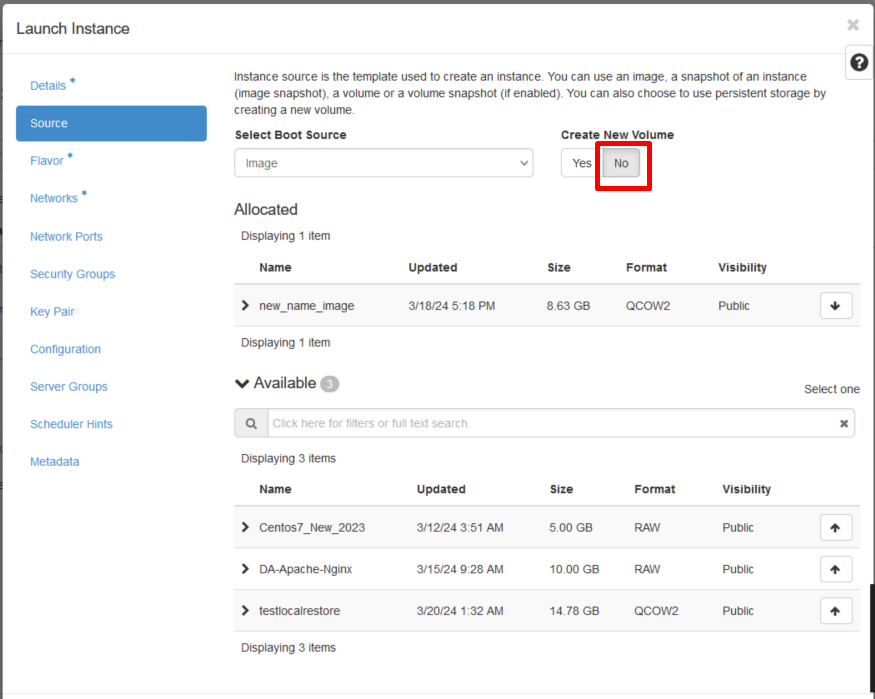
Thực hiện đẩy file backup về Openstack:

| **source /root/backup-local/daily/admin-sources.sh**  **openstack image create --disk-format qcow2 --container-format bare --public --file [path\_file.qcow2] --property hw\_qemu\_guest\_agent='yes' [new\_name\_image]** |
| --- |

Kiểm tra lại kết quả trên horizon



Đây là image vừa được đẩy lên, thực hiện tạo máy ảo mới.



Tạo máy ảo lưu ý chọn: No - Create New Volume

Thực hiện xóa Image sau khi đã tạo được máy ảo

AE lưu ý: Nếu muốn tạo máy ảo theo cách tạo volume ceph từ file backup local này thì sẽ có 2 lựa chọn:

* Convert file backup hiện tại đang từ QCOW2 → RAW
* Nâng cái timeout của cinder lên

# Thực hiện Backup instance local chỉ

Chỉ áp dụng cho 1 Instance