

VirtualBox 관리 Script

VirtualBox 관리 Shell script

- [Virtual Machine List](#)
- [Virtual Machine Power Off](#)
- [Virtual Machine Status 상태 확인](#)
- [Virtual Machine START headless](#)
- [Virtual Machine PAUSE / RESUMING](#)
- [Virtual Machine Snapshot Delete](#)
- [Snapshot All VM](#)

Virtual Machine List

Virtual Machine List

```
#!/bin/bash

echo -e "\n"
echo "#####"
echo "##### VM List START ! #####"
echo "#####"

echo -e "\n## Virtual Images list ##"
allVm=$(/usr/bin/VBoxManage list vms | awk '{print($1)}' | sort)

# Check Status
VmStatus(){
    vmStatus=$(/usr/bin/VBoxManage showvminfo $1 | grep -e ^State | sed s/\ \ //g )
    echo ${vmStatus}
}

# All VM Print
PrintVmList(){
    for i in $allVm
    do
        vmname=$(echo $i | sed -e "s/\"//g")
        echo " > " $vmname " , Status=" $(VmStatus $vmname)
    done
}

cd /home/hyunsu/BackUpCenter/20221014-exportVM/
PrintVmList

echo -e "\n"
echo "#####"
echo "##### VM List END ! #####"
echo "#####"
echo -e "\n"
```

사용 & 결과

```
[hyunsu@joang BackUpCenter]$ ./000-VMList.sh
```

```
#####
```

```
##### VM List START ! #####
#####

## Virtual Images list ##
> 0-RockyNAS , Status= State: running (since 2024-10-09T16:01:50.678000000)
> 1-RockyKubeMaster , Status= State: running (since 2024-10-09T22:39:41.958000000)
> 10-Deloitte001 , Status= State: running (since 2024-10-09T16:02:01.092000000)
> 4-RockyWorker1 , Status= State: running (since 2024-10-10T04:19:36.495000000)
> 9-GitRepo , Status= State: running (since 2024-10-09T16:02:16.701000000)
> Test-RockyDesktop , Status= State: powered off (since
2023-12-23T15:01:41.000000000)
> ex-RockyWorker2 , Status= State: running (since 2024-10-10T04:21:54.371000000)
> ex-RockyWorker3 , Status= State: running (since 2024-10-10T04:23:14.112000000)

#####
##### VM List END ! #####
#####
```

Virtual Machine Power Off

Virtual Machine power off

```
#!/bin/sh

export _SDATE=$(date +"%Y%m%d%H")

echo "==== Power Off VM ====="
echo ">> VMNAME : $1"
echo ">> DATE : $_SDATE"

# VM Pweroff
VmPause()
{
    vm=$1
    /usr/bin/VBoxManage controlvm $vm poweroff
    echo ">> " $vm " Poweroffed "
}

echo ">> VM Name = " $1
VmPause $1

echo "==== VM Power Off END ! ====="
```

실행 & 결과

```
[hyunsu@joang BackUpCenter]$ ./002-poweroffVm.sh
==== Power Off VM =====
>> VMNAME : 1-RockyKubeMaster
>> DATE : 2024101021
>> VM Name = 1-RockyKubeMaster
>> 1-RockyKubeMaster Poweroffed
==== VM Power Off END ! =====
```

Virtual Machine Status 상태 확인

Virtual Machine Status 상태 확인

```
#!/bin/bash

export _SDATE=$(date +"%Y%m%d%H")

echo "===== Show info VM ====="
echo ">> VMNAME : $1"
echo ">> DATE : $_SDATE"

# VM Pweroff
VmPause()
{
    vm=$1
    /usr/bin/VBoxManage showvminfo $vm
    echo "Running ? "
    /usr/bin/VBoxManage showvminfo $vm | grep State:
}

echo "##### VM Name = " $1 "#####"
VmPause $1

echo "===== Show info END ! ====="
```

실행 % 결과

```
[hyunsu@joang BackUpCenter]$ ./003-showvminfoVm.sh 1-RockyKubeMaster
===== Show info VM =====
>> VMNAME : 1-RockyKubeMaster
>> DATE : 2024101022
##### VM Name = 1-RockyKubeMaster #####
Name:                               1-RockyKubeMaster
Encryption:                          disabled
Groups:                               /Kubernetes-Hyunsu
Platform Architecture:               x86
Guest OS:                             Red Hat (64-bit)
UUID:                                 d416ace7-184b-4c22-bc26-a9afb3808039
Config file:                          /home/hyunsu/VirtualBox VMs/Kubernetes-Hyunsu/1-
RockyKubeMaster/1-RockyKubeMaster.vbox
Snapshot folder:                     /data/ext/4TB/backup/Snapshots/1-RockyKubeMaster
Log folder:                           /home/hyunsu/VirtualBox VMs/Kubernetes-Hyunsu/1-
```

RockyKubeMaster/Logs

Hardware UUID: d416ace7-184b-4c22-bc26-a9afb3808039
Memory size: 30048MB
Page Fusion: disabled
VRAM size: 16MB
CPU exec cap: 100%
CPUProfile: host
Chipset: piix3
Firmware: BIOS
Number of CPUs: 40
HPET: disabled
PAE: enabled
Long Mode: enabled
Triple Fault Reset: disabled
APIC: enabled
X2APIC: enabled
Nested VT-x/AMD-V: disabled
CPUID overrides: None
Hardware Virtualization: enabled
Nested Paging: enabled
Large Pages: enabled
VT-x VPID: enabled
VT-x Unrestricted Exec.: enabled
AMD-V Virt. Vmsave/Vmload: enabled
CPUID Portability Level: 0
Boot menu mode: message and menu
Boot Device 1: HardDisk
Boot Device 2: DVD
Boot Device 3: Not Assigned
Boot Device 4: Not Assigned
ACPI: enabled
IOAPIC: enabled
BIOS APIC mode: APIC
Time offset: 0ms
BIOS NVRAM File: /home/hyunsu/VirtualBox VMs/Kubernetes-Hyunsu/1-RockyKubeMaster/1-RockyKubeMaster.nvram
RTC: UTC
IOMMU: None
Paravirt. Provider: Default
Effective Paravirt. Prov.: KVM
State: running (since 2024-10-09T22:39:41.958000000)
Graphics Controller: VMSVGA
Monitor count: 1
3D Acceleration: disabled
Teleporter Enabled: disabled
Teleporter Port: 0
Teleporter Address:
Teleporter Password:
Tracing Enabled: disabled
Allow Tracing to Access VM: disabled

Tracing Configuration:

Autostart Enabled: disabled
Autostart Delay: 0
Default Frontend:
VM process priority: default
Storage Controllers:
#0: 'IDE', Type: PIIX4, Instance: 0, Ports: 2 (max 2), Bootable
Port 1, Unit 0: Empty
#1: 'SATA', Type: IntelAhci, Instance: 0, Ports: 1 (max 30), Bootable
Port 0, Unit 0: UUID: 0c33d809-198c-426b-bf17-b8556a91d066
Location: "/home/hyunsu/VirtualBox VMs/Kubernetes-Hyunsu/1-RockyKubeMaster/1-RockyKubeMaster-2024100500-disk001.vdi"
NIC 1: MAC: 08002773BA2F, Attachment: Bridged Interface
'enp3s0', Cable connected: on, Trace: off (file: none), Type: 82540EM, Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy: allow-all, Bandwidth group: none
NIC 2: disabled
NIC 3: disabled
NIC 4: disabled
NIC 5: disabled
NIC 6: disabled
NIC 7: disabled
NIC 8: disabled
Pointing Device: PS/2 Mouse
Keyboard Device: PS/2 Keyboard
UART 1: disabled
UART 2: disabled
UART 3: disabled
UART 4: disabled
LPT 1: disabled
LPT 2: disabled
Audio: enabled (Driver: Default, Controller: AC97, Codec: AD1980)
Audio playback: enabled
Audio capture: disabled
Clipboard Mode: disabled
Clipboard file transfers: disabled
Drag and drop Mode: disabled
Session name: headless
Video mode: 720x400x0 at 0,0 enabled
VRDE: disabled
OHCI USB: enabled
EHCI USB: enabled
xHCI USB: disabled
USB Device Filters: <none>
Available remote USB devices: <none>
Currently attached USB devices: <none>
Bandwidth groups: <none>
Shared folders: <none>
VRDE Connection: not active
Clients so far: 0

```
Recording status:          stopped
Recording enabled:        no
Recording screens:        1
  Screen 0:
    Enabled:              yes
    ID:                   0
    Record video:         yes
    Destination:          File
    File:                  /home/hyunsu/VirtualBox VMS/Kubernetes-Hyunsu/1-
RockyKubeMaster/1-RockyKubeMaster-screen0.webm
    Options:              vc_enabled=true,ac_enabled=false,ac_profile=med
    Video dimensions:     1024x768
    Video rate:           512kbps
    Video FPS:            25fps
* Guest:
Configured memory balloon: 0MB
OS type:                  Linux26_64
Additions run level:      2
Additions version:        7.0.10 r158379
Guest Facilities:
Facility "VirtualBox Base Driver": active/running (last update: 2024/10/09 22:39:52
UTC)
Facility "VirtualBox System Service": active/running (last update: 2024/10/09 22:40:00
UTC)
Facility "Seamless Mode": not active (last update: 2024/10/09 22:39:52 UTC)
Facility "Graphics Mode": not active (last update: 2024/10/09 22:39:52 UTC)

Running ?
State:                    running (since 2024-10-09T22:39:41.958000000)
===== Show info  END    ! =====
```

Virtual Machine START headless

Virtual Machine START headless

```
#!/bin/sh

echo "===== VM START START ! ====="
echo ">> VMNAME : $1"

# Start VM
StartVM()
{
    vm=$1
    /usr/bin/VBoxManage startvm $vm --type headless
    echo " > " $vm " Started VM"
    sleep 10
}

StartVM $1

echo "===== VM START END ! ====="
```

실행 & 결과

```
[hyunsu@joang BackUpCenter]$ ./004-StartVm.sh Test-RockyDesktop
===== VM START START ! =====
>> VMNAME : Test-RockyDesktop
Waiting for VM "Test-RockyDesktop" to power on...
 > Test-RockyDesktop Started VM
===== VM START END ! =====
[hyunsu@joang BackUpCenter]$
```

Virtual Machine PAUSE / RESUMING

Virtual Machine PAUSE / RESUMING

PAUSE

```
#!/bin/sh

export VMNAME=$1

echo "===== SNAPSHOT PAUSE START ! ====="
echo ">> VMNAME : $1"
echo ">> Pause $1 VM for snapshot "
VBoxManage controlvm $1 pause
echo "===== SNAPSHOT PAUSE END ! ====="
```

RESUMING

```
#!/bin/sh

export _SDATE=$(date +"%Y%m%d%H")
export VMNAME=$1

echo "===== SNAPSHOT Resuming START ! ====="
echo ">> VMNAME : $1"
echo ">> " $1 " Resuming VM"
VBoxManage controlvm $1 resume
echo ">> " $1 " Resume VM"
echo "===== SNAPSHOT Resuming END ! ====="
```

Virtual Machine Snapshot Delete

vm snapshot을 삭제한다.

전체 스냅샷 중에서 FinalSnapshotDueDay이전의 스냅샷을 지우되 FinalSnapshotCnt 이하로 지우지는 않는다.

```
#!/bin/bash

export _SDATE=$(date +"%Y%m%d%H")
export FinalSnapshotCnt=10
export FinalSnapshotDueDay=60
export VMNAME=$1
export NOW=`date +%Y%m%d%H`
export FinalSnapshotDay=`date +%Y%m%d%H --date="$FinalSnapshotDueDay days ago"`

echo "===== SNAPSHOT BACKUP&DELETE START ! ====="
echo ">> VMNAME : $1"
echo ">> DATE : $_SDATE"
echo ">> FinalBackupDay : $FinalSnapshotDueDay"
echo ">> FinalSnapshotDay : $FinalSnapshotDay"
echo ">>> ## Old $VMNAME snapshot delete ! ##"
TotalsnapshotCnt=`VBoxManage snapshot $VMNAME list | grep Name | awk '{print ($2)}' | wc -l`
echo ' >> Total snapshot count = ' $TotalsnapshotCnt
echo ' >> Final snapshot remain count = ' $FinalSnapshotCnt
TargetdeleteSnapshotCnt=`expr $TotalsnapshotCnt - $FinalSnapshotCnt`
echo ' >> Total snapshot count - Final snapshot count =' $TargetdeleteSnapshotCnt

BackupAllFiles=$(VBoxManage snapshot $VMNAME list | grep Name | awk '{print ($2)}' | sort -n)
cnt=1
echo ' > ' $FinalSnapshotDay ' day before delete target snapshot list'
for i in $BackupAllFiles
do
    snapshotDate=`echo $i | awk -F "-" '{print $1}'`
    echo $cnt " : " $i
    if [ $snapshotDate -le $FinalSnapshotDay ]
    then
        if [ $TargetdeleteSnapshotCnt -ge $cnt ]
        then
```

```
snapshotDate=`echo $i | awk -F "-" '{print $1}'`
#echo $snapshotDate " vs " $FinalSnapshotDay
if [ $snapshotDate -le $FinalSnapshotDay ]
then
    echo "Delete snapshot $i"
    VBoxManage snapshot $1 delete $i
fi
else
    echo " >> skip snapshot : " $i
fi
fi
cnt=$(expr $cnt + 1)
done

echo "===== SNAPSHOT BACKUP&DELETE END ! ====="
```

Snapshot All VM

모든 VM에 대하여 snapshot을 생성합니다.

1. 전체 VM 리스트를 출력
 - 이름만 분리하고 오름차순으로 정렬한다.
 - "l" 제거하고 VM명과 상태를 출력한다.
2. VM 전체를 Pause 한다
 - 전체 vm 리스트에 대한 pause를 실행
 - vm명에 "ex"가 있는 경우 skip하고자 하였으나 운영 상 스냅샷을 하는 것이 좋을 것 같아 나중에 수정함
 - 단 vm명에 "ex"가 있는 경우
 - 실행 상태 확인
 - 실행상태
 - 이미 pause 상태를 확인하고 pause인 경우 skip , pause가 아닌 경우 pause [pause 실행](#)
 - 실행 상태가 아닌 경우
 - skip
 - vm명에 "ex"가 없는 경우
 - 실행 상태 확인
 - 실행상태
 - 이미 pause 상태를 확인하고 pause인 경우 skip , pause가 아닌 경우 pause [Pause 실행](#)
 - 실행 상태가 아닌 경우
 - skip
 - 3. 스냅샷을 생성한다
 - 스냅샷도 Pause와 공일한데 단 "ex"는 시행하지 않는다.
 - 4. 오래된 스냅샷을 삭제한다. [Delete](#)
 - "ex", "Test"인 경우 skip
 - pause된 vm에 한정하여 삭제한다.
: 전체 스냅샷 중에서 FinalSnapshotDueDay이전의 스냅샷을 지우되 FinalSnapshotCnt 이하로 지우지는 않는다.
 - 5. Pause를 풀고 Resuming한다.
 - "ex"인 경우 : pause인 경우 재개한다. pause가 아닌 경우 실행하고 있는 경우는 skip 그렇지 않은 경우 재개
 - "Test"인 경우 : 모두 skip
 - 그 외에는 running 경우를 제외하고 모두 재개한다.

```
#!/bin/bash

echo -e "\n"
echo "#####"
echo "##### MAKE SNAPSHOT START ! #####"
echo "#####"
export _SDATE=$(date +"%Y-%m-%d %H")
echo "START" "$_SDATE"

echo -e "\n## Virtual Images list ##"
allVm=$(/usr/bin/VBoxManage list vms | awk '{print($1)}' | sort)
```

```

runningVm=$(/usr/bin/VBoxManage list runningvms | awk '{print($1)}' | sort -nr)

# Check Status
VmStatus(){
    vmStatus=$(/usr/bin/VBoxManage showvminfo $1 | grep -e ^State | sed s/\ \ //g )
    echo ${vmStatus}
}

# All VM Print
PrintVmList(){
    for i in $allVm
    do
        vmname=$(echo $i | sed -e "s/\"//g")
        echo " > " $vmname " , Status=" $(VmStatus $vmname)
    done
}

# Check Running
IfRunning(){
    runnable=1
    vmStatus=$(/usr/bin/VBoxManage showvminfo $1 | grep -e ^Name: -e ^State | sed s/\
 \ //g | grep running | awk '{print($2)}')
    if [ -z $vmStatus ] ; then
        runnable=1
    elif [ $vmStatus = "running" ] ; then
        runnable=0
    else
        runnable=1
    fi
    echo ${runnable}
}

# Check Pause
IfPause(){
    pausevm=1
    vmStatus=$(/usr/bin/VBoxManage showvminfo $1 | grep -e ^Name: -e ^State | sed s/\
 \ //g | grep paused | awk '{print($2)}')
    if [ -z $vmStatus ] ; then
        pausevm=1
    elif [ $vmStatus = "paused" ] ; then
        pausevm=0
    else
        pausevm=1
    fi
    echo ${pausevm}
}

# All VM PAUSE
PauseVM(){
    echo -e "\n## All Running Virtual Images Pause ##"
}

```

```

for i in $allVm
do
    vmname=$(echo $i | sed -e "s/\\/\\/g")
ex=$(echo $vmname | awk -F'-' '{print $1}')
    if [ "$ex" = "ex" ] ; then
        if [ $(IfRunning $vmname) = 0 ] ; then
            if [ $(IfPause $vmname) = 0 ] ; then
                echo " > " $vmname " is running but Already paused, Ex Skip
PAUSE START ! "
            else
                echo " > " $vmname " is running and not paused, Ex PAUSE
START ! "
                ./002-snapshotVMPause.sh $vmname
            fi
        else
            echo " > " $vmname " is not running, Ex PAUSE Skip Pause !"
        fi
    else
        if [ $(IfRunning $vmname) = 0 ] ; then
            if [ $(IfPause $vmname) = 0 ] ; then
                echo " > " $vmname " Already paused, notEx Skip PAUSE START !
"
            else
                echo " > " $vmname " is running and not paused, notEx PAUSE
START ! "
                ./002-snapshotVMPause.sh $vmname
            fi
        else
            echo " > " $vmname " is not running, notEx Skip Pause !"
        fi
    fi
done
}

# VM Snapshot
Snapshot(){
    echo -e "\n## Virtual Images Snapshot ##"
    for i in $allVm
    do
        vmname=$(echo $i | sed -e "s/\\/\\/g")
        ex=$(echo $vmname | awk -F'-' '{print $1}')
        if [ "$ex" = "ex" ] ; then
            if [ $(IfPause $vmname) = 0 ] ; then
                echo " > " $vmname " is pause, Ex Snapshot Skip ! "
            else
                echo " > " $vmname " is not pause, Ex Snapshot Skip ! "
            fi
        else
            if [ $(IfPause $vmname) = 0 ] ; then
                if [ "$ex" = "Test" ] ; then

```

```

        echo " > " $vmname " is pause, notEx but it is Test VM so
Snapshot Skip !"
    else
        echo " > " $vmname " is pause, notEx Snapshot START !"
        ./003-snapshotVM.sh $vmname
    fi
else
    echo " > " $vmname " is not pause, notEx Snapshot Skip ! "
fi
fi
done
}

# VM Delete
Delete(){
    echo "\n## Virtural Images Backup Snapshot Delete ##"
    for i in $allVm
    do
        vmname=$(echo $i | sed -e "s/\\/\\/g")
        ex=$(echo $vmname | awk -F'-' '{print $1}')
        if [ "$ex" = "ex" ] ; then
            echo " > " $vmname " Skip Snapshot Delete !"
        else
            if [ "$ex" = "Test" ] ; then
                echo " > " $vmname " is disable VM so Skip Snapshot Delete !"
            else
                if [ $(IfPause $vmname) = 0 ] ; then
                    echo " > " $vmname " is pause, Snapshot Delete START ! "
                    ./004-snapshottiDeleteVM.sh $vmname
                else
                    echo " > " $vmname " is not pause, notEx Skip Snapshot
Delete !"
                fi
            fi
        fi
    done
}

# VM Resuming
Resuming(){
    echo "\n## Virtural Images Snapshot ##"
    for i in $allVm
    do
        vmname=$(echo $i | sed -e "s/\\/\\/g")
        ex=$(echo $vmname | awk -F'-' '{print $1}')
        if [ "$ex" = "ex" ] ; then
            if [ $(IfPause $vmname) = 0 ] ; then
                echo " > " $vmname " is pause, Ex START ! "
                ./005-snapshotResuming.sh $vmname
            else

```

```

        echo " > " $vmname " is not pause, Ex Check Start VM !"
        if [ $(IfRunning $vmname) = 0 ] ; then
            echo " > " $vmname " is running, Ex Skip Start VM !"
        else
            echo " > " $vmname " is not running, Ex Start VM !"
            ./005-snapshotResuming.sh $vmname
        fi
    fi
else
    if [ "$ex" = "Test" ] ; then
        echo " > " $vmname " is disable VM so Not Start VM !"
    else
        if [ $(IfPause $vmname) = 0 ] ; then
            echo " > " $vmname " is pause, notEx Start VM !"
            ./005-snapshotResuming.sh $vmname
        else
            echo " > " $vmname " is not pause, notEx Check Start VM !"
            if [ $(IfRunning $vmname) = 0 ] ; then
                echo " > " $vmname " is running, Ex Skip Start VM !"
            else
                echo " > " $vmname " is not running, Ex Start VM !"
                ./005-snapshotResuming.sh $vmname
            fi
        fi
        sleep 5
    fi
fi
done
}

```

```
cd /home/hyunsu/BackUpCenter/20221014-snapshot/
```

```
PrintVmList
```

```
PauseVM
```

```
Snapshot
```

```
Delete
```

```
Resuming
```

```
echo -e "\n"
```

```
echo "#####"
```

```
echo "##### MAKE SNAPSHOT END ! #####"
```

```
echo "#####"
```

```
echo -e "\n"
```