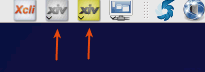
# Accessing CMS devices from a Linux workstation

Summary:

1. How to configure XIV GUI
2. Launching RDP for Windows servers
3. XIV scripts (HC + others)
4. Device list – dl
5. Host key swap – hks
6. Scripts for other devices
7. Firefox bookmarks for SVC, V7000, FSs
8. Preview mode for XIV configuration scripts

## 1. How to configure XIV GUI

I have configured my desktop to have two drawers for XIV launchers in the panel:



Two drawers because we have many sites.

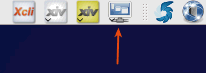
Every site has one launcher in one of the drawers and it will open XIV GUI for all XIVs in that site.

More details about that could be found in this document in the GSA drive.

[https://pokgsa.ibm.com/projects/s/sce-plus/XIV/How-to%20Configure%20XIV%20GUI.doc](https://pokgsa.ibm.com/projects/s/sce-plus/XIV/How-to Configure XIV GUI.doc)

## 2. Launching RDP for Windows servers

To launch RDP sessions click the drawer in the panel to open it:

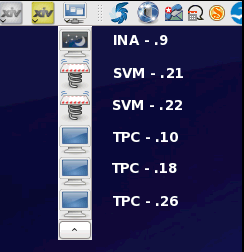


Once the drawer is opened click the launcher for the server you want to access.

The launchers are configured for my LDAP id and I will need only to type the password. This is an example of a command started from the launcher:

xfreerdp -d SSM -u us2d9999 -g 1500x900 --ignore-certificate --plugin cliprdr --plugin rdpdr --data disk:XIV:/home/dzunic/SCE+/XIV -- 146.89.141.10

Because IPs are the same in every site, this will work for every location.

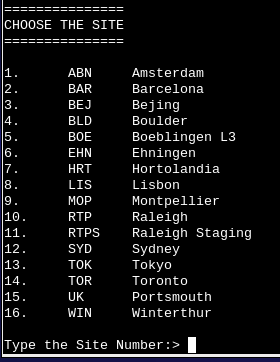


## 3. XIV scripts (HC + others)

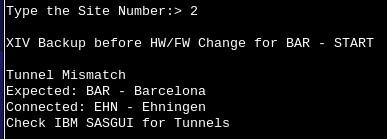
These scripts are used mostly for XIV health checking, but they could be used as templates for other devices. They are uploaded to the GSA drive and instructions are in the following document:

[https://pokgsa.ibm.com/projects/s/sce-plus/XIV/Scripts/XIV%20Scripts%20for%20Linux.odt](https://pokgsa.ibm.com/projects/s/sce-plus/XIV/Scripts/XIV Scripts for Linux.odt)

The common part of all scripts is that they prompt you to choose the site from the list and then they will compare it to the tunnel opened in SASGUI.

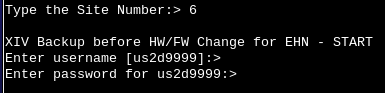


If the answer is not the same as the site in SASGUI, the script will display the message and exit.

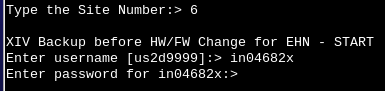


If the chosen site and the site in SASGUI are the same, the script will continue with the user id and password prompts. The script will list the default user name (currently us2d9999). You can change it by adjusting the variable DEFUSER at the beginning of each script.

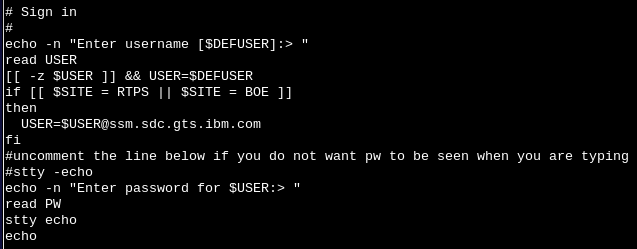
Pressing enter will accept the default user name:



Instead of pressing enter, you can type another user name and that name will be used further in the scripts:



By default the password will be displayed during typing. Only one thing needs to be changed in the script to make the password invisible:



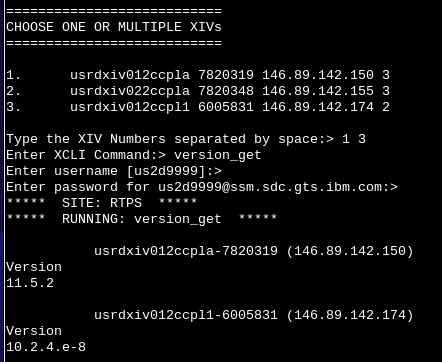
Majority of the scripts have hard coded commands.

There is also a script that will ask you to enter the command you want to execute.

Most of the scripts are running commands against all XIV devices in one site.

In case there is a need to run the script for only one or few devices, there are scripts for that.

They will display the list of devices to choose from before executing the commands.



The example of scripts that prompts you to enter a command:

xcli\_comm\_mXIV.ksh

xcli\_comm.ksh

The first one is for one or multiple devices and the second one for all devices in the site.

The script will ask you to enter the command and then it will process it. This is good for simple and short commands.

The example of the hard coded command scripts:

PreChangeBackup\_mXIV.ksh

PreChangeBackup.ksh

The first one is for one or multiple devices and the second one for all devices in the site.

This is good for long commands or set of commands that need to be executed more frequently.

For a particular activity you create the script by hard coded the commands in the basic script template.

The scripts could be easily adapted for different commands and other devices. Only part where the commands are listed need to be changed.

## 4. Device list – dl

The script, */home/dzunic/XIV/MyScripts/device\_list.ksh*, lists all devices, IPs and for XIV, switches and FS serial numbers, for the chosen site in a second.

Yes, it displays only names, IPs and some serial numbers. But most of the time we need only that. In case we need more information about devices and sites we could open the inventory documents.

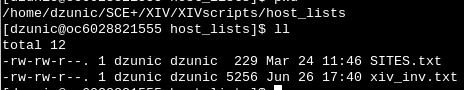
I use the alias “dl” for it that is defined in the environment file */home/dzunic/.bashrc*:

alias dl="/home/dzunic/SCE+/XIV/MyScripts/device\_list.ksh"

This script does not check if the tunnel is correct because you might have need to see devices for different sites than the one you are working on at the moment.

Data about devices is stored in the flat inventory files:

XIV - in the folder */home/dzunic/XIV/XIVscripts/host\_lists/xiv\_inv.txt*



SVC - */home/dzunic/SCE+/SVC/Scripts/svc\_inv.txt*

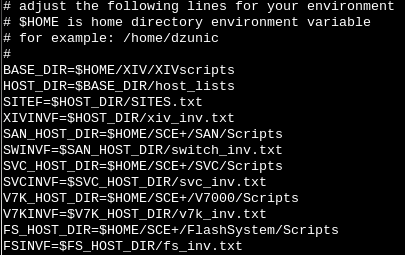
V7000 - */home/dzunic/SCE+/V7000/Scripts/v7k\_inv.txt*

Switches - */home/dzunic/SCE+/SAN/Scripts/switch\_inv.txt*

FS - */home/dzunic/SCE+/FlashSystem/Scripts/fs\_inv.txt*

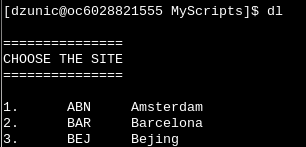
The file SITES.txt lists all sites and it is used by scripts for all devices.

At the beginning of all scripts there is an adjustment part where you need to do adjustment for file and folders locations on your workstation.

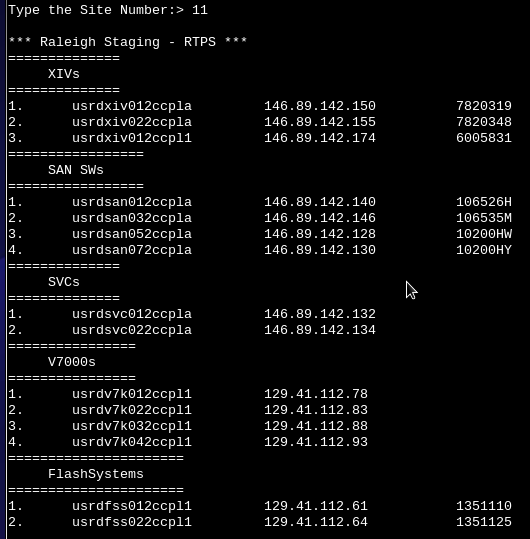


This script is very useful and fast when you need to get a short list of devices for one site.

Just type “dl”:



and then choose the site:



## 5. Host key swap – hks

For switches, SVCs, V7000s and FSs I collected host keys for putty and plink and known hosts for ssh for every site.

[dzunic@oc6028821555 ~]$ cd /home/dzunic/.putty

[dzunic@oc6028821555 .putty]$ ll

total 216

-rw-------. 1 dzunic dzunic 600 May 15 16:36 randomseed

drwx------. 2 dzunic dzunic 4096 Jul 6 2014 sessions

-rw-rw-r--. 1 dzunic dzunic 13851 May 29 17:26 sshhostkeys

-rw-rw-r--. 1 dzunic dzunic 13851 Apr 2 15:23 sshhostkeys\_ABN

-rw-rw-r--. 1 dzunic dzunic 3976 Mar 30 17:33 sshhostkeys\_BAR

-rw-rw-r--. 1 dzunic dzunic 8302 Sep 1 2014 sshhostkeys\_BEJ

-rw-rw-r--. 1 dzunic dzunic 10134 May 1 13:42 sshhostkeys\_BLD

-rw-rw-r--. 1 dzunic dzunic 3406 Mar 26 2014 sshhostkeys\_BOE

-rw-rw-r--. 1 dzunic dzunic 12924 Mar 30 17:29 sshhostkeys\_EHN

-rw-rw-r--. 1 dzunic dzunic 4514 Mar 26 2014 sshhostkeys\_HRT

-rw-rw-r--. 1 dzunic dzunic 3902 Mar 30 17:38 sshhostkeys\_LIS

-rw-rw-r--. 1 dzunic dzunic 3440 Sep 3 2014 sshhostkeys\_MOP

-rw-rw-r--. 1 dzunic dzunic 20081 May 15 16:36 sshhostkeys\_RTP

-rw-rw-r--. 1 dzunic dzunic 4992 May 1 13:34 sshhostkeys\_RTPS

-rw-rw-r--. 1 dzunic dzunic 6696 Jul 15 2014 sshhostkeys\_SYD

-rw-rw-r--. 1 dzunic dzunic 4516 Mar 26 2014 sshhostkeys\_TOK

-rw-rw-r--. 1 dzunic dzunic 8454 Mar 30 17:18 sshhostkeys\_TOR

-rw-rw-r--. 1 dzunic dzunic 3330 Apr 17 14:02 sshhostkeys\_UK

-rw-rw-r--. 1 dzunic dzunic 5087 Mar 30 17:23 sshhostkeys\_WIN

[dzunic@oc6028821555 .putty]$ cd /home/dzunic/.ssh

[dzunic@oc6028821555 .ssh]$ ll

total 140

-rw-------. 1 dzunic dzunic 1743 Sep 28 2012 id\_rsa

-rw-------. 1 dzunic dzunic 519 Sep 28 2012 id\_rsa\_com.pub

-rw-rw-r--. 1 dzunic dzunic 381 Sep 28 2012 id\_rsa.pub

-rw-r--r--. 1 dzunic dzunic 10293 May 29 17:26 known\_hosts

-rw-r--r--. 1 dzunic dzunic 10293 Apr 2 15:20 known\_hosts\_ABN

-rw-r--r--. 1 dzunic dzunic 2996 Mar 30 17:31 known\_hosts\_BAR

-rw-r--r--. 1 dzunic dzunic 6148 Sep 2 2014 known\_hosts\_BEJ

-rw-r--r--. 1 dzunic dzunic 4816 May 1 13:41 known\_hosts\_BLD

-rw-r--r--. 1 dzunic dzunic 2552 Jul 15 2014 known\_hosts\_BOE

-rw-r--r--. 1 dzunic dzunic 6500 Mar 30 17:26 known\_hosts\_EHN

-rw-r--r--. 1 dzunic dzunic 3312 Jul 15 2014 known\_hosts\_HRT

-rw-r--r--. 1 dzunic dzunic 2924 Mar 30 17:35 known\_hosts\_LIS

-rw-r--r--. 1 dzunic dzunic 2684 Jul 15 2014 known\_hosts\_MOP

-rw-r--r--. 1 dzunic dzunic 10111 May 15 16:30 known\_hosts\_RTP

-rw-r--r--. 1 dzunic dzunic 3714 May 1 13:30 known\_hosts\_RTPS

-rw-r--r--. 1 dzunic dzunic 4892 Jul 15 2014 known\_hosts\_SYD

-rw-r--r--. 1 dzunic dzunic 3312 Jul 15 2014 known\_hosts\_TOK

-rw-r--r--. 1 dzunic dzunic 6174 Mar 30 17:13 known\_hosts\_TOR

-rw-r--r--. 1 dzunic dzunic 2478 Apr 17 13:57 known\_hosts\_UK

-rw-r--r--. 1 dzunic dzunic 3758 Mar 30 17:22 known\_hosts\_WIN

Before running putty, plink or ssh, run the script */home/dzunic/SCE+/SAN/Scripts/host\_key\_swap.ksh* with alias "hks". It will copy the site files for the chosen site, for example if you are in ABN:

cp -p sshhostkeys\_ABN sshhostkeys

cp -p known\_hosts\_ABN known\_hosts

It will also compare a chosen site with the tunnel in SASGUI as described above for the XIV scripts.

The alias “hks” is defined in the environment file */home/dzunic/.bashrc*:

alias hks="/home/dzunic/SCE+/SAN/Scripts/host\_key\_swap.ksh"

All script templates for switches, SVCs, V7Ks and FSs do the host key swapping for the chosen site.

## 6. Scripts for other devices

There are 4 templates for each device type.

They could reside in any folder. In my workstation the templates are located in the following folders:

XIV - */home/dzunic/SCE+/XIV/Myscripts/templates*

SVC - */home/dzunic/SCE+/SVC/Scripts/*

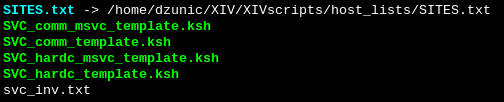
V7000 - */home/dzunic/SCE+/V7000/Scripts/*

Switches - */home/dzunic/SCE+/SAN/Scripts/*

FS - */home/dzunic/SCE+/FlashSystem/Scripts/*

The inventory files “device”\_inv.txt could also be located in any folder. You just need to update the templates with locations on your workstation.

For example these are the templates for SVCs:



First two templates are asking for the command to be entered. You can write your code in the last two templates. All templates will do the following:

* prompt to choose the site
* check if the SASGUI tunnel corresponds to the chosen site
* for V7000 and FS check if devices exist in the chosen site
* do the host key swapping for the chosen site
* prompt to choose a device (templates for one or multiple devices)
* prompt for the command (first 2 templates)
* prompt for the ID and password
* execute entered command or the code

You can add as many lines of the code as you want.

In addition to templates the following two files need to be in the folder:

SITES.txt – a link to the list of all sites in the XIV folder

[dzunic@oc6028821555 Scripts]$ cat SITES.txt

ABN Amsterdam

BAR Barcelona

BEJ Bejing

BLD Boulder

BOE Boeblingen L3

EHN Ehningen

HRT Hortolandia

LIS Lisbon

MOP Montpellier

RTP Raleigh

RTPS Raleigh Staging

SYD Sydney

TOK Tokyo

TOR Toronto

UK Portsmouth

WIN Winterthur

svc\_inv.txt – a list of the SVCs

[dzunic@oc6028821555 Scripts]$ cat svc\_inv.txt

EHN deehsvc011ccpla 146.89.142.132

EHN deehsvc021ccpla 146.89.142.134

EHN deehsvc011ccpl1 146.89.142.170

EHN deehsvc021ccpl1 146.89.142.172

EHN deehsvc011ccplb 129.41.112.4

EHN deehsvc021ccplb 129.41.112.6

EHN deehsvc011ccpl2 129.41.112.142

EHN deehsvc021ccpl2 129.41.112.144

RTP usrdsvc011ccpla 146.89.142.132

RTP usrdsvc021ccpla 146.89.142.134

...

A little knowledge of scripting is needed to use hard coded templates. First the templates have to be adjusted for the folders on your workstation. Then you need to add your code and some other things like redirecting your code to a file if you want to save the output. This is the example of the script that checks host ports for one SVC and redirects the output to a file.

The adjustment part:

# adjust the following lines for your environment

# $HOME is home directory environment variable

# for example: /home/dzunic

#

HOST\_DIR=$HOME/SCE+/SVC/Scripts

SITEF=$HOST\_DIR/SITES.txt

INVF=$HOST\_DIR/svc\_inv.txt

SASGUI\_DIR=$HOME/.sasgui

PUTTY\_DIR=$HOME/.putty

SSH\_DIR=$HOME/.ssh

OUT\_DIR=$HOST\_DIR/outs

# default user

DEFUSER=us2d9999

# END of adjustment part

The code could have as little as one line and up to hundreds of lines:

# type your description below --> RUNNING: your description

echo -e "\*\*\*\*\* SITE: $SITE \*\*\*\*\*\n\*\*\*\*\* RUNNING: checking host ports \*\*\*\*\*\n"

### Add your code below ###

DATE=`date +%Y%m%d`

TIME=`date +%H%M`

OUT\_FILE=$OUT\_DIR/"$SVCN"\_host\_ports\_"$DATE"\_$TIME

echo -e "$SVCN ($SVCIP)\n" | tee $OUT\_FILE

for i in `sshpass -p $PW ssh -o stricthostkeychecking=no -n $USER@$SVCIP lshost -nohdr | awk '{print $1}'`

do

echo -n "$i "

echo "id $i" >> $OUT\_FILE

sshpass -p $PW ssh -o stricthostkeychecking=no -n $USER@$SVCIP lshost $i | grep -Ew 'name|status|WWPN|state' |sed '$!N;s/\n/ /' | awk '{print substr($0, index($0,$2))}' >> $OUT\_FILE

echo >> $OUT\_FILE

done

echo -e "\n"

### End of your code ###

By executing this script you will be first prompted to choose the site and then to choose the SVC and type the user name and password:

...

15. UK Portsmouth

1. WIN Winterthur

Type the Site Number:> 11

==============

CHOOSE THE SVC

==============

1. usrdsvc012ccpla 146.89.142.132

2. usrdsvc022ccpla 146.89.142.134

Type the SVC Number:> 1

Enter username [us2d9999]:>

Enter password for us2d9999:>

...

The script will display host ids on the screen

\*\*\*\*\* SITE: RTPS \*\*\*\*\*

\*\*\*\*\* RUNNING: checking host ports \*\*\*\*\*

usrdsvc012ccpla (146.89.142.132)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43

and save its output in a file. This is a part of the output file:

usrdsvc012ccpla (146.89.142.132)

id 0

LogHost\_ESX\_C1\_32\_L1 status online

21000024FF393CE1 state active

21000024FF393CC4 state active

id 1

LogHost\_ESX\_C1\_32\_L2 status online

21000024FF393CE0 state active

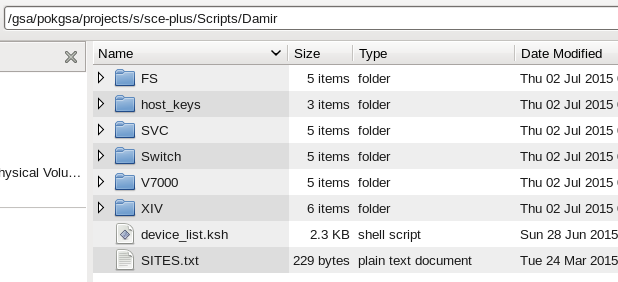
21000024FF393CC5 state active

...

Everything is very similar for other devices.

All scripts and inventory files are uploaded to the GSA drive:

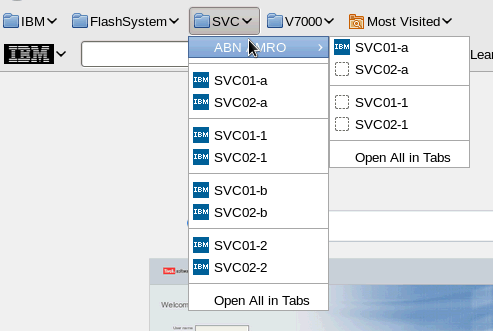
*/gsa/pokgsa/projects/s/sce-plus/Scripts/Damir*

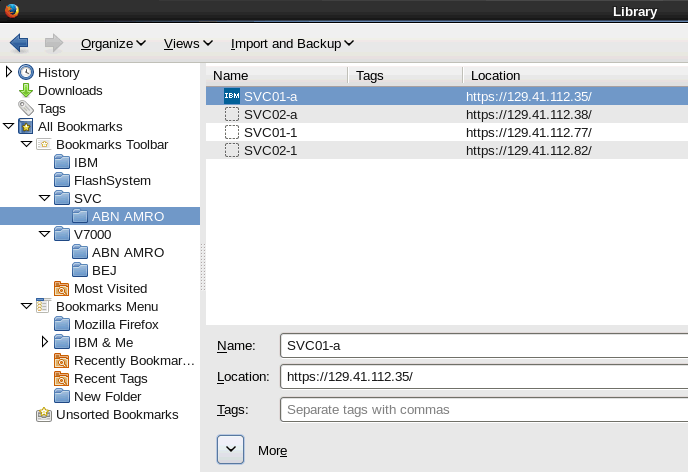


## 7. Firefox bookmarks for SVC, V7000, FSs

To connect to web GUI for SVC, V7000 and FS I created the Firefox bookmarks. Almost all devices in old sites use the same set of IP addresses. Unfortunately, new sites, like ABN and BEJ, are coming with different sets of IP addresses for some devices so the folders for these sites had to be created.

Clicking any of these bookmarks will open a tab for the chosen device in the current web browser window.





## 8. Preview mode for XIV configuration scripts

These scripts are mostly a list of individual cli commands.

In scripts for logical add XIV storage I am using a preview mode to make sure that everything is OK.

The script will substitute all variables and list all commands exactly the same way they will be executed.

This is a good way to check the scripts before real execution. The same idea could be used for other devices as well.

To illustrate the idea, this is is the beginning of one script:

[dzunic@oc6028821555 BLD\_091-101\_setup]$ cat 2-08\_1\_xiv\_gen3\_POD1\_09\_4TB\_volmapping.ksh

#

# Ask the preview question

#

echo -e " For preview type yes or y [default=yes]\n For execution type x "

read answer

answer=`echo $answer | awk '{print toupper($1)}'`

if [[ $answer = YES || $answer = Y || $answer = "" ]]

then

preview=echo

user=dummy

pw=dummy

elif [[ $answer = X ]]

then

preview=""

# default user

DEFUSER=us2d9999

echo -n "Enter username [$DEFUSER]:> "

read user

[[ -z $user ]] && user=$DEFUSER

#uncomment the line below if you do not want pw to be seen when you are typing

#stty -echo

echo -n "Enter password for $user:> "

read pw

stty echo

echo

else

echo " Wrong answer !!!!"

exit 33

fi

# Installation Site

site=usbd

# Server Instance Number

Instance\_No=09

# pod number

nn=01

# zone (a or b)

zone=a

# volume size (GB)

# 3 TB diss: vols=4852

# 4 TB disks: vols=6504

vols=6504

# POD1 XIV#09

# Please confirm Pool names are correct

XIV09p\_POOL="$site"\_"$nn""$zone"\_xiv\_"$Instance\_No"\_sas\_"$vols"gb\_r1

# Do not touch below line

svc1p="$site"svc011ccpl1

ip09p=146.89.142.214

echo "Mapping Volume POD1 XIV 09"

echo -e "\nMapping Volumes in the pool $XIV09p\_POOL\n"

$preview xcli -m $ip09p -u $user -p $pw map\_vol cluster=$svc1p vol="$XIV09p\_POOL"\_001 lun=1

$preview xcli -m $ip09p -u $user -p $pw map\_vol cluster=$svc1p vol="$XIV09p\_POOL"\_002 lun=2

$preview xcli -m $ip09p -u $user -p $pw map\_vol cluster=$svc1p vol="$XIV09p\_POOL"\_003 lun=3

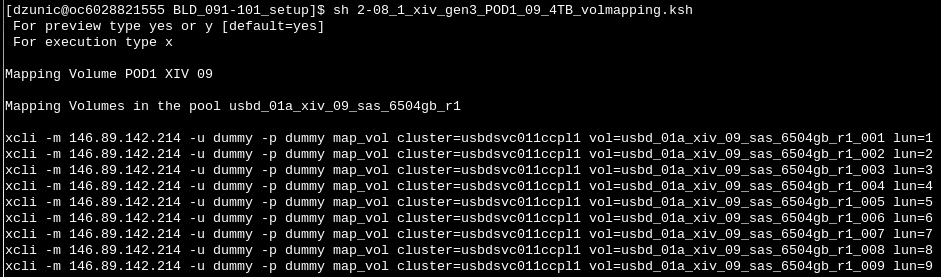
$preview xcli -m $ip09p -u $user -p $pw map\_vol cluster=$svc1p vol="$XIV09p\_POOL"\_004 lun=4

$preview xcli -m $ip09p -u $user -p $pw map\_vol cluster=$svc1p vol="$XIV09p\_POOL"\_005 lun=5

$preview xcli -m $ip09p -u $user -p $pw map\_vol cluster=$svc1p vol="$XIV09p\_POOL"\_006 lun=6

...

Executing this script in the preview mode will display:



If we are satisfied with the preview output, we can run the script in an execution mode:

