

Linux on System z Resource Automation Using Co:Z on z/OS

Kathy Hodge
EMC

Wednesday, March 12, 2014: 1:30 PM-2:30 PM
Session Number 14865

<http://www.share.org/>



EMC Software Quality Engineering



- Mainframe Enablers Product Interoperability Testing with Linux on System z
- z/OS Centric Staff and Software Products
- Opportunity to Interface with Linux from z/OS using Co:Z
 - EMC's Igor Pavlyutenkov- "I first learned about this product in 2007 at a Linux forum. At that time, I worked for Russian Railways and supported z/VM and z/Linux, and found it very helpful in my work."

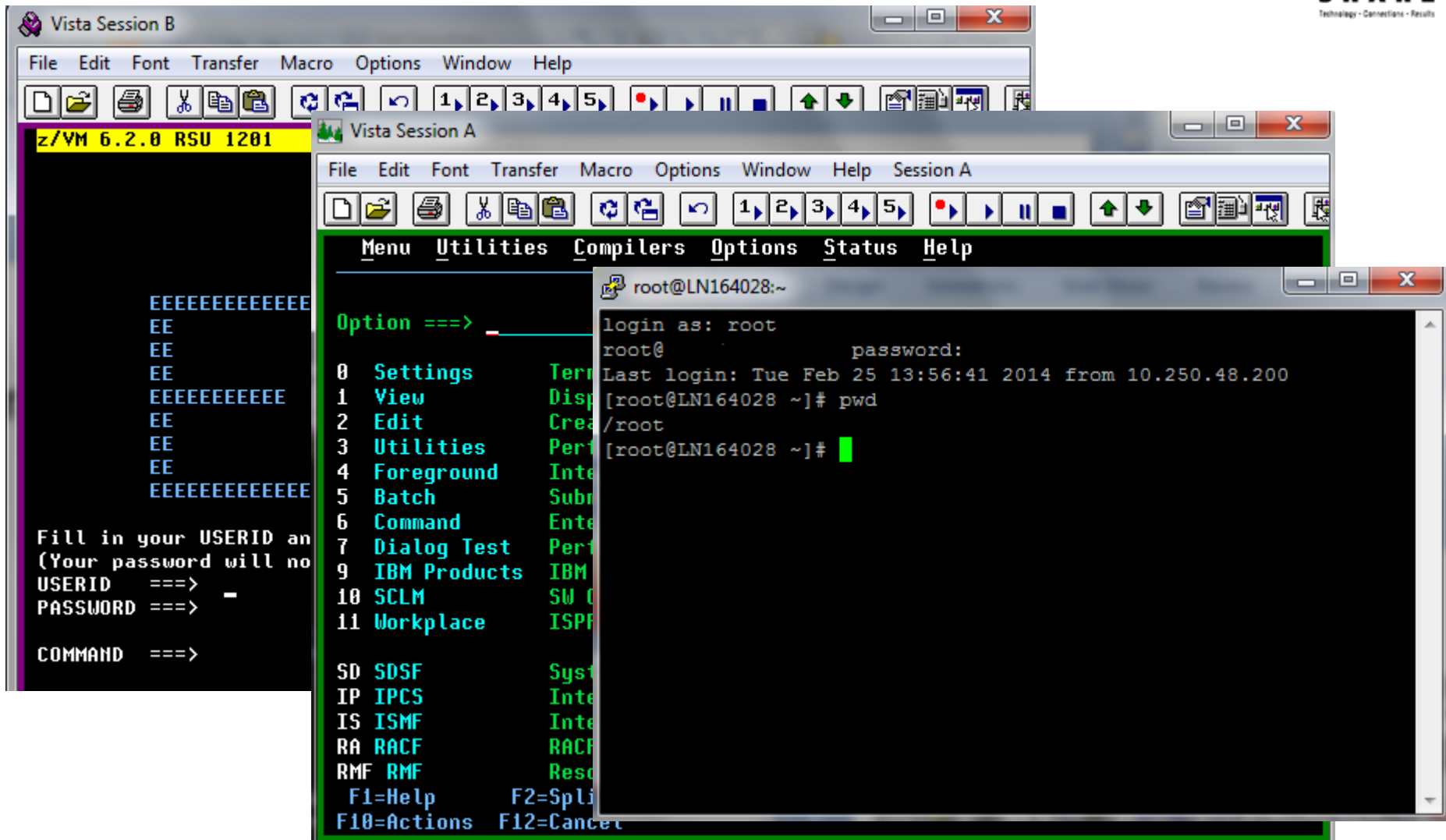


EMC Software Quality Engineering Challenges

- Linux Interoperability Test Scenario Configurations with One z/OS Job Stream
 - Reboot Linux Server
 - Configure Linux device environment
 - Invoke MFE z/OS Product tests
 - Verify test outcomes
 - Shutdown Linux Server
- One Operating System Interface for Automation
 - z/OS
- I/O Driven from Linux on System z



Multiple Operating System Interfaces Reduced to One



Linux on System z Resource Automation Software Components



- Dovetailed Technologies
 - Co:Z Launcher
- EMC Mainframe Enablers
 - TimeFinder/Clone Mainframe SNAP Facility
- IBM z/OS LPAR
- IBM z/VM Guest Server
- Red Hat Enterprise Linux for IBM System z



Dovetailed Technologies



- The Co:Z Co-Processing Toolkit for z/OS is a suite of mainframe technology products used to enhance the traditional IBM workload
- Co:Z Toolkit Components
 - Co:Z SFTP for z/OS
 - Co:Z Launcher for z/OS
 - Co:Z Dataset Pipes for z/OS
 - Co:Z Batch
 - Co:Z Load Balancer for z/OS
- Co:Z Toolkit is available free to use under Dovetailed Technologies Community License
- Enterprise License and support agreements are also available



Co:Z Launcher



- Batch utility which remotely launches a process on a distributed system
 - Linux, Windows, or other Unix/POSIX environments
- Redirects input and output from a process to traditional z/OS datasets or spool files
- Remote processes are securely launched using proven SSH (Secure Shell) technology to the target platform
- Performance benefits when using Linux on System z with IFLs and HiperSockets, however, not required
- For organizations that run both z/OS and System z Linux



Co:Z Launcher cont'd

- Co:Z Toolkit installation on z/OS
- Co:Z Target System Toolkit on each remote system
- Requires IBM Ported Tools for z/OS (OpenSSH) product to be installed and configured on USS
 - OpenSSH public/private keypair
 - RACF Digital Certificate
 - SSH_ASKPASS authentication

EMC Mainframe Enablers Products Under Test



- The EMC Mainframe Enablers (MFE) is a suite of components used to monitor and manage your EMC Symmetrix system
 - ResourcePak Base for z/OS
 - SRDF Host Component for z/OS
 - TimeFinder/Clone Mainframe SNAP Facility
 - TimeFinder/Mirror for z/OS
 - Consistency Groups for z/OS
 - TimeFinder Utility for z/OS
 - AutoSwap



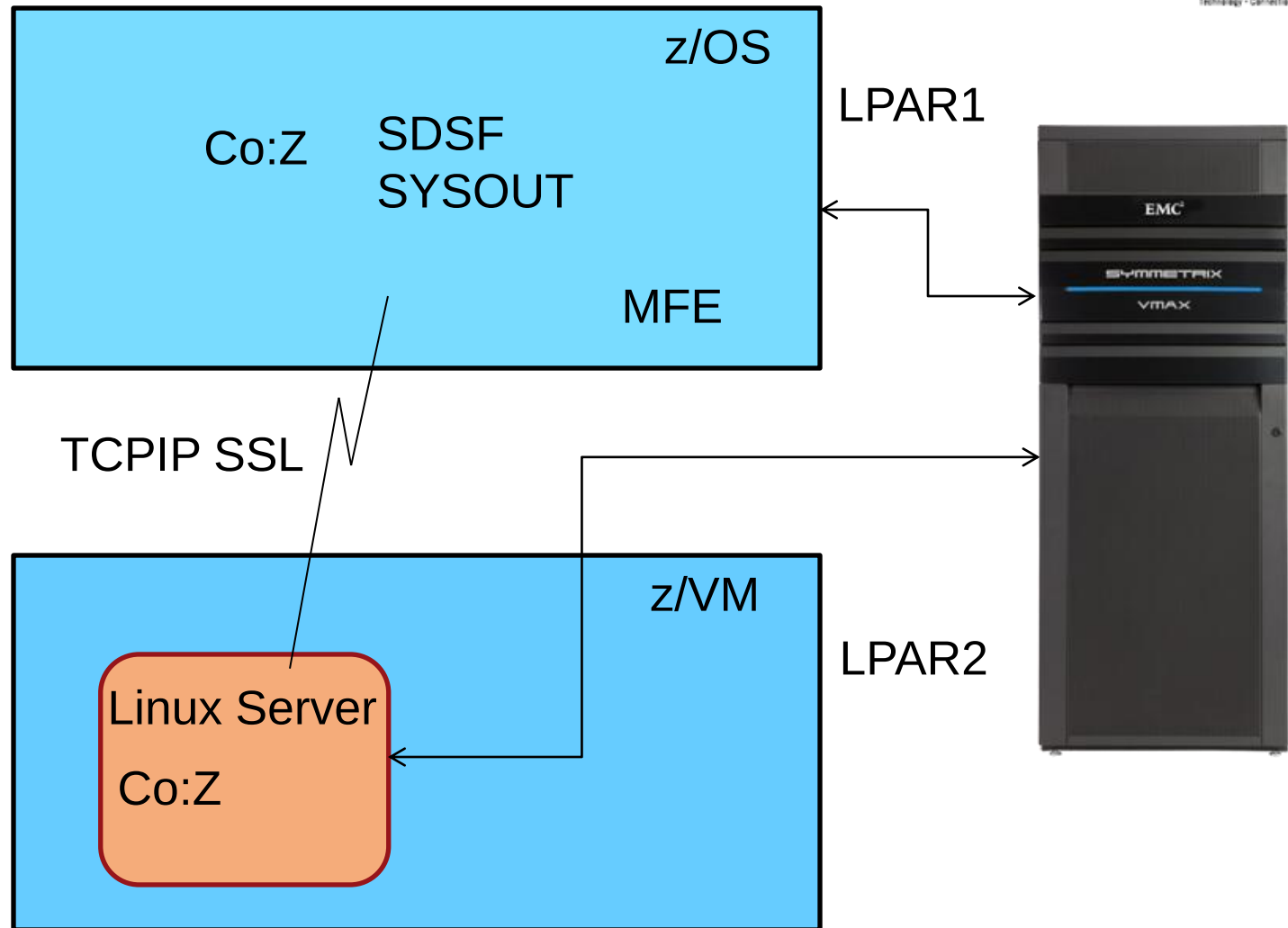


Red Hat Enterprise Linux for IBM System z

- Red Hat Enterprise Linux Server release 6.0
- IBM z/VM 6.2
- FBA Devices
 - Fibre-channel adapters on zSeries hosts running Linux
- CKD Devices
 - UCBs defined at z/OS and z/VM



Software Configuration



Co:Z Launcher Job Example

```
//COZLN2 JOB (0202),'COZ',CLASS=S,MSGCLASS=X
//MYLIB JCLLIB ORDER='APPLQA.COZ.PROC'
//STEP1 EXEC PROC=EXCOZ,
//  ARGS='root@10.92.2.28'
//COZCFG DD *
#agent-options=-LD,t      ← Default Logging with timestamp
#target-env-COZ_LOG=D,t   ← target components Logging options
server-env-PASSWD_DSN=APPLQA.PASSWD(LN164)
//STDOUT DD SYSOUT=*      ← Linux results output
//STDIN DD *              ← Linux commands follow
# This is input to the remote shell
echo "We are running on: " `uname -sr`
Isdasd
```

//

Complete your session evaluations online at www.SHARE.org/AnaheimEval

Co:Z Launcher EXCOZ PROC



Stored procedure for executing the Co:Z Launcher

```
//EXCOZ  PROC ARGS=,  
// LOG='-LI',  
// TARGET='user@lss.emc.com',  
// COZCFGD='COZCFGD',  
// REGSIZE='32M',  
// LEPARM=""  
//COZLNCH EXEC PGM=COZLNCH,REGION=&REGSIZE,  
// PARM='&LEPARM/&LOG &ARGS &TARGET'  
//STEPLIB DD DISP=SHR,DSN=COZ.PROD.LOADLIB  
//COZCONF DD DISP=SHR,DSN=APPLQA.COZ.PROC(&COZCFGD)  
// DD DDNAME=COZCFG  
//COZCFG DD DUMMY  
//SYSOUT DD SYSOUT=*  
//
```



Co:Z Launcher Job Execution on z/OS



Linux on z – LN164

z/OS

bash shell

```
//COZLN2 JOB
      (0202),'COZ',CLASS=S,MSGCLASS=X

//MYLIB JCLLIB
      ORDER='APPLQA.COZ.PROC'
//STEP1 EXEC PROC=EXCOZ,
//      ARGS='root@10.92.2.28'
//STDOUT DD SYSOUT=*
//STDIN DD

# This is input to the remote shell
echo "We are running on: " `uname
    -sr`

Isdasd

//
```

Co:Z Launcher Sends Commands to Linux



Linux on z – LN164

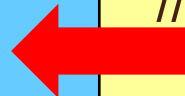
z/OS

bash shell

```
//COZLN2 JOB  
      (0202),'COZ',CLASS=S,MSGCLA  
      SS=X
```

```
//MYLIB JCLLIB  
      ORDER='APPLQA.COZ.PROC'  
//STEP1 EXEC PROC=EXCOZ,  
//      ARGS='root@10.92.2.28'  
//STDOUT DD SYSOUT=*  
//STDIN  DD
```

This is input to the remote shell
echo "We are running on: " `uname -sr`
lsdasd
//



//



Linux Resolves System Variables & Commands



Linux on z – LN164

z/OS

bash shell

This is input to the remote shell
echo "We are running on: Linux
2.6.32-71.el6.s390x
lsdasd
//



```
//COZLN2 JOB
      (0202),'COZ',CLASS=S,MSGCLA
      SS=X

//MYLIB JCLLIB
      ORDER='APPLQA.COZ.PROC'
//STEP1 EXEC PROC=EXCOZ,
//      ARGS='root@10.92.2.28'
//STDOUT DD SYSOUT=*
//STDIN DD

//
```

Co:Z Launcher Execution Results Back to z/OS



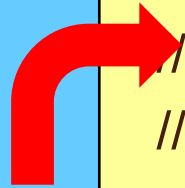
Linux on z – LN164

z/OS

bash shell

This is input to the remote shell
echo "We are running on: **Linux**
2.6.32-71.el6.s390x
lsdasd
//

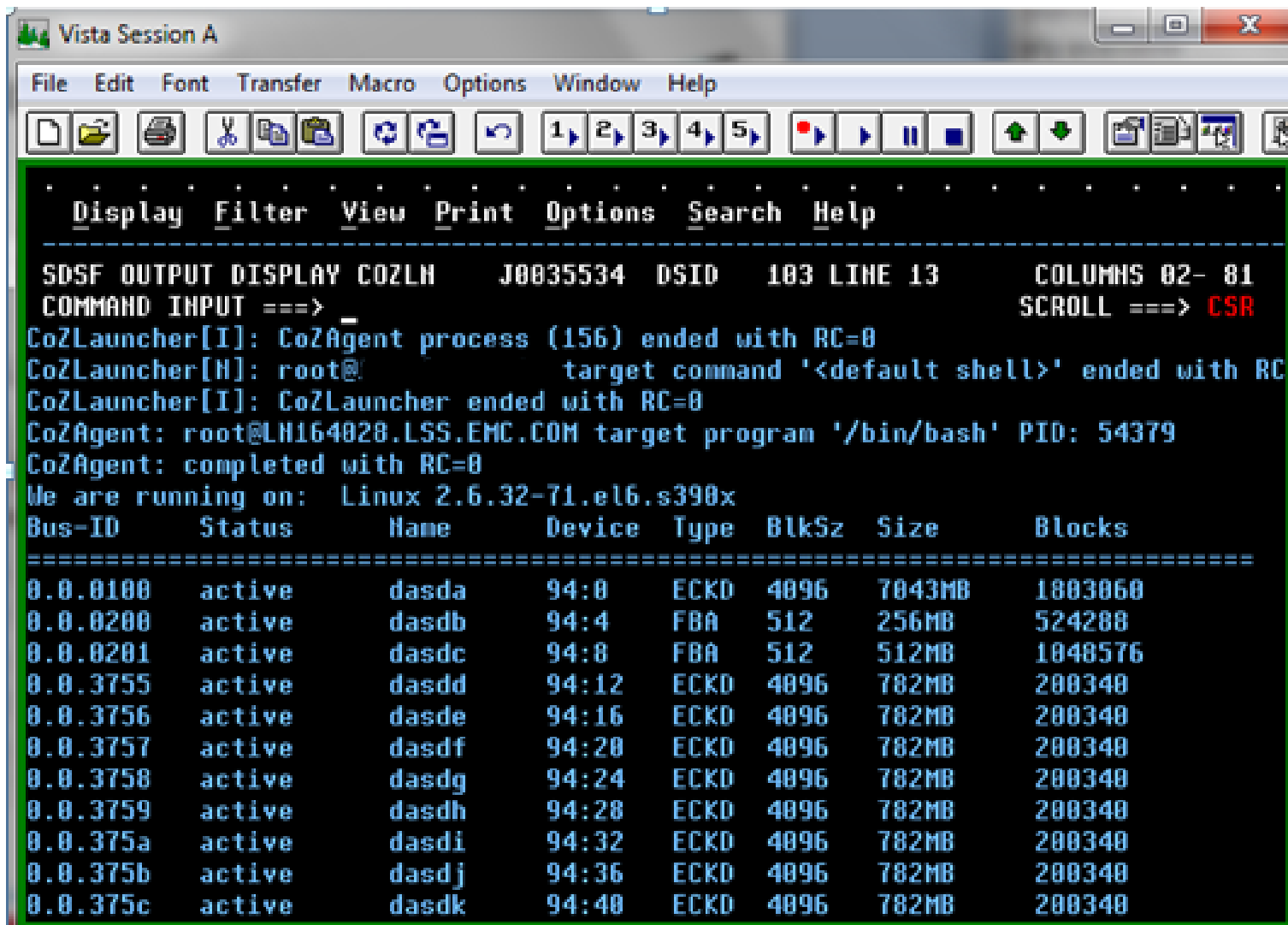
//COZLN2 JOB
 (0202),'COZ',CLASS=S,MSGCLA
 SS=X
//MYLIB JCLLIB
 ORDER='APPLQA.COZ.PROC'
//STEP1 EXEC PROC=EXCOZ,
// ARGS='root@10.92.2.28'
//STDOUT DD SYSOUT=*
//STDIN DD



//



Linux Results in SDSF SYSOUT on z/OS



```

SDSF OUTPUT DISPLAY COZLN      J0035534  DSID   103 LINE 13      COLUMNS 02- 81
COMMAND INPUT ==> _
CoZLauncher[I]: CoZAgent process (156) ended with RC=0
CoZLauncher[H]: root@          target command '<default shell>' ended with RC
CoZLauncher[I]: CoZLauncher ended with RC=0
CoZAgent: root@LN164020.LSS.EMC.COM target program '/bin/bash' PID: 54379
CoZAgent: completed with RC=0
We are running on: Linux 2.6.32-71.el6.s390x
Bus-ID      Status      Name      Device  Type  BlkSz  Size      Blocks
=====
0.0.0100    active      dasda     94:0    ECKD   4096   7043MB    1803060
0.0.0200    active      dasdb     94:4    FBA    512    256MB     524208
0.0.0201    active      dasdc     94:8    FBA    512    512MB     1048576
0.0.3755    active      dasdd     94:12   ECKD   4096   782MB     200340
0.0.3756    active      dasde     94:16   ECKD   4096   782MB     200340
0.0.3757    active      dasdf     94:20   ECKD   4096   782MB     200340
0.0.3758    active      dasdg     94:24   ECKD   4096   782MB     200340
0.0.3759    active      dasdh     94:28   ECKD   4096   782MB     200340
0.0.375a    active      dasdi     94:32   ECKD   4096   782MB     200340
0.0.375b    active      dasdj     94:36   ECKD   4096   782MB     200340
0.0.375c    active      dasdk     94:40   ECKD   4096   782MB     200340
    
```

Co:Z Launcher on z/OS for Linux on System z

Resource Automation

Co:Z Resource Automation



- Server Startup & Shutdown
- Create Test Configuration Scripts or Commands
 - Attach CKD devices to server
 - Vary devices online to z/VM
 - Vary devices online to Linux
 - Low level format CKD
 - Add partitions to CKD and FBA
- Test MFE Product
 - Consistent Backup Using Snap Volume
 - Same job stream as Co:Z Launch backup verification
 - Validate Test Results



Co:Z Launcher Server Startup

Use ***XAUTOLOG*** to logon another user's Virtual Machine automatically with Co:Z Launcher

```
//STEP1 EXEC PROC=EXCOZ,  
//      TARGET='root@10.92.2.222'  
//COZCFG DD *  
#agent-options=-LD,t  
#target-env-COZ_LOG=D,t  
server-env-PASSWD_DSN=//APPLQA.PASSWD(LN164)  
//STDOUT DD SYSOUT=*  
//STDIN DD *  
# This is input to the remote shell  
echo "We are running on: " `uname -sr`  
vmcp XAUTOLOG LN164028  
//
```



Co:Z Launcher Server Startup Output



fromdsn(APPLQA.PASSWD(LN164))[N]: 1 records/80 bytes read; 8 bytes written

cozagent[N]: version: 1.1.2 2013-03-19

fromdsn(DD:STDIN)[N]: 5 records/400 bytes read; 116 bytes written

todsn(DD:STDOUT)[N]: 65 bytes read; 2 records/63 bytes written

todsn(DD:STDERR)[N]: 0 bytes read; 0 records/0 bytes written

CoZLauncher[N]: root@10.92.2.222 target command '<default shell>' ended with RC=0

CoZLauncher[I]: CoZLauncher ended with RC=0

CoZAgent: root@LN164222 target program '/bin/bash' PID: 1899

CoZAgent: completed with

We are running on: Linux 2.6.16.21-0.8-default

Command accepted RC=0



Co:Z Launcher Server Shutdown

shutdown -h Requests the system either be halted or powered off after it shut down, all device relationships are severed

```
//STEP2 EXEC PROC=EXCOZ,  
//      TARGET='root@10.92.2.28'  
//STDIN DD *  
# This is input to the remote shell  
echo "We are running on: " `uname -sr`  
shutdown -h now  
//
```



Co:Z Invocation of Linux on System z Scripts or Commands



- Commands in Scripts on the Linux file system
 - `./home/appqa/ivp50/snap/ntc/att_ntc_ckd`
 - Advantages
 - Scripts maintained at Linux file system
- Commands Inline in z/OS job stream
 - Advantages
 - All code consolidated to one location
 - Command variables can be used for devices
 - Variable capability allows code reuse across different applications



Device Attach Script

Attach devices to Linux Server using an existing shell script

```
//STEP2 EXEC PROC=EXCOZ,  
//COZLNCH.STDOUT DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//STDIN DD *  
# This is input to the remote shell  
echo "We are running on: " `uname -sr`  
. /home/appqa/ivp50/snap/ntc/att_ntc_ckd  
//
```

Device Attach Commands Inline

```
//STEP2 EXEC PROC=EXCOZ,  
//COZLNCH.STDOUT DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//STDIN DD *
```



vmcp vary online 3A15-3A18

vmcp vary online 3A19-3A1C ← zVM vary online

vmcp att 3A15 ln164028 3755 ← Linux attach command

vmcp att 3A16 ln164028 3756

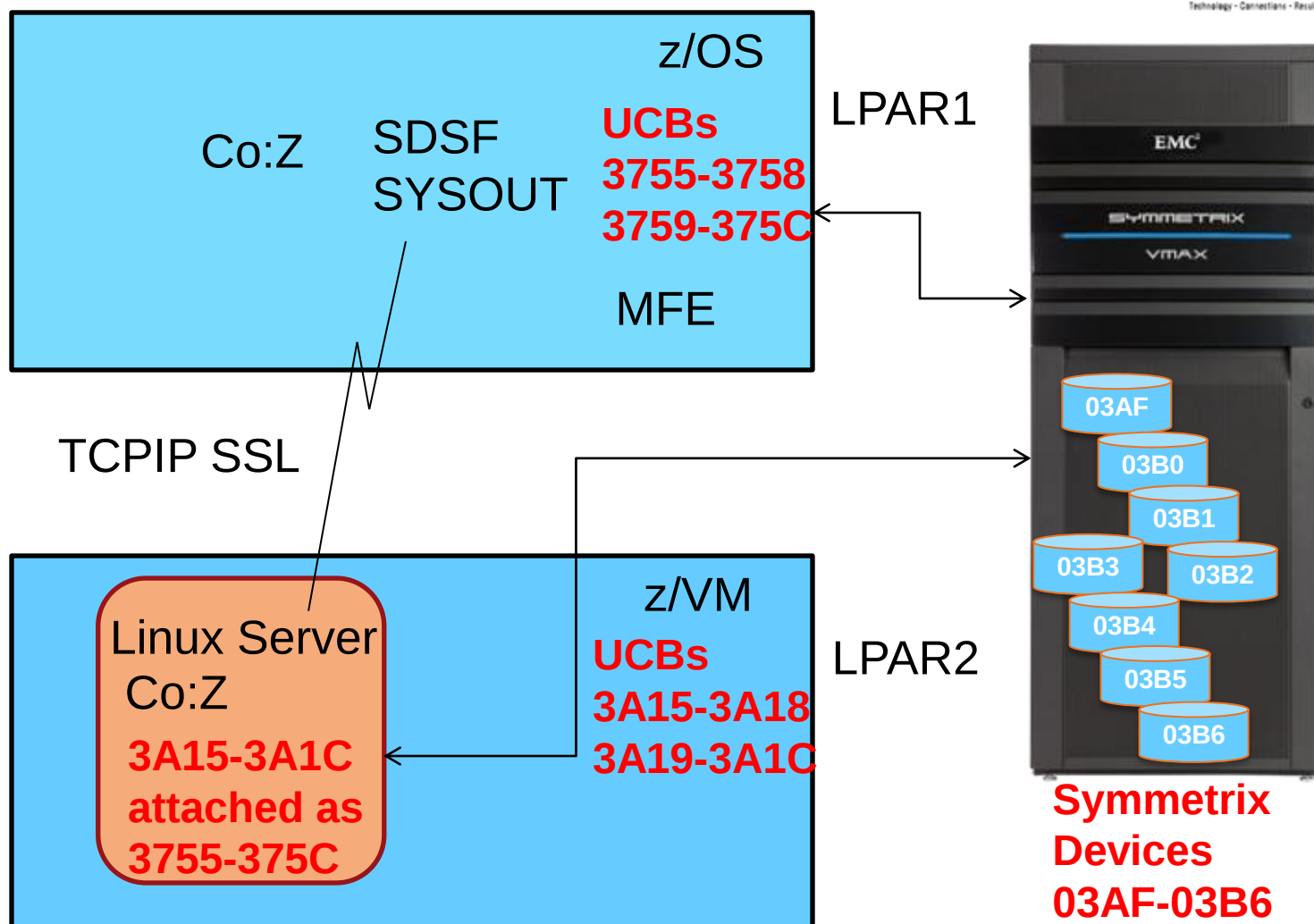
...

vmcp att 3A1C ln164028 375C

chccwdev -e 3755-3758 ← Linux vary online

chccwdev -e 3759-375C

Device Configuration Attached & Online



Device Attach Results

CoZAgent: root@LN164028.LSS.EMC.COM target program '/bin/bash' PID: 1807

CoZAgent: completed with RC=0

DASD 3A15 ATTACHED TO LN164028 3755 WITH DEVCTL

DASD 3A16 ATTACHED TO LN164028 3756 WITH DEVCTL

...

DASD 3A1B ATTACHED TO LN164028 375B WITH DEVCTL

DASD 3A1C ATTACHED TO LN164028 375C WITH DEVCTL

Setting device 0.0.3755 online

Done

Setting device 0.0.3756 online

Done

Setting device 0.0.3757 online

Done

Setting device 0.0.3758 online

CKD Device Low Level Format Scripts and Commands



Format devices using dasdfmt command applying labels to attached devices

```
//COZLNCH.STDOUT DD SYSOUT=*
```

```
//STDOUT DD SYSOUT=*
```

```
//STDIN DD *
```

```
# This is input to the remote shell
```

```
echo "We are running on: " `uname -sr`
```

```
# . /home/appqa/ivp50/snap/ntc/fmt_ntc_ckd ← script on Linux or
```

```
# . /home/appqa/ivp50/snap/ntc/fmt_ntc_std
```

```
dasdfmt -y -b 4096 -l TC03AF -f /dev/dasdd ← inline commands
```

```
dasdfmt -y -b 4096 -l TC03B0 -f /dev/dasde
```

```
dasdfmt -y -b 4096 -l TC03B1 -f /dev/dasdf
```

```
dasdfmt -y -b 4096 -l TC03B2 -f /dev/dasdg
```



CKD Device Low Level Format Output



CoZLauncher[I]: CoZAgent process (50332037) ended with RC=0

CoZLauncher[N]: 10.92.2.28 target command '<default shell>' ended

CoZLauncher[I]: CoZLauncher ended with RC=0

CoZAgent: root@LN164028.LSS.EMC.COM target program '/bin/bash' PID: 2749

CoZAgent: completed with RC=0

We are running on: Linux 2.6.32-71.el6.s390x

Finished formatting the device.

Rereading the partition table... ok

Finished formatting the device.

Rereading the partition table...



CKD DASD Formatted and Online to Linux



CoZAgent: root@LN164028.LSS.EMC.COM target program '/bin/bash' PID:4148

CoZAgent: completed with RC=0

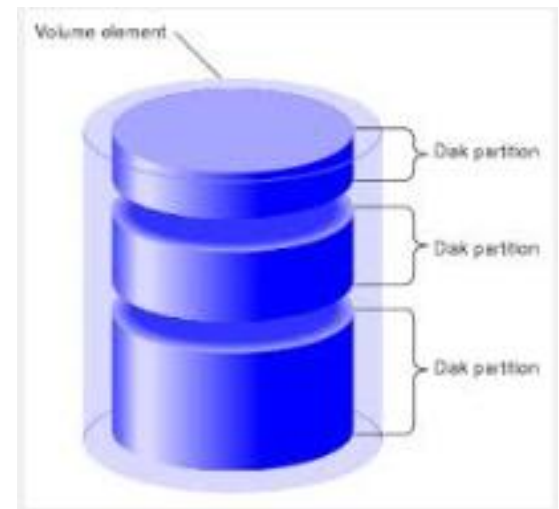
We are running on: Linux 2.6.32-71.el6.s390x

Bus-ID	Status	Name	Device	Type	BlkSz	Size	Blocks
0.0.0100	active	dasda	94:0	ECKD	4096	7043MB	1803060
0.0.0200	active	dasdb	94:4	FBA	512	256MB	524288
0.0.0201	active	dasdc	94:8	FBA	512	512MB	1048576
0.0.3755	active	dasdd	94:12	ECKD	4096	782MB	200340
0.0.3756	active	dasde	94:16	ECKD	4096	782MB	200340
0.0.3757	active	dasdf	94:20	ECKD	4096	782MB	200340
0.0.3758	active	dasdg	94:24	ECKD	4096	782MB	200340
0.0.3759	active	dasdh	94:28	ECKD	4096	782MB	200340
0.0.375a	active	dasdi	94:32	ECKD	4096	782MB	200340
0.0.375b	active	dasdj	94:36	ECKD	4096	782MB	200340
0.0.375c	active	dasdk	94:40	ECKD	4096	782MB	200340

Create CKD Partition Automation

Invoke shell script to format a list of dasd volumes

```
//COZLNCH.STDOUT DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//STDIN DD *  
# This is input to the remote shell  
echo "We are running on: " `uname -sr`  
./home/appqa/ivp50/snap/ntc/fdasd_auto
```



Create CKD Device Partition Shell Script

Shell script on Linux Server to loop through a device list and create partitions using fdasd command

```
[root@LN164028 ntc]# more fdasd_auto  
devlist=/home/appqa/ivp50/snap/ntc/ckd_devlist  
echo $devlist  
for disk in `cat $devlist`  
do  
    echo "$disk partition in process"  
/sbin/fdasd -a -k /dev/$disk &>/dev/null  
<< "    EOF"  
EOF  
done
```



CKD Device Partition Output

CoZLauncher[I]: CoZAgent process (16777637) ended with RC=0

CoZLauncher[N]: 10.92.2.28 target command '<default shell>' ended

CoZLauncher[I]: CoZLauncher ended with RC=0

CoZAgent: root@LN164028.LSS.EMC.COM target program '/bin/bash' PID: 3274

CoZAgent: completed with RC=0

We are running on: Linux 2.6.32-71.el6.s390x

reading volume label ...: VOL1

reading vtoc: ok

auto-creating one partition for the whole disk...

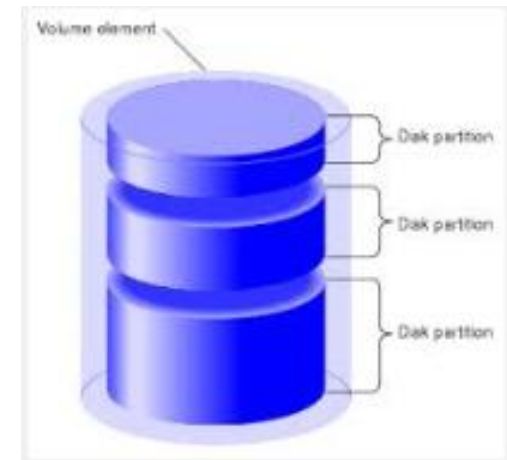
writing volume label...

writing VTOC...

rereading partition table...

reading volume label ...: VOL1

reading vtoc: ok



Make CKD File Systems

Invoke mke2fs command to create Linux File Systems

```
//COZLNCH.STDOUT DD SYSOUT=*  
//STDIN DD *  
# This is input to the remote shell  
echo "We are running on: " `uname -sr`  
#. /home/appqa/ivp50/snap/ntc/mke_ntc_ckd  
mke2fs -j /dev/dasdd1  
mke2fs -j /dev/dasde1  
mke2fs -j /dev/dasdf1  
mke2fs -j /dev/dasdg1  
  
//
```

Mount CKD Devices

Mount the formatted source devices units 3755-3758
attached as dasdd-dasdg

```
//COZLNCH.STDOUT DD SYSOUT=*
```

```
//STDIN DD *
```

```
# This is input to the remote shell
```

```
echo "We are running on: " `uname -sr`
```

```
#. /home/appqa/ivp50/snap/ntc/mnt_ntc_src
```

```
mount -t ext3 /dev/dasdd1 /home/sdasd/d01
```

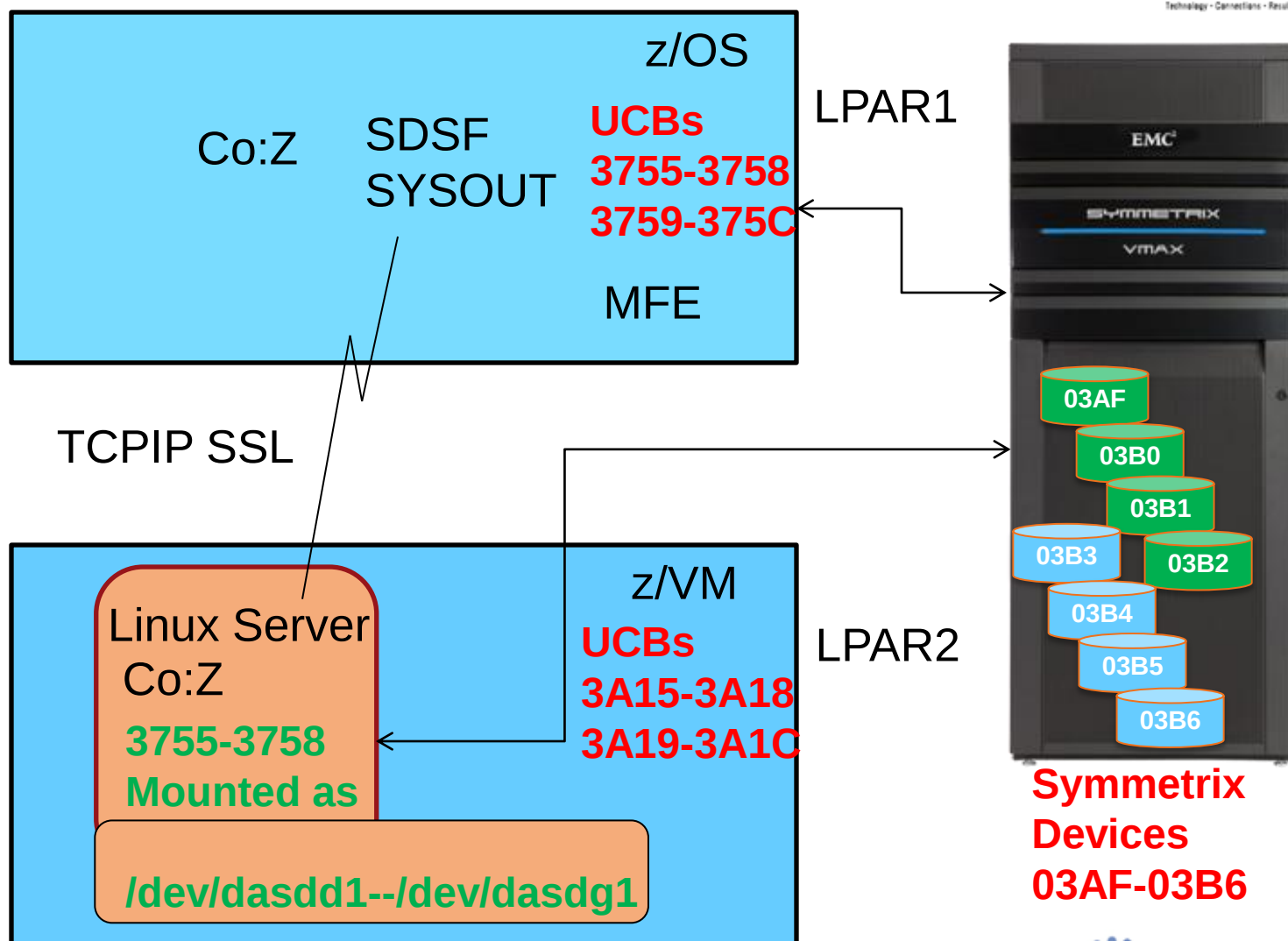
```
mount -t ext3 /dev/dasde1 /home/sdasd/d02
```

```
mount -t ext3 /dev/dasdf1 /home/sdasd/d03
```

```
mount -t ext3 /dev/dasdg1 /home/sdasd/d04
```

```
//
```

Device Configuration Mounted Volumes

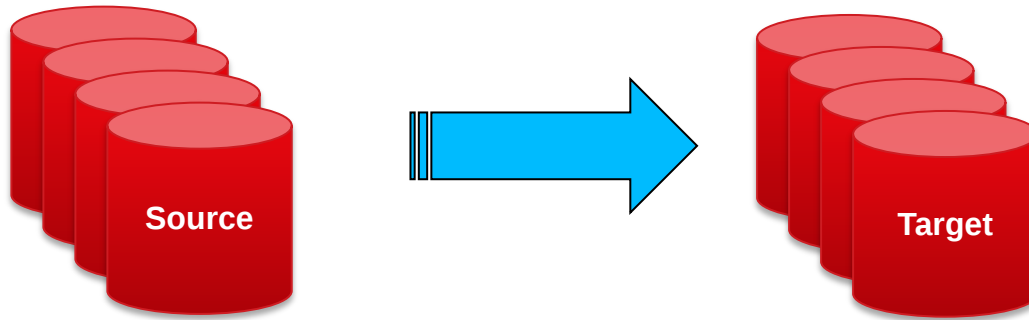


Consistent Backup Using Snap Volume

SNAP VOLUME (SOURCE (**UNIT(3755(4))**) -
TARGET (**UNIT(3759(4))**))
ACTIVATE(CONSISTENT(YES))

Source Volumes

Target Volumes



Source Devices Mounted

Source Devices – results from Co:Z Linux command `df -k`

CoZAgent: root@LN164028.LSS.EMC.COM target program '/bin/bash' PID:7911
 CoZAgent: completed with RC=0

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/dasda1	7098728	2272336	4465788	34%	/
tmpfs	252236	88	252148	1%	/dev/shm
/dev/dasdd1	788632	18456	730116	3%	/home/sdasd/d01
/dev/dasde1	788632	18456	730116	3%	/home/sdasd/d02
/dev/dasdf1	788632	18456	730116	3%	/home/sdasd/d03
/dev/dasdg1	788632	18456	730116	3%	/home/sdasd/d04

Backup Verification Process



Unmount Source Devices 3755-375B

Mount Target Devices Units 3759-395C

Run Verification Program against Target Devices

```
//COZLNCH.STDOUT DD SYSOUT=*
```

```
//STDIN DD *
```

```
. /home/appqa/ivp50/snap/ntc/umnt_ntc_src
```

```
. /home/appqa/ivp50/snap/ntc/mnt_ntc_trg
```

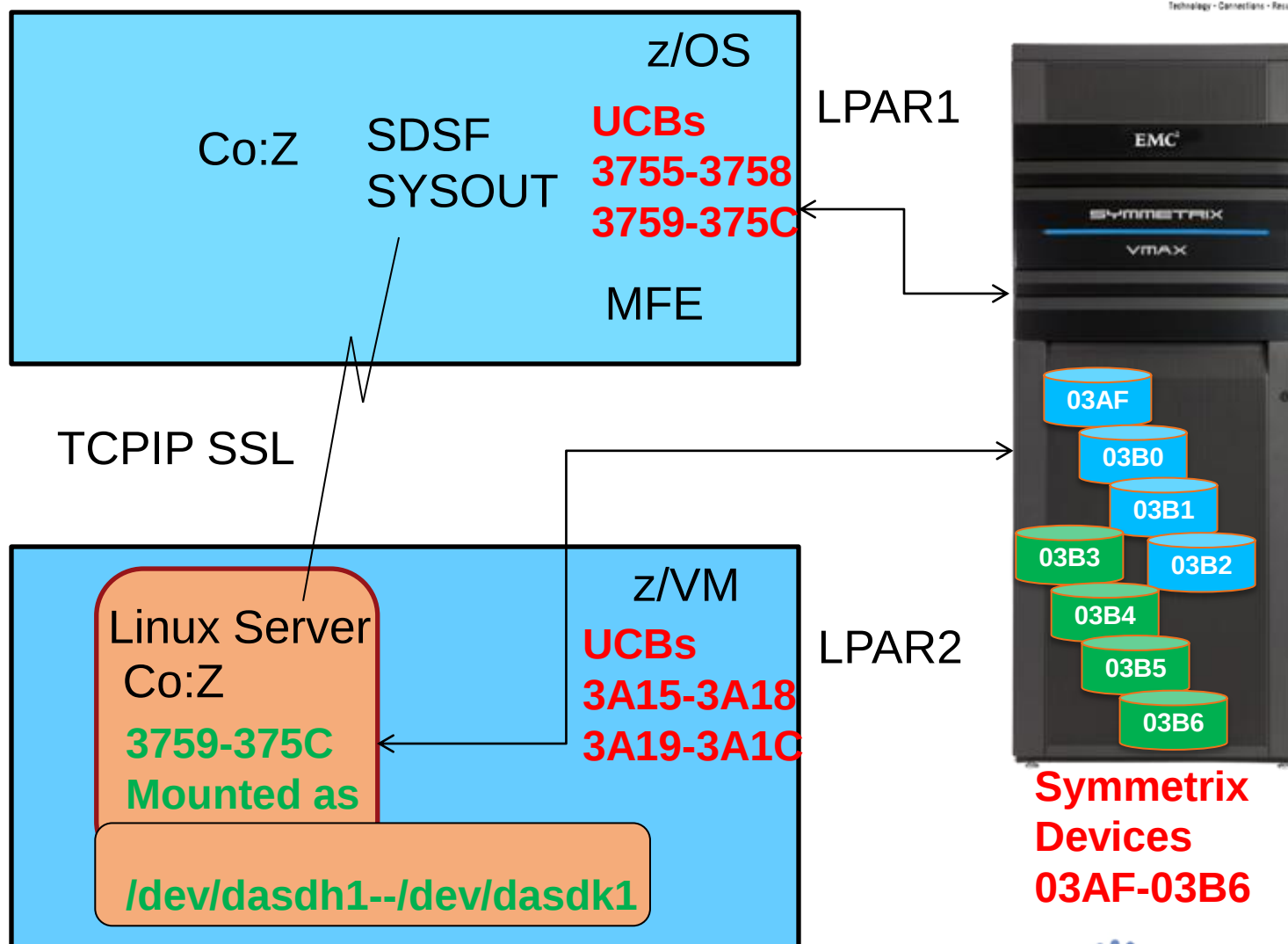
```
df -k
```

```
. /home/appqa/ivp50/verifpgm
```

```
. /home/appqa/ivp50/snap/ntc/umnt_ntc_trg
```

```
. /home/appqa/ivp50/snap/ntc/mnt_ntc_src
```

Device Configuration Mounted Target Volumes



Target Devices Mounted



Target Devices – df –k results from Co:Z

CoZAgent: root@LN164028.LSS.EMC.COM target program '/bin/bash' PID:7911

CoZAgent: completed with RC=0

Filesystem	1K-blocks	Used	Available	Use% Mounted on
/dev/dasda1	7098728	2267492	4470632	34% /
tmpfs	252236	88	252148	1% /dev/shm
/dev/dasdh1	788632	18456	730116	3% /home/sdasd/d01
/dev/dasdi1	788632	18456	730116	3% /home/sdasd/d02
/dev/dasdj1	788632	18456	730116	3% /home/sdasd/d03
/dev/dasdk1	788632	18456	730116	3% /home/sdasd/d04

Backup Verification

Co:Z results from **verifpgm**

CoZAgent: root@LN164028.LSS.EMC.COM target program
'/bin/bash' PID: 4313

CoZAgent: completed with RC=0

We are running on: Linux 2.6.32-71.el6.s390x

Verification Program

- Initializing resources.....
- Initialization complete.....
- Checking for consistency....
- Final result: **data is consistent**



Co:Z Launcher Opportunities Summary

- Linux Interoperability Test Scenario Configurations with One z/OS Job Stream
 - Remote server reboot
 - Remote device configuration
 - Remote script and command invocation
 - Local invocation of product tests
 - Remote verification of test outcomes at server
 - Remote server shutdown



Co:Z Launcher Opportunities Summary

- Invoke a script that already exists on a remote server
- Invoke a command to allow variables
- Linux command output is redirected to z/OS SDSF spool
- Free with Community License
- One operating system interface **z/OS!**



Reference Documentation

Dovetailed Technologies

<https://www.dovetail.com/>

Hybrid Batch Processing - YouTube Video

http://www.youtube.com/embed/WIZbN_vs7us

EMC Mainframe Enablers TimeFinder Clone Mainframe SNAP

https://support.emc.com/docu47019_Mainframe-Enablers-TimeFinder-Clone-Mainframe-Snap-Facility-Product-Guide.pdf?language=en_US

Practical Migration to Linux on System z

<http://www.redbooks.ibm.com/redbooks.nsf/searchsite?SearchView&query=linux>

Questions?



Co:Z® is a registered trademark of Dovetailed Technologies, LLC.
EMC2, EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries.

Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc.

z/OS® and z/VM® are registered trademarks of IBM Corporation

SSH® and Secure Shell® are registered trademarks of SSH Communications

All other trademarks used herein are the property of their respective owners.



Thank You!

