

 #SHAREorg



Virtual Linux Server Disaster Recovery Planning

Rick Barlow
Nationwide Insurance

August 2012
Session 11826



Agenda

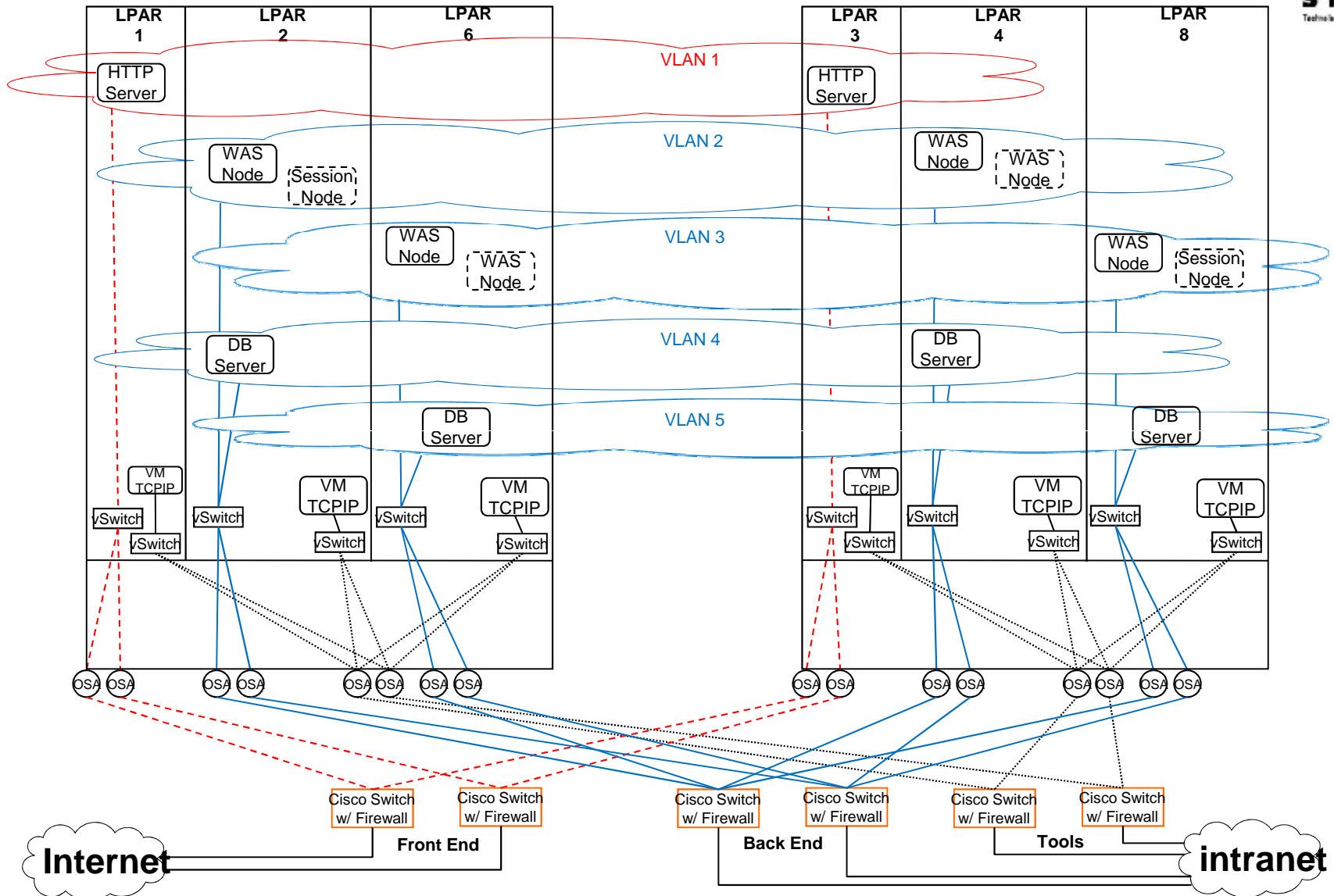
- Definitions
- Our Environment
- Business Recovery Philosophy at Nationwide
- Planning
- Execution

Definitions

- High Availability
 - “With any IT system it is desirable that the system and its components (be they hardware or software) are up and running and fully functional for as long as possible, at their highest availability. The most desirable high availability rate is known as “five 9s”™, or 99.999% availability. A good deal of planning for high availability centers around backup and failover processing and data storage and access.”
 - Deal with significant outage within data center
 - LPAR failure
 - Operating System outage
 - Application ABEND

High Availability

Production



Definitions

- Disaster Recovery
 - “Disaster recovery in information technology is the ability of an infrastructure to restart operations after a disaster. While many of today's larger computer systems contain built-in programs for disaster recovery, standalone recovery programs often provide enhanced features. Disaster recovery is used both in the context of data loss prevention and data recovery.”
 - Deal with complete outage
 - Natural catastrophe
 - Data center
 - Major hardware failure

Our Environment

- Four z196 installed in 2011

Current configuration:

- Production boxes
 - 36 IFLs – 18 on each z196
 - 568GB memory
 - 6 z/VM LPARs (plus system programmer configuration test)
 - Tier 4+ data center
 - Fully redundant power, telecom, generators, etc
- Development boxes
 - 21 IFLs – 10 on one z196 and 11 on the other
 - 665GB memory
 - 6 z/VM LPARs (plus system programmer test)

Planning

- Design
- Challenges
- Priorities
- Setup
- Automation
- Documentation
- Teamwork

Design

- What
 - Identify what needs to be recovered
 - Everything or subset?
 - Priorities – recovery order
- When
 - Need to know recovery objectives
- Where
 - Identify where recovery will occur
 - Second site
 - Vendor site
- How
 - Identify how to transfer programs and data
 - Identify how to perform recovery

Challenges

- Production configuration changes may require DR configuration changes
 - Processor model changes (not necessarily)
 - Capacity
 - Network configuration
 - Transition
- Consistent distribution and synchronization of boot and DR scripts
- Commitment to regular testing
- DR Infrastructure \neq Application DR

Priorities

- Prioritizing your application recovery must be done by the people in your organization that understand the business processes.
- Business requirements that drive recovery time-frame
 - regulated
 - financial / investments
 - responsive to customers

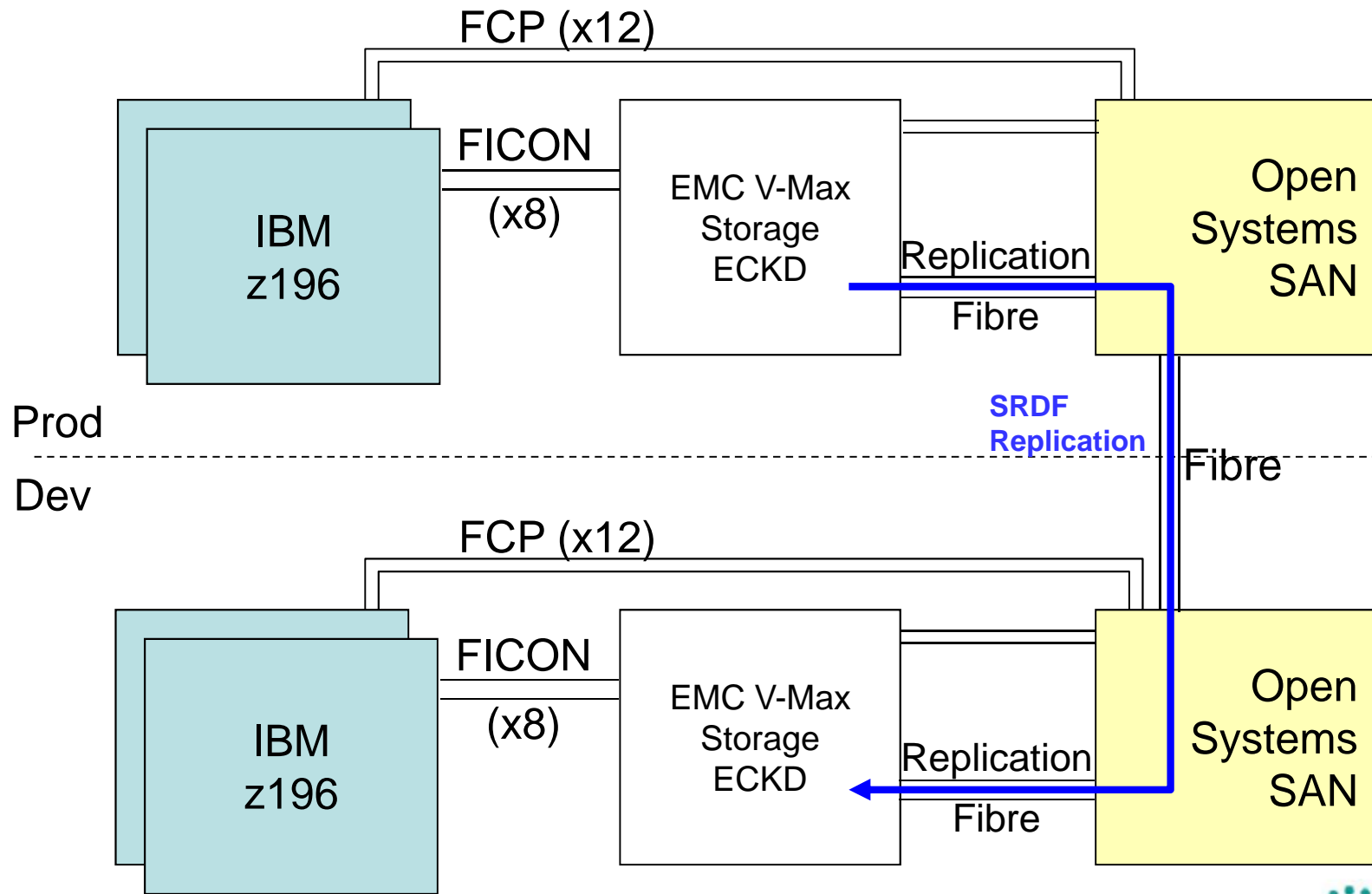
Setup

- Asynchronous replication with SRDF between production and recovery sites
- Replicated volumes at recovery site
 - z/VM different unit address between prod and dev/test/dr
 - SAN target WWPN and LUN differ between production and recovery
 - Changes for recovery are automated in Linux script run at boot
- Manual processes
 - Initiate Clone copies
 - Vary ECKD DR volumes online
 - Start Linux servers
 - Update DNS to reflect DR IP addresses for all servers
 - Middleware and application hard-coded parameters (e.g. IP addresses)

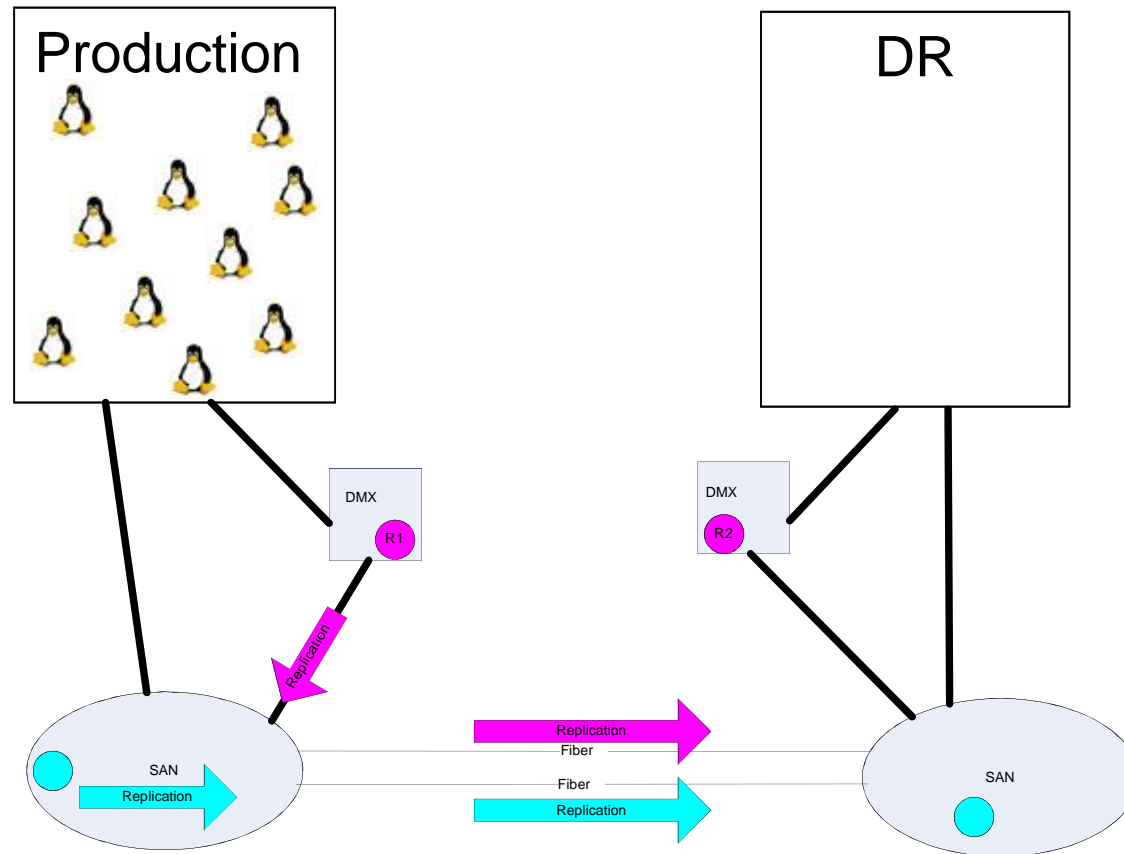
Setup

- The DR process would be the same for the following failures
 - System z failure
 - DASD / SAN storage frame failure
- Long distance (inter-data center) fiber failure
 - No DR required, (redundant links installed)

Setup

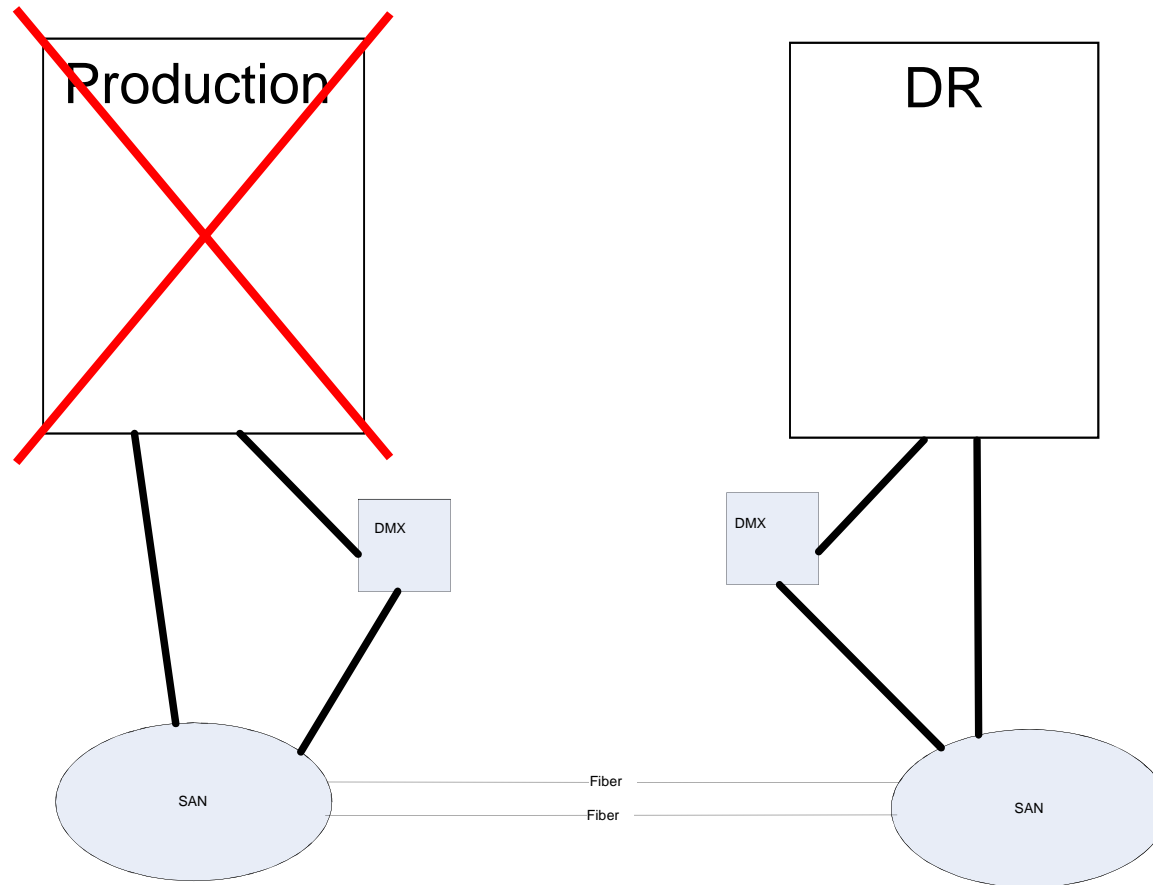


Normal Production



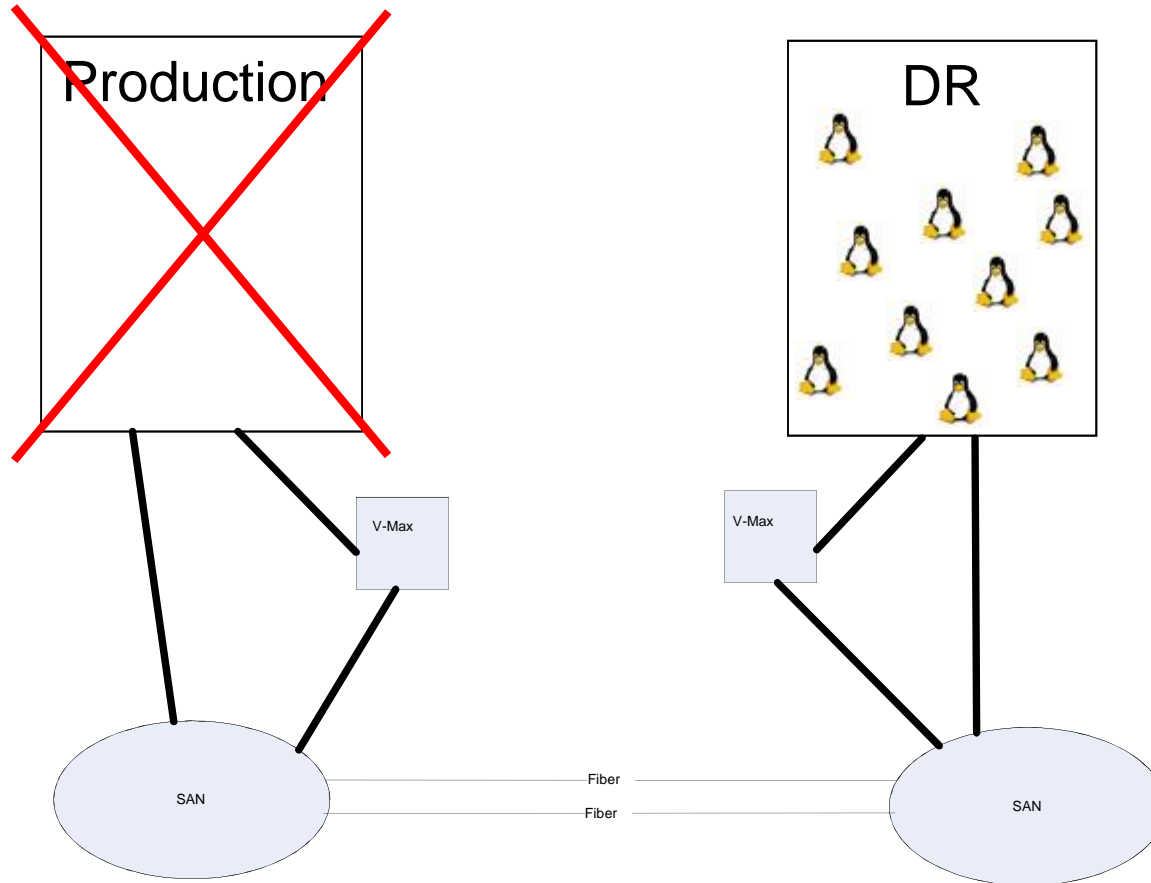
Failure Happens

If failure occurs:
manually stop
replication; initiate
DR Clones; Bring
ECKD volumes on-
line at DR site, start
Linux servers



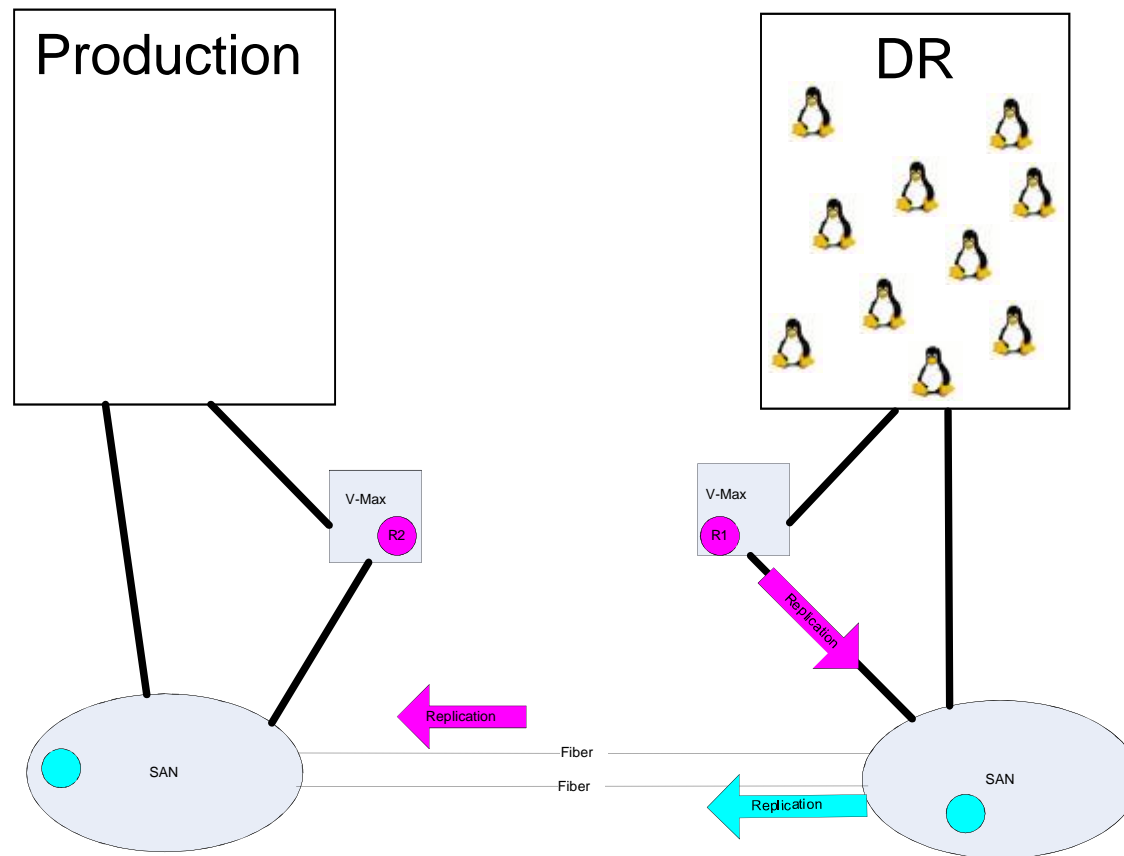
Recovery Begins

Servers identify DR configuration, change IP address, change SAN parameters, register new IP with DNS, start replication south to north.



Recovery Begins

Servers identify DR configuration, change IP address, change SAN parameters, register new IP with DNS, start replication south to north.



Automation

- Avoid manual processes
 - Dependence on key individuals
 - Prone to mistakes
 - Slow
- Automated processes
 - Requires only basic knowledge of environment and technologies in use
 - Accuracy
 - Repeatable
 - Faster
 - Does not mean build it once then ignore;
Requires regular review and updates

Automation

- Automation begins at provisioning
 - DR configuration stored with production configuration
 - CMS NAMES file
 - Contains all information about provisioned server
 - Copy stored on DR disk also
 - Also used to generate report of server definitions for easy lookup
 - Linux PARM file stored on CMS disk
 - Stored on disk accessible at boot time
 - Copy stored on DR disk also
 - Define everything needed to provision server and at boot time

Automation

- Extract from LINUX NAMES file for one guest

```
:nick.WS001
:userid.PZVWS001
:node.VN2
:desc.Prod web server 1
:env.PROD
:hostname.PZVMWS001
:load.LINUXWS
:ip.10.1.1.1
:vswitch.PRODVSU1
:vlan.2102
-----
:ip_nb.10.2.1.1
:vsw_nb.NETBKUP1
:vlan_nb.3940
```

```
:ip_dr.10.221.1.1
:vsw_dr.PRODVSU1
:vlan_dr.2102
:ip_drbu.10.222.1.1
:vsw_drbu.NETBKUP1
:vlan_drbu.3940
:oth_ip.10.1.1.5 10.1.1.15
:dr_oth_ip.10.221.1.5 10.221.1.15
-----
:status.2005-09-08
:gold.V1.2
:memory.256M
-----
:cpus.1
:share.200,LS-200
:comments.
:storage.2.3G
:storage_os.7.1G
:bootdev.251
```

Automation

- Extract from LINUX NAMES file for one guest (cont'd)

```
:storage_san.16.86G
```

```
:sanluns.R1:0100:5006048AD52D2588:0059000000000000:8.43 R1:0100:5006048AD52D2588  
:005A000000000000:8.43 R1:0200:5006048AD52D2587:0059000000000000:8.43 R1:0200:50  
06048AD52D2587:005A000000000000:8.43
```

```
:storage_san_dr.16.86G
```

```
:sanluns_dr.R2:0100:5006048AD52E4F87:006A000000000000:8.43 R2:0100:5006048AD52E4  
F87:006B000000000000:8.43 R2:0200:5006048AD52E4F88:006A000000000000:8.43 R2:0200  
:5006048AD52E4F88:006B000000000000:8.43
```

Automation

- PARM file

```
HOST=pzvmws001
ADMIN=10.1.1.1
BCKUP=10.2.1.1
DRADMIN=10.221.1.1
DRBCKUP=10.222.1.1
ENV=PROD
DRVIP=10.1.1.5,10.1.1.15
BOOTDEV=251
VIP=10.221.1.5,10.221.1.15
SAN_1=0100:5006048AD52D2588:005A000000000000,0200:5006048AD52D25
87:005A000000000000
SAN_2=0100:5006048AD52D2588:0059000000000000,0200:5006048AD52D25
87:0059000000000000
SAN_3=0100:5006048AD52E4F87:006A000000000000,0200:5006048AD52E4F
88:006A000000000000
SAN_4=0100:5006048AD52E4F87:006B000000000000,0200:5006048AD52E4F
88:006B000000000000
```

Automation

- Alternate Start-up Scripts
 - Identify production or DR mode
 - VMCP – interact with CP
 - CMSFS – read CMS files
 - Set parameters for environment
 - Hostname to /etc/HOSTNAME
 - IP addresses to /etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.xxxx
 - SAN LUN information
 - Color prompt by environment
 - Prod = Red
 - DR = Yellow
 - Tools = Green

Automation

- Extract from boot.config

```
# Setup variables
echo "1" > /sys/bus/ccw/devices/0.0.0191/online
sleep 5
# modprobe required just in case
modprobe cpint
PARMDEV=`grep 191 /proc/dasd/devices|awk '{print $7}'` NZVWS001 AT VN1
QUSERID=`hcp query userid`
GUEST=`echo $QUSERID|cut -d" " -f 1`
LOCO=`echo $QUSERID|cut -c14`
LPAR=`echo $QUSERID|cut -c13-15`
BOX=`echo $QUSERID|cut -c1`
cmsfscat -d /dev/$PARMDEV -a ${GUEST}.PARMFILE > /tmp/sourceinfo
. /tmp/sourceinfo

echo "0" > /sys/bus/ccw/devices/0.0.0191/online
```


Automation

- Result of cmsfscat

```
cat /tmp/sourceinfo
```

```
HOST=pzvmws001
```

```
ADMIN=10.1.1.1
```

```
BCKUP=10.2.1.1
```

```
DRADMIN=10.221.1.1
```

```
DRBCKUP=10.222.1.1
```

```
ENV=PROD
```

```
DRVIP=10.1.1.5,10.1.1.15
```

```
BOOTDEV=251
```

```
VIP=10.221.1.5,10.221.1.15
```

```
SAN_1=0100:5006048AD52D2588:005A000000000000,0200:5006048AD52D2587:005A000000000000
```

```
SAN_2=0100:5006048AD52D2588:0059000000000000,0200:5006048AD52D2587:0059000000000000
```

```
SAN_3=0100:5006048AD52E4F87:006A000000000000,0200:5006048AD52E4F88:006A000000000000
```

```
SAN_4=0100:5006048AD52E4F87:006B000000000000,0200:5006048AD52E4F88:006B000000000000
```

Automation

- More extract from boot.config

```
case "$ENV" in
  PROD)
    if [ "$LOCO" == "$BOX" ]
      then
        CLR="41"; #Red
      else
        CLR="43"; #Yellow/Gold
        ENV="DR";
      fi
    ;;
  DEV | JT | TOOLS | TOOL)
    CLR="42"; #Green
  ;;

```

```
ST)
  CLR="44"; #Blue
  ;;
PT)
  CLR="45"; #Purple
  ;;
UAT | IT)
  CLR="46"; #Turq
  ;;
*)
  CLR="42"; #Green
  ENV="UNK";
  ;;
esac
```

Examples:

```
barlowr@szvmjt002:JT:barlowr>
barlowr@nzvmws001:PROD:barlowr>
```

Documentation

- Document everything
 - Declaration criteria
 - Contact information (your list may be different)
 - Operating System
 - Middleware
 - Application
 - Network
 - Security
 - Lists of servers
 - Recovery process
 - Verification process
 - Fail-back process

Documentation

- DR Procedure example:
 - Confirm DISASTER declaration
 - Begin shutdown all test/development guests to insure sufficient capacity.
 - Bring up production DR guests identified by business units for each application environment.
 - Make appropriate emergency DNS changes to point users to DR environment per definitions for each application environment.

Documentation

- Return Procedure example:
 - Confirm DISASTER OVER declaration
 - Reverse disk replication; confirm synchronization
 - Follow instructions for confirmation of original production environment for each application.
 - Bring down DR guests identified by business units for each application environment.
 - Make appropriate DNS changes to point users to non-DR environment per definitions for each application environment.
 - Resume normal disk replication.

Teamwork

- Recovery coordinator
- z/VM System Programmers
- Linux System Administrators
- Middleware
 - WAS Administrators
 - Database Administrators
 - MQ Administrators
- Application Teams
 - Testing methodology
 - Expected results

Avoid processes that are dependent on subject matter experts (SME) when a disaster happens

Execution

- Test
- Document results
- Compare to plan
- Repeat

Execution

- Where...
 - ... to recover the systems
 - Your own second site
 - A recovery vendor
 - ... do the people go
 - Identify what personnel need to travel to recovery site
 - Document travel procedures
 - Identify alternate (local) office space
 - Some office locations may be able to access recovery site if connectivity is available

Execution

- Testing
 - Test as often as feasible
 - Frequency may depend on having your own site or contracting with a vendor
 - Tests should be as close as possible to real recovery conditions
 - Operating systems are easy
 - Some subsystems are **not** so easy (e.g. large database)
 - Multi-platform applications ~~can be~~ are more complex
 - Automate as much as possible to avoid manual effort

Document Results Compare to Plan



- Detailed plans for all test scenarios
- Carefully track tests
- Document action items and follow up for improvements
- Build on successes

Repeat

- Do it again
- Do it regularly
- Corporate emphasis may be required to *encourage* all applications to test

Contact Information

“And I thought we were busy *before* Linux showed up!”



Rick Barlow

Senior z/VM Systems Programmer

Phone: (614) 249-5213

Internet: Richard.Barlow@nationwide.com