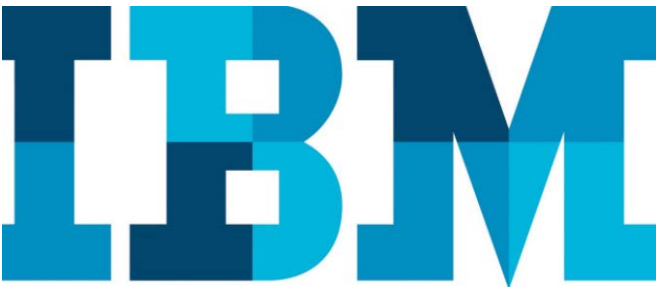
September 2020

How to inject failure: IBM VM Recovery Manager HA and DR

Simulate failure in HA, DR, and HADR cluster types for testing a solution before deployment

Table of contents



Overview

In this paper, we have described how to inject failures and how KSYS will react to those failures. Before deploying production VMs, users might face challenges in testing the product and simulating DR failure. This paper shows simulation of DR failure with respect to all the KSYS component such as HMC, Storage, VIOS, disk, application, and so on.

Introduction

IBM VM Recovery Manager provides a high availability (HA) and disaster recovery (DR) solution for virtual machines (VM) during partial and complete site failures. In case of down time or planned outage, users can migrate the VM from one host (server) to another host within a home site or to a backup site, depending on partial or complete site failure. VM Recovery Manager HADR provides an automated solution in case of partial site failures. This paper show failure injection techniques to simulate a disaster situation for testing a product before deploying the VMs for HA, DR, and HADR cluster types.

VM Recovery Manager provides the HA solution in case of application, VM, Virtual I/O Server (VIOS), and host failures and provides the DR solution in case of host, Hardware Management control (HMC), and storage failures. In case of application, VM, VIOS, and host failures, KSYS subsystem notifies the user through event notification or notify scripts and then automatically shuts down the VM in the current host and restarts the VM in another host in the host group.

KSYS provides two restart policies for users at host group levels: *advisory mode* and *auto mode* (default value is auto mode).

Auto mode: In case of HA failures, KSYS notifies the user and then migrates the VMs automatically from one host to another host within a host group.

Command to check restart progress during auto failures:

```
ksysmgr query system status (this command will show what
operation is in progress)
ksysmgr query system status monitor=yes (this command
will show the progress)
```

Advisory mode: In case of HA failures, KSYS notifies the user and the user can manually review the problem and restart the VMs within a site or across sites.

VM Recovery Manager HADR subsystems

The VM Recovery Manager HADR solution uses the following subsystems for cluster configuration:

- Controller System (KSYS)
- Site
- HMC
- Host
- VIOS and host monitor subsystem
- VMs or logical partitions (LPARs) and VM agent
- Storage
- Network

Refer to the article "Deployment steps for IBM VM Recovery Manager HADR cluster" for cluster deployment

Types of failure

This section describes the failure types handled by VM Recovery Manager HADR.

Types of application status:

- Green/Yellow: Healthy state
- Red: Failure state
- Gray: Not monitored state

Application failure

The VM agent uses the application HA monitoring framework to monitor the health of the application periodically by running the application-specific monitor scripts, by identifying whether the application has failed and by identifying whether the VM must be restarted in the same host or another host.

After you add an application, if the application fails or stops working correctly, the VM agent attempts to restart the application in the same VM for a number of times as specified in the `max_restart` attribute for the VM, which is set to 3 by default. If the application is still not working correctly, the KSYS subsystem notifies the user about the issue.

There are two types of applications: critical and non-critical. When an application is critical, if the application fails or stops working correctly, the VM agent attempts to restart the application. If the application is still not working correctly, the KSYS subsystem notifies the user about the issue and attempts to reboot the VM. If the application is still in incorrect state and its status is displayed as RED, the KSYS restarts the VM on another host within the host group. If the application is still not working correctly on all the other hosts, then the application will be marked as permanent failure. When an application is non-critical, if the application fails or stops working correctly, the VM agent attempts to restart the application and if it is still not in proper state, then KSYS leaves it in the same state.

VM failure

Calculating VM failure detection speed:

Fast: hostFDT + VMthreshold

Normal: hostFDT + VMthreshold*2

Slow: hostFDT + VMthreshold*3

If the operating system of a VM is not working correctly, or if the VM has stopped working because of an error, it is considered as VM failure. KSYS will wait for the number of missed heartbeats based on the `vm_failure_detection_speed` attribute of a VM and then notify the user and restart the VM on another host within the host group. The default value of `vm_failure_detection_speed` is `normal`. Users can change it to `slow` or `fast` using the `modify` command. The KSYS subsystem uses the VM monitor module to monitor the heartbeat from the VM to the host monitor subsystem in a VIOS.

Commands to change vm_failure_detection_speed at the VM, host, and host group levels:

```
ksysmgr modify vm <vmname>
vm_failure_detection_speed=<fast|normal|slow>
ksysmgr modify host <hostname>
vm_failure_detection_speed=<fast|normal|slow>
ksysmgr modify host_group <hgname> options
vm_failure_detection_speed=<fast|normal|slow>
```

VIOS failure

If the VIOS is in error state or in down state, it is considered as VIOS failure. If all the VIOS instances are down for a host then KSYS notifies the user and restarts the VMs belonging to that host to another host within the host group. If a few VIOS instances are down and the VM not in proper state, then the VM will not migrate automatically. KSYS will notify the user for VIOS failure and the user have to trigger restart on those VMs or on the host belonging to that VIOS.

Host failure

If a host is in the error state or in the down state, or all the VIOS instances of a host are down, it is considered as host failure. KSYS shuts down the VMs in the current host and restarts the VM from another host within the host group. If all the home site hosts belonging to a host group are down, then users can migrate the VMs manually to another site hosts within a host group. The KSYS subsystem will notify the user according to the host_failure_detection_time (hostFDT) value. The default value of host_failure_detection_time is 90 seconds. You can change the set the value between 90 and 600.

Command to change the value of host_failure_detection_time:

```
ksysmgr modify host_group <hgname> options
host_failure_detection_time=<90-600>
```

There are two MonitorMode types for VIOS:

- GLOBAL is the mode when disks and the VIOS instances are in the operative state.
- LOCAL is the mode when pool disks or the VIOS instances are in an erroneous state.

Disk failure

When the host monitor nodes lose network connectivity or lose access to the pool disk, all the VIOS instances operate in the LOCAL mode. If a VIOS instance is running in the LOCAL mode, the MonitorMode field is set to LOCAL. If the host monitor is not operating correctly, the MonitorMode field is set to DOWN.

Command to check monitor mode from KSYS:

```
lsrsrc IBM.VMR_VIOS Name ViosUuid CecUuid MonitorMode
```

Command to check pool disk:

```
(0) root @ rt07v1: /
# pooladm pool list
Pool Path
------------------------------
/var/vio/SSP/KSYS_HA_DR_1_1/D_E_F_A_U_L_T_061310
(0) root @ rt07v1: /
# pooladm pool lsdisk /var/vio/SSP/KSYS_HA_DR_1_1/D_E_F_A_U_L_T_061310


Pool:               /var/vio/SSP/KSYS_HA_DR_1_1/D_E_F_A_U_L_T_061310
Device Path
------------------------------
/dev/hdisk3
```

HMC failure

If HMC is unreachable then KSYS notifies the user and the user can manually migrate the VMs to a host in another site within a host group.

Storage failure

If storage is unreachable then KSYS notifies the user and the user can manually migrate VMs to another site hosts within a host group.

Description of file sets used for failure injection

The test team has used different types of file sets for failure injection and you can find these files in GitHub. The following sections provide details of the directories containing file sets.

APP directory

APP directory in mentioned file sets contains three scripts: `monitor_1`, `start_1`, and `stop_1` which will be configured at application level. These scripts will help in monitor, start and stop of an application. These file sets have been used during APP failure

Users can use the `modify ksysvmmgr` command to configure these scripts in a VM as shown in this section.

```
ksysvmmgr -s modify app  ziti030_app1 monitor_script=/apps/monitor_1 start_script=/apps/start_1
stop_script=/apps/stop_1
```

```
(0) root @ ziti005: /
# ksysvmmgr q app
Application name=ziti005_app1
 monitor_script=/apps/monitor_1
 stop_script=/apps/stop_1
start_script=/apps/start_1
```

VM CRASH

The CRASH directory contains two subdirectories, crash_vm and fail_io. The crash_vm directory contains the file sets related to loading and unloading the kernel extension along with the crash_vm scripts. We have used these file sets during VM failure. The fail_io directory contains file sets that crashed the disk in a VIOS instance or a VM. The test team used the crash_vm and fail_io file sets during testing to simulate a disk failure scenario.

HANG

This directory contains the syshang subdirectory, which in turn contains the file sets related to VM hang. The test team used these files during VM failure.

How to unload and load kernel extension file sets

After the completion of testing, users can unload the file that was earlier loaded (with `lke`) using the following command:

```
ulke <kmid from load>
```

For example:

```
(0) root @ ziti030: /syshang
# lke ./syshang_kext
        a0269000
(1) root @ ziti030: /syshang
# ulke a0269000               #### here use same id which we will get during load of kernel
extention
unload done
```

How to inject failure

The following sections explain how to inject failure at different levels for different kind of failures.

Failure in VM for application failure

Users can add a monitor script at the application level in a VM to monitor the health of an application. If any failure occurs in a monitor script, then the VM agent considers it as application failure. For an application failure, KDB kernel debugger should be disabled for a VM. If the KDB kernel debugger is enabled the VM enters a crash state during VM reboot, which is considered as a VM failure.

For example (monitor script):

```
#!/bin/sh
fName=$0
extn='.log'
if [ -f "/apps/monitor_1_1" ]
then
exit 1
```

In this monitor script, if the user creates a file with the name, *monitor_1_1*, the application will be considered to be in the failure state and application status in a VM will change to *FAILURE (RED)* and in KSYS it will change to *RED.*

Events related to app failure:

```
APP_FAILURE_DETECTED
VM_REBOOT_FOR_APPLICATION_FAILURE
VM_RESTART_FOR_APPLICATION_FAILURE
VM_APPLICATION_PERMANENT_FAILURE
```

Failure in VM for VM failure

You can crash the kernel debugger to fail VM. Refer to the commands to find how to enable and disable the kernel debugger in an IBM AIX® VM.

To enable the KDB kernel debugger:

```
bosdebug -D -M
bosboot -aD
bosdebug -L
```

To disable the KDB kernel debugger:

```
bosdebug -o
bosboot -a
shutdown -Fr
```

How to crash an AIX VM:

You can use the following commands to crash the VM.

Crash_vm file sets are in GitHub under the CRASH directory.

```
cd crash_vm
./lke crash_vm_kext
./crash_vm -c
```

How to hang an AIX VM:

You can use the following commands to hang an AIX VM. The syshang file sets are in GitHub under the HANG directory.

```
lke ./syshang_kext
./syshang
```

How to crash the LINUX VM:

You can use the following commands to force the Linux kernel to crash.

```
(0) root @ ziti001: /root
# systemctl disable kdump
Removed symlink /etc/systemd/system/multi-user.target.wants/kdump.service.

(0) root @ ziti001: /root
# systemctl stop kdump.service
```

```
(0) root @ ziti001: /root
# /bin/systemctl status kdump.service
â kdump.service - Crash recovery kernel arming
Loaded: loaded (/usr/lib/systemd/system/kdump.service; disabled; vendor preset: enabled)
Active: inactive (dead) since Thu 2018-11-01 01:39:20 CDT; 3s ago

Process: 8407 ExecStop=/usr/bin/kdumpctl stop (code=exited, status=0/SUCCESS)
Main PID: 1317 (code=exited, status=0/SUCCESS)

(0) root @ ziti001: /root
# sysctl -w kernel.panic="0"
kernel.panic = 0

(0) root @ ziti001: /root
# echo 1 > /proc/sys/kernel/sysrq

0) root @ ziti001: /root
# echo c > /proc/sysrq-trigger
```

How to hang a Linux VM:

Use the following steps to hang a Linux VM.

1. Configure kgdboc

```
echo ttyS0 > /sys/module/kgdboc/parameters/kgdboc
```

2. Stop kernel execution (break into the debugger)

```
echo g > /proc/sysrq-trigger
```

Event related to VM failure:
VM_FAILURE_DETECTED

For example:

Failing a VM using the scripts provided in GitHub.

```
(0) root @ ziti032: /
# cd crash_vm; ./lke crash_vm_kext; ./crash_vm -c
        a0a85000
```

Use the following command to check the reference code of a VM (in this case, the VM is in crash state)

```
(0) root @ ksys204: /
# ssh hscroot@e17vhmc5 lsrefcode -m ragu-9119-MME-SN106CDC7 -r lpar -F lpar_name:refcode | grep
ziti032
ziti032:0c20
```

KSYS will provide the event for VM failure in /var/ksys/events.log

```
---------EVENT START--------
VM_FAILURE_DETECTED event has occurred. Details are as follows:
Event:              VM_FAILURE_DETECTED
Type:               Critical Error Event
Time:               Mon Feb 10 04:51:26 CST 2020
Entity Affected:    VM
Resource Affected:  VM : ziti032 UUID 04E86341-E7C0-45DD-B903-CF9A1CEB7E2C
Description:        0000-187 VM: ziti032 failure detected.

---------EVENT END----------
```

Use the following command to check the operation progress.

```
(0) root @ ksys204: /
# ksysmgr q system status monitor=yes
Restart in progress for Host_group HG1
        Stopping HA monitoring for VM ziti032
        HA monitoring for VM ziti032 stopped
        Shutdown has started for VM ziti032
        Shutdown has completed for VM ziti032
        Restart has started for VM ziti032
        Starting HA monitoring for VM ziti032
        HA monitoring for VM ziti032 started
        Restart on Target host sock_8286-42A-2182C5V has completed for VM ziti032
        Configuration cleanup started for VM ziti032
        VM monitoring for VM ziti032 started
        Configuration cleanup completed for VM ziti032
1 out of 1 VMs have been successfully restarted
```

HOST and VIOS Failure

For VIOS failure, users can power down VIOS instances and for host failure all VIOSes belong to that host should be down or Host should be in unreachable state. User can power down host from hmc or power off VIOS for host failure.

HMC CLI command to power off/on Host and VIOS

Power on VIOS: `chsysstate -m <managed_systemname> -r lpar -o on -n <vios_name>`
Power off VIOS: `chsysstate -m <managed_systemname> -r lpar -o shutdown --immed -n <vios_name>`
Power on Host: `chsysstate -m <managed_systemname> -r sys -o on`
Power off Host: `chsysstate -m <managed_systemname> -r sys -o off`

Events related with VIOS and host failure:
VIOS_FAILURE
ALL_VIOS_UNREACHABLE
VIOS_NODE_FAILURE
HOST_FAILURE
HOST_FAILURE_DETECTED

Disk failure

For disk failure, users can remove the pool disk from all VIOS instances from storage. After the pool disk is removed, the VIOS monitor mode will be changed to the LOCAL mode. In the LOCAL mode, users can trigger discovery, verify, LPM, restart, and perform DR move. If VM failure or application failure happens when VIOS is in LOCAL mode, KSYS will take actions accordingly. But in case of host failure, KSYS will notify the user through event and the user should trigger the *restart host* operation.

You can use the following files to move VIOS instances in the LOCAL mode without removing disk from storage. Files are provided in GitHub. Before running the following command, check which disk is the pool disk.

Command to load kernel extension:

```
# ./lke fail_io_kext
       a05bf000
```

Command to fail disk:

```
# ./fail_io -e /dev/<pool_disk_name>
  I/O fail ON
```

Command to recover disk after testing is complete:

```
# ./fail_io -d /dev/<pool_disk_name>
  I/O fail OFF
```

Event related to disk failure:
VIOS_MONITOR_MODE

HMC failure

For failing HMC, users can power off HMC or if the IP address of HMC is unreachable, it is considered as HMC failure.

You can use the following commands to make HMC unreachable without failing HMC. But, this command is only for testing purposes, and it is not a functionality. For this command to work, you should use a non-pingable IP.

```
stopsrc -s IBM.VMR -c (this command will stop the KSYS daemon)
startsrc -s IBM.VMR -e "SKIP_HMC_IPCHECK=1" (this command will start KSYS daemon)
chrsrc -s 'Name="<hmc_name>"' IBM.VMR_HMC HmcIP=<new_ip> (this command will change HMC IP, give
new_ip as non pingable ip)
```

To revert IP back:

```
chrsrc -s 'Name="<hmc_name>"' IBM.VMR_HMC HmcIP=<original_ip>
```

Event related with HMC failure:
HMC_UNREACHABLE

Storage agent or storage failure

For failing storage or storage agent, you can switch off the storage agent or if the IP address of the storage is unreachable, it is considered as storage failure.

You can use the following commands to make storage unreachable without failing storage or the storage agent. But this command is only for testing purposes, and it is not a functionality. For this command to work, you should use a non-pingable IP.

```
stopsrc -s IBM.VMR -c (this command will stop the KSYS daemon)
startsrc -s IBM.VMR -e "SKIP_SA_IPCHECK=1" (this command will start KSYS daemon)
chrsrc -s ' SAname="<storage_agent_name>"' IBM.VMR_SA ipAddr=<new_ip> (this command will change
Storage Agent original IP to new IP, in new_ip give non pingable ip)
```

To revert IP back:

```
chrsrc -s 'SAname="<storage_agent_name>"' IBM.VMR_SA ipAddr=<original_ip>
```

Event related with storage failure:
STG_UNREACHABLE

Recommended migrations during failure

This section describes the migrations that are recommended during failure. Trigger a restart when partial site is in the down state and VMs can migrate within a site. In case of host failure within a site, restart will be triggered automatically from one host to another host. After the failed host becomes active auto cleanup will happen. In case all the hosts on the source site are down, triggering an unplanned move will only migrate the VMs to the target site. In such cases, users must clean up the VMs manually.

Restart

If the `restart_policy` attribute is set to advisory mode or if few VIOS instances are down in a host, users can restart the VM by using the `ksysmgr restart` command. Restart is supported at the VM level and the host level. Use the following command to trigger the restart of VMs or hosts within a site. Here, `to` is optional. Restarting the host will migrate all managed VMs belonging to that host to another home site host within a host group.

```
ksysmgr restart vm <vm_name1,vm_name2,..> to=<target_host_name>
```

```
ksysmgr restart host <host_name> to=<target_host_name>
```

Unplanned move

If all hosts are down or HMC is unreachable or storage is unreachable, then users can directly migrate the VMs from the home site to the backup site using the ksysmgr unplanned move command. Unplanned move is supported at the host group and at the site level. If an unplanned move is triggered at the site level, then KSYS will migrate all the host groups from one site to another site. Use the following command to trigger an unplanned move. Here, `dr_type` is optional, and the default value of `dr_type` is `planned`.

```
ksysmgr move host_group <host_group_name> to=<target_site_name> dr_type=unplanned
```

```
ksysmgr move site from=<source_site_name> to=<target_site_name> dr_type=unplanned
```

Conclusion

Each failure simulation method mentioned in this paper is tested in the lab and it worked fine. Users can refer to this paper to simulate resource failure scenarios for testing VM Recovery Manager with cluster type HA, DR, or HADR before the deployment of a production VM.

Get more information

Deployment steps for IBM VM Recovery Manager HADR cluster: https://developer.ibm.com/articles/deployment-steps-ibm-VM Recovery Manager-hadr-cluster/

About the authors

Neha Jain does functional verification testing in the VM Recovery Manager product team. She has more than 3 years of experience in the IBM Power platform. She has knowledge on disaster recovery and high availability, and has expertise with IBM i and IBM System Storage™ DS8000® storage. You can reach Neha at nehajain29@in.ibm.com.

Dishant Doriwala is a DR component test lead in the VM Recovery Manager product team. He has more than 7 years of experience working with the IBM Power platform including IBM PowerHA® SystemMirror® and VM Recovery Manager. You can reach Dishant at dishantdoriwala@in.ibm.com.

Srikanth Thanneeru is an advisory software engineer and is currently working as the test lead in the IBM VM Recovery Manager product team. Srikanth has around 10 years of experience with IBM AIX operating system functional testing. His areas of expertise include file systems, kernel, shared storage pools, high availability, and disaster recovery solution. You can reach Srikanth at sreekkanth@in.ibm.com.

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America