KVM HA with IPMI Fencing

NOTE: While we reference IPMI in this design doc, it could be any utility or shell script that can take several workflow actions and/or run external commands/apis.

Detailed description

This is an enhancement on how CloudStack deals with KVM Agent uncertainties. As of now, there is no clear and automated fail over process due to CloudStack's inability to fence a hypervisor. Here is a proposed scenario on how CloudStack would handle Agent/Link/Server Crash issues.

This feature would resolve the issue with crashing hypervisors – since this case has not been covered yet.

1) CloudStack notices that KVM host A no longer responds

2) CloudStack asks KVM neighbour host B to check on KVM host A
* In either case of KVM host B responding on state of KVM host A (or not) – yields no action – because we aren't certain what happened, did the agent die and we get no NFS Heartbeat response? Or did server crash? With this feature request, we are trying to address the server crash instance only.

3) Logic to figure out what really happened to KVM host A:

1. We need a job that will run this check with user specified interval in seconds when KVM host get disconnection
2. **b.** MS gets the list of VMs and their disks that were running on KVM host A with state "Running" and sends them to KVM host B
3. **c.** KVM host B has access to clustered storage and issues stat (or system level equivalent "c" call) to check when the last write operation has been made

i. If "rw" time stamp of any file is newer than the "disconnect" time stamp,– take no action, continue to monitor – until a failure occurs or host state changes to "Running"

ii. If looping through entire list of files yielded that rw is older than KVM host disconnect time stamp, allow for multiple (x) additional before taking next action –> x configurable value, if response came on the next check with update rw timestamp – take no action – continue to monitor, if no rw update came through, proceed with this logic

1. **1.** MS host puts KVM host A in Maintenance Mode to allow for power on of VMs that reside on this host
2. **2.** MS host checks for cluster level IPMI default action in case of host outage, 4 options should be present, all of which will send email/SNMP trap notification: do nothing, power off, reboot or issue led blink
3. **3.** We would like the IPMI interface to be as generic as possible to cover most IPMI vendors (while HP ILO is primary use case at the moment), proposed **cluster level setting** would allow for end user to specify IPMI parameters, perhaps under IMPI tab
    a. **a.** IPMI HA Default Fencing Action "do nothing, stop, reboot, blink"
    b. **b.** IPMI Username
    c. **c.** IPMI Password
    d. **d.** IPMI Exec {user configurable string}
    e. **e.** IPMI Action Stop {user configurable string}
    f. **f.** IPMI Action Start {user configurable string}
    g. **g.** IPMI Action Reboot {user configurable string}
    h. **h.** IPMI Action Blink {user configurable string}
    i. **i.** IPMI Action Test {user configurable string} -> simple operation to test IPMI interface
    j. **j.** IPMI Execution Syntax "$IPMI_EXEC --username $IPMI_Username --password $IPMI_password --command $IPMI_ACTION --host $IPMI_HOST"

1. **d.** We propose to add the IPMI tab on each hypervisor with following inputs, if **host level** settings aren't defined, inherit from cluster setting above

i. IPMI Username

ii. IPMI Password

iii. IPMI Hostname/IP

iv. IPMI Exec {user configurable string}

v. IPMI Action Stop {user configurable string}

vi. IPMI Action Start {user configurable string}

vii. IPMI Action Reboot {user configurable string}

viii. IPMI Action Blink {user configurable string}

ix. IPMI Action Test {user configurable string} -> simple operation to test IPMI interface

x. IPMI Execution Syntax “$IPMI_EXEC --username $IPMI_Username --password $IPMI_password --command $IPMI_ACTION --host $IPMI_HOST”

The reason for allowing a host level override for IPMI settings is due to the fact that a single cluster would contain a mix of hardware that may not conform to specific cluster level setting. In addition, the username and passwords may differ in some cases, so ability to override on the host level would be needed.

- Failure actions must be capped to avoid freak issue shutdowns, for example allow no more than 3 power downs within 1 hour -> both options configurable, if this condition is met – stop and notify

Detailed use-cases

Add an ability to identify when host outage occurs and fence the host using IPMI interface.

Recommended or proposed technical design & architecture

TBD

Supported Hypervisor(s). Which hypervisor should the new feature work with?

KVM with NFS primarily, however the storage check component should be modular, so in the future, Ceph or another integration can be written

Integration into CloudStack UI required?

Yes

Should users be able to access the new functionality via the UI?

Yes

Test Automation. Does the testing need to be automated and repeatable?

Yes