Building Veeam backup solution for new Hypervisors

Document version: 1.0

Overview

The document purpose is to simplify the creation of Veeam backup solutions for new hypervisors. The list of API capabilities described in the document is not exhaustive, some details might be clarified during development when hypervisor specifics became clearer.

The most important sections for the PoC of the integration are Backup Job and Entire VM Restore.

Cluster/host registration

To perform any actions with hypervisor, user first must add this hypervisor to Veeam Backup & Replication. Hypervisor is added via IP, hostname or FQDN. User specifies credentials to be used for the cluster and can change these credentials. Veeam validates hypervisor certificate on every API request.

Hypervisor API requirements

To be able to add hypervisor to Veeam Backup & Replication interface, we need to be able to get via API

1. For cluster/host:
    a. Globally unique cluster/host identifier (preferably UUID)
    b. Hypervisor version information

Workers architecture

To perform most data protection and disaster recovery operations, Veeam uses workers. Workers are Linux-based VMs that process backup workload and distribute backup traffic when transferring data to backup repositories. Each worker is launched on a specific host for the duration of a backup or restore operation. As soon as a backup or restore session starts, Veeam launches a worker, tests its configuration and installs system updates (if available). When the backup or restore session is complete, Veeam shuts down the worker VM so that it can be used for other sessions later.

In VMware backup terminology it is like “backup proxy”, the difference is that Veeam fully manages workers and shuts them down when they are not needed. A single worker can process multiple backup and restore tasks in parallel.

Worker deployment process

Veeam automatically deploys worker VM from the configuration specified by user. The process looks like that:

1. Veeam uploads vmdk/qcow2 image to a hypervisor image store
2. Veeam creates a worker VM with the configuration specified by user.
    a. User can specify these parameters:
        i. VM name
        ii. VM description
        iii. Disk storage (if hypervisor supports different storages)
        iv. Set of networks adapters connected to different networks
    b. Veeam specifies network adapter configuration and some other settings via cloud-init config which is passed by hypervisor to the worker VM.
    c. Veeam creates worker system disk by cloning the uploaded disk from image store.
    d. [For hypervisor clusters] Veeam specifies worker host affinity to spread load between different physical nodes.
3. Veeam turns on the worker VM
4. Veeam waiting for worker VM to receive IP addresses.
5. Veeam get worker VM IP addresses list from hypervisor API and connects to worker
6. Veeam performs worker update:
    a. Veeam takes a snapshot of the worker VM

    b. Veeam performs update
    c. If update was not successful, Veeam reverts worker to the snapshot
    d. Veeam removes the created snapshot and continue
7. Veeam performs backup or restore tasks
8. Veeam shuts down the worker when it is no longer needed.
9. Veeam may redeploy (delete and create again) worker on major updates

Hypervisor API requirements

We need the following API capabilities to be able to support workers:

1. For VMs we need to be able to:
    a. create/update/delete VM
    b. Start/stop VM
    c. Specify cloud-init config for VM on VM creation via hypervisor API
    d. Get VM IP addresses (multiple if VM has multiple adapters)
    e. Get list of all VMs
2. For VM snapshot
    a. Snapshot VM
    b. Revert VM or its disks to snapshot
    c. Delete VM snapshots
3. For disk images we need to be able to:
    a. Upload disk to image store in vmdk/qcow2 format
        i. Disks should have user-friendly names to
    b. Clone disk from the image store to a VM
    c. Delete disk from the image store
4. [For hypervisor clusters] If hypervisor supports clusters, we need to be able to:
    a. Get list of hosts under hypervisor
    b. Get information about maintenance mode for a node
    c. Specify VM affinity for a specific node of a cluster (specify that VM should be running on a specific node)

Backup Job

To back up VMs, you must configure a backup job. The backup job defines how, where and when to back up VM data. You can use one job to process one or more VMs. Jobs can be started manually or scheduled to run automatically at a specific time.

The goal is to back up VM configuration and disk content to be able to restore it later.

Backup job can utilize one (or both – depends on hypervisor) transport modes:

1. Network transport mode
    a. Disk content is read from snapshot via network
    b. For example: NBD, iSCSI, etc
    c. This mode is more flexible, but often slower than hot-add
2. Hot-add transport mode
    a. Disk from snapshot is cloned/attached directly to the worker VM, and Veeam components read data directly from the attached drive
    b. This mode is typically faster than reading disk content from network but requires a worker on each hypervisor we backup.

Backup process

1. Veeam resolves included clusters/hosts/tags to a list of VMs
2. For each VM Veeam:
    a. Creates a VM snapshot
        i. If hypervisor supports quiescence/VSS via in-guest tools, Veeam may disable it.
    b. Saves VM configuration
    c. Receives CBT (changed blocks tracking) via hypervisor API:
        i. [For full backup]
            1. CBT data for the full backup contains information about zeroed and data regions on VM disks
        ii. [For incremental backup]
            1. CBT data is requested from
            2. CBT data for the incremental backup contains information about zeroed, changed and unchanged regions.
    d. For each disk (except for CD-rom) -- reads disk content from the snapshot.
        i. [For Hot-Add transport mode]
            1. Source VM disk is cloned/attached from the snapshot to the worker VM which is running on the same cluster/host

2. Worker reads data from the attached disk and writes it to Veeam backup repository
3. The disk is detached from the worker

ii. [For Network transport mode]

1. Source VM disk from the snapshot is read by a worker VM which can be located anywhere
2. Worker reads data from the attached disk via network and writes it to Veeam backup repository

e. Deletes VM snapshot

i. [For CBT based on snapshots diff] If CBT is performed via a snapshot diff method, Veeam deletes “old” snapshot from the previous job run, not the lates snapshot

Hypervisor API requirements

Veeam need to be able to perform the following actions via hypervisor API to perform VM backup:

1. For VM snapshots:
    a. Create/delete VM snapshot
    b. Specify snapshot name/description -- some field where Veeam would be able to write custom data
    c. [Nice to have] Snapshot “expiration” -- datetime, after which snapshot will be automatically removed from hypervisor (used to automatically cleanup snapshots which have not been removed for some reason)
    d. Read disk content from a snapshot via Network protocols or by attaching the disk to a worker and reading directly from the device.
    e. Get list of all VM snapshots on hypervisor
    f. Get list of all VMs
2. For VM
    a. Read VM configuration
    b. Get VM unique identifier
    c. Get VM bios UUID (can be the same as unique identifier) -- bios UUID is used to avoid double licensing if VM is process via agents and on hypervisor level
3. For VM containers (tags, resource pools, folders, etc)
    a. Get list of VMs in the container

Entire VM restore

With Veeam Backup & Replication, you can restore an entire VM from a backup file to the latest state or to a previous point in time if the original VM fails.

When you restore an entire VM, Veeam Backup & Replication extracts the VM image from a backup to the production storage. Then Veeam Backup & Replication pulls the VM data from the backup repository to the selected storage, registers the VM on the chosen hypervisor and, if necessary, powers it on.

Entire VM restore has 2 modes:

- restore to original location
    - in this case we recreate VM on the original hypervisor with the same settings the original VM had
- restore to different location
    - in this case we preserve some settings, but regenerate VM ID, disk ID, skip some VM settings which we don't need to restore in this mode.
    - Also, user can change target hypervisor, VM name, production storage and networks before starting the restore process.
    - Veeam supports restore between different platforms, so, for example, VMware machines can be restored to another hypervisor with settings close to the original.

Entire VM restore process

1. [For restore to original] Shutdown and delete the original VM if it exists
2. Create new VM
    a. In case of restore to original it will have the same settings: VM ID, Bios UUID, MAC addresses, etc
    b. In case of restore to different it will have new IDs for all the entities
3. Create VM disk on the specified datastore
4. Worker writes data from Veeam backup repository to the target disk
5. Disk is reattached/cloned to the restored VM
6. The restored VM is powered on (if needed)

Hypervisor API requirements

Veeam need to be able to perform the following actions via hypervisor API to perform VM restore:

1. For VM
    a. Start/stop VM
    b. Delete VM – to delete original VM if it exists.
    c. Create VM and specify all the VM settings we backed up (to be able to recreate VM from backup with original settings).
    d. Specify VM ID on VM creation (to support restore to original)
2. For VM Disks
    a. Write data to the target VM disk (via network or by attaching the disk to a worker VM located on the same hypervisor).
    b. [Nice to have] Specify disk ID on disk creation (to avoid full backup after restore to original)
3. For VM adapters
    a. Specify adapter MAC address on adapter creation
4. For networks
    a. Get list of virtual networks
5. For VM storages (datastores)
    a. Get list of storages
6. For VM containers (tags, folders, resource pools, etc):
    a. Add newly created to the original container

File-level restore

File-level restore (FLR) is performed mostly by Veeam components: Veeam connects to a guest VM by IP and recovers specific guest files.

Hypervisor API requirements

For the file-level recovery Veeam needs:

1. For VM:
    a. Get a list of VM IP addresses – they will be used to connect to it.

Features for the next versions

Features described in this section are likely to be out of the v1 scope. Nevertheless, they are very popular among customers and may require some additional capabilities from the hypervisor API

Guest processing

Guest Processing or Application-Aware Processing is a set of Veeam features which allow to perform application-aware backups. It includes (but is not limited to): pre-freeze & post-thaw scripts, performing VSS via Veeam VSS components, performing DB transaction log backups.

The feature is mostly done on Veeam side, but we need a few API capabilities from hypervisor:

1. Option to disable VSS/quiescence for snapshots taken during backup job. We need it to avoid VSS conflicts.
2. Method to get VM's IP addresses via API
3. Method to get VM's FQDN (needed for Kerberos support) -- less critical than IP, but still nice to have.

Instant recovery

IR allows to restore any virtual or physical machine, but it does not copy data from backup to a hypervisor datastore.

Instant recovery covered in a separate document called "Instant Recovery for new Hypervisors".