VERITAS The truth in information.

Protecting Containerized Applications with Veritas NetBackup

Solution Paper

NetBackup Version 8.1.00 and later

Contents

About this document

This document provides information about the NetBackup solution for protecting persistent volume data in a container environment.

The document contains instructions for deploying the NetBackup Client Container image in container environment to protect persistent volumes using NetBackup version 8.1 or later.

Protecting Application Data with NetBackup Client Container

To protect applications deployed in containers, Veritas now provides a NetBackup Client that can be deployed as a container. The NetBackup Client Container leverages the NetBackup policy structure to run backups. Depending on the level of protection required, the NetBackup Client Container can be used to protect containerized applications in the following ways:

- Protect application data stored on persistent volumes
- Protect application data using a staging area

The NetBackup Client Container solution is available through a docker image.

NetBackup Client Container Compatibility

Veritas provides NetBackup Client Container images based on CentOS Linux 7 for NetBackup versions 8.1 or later.

NetBackup Client Container is compatible with all Master and Media server platforms and existing licensing entitlements.

The NetBackup Client Container supports standard NetBackup network topologies. For Kubernetes support of IPv6, check the appropriate issues on Github.

- IPv6 support
- IPv4/IPv6 dual stack support

Features Not Supported by the NetBackup Client Container

With this version, following features are not supported:

- Backup, Archive, and Recovery user interface
- Use as a VMware proxy
- Snapshot client
- Replication director
- SAN client

NetBackup Client Container Deployment Strategies in Kubernetes

The container environment is dynamic wherein, applications are added or removed regularly. For example, an application that is running on one node of a cluster on a certain point can be running on a different node when restarted. Typically, the orchestrator decides which application runs on which node and when. The only thing persistent in such an environment is the storage. To protect storage in such an environment, Veritas offers a dynamic solution through the NetBackup Client Container in a way that:

1. NetBackup Client Container can reside with the application and operate from the node where the application is running. For this, deploy one NetBackup Client Container per application pod as a sidecar container.
   See, Deploy NetBackup Client Container in the Application Pod.
2. NetBackup Client Container protects the persistent storage from a single point from where it has access to volumes. For this deploy one NetBackup Client Container to protect multiple applications pods.
   See, Deploy One NetBackup Client for Multiple Application Pods.

Also, a dump and sweep approach can be used for both the deployments by mounting a dump volume.
See, Dump and Sweep Approach for the NetBackup Client Container.

Deploy NetBackup Client Container in the Application Pod

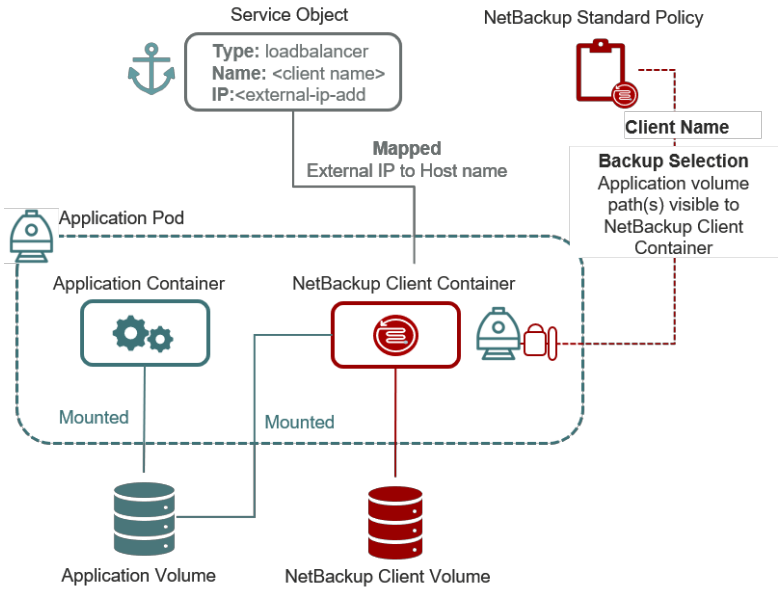
A NetBackup client container can be deployed per application pod. In this method:

- The NetBackup Client Container runs as a sidecar in the application pod. This ensures that the application and NetBackup Client Containers share the same lifecycle.
- The persistent volume(s) requiring protection must be mounted on both the application and NetBackup Client Containers.

This solution offers:

- Simplicity of management for the NetBackup administrator.
- Best throughput.
- Efficient use of NetBackup core technologies like accelerator, client direct backup, etc.
- Capability to catalog each application's data under its unique name.
- A typical NetBackup client restore experience.

The following diagram illustrates a typical deployment of NetBackup Client Container as a side car.



Deploy One NetBackup Client for Multiple Application Pods

Deploying one NetBackup Client Container to protect multiple applications pods is suitable when:

- The application owner does not want to incorporate the NetBackup Client Container image into their pod.
- The number of external IP interfaces available to the cluster are limited.
- To minimize the NetBackup footprint on the cluster.

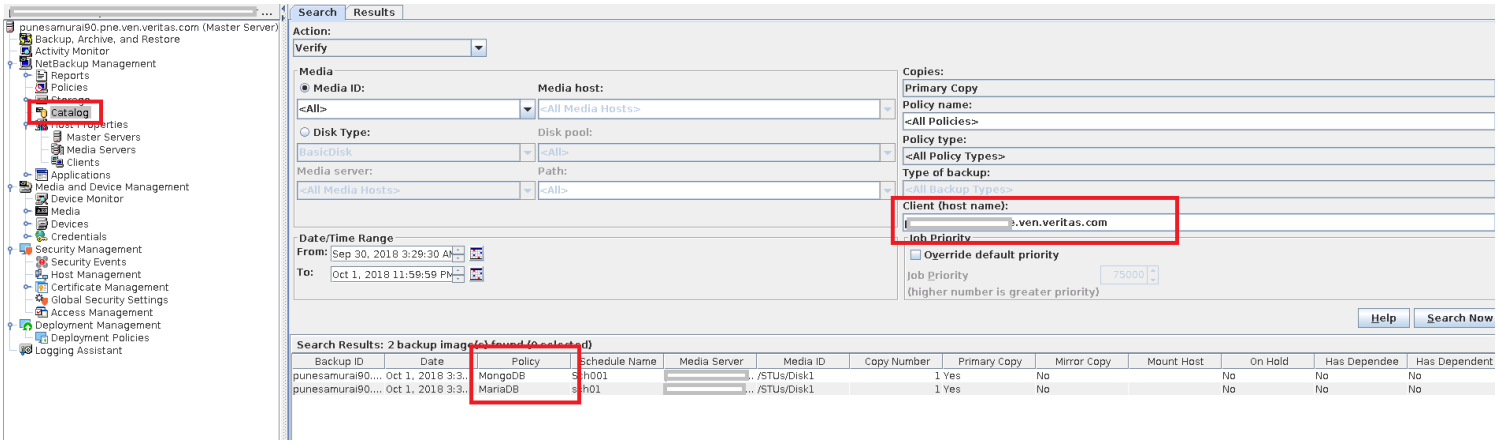
One NetBackup Client Container can be deployed per node of the Kubernetes cluster or one NetBackup Client Container per cluster depending on:

- Access mode of the persistent volumes: ‘ReadWriteOnce’ volumes are available only on one node and ‘ReadWriteMany’ volumes are available on multiple nodes of the cluster.
- Desired client throughput.

In this approach, mount all the volumes that need protection in the NetBackup Client Container. It is not possible to mount volumes after the pod has started. The administrator must know in advance the volumes that need protection and must define them in the NetBackup Client Container template before the pod is created.

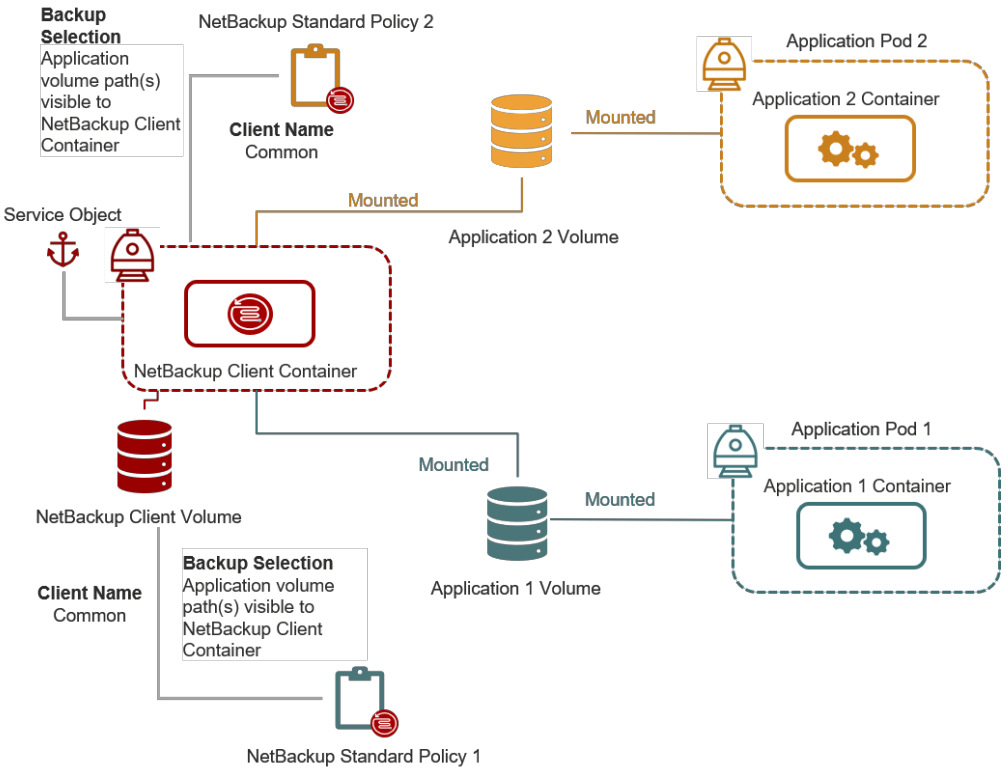
All protected applications are cataloged under the same NetBackup client name. Thus, create one NetBackup policy per application and assign keywords specific to that application.

The following example shows how different policies are created for each application protected by one NetBackup Client Container.



Also, tune the number of jobs per NetBackup Client Container depending on your environment.

The following diagram illustrates the scenario.

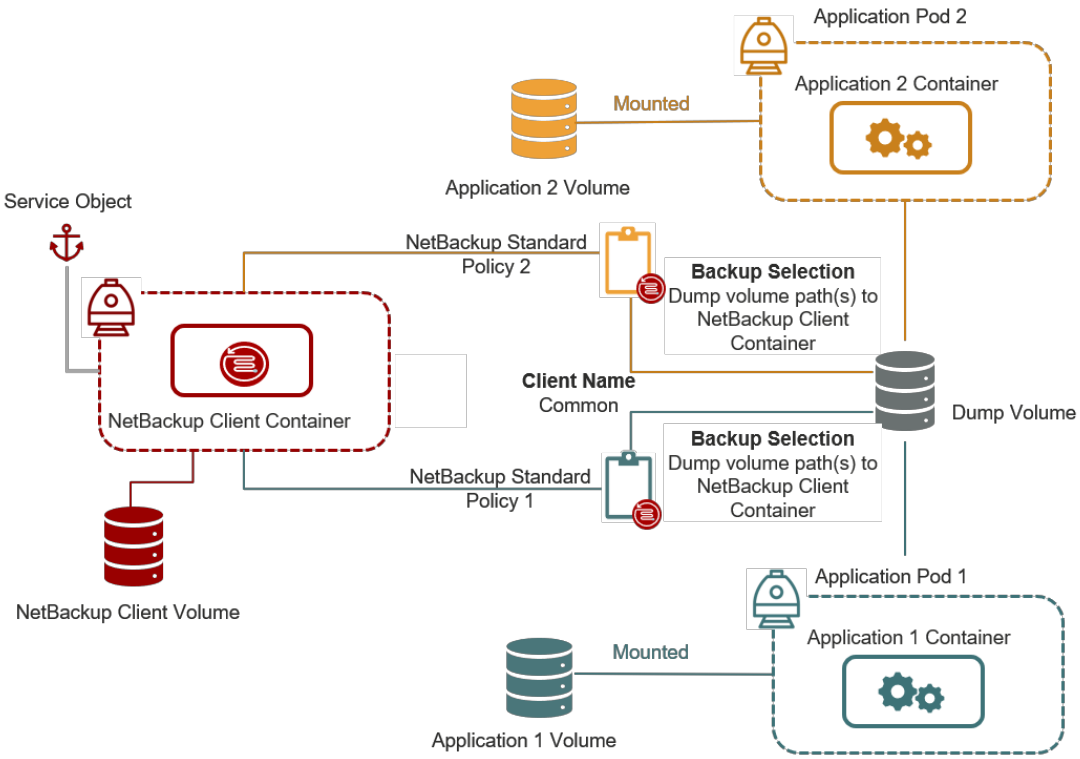


Dump and Sweep Approach for the NetBackup Client Container

With the dump and sweep approach the application pods that need protection mount the dump volume in addition to their data volume(s). The application owner dumps the application data to the dump volume. NetBackup sweeps the dump volume periodically using a NetBackup Standard policy.

All protected applications are cataloged under the same NetBackup Client name. Thus, create one NetBackup policy per application and assign keywords specific to that application.

The following diagram illustrates the dump and sweep approach when deploying one NetBackup Container for multiple application pods.

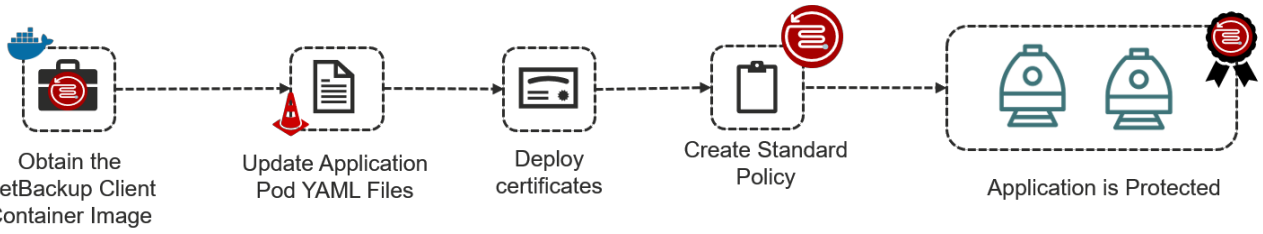


Deploying the NetBackup Client Container

The NetBackup Client Container solution is available through a docker image.

You can choose the pre-built image that is provided by Veritas or choose to build one using the NetBackup client files and docker files.

The following diagram illustrates the different tasks you need to perform to deploy the NetBackup Client Container and protect the applications.



As illustrated in the diagram, refer to the following topics to complete the tasks:

1. Obtaining the NetBackup Client Container
2. Updating the Application POD YAML file
3. Deploying Certificates
4. Creating a Standard NetBackup Policy

Obtaining the NetBackup Client Container

From Docker Store

Run the following command to obtain the NetBackup Client Container docker image.

```
docker pull store/veritasnetbackup/client:8.1.x
```

From Veritas Support Site

1. Go to https://www.veritas.com/support site.
2. Click **Licensing**. You are directed to the **Veritas Account Manager** page to access your Veritas account.
3. Enter your user credentials to access your Veritas account. You are directed to the **Veritas Entitlement Management System** site.
4. On the **Entitlements** menu, use your Entitlement ID to locate and download the Client Container Image for NetBackup version you are need.
    - NetBackup_8.1_Client_Container.tar.gz
    - NetBackup_8.1.1_Client_Container.tar.gz
    - NetBackup_8.1.2_Client_Container.tar.gz
5. In the **Actions** column against the software you want to download, click **Download**.
6. Extract the tar.
7. Add the image to docker repository by running the following command:
   `docker load -i NetBackup_8.1.x_Client_Container.tar`

Prerequisites for NetBackup Client Container

The following prerequisites must be met before deploying the NetBackup Client Container:

- External IP address or hostname: This hostname is used to configure the NetBackup Client Container on the NetBackup Master Server.
- The NetBackup Client Container must not be deployed in a pod that is managed by a controller object which produces replicas (for example, deployments, replica sets).
- Expose the following TCP ports:

  - PBX: 1556
  - vnetd-nbrntd: 13724
- Persistent storage for NetBackup data, logs, and configuration. One persistent volume can be provided for all or one for each.
- Storage for dump and sweep staging area, if using that method.

Updating the Application pod YAML file

As part of deploying NetBackup Client Container, update the application pod YAML file with details of the NetBackup Client Container and then run the command for creating the pod. For example,

```
kubectl apply -f <file_name>.yaml
```

Following is a sample YAML file for Kubernetes orchestration:

```
apiVersion: v1
kind: Pod
metadata:
  name: application-pod
  labels:
    pod: application-pod
spec:
  hostname: <client-name>
  volumes:
  - name: nb-client-vol
    persistentVolumeClaim:
      claimName: nb-client-pvc
  - name: application-vol
    persistentVolumeClaim:
      claimName: application-pvc
  containers:
  - name: nginx
    image: nginx:latest
    volumeMounts:
      - mountPath: /usr/share/nginx/html
        name: application-vol
  - name: nb-client
    image: store/veritasnetbackup/client:8.1.2
    command: [ "/entrypoint.sh" ]
    args: [ "-M", "<master-server-name>", "-c", "<client-name>" ]
    livenessProbe:
      exec:
        command:
        - /health.sh
```

```
        initialDelaySeconds: 60
        periodSeconds: 180
      volumeMounts:
        - mountPath: /mnt/nblogs
          subPath: nblogs
          name: nb-client-vol
        - mountPath: /mnt/nbcfg
          subPath: nbcfg
          name: nb-client-vol
        - mountPath: /mnt/nbdata
          subPath: nbdata
          name: nb-client-vol
        - mountPath: /backup
          name: application-vol
---
apiVersion: v1
kind: Service
metadata:
  name: <client-name>
spec:
  type: LoadBalancer
  loadBalancerIP: <external-ip-address>
  selector:
    pod: application-pod
  ports:
  - name: pbx
    port: 1556
  - name: vnetd-nbrntd
    port: 13724
```

The NetBackup Client Container entry point

The NetBackup Client Container entry point is `entrypoint.sh`.

Command

```
entrypoint.sh -c <client-name> [-i] [-M <master-server>] [-m <media-server-list>]
```

Options

`-c <client-name>`
Required parameter. Use the client name as configured in the client's **bp.conf**. Certificate is generated for this client name.

`-i`
Run the container in interactive mode.

`-M <master-server>`
Required when the container is started for the first time. It is needed to create **bp.conf**. In subsequent runs, it is optional.

`-m <media-server-list>`
Space delineated list of the media servers that are added to the client's **bp.conf**. This must be the last argument.

Deploying Certificates

Deploy certificates using one of the following methods:

- Deploy certificate manually. Follow the steps mentioned in the following technote.
  https://www.veritas.com/support/en_US/doc/21733320-127424841-0/v121744015-127424841
  You must execute the steps described in the technote in the container.
- Deploy certificate using secrets. Follow the steps.
  1. Get RSA key and token from NetBackup admin for the NetBackup Client Container hostname.
  2. Create files for the rsa_key and token that contain the rsa_key and token respectively.
  3. Create secret. Run the following command:
     ```
     kubectl create secret generic rsa-token-key --from-file=rsa_key=/rsa_key --from-file=token=/token
     ```
  4. Mount the secret at **/etc/nb-secret-vol** under NetBackup Client Container through YAML definition.

     Following is a sample snippet of the YAML, considering the secret is created with name rsa-token-key
     ```
     <snip: under NetBackup Client container>
     volumeMounts:
         - mountPath: "/etc/nb-secret-vol"
           name: client-secret-volume
     </snip>

     <snip: under volumes>
     volumes:
       - name: client-secret-volume
         secret:
           secretName: rsa-token-key
              </snip>
     ```

Note: After deleting the pod along with persistent volumes, deploy certificate using reissue token if the same client name or IP is used that was already configured in NetBackup.

Creating a NetBackup Policy

NetBackup Client Container uses the Standard or other policy type for running backup jobs.

Parameter	Value
Policy Type	Standard or other policy type
Attributes	Select according to the volume. For example, Select **Follow NFS**, if the persistent volume type is NFS.
Schedules	As appropriate for the application. In case of dump and sweep, there must be coordination between the application and backup administrators. The application dump and backup activities must take place at different times.
Client Name	As defined by the Kubernetes service object.
Backup Selections	Application volume path(s) as visible in the NetBackup Client Container. In case of dump and sweep, the dump volume path as visible in the NetBackup Client Container. It is recommended to structure the data in the dump volume such that there is one directory per application.

For information on creating policies, see Veritas NetBackup Administrator's Guide, Volume I.

Security considerations

Data stored on persistent volume provided for mount path **/mnt/nbcfg** stores critical security data such as keys and certificates for secure communication. If someone gets access to this data, they can impersonate the client. Thus, data must not be shared with any container other than the NetBackup Client Container.

Restoring persistent volume backups

For restoring persistent volume backups, consider the following:

- To restore to a volume that is already mounted on NetBackup Client Container (Original or alternate volume):

    - Use the **Backup Archive and Restore** console or the `bprestore` command to restore.
    - As the backups for multiple application pod are cataloged under same client name, ensure that separate policy is used per application pod. Also, ensure that keywords are added for identifying the backups corresponding to specific application. Use these as filters to identify the backup to be restored.
- To restore to a volume that is not mounted on the NetBackup Client Container:
    - Deploy a new client container which will mount a destination volume for restore.
    - Deploy the certificates. See, Deploying Certificates.
    - Use the **Backup Archive and Restore** console or the `bprestore` command to restore.

Note: Recovery flow for dump and sweep backup is a two-step process -- restores the dump and uses the corresponding application tools for recovery.

Code samples for reference

You can refer to code samples for NetBackup client container usage uploaded at the following location:

https://github.com/VeritasOS/netbackup-client-container-code-samples