



Simplifying Backup, Recovery, and Continuous Business Resilience in Kubernetes with Veeam Kasten

Implementation Guide for Data Protection in Kubernetes with Veeam Kasten and Lightbits

August 2025

Abstract

This white paper goes beyond simple installation, guiding you through the essential steps to connect Veeam Kasten with your existing Kubernetes and Lightbits infrastructure, ensuring seamless integration. We will illustrate how to execute backups and restores with confidence, ensuring that all components work together seamlessly. By following this approach, you can protect your critical application data, simplify disaster recovery, and maintain business continuity - all while leveraging the performance and efficiency of Lightbits software-defined storage. Get ready to transform your data protection strategy from a complex challenge into a streamlined, automated process.

Table of Contents

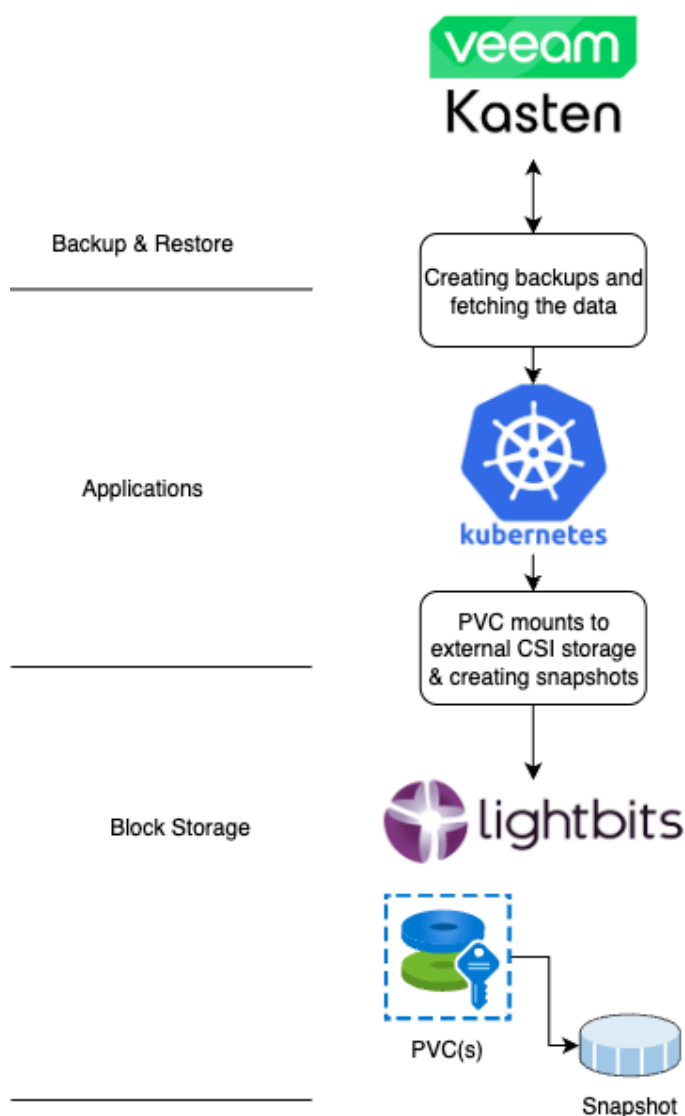
1. Introduction.....	3
2. Installing and Configuring Veeam Kasten.....	4
2.1 Configuration of volumesnapshotclass.....	7
2.2 Veeam Kasten Dashboard on its Own Internal IP Address.....	9
3. Backup and Restore.....	10
3.1 Backup volumeMode FileSystem.....	10
3.2 Restore volumeMode FileSystem.....	14
3.3 Backup volumeMode Block.....	17
3.4 Restore volumeMode Block.....	20
3.5 Snapshot Liaison with the Backup and the Snapshot in Lightbits.....	24
4. Conclusion.....	25
Veeam Kasten and Lightbits Synergy.....	25
About Lightbits Labs.....	26

1. Introduction

In this white paper, we will illustrate how to achieve a fully integrated backup and restore solution for Kubernetes, utilizing its Physical Volume Claims (PVC) on top of Lightbits software-defined block storage.

We will use Veeam Kasten as the backup and restore engine. The paper focuses on the Veeam Kasten installation and its integration with Kubernetes and Lightbits software-defined storage.

The diagram below illustrates the architecture for this implementation:



2. Installing and Configuring Veeam Kasten

For the Veeam Kasten Kubernetes server, we will use Ubuntu 24.04.

Before we install helm and Veeam Kasten, it is essential to have a default CSI storage class. In this example, we have been using a local-path storage class.

```
Shell
kubectl apply -f
https://raw.githubusercontent.com/rancher/local-path-provisioner/master/deploy/
local-path-storage.yaml
```

To verify:

```
Shell
kubectl get sc

Output:
NAME                PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION AGE
local-path (default) rancher.io/local-path   Delete          WaitForFirstConsumer
false                21h
```

To install Helm, download the correct version from <https://github.com/helm/helm/releases>. We have used [helm-v3.18.6-linux-amd64.tar.gz](https://github.com/helm/helm/releases/download/v3.18.6/helm-v3.18.6-linux-amd64.tar.gz). Unzip the [helm-v3.18.6-linux-amd64.tar.gz](https://github.com/helm/helm/releases/download/v3.18.6/helm-v3.18.6-linux-amd64.tar.gz) file as follows:

```
Shell
tar -zxvf helm-v3.18.6-linux-amd64.tar.gz

Output:
linux-amd64/
linux-amd64/README.md
linux-amd64/helm
linux-amd64/LICENSE
```



Move the Helm binary to the `/usr/local/bin` directory:

```
Shell
sudo mv linux-amd64/helm /usr/local/bin/helm
```

Verify that Helm works:

```
Shell
helm help

Output:
The Kubernetes package manager

Common actions for Helm:

- helm search:  search for charts
- helm pull:    download a chart to your local directory to view
- helm install: upload the chart to Kubernetes
- helm list:    list releases of charts

Environment variables:
etc...
```

Then add the Veeam Kasten repository:

```
Shell
helm repo add kasten https://charts.kasten.io/

Output:
"kasten" has been added to your repositories
```

Download the RPM-KASTEN for security:

```
Shell
wget https://repository.veeam.com/keys/RPM-KASTEN
```



Create a namespace in Kubernetes called kasten-io:

```
Shell
kubectl create namespace kasten-io

Output:
namespace/kasten-io created
```

Install Veeam Kasten with Helm into the Kasten namespace:

```
Shell
$ helm install k10 kasten/k10 --namespace=kasten-io --verify
--keyring=./RPM-KASTEN

Output:
NAME: k10
LAST DEPLOYED: Thu Aug 28 09:48:02 2025
NAMESPACE: kasten-io
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing Kasten's K10 Data Management Platform 8.0.7!

Documentation can be found at https://docs.kasten.io/.

How to access the K10 Dashboard:

To establish a connection to it use the following `kubectl` command:

`kubectl --namespace kasten-io port-forward service/gateway 8080:80`

The Kasten dashboard will be available at: `http://127.0.0.1:8080/k10/#/`
```

Verify that the Kasten implementation is working correctly:

```
Shell
kubectl get pods --namespace kasten-io --watch
```

Output:

NAME	READY	STATUS	RESTARTS	AGE
aggregatedapis-svc-646b5468c-rdc9n	1/1	Running	0	13m
auth-svc-64b9b847fb-nq4fr	1/1	Running	0	13m
catalog-svc-85499778c5-hkzzb	2/2	Running	0	13m
controllermanager-svc-7f68c79cc6-rrbzm	1/1	Running	0	13m
crypto-svc-f6b654f88-rkjsq	0/4	Pending	0	13m
dashboardbff-svc-b59d55b44-5cjn	0/2	Pending	0	13m
executor-svc-9cdfdc89-2h7kz	1/1	Running	0	13m
executor-svc-9cdfdc89-2rhdm	1/1	Running	0	13m
executor-svc-9cdfdc89-klfj5	1/1	Running	0	13m
frontend-svc-74d4c4c784-z42qr	1/1	Running	0	13m
gateway-67969974fd-8h4g8	1/1	Running	0	13m
jobs-svc-799b8b97d4-c7kzd	1/1	Running	0	13m
kanister-svc-77bb4bb868-xnlbv	1/1	Running	0	13m
logging-svc-884b44498-rrhrb	1/1	Running	0	13m
metering-svc-575dd4b954-mrhj8	1/1	Running	0	13m
prometheus-server-79944c88f4-mztq7	2/2	Running	0	13m
state-svc-7c7f545865-444h4	2/2	Running	0	13m

2.1 Configuration of volumesnapshotclass

Check the default `volumesnapshotclass` in Kubernetes:

Shell

```
kubectl get volumesnapshotclass
```

Output:

NAME	DRIVER	DELETIONPOLICY	AGE
example-snapshot-sc	csi.lightbitslabs.com	Delete	16h

There is no default `volumesnapshotclass` appointed in Kubernetes yet. Make the following changes:

Shell

```
kubectl edit volumesnapshotclass example-snapshot-sc
```



Add the following annotation underneath the name, as shown below:

```
Shell
metadata:
  name: example-snapshot-sc
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: "true"
    k10.kasten.io/is-default-class: "true"
    k10.kasten.io/is-snapshot-class: "true"
```

Verify that `volumesnapshotclass` is the default:

```
Shell
kubectl describe volumesnapshotclass example-snapshot-sc

Output:
Name:          example-snapshot-sc
Namespace:
Labels:        <none>
Annotations:   k10.kasten.io/is-default-class: true
               k10.kasten.io/is-snapshot-class: true
               snapshot.storage.kubernetes.io/is-default-class: true
API Version:   snapshot.storage.k8s.io/v1
Deletion Policy: Delete
Driver:        csi.lightbitslabs.com
Kind:          VolumeSnapshotClass
Metadata:
  Creation Timestamp: 2025-08-28T07:56:33Z
  Generation:        1
  Resource Version:   5904
  UID:                413ba814-911d-4fb6-80f6-eefdbb46b595
Parameters:
  csi.storage.k8s.io/snapshotter-list-secret-name: example-secret
  csi.storage.k8s.io/snapshotter-list-secret-namespace: default
  csi.storage.k8s.io/snapshotter-secret-name: example-secret
  csi.storage.k8s.io/snapshotter-secret-namespace: default
  snapshot.storage.kubernetes.io/deletion-secret-name: example-secret
  snapshot.storage.kubernetes.io/deletion-secret-namespace: default
Events: <none>
```


2.2 Veeam Kasten Dashboard on its Own Internal IP Address

By default, the Kasten dashboard is only visible internally, accessible via the IP address 127.0.0.1.

Verify that the dashboard is functional:

```
Shell
kubectl --namespace kasten-io port-forward service/gateway 8080:80

Output:
Forwarding from 127.0.0.1:8080 -> 8000
Forwarding from [::1]:8080 -> 8000
```

To access the dashboard with the IP address of the server - in this case 192.168.1.56 - do the following:

```
Shell
kubectl edit service gateway -n kasten-io
```

Change the annotation type from `clusterIP` to `NodePort` (the third line from the bottom):

```
Shell
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
```

Check the port on which the service is running now:

```
Shell
kubectl get svc gateway -n kasten-io

Output:
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
```

gateway	NodePort	10.43.82.119	<none>	80:32604/TCP	32m
---------	----------	--------------	--------	--------------	-----

Restart `port-forward` as follows:

Shell

```
kubect1 port-forward --address 0.0.0.0 service/gateway 8080:80 -n kasten-io
```

Output:

```
Forwarding from 0.0.0.0:8080 -> 8000
```

The Kasten dashboard is now open on `192.168.1.56:32604/k10/#/`.

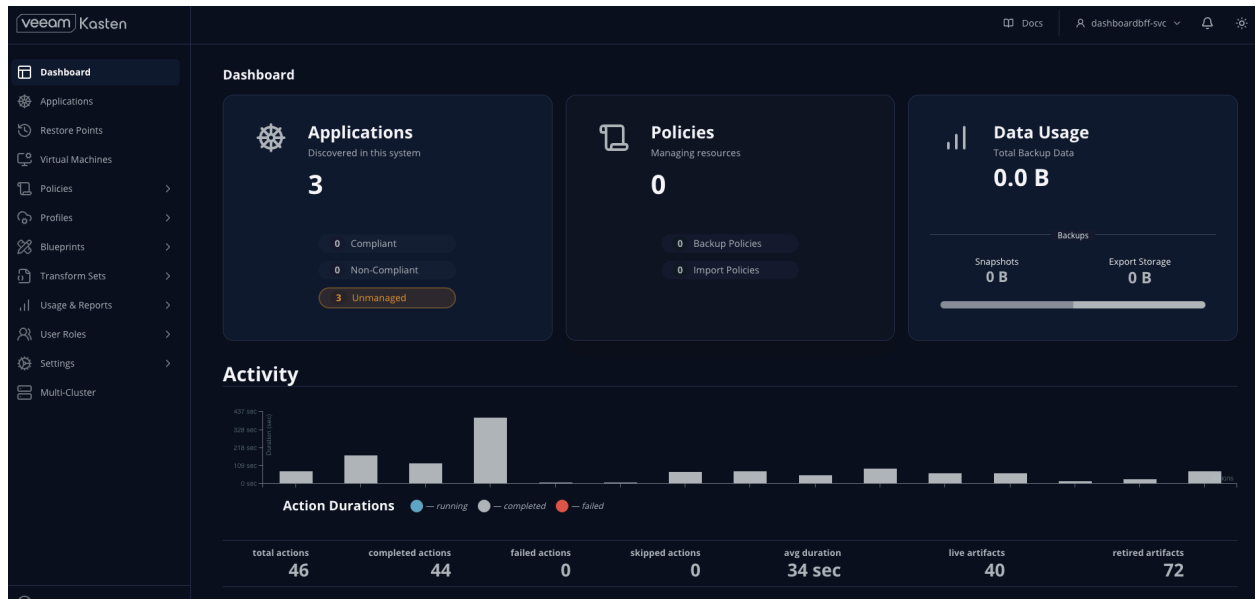
Note: The port may differ in your specific situation.

3. Backup and Restore

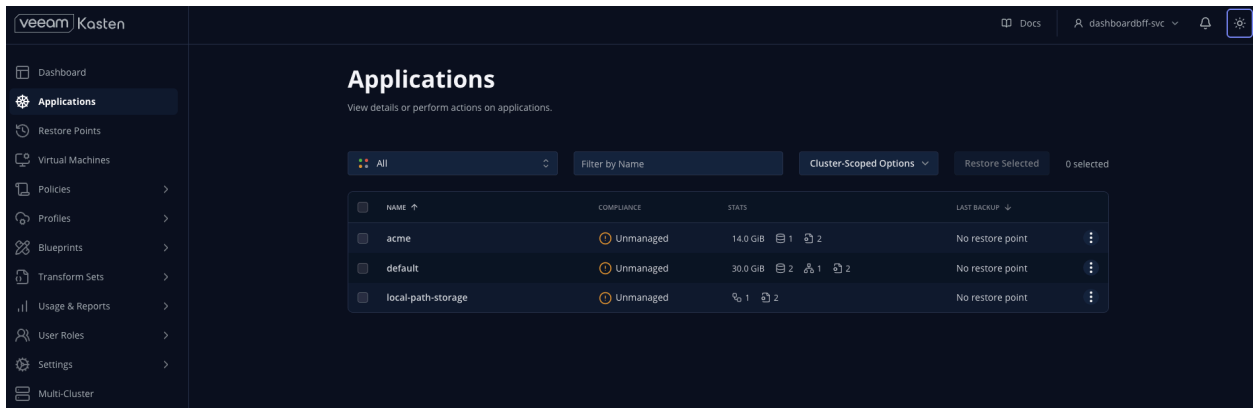
Veeam Kasten utilizes the browser to perform backup and restore activities. In our example, the address is: `http://192.168.1.56:31376/k10/#/dashboard`. Below, we will detail four examples: two for backup and two for restore.

3.1 Backup volumeMode FileSystem

When the browser uses the `http://192.168.1.56:31376/k10/#/dashboard` URL, the following screen displays:



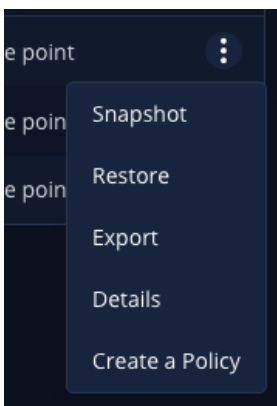
To create our first backup, select applications in the left menu bar. The following screen displays:



Applications
View details or perform actions on applications.

All Filter by Name Cluster-Scoped Options Restore Selected 0 selected

NAME	COMPLIANCE	STATUS	LAST BACKUP
acme	Unmanaged	14.0 GiB 1 2	No restore point
default	Unmanaged	30.0 GiB 2 1 2	No restore point
local-path-storage	Unmanaged	1 2	No restore point



We will use the **Acme** application. From the top right, from the three-dots menu, select **Snapshot**.

Click > **Snapshot all resources** and > **Filter Resources**. Then click **Add the filter** and in the filter, enter the app name you want to back up. In our example, it is `app: acme-fs-pod`. Notice the screen that appears on the right side.

Add Resource Filter

Filter resources by group, version, resource type, name, or label.

☐ Group
☐ Version

☐ Resource
☐ Name

☒ Label

Known labels will be listed in the dropdown. You can also add a label by typing the correct format `app:mysql` and pressing **ENTER**.

app:acme-fs-pod

> Apply to resources where labels include app:acme-fs-pod

Click **Add Filter**.

Include Filters

Include only resources that match any of these filters.
If no include filters exist, all resources are included.

Label app:acme-fs-pod

Exclude Filters

Exclude resources from any of the included resources.

No Filters

Snapshot Expiration (Optional)

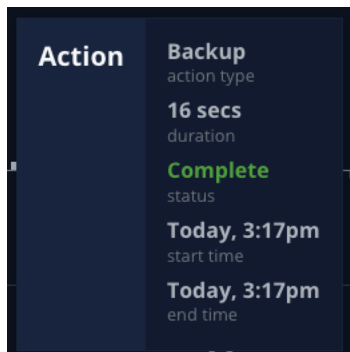
If specified the snapshot will be deleted after the selected date and time.

Click **Snapshot Application**.

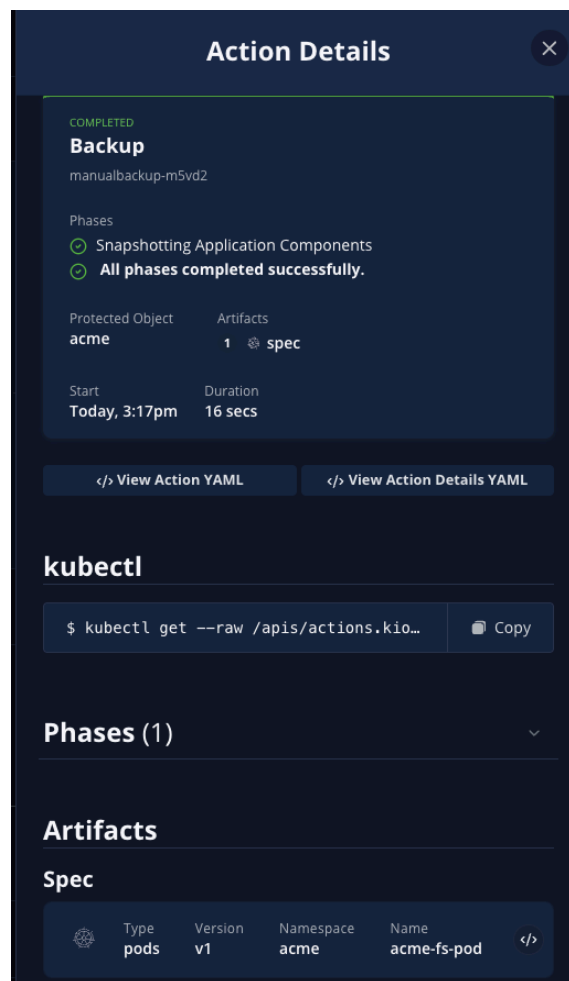
Go to the **Dashboard** (left menu item), and on the action scroll bar, click the latest backup action in the bar:



From the pop-up menu, click **Complete** to find all the details of the backup operation.



The details of the backup operation are displayed in the following screen:

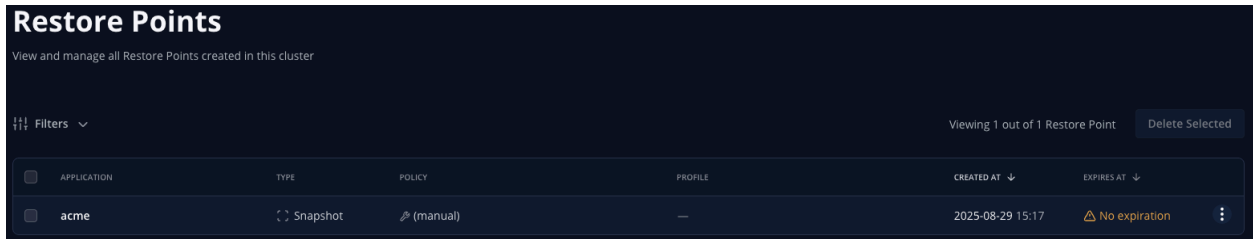


3.2 Restore volumeMode FileSystem

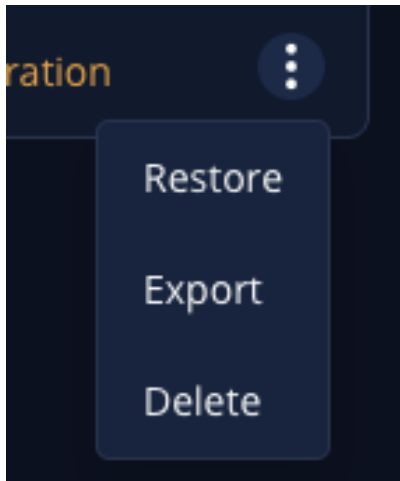
To make sure the restore operation works, first delete the pod. In our example, `acme-fs-pod` in namespace `acme`.

```
Shell
kubectl delete pod acme-fs-pod -n acme
```

In the Veeam Kasten dashboard, click the left menu bar and select the **Restore Points** item. The following screen will appear:

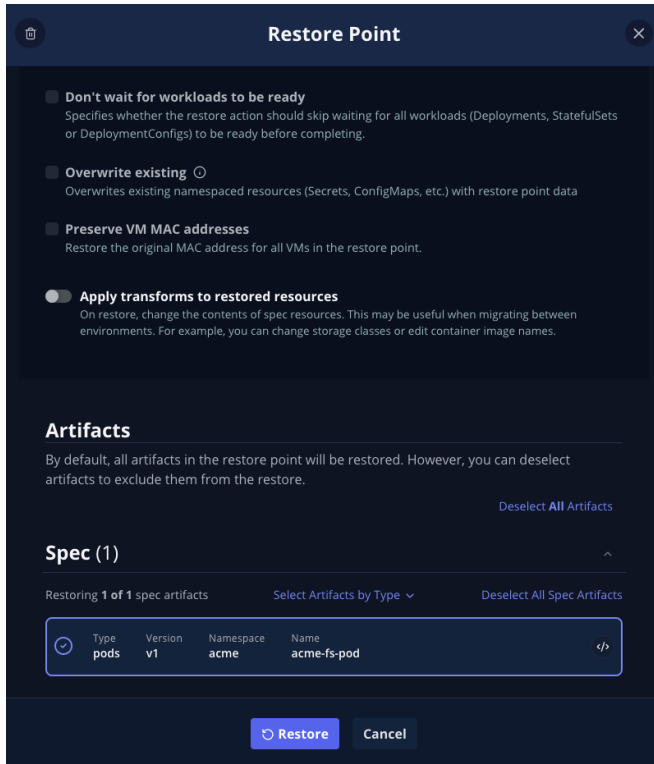


APPLICATION	TYPE	POLICY	PROFILE	CREATED AT	EXPIRES AT
acme	Snapshot	(manual)	—	2025-08-29 15:17	No expiration



To restore the `acme` application, go to the right of the three buttons and click the **Restore** option.

The following screen displays:



Restore Point

☒ **Don't wait for workloads to be ready**
Specifies whether the restore action should skip waiting for all workloads (Deployments, StatefulSets or DeploymentConfigs) to be ready before completing.

☒ **Overwrite existing** ⓘ
Overwrites existing namespaced resources (Secrets, ConfigMaps, etc.) with restore point data.

☒ **Preserve VM MAC addresses**
Restore the original MAC address for all VMs in the restore point.

☐ **Apply transforms to restored resources**
On restore, change the contents of spec resources. This may be useful when migrating between environments. For example, you can change storage classes or edit container image names.

Artifacts
By default, all artifacts in the restore point will be restored. However, you can deselect artifacts to exclude them from the restore.
[Deselect All Artifacts](#)

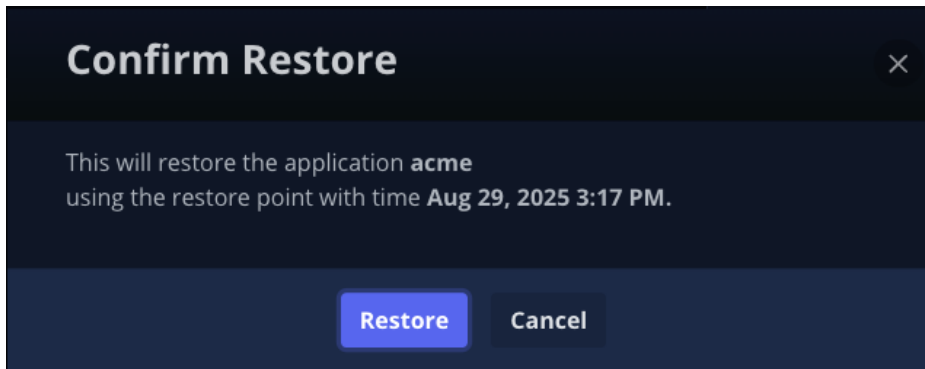
Spec (1)

Restoring 1 of 1 spec artifacts Select Artifacts by Type ▾ [Deselect All Spec Artifacts](#)

Type	Version	Namespace	Name
pods	v1	acme	acme-fs-pod

[Restore](#) [Cancel](#)

Click **Restore**, and the following pop-up displays:



Confirm Restore

This will restore the application **acme**
using the restore point with time **Aug 29, 2025 3:17 PM**.

[Restore](#) [Cancel](#)

Click **Restore**, and the pod `acme-fs-pod` will be restored on the PVC.



Go to the **Dashboard** (left menu item), and on the action scroll bar, click on the latest restore action in the bar.



In the pop-up menu, click **Complete** to find all the details of the restore operation.

Action	Restore
	action type
	16 secs
	duration
	Complete
	status
	Today, 3:30pm
	start time
	Today, 3:30pm
	end time

Action Details

COMPLETED

Restore

acme-97495

Phases

Restoring Application Components

All phases completed successfully.

Target Namespace

acme

Artifacts

none

Restore Point

manualbackup-m5vd2

Start

Today, 3:30pm

Duration

16 secs

</> View Action YAML

</> View Action Details YAML

kubectl

\$ kubectl get --raw /apis/actions.kio...

Copy

Phases (1)

Restoring Application Components

Started

Today, 3:30pm

Duration

14 secs

Attempts

1



Verify the status of the pod in Kubernetes:

Shell

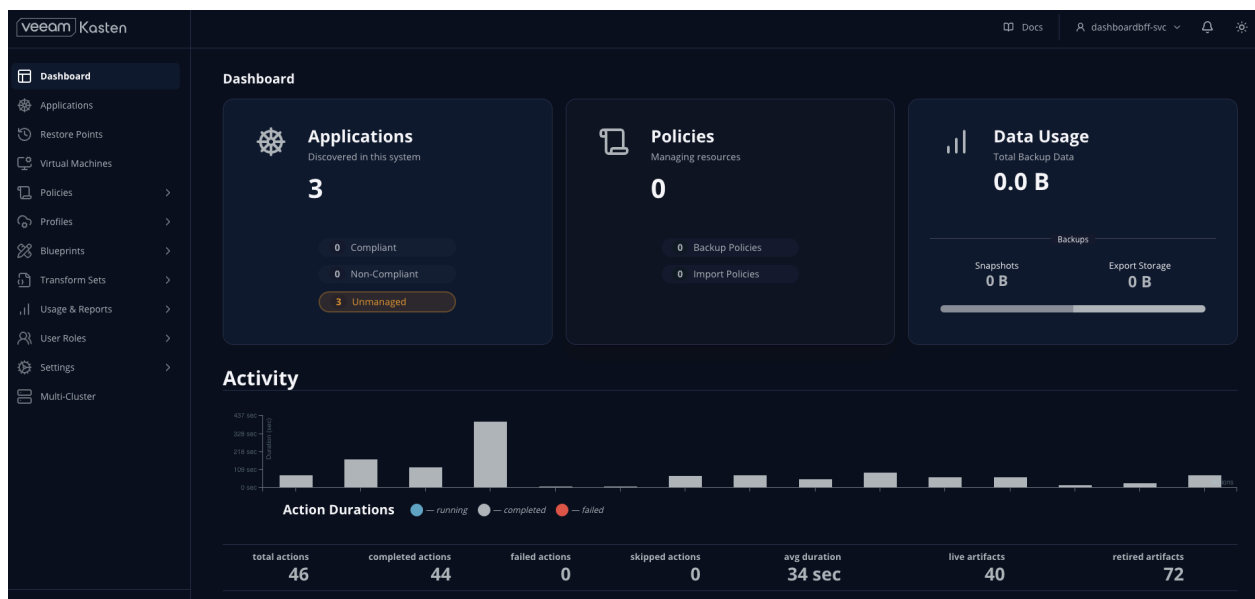
```
kubectl get pod acme-fs-pod -n acme
```

Output:

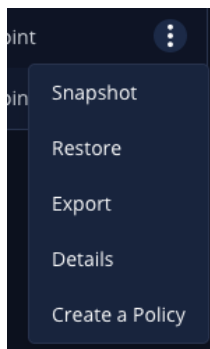
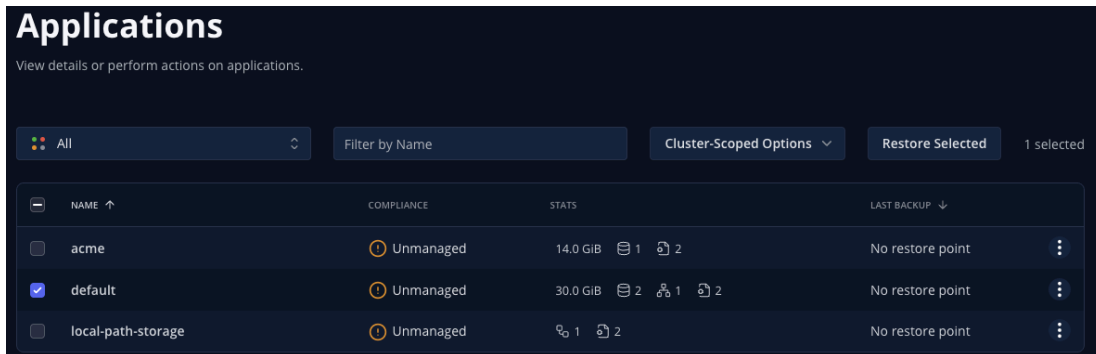
NAME	READY	STATUS	RESTARTS	AGE
acme-fs-pod	1/1	Running	0	60s

3.3 Backup volumeMode Block

When the browser uses the <http://192.168.1.56:31376/k10/#/dashboard> URL, the following screen displays:

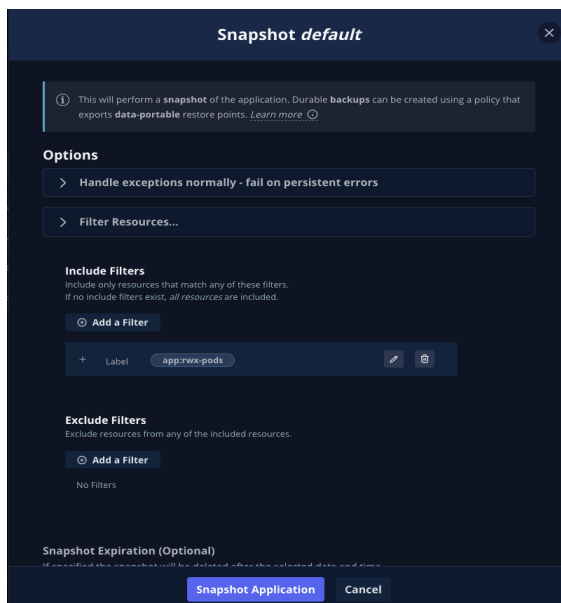


To create our first backup, select **Applications** from the left menu bar. The following screen displays:

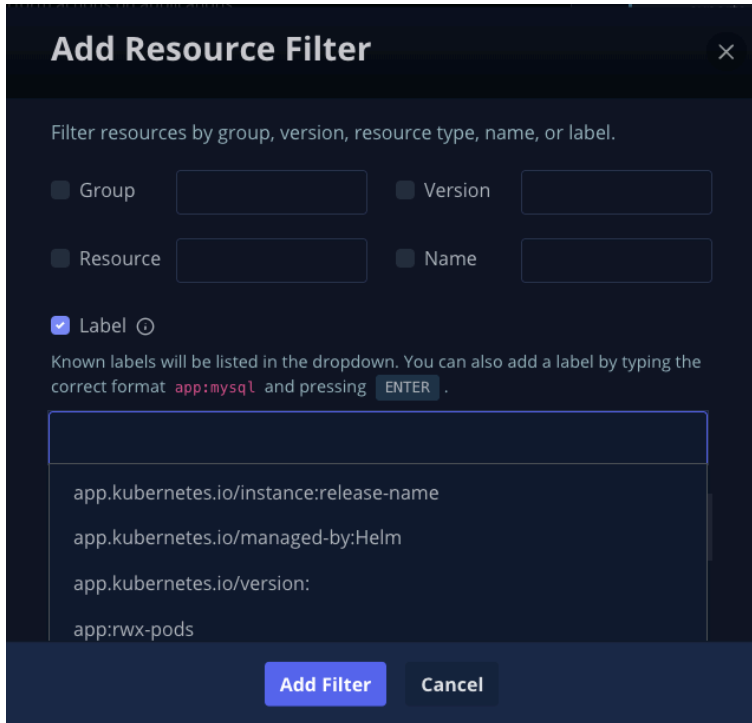


We will use the default application. From the top right, from the three-dots menu, select **Snapshot**:

Click > **Snapshot all resources** and > **Filter Resources**. Then click **Add the filter** and in the filter, enter the app name you want to back up. In our example, it is `app: rwx-pods`. Notice the screen that appears on the right side.



Click **Add Filter** and select the **app:rwx-pods**.



Add Resource Filter [X]

Filter resources by group, version, resource type, name, or label.

☐ Group ☐ Version

☐ Resource ☐ Name

☒ Label ⓘ

Known labels will be listed in the dropdown. You can also add a label by typing the correct format `app:mysql` and pressing **ENTER**.

app.kubernetes.io/instance:release-name
app.kubernetes.io/managed-by:Helm
app.kubernetes.io/version:
app:rwx-pods

Add Filter **Cancel**

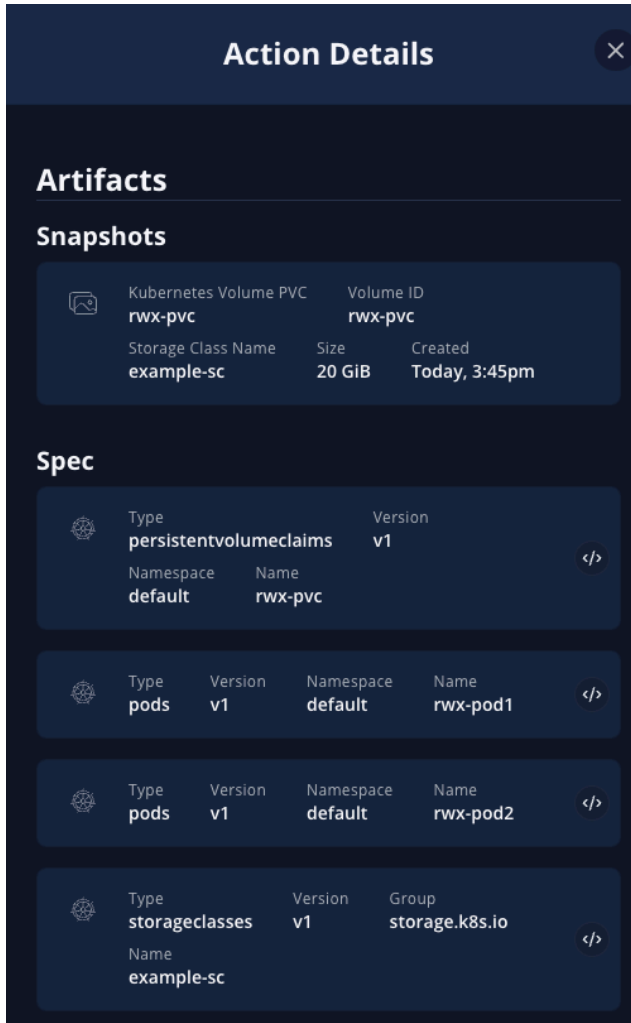
Click **Snapshot Application**. The backup will now be created.

Veeam Kasten will issue a snapshot to be created on Lightbits. The snapshot will be used to copy the pods for Veeam Kasten to back up. The snapshot will be used to restore the pods and the PVC at a later stage.

Go to the **Dashboard** (located in the left menu), and examine the activity bar.



When you hover over the latest backup activity, the pop-up menu on the right will appear. Click **Complete** to view the backup details.



Action Details

Artifacts

Snapshots

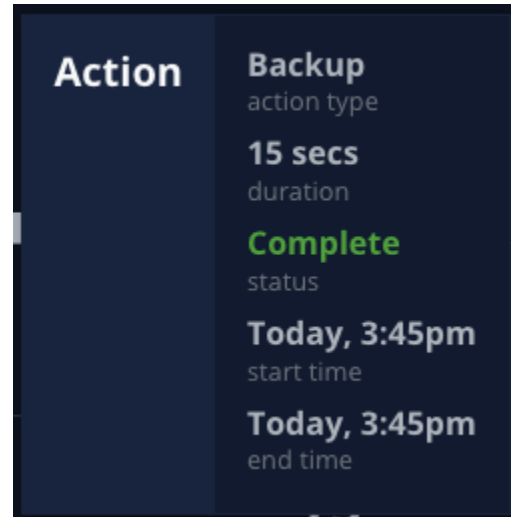
Kubernetes Volume PVC	Volume ID
rwx-pvc	rwx-pvc
Storage Class Name	Size
example-sc	20 GiB
Created	
Today, 3:45pm	

Spec

Type	Version
persistentvolumeclaims	v1
Namespace	Name
default	rwx-pvc

Type	Version	Namespace	Name
pods	v1	default	rwx-pod1
pods	v1	default	rwx-pod2

Type	Version	Group
storageclasses	v1	storage.k8s.io
Name		
example-sc		



Action

Backup
action type

15 secs
duration

Complete
status

Today, 3:45pm
start time

Today, 3:45pm
end time

The action details on the left screen provide information about the pods, PVC, and storage class used to create the snapshot. All are part of the backup.

3.4 Restore volumeMode Block

To ensure that the restore operation works, we delete the pods and the PVC in our example, specifically the pods `rwx-pod1` and `rwx-pod2`, and the PVC `rwx-pvc` - all within the default namespace.

Before we delete the pods, let's first have an overview of how long the runtime is for the pods before we delete them:



Shell

```
kubectl get pods
```

Output:

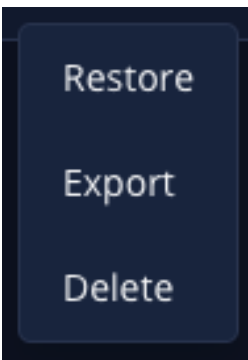
NAME	READY	STATUS	RESTARTS	AGE
rw-x-pod1	1/1	Running	0	85m
rw-x-pod2	1/1	Running	0	85m

Shell

```
kubectl delete pod rw-x-pod1
kubectl delete pod rw-x-pod2
kubectl delete pvc rw-x-pvc
```


In the Veeam Kasten dashboard, click on the left menu bar and select **Restore Points**. The following screen displays:

Restore Points						
View and manage all Restore Points created in this cluster						
Filters			Viewing 1 out of 1 Restore Point		Delete Selected	
APPLICATION	TYPE	POLICY	PROFILE	CREATED AT	EXPIRES AT	
default	Snapshot	(manual)	—	2025-08-29 15:45	No expiration	




Click the three-dots menu option on the right and click **Restore** in the pop-up menu.

The following screen displays:



Restore Point



By selecting an artifact in the restore point, you can select artifacts to exclude them from the restore.

Deselect All Artifacts

Snapshot (1)

Restoring 1 of 1 volumes

Deselect Volume Snapshots




<input checked="" type="checkbox"/>	Kubernetes Volume PVC rwx-pvc	Volume ID rwx-pvc	Storage Class Name example-sc	Size 20 GiB	Created Today, 3:45pm
-------------------------------------	----------------------------------	----------------------	----------------------------------	----------------	--------------------------

Spec (3)

Restoring 3 of 3 spec artifacts

Select Artifacts by Type ▾

Deselect All Spec Artifacts

<input checked="" type="checkbox"/>	Type pods	Version v1	Namespace default	Name rwx-pod1	
<input checked="" type="checkbox"/>	Type pods	Version v1	Namespace default	Name rwx-pod2	
<input checked="" type="checkbox"/>	Type storageclasses	Version v1	Group storage.k8s.io	Name example-sc	


Restore

Cancel

The screen above shows the pods and the PVC, which will be recreated. To create the `rwx-pvc`, the storage class `example-sc` will be used.

Click **Restore**. The pods and the PVC will be restored. The PVC will be created, and the pods will be restored from the associated snapshot in Lightbits.

Confirm Restore

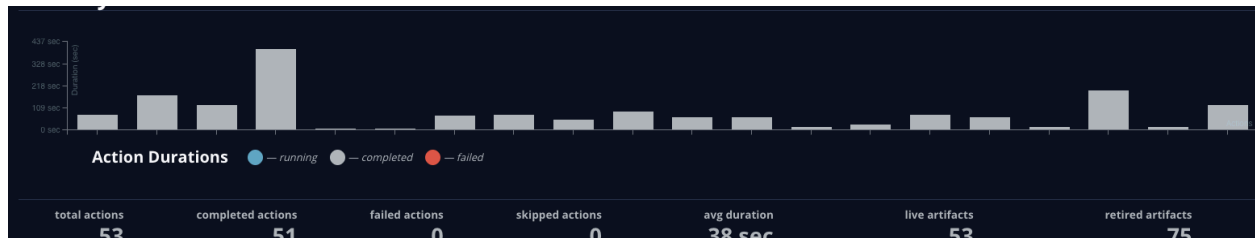


This will restore the application default
using the restore point with time Aug 29, 2025 3:45 PM.

Restore

Cancel

Go to the **Dashboard** (located in the left menu bar), and examine the activity bar.



Hover over the last action item, and the following pop-up screen (the screen on the right) displays:

Action Details

Restore Point

manualbackup-69swv

Processed Volumes

1 / 1

Start

Today, 4:05pm

Duration

2 mins, 1 sec

</> View Action YAML

</> View Action Details YAML

kubectl

\$ kubectl get --raw /apis/actions.kio...

Copy

Phases (1)

Restoring Application Components

Started

Today, 4:05pm

Duration

1 min, 56 secs

Attempts

1

Artifacts

Volumes

Volume Id	Availability Zone	Type	Created
rw-x-pvc		CSI	

Action

Restore

action type

2 mins, 1 sec

duration

Complete

status

Today, 4:05pm

start time

Today, 4:07pm

end time

The screen on the left shows that the restore was successful and that the `Volume-id` with the name `rw-x-pvc` was successfully created with CSI.



Check the runtime of the now-restored pods on the newly-created PVC:

Shell

```
kubectl get pod
```

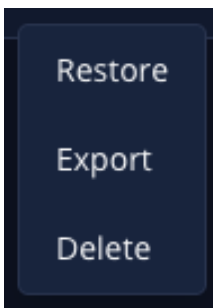
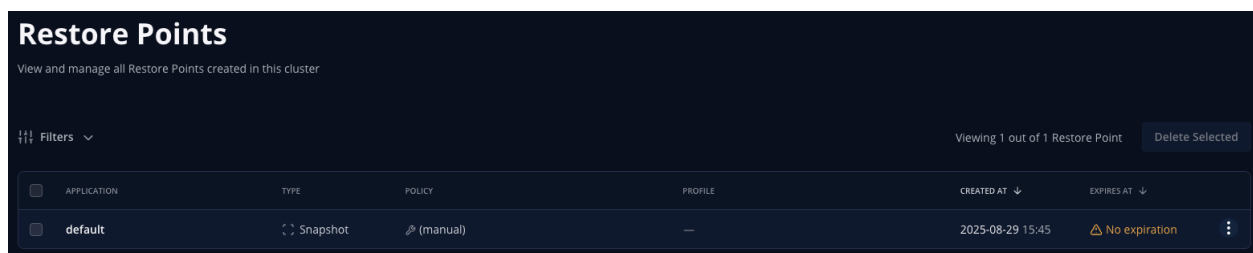
Output:

NAME	READY	STATUS	RESTARTS	AGE
rwX-pod1	1/1	Running	0	19s
rwX-pod2	1/1	Running	0	19s

3.5 Snapshot Liaison with the Backup and the Snapshot in Lightbits

When a backup is actioned, Lightbits receives the command to create a snapshot. Veeam Kasten is directly associated with that snapshot. Even when the volume is deleted in Lightbits, the snapshot remains. The integration with CSI even goes this deep, so that when this backup is deleted from Veeam Kasten, the snapshot will also be automatically deleted in Lightbits. This prevents orphaned snapshots in the Lightbits cluster.

To delete the backup in Veeam Kasten, navigate to the dashboard and select **Restore Points** from the left menu. The following screen appears:



From the three buttons on the right, click one of them, and in the pop-up menu, click **Delete**.



The following screen appears:

A dark-themed dialog box titled "Delete Restore Point". Below the title is the question "Are you sure you want to delete 1 Restore Point?". Underneath is the instruction "To confirm, type DELETE below". A text input field contains the word "DELETE" with a cursor at the end. At the bottom are two buttons: "Cancel" (dark blue) and "Delete" (red).

Type the word **DELETE** in the field and click **Delete**. The restore point will be deleted, and the snapshot in Lightbits will be deleted as well.

4. Conclusion

This comprehensive guide has demonstrated how the powerful combination of **Veeam Kasten** and **Lightbits** transforms Kubernetes data protection from a complex, labor-intensive chore into a seamless, automated process.

By integrating these leading technologies, organizations can establish a robust backup and restore solution that not only safeguards critical application data but also simplifies disaster recovery and ensures business continuity.

Veeam Kasten and Lightbits Synergy

The proven, step-by-step approach illustrated in this paper highlights the synergy between **Veeam Kasten's** intelligent backup capabilities and **Lightbits'** high-performance, efficient data platform. The result is an integrated solution that leverages external CSI snapshots for lightning-fast block-level backups and restores, as well as efficient file-system backups.

Whether you are recovering an entire namespace or a single pod, the process is streamlined and reliable. The ease of configuration and the demonstrated performance gains of this architecture empower IT teams to focus less on data loss and more on innovation.



By embracing this strategic data protection framework, organizations can confidently scale their Kubernetes environments, knowing that their data is secure, accessible, and prepared for any challenges that lie ahead. The integration of **Veeam Kasten** and **Lightbits** offers a sophisticated yet easy-to-manage solution that ensures data integrity and business resilience in the dynamic world of cloud-native applications. This framework represents a significant step forward in simplifying and enhancing the crucial task of data protection for modern enterprises.

About Lightbits Labs

Lightbits Labs® (Lightbits) invented the NVMe over TCP protocol and offers best-in-class software-defined block storage that enables data center infrastructure modernization for organizations building a private or public cloud. Built from the ground up for low consistent latency, scalability, resiliency, and cost-efficiency, Lightbits software delivers the best price/performance for real-time analytics, transactional, and AI/ML workloads. Lightbits Labs is backed by enterprise technology leaders [Cisco Investments, Dell Technologies Capital, Intel Capital, Lenovo, and Micron] and is on a mission to deliver the fastest and most cost-efficient data storage for performance-sensitive workloads at scale.

 www.lightbitslabs.com

US Offices
1830 The Alameda,
San Jose, CA 95126,
USA

 info@lightbitslabs.com

Israel Office
17 Atir Yeda Street,
Kfar Saba 4464313,
Israel

The information in this document and any document referenced herein is provided for informational purposes only, is provided as is and with all faults and cannot be understood as substituting for customized service and information that might be developed by Lightbits Labs Ltd for a particular user based upon that user's particular environment. Reliance upon this document and any document referenced herein is at the user's own risk.

The software is provided "As is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the contributors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings with the software.

Unauthorized copying or distributing of included software files, via any medium is strictly prohibited.