

[🏠 Home](#)[Getting Started](#) >[FAQs](#) >[Release notes](#) >[How-Tos and Troubleshooting](#) >[Edge Management](#) >[ZEDEDA Configuration](#) v[Identity and Access Management](#) >[Storage](#) v[📄 Storage Overview](#)[📄 ZEDEDA Store](#)[📄 Data Stores](#)[📄 Volume Instances](#)[📄 Persistent volume instances](#)[📄 ZEDEDA Redundant Storage User Guide](#)[Administration](#) >[Marketplace](#) >[Virtual Machines on EVE-OS](#) >[Integrations](#) >[Developer Program](#) >[Alerts](#) >

ZEDEDA Store

Updated a few seconds ago

Overview

ZEDEDA Store is a feature of ZEDEDATA Cloud that provides an interface to your third-party storage solutions, such as HashiCorp Vault. With it, you can give your edge assets



automated and secure access to your licenses, tokens, certificates, passwords, and other sensitive data. Whichever storage solutions you use, even if you have many, you can integrate and manage them all as a cohesive whole within ZEDEDATA Cloud.

In ZEDEDATA Store, your data are called **resources**. Each resource is categorized in one of two ways: either as a secret, or as an artifact. **Secrets** cannot be read in plain text after they have been created. **Artifacts**, however, can be read in plain text. All resources can be updated after they have been created. Each store is also classified as either a secret store or an artifact store. The type of store determines the type of the resources in it. For example, when you add a resource to a secret store, that resource becomes a secret.

Note that ZEDEDATA Store does not contain your resources directly. It only contains metadata about your resources. Your resources are always stored in your underlying storage solutions, such as Hashicorp Vault. The **metadata** that ZEDEDATA Store includes the following:

- expires_on
- created_on
- modified_on
- used_by_id

Because ZEDEDATA Store does not directly contain your resources, your storage provider is responsible for writing, reading, encrypting, and decrypting your data. Additionally, ZEDEDATA will not provision or manage your third-party storage services.

Currently, ZEDEDATA Store must be used with the ZCLI.

How resources are used

To use a resource, you must add it to one or more of your edge applications by referencing it in your applications' custom configuration pages. You must have as many copies of that resource in your store as you want to be available to your applications. Each application may use one of them. When an instance of one of those applications is created, it will claim one of the available resources. If all of your resources are in use, new application instances that depend on them will fail to spin up. If an app crashes, its resources are still in use and will be used by the app when it is rebooted. Only deleting an app will free its resources.

Additionally, resources can be configured to expire. If an application instance's resource expires, that application will continue running. New application instances, however, will fail to boot up. Expiry dates do not reset after a resource has been used and released.

Access to one, or multiple, resources is possible through the custom configuration section. Use either the `$datastore.store-name.resource-name` or `$datastore.store-name.#` statement. The ending '#' character references the next available resource.

Secret management

In this workflow, you'll create a new secret store, add a secret to it, and then reference that secret in your edge application's custom configuration.

1. Create a new store.

```
zcli datastore create secrets-app1 --fqdn="https://  
vault.company.com" \  
--dpath=YOUR_VAULT_MOUNT_PATH --dstype=secret-store:vault \  
--apikey=YOUR_API_KEY --apipass=YOUR_API_PASS
```

2. Create secrets in your new store. Remember that the resource automatically becomes a secret because you are adding it to a secrets store.

```
zcli datastore resource create secrets-app1 CLIENT_SECRET "MY  
SECRET"
```

3. Reference your secret: enter the following text in the custom configuration section of your edge application.

```
$datastore.secrets-app1.CLIENT_SECRET
```

License assignment

In this workflow, you'll deploy a new application that requires a license. you'll upload 100 licenses to a store and configure the new application to use that store.

1. Create a new store.

```
zcli datastore create licenses-app1 --dstype=artifact-  
store:vault \  
--apikey=YOUR_API_KEY --apipass=YOUR_API_PASS
```

2. Create some artifacts in your new store.

```
zcli datastore resource create licenses-app1 license1 --  
value=VALUE1  
zcli datastore resource create licenses-app1 license2 --  
value=VALUE2  
zcli datastore resource create licenses-app1 license3 --  
value=VALUE3
```

3. Create a new application and reference your artifacts. Enter the following text in the custom configuration section of your edge application.

```
$datastore.licenses-app1.#
```

ZEDEDA Store ZCLI Commands

Create a new store

```
zcli datastore create <name> [--title=<title>] --  
dstype=<dstype>  
[--description=<description>] [--fqdn=<fqdn>] [--  
region=<region>]  
[--apikey=<apikey>] [--apipass=<password>] [--dpath=<dpath>]  
[--origin-type=<origin-type>] [--project-scope=<project-names>]
```

Show a store

```
zcli datastore show-project-scope <name>
```

```
zcli datastore show project-scope <name>
```

```
zcli datastore show [--dstype=<dstype>] [[[<name> | --  
uuid=<uuid>]  
[--detail]] | [--summary] | [--name-pattern=<name-pattern>]  
[--page-size=<page-size>]]]
```

Delete a store

```
zcli datastore delete <name> [--dstype=<dstype>] [-f] [--  
delete-resources]
```

Update a store

```
zcli datastore update <name> [--dstype=<dstype>] [--  
title=title]  
[--description=<description>] [--clear-text=<true|false>]  
[--project-scope=<project-names>] [--origin-type=<origin-type>]  
[--fqdn=<fqdn>] [--apikey=<apikey>] [--dpath=<dpath>]
```

Create a resource

```
zcli datastore create-resource <name> --resource-  
name=<resource-name>  
--value=<value> [--title=<title>] [--description=<description>]  
[--expires-on=<expires-on>]
```

Show a store's resource

```
zcli datastore show-resource [<name>] [--resource-  
name=<resource-name>] [--detail]
```

Delete a resource

Delete a resource

```
zcli datastore delete <name> [--dstype=<dstype>] [-f] [--delete-resources]
```

Update a resource

```
zcli datastore update <name> [--dstype=<dstype>] [--title=title]
[--description=<description>] [--clear-text=<true|false>]
[--project-scope=<project-names>] [--origin-type=<origin-type>]
[--fqdn=<fqdn>] [--apikey=<apikey>] [--dpath=<dpath>]
```