dvvy **Admin Guide**

This document provides information relating to the installation and configuration of the dvvy Splunk app. If you have any questions, please email support@redfactor.io.

1. Define data ownership & update inputs.

Create a Group Tag structure for all of the teams, departments, lines of business, etc. using Splunk. This structure should reflect the way you intend to report on utilization and charges. Below is a simple example:

Group	Group Tag Value
Network Operations Center (NOC)	100
Security Operations Center (SOC)	101
Application Development	102
Systems Engineering	103

Although you could use names (i.e., SOC, NOC, etc.) we recommend numeric values for the greatest flexibility. The dvvy app allows you to provide a mapping of the Group Tag value to a friendly name that will be used in dvvy's dashboards.

Once you have defined your tags, add pipeline keys to the `inputs.conf` files on your forwarders for the data sources to be tracked. Doing so will add index-time fields that will be used to establish data ownership within dvvy. To illustrate:

Your NOC and SOC teams have forwarders on their Linux servers monitoring `/var/log/syslog.` Each forwarder sends the data to the `linux` index with a source type of `syslog`.

Existing inputs.conf configuration:

```
[monitor:///var/log/syslog]
disabled = false
index = linux
sourcetype = syslog
```

New inputs.conf configuration:

NOC (Group Tag Value: 100)	SOC (Group Tag Value: 101)

`[monitor:///var/log/syslog]`
`_meta = group_id::100`
`disabled = false`
`index = linux`
`sourcetype = syslog`
	`[monitor:///var/log/syslog]`
	`_meta = group_id::101`
	`disabled = false`
	`index = linux`
	`sourcetype = syslog`

This will create a key/value pair of group_id=100 for the NOC-sourced data and group_id=101 for the SOC-sourced data. Throughout the dvvy app, the Group Tag Value is captured as *group*.

2. Install dvvy

As a user with admin rights, perform the following on your search head, install dvvy through *Apps > Install app from file* or manually extract app tarball in your `$SPLUNK_HOME/etc/apps` directory.

3. Add your license

If you do not have a license, request a trial activation key on the RedFactor website:

https://redfactor.io/products/dvvy/trial/

Once you have received your key via email, create a file named `dvvy.lic` in `$SPLUNK_HOME$/etc/apps/dvvy/appserver/static.` Add the key string to the empty file and save it. If you have purchased dvvy and have a license file, simply copy that file to the path above.

The license message should no longer be visible after creating the license and refreshing your browser window. If it is, perform a debug/refresh:

```
https://<splunk url>:<splunkweb port>/en-US/debug/refresh
```

followed by a _bump:

```
https://<splunk url>:<splunkweb port>/en-US/_bump
```

These will cause Splunk to read the license file without performing a full restart. If your locale isn’t en-US, you will need to update it accordingly.

4. Set configuration options

Set dvvy configuration options in the *dvvyConfig* collection:

- `currencyUnit:` $ or other currency symbol reflected in dashboards
- `currencyUnitPosition`: The position of the currency symbol in dashboards (`before` or `after`)
- `dvvyAdmin:` A list of users that have access to all dvvy data in the app dashboards
- `indexerTargetGB`: The daily indexing target per indexer in GB based on your architecture and sizing
- groupTag: The name of the index-time field that establishes data source ownership defined in step #1

5. Groups configuration

Define your organization in the *dvvyGroups* collection

- `adminContact`: A comma-separated list of administrative contacts
- `costIndexer:` The daily price of indexer resources
- costLicenseGB: The daily price of 1 GB of license usage
- `costStorageColdTB:` The daily price per TB of cold storage
- `costStorageHotWarmTB:` The daily price per TB of hot/warm storage
- `group:` The group name or ID that maps to the groupTag field value (i.e., 100 or 101 from the earlier example)
- `groupDisplay:` The group display name for dashboards
- `licenseEntitlementGB:` The amount of license allocated to a given group for reporting purposes
- `storageEntitlementGB:` The amount of storage allocated to a given group for reporting purposes
- `techContact:` A comma-separated list of technical contacts

  See *Recommended Configuration Workflow* below.

6. Add data to track

Data must be explicitly added to dvvy for tracking to occur. Once data is flowing with your Group Tag defined and the dvvyConfig and dvvyGroups configuration is in place, run the *Update dvvyData Collection* report. It will add the required information for all indexed data sources that have a Group Tag defined.

Required Data Configuration:

The *Update dvvyData Collection* will run daily and append the following for any untracked data source:

- `group:` The group ID/name
- `idx:` The index
- `st:` The source type

Optional Data Configuration:

The following optional data may be manually added to each data source:

- `costCenter:` The cost center name
- `description:` The description of the data source
- `tag:` A comma-separated list of tags for reporting purposes
- `useCase:` A comma-separated list of use cases
- `timestamp:` Timestamp of when the data source was added to dvvy (automatically populated by *Update dvvyData Collection*

7. Create summary indexes

Create the following indexes and adjust the index configuration to ensure sufficient data retention. The indexes to be created are as follows:

- `dvvy_event_summary`
- `dvvy_license_summary`
- `dvvy_storage_summary`
- `dvvy_license_cost_summary`
- `dvvy_storage_cost_summary`
- `dvvy_indexer_cost_summary`

8. Add cost center information (optional)

Optionally add cost centers to the *dvvyCostCenters* collection:

- `costCenter:` The cost center code
- `costCenterDescription:` The cost center display name for dashboards

If cost centers aren't required, consider using this field as an internal identifier or reference number to enhance reporting.

9. Validate & schedule searches

Verify proper function of the following scheduled searches. Remove the `| collect` command prior to doing so in order to avoid writing data to summaries prematurely.

- Populate Event Summary: Gathers group event count and percentage information that is written to *dvvy_event_summary*
- Populate License Summary: Gathers license usage data every 5 minutes that is written to *dvvy_license_summary* with searches against the `_internal` index
- Populate Storage Summary*:* Gathers storage usage daily that is written to *dvvy_storage_summary* using `| dbinspect`
- Populate License Cost Summary: Calculates daily license charges that are written to *dvvy_license_cost_summary*
- Populate Storage Cost Summary: Calculates daily storage charges that are written to *dvvy_storage_cost_summary*
- Populate Indexer Cost Summary: Calculates indexer charges that are written to *dvvy_indexer_cost_summary* based on license use and the `indexerTargetGB` value in *dvvyConfig*

Once you have successfully validated that all of the above searches are fully operational, confirm that the schedule for each search meets your requirements. Revise the schedule as needed.

Search Order

Follow the guidelines below when adjusting the schedule of searches.

- It is recommended that you do not update the schedule of the *Populate License Summary* schedule
- *Populate Storage Summary* should run daily for the current day's usage
- *Populate Storage Summary* must run before *Populate Storage Cost Summary*
- *Populate Event Summary* must run prior to the *Populate License Cost Summary*, *Populate Storage Cost Summary*, and *Populate Indexer Cost Summary* searches

All searches will begin running as scheduled after installation. Disable until you are ready.

10. Review dashboards

Confirm dashboards are populating. Data access is scoped by group affiliation as defined in *dvvyGroups* and all dashboards search against *dvvy_license_cost_summary*, *dvvy_storage_cost_summary*, and *dvvy_indexer_cost_summary* indexes. Any users listed in `dvvyAdmin` in the *dvvyConfig* collection will have access to all app data in the dashboards.

Recommended Configuration Workflow

If you're not already using it, we recommend that you install the Lookup File Editor app for the purpose of interacting with dvvy's KV store collections. The app is incredibly useful and will make editing, importing, and exporting collection data a snap.

Consider creating a spreadsheet in .csv file format for *dvvyGroups* and *dvvyCostCenters* with the same field names on the first line. Once complete, simply load the data using the Lookup File Editor import function.

Summary Indexes

The dvvy app store usage and calculated charges in summary indexing. In doing so, reporting performance is improved and the data is written read-only so it can't be manipulated.

Summary Name	**Description**
`dvvy_license_summary`	License usage data at 5-minute granularity
`dvvy_storage_summary`	Daily storage usage data
`dvvy_event_summary`	Daily count and percentage of events by group
`dvvy_license_cost_summary`	Daily roll-up of license charges
`dvvy_storage_cost_summary`	Daily roll-up of storage charges
`dvvy_indexer_cost_summary`	Daily roll-up of indexer charges

KV Store Collections

The dvvy app leverages KV store collections for all request operations. The table below lists the collections and their role.

Collection Name	**Description**
`dvvyConfig`	Contains app configuration options
`dvvyCostCenters`	Contains cost center information
`dvvyData`	Contains data tracked by dvvy
`dvvyGroups`	Contains organization information

KV Store Backup

Since some of the app data is stored in KV store collections and collections are susceptible to accidental deletion or overwrite (e.g., unintentional `outputlookup` by an admin), **it is highly recommended that you frequently backup all dvvy collections to prevent data loss.**

Backup the collections via the CLI:

```
$SPLUNK_HOME/bin/splunk backup kvstore [-archiveName <archive>] [-collectionName
<collection>] [-appName <app>]
```

Using the above syntax as a guide, running this command:

```
/opt/splunk/bin/splunk backup kvstore -archiveName dvvyData -collectionName dvvyData -
appName dvvy
```

will generate a .tar.gz archive in:

```
/opt/splunk/var/lib/splunk/kvstorebackup
```

The archive will contain a JSON file with the contents of the dvvyData collection which could be used to restore.

Consider scheduling this command with cron (or equivalent) or creating a simple shell script to streamline the process of backing up each dvvy collection. Please see Backup and Restore the KV store in the Admin Manual for official Splunk guidance on the topic.

Data can also be manually exported when working with the collections via the Config dashboards (which leverages the Lookup File Editor), the Lookup File Editor directly, or ad hoc with SPL (with `outputcsv`). Although these methods do create backups of collection contents, they aren't recommended for routine backup operations.

Help = Available

Initial deployment support is included with your paid dvvy license. If you would like more information or if you would like an assist with installation or configuration, please email support@redfactor.io.