HELM Installation & Configuration

Overview

Helm Overview

Installing Helm

Configuring Helm

HOW TO INSTALL

HELM

The
package manager
for Kubernetes

ON KUBERNETES CLUSTER

Setting up Your Environment

Prerequisite

Kubernetes

Installing Helm

- **From the binary releases**
- **From script**
- **Via snap**
- **Through a package manager**
  - Chocolately

Demo

Configuring our environment

Installing Helm

Installing Helm

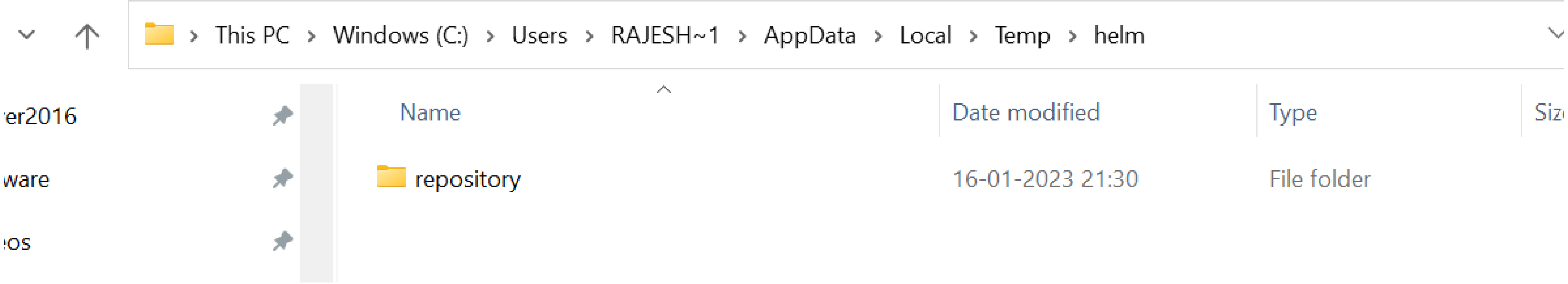
https://www.devopsschool.com/blog/helm-tutorials-how-to-install-helm/

Configuring Helm

Helm: Cache Path

The Helm cache path refers to the directory where Helm stores cached data, such as **downloaded charts** and **extracted chart files**.

This cache is used to improve performance by avoiding redundant downloads and extracting chart files.



This PC › Windows (C:) › Users › RAJESH~1 › AppData › Local › Temp › helm

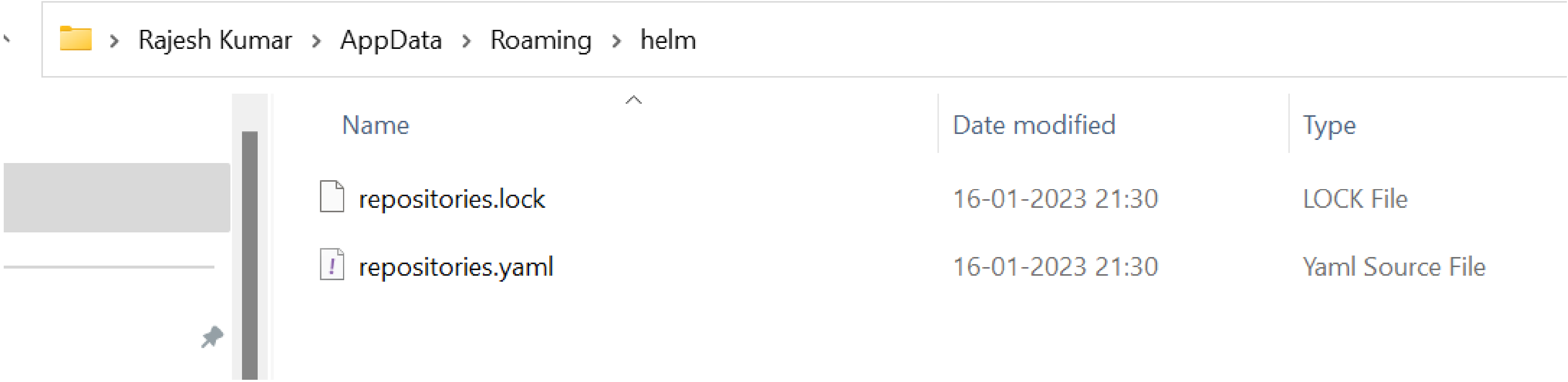
Name	Date modified	Type	Siz
repository	16-01-2023 21:30	File folder	

Helm: Configuration Path

The Helm configuration path refers to the directory where Helm stores its **configuration files**. These configuration files include settings such as

- repositories,
- credentials, and
- client-side configuration options.

In the specified directories, you can find files like config.yaml and repositories.yaml, which hold the Helm configuration information.



Rajesh Kumar › AppData › Roaming › helm

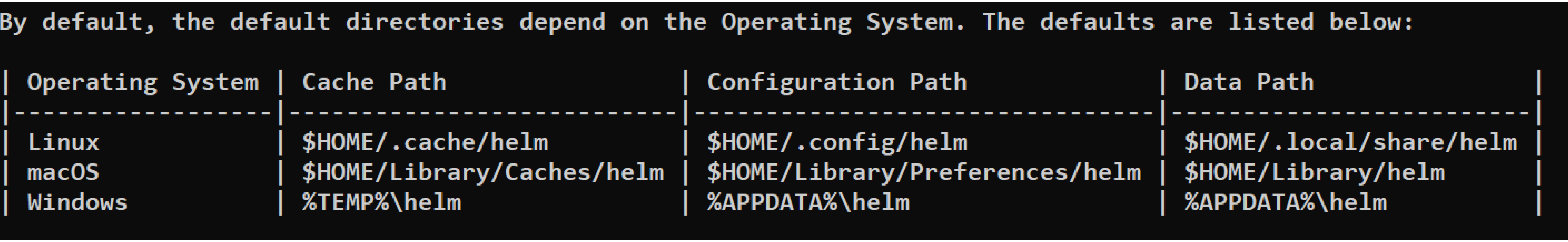
Name	Date modified	Type
repositories.lock	16-01-2023 21:30	LOCK File
repositories.yaml	16-01-2023 21:30	Yaml Source File

Helm: Data Path

The Helm data path refers to the directory where Helm stores various data files, including **cached chart archives**, **downloaded plugins**, and **other temporary data generated during Helm operations**.

In the specified directories, you can find subdirectories such as repository (containing cached chart archives), plugins (storing downloaded plugins), and other temporary files used by Helm.

Helm: Default Cache | Configuration | Data Path



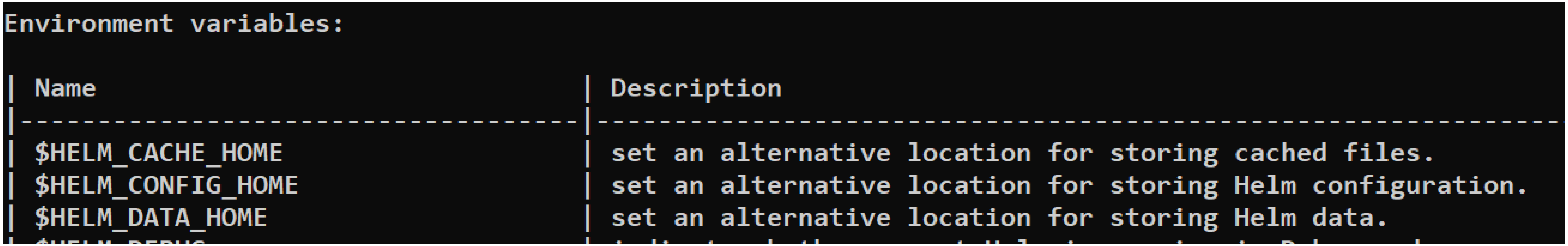
```
By default, the default directories depend on the Operating System. The defaults are listed below:

| Operating System | Cache Path                | Configuration Path             | Data Path               |
|------------------|---------------------------|--------------------------------|-------------------------|
| Linux            | $HOME/.cache/helm         | $HOME/.config/helm             | $HOME/.local/share/helm |
| macOS            | $HOME/Library/Caches/helm | $HOME/Library/Preferences/helm | $HOME/Library/helm      |
| Windows          | %TEMP%\helm               | %APPDATA%\helm                 | %APPDATA%\helm          |
```

Helm: Change Cache | Configuration | Data Path

Helm stores cache, configuration, and data based on the following configuration order:

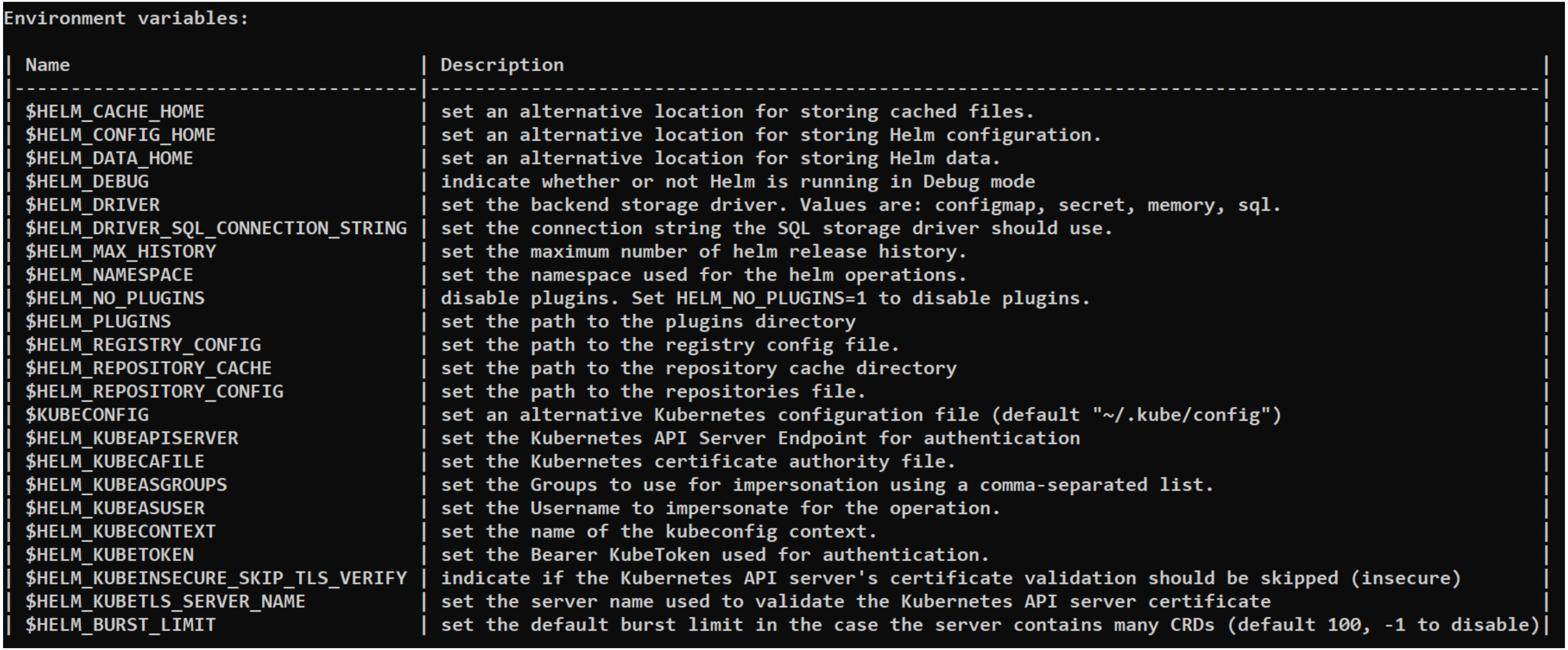
- If a HELM_*_HOME environment variable is set, it will be used
- Otherwise, on systems supporting the XDG base directory specification, the XDG variables will be used
- When no other location is set a default location will be used based on the operating system



```
Environment variables:

| Name                               | Description
|------------------------------------|------------------------------------------------------
| $HELM_CACHE_HOME                   | set an alternative location for storing cached files.
| $HELM_CONFIG_HOME                  | set an alternative location for storing Helm configuration.
| $HELM_DATA_HOME                    | set an alternative location for storing Helm data.
```

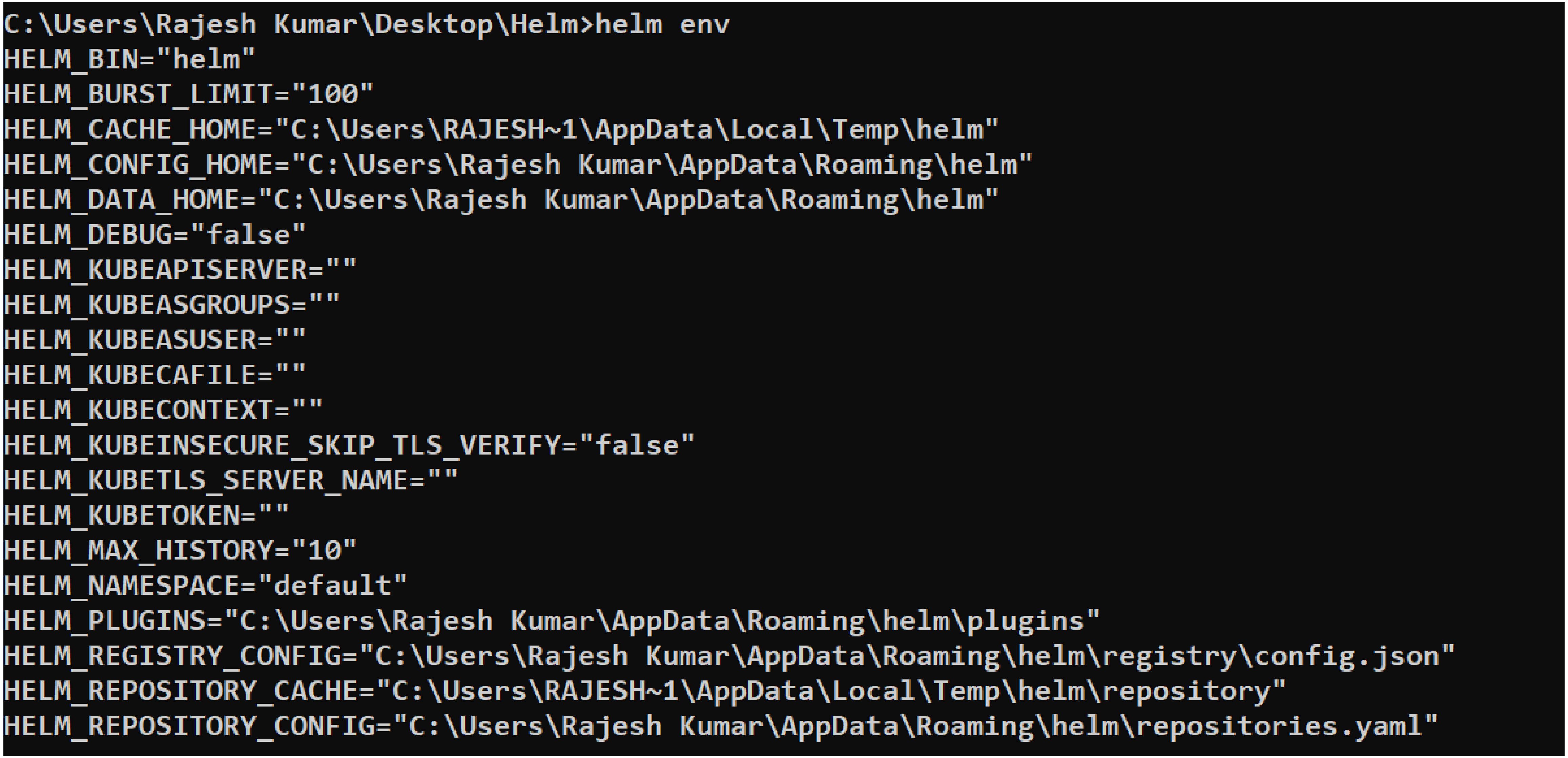
Helm: Environment variables



```
Environment variables:

| Name                               | Description                                                                                       |
|------------------------------------|---------------------------------------------------------------------------------------------------|
| $HELM_CACHE_HOME                   | set an alternative location for storing cached files.                                             |
| $HELM_CONFIG_HOME                  | set an alternative location for storing Helm configuration.                                       |
| $HELM_DATA_HOME                    | set an alternative location for storing Helm data.                                                |
| $HELM_DEBUG                        | indicate whether or not Helm is running in Debug mode                                             |
| $HELM_DRIVER                       | set the backend storage driver. Values are: configmap, secret, memory, sql.                       |
| $HELM_DRIVER_SQL_CONNECTION_STRING | set the connection string the SQL storage driver should use.                                      |
| $HELM_MAX_HISTORY                  | set the maximum number of helm release history.                                                   |
| $HELM_NAMESPACE                    | set the namespace used for the helm operations.                                                   |
| $HELM_NO_PLUGINS                   | disable plugins. Set HELM_NO_PLUGINS=1 to disable plugins.                                        |
| $HELM_PLUGINS                      | set the path to the plugins directory                                                             |
| $HELM_REGISTRY_CONFIG              | set the path to the registry config file.                                                         |
| $HELM_REPOSITORY_CACHE             | set the path to the repository cache directory                                                    |
| $HELM_REPOSITORY_CONFIG            | set the path to the repositories file.                                                            |
| $KUBECONFIG                        | set an alternative Kubernetes configuration file (default "~/.kube/config")                       |
| $HELM_KUBEAPISERVER                | set the Kubernetes API Server Endpoint for authentication                                         |
| $HELM_KUBECAFILE                   | set the Kubernetes certificate authority file.                                                    |
| $HELM_KUBEASGROUPS                 | set the Groups to use for impersonation using a comma-separated list.                             |
| $HELM_KUBEASUSER                   | set the Username to impersonate for the operation.                                                |
| $HELM_KUBECONTEXT                  | set the name of the kubeconfig context.                                                           |
| $HELM_KUBETOKEN                    | set the Bearer KubeToken used for authentication.                                                 |
| $HELM_KUBEINSECURE_SKIP_TLS_VERIFY | indicate if the Kubernetes API server's certificate validation should be skipped (insecure)       |
| $HELM_KUBETLS_SERVER_NAME          | set the server name used to validate the Kubernetes API server certificate                        |
| $HELM_BURST_LIMIT                  | set the default burst limit in the case the server contains many CRDs (default 100, -1 to disable)|
```

Helm: helm client environment information



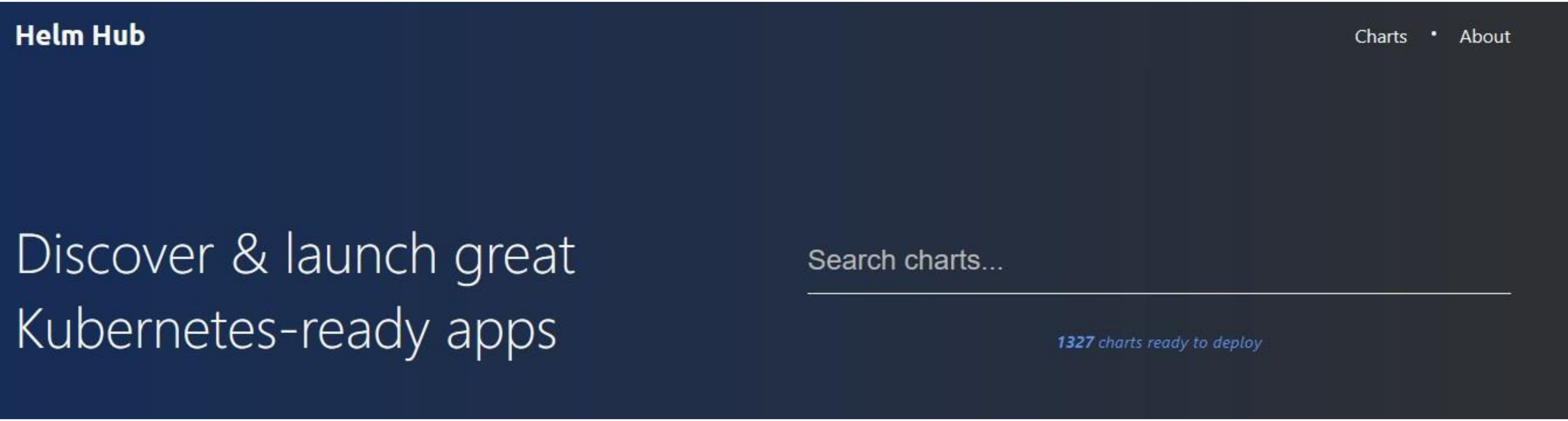
```
C:\Users\Rajesh Kumar\Desktop\Helm>helm env
HELM_BIN="helm"
HELM_BURST_LIMIT="100"
HELM_CACHE_HOME="C:\Users\RAJESH~1\AppData\Local\Temp\helm"
HELM_CONFIG_HOME="C:\Users\Rajesh Kumar\AppData\Roaming\helm"
HELM_DATA_HOME="C:\Users\Rajesh Kumar\AppData\Roaming\helm"
HELM_DEBUG="false"
HELM_KUBEAPISERVER=""
HELM_KUBEASGROUPS=""
HELM_KUBEASUSER=""
HELM_KUBECAFILE=""
HELM_KUBECONTEXT=""
HELM_KUBEINSECURE_SKIP_TLS_VERIFY="false"
HELM_KUBETLS_SERVER_NAME=""
HELM_KUBETOKEN=""
HELM_MAX_HISTORY="10"
HELM_NAMESPACE="default"
HELM_PLUGINS="C:\Users\Rajesh Kumar\AppData\Roaming\helm\plugins"
HELM_REGISTRY_CONFIG="C:\Users\Rajesh Kumar\AppData\Roaming\helm\registry\config.json"
HELM_REPOSITORY_CACHE="C:\Users\RAJESH~1\AppData\Local\Temp\helm\repository"
HELM_REPOSITORY_CONFIG="C:\Users\Rajesh Kumar\AppData\Roaming\helm\repositories.yaml"
```

Helm: Popular Repository

1. Helm Hub: Official Helm repository managed by the Helm community.
   URL: https://hub.helm.sh/
2. Artifact Hub: A community-driven repository for Helm charts and other artifacts.
   URL: https://artifacthub.io/
3. Bitnami Charts: Helm charts maintained by Bitnami for various applications and infrastructure components.
   URL: https://github.com/bitnami/charts
4. Stable Charts: Official Helm Charts repository maintained by the Helm community.
   URL: https://github.com/helm/charts/tree/main/stable
5. Incubator Charts: Official Helm Charts repository for incubating charts maintained by the Helm community.
   URL: https://github.com/helm/charts/tree/main/incubator
6. Codecentric Charts: A collection of Helm charts provided by Codecentric AG.
   URL: https://github.com/codecentric/helm-charts

Adding the Stable Repository

```
helm repo add stable https://charts.helm.sh/stable
```



Discover & launch great Kubernetes-ready apps

1327 charts ready to deploy

Helm: Charts and Repository Commands

pull	download a chart from a repository and (optionally) unpack it in local directory
push	push a chart to remote
registry	login to or logout from a registry
repo	add, list, remove, update, and index chart repositories
search	search for a keyword in charts
show	show information of a chart

Helm: helm pull

Pull a specific chart version from a remote repository:

```
$ helm repo add stable https://charts.helm.sh/stable
$ helm pull stable/mysql --version 1.6.1
```

Pull a chart and save it with a specific name and version:

```
$ helm pull stable/nginx-ingress --version 1.2.3 --untar --untardir ./nginx-ingress
```

Pull a chart and save it without extracting the archive:

```
$ helm pull stable/redis --version 7.2.1 --untar=false
```

Pull a chart from a specific repository using a custom configuration file:

```
$ helm --kubeconfig=/path/to/kubeconfig.yaml --repository-config=/path/to/repo.yaml pull
stable/mariadb
```

Pull a private chart from a repository using credentials:

```
$ helm pull private-chart --repo https://example.com/charts --username myuser --password mypassword
```

Helm: helm push

https://www.devopsschool.com/blog/helm-tutorials-helm-push-command-to-push-chart-to-registry/

Helm: List of Private Registry

https://www.devopsschool.com/blog/list-of-helm-chart-private-registry-software/

Helm: How to publish Chart at Artifacthub?

https://www.devopsschool.com/blog/list-of-helm-chart-private-registry-software/

Helm: helm show

The helm show command is used to display information about a chart, including its values, templates, and metadata. Here are the different subcommands available with helm show

helm show chart: Displays the information and metadata of a chart.

```
$ helm show chart mychart
```

helm show readme: Shows the README file of a chart.

```
helm show readme mychart
```

helm show values: Displays the default values.yaml file of a chart.

```
$ helm show values mychart
```

helm show all: Shows all the information available for a chart, including its metadata, README, and values.

```
$ helm show all mychart
```

Helm: helm show

```
helm show -h
helm install mysql  stable/mysql --version 1.6.1
helm show stable/mysql
helm show all
helm show all stable/mysql
helm show all stable/mysql
helm show chart stable/mysql
helm show crds stable/mysql
helm show readme stable/mysql
helm show values stable/mysql
```

```
helm show -h
  all         show all information of the chart
  chart       show the chart's definition
  crds        show the chart's CRDs
  readme      show the chart's README
  values      show the chart's values
```

Helm: helm search

The helm search command is used to search for charts in the available repositories. Here are the different subcommands available with helm search:

helm search repo: Searches for charts in the configured Helm repositories.

```
$ helm search repo mysql
```

helm search hub: Searches for charts in the official Helm Hub repository.

```
$ helm search hub nginx
```

helm search all: Searches for charts in both the configured Helm repositories and the Helm Hub repository. By default, the helm search command displays charts that match the provided search term. The output typically includes the chart name, version, description, and repository information.

```
$ helm search all redis
```

For example, to search for charts that match a specific version constraint:

```
$ helm search repo mysql --version ">=1.0.0,<=2.0.0“
```

Demo

- Exploring the stable repository
- Register a Account at artifacthub.io