

# Package ‘h2osteam’

May 16, 2023

**Version** 1.9.4

**License** Apache License (== 2.0)

**Title** R Client for Enterprise Steam

**Description** Enterprise Steam is a service for securely using H2O.ai products such as H2O, Sparkling Water and Driverless AI in an enterprise environment. Enterprise Steam offers security, resource control, and resource monitoring out of the box in a multi-tenant architecture so that organizations can focus on the core of their data science practice. Enterprise Steam enables streamlined adoption of H2O.ai products in a secure manner that complies with company policies. This is a R Client for Enterprise Steam.

**Author** H2O.ai

**Maintainer** H2O.ai <support@h2o.ai>

**Date** 2023-05-16

**Depends** R (>= 2.13.0)

**Imports** http, jsonlite, utils, urltools, methods, markdown

**Suggests** h2o, knitr

**Collate** backend.r core.r h2o.r login.r h2ok8s.r

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**VignetteBuilder** knitr

## R topics documented:

h2osteam.delete_h2o_cluster . . . . .	2
h2osteam.delete_h2o_kubernetes_cluster . . . . .	2
h2osteam.get_h2o_cluster . . . . .	3
h2osteam.get_h2o_clusters . . . . .	4
h2osteam.get_h2o_engines . . . . .	4
h2osteam.get_h2o_kubernetes_cluster . . . . .	5
h2osteam.get_h2o_kubernetes_clusters . . . . .	6
h2osteam.get_h2o_kubernetes_engines . . . . .	6
h2osteam.get_profiles . . . . .	7
h2osteam.import_h2o_engine . . . . .	7
h2osteam.launch_h2o_cluster . . . . .	8
h2osteam.launch_h2o_kubernetes_cluster . . . . .	10
h2osteam.login . . . . .	11
h2osteam.stop_h2o_cluster . . . . .	12
h2osteam.stop_h2o_kubernetes_cluster . . . . .	13

**Index****14**

---

`h2osteam.delete_h2o_cluster`*Delete (Terminate) a stopped H2O cluster on Hadoop.*

---

**Description**

Delete (Terminate) a stopped H2O cluster on Hadoop.

**Usage**

```
h2osteam.delete_h2o_cluster(conn, name)
```

**Arguments**

<code>conn</code>	The <code>SteamConnection</code> object obtained by <code>h2osteam.login</code> .
<code>name</code>	The name of the H2O cluster.

**Value**

None

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
h2osteam.delete_h2o_cluster(conn, "my-cluster")

## End(Not run)
```

---

`h2osteam.delete_h2o_kubernetes_cluster`*Delete (Terminate) a stopped H2O cluster on Kubernetes.*

---

**Description**

Delete (Terminate) a stopped H2O cluster on Kubernetes.

**Usage**

```
h2osteam.delete_h2o_kubernetes_cluster(conn, name)
```

**Arguments**

<code>conn</code>	The <code>SteamConnection</code> object obtained by <code>h2osteam.login</code> .
<code>name</code>	The name of the H2O kubernetes cluster.

**Value**

None

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
h2osteam.delete_h2o_kubernetes_cluster(conn, "my-cluster")

## End(Not run)
```

---

```
h2osteam.get_h2o_cluster
```

*Get connection configuration of H2O cluster on Hadoop.*

---

**Description**

Get connection configuration of H2O cluster on Hadoop.

**Usage**

```
h2osteam.get_h2o_cluster(conn, name)
```

**Arguments**

conn	The SteamConnection object obtained by h2osteam.login.
name	The name of the H2O cluster.

**Value**

H2O cluster connection configuration that can be passed to h2o.connect (config = config).

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
config <- h2osteam.get_h2o_cluster(conn, "my-cluster")
h2o.connect(config = config)

## End(Not run)
```

---

```
h2osteam.get_h2o_clusters
```

*Get information about all clusters from the Enterprise Steam interface*

---

### Description

Get information about all clusters from the Enterprise Steam interface

### Usage

```
h2osteam.get_h2o_clusters(conn)
```

### Arguments

conn                      The SteamConnection object obtained by h2osteam.login.

### Value

The list of H2O clusters.

### Examples

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
clusters <- h2osteam.get_h2o_clusters(conn)
print(clusters)

## End(Not run)
```

---

```
h2osteam.get_h2o_engines
```

*Lists H2O engines from Enterprise Steam server.*

---

### Description

Lists H2O engines from Enterprise Steam server.

### Usage

```
h2osteam.get_h2o_engines(conn)
```

### Arguments

conn                      The SteamConnection object obtained by h2osteam.login.

### Value

list of available H2O engines

## Examples

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
h2osteam.get_h2o_engines(conn)

## End(Not run)
```

---

```
h2osteam.get_h2o_kubernetes_cluster
```

*Get connection configuration of H2O cluster on Kubernetes.*

---

## Description

Get connection configuration of H2O cluster on Kubernetes.

## Usage

```
h2osteam.get_h2o_kubernetes_cluster(conn, name)
```

## Arguments

conn	The SteamConnection object obtained by h2osteam.login.
name	The name of the H2O kubernetes cluster.

## Value

H2O cluster connection configuration that can be passed to h2o.connect(config = config).

## Examples

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
config <- h2osteam.get_h2o_kubernetes_cluster(conn, "my-cluster")
h2o.connect(config = config)

## End(Not run)
```

```
h2osteam.get_h2o_kubernetes_clusters
```

*Get information about all H2O clusters on Kubernetes*

---

**Description**

Get information about all H2O clusters on Kubernetes

**Usage**

```
h2osteam.get_h2o_kubernetes_clusters(conn)
```

**Arguments**

conn                      The SteamConnection object obtained by h2osteam.login.

**Value**

The list of H2O clusters.

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
clusters <- h2osteam.get_h2o_kubernetes_clusters(conn)
print(clusters)

## End(Not run)
```

---

```
h2osteam.get_h2o_kubernetes_engines
```

*Lists H2O Kubernetes engines from Enterprise Steam server.*

---

**Description**

Lists H2O Kubernetes engines from Enterprise Steam server.

**Usage**

```
h2osteam.get_h2o_kubernetes_engines(conn)
```

**Arguments**

conn                      The SteamConnection object obtained by h2osteam.login.

**Value**

list of available H2O Kubernetes engines

### Examples

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
h2osteam.get_h2o_kubernetes_engines(conn)

## End(Not run)
```

---

```
h2osteam.get_profiles
```

*Get details about the profiles available to the logged-in user.*

---

### Description

Get details about the profiles available to the logged-in user.

### Usage

```
h2osteam.get_profiles(conn)
```

### Arguments

conn                    The SteamConnection object obtained by h2osteam.login.

### Value

Profiles available to the logged-in user.

### Examples

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
profiles <- h2osteam.get_profiles(conn)
print(profiles)

## End(Not run)
```

---

```
h2osteam.import_h2o_engine
```

*Import H2O engine from Enterprise Steam server and makes it available to users.*

---

### Description

Only Enterprise Steam administrators are authorized to call this method.

**Usage**

```
h2osteam.import_h2o_engine(conn, path)
```

**Arguments**

conn	The SteamConnection object obtained by h2osteam.login.
path	Full path to the H2O engine on disk of the Enterprise Steam server.

**Value**

None

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
h2osteam.import_h2o_engine(conn, "/tmp/h2o-3.26.0.6-cdh6.3.zip")

## End(Not run)
```

---

```
h2osteam.launch_h2o_cluster
```

*Launch a new H2O cluster on Hadoop.*

---

**Description**

Launches a new H2O cluster on Hadoop using the parameters described below. You do not need to specify all parameters. In that case they will be filled based on the default value of the selected profile. The process of launching a cluster can take up to 5 minutes.

**Usage**

```
h2osteam.launch_h2o_cluster(
  conn,
  name = NA_character_,
  version = NA_character_,
  profile_name = NA_character_,
  nodes = NA_integer_,
  node_cpus = NA_integer_,
  yarn_vcores = NA_integer_,
  node_memory_gb = NA_integer_,
  extra_memory_percent = NA_integer_,
  max_idle_h = NA_integer_,
  max_uptime_h = NA_integer_,
  timeout_s = NA_integer_,
  yarn_queue = "",
  leader_node_id = 0
)
```

**Arguments**

conn	The SteamConnection object obtained by h2osteam.login.
name	Name of the cluster.
version	Version of H2O that will be used in the cluster.
profile_name	(Optional) Specify name of an existing profile that will be used for this cluster.
nodes	(Optional) Number of nodes of the H2O cluster.
node_cpus	(Optional) Number of CPUs/threads used by H2O on a single node. Specify '0' to use all available CPUs/threads.
yarn_vcores	(Optional) Number of YARN virtual cores per cluster node. Should match node_cpus.
node_memory_gb	(Optional) Amount of memory in GB allocated for a single H2O node.
extra_memory_percent	(Optional) Percentage of extra memory that will be allocated outside of H2O JVM for algos like XGBoost.
max_idle_h	(Optional) Maximum amount of time in hours the cluster can be idle before shutting down.
max_uptime_h	(Optional) Maximum amount of time in hours the cluster will be up before shutting down.
timeout_s	(Optional) Maximum amount of time in seconds to wait for the H2O cluster to start.
yarn_queue	(Optional) Name of the YARN queue where the cluster will be placed.
leader_node_id	(Optional) ID of the H2O leader node.

**Value**

H2O cluster connection configuration that can be passed to `h2o.connect(config = config)`.

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
config <- h2osteam.launch_h2o_cluster(conn, name = "test-cluster", version = "3.30.0.1",
                                     nodes = 5, node_memory_gb = 20)
h2o.connect(config = config)

## End(Not run)
```

---

```
h2osteam.launch_h2o_kubernetes_cluster
```

*Launch a new H2O cluster on Kubernetes.*

---

## Description

Launches a new H2O cluster on Kubernetes using the parameters described below. You do not need to specify all parameters. In that case they will be filled based on the default value of the selected profile. The process of launching a cluster can take up to 5 minutes.

## Usage

```
h2osteam.launch_h2o_kubernetes_cluster(
  conn,
  name = NA_character_,
  profile_name = NA_character_,
  version = NA_character_,
  dataset_size_gb = 0,
  dataset_dimension = c(0, 0),
  nodes = NA_integer_,
  node_cpus = NA_integer_,
  node_gpus = NA_integer_,
  node_memory_gb = NA_integer_,
  max_idle_h = NA_integer_,
  max_uptime_h = NA_integer_,
  timeout_s = NA_integer_,
  volumes = NA_character_
)
```

## Arguments

conn	The SteamConnection object obtained by h2osteam.login.
name	Name of the cluster.
profile_name	(Optional) Specify name of an existing profile that will be used for this cluster.
version	Version of H2O that will be used in the cluster.
dataset_size_gb	(Optional) Specify size of your uncompressed dataset. For compressed data source, use dataset_dimension parameter. Cluster parameters will be preset to accommodate your dataset within selected profile limits. Does not override user-specified values.
dataset_dimension	(Optional) Array of (n_rows, n_cols) representing an estimation of dataset dimensions. Use this parameter when you intend to use compressed data source like Parquet format. Cluster parameters will be preset to accommodate your dataset within selected profile limits. Does not override user-specified values.
nodes	(Optional) Number of nodes of the H2O cluster.
node_cpus	(Optional) Number of CPUs/threads used by H2O on a single node. Specify '0' to use all available CPUs/threads.
node_gpus	(Optional) Number of GPUs per node

node_memory_gb	(Optional) Amount of memory in GB allocated for a single H2O node.
max_idle_h	(Optional) Maximum amount of time in hours the cluster can be idle before shutting down.
max_uptime_h	(Optional) Maximum amount of time in hours the cluster will be up before shutting down.
timeout_s	(Optional) Maximum amount of time in seconds to wait for the H2O cluster to start.
volumes	(Optional) Specify unbound volumes to mount with this instance.

**Value**

H2O cluster connection configuration that can be passed to `h2o.connect (config = config)`.

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
config <- h2osteam.launch_h2o_kubernetes_cluster(conn, name = "test-cluster", version = "
                                                nodes = 5, node_memory_gb = 20)
h2o.connect(config = config)

## End(Not run)
```

---

h2osteam.login	<i>Connect and login to an existing Enterprise Server server.</i>
----------------	---

---

**Description**

Connect and login to an existing Enterprise Server server.

**Usage**

```
h2osteam.login(
  url = NA_character_,
  username = NA_character_,
  password = NA_character_,
  verify_ssl = TRUE,
  cacert = ""
)
```

**Arguments**

url	Object of class <code>character</code> representing the Full URL (including schema and port) of the Enterprise Steam server.
username	Object of class <code>character</code> representing the username.
password	Object of class <code>character</code> representing user's password or access token.
verify_ssl	(Optional) A logical value indicating whether to verify SSL when connecting to the server.
cacert	(Optional) Path to a CA bundle file with root and intermediate certificates of trusted CAs.

**Value**

This method will connect to the Enterprise Steam server and return a `SteamConnection` object that must be passed to subsequent methods.

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.example.com:9555",
                      username = "AzureDiamond", password = "hunter2")

## End(Not run)
```

---

```
h2osteam.stop_h2o_cluster
```

*Stop a running H2O cluster on Hadoop.*

---

**Description**

Stop a running H2O cluster on Hadoop.

**Usage**

```
h2osteam.stop_h2o_cluster(conn, name)
```

**Arguments**

<code>conn</code>	The <code>SteamConnection</code> object obtained by <code>h2osteam.login</code> .
<code>name</code>	The name of the H2O cluster.

**Value**

None

**Examples**

```
## Not run:
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",
                      username = "user01", password = "token-here")
h2osteam.stop_h2o_cluster(conn, "my-cluster")

## End(Not run)
```

---

`h2osteam.stop_h2o_kubernetes_cluster`*Stop a running H2O cluster on Kubernetes.*

---

**Description**

Stop a running H2O cluster on Kubernetes.

**Usage**

```
h2osteam.stop_h2o_kubernetes_cluster(conn, name)
```

**Arguments**

<code>conn</code>	The SteamConnection object obtained by <code>h2osteam.login</code> .
<code>name</code>	The name of the H2O kubernetes cluster.

**Value**

None

**Examples**

```
## Not run:  
conn <- h2osteam.login(url = "https://steam.h2o.ai:9555",  
                      username = "user01", password = "token-here")  
h2osteam.stop_h2o_kubernetes_cluster(conn, "my-cluster")  
  
## End(Not run)
```

# Index

h2osteam.delete\_h2o\_cluster, [2](#)  
h2osteam.delete\_h2o\_kubernetes\_cluster,  
[2](#)  
h2osteam.get\_h2o\_cluster, [3](#)  
h2osteam.get\_h2o\_clusters, [4](#)  
h2osteam.get\_h2o\_engines, [4](#)  
h2osteam.get\_h2o\_kubernetes\_cluster,  
[5](#)  
h2osteam.get\_h2o\_kubernetes\_clusters,  
[6](#)  
h2osteam.get\_h2o\_kubernetes\_engines,  
[6](#)  
h2osteam.get\_profiles, [7](#)  
h2osteam.import\_h2o\_engine, [7](#)  
h2osteam.launch\_h2o\_cluster, [8](#)  
h2osteam.launch\_h2o\_kubernetes\_cluster,  
[10](#)  
h2osteam.login, [11](#)  
h2osteam.stop\_h2o\_cluster, [12](#)  
h2osteam.stop\_h2o\_kubernetes\_cluster,  
[13](#)