

# Configuration APIs

---

[ConfigMap \[v1\]](#)

[Secret \[v1\]](#)

# ConfigMap [v1]

## Description

ConfigMap holds configuration data for pods to consume.

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>

Property	Type	Description
<code>binaryData</code>	<code>object</code>	<p>BinaryData contains the binary data. Each key must consist of alphanumeric characters, '-', '_' or '.'.</p> <p>BinaryData can contain byte sequences that are not in the UTF-8 range. The keys stored in BinaryData must not overlap with the ones in the Data field, this is enforced during validation process. Using this field will require 1.10+ apiserver and kubelet.</p>
<code>data</code>	<code>object</code>	<p>Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '.'. Values with non-UTF-8 byte sequences must use the BinaryData field. The keys stored in Data must not overlap with the keys in the BinaryData field, this is enforced during validation process.</p>
<code>immutable</code>	<code>boolean</code>	<p>Immutable, if set to true, ensures that data stored in the ConfigMap cannot be updated (only object metadata can be modified). If not set to true, the field can be modified at any time. Defaulted to nil.</p>
<code>kind</code>	<code>string</code>	<p>Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a></p>

Property	Type	Description
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.

## .binaryData

### Description

BinaryData contains the binary data. Each key must consist of alphanumeric characters, '-', '\_' or '.'. BinaryData can contain byte sequences that are not in the UTF-8 range. The keys stored in BinaryData must not overlap with the ones in the Data field, this is enforced during validation process. Using this field will require 1.10+ apiserver and kubelet.

### Type

`object`

## .data

### Description

Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '\_' or '.'. Values with non-UTF-8 byte sequences must use the BinaryData field. The keys stored in Data must not overlap with the keys in the BinaryData field, this is enforced during validation process.

### Type

`object`

## API Endpoints

The following API endpoints are available:

- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/configmaps`
- `DELETE`: delete collection of ConfigMap

- **GET** : list objects of kind ConfigMap
- **POST** : create a new ConfigMap
- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/configmaps/{name}`
  - **DELETE** : delete the specified ConfigMap
  - **GET** : read the specified ConfigMap
  - **PATCH** : partially update the specified ConfigMap
  - **PUT** : replace the specified ConfigMap

## /kubernetes/{cluster}/api/v1/namespaces/{namespace}/configmaps

### HTTP method

**DELETE**

### Description

delete collection of ConfigMap

### HTTP responses

HTTP code	Response body
200 - OK	<b>Status</b> schema
401 - Unauthorized	Empty

### HTTP method

**GET**

### Description

list objects of kind ConfigMap

### HTTP responses

HTTP code	Response body
200 - OK	<b>ConfigMapList</b> schema

HTTP code	Response body
401 - Unauthorized	Empty

## HTTP method

POST

## Description

create a new ConfigMap

## Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

## Body parameters

Parameter	Type	Description
body	ConfigMap schema	application/json formatted

## HTTP responses

HTTP code	Response body
200 - OK	ConfigMap schema
201 - Created	ConfigMap schema
202 - Accepted	ConfigMap schema
401 - Unauthorized	Empty

# /kubernetes/{cluster}/api/v1/namespaces/{namespace}/configmaps/{name}

## HTTP method

DELETE

## Description

delete the specified ConfigMap

## Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

## HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
202 - Accepted	<code>Status</code> schema
401 - Unauthorized	Empty

## HTTP method

GET

## Description

read the specified ConfigMap

## HTTP responses

HTTP code	Response body
200 - OK	<code>ConfigMap</code> schema
401 - Unauthorized	Empty

## HTTP method

PATCH

## Description

partially update the specified ConfigMap

## Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

## HTTP responses

HTTP code	Response body
200 - OK	<code>ConfigMap</code> schema
401 - Unauthorized	Empty

## HTTP method

PUT

## Description

replace the specified ConfigMap

## Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<p><code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are:</p> <ul style="list-style-type: none"> <li>- Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23.</li> <li>- Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+.</li> <li>- Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.</li> </ul>

## Body parameters

Parameter	Type	Description
<code>body</code>	<code>ConfigMap</code> schema	<code>application/json</code> formatted

## HTTP responses

HTTP code	Response body
200 - OK	<code>ConfigMap</code> schema
201 - Created	<code>ConfigMap</code> schema
401 - Unauthorized	Empty

# Secret [v1]

## Description

Secret holds secret data of a certain type. The total bytes of the values in the Data field must be less than MaxSecretSize bytes.

## Type

object

## Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>
<code>data</code>	<code>object</code>	Data contains the secret data. Each key must consist of alphanumeric characters, '-', '_' or '!'. The serialized form of the secret data is a base64 encoded string, representing the arbitrary (possibly non-string) data value here. Described in <a href="https://tools.ietf.org/html/rfc4648#section-4">https://tools.ietf.org/html/rfc4648#section-4</a>

Property	Type	Description
<code>immutable</code>	<code>boolean</code>	Immutable, if set to true, ensures that data stored in the Secret cannot be updated (only object metadata can be modified). If not set to true, the field can be modified at any time. Defaulted to nil.
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>stringData</code>	<code>object</code>	<code>stringData</code> allows specifying non-binary secret data in string form. It is provided as a write-only input field for convenience. A keys and values are merged into the data field on write, overwriting any existing values. The <code>stringData</code> field is never output when reading from the API.
<code>type</code>	<code>string</code>	Used to facilitate programmatic handling of secret data. More info: <a href="https://kubernetes.io/docs/concepts/configuration/secret/#secret-types">https://kubernetes.io/docs/concepts/configuration/secret/#secret-types</a>

**.data**

## Description

Data contains the secret data. Each key must consist of alphanumeric characters, '-', '\_' or '.'. The serialized form of the secret data is a base64 encoded string, representing the arbitrary (possibly non-string) data value here. Described in <https://tools.ietf.org/html/rfc4648#section-4>

## Type

object

## .stringData

### Description

stringData allows specifying non-binary secret data in string form. It is provided as a write-only input field for convenience. All keys and values are merged into the data field on write, overwriting any existing values. The stringData field is never output when reading from the API.

### Type

object

## API Endpoints

The following API endpoints are available:

- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/secrets`
  - `DELETE` : delete collection of Secret
  - `GET` : list objects of kind Secret
  - `POST` : create a new Secret
- `/kubernetes/{cluster}/api/v1/namespaces/{namespace}/secrets/{name}`
  - `DELETE` : delete the specified Secret
  - `GET` : read the specified Secret
  - `PATCH` : partially update the specified Secret
  - `PUT` : replace the specified Secret

# /kubernetes/{cluster}/api/v1/namespaces/{namespace}/secrets

## HTTP method

DELETE

## Description

delete collection of Secret

## HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> schema
401 - Unauthorized	Empty

## HTTP method

GET

## Description

list objects of kind Secret

## HTTP responses

HTTP code	Response body
200 - OK	<code>SecretList</code> schema
401 - Unauthorized	Empty

## HTTP method

POST

## Description

create a new Secret

## Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

## Body parameters

Parameter	Type	Description
<code>body</code>	<code>Secret</code> schema	<code>application/json</code> formatted

## HTTP responses

HTTP code	Response body
200 - OK	<code>Secret</code> schema
201 - Created	<code>Secret</code> schema
202 - Accepted	<code>Secret</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

## /kubernetes/{cluster}/api/v1/namespaces/{namespace}/secrets/{name}

### HTTP method

DELETE

### Description

delete the specified Secret

### Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

### HTTP responses

HTTP code	Response body
200 - OK	Status schema
202 - Accepted	Status schema
401 - Unauthorized	Empty

### HTTP method

GET

### Description

read the specified Secret

### HTTP responses

HTTP code	Response body
200 - OK	<code>Secret</code> schema
401 - Unauthorized	Empty

## HTTP method

PATCH

## Description

partially update the specified Secret

## Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

## HTTP responses

HTTP code	Response body
200 - OK	<code>Secret</code> schema
401 - Unauthorized	Empty

## HTTP method

PUT

## Description

replace the specified Secret

## Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

## Body parameters

Parameter	Type	Description
body	Secret schema	application/json formatted

## HTTP responses

HTTP code	Response body
200 - OK	Secret schema
201 - Created	Secret schema
401 - Unauthorized	Empty