# Metric APIs

Metrics Node [v1beta1]

Metrics Pod [v1beta1]

# Metrics Node [v1beta1]

## /kubernetes/{cluster}/apis/metrics.k8s.io/v1beta1/nodes

### Common Parameters

- `allowWatchBookmarks` (*in query*): `boolean`
  allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
- `continue` (*in query*): `string`
  The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key".

This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.

- `fieldSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their fields. Defaults to everything.
- `labelSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their labels. Defaults to everything.
- `limit` (*in query*): `integer`
  limit is a maximum number of responses to return for a list call. If more items exist, the server will set the `continue` field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true.
  The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
- `pretty` (*in query*): `string`
  If 'true', then the output is pretty printed. Defaults to 'false' unless the user-agent indicates a browser or command-line HTTP tool (curl and wget).
- `resourceVersion` (*in query*): `string`
  resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions ↗ for details.

  Defaults to unset
- `resourceVersionMatch` (*in query*): `string`
  resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set

See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions for details.
Defaults to unset

- `sendInitialEvents` (*in query*): `boolean`
  `sendInitialEvents=true` may be set together with `watch=true`. In that case, the watch stream will begin with synthetic events to produce the current state of objects in the collection. Once all such events have been sent, a synthetic "Bookmark" event will be sent. The bookmark will report the ResourceVersion (RV) corresponding to the set of objects, and be marked with `"k8s.io/initial-events-end": "true"` annotation. Afterwards, the watch stream will proceed as usual, sending watch events corresponding to changes (subsequent to the RV) to objects watched.
  When `sendInitialEvents` option is set, we require `resourceVersionMatch` option to also be set. The semantic of the watch request is as following: - `resourceVersionMatch` = NotOlderThan is interpreted as "data at least as new as the provided `resourceVersion`" and the bookmark event is send when the state is synced to a `resourceVersion` at least as fresh as the one provided by the ListOptions. If `resourceVersion` is unset, this is interpreted as "consistent read" and the bookmark event is send when the state is synced at least to the moment when request started being processed.

  - `resourceVersionMatch` set to any other value or unset Invalid error is returned.

  Defaults to true if `resourceVersion=""` or `resourceVersion="0"` (for backward compatibility reasons) and to false otherwise.

- `timeoutSeconds` (*in query*): `integer`
  Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

- `watch` (*in query*): `boolean`
  Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

## `get`

list objects of kind NodeMetrics

### Response

- `200` NodeMetricsList: OK

# NodeMetricsList

NodeMetricsList is a list of NodeMetrics.

- `apiVersion`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
- `items`: `[]NodeMetrics`
  List of node metrics.
- `kind`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `metadata`: ListMeta
  ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

# NodeMetrics

NodeMetrics sets resource usage metrics of a node.

- `apiVersion`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
- `kind`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

- `metadata`: ObjectMeta

  ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.

- `timestamp`: `string`

  Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- `usage`: `map[string]Quantity`

  The memory usage is the memory working set.

- `window`: `string`

  Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

# Time

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

# Quantity

Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML, in addition to String() and AsInt64() accessors.

The serialization format is:

````
    (Note that <suffix> may be empty, from the "" case in <decimalSI>.)

<digit>           ::= 0 | 1 | ... | 9 <digits>          ::= <digit> | <digit><d

    (International System of units; See: http://physics.nist.gov/cuu/Units/bina

<decimalSI>       ::= m | "" | k | M | G | T | P | E

    (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.)

<decimalExponent> ::= "e" <signedNumber> | "E" <signedNumber> ```

No matter which of the three exponent forms is used, no quantity may represent 

When a Quantity is parsed from a string, it will remember the type of suffix it

Before serializing, Quantity will be put in "canonical form". This means that E

- No precision is lost - No fractional digits will be emitted - The exponent (o

The sign will be omitted unless the number is negative.

Examples:

- 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi"

Note that the quantity will NEVER be internally represented by a floating point

Non-canonical values will still parse as long as they are well formed, but will

This format is intended to make it difficult to use these numbers without writi
````

# Duration

Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

# ListMeta

ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

- `continue`: `string`
  continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a consistent list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response, unless you have received this token from an error message.
- `remainingItemCount`: `integer`
  remainingItemCount is the number of subsequent items in the list which are not included in this list response. If the list request contained label or field selectors, then the number of remaining items is unknown and the field will be left unset and omitted during serialization. If the list is complete (either because it is not chunking or because this is the last chunk), then there are no more remaining items and this field will be left unset and omitted during serialization. Servers older than v1.15 do not set this field. The intended use of the remainingItemCount is *estimating* the size of a collection. Clients should not rely on the remainingItemCount to be set or to be exact.
- `resourceVersion`: `string`
  String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
- `selfLink`: `string`
  Deprecated: selfLink is a legacy read-only field that is no longer populated by the system.

# /kubernetes/{cluster}/apis/metrics.k8s.io/v1beta1/nodes/{name}

## Common Parameters

- `name` (*in path*): `string` required
  name of the NodeMetrics
- `pretty` (*in query*): `string`
  If 'true', then the output is pretty printed. Defaults to 'false' unless the user-agent indicates a browser or command-line HTTP tool (curl and wget).

## `get`

read the specified NodeMetrics

### Response

- `200` NodeMetrics: OK

# NodeMetrics

NodeMetrics sets resource usage metrics of a node.

- `apiVersion`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
- `kind`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `metadata`: ObjectMeta
  ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
- `timestamp`: `string`

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- `usage` : `map[string]Quantity`

  The memory usage is the memory working set.

- `window` : `string`

  Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

## Time

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

## Quantity

Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML, in addition to String() and AsInt64() accessors.

The serialization format is:

~~~~
    (Note that <suffix> may be empty, from the "" case in <decimalSI>.)

<digit>           ::= 0 | 1 | ... | 9 <digits>          ::= <digit> | <digit><d

    (International System of units; See: http://physics.nist.gov/cuu/Units/bina

<decimalSI>       ::= m | "" | k | M | G | T | P | E

    (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.)

<decimalExponent> ::= "e" <signedNumber> | "E" <signedNumber> ```

No matter which of the three exponent forms is used, no quantity may represent 

When a Quantity is parsed from a string, it will remember the type of suffix it

Before serializing, Quantity will be put in "canonical form". This means that E

- No precision is lost - No fractional digits will be emitted - The exponent (o

The sign will be omitted unless the number is negative.

Examples:

- 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi"

Note that the quantity will NEVER be internally represented by a floating point

Non-canonical values will still parse as long as they are well formed, but will

This format is intended to make it difficult to use these numbers without writi
~~~~

# Duration

Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

# Metrics Pod [v1beta1]

## /kubernetes/{cluster}/apis/metrics.k8s.io/v1beta1/namespaces/{namespace}/pods

### Common Parameters

- `allowWatchBookmarks` (*in query*): `boolean`
  allowWatchBookmarks requests watch events with type "BOOKMARK". Servers that do not implement bookmarks may ignore this flag and bookmarks are sent at the server's discretion. Clients should not assume bookmarks are returned at any specific interval, nor may they assume the server will send any BOOKMARK event during a session. If this is not a watch, this field is ignored.
- `continue` (*in query*): `string`
  The continue option should be set when retrieving more results from the server. Since this value is server defined, clients may only use the continue value from a previous query result with identical query parameters (except for the value of continue) and the server may reject a continue value it does not recognize. If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error together with a continue token. If the client needs a consistent list, it must restart their list without the continue field. Otherwise, the client may send another list request with the token received with the 410 error, the server will respond with a list starting from the next key, but from the latest snapshot, which is inconsistent from the previous list results - objects that are created, modified, or deleted after the first list request will be included in the response, as long as their keys are after the "next key".

This field is not supported when watch is true. Clients may start a watch from the last resourceVersion value returned by the server and not miss any modifications.

- `fieldSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their fields. Defaults to everything.
- `labelSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their labels. Defaults to everything.
- `limit` (*in query*): `integer`
  limit is a maximum number of responses to return for a list call. If more items exist, the server will set the `continue` field on the list metadata to a value that can be used with the same initial query to retrieve the next set of results. Setting a limit may return fewer than the requested amount of items (up to zero items) in the event all requested objects are filtered out and clients should only use the presence of the continue field to determine whether more results are available. Servers may choose not to support the limit argument and will return all of the available results. If limit is specified and the continue field is empty, clients may assume that no more results are available. This field is not supported if watch is true.
  The server guarantees that the objects returned when using continue will be identical to issuing a single list call without a limit - that is, no objects created, modified, or deleted after the first request is issued will be included in any subsequent continued requests. This is sometimes referred to as a consistent snapshot, and ensures that a client that is using limit to receive smaller chunks of a very large result can ensure they see all possible objects. If objects are updated during a chunked list the version of the object that was present at the time the first list result was calculated is returned.
- `namespace` (*in path*): `string` required
  object name and auth scope, such as for teams and projects
- `pretty` (*in query*): `string`
  If 'true', then the output is pretty printed. Defaults to 'false' unless the user-agent indicates a browser or command-line HTTP tool (curl and wget).
- `resourceVersion` (*in query*): `string`
  resourceVersion sets a constraint on what resource versions a request may be served from. See https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions ↗ for details.
  Defaults to unset
- `resourceVersionMatch` (*in query*): `string`

resourceVersionMatch determines how resourceVersion is applied to list calls. It is highly recommended that resourceVersionMatch be set for list calls where resourceVersion is set See [https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions](https://kubernetes.io/docs/reference/using-api/api-concepts/#resource-versions) ↗ for details.
Defaults to unset

- `sendInitialEvents` (*in query*): `boolean`
  `sendInitialEvents=true` may be set together with `watch=true`. In that case, the watch stream will begin with synthetic events to produce the current state of objects in the collection. Once all such events have been sent, a synthetic "Bookmark" event will be sent. The bookmark will report the ResourceVersion (RV) corresponding to the set of objects, and be marked with `"k8s.io/initial-events-end": "true"` annotation. Afterwards, the watch stream will proceed as usual, sending watch events corresponding to changes (subsequent to the RV) to objects watched.
  When `sendInitialEvents` option is set, we require `resourceVersionMatch` option to also be set. The semantic of the watch request is as following: - `resourceVersionMatch` = NotOlderThan is interpreted as "data at least as new as the provided `resourceVersion`" and the bookmark event is send when the state is synced to a `resourceVersion` at least as fresh as the one provided by the ListOptions. If `resourceVersion` is unset, this is interpreted as "consistent read" and the bookmark event is send when the state is synced at least to the moment when request started being processed.

  - `resourceVersionMatch` set to any other value or unset Invalid error is returned.

  Defaults to true if `resourceVersion=""` or `resourceVersion="0"` (for backward compatibility reasons) and to false otherwise.

- `timeoutSeconds` (*in query*): `integer`
  Timeout for the list/watch call. This limits the duration of the call, regardless of any activity or inactivity.

- `watch` (*in query*): `boolean`
  Watch for changes to the described resources and return them as a stream of add, update, and remove notifications. Specify resourceVersion.

## `get`

list objects of kind PodMetrics

### Response

- `200` PodMetricsList: OK

# PodMetricsList

PodMetricsList is a list of PodMetrics.

- `apiVersion`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
- `items`: `[]PodMetrics`
  List of pod metrics.
- `kind`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `metadata`: ListMeta
  ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

# PodMetrics

PodMetrics sets resource usage metrics of a pod.

- `apiVersion`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗
- `containers`: `[]ContainerMetrics`

Metrics for all containers are collected within the same time window.

- `kind` : `string`

  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

- `metadata` : ObjectMeta

  ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.

- `timestamp` : `string`

  Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

- `window` : `string`

  Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

## ContainerMetrics

ContainerMetrics sets resource usage metrics of a container.

- `name` : `string`

  Container name corresponding to the one from pod.spec.containers.

- `usage` : `map[string]Quantity`

  The memory usage is the memory working set.

## Quantity

Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML, in addition to String() and AsInt64() accessors.

The serialization format is:

```
    (Note that <suffix> may be empty, from the "" case in <decimalSI>.)

<digit>           ::= 0 | 1 | ... | 9 <digits>          ::= <digit> | <digit><d

    (International System of units; See: http://physics.nist.gov/cuu/Units/bina

<decimalSI>       ::= m | "" | k | M | G | T | P | E

    (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.)

<decimalExponent> ::= "e" <signedNumber> | "E" <signedNumber> ```

No matter which of the three exponent forms is used, no quantity may represent 

When a Quantity is parsed from a string, it will remember the type of suffix it

Before serializing, Quantity will be put in "canonical form". This means that E

- No precision is lost - No fractional digits will be emitted - The exponent (o

The sign will be omitted unless the number is negative.

Examples:

- 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi"

Note that the quantity will NEVER be internally represented by a floating point

Non-canonical values will still parse as long as they are well formed, but will

This format is intended to make it difficult to use these numbers without writi
```

# Time

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

# Duration

Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

# ListMeta

ListMeta describes metadata that synthetic resources must have, including lists and various status objects. A resource may have only one of {ObjectMeta, ListMeta}.

- `continue`: `string`
  continue may be set if the user set a limit on the number of items returned, and indicates that the server has more data available. The value is opaque and may be used to issue another request to the endpoint that served this list to retrieve the next set of available objects. Continuing a consistent list may not be possible if the server configuration has changed or more than a few minutes have passed. The resourceVersion field returned when using this continue value will be identical to the value in the first response, unless you have received this token from an error message.
- `remainingItemCount`: `integer`
  remainingItemCount is the number of subsequent items in the list which are not included in this list response. If the list request contained label or field selectors, then the number of remaining items is unknown and the field will be left unset and omitted during serialization. If the list is complete (either because it is not chunking or because this is the last chunk), then there are no more remaining items and this field will be left unset and omitted during serialization. Servers older than v1.15 do not set this field. The intended use of the remainingItemCount is *estimating* the size of a collection. Clients should not rely on the remainingItemCount to be set or to be exact.
- `resourceVersion`: `string`
  String that identifies the server's internal version of this object that can be used by clients to determine when objects have changed. Value must be treated as opaque by clients and passed unmodified back to the server. Populated by the system. Read-only. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#concurrency-control-and-consistency ↗
- `selfLink`: `string`

Deprecated: selfLink is a legacy read-only field that is no longer populated by the system.

# /kubernetes/{cluster}/apis/metrics.k8s.io/v1beta1/namespaces/{namespace}/pods/{name}

## Common Parameters

- `name` (*in path*): `string` required
  name of the PodMetrics
- `namespace` (*in path*): `string` required
  object name and auth scope, such as for teams and projects
- `pretty` (*in query*): `string`
  If 'true', then the output is pretty printed. Defaults to 'false' unless the user-agent indicates a browser or command-line HTTP tool (curl and wget).

## `get`

read the specified PodMetrics

### Response

- `200` PodMetrics: OK

# PodMetrics

PodMetrics sets resource usage metrics of a pod.

- `apiVersion`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources ↗

- `containers`: `[]ContainerMetrics`
  Metrics for all containers are collected within the same time window.
- `kind`: `string`
  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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `metadata`: ObjectMeta
  ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
- `timestamp`: `string`
  Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.
- `window`: `string`
  Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.

# ContainerMetrics

ContainerMetrics sets resource usage metrics of a container.

- `name`: `string`
  Container name corresponding to the one from pod.spec.containers.
- `usage`: `map[string]Quantity`
  The memory usage is the memory working set.

# Quantity

Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON and YAML, in addition to String() and AsInt64() accessors.

The serialization format is:

```
    (Note that <suffix> may be empty, from the "" case in <decimalSI>.)

<digit>           ::= 0 | 1 | ... | 9 <digits>          ::= <digit> | <digit><d

    (International System of units; See: http://physics.nist.gov/cuu/Units/bina

<decimalSI>       ::= m | "" | k | M | G | T | P | E

    (Note that 1024 = 1Ki but 1000 = 1k; I didn't choose the capitalization.)

<decimalExponent> ::= "e" <signedNumber> | "E" <signedNumber> ```

No matter which of the three exponent forms is used, no quantity may represent 

When a Quantity is parsed from a string, it will remember the type of suffix it

Before serializing, Quantity will be put in "canonical form". This means that E

- No precision is lost - No fractional digits will be emitted - The exponent (o

The sign will be omitted unless the number is negative.

Examples:

- 1.5 will be serialized as "1500m" - 1.5Gi will be serialized as "1536Mi"

Note that the quantity will NEVER be internally represented by a floating point

Non-canonical values will still parse as long as they are well formed, but will

This format is intended to make it difficult to use these numbers without writi
```

# Time

Time is a wrapper around time.Time which supports correct marshaling to YAML and JSON. Wrappers are provided for many of the factory methods that the time package offers.

# Duration

Duration is a wrapper around time.Duration which supports correct marshaling to YAML and JSON. In particular, it marshals into strings, which can be used as map keys in json.