AutoScaling APIs

HorizontalPodAutoscaler [autoscaling/v2]

HorizontalPodAutoscaler [autoscaling/v2]

描述

HorizontalPodAutoscaler is the configuration for a horizontal pod autoscaler, which automatically manages the replica count of any resource implementing the scale subresource based on the metrics specified.

类型

`object`

规格

属性	类型	描述
`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.
`spec`	`object`	HorizontalPodAutoscalerSpec describes the desired functionality of the HorizontalPodAutoscaler.
`status`	`object`	HorizontalPodAutoscalerStatus describes the current status of a horizontal pod autoscaler.

.spec

描述

HorizontalPodAutoscalerSpec describes the desired functionality of the HorizontalPodAutoscaler.

类型

`object`

必填

`scaleTargetRef` `maxReplicas`

属性	类型	描述
`behavior`	`object`	HorizontalPodAutoscalerBehavior configures the scaling behavior of the target in both Up and Down directions (scaleUp and scaleDown fields respectively).
`maxReplicas`	`integer`	maxReplicas is the upper limit for the number of replicas to which the autoscaler can scale up. It cannot be less that minReplicas.
`metrics`	`array`	metrics contains the specifications for which to use to calculate the desired replica count (the maximum replica count across all metrics will be used). The desired replica count is calculated multiplying the ratio between the target value and the current value by the current number of pods. Ergo, metrics used must decrease as the pod count is increased, and vice-versa. See the individual metric source types for more information about how each type of metric must respond. If not set, the default metric will be set to 80% average CPU utilization.
`minReplicas`	`integer`	minReplicas is the lower limit for the number of replicas to which the autoscaler can scale down. It defaults to 1 pod. minReplicas is allowed to be 0 if the alpha feature gate HPAScaleToZero is enabled and at least one Object or External metric is configured. Scaling is active as long as at least one metric value is available.
`scaleTargetRef`	`object`	CrossVersionObjectReference contains enough information to let you identify the referred resource.

.spec.behavior

描述

HorizontalPodAutoscalerBehavior configures the scaling behavior of the target in both Up and Down directions (scaleUp and scaleDown fields respectively).

类型

`object`

属性	类型	描述
`scaleDown`	`object`	HPAScalingRules configures the scaling behavior for one direction. These Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen.
`scaleUp`	`object`	HPAScalingRules configures the scaling behavior for one direction. These Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen.

.spec.behavior.scaleDown

描述

HPAScalingRules configures the scaling behavior for one direction. These Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen.

类型

`object`

属性	类型	描述
`policies`	`array`	policies is a list of potential scaling polices which can be used during scaling. At least one policy must be specified, otherwise the HPAScalingRules will be discarded as invalid
`selectPolicy`	`string`	selectPolicy is used to specify which policy should be used. If not set, the default value Max is used.
`stabilizationWindowSeconds`	`integer`	stabilizationWindowSeconds is the number of seconds for which past recommendations should be considered while scaling up or scaling down. StabilizationWindowSeconds must be greater than or equal to zero and less than or equal to 3600 (one hour). If not set, use the default values: - For scale up: 0 (i.e. no stabilization is done). - For scale down: 300 (i.e. the stabilization window is 300 seconds long).

.spec.behavior.scaleDown.policies

描述

policies is a list of potential scaling polices which can be used during scaling. At least one policy must be specified, otherwise the HPAScalingRules will be discarded as invalid

类型

`array`

.spec.behavior.scaleDown.policies[]

描述

HPAScalingPolicy is a single policy which must hold true for a specified past interval.

类型

`object`

必填

`type` `value` `periodSeconds`

属性	类型	描述
`periodSeconds`	`integer`	periodSeconds specifies the window of time for which the policy should hold true. PeriodSeconds must be greater than zero and less than or equal to 1800 (30 min).
`type`	`string`	type is used to specify the scaling policy.

属性	类型	描述
`value`	`integer`	value contains the amount of change which is permitted by the policy. It must be greater than zero

.spec.behavior.scaleUp

描述

HPAScalingRules configures the scaling behavior for one direction. These Rules are applied after calculating DesiredReplicas from metrics for the HPA. They can limit the scaling velocity by specifying scaling policies. They can prevent flapping by specifying the stabilization window, so that the number of replicas is not set instantly, instead, the safest value from the stabilization window is chosen.

类型

`object`

属性	类型	描述
`policies`	`array`	policies is a list of potential scaling polices which can be used during scaling. At least one policy must be specified, otherwise the HPAScalingRules will be discarded as invalid
`selectPolicy`	`string`	selectPolicy is used to specify which policy should be used. If not set, the default value Max is used.
`stabilizationWindowSeconds`	`integer`	stabilizationWindowSeconds is the number of seconds for which past recommendations should be considered while scaling up or scaling down. StabilizationWindowSeconds must be greater than or equal to zero and less than or equal to 3600 (one hour). If not set, use the default values: - For scale up: 0 (i.e. no stabilization is done). - For scale down: 300 (i.e. the stabilization window is 300 seconds long).

.spec.behavior.scaleUp.policies

描述

policies is a list of potential scaling polices which can be used during scaling. At least one policy must be specified, otherwise the HPAScalingRules will be discarded as invalid

类型

`array`

.spec.behavior.scaleUp.policies[]

描述

HPAScalingPolicy is a single policy which must hold true for a specified past interval.

类型

`object`

必填

`type` `value` `periodSeconds`

属性	类型	描述
`periodSeconds`	`integer`	periodSeconds specifies the window of time for which the policy should hold true. PeriodSeconds must be greater than zero and less than or equal to 1800 (30 min).
`type`	`string`	type is used to specify the scaling policy.
`value`	`integer`	value contains the amount of change which is permitted by the policy. It must be greater than zero

.spec.metrics

描述

metrics contains the specifications for which to use to calculate the desired replica count (the maximum replica count across all metrics will be used). The desired replica count is calculated multiplying the ratio between the target value and the current value by the current number of pods. Ergo, metrics used must decrease as the pod count is increased, and vice-versa. See the individual metric source types for more information about how each type of metric must respond. If not set, the default metric will be set to 80% average CPU utilization.

类型

`array`

.spec.metrics[]

描述

MetricSpec specifies how to scale based on a single metric (only `type` and one other matching field should be set at once).

类型

`object`

必填

`type`

属性	类型	描述
`containerResource`	`object`	ContainerResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.
`external`	`object`	ExternalMetricSource indicates how to scale on a metric not associated with any Kubernetes object (for example length of queue in cloud messaging service, or QPS from loadbalancer running outside of cluster).
`object`	`object`	ObjectMetricSource indicates how to scale on a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).

属性	类型	描述
`pods`	`object`	PodsMetricSource indicates how to scale on a metric describing each pod in the current scale target (for example, transactions-processed-per-second). The values will be averaged together before being compared to the target value.
`resource`	`object`	ResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.
`type`	`string`	type is the type of metric source. It should be one of "ContainerResource", "External", "Object", "Pods" or "Resource", each mapping to a matching field in the object.

.spec.metrics[].containerResource

描述

ContainerResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.

类型

`object`

必填

`name` `target` `container`

属性	类型	描述
`container`	`string`	container is the name of the container in the pods of the scaling target
`name`	`string`	name is the name of the resource in question.
`target`	`object`	MetricTarget defines the target value, average value, or average utilization of a specific metric

.spec.metrics[].containerResource.target

描述

MetricTarget defines the target value, average value, or average utilization of a specific metric

类型

`object`

必填

`type`

属性	类型	描述
`averageUtilization`	`integer`	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`type`	`string`	type represents whether the metric type is Utilization, Value, or AverageValue
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.spec.metrics[].external

描述

ExternalMetricSource indicates how to scale on a metric not associated with any Kubernetes object (for example length of queue in cloud messaging service, or QPS from loadbalancer running outside of cluster).

类型

`object`

必填

`metric` `target`

属性	类型	描述
`metric`	`object`	MetricIdentifier defines the name and optionally selector for a metric
`target`	`object`	MetricTarget defines the target value, average value, or average utilization of a specific metric

.spec.metrics[].external.metric

描述

MetricIdentifier defines the name and optionally selector for a metric

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	name is the name of the given metric
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.spec.metrics[].external.metric.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`object`	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.metrics[].external.metric.selector.matchExpressions

描述

matchExpressions is a list of label selector requirements. The requirements are ANDed.

类型

`array`

.spec.metrics[].external.metric.selector.matchExpressions[]

描述

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`values`	`array`	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.metrics[].external.metric.selector.matchExpressions[].values

描述

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

类型

`array`

.spec.metrics[].external.metric.selector.matchExpressions[].values[]

类型

`string`

.spec.metrics[].external.metric.selector.matchLabels

描述

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

类型

`object`

.spec.metrics[].external.target

描述

MetricTarget defines the target value, average value, or average utilization of a specific metric

类型

`object`

必填

`type`

属性	类型	描述
`averageUtilization`	`integer`	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented

属性	类型	描述
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`type`	`string`	type represents whether the metric type is Utilization, Value, or AverageValue
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.spec.metrics[].object

描述

ObjectMetricSource indicates how to scale on a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).

类型

`object`

必填

`describedObject` `target` `metric`

属性	类型	描述
`describedObject`	`object`	CrossVersionObjectReference contains enough information to let you identify the referred resource.
`metric`	`object`	MetricIdentifier defines the name and optionally selector for a metric
`target`	`object`	MetricTarget defines the target value, average value, or average utilization of a specific metric

.spec.metrics[].object.describedObject

描述

CrossVersionObjectReference contains enough information to let you identify the referred resource.

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiVersion`	`string`	apiVersion is the API version of the referent
`kind`	`string`	kind is the kind of the referent; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
`name`	`string`	name is the name of the referent; More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

.spec.metrics[].object.metric

描述

MetricIdentifier defines the name and optionally selector for a metric

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	name is the name of the given metric
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.spec.metrics[].object.metric.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`object`	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.metrics[].object.metric.selector.matchExpressions

描述

matchExpressions is a list of label selector requirements. The requirements are ANDed.

类型

`array`

.spec.metrics[].object.metric.selector.matchExpressions[]

描述

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`values`	`array`	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.metrics[].object.metric.selector.matchExpressions[].values

描述

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

类型

`array`

.spec.metrics[].object.metric.selector.matchExpressions[].values[]

类型

`string`

.spec.metrics[].object.metric.selector.matchLabels

描述

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

类型

`object`

.spec.metrics[].object.target

描述

MetricTarget defines the target value, average value, or average utilization of a specific metric

类型

`object`

必填

`type`

属性	类型	描述
`averageUtilization`	`integer`	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`type`	`string`	type represents whether the metric type is Utilization, Value, or AverageValue
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.spec.metrics[].pods

描述

PodsMetricSource indicates how to scale on a metric describing each pod in the current scale target (for example, transactions-processed-per-second). The values will be averaged together before being compared to the target value.

类型

`object`

必填

`metric` `target`

属性	类型	描述
`metric`	`object`	MetricIdentifier defines the name and optionally selector for a metric
`target`	`object`	MetricTarget defines the target value, average value, or average utilization of a specific metric

.spec.metrics[].pods.metric

描述

MetricIdentifier defines the name and optionally selector for a metric

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	name is the name of the given metric
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.spec.metrics[].pods.metric.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`object`	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.spec.metrics[].pods.metric.selector.matchExpressions

描述

matchExpressions is a list of label selector requirements. The requirements are ANDed.

类型

`array`

.spec.metrics[].pods.metric.selector.matchExpressions[]

描述

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`values`	`array`	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.spec.metrics[].pods.metric.selector.matchExpressions[].values

描述

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

类型

`array`

.spec.metrics[].pods.metric.selector.matchExpressions[].values[]

类型

`string`

.spec.metrics[].pods.metric.selector.matchLabels

描述

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

类型

`object`

.spec.metrics[].pods.target

描述

MetricTarget defines the target value, average value, or average utilization of a specific metric

类型

`object`

必填

`type`

属性	类型	描述
`averageUtilization`	`integer`	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented

属性	类型	描述
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`type`	`string`	type represents whether the metric type is Utilization, Value, or AverageValue
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.spec.metrics[].resource

描述

ResourceMetricSource indicates how to scale on a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). The values will be averaged together before being compared to the target. Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source. Only one "target" type should be set.

类型

`object`

必填

`name` `target`

属性	类型	描述
`name`	`string`	name is the name of the resource in question.
`target`	`object`	MetricTarget defines the target value, average value, or average utilization of a specific metric

.spec.metrics[].resource.target

描述

MetricTarget defines the target value, average value, or average utilization of a specific metric

类型

`object`

必填

`type`

属性	类型	描述
`averageUtilization`	`integer`	averageUtilization is the target value of the average of the resource metric across all relevant pods, represented
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`type`	`string`	type represents whether the metric type is Utilization, Value, or AverageValue
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a

属性	类型	描述
		The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.spec.scaleTargetRef

描述

CrossVersionObjectReference contains enough information to let you identify the referred resource.

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiVersion`	`string`	apiVersion is the API version of the referent
`kind`	`string`	kind is the kind of the referent; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

属性	类型	描述
`name`	`string`	name is the name of the referent; More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.status

描述

HorizontalPodAutoscalerStatus describes the current status of a horizontal pod autoscaler.

类型

`object`

必填

`desiredReplicas`

属性	类型	描述
`conditions`	`array`	conditions is the set of conditions required for this autoscaler to scale its target, and indicates whether or not those conditions are met.
`currentMetrics`	`array`	currentMetrics is the last read state of the metrics used by this autoscaler.
`currentReplicas`	`integer`	currentReplicas is current number of replicas of pods managed by this autoscaler, as last seen by the autoscaler.
`desiredReplicas`	`integer`	desiredReplicas is the desired number of replicas of pods managed by this autoscaler, as last calculated by the autoscaler.
`lastScaleTime`	`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.
`observedGeneration`	`integer`	observedGeneration is the most recent generation observed by this autoscaler.

.status.conditions

描述

conditions is the set of conditions required for this autoscaler to scale its target, and indicates whether or not those conditions are met.

类型

`array`

.status.conditions[]

描述

HorizontalPodAutoscalerCondition describes the state of a HorizontalPodAutoscaler at a certain point.

类型

`object`

必填

`type` `status`

属性	类型	描述
`lastTransitionTime`	`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.
`message`	`string`	message is a human-readable explanation containing details about the transition
`reason`	`string`	reason is the reason for the condition's last transition.
`status`	`string`	status is the status of the condition (True, False, Unknown)
`type`	`string`	type describes the current condition

.status.currentMetrics

描述

currentMetrics is the last read state of the metrics used by this autoscaler.

类型

`array`

.status.currentMetrics[]

描述

MetricStatus describes the last-read state of a single metric.

类型

`object`

必填

`type`

属性	类型	描述
`containerResource`	`object`	ContainerResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing a single container in each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.

属性	类型	描述
`external`	`object`	ExternalMetricStatus indicates the current value of a global metric not associated with any Kubernetes object.
`object`	`object`	ObjectMetricStatus indicates the current value of a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).
`pods`	`object`	PodsMetricStatus indicates the current value of a metric describing each pod in the current scale target (for example, transactions-processed-per-second).
`resource`	`object`	ResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.
`type`	`string`	type is the type of metric source. It will be one of "ContainerResource", "External", "Object", "Pods" or "Resource", each corresponds to a matching field in the object.

.status.currentMetrics[].containerResource

描述

ContainerResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing a single container in each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.

类型

`object`

必填

`name` `current` `container`

属性	类型	描述
`container`	`string`	container is the name of the container in the pods of the scaling target
`current`	`object`	MetricValueStatus holds the current value for a metric
`name`	`string`	name is the name of the resource in question.

.status.currentMetrics[].containerResource.current

描述

MetricValueStatus holds the current value for a metric

类型

`object`

属性	类型	描述
`averageUtilization`	`integer`	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, rep
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.status.currentMetrics[].external

描述

ExternalMetricStatus indicates the current value of a global metric not associated with any Kubernetes object.

类型

`object`

必填

`metric` `current`

属性	类型	描述
`current`	`object`	MetricValueStatus holds the current value for a metric
`metric`	`object`	MetricIdentifier defines the name and optionally selector for a metric

.status.currentMetrics[].external.current

描述

MetricValueStatus holds the current value for a metric

类型

`object`

属性	类型	描述
`averageUtilization`	`integer`	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, rep
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.status.currentMetrics[].external.metric

描述

MetricIdentifier defines the name and optionally selector for a metric

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	name is the name of the given metric
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.status.currentMetrics[].external.metric.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`object`	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.status.currentMetrics[].external.metric.selector.matchExpressions

描述

matchExpressions is a list of label selector requirements. The requirements are ANDed.

类型

`array`

.status.currentMetrics[].external.metric.selector.matchExpressions[]

描述

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`values`	`array`	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.status.currentMetrics[].external.metric.selector.matchExpressions[].values

描述

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

类型

`array`

.status.currentMetrics[].external.metric.selector.matchExpressions[].values[]

类型

`string`

.status.currentMetrics[].external.metric.selector.matchLabels

描述

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

类型

`object`

.status.currentMetrics[].object

描述

ObjectMetricStatus indicates the current value of a metric describing a kubernetes object (for example, hits-per-second on an Ingress object).

类型

`object`

必填

`metric` `current` `describedObject`

属性	类型	描述
`current`	`object`	MetricValueStatus holds the current value for a metric
`describedObject`	`object`	CrossVersionObjectReference contains enough information to let you identify the referred resource.
`metric`	`object`	MetricIdentifier defines the name and optionally selector for a metric

.status.currentMetrics[].object.current

描述

MetricValueStatus holds the current value for a metric

类型

`object`

属性	类型	描述
`averageUtilization`	`integer`	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, rep

属性	类型	描述
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.status.currentMetrics[].object.describedObject

描述

CrossVersionObjectReference contains enough information to let you identify the referred resource.

类型

`object`

必填

`kind` `name`

属性	类型	描述
`apiVersion`	`string`	apiVersion is the API version of the referent
`kind`	`string`	kind is the kind of the referent; More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗

属性	类型	描述
`name`	`string`	name is the name of the referent; More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

.status.currentMetrics[].object.metric

描述

MetricIdentifier defines the name and optionally selector for a metric

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	name is the name of the given metric
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.status.currentMetrics[].object.metric.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`object`	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.status.currentMetrics[].object.metric.selector.matchExpressions

描述

matchExpressions is a list of label selector requirements. The requirements are ANDed.

类型

`array`

.status.currentMetrics[].object.metric.selector.matchExpressions[]

描述

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`values`	`array`	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.status.currentMetrics[].object.metric.selector.matchExpressions[].values

描述

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

类型

`array`

.status.currentMetrics[].object.metric.selector.matchExpressions[].values[]

类型

`string`

.status.currentMetrics[].object.metric.selector.matchLabels

描述

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

类型

`object`

.status.currentMetrics[].pods

描述

PodsMetricStatus indicates the current value of a metric describing each pod in the current scale target (for example, transactions-processed-per-second).

类型

`object`

必填

`metric` `current`

属性	类型	描述
`current`	`object`	MetricValueStatus holds the current value for a metric
`metric`	`object`	MetricIdentifier defines the name and optionally selector for a metric

.status.currentMetrics[].pods.current

描述

MetricValueStatus holds the current value for a metric

类型

`object`

属性	类型	描述
`averageUtilization`	`integer`	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, rep
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

.status.currentMetrics[].pods.metric

描述

MetricIdentifier defines the name and optionally selector for a metric

类型

`object`

必填

`name`

属性	类型	描述
`name`	`string`	name is the name of the given metric
`selector`	`object`	A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

.status.currentMetrics[].pods.metric.selector

描述

A label selector is a label query over a set of resources. The result of matchLabels and matchExpressions are ANDed. An empty label selector matches all objects. A null label selector matches no objects.

类型

`object`

属性	类型	描述
`matchExpressions`	`array`	matchExpressions is a list of label selector requirements. The requirements are ANDed.
`matchLabels`	`object`	matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

.status.currentMetrics[].pods.metric.selector.matchExpressions

描述

matchExpressions is a list of label selector requirements. The requirements are ANDed.

类型

`array`

.status.currentMetrics[].pods.metric.selector.matchExpressions[]

描述

A label selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

类型

`object`

必填

`key` `operator`

属性	类型	描述
`key`	`string`	key is the label key that the selector applies to.
`operator`	`string`	operator represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists and DoesNotExist.
`values`	`array`	values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

.status.currentMetrics[].pods.metric.selector.matchExpressions[].values

描述

values is an array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. This array is replaced during a strategic merge patch.

类型

`array`

.status.currentMetrics[].pods.metric.selector.matchExpressions[].values[]

类型

`string`

.status.currentMetrics[].pods.metric.selector.matchLabels

描述

matchLabels is a map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

类型

`object`

.status.currentMetrics[].resource

描述

ResourceMetricStatus indicates the current value of a resource metric known to Kubernetes, as specified in requests and limits, describing each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.

类型

`object`

必填

`name` `current`

属性	类型	描述
`current`	`object`	MetricValueStatus holds the current value for a metric
`name`	`string`	name is the name of the resource in question.

.status.currentMetrics[].resource.current

描述

MetricValueStatus holds the current value for a metric

类型

`object`

属性	类型	描述
`averageUtilization`	`integer`	currentAverageUtilization is the current value of the average of the resource metric across all relevant pods, rep
`averageValue`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a

属性	类型	描述
		The serialization format is: (Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing
`value`	`string|number`	Quantity is a fixed-point representation of a number. It provides convenient marshaling/unmarshaling in JSON a The serialization format is:

属性	类型	描述
		(Note that <suffix> may be empty, from the "" case in <decimalSI>.) <digit> ::= 0 | 1 | ... | 9 <digits> ::= <digit> | <digit><digi (International System of units; See: http://physics.nist.gov/cuu/Units/binary. <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 a n When a Quantity is parsed from a string, it will remember the type of suffix it ha Before serializing, Quantity will be put in "canonical form". This means that Expo - No precision is lost - No fractional digits will be emitted - The exponent (or s 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 nu Non-canonical values will still parse as long as they are well formed, but will be This format is intended to make it difficult to use these numbers without writing

API 端点

可用的 API 端点如下：

- `/kubernetes/{cluster}/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers`
  - `DELETE` : delete collection of HorizontalPodAutoscaler
  - `GET` : list objects of kind HorizontalPodAutoscaler
  - `POST` : create a new HorizontalPodAutoscaler
- `/kubernetes/{cluster}/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}`
  - `DELETE` : delete the specified HorizontalPodAutoscaler
  - `GET` : read the specified HorizontalPodAutoscaler
  - `PATCH` : partially update the specified HorizontalPodAutoscaler
  - `PUT` : replace the specified HorizontalPodAutoscaler
- `/kubernetes/{cluster}/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}/status`
  - `GET` : read status of the specified HorizontalPodAutoscaler
  - `PATCH` : partially update status of the specified HorizontalPodAutoscaler
  - `PUT` : replace status of the specified HorizontalPodAutoscaler

/kubernetes/{cluster}/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers

HTTP 方法

`DELETE`

描述

delete collection of HorizontalPodAutoscaler

HTTP 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

list objects of kind HorizontalPodAutoscaler

HTTP 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscalerList` schema
401 - Unauthorized	Empty

HTTP 方法

`POST`

描述

create a new HorizontalPodAutoscaler

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation 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 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.

请求体参数

参数	类型	描述
`body`	`HorizontalPodAutoscaler` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema
201 - Created	`HorizontalPodAutoscaler` schema
202 - Accepted	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}

HTTP 方法

`DELETE`

描述

delete the specified HorizontalPodAutoscaler

查询参数

参数	类型	描述
`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 响应

HTTP 状态码	响应体
200 - OK	`Status` schema
202 - Accepted	`Status` schema
401 - Unauthorized	Empty

HTTP 方法

`GET`

描述

read the specified HorizontalPodAutoscaler

HTTP 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update the specified HorizontalPodAutoscaler

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation 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 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 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace the specified HorizontalPodAutoscaler

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation 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 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.

请求体参数

参数	类型	描述
`body`	`HorizontalPodAutoscaler` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema

HTTP 状态码	响应体
201 - Created	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty

/kubernetes/{cluster}/apis/autoscaling/v2/namespaces/{namespace}/horizontalpodautoscalers/{name}/status

HTTP 方法

`GET`

描述

read status of the specified HorizontalPodAutoscaler

HTTP 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty

HTTP 方法

`PATCH`

描述

partially update status of the specified HorizontalPodAutoscaler

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation 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 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 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty

HTTP 方法

`PUT`

描述

replace status of the specified HorizontalPodAutoscaler

查询参数

参数	类型	描述
`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
`fieldValidation`	`string`	fieldValidation 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 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.

请求体参数

参数	类型	描述
`body`	`HorizontalPodAutoscaler` schema	`application/json` formatted

HTTP 响应

HTTP 状态码	响应体
200 - OK	`HorizontalPodAutoscaler` schema
201 - Created	`HorizontalPodAutoscaler` schema
401 - Unauthorized	Empty