Cluster Lifecycle APIs

Cluster [platform.tkestack.io/v1 ClusterCredential [platform.tke Machine [pl

Cluster [platform.tkestack.io/v1]

/apis/platform.tkestack.io/v1/clusters

Common Parameters

- `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`

list objects of kind Cluster

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 If the specified continue value is no longer valid whether due to expiration (generally five to fifteen minutes) or a configuration change on the server, the server will respond with a 410 ResourceExpired error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.
- `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.

Response

- `200` ClusterList: OK

`post`

create a Cluster

Parameters

- `dryRun` (*in query*): `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
- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗.
- `fieldValidation` (*in query*): `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.

Request Body

Cluster

Response

- `200` Cluster: OK
- `201` Cluster: Created
- `202` Cluster: Accepted

`delete`

delete collection of Cluster

Parameters

- `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.
- `dryRun` (*in query*): `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
- `fieldSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their fields. Defaults to everything.
- `gracePeriodSeconds` (*in query*): `integer`
  The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace

period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

- `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.
- `orphanDependents` (*in query*): `boolean`
  Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
- `propagationPolicy` (*in query*): `string`
  Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.
- `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.

Request Body

DeleteOptions

Response

- `200` Status: OK

/apis/platform.tkestack.io/v1/clusters/{name}

Common Parameters

- `name` (*in path*): `string` required
  name of the Cluster
- `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 Cluster

Response

- `200` Cluster: OK

`put`

replace the specified Cluster

Parameters

- `dryRun` (*in query*): `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
- `fieldManager` (*in query*): `string`

fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗.

- `fieldValidation` (*in query*): `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.

Request Body

Cluster

Response

- `200` Cluster: OK
- `201` Cluster: Created

`delete`

delete a Cluster

Parameters

- `dryRun` (*in query*): `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
- `gracePeriodSeconds` (*in query*): `integer`

The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

- `orphanDependents` (*in query*): `boolean`
  Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
- `propagationPolicy` (*in query*): `string`
  Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Request Body

DeleteOptions

Response

- `200` Status: OK
- `202` Status: Accepted

`patch`

partially update the specified Cluster

Parameters

- `dryRun` (*in query*): `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

- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).
- `fieldValidation` (*in query*): `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.
- `force` (*in query*): `boolean`
  Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Request Body

Patch

Response

- `200` Cluster: OK
- `201` Cluster: Created

ClusterList

ClusterList is the whole list of all clusters which owned by a tenant.

- `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`: `[]Cluster`
  List of clusters
- `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}.

Cluster

Cluster is a Kubernetes cluster in

- `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`: ClusterSpec

ClusterSpec is a description of a cluster.

- `status` : ClusterStatus
  ClusterStatus represents information about the status of a cluster.

ClusterSpec

ClusterSpec is a description of a cluster.

- `apiServerExtraArgs` : `map[string]string`
  api server extra arguments
- `clusterCIDR` : `string`
  ClusterCIDR is used to set a separated CIDR for k8s pod
- `clusterCredentialRef` : LocalObjectReference
  LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
- `controllerManagerExtraArgs` : `map[string]string`
  controller manager extra arguments
- `displayName` : `string`
  DisplayName is the display name of the cluster.
- `dnsDomain` : `string`
  DNSDomain is the dns domain used by k8s services. Defaults to "cluster.local".
- `dockerExtraArgs` : `map[string]string`
  docker extra arguments
- `etcd` : Etcd
  Etcd contains elements describing Etcd configuration.
- `features` : ClusterFeature
  ClusterFeature records the features that are enabled by the cluster.
- `finalizers` : `[]string`
  Finalizers is an opaque list of values that must be empty to permanently remove object from storage.
- `hostnameAsNodename` : `boolean`
  If true will use hostname as nodename, if false will use machine IP as nodename.

- `kubeletExtraArgs` : `map[string]string`
  kubelet extra arguments
- `machines` : `[]ClusterMachine`
  cluster machine, represent master nodes in the cluster
- `networkArgs` : `map[string]string`
  network arguments
- `networkDevice` : `string`
  network device
- `networkType` : `string`
  network type
- `properties` : ClusterProperty
  ClusterProperty records the attribute information of the cluster.
- `publicAlternativeNames` : `[]string`
  public alternative names
- `scalingMachines` : `[]ClusterMachine`
  ScalingMachines is a set of the machines that we want to add/delete from the cluster.
- `schedulerExtraArgs` : `map[string]string`
  scheduler extra arguments
- `serviceCIDR` : `string`
  ServiceCIDR is used to set a separated CIDR for k8s service, it's exclusive with MaxClusterServiceNum.
- `tenantID` : `string`
  tenant id
- `type` : `string`
  cluster type, baremetal or imported
- `version` : `string`
  cluster version

LocalObjectReference

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

- `name` : `string`
  Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗

Etcd

Etcd contains elements describing Etcd configuration.

- `external` : ExternalEtcd
  ExternalEtcd describes an external etcd cluster. Kubeadm has no knowledge of where certificate files live and they must be supplied.
- `local` : LocalEtcd
  LocalEtcd describes that kubeadm should run an etcd cluster locally

ExternalEtcd

ExternalEtcd describes an external etcd cluster. Kubeadm has no knowledge of where certificate files live and they must be supplied.

- `caFile` : `string`
  CAFile is an SSL Certificate Authority file used to secure etcd communication. Required if using a TLS connection.
- `certFile` : `string`
  CertFile is an SSL certification file used to secure etcd communication. Required if using a TLS connection.
- `endpoints` : `[]string`
  Endpoints of etcd members. Required for ExternalEtcd.
- `keyFile` : `string`
  KeyFile is an SSL key file used to secure etcd communication. Required if using a TLS connection.

LocalEtcd

LocalEtcd describes that kubeadm should run an etcd cluster locally

- `dataDir` : `string`
  DataDir is the directory etcd will place its data. Defaults to "/var/lib/etcd".
- `extraArgs` : `map[string]string`
  ExtraArgs are extra arguments provided to the etcd binary when run inside a static pod.
- `peerCertSANs` : `[]string`
  PeerCertSANs sets extra Subject Alternative Names for the etcd peer signing cert.
- `serverCertSANs` : `[]string`
  ServerCertSANs sets extra Subject Alternative Names for the etcd server signing cert.

ClusterFeature

ClusterFeature records the features that are enabled by the cluster.

- `authzWebhookAddr` : AuthzWebhookAddr
- `csiOperator` : CSIOperatorFeature
- `enableMasterSchedule` : `boolean`
  enable master node schedule or not
- `enableMetricsServer` : `boolean`
  enable metrics server or not
- `files` : `[]File`
  files
- `gpuType` : `string`
  GPU type
- `ha` : HA
- `hooks` : `map[string]string`
  hooks
- `internalLB` : `boolean`
  internalLB on-off

- `ipv6DualStack` : `boolean`
  enable IPV4/IPV6 dual stack or not
- `ipvs` : `boolean`
  ipvs on-off
- `kubeProxyConfiguration` : `string`
  KubeProxyConfiguration kube-proxy configuration
- `publicLB` : `boolean`
  publicLB on-off
- `skipConditions` : `[]string`
  SkipConditions is a set of conditions that needs to be skipped
- `upgrade` : Upgrade

AuthzWebhookAddr

- `builtin` : BuiltinAuthzWebhookAddr
- `external` : ExternalAuthzWebhookAddr

BuiltinAuthzWebhookAddr

ExternalAuthzWebhookAddr

- `ip` : `string`
  ip
- `port` : `integer`
  port

CSIOperatorFeature

- `version` : `string`
  version

File

- `dst` : `string`
  destination file, only support regular file
- `src` : `string`
  source file, only support regular file

HA

- `thirdParty` : ThirdPartyHA
- `tke` : TKEHA

ThirdPartyHA

- `vip` : `string`
  virtual IP of the HA
- `vport` : `integer`
  virtual port of the HA

TKEHA

- `vip` : `string`
  virtual IP of the HA
- `vrid` : `integer`
  vrid of the HA

Upgrade

- `mode` : `string`
  Upgrade mode, default value is Auto.

- `strategy` : UpgradeStrategy
  UpgradeStrategy used to control the upgrade process.

UpgradeStrategy

UpgradeStrategy used to control the upgrade process.

- `drainNodeBeforeUpgrade` : `boolean`
  Whether drain node before upgrade. Draining node before upgrade is recommended. But not all pod running as cows, a few running as pets. If your pod can not accept be expelled from current node, this value should be false.
- `maxUnready` : IntOrString
  IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

IntOrString

IntOrString is a type that can hold an int32 or a string. When used in JSON or YAML marshalling and unmarshalling, it produces or consumes the inner type. This allows you to have, for example, a JSON field that can accept a name or number.

ClusterMachine

ClusterMachine is the master machine definition of cluster.

- `annotations` : `map[string]string`
  Annotations
- `displayName` : `string`
  DisplayName
- `ip` : `string`
  IP
- `ipv6` : `string`

IPv6

- `labels` : `map[string]string`
- `networkDevice` : `string`
  NetworkDevice
- `passPhrase` : `string`
- `password` : `string`
- `port` : `integer`
- `privateKey` : `string`
- `proxy` : MachineProxy
  MachineProxy is the proxy to connect Machine's ssh.
- `publicIP` : `string`
  PublicIP
- `role` : `string`
  Role master/node
- `taints` : `[]Taint`
  If specified, the node's taints.
- `username` : `string`

MachineProxy

MachineProxy is the proxy to connect Machine's ssh.

- `address` : `string`
- `password` : `string`
- `type` : `string`
- `username` : `string`

Taint

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

- `effect` : `string`
  Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.
  Possible enum values:
  - `"NoExecute"` Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController.
  - `"NoSchedule"` Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler.
  - `"PreferNoSchedule"` Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.
- `key` : `string`
  Required. The taint key to be applied to a node.
- `timeAdded` : `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.
- `value` : `string`
  The taint value corresponding to the taint key.

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.

ClusterProperty

ClusterProperty records the attribute information of the cluster.

- `maxClusterServiceNum` : `integer`
  max number of cluster service

- `maxNodePodNum` : `integer`
  max number of node pod
- `oversoldRatio` : `map[string]string`
  oversold ratio

ClusterStatus

ClusterStatus represents information about the status of a cluster.

- `addresses` : `[]ClusterAddress`
  List of addresses reachable to the cluster.
- `clusterCIDR` : `string`
  ClusterCIDR is used to set a separated CIDR for k8s pod
- `components` : `[]ClusterComponent`
  cluster component
- `conditions` : `[]ClusterCondition`
  cluster conditions
- `dnsIP` : `string`
  dns ip
- `locked` : `boolean`
  lock status of the cluster
- `message` : `string`
  A human readable message indicating details about why the cluster is in this condition.
- `nodeCIDRMaskSize` : `integer`
  node CIDR mask size
- `nodeCIDRMaskSizeIPv4` : `integer`
  node CIDR mask size of the IPV4
- `nodeCIDRMaskSizeIPv6` : `integer`
  node CIDR mask size of the IPV6
- `phase` : `string`
  cluster phase
- `reason` : `string`

A brief CamelCase message indicating details about why the cluster is in this state.

- `registryIPs` : `[]string`
  registry ip
- `resource` : ClusterResource
  ClusterResource records the current available and maximum resource quota information for the cluster.
- `secondaryClusterCIDR` : `string`
  secondary ClusterCIDR, for dual stack IPV4/IPV6.
- `secondaryServiceCIDR` : `string`
  secondary ServiceCIDR, for dual stack IPV4/IPV6.
- `serviceCIDR` : `string`
  ServiceCIDR is used to set a separated CIDR for k8s service, it's exclusive with MaxClusterServiceNum.
- `version` : `string`
  cluster version

ClusterAddress

ClusterAddress contains information for the cluster's address.

- `host` : `string`
  The cluster address.
- `path` : `string`
  path
- `port` : `integer`
  port
- `type` : `string`
  Cluster address type, one of Public, ExternalIP or InternalIP.

ClusterComponent

ClusterComponent records the number of copies of each component of the cluster master.

- `replicas` : ClusterComponentReplicas
  ClusterComponentReplicas records the number of copies of each state of each component of the cluster master.
- `type` : `string`
  type

ClusterComponentReplicas

ClusterComponentReplicas records the number of copies of each state of each component of the cluster master.

- `available` : `integer`
  Available
- `current` : `integer`
  Current
- `desired` : `integer`
  Desired
- `updated` : `integer`
  Updated

ClusterCondition

ClusterCondition contains details for the current condition of this cluster.

- `lastProbeTime` : `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.
- `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`

Human-readable message indicating details about last transition.

- `reason` : `string`
  Unique, one-word, CamelCase reason for the condition's last transition.
- `status` : `string`
  Status is the status of the condition. Can be True, False, Unknown.
- `type` : `string`
  Type is the type of the condition.

ClusterResource

ClusterResource records the current available and maximum resource quota information for the cluster.

- `allocatable` : `map[string]Quantity`
  Allocatable represents the resources of a cluster that are available for scheduling. Defaults to Capacity.
- `allocated` : `map[string]Quantity`
  Allocated represents the resources of a cluster that have been allocated
- `capacity` : `map[string]Quantity`
  Capacity represents the total resources of a cluster.

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 a

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
```

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.

Status

Status is a return value for calls that don't return other objects.

- `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 ↗
- `code` : `integer`

Suggested HTTP return code for this status, 0 if not set.

- `details` : StatusDetails
  StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.
- `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 ↗
- `message` : `string`
  A human-readable description of the status of this operation.
- `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}.
- `reason` : `string`
  A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available. A Reason clarifies an HTTP status code but does not override it.
- `status` : `string`
  Status of the operation. One of: "Success" or "Failure". More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

StatusDetails

StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.

- `causes` : `[]StatusCause`

The Causes array includes more details associated with the StatusReason failure. Not all StatusReasons may provide detailed causes.

- `group` : `string`
  The group attribute of the resource associated with the status StatusReason.
- `kind` : `string`
  The kind attribute of the resource associated with the status StatusReason. On some operations may differ from the requested resource Kind. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `name` : `string`
  The name attribute of the resource associated with the status StatusReason (when there is a single name which can be described).
- `retryAfterSeconds` : `integer`
  If specified, the time in seconds before the operation should be retried. Some errors may indicate the client must take an alternate action - for those errors this field may indicate how long to wait before taking the alternate action.
- `uid` : `string`
  UID of the resource. (when there is a single resource which can be described). More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids ↗

StatusCause

StatusCause provides more information about an api.Status failure, including cases when multiple errors are encountered.

- `field` : `string`
  The field of the resource that has caused this error, as named by its JSON serialization. May include dot and postfix notation for nested attributes. Arrays are zero-indexed. Fields may appear more than once in an array of causes due to fields having multiple errors. Optional.
  Examples: "name" - the field "name" on the current resource "items[0].name" - the field "name" on the first array entry in "items"
- `message` : `string`

A human-readable description of the cause of the error. This field may be presented as-is to a reader.

- `reason` : `string`
  A machine-readable description of the cause of the error. If this value is empty there is no information available.

Patch

Patch is provided to give a concrete name and type to the Kubernetes PATCH request body.

ClusterCredential [platform.tkestack.io/v1]

/apis/platform.tkestack.io/v1/clustercredentials

Common Parameters

- `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`

list objects of kind ClusterCredential

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.
- `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.

Response

- `200` ClusterCredentialList: OK

`post`

create a ClusterCredential

Parameters

- `dryRun` (*in query*): `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
- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗.
- `fieldValidation` (*in query*): `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.

Request Body

ClusterCredential required

Response

- `200` ClusterCredential: OK
- `201` ClusterCredential: Created
- `202` ClusterCredential: Accepted

`delete`

delete collection of ClusterCredential

Parameters

- `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.
- `dryRun` (*in query*): `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
- `fieldSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their fields. Defaults to everything.
- `gracePeriodSeconds` (*in query*): `integer`
  The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace

period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

- `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.
- `orphanDependents` (*in query*): `boolean`
  Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
- `propagationPolicy` (*in query*): `string`
  Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.
- `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.

Request Body

DeleteOptions

Response

- `200` Status: OK

/apis/platform.tkestack.io/v1/clustercredentials/{name}

Common Parameters

- `name` (*in path*): `string` required
  name of the ClusterCredential
- `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 ClusterCredential

Response

- `200` ClusterCredential: OK

`put`

replace the specified ClusterCredential

Parameters

- `dryRun` (*in query*): `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

- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗.
- `fieldValidation` (*in query*): `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.

Request Body

ClusterCredential required

Response

- `200` ClusterCredential: OK
- `201` ClusterCredential: Created

`delete`

delete a ClusterCredential

Parameters

- `dryRun` (*in query*): `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

- `gracePeriodSeconds` (*in query*): `integer`
  The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.
- `orphanDependents` (*in query*): `boolean`
  Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
- `propagationPolicy` (*in query*): `string`
  Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Request Body

DeleteOptions

Response

- `200` Status: OK
- `202` Status: Accepted

`patch`

partially update the specified ClusterCredential

Parameters

- `dryRun` (*in query*): `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

- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).
- `fieldValidation` (*in query*): `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.
- `force` (*in query*): `boolean`
  Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Request Body

Patch

Response

- `200` ClusterCredential: OK
- `201` ClusterCredential: Created

ClusterCredentialList

ClusterCredentialList is the whole list of all ClusterCredential which owned by a tenant.

- `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`: `[]ClusterCredential`
  List of clusters
- `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}.

ClusterCredential

ClusterCredential records the credential information needed to access the cluster.

- `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 ↗
- `bootstrapToken`: `string`
  For kubeadm init or join
- `caCert`: `string`
  For connect the cluster
- `caKey`: `string`
  CA key
- `certificateKey`: `string`

For kubeadm init or join

- `clientCert` : `string`
  For kube-apiserver X509 auth
- `clientKey` : `string`
  For kube-apiserver X509 auth
- `clusterName` : `string`
  cluster name
- `etcdAPIClientCert` : `string`
  ETCD api client certificate
- `etcdAPIClientKey` : `string`
  ETCD api client key
- `etcdCACert` : `string`
  For TKE in global reuse
- `etcdCAKey` : `string`
  ETCD CA key
- `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.
- `registryPassword` : `string`
  For external registry
- `registryUsername` : `string`
  For external registry
- `tenantID` : `string`
  tenant id
- `token` : `string`
  For kube-apiserver token auth

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.

Status

Status is a return value for calls that don't return other objects.

- `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 ↗
- `code` : `integer`
  Suggested HTTP return code for this status, 0 if not set.
- `details` : StatusDetails
  StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.
- `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 ↗
- `message` : `string`
  A human-readable description of the status of this operation.
- `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}.
- `reason` : `string`
  A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available. A Reason clarifies an HTTP status code but does not override it.
- `status` : `string`
  Status of the operation. One of: "Success" or "Failure". More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

StatusDetails

StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.

- `causes` : `[]StatusCause`
  The Causes array includes more details associated with the StatusReason failure. Not all StatusReasons may provide detailed causes.
- `group` : `string`
  The group attribute of the resource associated with the status StatusReason.
- `kind` : `string`
  The kind attribute of the resource associated with the status StatusReason. On some operations may differ from the requested resource Kind. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `name` : `string`
  The name attribute of the resource associated with the status StatusReason (when there is a single name which can be described).
- `retryAfterSeconds` : `integer`
  If specified, the time in seconds before the operation should be retried. Some errors may indicate the client must take an alternate action - for those errors this field may indicate how long to wait before taking the alternate action.
- `uid` : `string`
  UID of the resource. (when there is a single resource which can be described). More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids ↗

StatusCause

StatusCause provides more information about an api.Status failure, including cases when multiple errors are encountered.

- `field` : `string`
  The field of the resource that has caused this error, as named by its JSON serialization. May include dot and postfix notation for nested attributes. Arrays are zero-indexed. Fields

may appear more than once in an array of causes due to fields having multiple errors. Optional.
Examples: "name" - the field "name" on the current resource "items[0].name" - the field "name" on the first array entry in "items"

- `message` : `string`
  A human-readable description of the cause of the error. This field may be presented as-is to a reader.
- `reason` : `string`
  A machine-readable description of the cause of the error. If this value is empty there is no information available.

Patch

Patch is provided to give a concrete name and type to the Kubernetes PATCH request body.

Machine [platform.tkestack.io/v1]

/apis/platform.tkestack.io/v1/machines

Common Parameters

- `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`

list objects of kind Machine

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.
- `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.

Response

- `200` MachineList: OK

`post`

create a Machine

Parameters

- `dryRun` (*in query*): `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
- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗.
- `fieldValidation` (*in query*): `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.

Request Body

Machine

Response

- `200` Machine: OK
- `201` Machine: Created
- `202` Machine: Accepted

`delete`

delete collection of Machine

Parameters

- `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.
- `dryRun` (*in query*): `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
- `fieldSelector` (*in query*): `string`
  A selector to restrict the list of returned objects by their fields. Defaults to everything.
- `gracePeriodSeconds` (*in query*): `integer`
  The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace

period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

- `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.
- `orphanDependents` (*in query*): `boolean`
  Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
- `propagationPolicy` (*in query*): `string`
  Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.
- `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.

Request Body

DeleteOptions

Response

- `200` Status: OK

/apis/platform.tkestack.io/v1/machines/{name}

Common Parameters

- `name` (*in path*): `string` required
  name of the Machine
- `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 Machine

Response

- `200` Machine: OK

`put`

replace the specified Machine

Parameters

- `dryRun` (*in query*): `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
- `fieldManager` (*in query*): `string`

fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗.

- `fieldValidation` (*in query*): `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.

Request Body

Machine

Response

- `200` Machine: OK
- `201` Machine: Created

`delete`

delete a Machine

Parameters

- `dryRun` (*in query*): `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
- `gracePeriodSeconds` (*in query*): `integer`

The duration in seconds before the object should be deleted. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period for the specified type will be used. Defaults to a per object value if not specified. zero means delete immediately.

- `orphanDependents` (*in query*): `boolean`
  Deprecated: please use the PropagationPolicy, this field will be deprecated in 1.7. Should the dependent objects be orphaned. If true/false, the "orphan" finalizer will be added to/removed from the object's finalizers list. Either this field or PropagationPolicy may be set, but not both.
- `propagationPolicy` (*in query*): `string`
  Whether and how garbage collection will be performed. Either this field or OrphanDependents may be set, but not both. The default policy is decided by the existing finalizer set in the metadata.finalizers and the resource-specific default policy. Acceptable values are: 'Orphan' - orphan the dependents; 'Background' - allow the garbage collector to delete the dependents in the background; 'Foreground' - a cascading policy that deletes all dependents in the foreground.

Request Body

DeleteOptions

Response

- `200` Status: OK
- `202` Status: Accepted

`patch`

partially update the specified Machine

Parameters

- `dryRun` (*in query*): `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

- `fieldManager` (*in query*): `string`
  fieldManager is a name associated with the actor or entity that is making these changes. The value must be less than or 128 characters long, and only contain printable characters, as defined by https://golang.org/pkg/unicode/#IsPrint ↗. This field is required for apply requests (application/apply-patch) but optional for non-apply patch types (JsonPatch, MergePatch, StrategicMergePatch).
- `fieldValidation` (*in query*): `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.
- `force` (*in query*): `boolean`
  Force is going to "force" Apply requests. It means user will re-acquire conflicting fields owned by other people. Force flag must be unset for non-apply patch requests.

Request Body

Patch

Response

- `200` Machine: OK
- `201` Machine: Created

MachineList

MachineList is the whole list of all machine in an cluster.

- `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`: `[]Machine`
  List of clusters
- `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}.

Machine

Machine instance in Kubernetes cluster

- `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`: MachineSpec

MachineSpec is a description of machine.

- `status` : MachineStatus
  MachineStatus represents information about the status of an machine.

MachineSpec

MachineSpec is a description of machine.

- `annotations` : `map[string]string`
  Annotations
- `clusterName` : `string`
  cluster name
- `displayName` : `string`
  DisplayName
- `finalizers` : `[]string`
  Finalizers is an opaque list of values that must be empty to permanently remove object from storage.
- `ip` : `string`
  ip
- `ipv6` : `string`
  IPv6
- `labels` : `map[string]string`
  labels
- `networkDevice` : `string`
  NetworkDevice
- `passPhrase` : `string`
  pass phrase
- `password` : `string`
  password
- `port` : `integer`
  port
- `privateKey` : `string`

private key

- `proxy` : MachineProxy
  MachineProxy is the proxy to connect Machine's ssh.
- `publicIP` : `string`
  PublicIP
- `role` : `string`
  Role master/node
- `taints` : `[]Taint`
  If specified, the node's taints.
- `tenantID` : `string`
  tenant id
- `type` : `string`
  type
- `username` : `string`
  user name

MachineProxy

MachineProxy is the proxy to connect Machine's ssh.

- `address` : `string`
- `password` : `string`
- `type` : `string`
- `username` : `string`

Taint

The node this Taint is attached to has the "effect" on any pod that does not tolerate the Taint.

- `effect` : `string`
  Required. The effect of the taint on pods that do not tolerate the taint. Valid effects are NoSchedule, PreferNoSchedule and NoExecute.

Possible enum values:

- `"NoExecute"` Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController.
- `"NoSchedule"` Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler.
- `"PreferNoSchedule"` Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.

- `key` : `string`
  Required. The taint key to be applied to a node.
- `timeAdded` : `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.
- `value` : `string`
  The taint value corresponding to the taint key.

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.

MachineStatus

MachineStatus represents information about the status of an machine.

- `addresses` : `[]MachineAddress`
  List of addresses reachable to the machine.
- `conditions` : `[]MachineCondition`
  machine condition
- `locked` : `boolean`

lock status

- `machineInfo` : MachineSystemInfo
  MachineSystemInfo is a set of ids/uuids to uniquely identify the node.
- `message` : `string`
  A human readable message indicating details about why the machine is in this condition.
- `phase` : `string`
  machine phase
- `reason` : `string`
  A brief CamelCase message indicating details about why the machine is in this state.

MachineAddress

MachineAddress contains information for the machine's address.

- `address` : `string`
  The machine address.
- `type` : `string`
  Machine address type, one of Public, ExternalIP or InternalIP.

MachineCondition

MachineCondition contains details for the current condition of this Machine.

- `lastProbeTime` : `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.
- `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`
  Human-readable message indicating details about last transition.

- `reason` : `string`
  Unique, one-word, CamelCase reason for the condition's last transition.
- `status` : `string`
  Status is the status of the condition. Can be True, False, Unknown.
- `type` : `string`
  Type is the type of the condition.

MachineSystemInfo

MachineSystemInfo is a set of ids/uuids to uniquely identify the node.

- `architecture` : `string`
  The Architecture reported by the node
- `bootID` : `string`
  Boot ID reported by the node.
- `containerRuntimeVersion` : `string`
  ContainerRuntime Version reported by the node.
- `kernelVersion` : `string`
  Kernel Version reported by the node.
- `kubeProxyVersion` : `string`
  KubeProxy Version reported by the node.
- `kubeletVersion` : `string`
  Kubelet Version reported by the node.
- `machineID` : `string`
  MachineID reported by the node. For unique machine identification in the cluster this field is preferred. Learn more from man(5) machine-id: http://man7.org/linux/man-pages/man5/machine-id.5.html ↗
- `operatingSystem` : `string`
  The Operating System reported by the node
- `osImage` : `string`
  OS Image reported by the node.
- `systemUUID` : `string`

SystemUUID reported by the node. For unique machine identification MachineID is preferred. This field is specific to Red Hat hosts https://access.redhat.com/documentation/en-US/Red_Hat_Subscription_Management/1/html/RHSM/getting-system-uuid.html ↗

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.

Status

Status is a return value for calls that don't return other objects.

- `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 ↗
- `code` : `integer`
  Suggested HTTP return code for this status, 0 if not set.
- `details` : StatusDetails
  StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.
- `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 ↗
- `message` : `string`
  A human-readable description of the status of this operation.
- `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}.
- `reason` : `string`
  A machine-readable description of why this operation is in the "Failure" status. If this value is empty there is no information available. A Reason clarifies an HTTP status code but does not override it.
- `status` : `string`
  Status of the operation. One of: "Success" or "Failure". More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status ↗

StatusDetails

StatusDetails is a set of additional properties that MAY be set by the server to provide additional information about a response. The Reason field of a Status object defines what attributes will be set. Clients must ignore fields that do not match the defined type of each attribute, and should assume that any attribute may be empty, invalid, or under defined.

- `causes` : `[]StatusCause`
  The Causes array includes more details associated with the StatusReason failure. Not all StatusReasons may provide detailed causes.
- `group` : `string`
  The group attribute of the resource associated with the status StatusReason.
- `kind` : `string`
  The kind attribute of the resource associated with the status StatusReason. On some operations may differ from the requested resource Kind. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
- `name` : `string`
  The name attribute of the resource associated with the status StatusReason (when there is a single name which can be described).
- `retryAfterSeconds` : `integer`
  If specified, the time in seconds before the operation should be retried. Some errors may indicate the client must take an alternate action - for those errors this field may indicate how long to wait before taking the alternate action.
- `uid` : `string`
  UID of the resource. (when there is a single resource which can be described). More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names#uids ↗

StatusCause

StatusCause provides more information about an api.Status failure, including cases when multiple errors are encountered.

- `field` : `string`

The field of the resource that has caused this error, as named by its JSON serialization. May include dot and postfix notation for nested attributes. Arrays are zero-indexed. Fields may appear more than once in an array of causes due to fields having multiple errors. Optional.
Examples: "name" - the field "name" on the current resource "items[0].name" - the field "name" on the first array entry in "items"

- `message` : `string`
  A human-readable description of the cause of the error. This field may be presented as-is to a reader.
- `reason` : `string`
  A machine-readable description of the cause of the error. If this value is empty there is no information available.

Patch

Patch is provided to give a concrete name and type to the Kubernetes PATCH request body.