



Service

Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

```
apiVersion: v1

import "k8s.io/api/core/v1"
```

Service

Service is a named abstraction of software service (for example, mysql) consisting of local port (for example 3306) that the proxy listens on, and the selector that determines which pods will answer requests sent through the proxy.

- **apiVersion:** v1
- **kind:** Service
- **metadata** ([ObjectMeta](#))

Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata>
- **spec** ([ServiceSpec](#))

Spec defines the behavior of a service. <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>
- **status** ([ServiceStatus](#))

Most recently observed status of the service. Populated by the system. Read-only. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

ServiceSpec

ServiceSpec describes the attributes that a user creates on a service.

- **selector** (map[string]string)

Route service traffic to pods with label keys and values matching this selector. If empty or not present, the service is assumed to have an external process managing its endpoints, which Kubernetes will not modify. Only applies to types ClusterIP, NodePort, and LoadBalancer. Ignored if type is ExternalName. More info: <https://kubernetes.io/docs/concepts/services-networking/service/>
- **ports** ([]ServicePort)

Patch strategy: merge on key port

Map: unique values on keys port, protocol will be kept during a merge

The list of ports that are exposed by this service. More info:

<https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

ServicePort contains information on service's port.

- **ports.port** (int32), required

The port that will be exposed by this service.

- **ports.targetPort** (IntOrString)

Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME. If this is a string, it will be looked up as a named port in the target Pod's container ports. If this is not specified, the value of the 'port' field is used (an identity map). This field is ignored for services with clusterIP=None, and should be omitted or set equal to the 'port' field.

More info: <https://kubernetes.io/docs/concepts/services-networking/service/#defining-a-service>

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.

- **ports.protocol** (string)

The IP protocol for this port. Supports "TCP", "UDP", and "SCTP". Default is TCP.

- **ports.name** (string)

The name of this port within the service. This must be a DNS_LABEL. All ports within a ServiceSpec must have unique names. When considering the endpoints for a Service, this must match the 'name' field in the EndpointPort. Optional if only one ServicePort is defined on this service.

- **ports.nodePort** (int32)

The port on each node on which this service is exposed when type is NodePort or LoadBalancer. Usually assigned by the system. If a value is specified, in-range, and not in use it will be used, otherwise the operation will fail. If not specified, a port will be allocated if this Service requires one. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type from NodePort to ClusterIP). More info: <https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport>

- **ports.appProtocol** (string)

The application protocol for this port. This field follows standard Kubernetes label syntax. Un-prefixed names are reserved for IANA standard service names (as per RFC-6335 and <http://www.iana.org/assignments/service-names>). Non-standard protocols should use prefixed names such as mycompany.com/my-custom-protocol.

- **type** (string)

type determines how the Service is exposed. Defaults to ClusterIP. Valid options are ExternalName, ClusterIP, NodePort, and LoadBalancer. "ClusterIP" allocates a cluster-internal IP address for load-balancing to endpoints. Endpoints are determined by the selector or if that is not specified, by manual construction of an Endpoints object or EndpointSlice objects. If clusterIP is "None", no virtual IP is allocated and the endpoints are published as a set of endpoints rather than a virtual IP. "NodePort" builds on ClusterIP and allocates a port on every node which routes to the same endpoints as the clusterIP. "LoadBalancer" builds on NodePort and creates an external load-balancer (if supported in the current cloud) which routes to the same endpoints as the clusterIP. "ExternalName" aliases this service to the specified externalName. Several other fields do not apply to ExternalName services. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types>

- **ipFamilies** ([]string)

Atomic: will be replaced during a merge

IPFamilies is a list of IP families (e.g. IPv4, IPv6) assigned to this service, and is gated by the "IPv6DualStack" feature gate. This field is usually assigned automatically based on cluster configuration and the ipFamilyPolicy field. If this field is specified manually, the requested family is available in the cluster, and ipFamilyPolicy allows it, it will be used; otherwise creation of the service will fail. This field is conditionally mutable: it allows for adding or removing a secondary IP family, but it does not allow changing the primary IP family of the Service. Valid values are "IPv4" and "IPv6". This field only applies to Services of types ClusterIP, NodePort, and LoadBalancer, and does apply to "headless" services. This field will be wiped when updating a Service to type ExternalName.

This field may hold a maximum of two entries (dual-stack families, in either order). These families must correspond to the values of the clusterIPs field, if specified. Both clusterIPs and ipFamilies are governed by the ipFamilyPolicy field.

- **ipFamilyPolicy** (string)

IPFamilyPolicy represents the dual-stack-ness requested or required by this Service, and is gated by the "IPv6DualStack" feature gate. If there is no value provided, then this field will be set to SingleStack. Services can be "SingleStack" (a single IP family), "PreferDualStack" (two IP families on dual-stack configured clusters or a single IP family on single-stack clusters), or "RequireDualStack" (two IP families on dual-stack configured clusters, otherwise fail). The ipFamilies and clusterIPs fields depend on the value of this field. This field will be wiped when updating a service to type ExternalName.

- **clusterIP** (string)

clusterIP is the IP address of the service and is usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be blank) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. More info:

<https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

- **clusterIPs** ([]string)

Atomic: will be replaced during a merge

ClusterIPs is a list of IP addresses assigned to this service, and are usually assigned randomly. If an address is specified manually, is in-range (as per system configuration), and is not in use, it will be allocated to the service; otherwise creation of the service will fail. This field may not be changed through updates unless the type field is also being changed to ExternalName (which requires this field to be empty) or the type field is being changed from ExternalName (in which case this field may optionally be specified, as describe above). Valid values are "None", empty string (""), or a valid IP address. Setting this to "None" makes a "headless service" (no virtual IP), which is useful when direct endpoint connections are preferred and proxying is not required. Only applies to types ClusterIP, NodePort, and LoadBalancer. If this field is specified when creating a Service of type ExternalName, creation will fail. This field will be wiped when updating a Service to type ExternalName. If this field is not specified, it will be initialized from the clusterIP field. If this field is specified, clients must ensure that clusterIPs[0] and clusterIP have the same value.

Unless the "IPv6DualStack" feature gate is enabled, this field is limited to one value, which must be the same as the clusterIP field. If the feature gate is enabled, this field may hold a maximum of two entries (dual-stack IPs, in either order). These IPs must correspond to the

values of the `ipFamilies` field. Both `clusterIPs` and `ipFamilies` are governed by the `ipFamilyPolicy` field. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

- **externalIPs** ([]string)

`externalIPs` is a list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.

- **sessionAffinity** (string)

Supports "ClientIP" and "None". Used to maintain session affinity. Enable client IP based session affinity. Must be ClientIP or None. Defaults to None. More info: <https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies>

- **loadBalancerIP** (string)

Only applies to Service Type: LoadBalancer. LoadBalancer will get created with the IP specified in this field. This feature depends on whether the underlying cloud-provider supports specifying the `loadBalancerIP` when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature.

- **loadBalancerSourceRanges** ([]string)

If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature." More info: <https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/>

- **loadBalancerClass** (string)

`loadBalancerClass` is the class of the load balancer implementation this Service belongs to. If specified, the value of this field must be a label-style identifier, with an optional prefix, e.g. "internal-vip" or "example.com/internal-vip". Unprefixed names are reserved for end-users. This field can only be set when the Service type is 'LoadBalancer'. If not set, the default load balancer implementation is used, today this is typically done through the cloud provider integration, but should apply for any default implementation. If set, it is assumed that a load balancer implementation is watching for Services with a matching class. Any default load balancer implementation (e.g. cloud providers) should ignore Services that set this field. This field can only be set when creating or updating a Service to type 'LoadBalancer'. Once set, it can not be changed. This field will be wiped when a service is updated to a non 'LoadBalancer' type.

- **externalName** (string)

`externalName` is the external reference that discovery mechanisms will return as an alias for this service (e.g. a DNS CNAME record). No proxying will be involved. Must be a lowercase RFC-1123 hostname (<https://tools.ietf.org/html/rfc1123>) and requires `type` to be "ExternalName".

- **externalTrafficPolicy** (string)

`externalTrafficPolicy` denotes if this Service desires to route external traffic to node-local or cluster-wide endpoints. "Local" preserves the client source IP and avoids a second hop for LoadBalancer and Nodeport type services, but risks potentially imbalanced traffic spreading. "Cluster" obscures the client source IP and may cause a second hop to another node, but should have good overall load-spreading.

- **internalTrafficPolicy** (string)

`InternalTrafficPolicy` specifies if the cluster internal traffic should be routed to all endpoints or node-local endpoints only. "Cluster" routes internal traffic to a Service to all endpoints. "Local" routes traffic to node-local endpoints only, traffic is dropped if no node-local endpoints are ready. The default value is "Cluster".

- **healthCheckNodePort** (int32)

healthCheckNodePort specifies the healthcheck nodePort for the service. This only applies when type is set to LoadBalancer and externalTrafficPolicy is set to Local. If a value is specified, is in-range, and is not in use, it will be used. If not specified, a value will be automatically allocated. External systems (e.g. load-balancers) can use this port to determine if a given node holds endpoints for this service or not. If this field is specified when creating a Service which does not need it, creation will fail. This field will be wiped when updating a Service to no longer need it (e.g. changing type).

- **publishNotReadyAddresses** (boolean)

publishNotReadyAddresses indicates that any agent which deals with endpoints for this Service should disregard any indications of ready/not-ready. The primary use case for setting this field is for a StatefulSet's Headless Service to propagate SRV DNS records for its Pods for the purpose of peer discovery. The Kubernetes controllers that generate Endpoints and EndpointSlice resources for Services interpret this to mean that all endpoints are considered "ready" even if the Pods themselves are not. Agents which consume only Kubernetes generated endpoints through the Endpoints or EndpointSlice resources can safely assume this behavior.

- **sessionAffinityConfig** (SessionAffinityConfig)

sessionAffinityConfig contains the configurations of session affinity.

SessionAffinityConfig represents the configurations of session affinity.

- **sessionAffinityConfig.clientIP** (ClientIPConfig)

clientIP contains the configurations of Client IP based session affinity.

ClientIPConfig represents the configurations of Client IP based session affinity.

- **sessionAffinityConfig.clientIP.timeoutSeconds** (int32)

timeoutSeconds specifies the seconds of ClientIP type session sticky time. The value must be >0 && <=86400(for 1 day) if ServiceAffinity == "ClientIP". Default value is 10800(for 3 hours).

- **allocateLoadBalancerNodePorts** (boolean)

allocateLoadBalancerNodePorts defines if NodePorts will be automatically allocated for services with type LoadBalancer. Default is "true". It may be set to "false" if the cluster load-balancer does not rely on NodePorts. If the caller requests specific NodePorts (by specifying a value), those requests will be respected, regardless of this field. This field may only be set for services with type LoadBalancer and will be cleared if the type is changed to any other type. This field is beta-level and is only honored by servers that enable the ServiceLBNodePortControl feature.

ServiceStatus

ServiceStatus represents the current status of a service.

- **conditions** ([]Condition)

Patch strategy: merge on key `type`

Map: unique values on key type will be kept during a merge

Current service state

Condition contains details for one aspect of the current state of this API Resource.

- **conditions.lastTransitionTime** (Time), required

lastTransitionTime is the last time the condition transitioned from one status to another. This should be when the underlying condition changed. If that is not known, then using the time when the API field changed is acceptable.

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.

- **conditions.message** (string), required

message is a human readable message indicating details about the transition. This may be an empty string.

- **conditions.reason** (string), required

reason contains a programmatic identifier indicating the reason for the condition's last transition. Producers of specific condition types may define expected values and meanings for this field, and whether the values are considered a guaranteed API. The value should be a CamelCase string. This field may not be empty.

- **conditions.status** (string), required

status of the condition, one of True, False, Unknown.

- **conditions.type** (string), required

type of condition in CamelCase or in foo.example.com/CamelCase.

- **conditions.observedGeneration** (int64)

observedGeneration represents the .metadata.generation that the condition was set based upon. For instance, if .metadata.generation is currently 12, but the .status.conditions[x].observedGeneration is 9, the condition is out of date with respect to the current state of the instance.

- **loadBalancer** (LoadBalancerStatus)

LoadBalancer contains the current status of the load-balancer, if one is present.

LoadBalancerStatus represents the status of a load-balancer.

- **loadBalancer.ingress** ([]LoadBalancerIngress)

Ingress is a list containing ingress points for the load-balancer. Traffic intended for the service should be sent to these ingress points.

LoadBalancerIngress represents the status of a load-balancer ingress point: traffic intended for the service should be sent to an ingress point.

- **loadBalancer.ingress.hostname** (string)

Hostname is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)

- **loadBalancer.ingress.ip** (string)

IP is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)

- **loadBalancer.ingress.ports** ([]PortStatus)

Atomic: will be replaced during a merge

Ports is a list of records of service ports. If used, every port defined in the service should have an entry in it

- **loadBalancer.ingress.ports.port** (int32), required

Port is the port number of the service port of which status is recorded here

- **loadBalancer.ingress.ports.protocol** (string), required

Protocol is the protocol of the service port of which status is recorded here
The supported values are: "TCP", "UDP", "SCTP"

- **loadBalancer.ingress.ports.error** (string)

Error is to record the problem with the service port The format of the error shall comply with the following rules: - built-in error values shall be specified in this file and those shall use CamelCase names

- cloud provider specific error values must have names that comply with the format foo.example.com/CamelCase.

ServiceList

ServiceList holds a list of services.

- **apiVersion:** v1
- **kind:** ServiceList
- **metadata** ([ListMeta](#))

Standard list metadata. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>

- **items** ([\[\]Service](#)), required

List of services

Operations

get read the specified Service

HTTP Request

GET /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Service](#)): OK

401: Unauthorized

get read status of the specified Service

HTTP Request

GET /api/v1/namespaces/{namespace}/services/{name}/status

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Service](#)): OK

401: Unauthorized

list list or watch objects of kind Service

HTTP Request

GET /api/v1/namespaces/{namespace}/services

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean

[watch](#)

Response

200 ([ServiceList](#)): OK

401: Unauthorized

list list or watch objects of kind Service

HTTP Request

GET /api/v1/services

Parameters

- **allowWatchBookmarks** (*in query*): boolean
[allowWatchBookmarks](#)
- **continue** (*in query*): string
[continue](#)
- **fieldSelector** (*in query*): string
[fieldSelector](#)
- **labelSelector** (*in query*): string
[labelSelector](#)
- **limit** (*in query*): integer
[limit](#)
- **pretty** (*in query*): string
[pretty](#)
- **resourceVersion** (*in query*): string
[resourceVersion](#)
- **resourceVersionMatch** (*in query*): string
[resourceVersionMatch](#)
- **timeoutSeconds** (*in query*): integer
[timeoutSeconds](#)
- **watch** (*in query*): boolean
[watch](#)

Response

200 ([ServiceList](#)): OK

401: Unauthorized

create create a Service

HTTP Request

POST /api/v1/namespaces/{namespace}/services

Parameters

- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Service](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **pretty** (*in query*): string
[pretty](#)

Response

- 200 ([Service](#)): OK
- 201 ([Service](#)): Created
- 202 ([Service](#)): Accepted
- 401: Unauthorized

update replace the specified Service

HTTP Request

PUT /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Service](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **pretty** (*in query*): string
[pretty](#)

Response

- 200 ([Service](#)): OK
- 201 ([Service](#)): Created
- 401: Unauthorized

update replace status of the specified Service

HTTP Request

PUT /api/v1/namespaces/{namespace}/services/{name}/status

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Service](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **pretty** (*in query*): string
[pretty](#)

Response

200 ([Service](#)): OK

201 ([Service](#)): Created

401: Unauthorized

patch partially update the specified Service

HTTP Request

PATCH /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string

[pretty](#)

Response

- 200 ([Service](#)): OK
- 201 ([Service](#)): Created
- 401: Unauthorized

patch partially update status of the specified Service

HTTP Request

PATCH /api/v1/namespaces/{namespace}/services/{name}/status

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)
- **body**: [Patch](#), required
- **dryRun** (*in query*): string
[dryRun](#)
- **fieldManager** (*in query*): string
[fieldManager](#)
- **force** (*in query*): boolean
[force](#)
- **pretty** (*in query*): string
[pretty](#)

Response

- 200 ([Service](#)): OK
- 201 ([Service](#)): Created
- 401: Unauthorized

delete delete a Service

HTTP Request

DELETE /api/v1/namespaces/{namespace}/services/{name}

Parameters

- **name** (*in path*): string, required
name of the Service
- **namespace** (*in path*): string, required
[namespace](#)

- **body:** [DeleteOptions](#)
- **dryRun** (*in query*): string
[dryRun](#)
- **gracePeriodSeconds** (*in query*): integer
[gracePeriodSeconds](#)
- **pretty** (*in query*): string
[pretty](#)
- **propagationPolicy** (*in query*): string
[propagationPolicy](#).

Response

200 ([Status](#)): OK

202 ([Status](#)): Accepted

401: Unauthorized

Feedback

Was this page helpful?

Yes

No

Last modified June 26, 2021 at 11:28 AM PST : [v1.22 beta.0 \(02d18a3ce\)](#).