--hostname-override string

If non-empty, will be used as the name of the Node that kube-proxy is running on. If unset, the node name is assumed to be the same as the node's hostname.

--init-only

If true, perform any initialization steps that must be done with full root privileges, and then exit. After doing this, you can run kube-proxy again with only the CAP_NET_ADMIN capability.

--iptables-localhost-nodeports Default: true

If false, kube-proxy will disable the legacy behavior of allowing NodePort services to be accessed via localhost. (Applies only to iptables mode and IPv4; localhost NodePorts are never allowed with other proxy modes or with IPv6.)

--iptables-masquerade-bit int32 Default: 14

If using the iptables or ipvs proxy mode, the bit of the fwmark space to mark packets requiring SNAT with. Must be within the range [0, 31].

--iptables-min-sync-period duration Default: 1s

The minimum period between iptables rule resyncs (e.g. '5s', '1m', '2h22m'). A value of 0 means every Service or EndpointSlice change will result in an immediate iptables resync.

--iptables-sync-period duration Default: 30s

An interval (e.g. '5s', '1m', '2h22m') indicating how frequently various re-synchronizing and cleanup operations are performed. Must be greater than 0.

--ipvs-exclude-cidrs strings

A comma-separated list of CIDRs which the ipvs proxier should not touch when cleaning up IPVS rules.

--ipvs-min-sync-period duration Default: 1s

The minimum period between IPVS rule resyncs (e.g. '5s', '1m', '2h22m'). A value of 0 means every Service or EndpointSlice change will result in an immediate IPVS resync.

--ipvs-scheduler string

The ipvs scheduler type when proxy mode is ipvs

--ipvs-strict-arp

Enable strict ARP by setting arp_ignore to 1 and arp_announce to 2

--ipvs-sync-period duration Default: 30s

An interval (e.g. '5s', '1m', '2h22m') indicating how frequently various re-synchronizing and cleanup operations are performed. Must be greater than 0.

--ipvs-tcp-timeout duration

The timeout for idle IPVS TCP connections, 0 to leave as-is. (e.g. '5s', '1m', '2h22m').

--ipvs-tcpfin-timeout duration

The timeout for IPVS TCP connections after receiving a FIN packet, 0 to leave as-is. (e.g. '5s', '1m', '2h22m').

--ipvs-udp-timeout duration

The timeout for IPVS UDP packets, 0 to leave as-is. (e.g. '5s', '1m', '2h22m').

--kube-api-burst int32 Default: 10

Burst to use while talking with kubernetes apiserver

--kube-api-content-type string Default: "application/vnd.kubernetes.protobuf"

Content type of requests sent to apiserver.

--kube-api-qps float Default: 5

QPS to use while talking with kubernetes apiserver

--kubeconfig string

Path to kubeconfig file with authorization information (the master location can be overridden by the master flag).

--log-flush-frequency duration Default: 5s

Maximum number of seconds between log flushes

--log-text-info-buffer-size quantity

[Alpha] In text format with split output streams, the info messages can be buffered for a while to increase performance. The default value of zero bytes disables buffering. The size can be specified as number of bytes (512), multiples of 1000 (1K), multiples of 1024 (2Ki), or powers of those (3M, 4G, 5Mi, 6Gi). Enable the LoggingAlphaOptions feature gate to use this.

--log-text-split-stream

[Alpha] In text format, write error messages to stderr and info messages to stdout. The default is to write a single stream to stdout. Enable the LoggingAlphaOptions feature gate to use this.

--log_backtrace_at <a string in the form 'file:N'> Default: :0

when logging hits line file:N, emit a stack trace

--log_dir string

If non-empty, write log files in this directory (no effect when -logtostderr=true)

--log_file string

If non-empty, use this log file (no effect when -logtostderr=true)

--log_file_max_size uint Default: 1800

Defines the maximum size a log file can grow to (no effect when -logtostderr=true). Unit is megabytes. If the value is 0, the maximum file size is unlimited.

--logging-format string Default: "text"

Sets the log format. Permitted formats: "text".

--logtostderr Default: true

log to standard error instead of files

--masquerade-all

SNAT all traffic sent via Service cluster IPs. This may be required with some CNI plugins. Only supported on Linux.

--master string

The address of the Kubernetes API server (overrides any value in kubeconfig)

--metrics-bind-address ipport Default: 127.0.0.1:10249

The IP address and port for the metrics server to serve on, defaulting to "127.0.0.1:10249". (Set to "0.0.0.0:10249" / "[::]:10249" to bind on all interfaces.) Set empty to disable. This parameter is ignored if a config file is specified by --config.

--nodeport-addresses strings

A list of CIDR ranges that contain valid node IPs, or alternatively, the single string 'primary'. If set to a list of CIDRs, connections to NodePort services will only be accepted on node IPs in one of the indicated ranges. If set to 'primary', NodePort services will only be accepted on the node's primary IP(s) according to the Node object. If unset, NodePort connections will be accepted on all local IPs. This parameter is ignored if a config file is specified by --config.

--one_output

If true, only write logs to their native severity level (vs also writing to each lower severity level; no effect when -logtostderr=true)

--oom-score-adj int32 Default: -999

The oom-score-adj value for kube-proxy process. Values must be within the range [-1000, 1000]. This parameter is ignored if a config file is specified by --config.

--pod-bridge-interface string

A bridge interface name. When --detect-local-mode is set to BridgeInterface, kube-proxy will consider traffic to be local if it originates from this bridge.

--pod-interface-name-prefix string

An interface name prefix. When --detect-local-mode is set to InterfaceNamePrefix, kube-proxy will consider traffic to be local if it originates from any interface whose name begins with this prefix.

--profiling

If true enables profiling via web interface on /debug/pprof handler. This parameter is ignored if a config file is specified by --config.

--proxy-mode ProxyMode

Which proxy mode to use: on Linux this can be 'iptables' (default) or 'ipvs'. On Windows the only supported value is 'kernelspace'.This parameter is ignored if a config file is specified by --config.

--show-hidden-metrics-for-version string

The previous version for which you want to show hidden metrics. Only the previous minor version is meaningful, other values will not be allowed. The format is <major>.<minor>, e.g.: '1.16'. The purpose of this format is make sure you have the opportunity to notice if the next release hides additional metrics, rather than being surprised when they are permanently removed in the release after that. This parameter is ignored if a config file is specified by --config.

--skip_headers

If true, avoid header prefixes in the log messages

--skip_log_headers

If true, avoid headers when opening log files (no effect when -logtostderr=true)

--stderrthreshold int Default: 2

logs at or above this threshold go to stderr when writing to files and stderr (no effect when -logtostderr=true or -alsologtostderr=true)

-v, --v int

number for the log level verbosity

--version version[=true]

--version, --version=raw prints version information and quits; --version=vX.Y.Z... sets the reported version

--vmodule pattern=N,...

comma-separated list of pattern=N settings for file-filtered logging (only works for text log format)

--write-config-to string

If set, write the default configuration values to this file and exit.

12.7 - kube-scheduler

Synopsis

The Kubernetes scheduler is a control plane process which assigns Pods to Nodes. The scheduler determines which Nodes are valid placements for each Pod in the scheduling queue according to constraints and available resources. The scheduler then ranks each valid Node and binds the Pod to a suitable Node. Multiple different schedulers may be used within a cluster; kube-scheduler is the reference implementation. See scheduling for more information about scheduling and the kube-scheduler component.

```
kube-scheduler [flags]
```

Options

--allow-metric-labels stringToString Default: []

The map from metric-label to value allow-list of this label. The key's format is <MetricName>,<LabelName>. The value's format is <allowed_value>,<allowed_value>...e.g. metric1,label1='v1,v2,v3', metric1,label2='v1,v2,v3' metric2,label1='v1,v2,v3'.

--allow-metric-labels-manifest string

The path to the manifest file that contains the allow-list mapping. The format of the file is the same as the flag --allow-metric-labels. Note that the flag --allow-metric-labels will override the manifest file.

--authentication-kubeconfig string

kubeconfig file pointing at the 'core' kubernetes server with enough rights to create tokenreviews.authentication.k8s.io. This is optional. If empty, all token requests are considered to be anonymous and no client CA is looked up in the cluster.

--authentication-skip-lookup

If false, the authentication-kubeconfig will be used to lookup missing authentication configuration from the cluster.

--authentication-token-webhook-cache-ttl duration Default: 10s

The duration to cache responses from the webhook token authenticator.

--authentication-tolerate-lookup-failure Default: true

If true, failures to look up missing authentication configuration from the cluster are not considered fatal. Note that this can result in authentication that treats all requests as anonymous.

--authorization-always-allow-paths strings Default: "/healthz,/readyz,/livez"

A list of HTTP paths to skip during authorization, i.e. these are authorized without contacting the 'core' kubernetes server.

--authorization-kubeconfig string

kubeconfig file pointing at the 'core' kubernetes server with enough rights to create subjectaccessreviews.authorization.k8s.io. This is optional. If empty, all requests not skipped by authorization are forbidden.

--authorization-webhook-cache-authorized-ttl duration Default: 10s

The duration to cache 'authorized' responses from the webhook authorizer.

--authorization-webhook-cache-unauthorized-ttl duration Default: 10s

The duration to cache 'unauthorized' responses from the webhook authorizer.

--bind-address string Default: 0.0.0.0

The IP address on which to listen for the --secure-port port. The associated interface(s) must be reachable by the rest of the cluster, and by CLI/web clients. If blank or an unspecified address (0.0.0.0 or ::), all interfaces and IP address families will be used.

--cert-dir string

The directory where the TLS certs are located. If --tls-cert-file and --tls-private-key-file are provided, this flag will be ignored.

--client-ca-file string

If set, any request presenting a client certificate signed by one of the authorities in the client-ca-file is authenticated with an identity corresponding to the CommonName of the client certificate.

--config string

The path to the configuration file.

--contention-profiling Default: true

DEPRECATED: enable block profiling, if profiling is enabled. This parameter is ignored if a config file is specified in --config.

--disable-http2-serving

If true, HTTP2 serving will be disabled [default=false]

--disabled-metrics strings

This flag provides an escape hatch for misbehaving metrics. You must provide the fully qualified metric name in order to disable it. Disclaimer: disabling metrics is higher in precedence than showing hidden metrics.

--emulated-version strings

The versions different components emulate their capabilities (APIs, features, ...) of. If set, the component will emulate the behavior of this version instead of the underlying binary version.
Version format could only be major.minor, for example: '--emulated-version=wardle=1.2,kube=1.31'. Options are:
kube=1.31..1.31 (default=1.31)If the component is not specified, defaults to "kube"

--feature-gates colonSeparatedMultimapStringString

Comma-separated list of component:key=value pairs that describe feature gates for alpha/experimental features of different components.
If the component is not specified, defaults to "kube". This flag can be repeatedly invoked. For example: --feature-gates 'wardle:featureA=true,wardle:featureB=false' --feature-gates 'kube:featureC=true'Options are:
kube:APIResponseCompression=true|false (BETA - default=true)
kube:APIServerIdentity=true|false (BETA - default=true)
kube:APIServerTracing=true|false (BETA - default=true)
kube:APIServingWithRoutine=true|false (ALPHA - default=false)
kube:AllAlpha=true|false (ALPHA - default=false)
kube:AllBeta=true|false (BETA - default=false)
kube:AnonymousAuthConfigurableEndpoints=true|false (ALPHA - default=false)
kube:AnyVolumeDataSource=true|false (BETA - default=true)
kube:AuthorizeNodeWithSelectors=true|false (ALPHA - default=false)
kube:AuthorizeWithSelectors=true|false (ALPHA - default=false)
kube:CPUManagerPolicyAlphaOptions=true|false (ALPHA - default=false)
kube:CPUManagerPolicyBetaOptions=true|false (BETA - default=true)
kube:CPUManagerPolicyOptions=true|false (BETA - default=true)
kube:CRDValidationRatcheting=true|false (BETA - default=true)
kube:CSIMigrationPortworx=true|false (BETA - default=true)
kube:CSIVolumeHealth=true|false (ALPHA - default=false)
kube:CloudControllerManagerWebhook=true|false (ALPHA - default=false)
kube:ClusterTrustBundle=true|false (ALPHA - default=false)
kube:ClusterTrustBundleProjection=true|false (ALPHA - default=false)
kube:ComponentSLIs=true|false (BETA - default=true)
kube:ConcurrentWatchObjectDecode=true|false (BETA - default=false)
kube:ConsistentListFromCache=true|false (BETA - default=true)
kube:ContainerCheckpoint=true|false (BETA - default=true)
kube:ContextualLogging=true|false (BETA - default=true)
kube:CoordinatedLeaderElection=true|false (ALPHA - default=false)
kube:CronJobsScheduledAnnotation=true|false (BETA - default=true)
kube:CrossNamespaceVolumeDataSource=true|false (ALPHA - default=false)
kube:CustomCPUCFSQuotaPeriod=true|false (ALPHA - default=false)
kube:CustomResourceFieldSelectors=true|false (BETA - default=true)
kube:DRAControlPlaneController=true|false (ALPHA - default=false)
kube:DisableAllocatorDualWrite=true|false (ALPHA - default=false)
kube:DisableNodeKubeProxyVersion=true|false (BETA - default=true)
kube:DynamicResourceAllocation=true|false (ALPHA - default=false)
kube:EventedPLEG=true|false (ALPHA - default=false)
kube:GracefulNodeShutdown=true|false (BETA - default=true)
kube:GracefulNodeShutdownBasedOnPodPriority=true|false (BETA - default=true)
kube:HPAScaleToZero=true|false (ALPHA - default=false)
kube:HonorPVReclaimPolicy=true|false (BETA - default=true)
kube:ImageMaximumGCAge=true|false (BETA - default=true)
kube:ImageVolume=true|false (ALPHA - default=false)
kube:InPlacePodVerticalScaling=true|false (ALPHA - default=false)
kube:InTreePluginPortworxUnregister=true|false (ALPHA - default=false)
kube:InformerResourceVersion=true|false (ALPHA - default=false)
kube:JobBackoffLimitPerIndex=true|false (BETA - default=true)
kube:JobManagedBy=true|false (ALPHA - default=false)
kube:JobPodReplacementPolicy=true|false (BETA - default=true)
kube:JobSuccessPolicy=true|false (BETA - default=true)
kube:KubeletCgroupDriverFromCRI=true|false (BETA - default=true)
kube:KubeletInUserNamespace=true|false (ALPHA - default=false)
kube:KubeletPodResourcesDynamicResources=true|false (ALPHA - default=false)
kube:KubeletPodResourcesGet=true|false (ALPHA - default=false)
kube:KubeletSeparateDiskGC=true|false (BETA - default=true)
kube:KubeletTracing=true|false (BETA - default=true)
kube:LoadBalancerIPMode=true|false (BETA - default=true)
kube:LocalStorageCapacityIsolationFSQuotaMonitoring=true|false (BETA - default=false)
kube:LoggingAlphaOptions=true|false (ALPHA - default=false)
kube:LoggingBetaOptions=true|false (BETA - default=true)
kube:MatchLabelKeysInPodAffinity=true|false (BETA - default=true)
kube:MatchLabelKeysInPodTopologySpread=true|false (BETA - default=true)
kube:MaxUnavailableStatefulSet=true|false (ALPHA - default=false)
kube:MemoryManager=true|false (BETA - default=true)
kube:MemoryQoS=true|false (ALPHA - default=false)
kube:MultiCIDRServiceAllocator=true|false (BETA - default=false)
kube:MutatingAdmissionPolicy=true|false (ALPHA - default=false)
kube:NFTablesProxyMode=true|false (BETA - default=true)

kube:NodeInclusionPolicyInPodTopologySpread=true|false (BETA - default=true)
kube:NodeLogQuery=true|false (BETA - default=false)
kube:NodeSwap=true|false (BETA - default=true)
kube:OpenAPIEnums=true|false (BETA - default=true)
kube:PodAndContainerStatsFromCRI=true|false (ALPHA - default=false)
kube:PodDeletionCost=true|false (BETA - default=true)
kube:PodIndexLabel=true|false (BETA - default=true)
kube:PodLifecycleSleepAction=true|false (BETA - default=true)
kube:PodReadyToStartContainersCondition=true|false (BETA - default=true)
kube:PortForwardWebsockets=true|false (BETA - default=true)
kube:ProcMountType=true|false (BETA - default=false)
kube:QOSReserved=true|false (ALPHA - default=false)
kube:RecoverVolumeExpansionFailure=true|false (ALPHA - default=false)
kube:RecursiveReadOnlyMounts=true|false (BETA - default=true)
kube:RelaxedEnvironmentVariableValidation=true|false (ALPHA - default=false)
kube:ReloadKubeletServerCertificateFile=true|false (BETA - default=true)
kube:ResilientWatchCacheInitialization=true|false (BETA - default=true)
kube:ResourceHealthStatus=true|false (ALPHA - default=false)
kube:RetryGenerateName=true|false (BETA - default=true)
kube:RotateKubeletServerCertificate=true|false (BETA - default=true)
kube:RuntimeClassInImageCriApi=true|false (ALPHA - default=false)
kube:SELinuxMount=true|false (ALPHA - default=false)
kube:SELinuxMountReadWriteOncePod=true|false (BETA - default=true)
kube:SchedulerQueueingHints=true|false (BETA - default=false)
kube:SeparateCacheWatchRPC=true|false (BETA - default=true)
kube:SeparateTaintEvictionController=true|false (BETA - default=true)
kube:ServiceAccountTokenJTI=true|false (BETA - default=true)
kube:ServiceAccountTokenNodeBinding=true|false (BETA - default=true)
kube:ServiceAccountTokenNodeBindingValidation=true|false (BETA - default=true)
kube:ServiceAccountTokenPodNodeInfo=true|false (BETA - default=true)
kube:ServiceTrafficDistribution=true|false (BETA - default=true)
kube:SidecarContainers=true|false (BETA - default=true)
kube:SizeMemoryBackedVolumes=true|false (BETA - default=true)
kube:StatefulSetAutoDeletePVC=true|false (BETA - default=true)
kube:StorageNamespaceIndex=true|false (BETA - default=true)
kube:StorageVersionAPI=true|false (ALPHA - default=false)
kube:StorageVersionHash=true|false (BETA - default=true)
kube:StorageVersionMigrator=true|false (ALPHA - default=false)
kube:StrictCostEnforcementForVAP=true|false (BETA - default=false)
kube:StrictCostEnforcementForWebhooks=true|false (BETA - default=false)
kube:StructuredAuthenticationConfiguration=true|false (BETA - default=true)
kube:StructuredAuthorizationConfiguration=true|false (BETA - default=true)
kube:SupplementalGroupsPolicy=true|false (ALPHA - default=false)
kube:TopologyAwareHints=true|false (BETA - default=true)
kube:TopologyManagerPolicyAlphaOptions=true|false (ALPHA - default=false)
kube:TopologyManagerPolicyBetaOptions=true|false (BETA - default=true)
kube:TopologyManagerPolicyOptions=true|false (BETA - default=true)
kube:TranslateStreamCloseWebsocketRequests=true|false (BETA - default=true)
kube:UnauthenticatedHTTP2DOSMitigation=true|false (BETA - default=true)
kube:UnknownVersionInteroperabilityProxy=true|false (ALPHA - default=false)
kube:UserNamespacesPodSecurityStandards=true|false (ALPHA - default=false)
kube:UserNamespacesSupport=true|false (BETA - default=false)
kube:VolumeAttributesClass=true|false (BETA - default=false)
kube:VolumeCapacityPriority=true|false (ALPHA - default=false)
kube:WatchCacheInitializationPostStartHook=true|false (BETA - default=false)
kube:WatchFromStorageWithoutResourceVersion=true|false (BETA - default=false)
kube:WatchList=true|false (ALPHA - default=false)
kube:WatchListClient=true|false (BETA - default=false)
kube:WinDSR=true|false (ALPHA - default=false)
kube:WinOverlay=true|false (BETA - default=true)
kube:WindowsHostNetwork=true|false (ALPHA - default=true)

-h, --help

help for kube-scheduler

--http2-max-streams-per-connection int

The limit that the server gives to clients for the maximum number of streams in an HTTP/2 connection. Zero means to use golang's default.

--kube-api-burst int32 Default: 100

DEPRECATED: burst to use while talking with kubernetes apiserver. This parameter is ignored if a config file is specified in --config.

--kube-api-content-type string Default: "application/vnd.kubernetes.protobuf"

DEPRECATED: content type of requests sent to apiserver. This parameter is ignored if a config file is specified in --config.

--kube-api-qps float Default: 50

DEPRECATED: QPS to use while talking with kubernetes apiserver. This parameter is ignored if a config file is specified in --config.

--kubeconfig string

DEPRECATED: path to kubeconfig file with authorization and master location information. This parameter is ignored if a config file is specified in --config.

--leader-elect Default: true

Start a leader election client and gain leadership before executing the main loop. Enable this when running replicated components for high availability.

--leader-elect-lease-duration duration Default: 15s

The duration that non-leader candidates will wait after observing a leadership renewal until attempting to acquire leadership of a led but unrenewed leader slot. This is effectively the maximum duration that a leader can be stopped before it is replaced by another candidate. This is only applicable if leader election is enabled.

--leader-elect-renew-deadline duration Default: 10s

The interval between attempts by the acting master to renew a leadership slot before it stops leading. This must be less than the lease duration. This is only applicable if leader election is enabled.

--leader-elect-resource-lock string Default: "leases"

The type of resource object that is used for locking during leader election. Supported options are 'leases', 'endpointsleases' and 'configmapsleases'.

--leader-elect-resource-name string Default: "kube-scheduler"

The name of resource object that is used for locking during leader election.

--leader-elect-resource-namespace string Default: "kube-system"

The namespace of resource object that is used for locking during leader election.

--leader-elect-retry-period duration Default: 2s

The duration the clients should wait between attempting acquisition and renewal of a leadership. This is only applicable if leader election is enabled.

--log-flush-frequency duration Default: 5s

Maximum number of seconds between log flushes

--log-text-info-buffer-size quantity

[Alpha] In text format with split output streams, the info messages can be buffered for a while to increase performance. The default value of zero bytes disables buffering. The size can be specified as number of bytes (512), multiples of 1000 (1K), multiples of 1024 (2Ki), or powers of those (3M, 4G, 5Mi, 6Gi). Enable the LoggingAlphaOptions feature gate to use this.

--log-text-split-stream

[Alpha] In text format, write error messages to stderr and info messages to stdout. The default is to write a single stream to stdout. Enable the LoggingAlphaOptions feature gate to use this.

--logging-format string Default: "text"

Sets the log format. Permitted formats: "text".

--master string

The address of the Kubernetes API server (overrides any value in kubeconfig)

--permit-address-sharing

If true, SO_REUSEADDR will be used when binding the port. This allows binding to wildcard IPs like 0.0.0.0 and specific IPs in parallel, and it avoids waiting for the kernel to release sockets in TIME_WAIT state. [default=false]

--permit-port-sharing

If true, SO_REUSEPORT will be used when binding the port, which allows more than one instance to bind on the same address and port. [default=false]

--pod-max-in-unschedulable-pods-duration duration Default: 5m0s

DEPRECATED: the maximum time a pod can stay in unschedulablePods. If a pod stays in unschedulablePods for longer than this value, the pod will be moved from unschedulablePods to backoffQ or activeQ. This flag is deprecated and will be removed in a future version.

--profiling Default: true

DEPRECATED: enable profiling via web interface host:port/debug/pprof/. This parameter is ignored if a config file is specified in --config.

--requestheader-allowed-names strings

List of client certificate common names to allow to provide usernames in headers specified by --requestheader-username-headers. If empty, any client certificate validated by the authorities in --requestheader-client-ca-file is allowed.

--requestheader-client-ca-file string

Root certificate bundle to use to verify client certificates on incoming requests before trusting usernames in headers specified by --requestheader-username-headers. WARNING: generally do not depend on authorization being already done for incoming requests.

--requestheader-extra-headers-prefix strings Default: "x-remote-extra-"

List of request header prefixes to inspect. X-Remote-Extra- is suggested.

--requestheader-group-headers strings Default: "x-remote-group"

List of request headers to inspect for groups. X-Remote-Group is suggested.

--requestheader-username-headers strings Default: "x-remote-user"

List of request headers to inspect for usernames. X-Remote-User is common.

--secure-port int Default: 10259

The port on which to serve HTTPS with authentication and authorization. If 0, don't serve HTTPS at all.

--show-hidden-metrics-for-version string

The previous version for which you want to show hidden metrics. Only the previous minor version is meaningful, other values will not be allowed. The format is <major>.<minor>, e.g.: '1.16'. The purpose of this format is make sure you have the opportunity to notice if the next release hides additional metrics, rather than being surprised when they are permanently removed in the release after that.

--tls-cert-file string

File containing the default x509 Certificate for HTTPS. (CA cert, if any, concatenated after server cert). If HTTPS serving is enabled, and --tls-cert-file and --tls-private-key-file are not provided, a self-signed certificate and key are generated for the public address and saved to the directory specified by --cert-dir.

--tls-cipher-suites strings

Comma-separated list of cipher suites for the server. If omitted, the default Go cipher suites will be used.
Preferred values: TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305, TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305, TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256.
Insecure values: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_RC4_128_SHA, TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_RC4_128_SHA, TLS_RSA_WITH_3DES_EDE_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_RC4_128_SHA.

--tls-min-version string

Minimum TLS version supported. Possible values: VersionTLS10, VersionTLS11, VersionTLS12, VersionTLS13

--tls-private-key-file string

File containing the default x509 private key matching --tls-cert-file.

--tls-sni-cert-key string

A pair of x509 certificate and private key file paths, optionally suffixed with a list of domain patterns which are fully qualified domain names, possibly with prefixed wildcard segments. The domain patterns also allow IP addresses, but IPs should only be used if the apiserver has visibility to the IP address requested by a client. If no domain patterns are provided, the names of the certificate are extracted. Non-wildcard matches trump over wildcard matches, explicit domain patterns trump over extracted names. For multiple key/certificate pairs, use the --tls-sni-cert-key multiple times. Examples: "example.crt,example.key" or "foo.crt,foo.key:*.foo.com,foo.com".

-v, --v int

number for the log level verbosity

--version version[=true]

--version, --version=raw prints version information and quits; --version=vX.Y.Z... sets the reported version

--vmodule pattern=N,...

comma-separated list of pattern=N settings for file-filtered logging (only works for text log format)

--write-config-to string

If set, write the configuration values to this file and exit.

13 - Debug cluster

13.1 - Flow control

API Priority and Fairness controls the behavior of the Kubernetes API server in an overload situation. You can find more information about it in the API Priority and Fairness documentation.

Diagnostics

Every HTTP response from an API server with the priority and fairness feature enabled has two extra headers: `X-Kubernetes-PF-FlowSchema-UID` and `X-Kubernetes-PF-PriorityLevel-UID`, noting the flow schema that matched the request and the priority level to which it was assigned, respectively. The API objects' names are not included in these headers (to avoid revealing details in case the requesting user does not have permission to view them). When debugging, you can use a command such as:

```
kubectl get flowschemas -o custom-columns="uid:{metadata.uid},name:{m
kubectl get prioritylevelconfigurations -o custom-columns="uid:{metad
```

to get a mapping of UIDs to names for both FlowSchemas and PriorityLevelConfigurations.

Debug endpoints

With the `APIPriorityAndFairness` feature enabled, the `kube-apiserver` serves the following additional paths at its HTTP(S) ports.

You need to ensure you have permissions to access these endpoints. You don't have to do anything if you are using admin. Permissions can be granted if needed following the RBAC doc to access `/debug/api_priority_and_fairness/` by specifying `nonResourceURLs`.

- `/debug/api_priority_and_fairness/dump_priority_levels` - a listing of all the priority levels and the current state of each. You can fetch like this:

```
kubectl get --raw /debug/api_priority_and_fairness/dump_priority
```

The output will be in CSV and similar to this:

```
PriorityLevelName, ActiveQueues, IsIdle, IsQuiescing, WaitingReq
catch-all,         0,            true,   false,       0,
exempt,            0,            true,   false,       0,
global-default,    0,            true,   false,       0,
leader-election,   0,            true,   false,       0,
node-high,         0,            true,   false,       0,
system,            0,            true,   false,       0,
workload-high,     0,            true,   false,       0,
workload-low,      0,            true,   false,       0,
```

Explanation for selected column names:

  - `IsQuiescing` indicates if this priority level will be removed

when its queues have been drained.

- `/debug/api_priority_and_fairness/dump_queues` - a listing of all the queues and their current state. You can fetch like this:

```
kubectl get --raw /debug/api_priority_and_fairness/dump_queues
```

The output will be in CSV and similar to this:

```
PriorityLevelName, Index,  PendingRequests, ExecutingRequests, S
workload-low,      14,     27,              0,                 0
workload-low,      74,     26,              0,                 0
...
leader-election,   0,      0,               0,                 0
leader-election,   1,      0,               0,                 0
...
workload-high,     0,      0,               0,                 0
workload-high,     1,      0,               0,                 0
```

Explanation for selected column names:

  - `NextDispatchR` : The R progress meter reading, in units of seat-seconds, at which the next request will be dispatched.
  - `InitialSeatsSum` : The sum of InitialSeats associated with all requests in a given queue.
  - `MaxSeatsSum` : The sum of MaxSeats associated with all requests in a given queue.
  - `TotalWorkSum` : The sum of total work, in units of seat-seconds, of all waiting requests in a given queue.

  Note: `seat-second` (abbreviate as `ss`) is a measure of work, in units of seat-seconds, in the APF world.

- `/debug/api_priority_and_fairness/dump_requests` - a listing of all the requests including requests waiting in a queue and requests being executing. You can fetch like this:

```
kubectl get --raw /debug/api_priority_and_fairness/dump_requests
```

The output will be in CSV and similar to this:

```
PriorityLevelName, FlowSchemaName,   QueueIndex, RequestIndexInQ
exempt,            exempt,           -1,         -1,
workload-low,      service-accounts, 14,         0,
workload-low,      service-accounts, 14,         1,
```

You can get a more detailed listing with a command like this:

```
kubectl get --raw '/debug/api_priority_and_fairness/dump_request
```

The output will be in CSV and similar to this:

```
PriorityLevelName, FlowSchemaName,   QueueIndex, RequestIndexInQ
exempt,            exempt,           -1,         -1,
workload-low,      service-accounts, 14,         0,
workload-low,      service-accounts, 14,         1,
```

Explanation for selected column names:

- `QueueIndex` : The index of the queue. It will be -1 for priority levels without queues.
- `RequestIndexInQueue` : The index in the queue for a given request. It will be -1 for executing requests.
- `InitialSeats` : The number of seats will be occupied during the initial (normal) stage of execution of the request.
- `FinalSeats` : The number of seats will be occupied during the final stage of request execution, accounting for the associated WATCH notifications.
- `AdditionalLatency` : The extra time taken during the final stage of request execution. FinalSeats will be occupied during this time period. It does not mean any latency that a user will observe.
- `StartTime` : The time a request starts to execute. It will be 0001-01-01T00:00:00Z for queued requests.

Debug logging

At `-v=3` or more verbosity, the API server outputs an httplog line for every request in the API server log, and it includes the following attributes.

- `apf_fs` : the name of the flow schema to which the request was classified.
- `apf_pl` : the name of the priority level for that flow schema.
- `apf_iseats` : the number of seats determined for the initial (normal) stage of execution of the request.
- `apf_fseats` : the number of seats determined for the final stage of execution (accounting for the associated `watch` notifications) of the request.
- `apf_additionalLatency` : the duration of the final stage of execution of the request.

At higher levels of verbosity there will be log lines exposing details of how APF handled the request, primarily for debugging purposes.

Response headers

APF adds the following two headers to each HTTP response message. They won't appear in the audit log. They can be viewed from the client side. For client using `klog` , use verbosity `-v=8` or higher to view these headers.

- `X-Kubernetes-PF-FlowSchema-UID` holds the UID of the FlowSchema object to which the corresponding request was classified.
- `X-Kubernetes-PF-PriorityLevel-UID` holds the UID of the PriorityLevelConfiguration object associated with that FlowSchema.

What's next

For background information on design details for API priority and fairness, see the enhancement proposal.

14 - Configuration APIs

14.1 - Client Authentication (v1)

Resource Types

- ExecCredential

`ExecCredential`

ExecCredential is used by exec-based plugins to communicate credentials to HTTP transports.

Field	Description
`apiVersion` string	`client.authentication.k8s.io/v1`
`kind` string	`ExecCredential`
`spec` **[Required]** ExecCredentialSpec	Spec holds information passed to the plugin by the transport.
`status` ExecCredentialStatus	Status is filled in by the plugin and holds the credentials that the transport should use to contact the API.

`Cluster`

Appears in:

- ExecCredentialSpec

Cluster contains information to allow an exec plugin to communicate with the kubernetes cluster being authenticated to.

To ensure that this struct contains everything someone would need to communicate with a kubernetes cluster (just like they would via a kubeconfig), the fields should shadow "k8s.io/client-go/tools/clientcmd/api/v1".Cluster, with the exception of CertificateAuthority, since CA data will always be passed to the plugin as bytes.

Field	Description
`server` **[Required]** string	Server is the address of the kubernetes cluster (https://hostname:port).
`tls-server-name` string	TLSServerName is passed to the server for SNI and is used in the client to check server certificates against. If ServerName is empty, the hostname used to contact the server is used.

Field	Description
`insecure-skip-tls-verify` `bool`	InsecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.
`certificate-authority-data` `[]byte`	CAData contains PEM-encoded certificate authority certificates. If empty, system roots should be used.
`proxy-url` `string`	ProxyURL is the URL to the proxy to be used for all requests to this cluster.
`disable-compression` `bool`	DisableCompression allows client to opt-out of response compression for all requests to the server. This is useful to speed up requests (specifically lists) when client-server network bandwidth is ample, by saving time on compression (server-side) and decompression (client-side): https://github.com/kubernetes/kubernetes/issues/112296.
`config` `k8s.io/apimachinery/pkg/runtime.RawExtension`	Config holds additional config data that is specific to the exec plugin with regards to the cluster being authenticated to. This data is sourced from the clientcmd Cluster object's extensions[client.authentication.k8s.io/exec] field: clusters: • name: my-cluster cluster: ... extensions: ◦ name: client.authentication.k8s.io/exec # reserved extension name for per cluster exec config extension: audience: 06e3fbd18de8 # arbitrary config In some environments, the user config may be exactly the same across many clusters (i.e. call this exec plugin) minus some details that are specific to each cluster such as the audience. This field allows the per cluster config to be directly specified with the cluster info. Using this field to store secret data is not recommended as one of the prime benefits of exec plugins is that no secrets need to be stored directly in the kubeconfig.

`ExecCredentialSpec`

Appears in:

- ExecCredential

ExecCredentialSpec holds request and runtime specific information provided by the transport.

Field	Description

Field	Description
`cluster` [Cluster](#)	Cluster contains information to allow an exec plugin to communicate with the kubernetes cluster being authenticated to. Note that Cluster is non-nil only when provideClusterInfo is set to true in the exec provider config (i.e., ExecConfig.ProvideClusterInfo).

`interactive`
[Required]
`bool`
	Interactive declares whether stdin has been passed to this exec plugin.

ExecCredentialStatus

Appears in:

- ExecCredential

ExecCredentialStatus holds credentials for the transport to use.

Token and ClientKeyData are sensitive fields. This data should only be transmitted in-memory between client and exec plugin process. Exec plugin itself should at least be protected via file permissions.

Field	Description
`expirationTimestamp` meta/v1.Time	ExpirationTimestamp indicates a time when the provided credentials expire.
`token` **[Required]** `string`	Token is a bearer token used by the client for request authentication.

`clientCertificateData`
[Required]
`string`
	PEM-encoded client TLS certificates (including intermediates, if any).

`clientKeyData`
[Required]
`string`
	PEM-encoded private key for the above certificate.

14.2 - Client Authentication (v1beta1)

Resource Types

- ExecCredential

ExecCredential

ExecCredential is used by exec-based plugins to communicate credentials to HTTP transports.

Field	Description
`apiVersion` string	`client.authentication.k8s.io/v1beta1`
`kind` string	`ExecCredential`
`spec` **[Required]** ExecCredentialSpec	Spec holds information passed to the plugin by the transport.
`status` ExecCredentialStatus	Status is filled in by the plugin and holds the credentials that the transport should use to contact the API.

Cluster

Appears in:

- ExecCredentialSpec

Cluster contains information to allow an exec plugin to communicate with the kubernetes cluster being authenticated to.

To ensure that this struct contains everything someone would need to communicate with a kubernetes cluster (just like they would via a kubeconfig), the fields should shadow "k8s.io/client-go/tools/clientcmd/api/v1".Cluster, with the exception of CertificateAuthority, since CA data will always be passed to the plugin as bytes.

Field	Description
`server` **[Required]** string	Server is the address of the kubernetes cluster (https://hostname:port).
`tls-server-name` string	TLSServerName is passed to the server for SNI and is used in the client to check server certificates against. If ServerName is empty, the hostname used to contact the server is used.
`insecure-skip-tls-verify` bool	InsecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.

Field	Description
`certificate-authority-data` `[]byte`	CAData contains PEM-encoded certificate authority certificates. If empty, system roots should be used.
`proxy-url` `string`	ProxyURL is the URL to the proxy to be used for all requests to this cluster.
`disable-compression` `bool`	DisableCompression allows client to opt-out of response compression for all requests to the server. This is useful to speed up requests (specifically lists) when client-server network bandwidth is ample, by saving time on compression (server-side) and decompression (client-side): https://github.com/kubernetes/kubernetes/issues/112296.
`config` k8s.io/apimachinery/pkg/runtime.RawExtension	Config holds additional config data that is specific to the exec plugin with regards to the cluster being authenticated to. This data is sourced from the clientcmd Cluster object's extensions[client.authentication.k8s.io/exec] field: clusters: • name: my-cluster cluster: ... extensions: ◦ name: client.authentication.k8s.io/exec # reserved extension name for per cluster exec config extension: audience: 06e3fbd18de8 # arbitrary config In some environments, the user config may be exactly the same across many clusters (i.e. call this exec plugin) minus some details that are specific to each cluster such as the audience. This field allows the per cluster config to be directly specified with the cluster info. Using this field to store secret data is not recommended as one of the prime benefits of exec plugins is that no secrets need to be stored directly in the kubeconfig.

ExecCredentialSpec

Appears in:

- ExecCredential

ExecCredentialSpec holds request and runtime specific information provided by the transport.

Field	Description
`cluster` [Cluster](#)	Cluster contains information to allow an exec plugin to communicate with the kubernetes cluster being authenticated to. Note that Cluster is non-nil only when provideClusterInfo is set to true in the exec provider config (i.e., ExecConfig.ProvideClusterInfo).

Field	Description
`interactive` **[Required]** `bool`	Interactive declares whether stdin has been passed to this exec plugin.

ExecCredentialStatus

Appears in:

- ExecCredential

ExecCredentialStatus holds credentials for the transport to use.

Token and ClientKeyData are sensitive fields. This data should only be transmitted in-memory between client and exec plugin process. Exec plugin itself should at least be protected via file permissions.

Field	Description
`expirationTimestamp` `meta/v1.Time`	ExpirationTimestamp indicates a time when the provided credentials expire.
`token` **[Required]** `string`	Token is a bearer token used by the client for request authentication.
`clientCertificateData` **[Required]** `string`	PEM-encoded client TLS certificates (including intermediates, if any).
`clientKeyData` **[Required]** `string`	PEM-encoded private key for the above certificate.

14.3 - Event Rate Limit Configuration (v1alpha1)

Resource Types

- Configuration

Configuration

Configuration provides configuration for the EventRateLimit admission controller.

Field	Description
`apiVersion` string	`eventratelimit.admission.k8s.io/v1alpha1`
`kind` string	`Configuration`
`limits` **[Required]** `[]Limit`	limits are the limits to place on event queries received. Limits can be placed on events received server-wide, per namespace, per user, and per source+object. At least one limit is required.

Limit

Appears in:

- Configuration

Limit is the configuration for a particular limit type

Field	Description
`type` **[Required]** `LimitType`	type is the type of limit to which this configuration applies
`qps` **[Required]** `int32`	qps is the number of event queries per second that are allowed for this type of limit. The qps and burst fields are used together to determine if a particular event query is accepted. The qps determines how many queries are accepted once the burst amount of queries has been exhausted.

Field	Description
`burst` **[Required]** `int32`	burst is the burst number of event queries that are allowed for this type of limit. The qps and burst fields are used together to determine if a particular event query is accepted. The burst determines the maximum size of the allowance granted for a particular bucket. For example, if the burst is 10 and the qps is 3, then the admission control will accept 10 queries before blocking any queries. Every second, 3 more queries will be allowed. If some of that allowance is not used, then it will roll over to the next second, until the maximum allowance of 10 is reached.
`cacheSize` `int32`	cacheSize is the size of the LRU cache for this type of limit. If a bucket is evicted from the cache, then the allowance for that bucket is reset. If more queries are later received for an evicted bucket, then that bucket will re-enter the cache with a clean slate, giving that bucket a full allowance of burst queries. The default cache size is 4096. If limitType is 'server', then cacheSize is ignored.

`LimitType`

(Alias of `string`)

Appears in:

- Limit

LimitType is the type of the limit (e.g., per-namespace)

14.4 - Image Policy API (v1alpha1)

Resource Types

- ImageReview

ImageReview

ImageReview checks if the set of images in a pod are allowed.

Field	Description
`apiVersion` string	`imagepolicy.k8s.io/v1alpha1`
`kind` string	`ImageReview`
`metadata` meta/v1.ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata Refer to the Kubernetes API documentation for the fields of the `metadata` field.
`spec` **[Required]** ImageReviewSpec	Spec holds information about the pod being evaluated
`status` ImageReviewStatus	Status is filled in by the backend and indicates whether the pod should be allowed.

ImageReviewContainerSpec

Appears in:

- ImageReviewSpec

ImageReviewContainerSpec is a description of a container within the pod creation request.

Field	Description
`image` `string`	This can be in the form image:tag or image@SHA:012345679abcdef.

ImageReviewSpec

Appears in:

- ImageReview

ImageReviewSpec is a description of the pod creation request.

Field	Description
`containers` [`[]ImageReviewContainerSpec`]	Containers is a list of a subset of the information in each container of the Pod being created.
`annotations` `map[string]string`	Annotations is a list of key-value pairs extracted from the Pod's annotations. It only includes keys which match the pattern `*.image-policy.k8s.io/*`. It is up to each webhook backend to determine how to interpret these annotations, if at all.
`namespace` `string`	Namespace is the namespace the pod is being created in.

ImageReviewStatus

Appears in:

- ImageReview

ImageReviewStatus is the result of the review for the pod creation request.

Field	Description
`allowed` **[Required]** `bool`	Allowed indicates that all images were allowed to be run.
`reason` `string`	Reason should be empty unless Allowed is false in which case it may contain a short description of what is wrong. Kubernetes may truncate excessively long errors when displaying to the user.
`auditAnnotations` `map[string]string`	AuditAnnotations will be added to the attributes object of the admission controller request using 'AddAnnotation'. The keys should be prefix-less (i.e., the admission controller will add an appropriate prefix).

14.5 - kube-apiserver Admission (v1)

Resource Types

- AdmissionReview

AdmissionReview

AdmissionReview describes an admission review request/response.

Field	Description
`apiVersion` string	`admission.k8s.io/v1`
`kind` string	`AdmissionReview`
`request` AdmissionRequest	Request describes the attributes for the admission request.
`response` AdmissionResponse	Response describes the attributes for the admission response.

AdmissionRequest

Appears in:

- AdmissionReview

AdmissionRequest describes the admission.Attributes for the admission request.

Field	Description
`uid` **[Required]** k8s.io/apimachinery/pkg/types.UID	UID is an identifier for the individual request/response. It allows us to distinguish instances of requests which are otherwise identical (parallel requests, requests when earlier requests did not modify etc) The UID is meant to track the round trip (request/response) between the KAS and the WebHook, not the user request. It is suitable for correlating log entries between the webhook and apiserver, for either auditing or debugging.
`kind` **[Required]** meta/v1.GroupVersionKind	Kind is the fully-qualified type of object being submitted (for example, v1.Pod or autoscaling.v1.Scale)
`resource` **[Required]** meta/v1.GroupVersionResource	Resource is the fully-qualified resource being requested (for example, v1.pods)

Field	Description
`subResource` `string`	SubResource is the subresource being requested, if any (for example, "status" or "scale")
`requestKind` [meta/v1.GroupVersionKind](#)	RequestKind is the fully-qualified type of the original API request (for example, v1.Pod or autoscaling.v1.Scale). If this is specified and differs from the value in "kind", an equivalent match and conversion was performed. For example, if deployments can be modified via apps/v1 and apps/v1beta1, and a webhook registered a rule of `apiGroups:["apps"], apiVersions:["v1"], resources: ["deployments"]` and `matchPolicy: Equivalent`, an API request to apps/v1beta1 deployments would be converted and sent to the webhook with `kind: {group:"apps", version:"v1", kind:"Deployment"}` (matching the rule the webhook registered for), and `requestKind: {group:"apps", version:"v1beta1", kind:"Deployment"}` (indicating the kind of the original API request). See documentation for the "matchPolicy" field in the webhook configuration type for more details.
`requestResource` [meta/v1.GroupVersionResource](#)	RequestResource is the fully-qualified resource of the original API request (for example, v1.pods). If this is specified and differs from the value in "resource", an equivalent match and conversion was performed. For example, if deployments can be modified via apps/v1 and apps/v1beta1, and a webhook registered a rule of `apiGroups:["apps"], apiVersions:["v1"], resources: ["deployments"]` and `matchPolicy: Equivalent`, an API request to apps/v1beta1 deployments would be converted and sent to the webhook with `resource: {group:"apps", version:"v1", resource:"deployments"}` (matching the resource the webhook registered for), and `requestResource: {group:"apps", version:"v1beta1", resource:"deployments"}` (indicating the resource of the original API request). See documentation for the "matchPolicy" field in the webhook configuration type.
`requestSubResource` `string`	RequestSubResource is the name of the subresource of the original API request, if any (for example, "status" or "scale") If this is specified and differs from the value in "subResource", an equivalent match and conversion was performed. See documentation for the "matchPolicy" field in the webhook configuration type.
`name` `string`	Name is the name of the object as presented in the request. On a CREATE operation, the client may omit name and rely on the server to generate the name. If that is the case, this field will contain an empty string.

Field	Description
`namespace` `string`	Namespace is the namespace associated with the request (if any).
`operation` **[Required]** Operation	Operation is the operation being performed. This may be different than the operation requested. e.g. a patch can result in either a CREATE or UPDATE Operation.
`userInfo` **[Required]** authentication/v1.UserInfo	UserInfo is information about the requesting user
`object` k8s.io/apimachinery/pkg/runtime.RawExtension	Object is the object from the incoming request.
`oldObject` k8s.io/apimachinery/pkg/runtime.RawExtension	OldObject is the existing object. Only populated for DELETE and UPDATE requests.
`dryRun` `bool`	DryRun indicates that modifications will definitely not be persisted for this request. Defaults to false.
`options` k8s.io/apimachinery/pkg/runtime.RawExtension	Options is the operation option structure of the operation being performed. e.g. `meta.k8s.io/v1.DeleteOptions` or `meta.k8s.io/v1.CreateOptions`. This may be different than the options the caller provided. e.g. for a patch request the performed Operation might be a CREATE, in which case the Options will a `meta.k8s.io/v1.CreateOptions` even though the caller provided `meta.k8s.io/v1.PatchOptions`.

AdmissionResponse

Appears in:

- AdmissionReview

AdmissionResponse describes an admission response.

Field	Description
`uid` **[Required]** k8s.io/apimachinery/pkg/types.UID	UID is an identifier for the individual request/response. This must be copied over from the corresponding AdmissionRequest.
`allowed` **[Required]** `bool`	Allowed indicates whether or not the admission request was permitted.

Field	Description
`status` [meta/v1.Status](#)	Result contains extra details into why an admission request was denied. This field IS NOT consulted in any way if "Allowed" is "true".
`patch` `[]byte`	The patch body. Currently we only support "JSONPatch" which implements RFC 6902.
`patchType` [PatchType](#)	The type of Patch. Currently we only allow "JSONPatch".
`auditAnnotations` `map[string]string`	AuditAnnotations is an unstructured key value map set by remote admission controller (e.g. error=image-blacklisted). MutatingAdmissionWebhook and ValidatingAdmissionWebhook admission controller will prefix the keys with admission webhook name (e.g. imagepolicy.example.com/error=image-blacklisted). AuditAnnotations will be provided by the admission webhook to add additional context to the audit log for this request.
`warnings` `[]string`	warnings is a list of warning messages to return to the requesting API client. Warning messages describe a problem the client making the API request should correct or be aware of. Limit warnings to 120 characters if possible. Warnings over 256 characters and large numbers of warnings may be truncated.

Operation

(Alias of `string`)

Appears in:

- AdmissionRequest

Operation is the type of resource operation being checked for admission control

PatchType

(Alias of `string`)

Appears in:

- AdmissionResponse

PatchType is the type of patch being used to represent the mutated object

14.6 - kube-apiserver Audit Configuration (v1)

Resource Types

- Event
- EventList
- Policy
- PolicyList

Event

Appears in:

- EventList

Event captures all the information that can be included in an API audit log.

Field	Description
`apiVersion` string	`audit.k8s.io/v1`
`kind` string	`Event`
`level` **[Required]** Level	AuditLevel at which event was generated
`auditID` **[Required]** k8s.io/apimachinery/pkg/types.UID	Unique audit ID, generated for each request.
`stage` **[Required]** Stage	Stage of the request handling when this event instance was generated.
`requestURI` **[Required]** string	RequestURI is the request URI as sent by the client to a server.
`verb` **[Required]** string	Verb is the kubernetes verb associated with the request. For non-resource requests, this is the lower-cased HTTP method.
`user` **[Required]** authentication/v1.UserInfo	Authenticated user information.
`impersonatedUser` authentication/v1.UserInfo	Impersonated user information.

Field	Description
`sourceIPs` `[]string`	Source IPs, from where the request originated and intermediate proxies. The source IPs are listed from (in order): 1. X-Forwarded-For request header IPs 2. X-Real-Ip header, if not present in the X-Forwarded-For list 3. The remote address for the connection, if it doesn't match the last IP in the list up to here (X-Forwarded-For or X-Real-Ip). Note: All but the last IP can be arbitrarily set by the client.
`userAgent` `string`	UserAgent records the user agent string reported by the client. Note that the UserAgent is provided by the client, and must not be trusted.
`objectRef` `ObjectReference`	Object reference this request is targeted at. Does not apply for List-type requests, or non-resource requests.
`responseStatus` `meta/v1.Status`	The response status, populated even when the ResponseObject is not a Status type. For successful responses, this will only include the Code and StatusSuccess. For non-status type error responses, this will be auto-populated with the error Message.
`requestObject` `k8s.io/apimachinery/pkg/runtime.Unknown`	API object from the request, in JSON format. The RequestObject is recorded as-is in the request (possibly re-encoded as JSON), prior to version conversion, defaulting, admission or merging. It is an external versioned object type, and may not be a valid object on its own. Omitted for non-resource requests. Only logged at Request Level and higher.
`responseObject` `k8s.io/apimachinery/pkg/runtime.Unknown`	API object returned in the response, in JSON. The ResponseObject is recorded after conversion to the external type, and serialized as JSON. Omitted for non-resource requests. Only logged at Response Level.
`requestReceivedTimestamp` `meta/v1.MicroTime`	Time the request reached the apiserver.
`stageTimestamp` `meta/v1.MicroTime`	Time the request reached current audit stage.

Field	Description
`annotations` `map[string]string`	Annotations is an unstructured key value map stored with an audit event that may be set by plugins invoked in the request serving chain, including authentication, authorization and admission plugins. Note that these annotations are for the audit event, and do not correspond to the metadata.annotations of the submitted object. Keys should uniquely identify the informing component to avoid name collisions (e.g. podsecuritypolicy.admission.k8s.io/policy). Values should be short. Annotations are included in the Metadata level.

EventList

EventList is a list of audit Events.

Field	Description
`apiVersion` string	`audit.k8s.io/v1`
`kind` string	`EventList`
`metadata` `meta/v1.ListMeta`	No description provided.
`items` **[Required]** `[]Event`	No description provided.

Policy

Appears in:

- PolicyList

Policy defines the configuration of audit logging, and the rules for how different request categories are logged.

Field	Description
`apiVersion` string	`audit.k8s.io/v1`
`kind` string	`Policy`
`metadata` `meta/v1.ObjectMeta`	ObjectMeta is included for interoperability with API infrastructure. Refer to the Kubernetes API documentation for the fields of the `metadata` field.

Field	Description
`rules` **[Required]** []PolicyRule	Rules specify the audit Level a request should be recorded at. A request may match multiple rules, in which case the FIRST matching rule is used. The default audit level is None, but can be overridden by a catch-all rule at the end of the list. PolicyRules are strictly ordered.
`omitStages` []Stage	OmitStages is a list of stages for which no events are created. Note that this can also be specified per rule in which case the union of both are omitted.
`omitManagedFields` `bool`	OmitManagedFields indicates whether to omit the managed fields of the request and response bodies from being written to the API audit log. This is used as a global default - a value of 'true' will omit the managed fileds, otherwise the managed fields will be included in the API audit log. Note that this can also be specified per rule in which case the value specified in a rule will override the global default.

PolicyList

PolicyList is a list of audit Policies.

Field	Description
`apiVersion` string	`audit.k8s.io/v1`
`kind` string	`PolicyList`
`metadata` meta/v1.ListMeta	No description provided.
`items` **[Required]** []Policy	No description provided.

GroupResources

Appears in:

- PolicyRule

GroupResources represents resource kinds in an API group.

Field	Description
`group` `string`	Group is the name of the API group that contains the resources. The empty string represents the core API group.

Field	Description
`resources` `[]string`	Resources is a list of resources this rule applies to. For example: • `pods` matches pods. • `pods/log` matches the log subresource of pods. • `*` matches all resources and their subresources. • `pods/*` matches all subresources of pods. • `*/scale` matches all scale subresources. If wildcard is present, the validation rule will ensure resources do not overlap with each other. An empty list implies all resources and subresources in this API groups apply.
`resourceNames` `[]string`	ResourceNames is a list of resource instance names that the policy matches. Using this field requires Resources to be specified. An empty list implies that every instance of the resource is matched.

`Level`

(Alias of `string`)

Appears in:

- Event
- PolicyRule

Level defines the amount of information logged during auditing

`ObjectReference`

Appears in:

- Event

ObjectReference contains enough information to let you inspect or modify the referred object.

Field	Description
`resource` `string`	No description provided.
`namespace` `string`	No description provided.
`name` `string`	No description provided.
`uid` `k8s.io/apimachinery/pkg/types.UID`	No description provided.

Field	Description
`apiGroup` `string`	APIGroup is the name of the API group that contains the referred object. The empty string represents the core API group.
`apiVersion` `string`	APIVersion is the version of the API group that contains the referred object.
`resourceVersion` `string`	No description provided.
`subresource` `string`	No description provided.

PolicyRule

Appears in:

- Policy

PolicyRule maps requests based off metadata to an audit Level. Requests must match the rules of every field (an intersection of rules).

Field	Description
`level` **[Required]** `Level`	The Level that requests matching this rule are recorded at.
`users` `[]string`	The users (by authenticated user name) this rule applies to. An empty list implies every user.
`userGroups` `[]string`	The user groups this rule applies to. A user is considered matching if it is a member of any of the UserGroups. An empty list implies every user group.
`verbs` `[]string`	The verbs that match this rule. An empty list implies every verb.
`resources` `[]GroupResources`	Resources that this rule matches. An empty list implies all kinds in all API groups.
`namespaces` `[]string`	Namespaces that this rule matches. The empty string "" matches non-namespaced resources. An empty list implies every namespace.
`nonResourceURLs` `[]string`	NonResourceURLs is a set of URL paths that should be audited. `*` s are allowed, but only as the full, final step in the path. Examples: • `/metrics` - Log requests for apiserver metrics • `/healthz*` - Log all health checks

Field	Description
`omitStages` [`[]Stage`](#)	OmitStages is a list of stages for which no events are created. Note that this can also be specified policy wide in which case the union of both are omitted. An empty list means no restrictions will apply.
`omitManagedFields` `bool`	OmitManagedFields indicates whether to omit the managed fields of the request and response bodies from being written to the API audit log. • a value of 'true' will drop the managed fields from the API audit log • a value of 'false' indicates that the managed fileds should be included in the API audit log Note that the value, if specified, in this rule will override the global default If a value is not specified then the global default specified in Policy.OmitManagedFields will stand.

Stage

(Alias of `string`)

Appears in:

- Event
- Policy
- PolicyRule

Stage defines the stages in request handling that audit events may be generated.

14.7 - kube-apiserver Configuration (v1)

Package v1 is the v1 version of the API.

Resource Types

- AdmissionConfiguration
- EncryptionConfiguration

AdmissionConfiguration

AdmissionConfiguration provides versioned configuration for admission controllers.

Field	Description
`apiVersion` string	`apiserver.config.k8s.io/v1`
`kind` string	`AdmissionConfiguration`
`plugins` []AdmissionPluginConfiguration	Plugins allows specifying a configuration per admission control plugin.

EncryptionConfiguration

EncryptionConfiguration stores the complete configuration for encryption providers. It also allows the use of wildcards to specify the resources that should be encrypted. Use '*.<group>' to encrypt all resources within a group or '*.*' to encrypt all resources. '*.' can be used to encrypt all resource in the core group. '*.*' will encrypt all resources, even custom resources that are added after API server start. Use of wildcards that overlap within the same resource list or across multiple entries are not allowed since part of the configuration would be ineffective. Resource lists are processed in order, with earlier lists taking precedence.

Example:

```
kind: EncryptionConfiguration
apiVersion: apiserver.config.k8s.io/v1
resources:
- resources:
  - events
  providers:
  - identity: {}  # do not encrypt events even though *.* is specifie
- resources:
  - secrets
  - configmaps
  - pandas.awesome.bears.example
  providers:
  - aescbc:
      keys:
      - name: key1
        secret: c2VjcmV0IGlzIHNlY3VyZQ==
- resources:
  - '*.apps'
  providers:
  - aescbc:
      keys:
      - name: key2
        secret: c2VjcmV0IGlzIHNlY3VyZSwgb3IgaXMgaXQ/Cg==
- resources:
  - '*.*'
  providers:
  - aescbc:
      keys:
      - name: key3
        secret: c2VjcmV0IGlzIHNlY3VyZSwgSSB0aGluaw==
```

Field	Description
`apiVersion` string	`apiserver.config.k8s.io/v1`
`kind` string	`EncryptionConfiguration`
`resources` **[Required]** [][]ResourceConfiguration	resources is a list containing resources, and their corresponding encryption providers.

AESConfiguration

Appears in:

- ProviderConfiguration

AESConfiguration contains the API configuration for an AES transformer.

Field	Description
keys **[Required]** [][]Key	keys is a list of keys to be used for creating the AES transformer. Each key has to be 32 bytes long for AES-CBC and 16, 24 or 32 bytes for AES-GCM.

AdmissionPluginConfiguration

Appears in:

- AdmissionConfiguration

AdmissionPluginConfiguration provides the configuration for a single plug-in.

Field	Description
`name` **[Required]** `string`	Name is the name of the admission controller. It must match the registered admission plugin name.
`path` `string`	Path is the path to a configuration file that contains the plugin's configuration
`configuration` k8s.io/apimachinery/pkg/runtime.Unknown	Configuration is an embedded configuration object to be used as the plugin's configuration. If present, it will be used instead of the path to the configuration file.

IdentityConfiguration

Appears in:

- ProviderConfiguration

IdentityConfiguration is an empty struct to allow identity transformer in provider configuration.

KMSConfiguration

Appears in:

- ProviderConfiguration

KMSConfiguration contains the name, cache size and path to configuration file for a KMS based envelope transformer.

Field	Description
`apiVersion` `string`	apiVersion of KeyManagementService
`name` **[Required]** `string`	name is the name of the KMS plugin to be used.
`cachesize` `int32`	cachesize is the maximum number of secrets which are cached in memory. The default value is 1000. Set to a negative value to disable caching. This field is only allowed for KMS v1 providers.
`endpoint` **[Required]** `string`	endpoint is the gRPC server listening address, for example "unix:///var/run/kms-provider.sock".
`timeout` meta/v1.Duration	timeout for gRPC calls to kms-plugin (ex. 5s). The default is 3 seconds.

Key

Appears in:

- AESConfiguration
- SecretboxConfiguration

Key contains name and secret of the provided key for a transformer.

Field	Description
`name` **[Required]** `string`	name is the name of the key to be used while storing data to disk.
`secret` **[Required]** `string`	secret is the actual key, encoded in base64.

ProviderConfiguration

Appears in:

- ResourceConfiguration

ProviderConfiguration stores the provided configuration for an encryption provider.

Field	Description
`aesgcm` **[Required]** AESConfiguration	aesgcm is the configuration for the AES-GCM transformer.
`aescbc` **[Required]** AESConfiguration	aescbc is the configuration for the AES-CBC transformer.
`secretbox` **[Required]** SecretboxConfiguration	secretbox is the configuration for the Secretbox based transformer.
`identity` **[Required]** IdentityConfiguration	identity is the (empty) configuration for the identity transformer.
`kms` **[Required]** KMSConfiguration	kms contains the name, cache size and path to configuration file for a KMS based envelope transformer.

ResourceConfiguration

Appears in:

- EncryptionConfiguration

ResourceConfiguration stores per resource configuration.

Field	Description

Field	Description
`resources` **[Required]** `[]string`	resources is a list of kubernetes resources which have to be encrypted. The resource names are derived from `resource` or `resource.group` of the group/version/resource. eg: pandas.awesome.bears.example is a custom resource with 'group': awesome.bears.example, 'resource': pandas. Use '*.*' to encrypt all resources and '*.<group>' to encrypt all resources in a specific group. eg: '*.awesome.bears.example' will encrypt all resources in the group 'awesome.bears.example'. eg: '*.' will encrypt all resources in the core group (such as pods, configmaps, etc).
`providers` **[Required]** []ProviderConfiguration	providers is a list of transformers to be used for reading and writing the resources to disk. eg: aesgcm, aescbc, secretbox, identity, kms.

SecretboxConfiguration

Appears in:

- ProviderConfiguration

SecretboxConfiguration contains the API configuration for an Secretbox transformer.

Field	Description
keys **[Required]** []Key	keys is a list of keys to be used for creating the Secretbox transformer. Each key has to be 32 bytes long.

14.8 - kube-apiserver Configuration (v1alpha1)

Package v1alpha1 is the v1alpha1 version of the API.

Resource Types

- AdmissionConfiguration
- AuthenticationConfiguration
- AuthorizationConfiguration
- EgressSelectorConfiguration
- TracingConfiguration

TracingConfiguration

Appears in:

- KubeletConfiguration
- TracingConfiguration

TracingConfiguration provides versioned configuration for OpenTelemetry tracing clients.

Field	Description
`endpoint` `string`	Endpoint of the collector this component will report traces to. The connection is insecure, and does not currently support TLS. Recommended is unset, and endpoint is the otlp grpc default, localhost:4317.
`samplingRatePerMillion` `int32`	SamplingRatePerMillion is the number of samples to collect per million spans. Recommended is unset. If unset, sampler respects its parent span's sampling rate, but otherwise never samples.

AdmissionConfiguration

AdmissionConfiguration provides versioned configuration for admission controllers.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1alpha1`
`kind` string	`AdmissionConfiguration`
`plugins` []AdmissionPluginConfiguration	Plugins allows specifying a configuration per admission control plugin.

AuthenticationConfiguration

AuthenticationConfiguration provides versioned configuration for authentication.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1alpha1`
`kind` string	`AuthenticationConfiguration`
`jwt` **[Required]** [][]JWTAuthenticator	jwt is a list of authenticator to authenticate Kubernetes users using JWT compliant tokens. The authenticator will attempt to parse a raw ID token, verify it's been signed by the configured issuer. The public key to verify the signature is discovered from the issuer's public endpoint using OIDC discovery. For an incoming token, each JWT authenticator will be attempted in the order in which it is specified in this list. Note however that other authenticators may run before or after the JWT authenticators. The specific position of JWT authenticators in relation to other authenticators is neither defined nor stable across releases. Since each JWT authenticator must have a unique issuer URL, at most one JWT authenticator will attempt to cryptographically validate the token. The minimum valid JWT payload must contain the following claims: { "iss": "https://issuer.example.com", "aud": ["audience"], "exp": 1234567890, "": "username" }
`anonymous` **[Required]** AnonymousAuthConfig	If present --anonymous-auth must not be set

AuthorizationConfiguration

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1alpha1`
`kind` string	`AuthorizationConfiguration`
`authorizers` **[Required]** [][]AuthorizerConfiguration	Authorizers is an ordered list of authorizers to authorize requests against. This is similar to the --authorization-modes kube-apiserver flag Must be at least one.

EgressSelectorConfiguration

EgressSelectorConfiguration provides versioned configuration for egress selector clients.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1alpha1`
`kind` string	`EgressSelectorConfiguration`
`egressSelections` **[Required]** [`[]EgressSelection`](#)	connectionServices contains a list of egress selection client configurations

TracingConfiguration

TracingConfiguration provides versioned configuration for tracing clients.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1alpha1`
`kind` string	`TracingConfiguration`
`TracingConfiguration` **[Required]** [`TracingConfiguration`](#)	(Members of `TracingConfiguration` are embedded into this type.) Embed the component config tracing configuration struct

AdmissionPluginConfiguration

Appears in:

- AdmissionConfiguration

AdmissionPluginConfiguration provides the configuration for a single plug-in.

Field	Description
`name` **[Required]** `string`	Name is the name of the admission controller. It must match the registered admission plugin name.
`path` `string`	Path is the path to a configuration file that contains the plugin's configuration
`configuration` [`k8s.io/apimachinery/pkg/runtime.Unknown`](#)	Configuration is an embedded configuration object to be used as the plugin's configuration. If present, it will be used instead of the path to the configuration file.

AnonymousAuthCondition

Appears in:

- AnonymousAuthConfig

AnonymousAuthCondition describes the condition under which anonymous auth should be enabled.

Field	Description
`path` **[Required]** `string`	Path for which anonymous auth is enabled.

AnonymousAuthConfig

Appears in:

- AuthenticationConfiguration

AnonymousAuthConfig provides the configuration for the anonymous authenticator.

Field	Description
`enabled` **[Required]** `bool`	No description provided.
`conditions` **[Required]** []AnonymousAuthCondition	If set, anonymous auth is only allowed if the request meets one of the conditions.

AudienceMatchPolicyType

(Alias of `string`)

Appears in:

- Issuer

AudienceMatchPolicyType is a set of valid values for issuer.audienceMatchPolicy

AuthorizerConfiguration

Appears in:

- AuthorizationConfiguration

Field	Description
`type` **[Required]** `string`	Type refers to the type of the authorizer "Webhook" is supported in the generic API server Other API servers may support additional authorizer types like Node, RBAC, ABAC, etc.

Field	Description
`name` **[Required]** `string`	Name used to describe the webhook This is explicitly used in monitoring machinery for metrics Note: Names must be DNS1123 labels like `myauthorizername` or subdomains like `myauthorizer.example.domain` Required, with no default
`webhook` **[Required]** WebhookConfiguration	Webhook defines the configuration for a Webhook authorizer Must be defined when Type=Webhook Must not be defined when Type!=Webhook

ClaimMappings

Appears in:

- JWTAuthenticator

ClaimMappings provides the configuration for claim mapping

Field	Description
`username` **[Required]** PrefixedClaimOrExpression	username represents an option for the username attribute. The claim's value must be a singular string. Same as the --oidc-username-claim and --oidc-username-prefix flags. If username.expression is set, the expression must produce a string value. If username.expression uses 'claims.email', then 'claims.email_verified' must be used in username.expression or extra[*].valueExpression or claimValidationRules[*].expression. An example claim validation rule expression that matches the validation automatically applied when username.claim is set to 'email' is 'claims.?email_verified.orValue(true)'. In the flag based approach, the --oidc-username-claim and --oidc-username-prefix are optional. If --oidc-username-claim is not set, the default value is "sub". For the authentication config, there is no defaulting for claim or prefix. The claim and prefix must be set explicitly. For claim, if --oidc-username-claim was not set with legacy flag approach, configure username.claim="sub" in the authentication config. For prefix: (1) --oidc-username-prefix="-", no prefix was added to the username. For the same behavior using authentication config, set username.prefix="" (2) --oidc-username-prefix="" and --oidc-username-claim != "email", prefix was "<value of --oidc-issuer-url>#". For the same behavior using authentication config, set username.prefix="#" (3) --oidc-username-prefix="". For the same behavior using authentication config, set username.prefix=""

Field	Description
`groups` [PrefixedClaimOrExpression](#)	groups represents an option for the groups attribute. The claim's value must be a string or string array claim. If groups.claim is set, the prefix must be specified (and can be the empty string). If groups.expression is set, the expression must produce a string or string array value. "", [], and null values are treated as the group mapping not being present.
`uid` [ClaimOrExpression](#)	uid represents an option for the uid attribute. Claim must be a singular string claim. If uid.expression is set, the expression must produce a string value.
`extra` []ExtraMapping	extra represents an option for the extra attribute. expression must produce a string or string array value. If the value is empty, the extra mapping will not be present. hard-coded extra key/value • key: "foo" valueExpression: "'bar'" This will result in an extra attribute - foo: ["bar"] hard-coded key, value copying claim value • key: "foo" valueExpression: "claims.some_claim" This will result in an extra attribute - foo: [value of some_claim] hard-coded key, value derived from claim value • key: "admin" valueExpression: '(has(claims.is_admin) && claims.is_admin) ? "true":"" This will result in: • if is_admin claim is present and true, extra attribute - admin: ["true"] • if is_admin claim is present and false or is_admin claim is not present, no extra attribute will be added

ClaimOrExpression

Appears in:

- ClaimMappings

ClaimOrExpression provides the configuration for a single claim or expression.

Field	Description
`claim` `string`	claim is the JWT claim to use. Either claim or expression must be set. Mutually exclusive with expression.

Field	Description
`expression` `string`	expression represents the expression which will be evaluated by CEL. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/ Mutually exclusive with claim.

ClaimValidationRule

Appears in:

- JWTAuthenticator

ClaimValidationRule provides the configuration for a single claim validation rule.

Field	Description
`claim` `string`	claim is the name of a required claim. Same as --oidc-required-claim flag. Only string claim keys are supported. Mutually exclusive with expression and message.
`requiredValue` `string`	requiredValue is the value of a required claim. Same as --oidc-required-claim flag. Only string claim values are supported. If claim is set and requiredValue is not set, the claim must be present with a value set to the empty string. Mutually exclusive with expression and message.
`expression` `string`	expression represents the expression which will be evaluated by CEL. Must produce a boolean. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Must return true for the validation to pass. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/ Mutually exclusive with claim and requiredValue.

Field	Description
`message` `string`	message customizes the returned error message when expression returns false. message is a literal string. Mutually exclusive with claim and requiredValue.

Connection

Appears in:

- EgressSelection

Connection provides the configuration for a single egress selection client.

Field	Description
`proxyProtocol` **[Required]** ProtocolType	Protocol is the protocol used to connect from client to the konnectivity server.
`transport` Transport	Transport defines the transport configurations we use to dial to the konnectivity server. This is required if ProxyProtocol is HTTPConnect or GRPC.

EgressSelection

Appears in:

- EgressSelectorConfiguration

EgressSelection provides the configuration for a single egress selection client.

Field	Description
`name` **[Required]** `string`	name is the name of the egress selection. Currently supported values are "controlplane", "master", "etcd" and "cluster" The "master" egress selector is deprecated in favor of "controlplane"
`connection` **[Required]** Connection	connection is the exact information used to configure the egress selection

ExtraMapping

Appears in:

- ClaimMappings

ExtraMapping provides the configuration for a single extra mapping.

Field	Description

Field	Description
`key` **[Required]** `string`	key is a string to use as the extra attribute key. key must be a domain-prefix path (e.g. example.org/foo). All characters before the first "/" must be a valid subdomain as defined by RFC 1123. All characters trailing the first "/" must be valid HTTP Path characters as defined by RFC 3986. key must be lowercase. Required to be unique.
`valueExpression` **[Required]** `string`	valueExpression is a CEL expression to extract extra attribute value. valueExpression must produce a string or string array value. "", [], and null values are treated as the extra mapping not being present. Empty string values contained within a string array are filtered out. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/

Issuer

Appears in:

- JWTAuthenticator

Issuer provides the configuration for an external provider's specific settings.

Field	Description
`url` **[Required]** `string`	url points to the issuer URL in a format https://url or https://url/path. This must match the "iss" claim in the presented JWT, and the issuer returned from discovery. Same value as the --oidc-issuer-url flag. Discovery information is fetched from "{url}/.well-known/openid-configuration" unless overridden by discoveryURL. Required to be unique across all JWT authenticators. Note that egress selection configuration is not used for this network connection.

Field	Description
`discoveryURL` `string`	discoveryURL, if specified, overrides the URL used to fetch discovery information instead of using "{url}/.well-known/openid-configuration". The exact value specified is used, so "/.well-known/openid-configuration" must be included in discoveryURL if needed. The "issuer" field in the fetched discovery information must match the "issuer.url" field in the AuthenticationConfiguration and will be used to validate the "iss" claim in the presented JWT. This is for scenarios where the well-known and jwks endpoints are hosted at a different location than the issuer (such as locally in the cluster). Example: A discovery url that is exposed using kubernetes service 'oidc' in namespace 'oidc-namespace' and discovery information is available at '/.well-known/openid-configuration'. discoveryURL: "https://oidc.oidc-namespace/.well-known/openid-configuration" certificateAuthority is used to verify the TLS connection and the hostname on the leaf certificate must be set to 'oidc.oidc-namespace'. curl https://oidc.oidc-namespace/.well-known/openid-configuration (.discoveryURL field) { issuer: "https://oidc.example.com" (.url field) } discoveryURL must be different from url. Required to be unique across all JWT authenticators. Note that egress selection configuration is not used for this network connection.
`certificateAuthority` `string`	certificateAuthority contains PEM-encoded certificate authority certificates used to validate the connection when fetching discovery information. If unset, the system verifier is used. Same value as the content of the file referenced by the --oidc-ca-file flag.
`audiences` **[Required]** `[]string`	audiences is the set of acceptable audiences the JWT must be issued to. At least one of the entries must match the "aud" claim in presented JWTs. Same value as the --oidc-client-id flag (though this field supports an array). Required to be non-empty.

Field	Description
`audienceMatchPolicy` [AudienceMatchPolicyType]	audienceMatchPolicy defines how the "audiences" field is used to match the "aud" claim in the presented JWT. Allowed values are: 1. "MatchAny" when multiple audiences are specified and 2. empty (or unset) or "MatchAny" when a single audience is specified. - MatchAny: the "aud" claim in the presented JWT must match at least one of the entries in the "audiences" field. For example, if "audiences" is ["foo", "bar"], the "aud" claim in the presented JWT must contain either "foo" or "bar" (and may contain both). - "": The match policy can be empty (or unset) when a single audience is specified in the "audiences" field. The "aud" claim in the presented JWT must contain the single audience (and may contain others). For more nuanced audience validation, use claimValidationRules. example: claimValidationRule[].expression: 'sets.equivalent(claims.aud, ["bar", "foo", "baz"])' to require an exact match.

JWTAuthenticator

Appears in:

- AuthenticationConfiguration

JWTAuthenticator provides the configuration for a single JWT authenticator.

Field	Description
`issuer` **[Required]** Issuer	issuer contains the basic OIDC provider connection options.
`claimValidationRules` []ClaimValidationRule	claimValidationRules are rules that are applied to validate token claims to authenticate users.
`claimMappings` **[Required]** ClaimMappings	claimMappings points claims of a token to be treated as user attributes.
`userValidationRules` []UserValidationRule	userValidationRules are rules that are applied to final user before completing authentication. These allow invariants to be applied to incoming identities such as preventing the use of the system: prefix that is commonly used by Kubernetes components. The validation rules are logically ANDed together and must all return true for the validation to pass.

PrefixedClaimOrExpression

Appears in:

- ClaimMappings

PrefixedClaimOrExpression provides the configuration for a single prefixed claim or expression.

Field	Description
`claim` `string`	claim is the JWT claim to use. Mutually exclusive with expression.
`prefix` `string`	prefix is prepended to claim's value to prevent clashes with existing names. prefix needs to be set if claim is set and can be the empty string. Mutually exclusive with expression.
`expression` `string`	expression represents the expression which will be evaluated by CEL. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/ Mutually exclusive with claim and prefix.

ProtocolType

(Alias of `string`)

Appears in:

- Connection

ProtocolType is a set of valid values for Connection.ProtocolType

TCPTransport

Appears in:

- Transport

TCPTransport provides the information to connect to konnectivity server via TCP

Field	Description

Field	Description
`url` **[Required]** `string`	URL is the location of the konnectivity server to connect to. As an example it might be "https://127.0.0.1:8131"
`tlsConfig` [TLSConfig](#)	TLSConfig is the config needed to use TLS when connecting to konnectivity server

TLSConfig

Appears in:

- TCPTransport

TLSConfig provides the authentication information to connect to konnectivity server Only used with TCPTransport

Field	Description
`caBundle` `string`	caBundle is the file location of the CA to be used to determine trust with the konnectivity server. Must be absent/empty if TCPTransport.URL is prefixed with http:// If absent while TCPTransport.URL is prefixed with https://, default to system trust roots.
`clientKey` `string`	clientKey is the file location of the client key to be used in mtls handshakes with the konnectivity server. Must be absent/empty if TCPTransport.URL is prefixed with http:// Must be configured if TCPTransport.URL is prefixed with https://
`clientCert` `string`	clientCert is the file location of the client certificate to be used in mtls handshakes with the konnectivity server. Must be absent/empty if TCPTransport.URL is prefixed with http:// Must be configured if TCPTransport.URL is prefixed with https://

Transport

Appears in:

- Connection

Transport defines the transport configurations we use to dial to the konnectivity server

Field	Description
`tcp` [TCPTransport](#)	TCP is the TCP configuration for communicating with the konnectivity server via TCP ProxyProtocol of GRPC is not supported with TCP transport at the moment Requires at least one of TCP or UDS to be set

Field	Description
uds UDSTransport	UDS is the UDS configuration for communicating with the konnectivity server via UDS Requires at least one of TCP or UDS to be set

UDSTransport

Appears in:

- Transport

UDSTransport provides the information to connect to konnectivity server via UDS

Field	Description
udsName **[Required]** string	UDSName is the name of the unix domain socket to connect to konnectivity server This does not use a unix:// prefix. (Eg: /etc/srv/kubernetes/konnectivity-server/konnectivity-server.socket)

UserValidationRule

Appears in:

- JWTAuthenticator

UserValidationRule provides the configuration for a single user info validation rule.

Field	Description
expression **[Required]** string	expression represents the expression which will be evaluated by CEL. Must return true for the validation to pass. CEL expressions have access to the contents of UserInfo, organized into CEL variable: • 'user' - authentication.k8s.io/v1, Kind=UserInfo object Refer to https://github.com/kubernetes/api/blob/release-1.28/authentication/v1/types.go#L105-L122 for the definition. API documentation: https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.28/#userinfo-v1-authentication-k8s-io Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/
message string	message customizes the returned error message when rule returns false. message is a literal string.

WebhookConfiguration

Appears in:

- AuthorizerConfiguration

Field	Description
`authorizedTTL` **[Required]** meta/v1.Duration	The duration to cache 'authorized' responses from the webhook authoriz Same as setting `--authorization-webhook-cache-authorized-ttl` flag Default: 5m0s
`unauthorizedTTL` **[Required]** meta/v1.Duration	The duration to cache 'unauthorized' responses from the webhook authoriz Same as setting `--authorization-webhook-cache-unauthorized-ttl` fl Default: 30s
`timeout` **[Required]** meta/v1.Duration	Timeout for the webhook request Max allowed value is 30s. Required, no defa value.
`subjectAccessReviewVersion` **[Required]** `string`	The API version of the authorization.k8 SubjectAccessReview to send to and e from the webhook. Same as setting `--authorization-webhook-version` fla Valid values: v1beta1, v1 Required, no default value
`matchConditionSubjectAccessReviewVersion` **[Required]** `string`	MatchConditionSubjectAccessReviewV specifies the SubjectAccessReview vers the CEL expressions are evaluated aga Valid values: v1 Required, no default v
`failurePolicy` **[Required]** `string`	Controls the authorization decision wh webhook request fails to complete or returns a malformed response or erro evaluating matchConditions. Valid valu • NoOpinion: continue to subsequ authorizers to see if one of them allows the request • Deny: reject the request without consulting subsequent authorize Required, with no default.
`connectionInfo` **[Required]** WebhookConnectionInfo	ConnectionInfo defines how we talk to webhook

Field	Description
`matchConditions` **[Required]** []WebhookMatchCondition	matchConditions is a list of conditions must be met for a request to be sent t webhook. An empty list of matchCond matches all requests. There are a max of 64 match conditions allowed. The exact matching logic is (in order): 1. If at least one matchCondition evaluates to FALSE, then the web is skipped. 2. If ALL matchConditions evaluate TRUE, then the webhook is called 3. If at least one matchCondition evaluates to an error (but none a FALSE): ◦ If failurePolicy=Deny, then t webhook rejects the reques ◦ If failurePolicy=NoOpinion, the error is ignored and the webhook is skipped

WebhookConnectionInfo

Appears in:

- WebhookConfiguration

Field	Description
`type` **[Required]** `string`	Controls how the webhook should communicate with the server. Valid values: • KubeConfigFile: use the file specified in kubeConfigFile to locate the server. • InClusterConfig: use the in-cluster configuration to call the SubjectAccessReview API hosted by kube-apiserver. This mode is not allowed for kube-apiserver.

`kubeConfigFile`
[Required]
`string`
	Path to KubeConfigFile for connection info Required, if connectionInfo.Type is KubeConfig

WebhookMatchCondition

Appears in:

- WebhookConfiguration

Field	Description

Field	Description
`expression` **[Required]** `string`	expression represents the expression which will be evaluated by CEL. Must evaluate to bool. CEL expressions have access to the contents of the SubjectAccessReview in v1 version. If version specified by subjectAccessReviewVersion in the request variable is v1beta1, the contents would be converted to the v1 version before evaluating the CEL expression. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/

14.9 - kube-apiserver Configuration (v1beta1)

Package v1beta1 is the v1beta1 version of the API.

Resource Types

- AuthenticationConfiguration
- AuthorizationConfiguration
- EgressSelectorConfiguration
- TracingConfiguration

TracingConfiguration

Appears in:

- KubeletConfiguration
- TracingConfiguration
- TracingConfiguration

TracingConfiguration provides versioned configuration for OpenTelemetry tracing clients.

Field	Description
`endpoint` `string`	Endpoint of the collector this component will report traces to. The connection is insecure, and does not currently support TLS. Recommended is unset, and endpoint is the otlp grpc default, localhost:4317.
`samplingRatePerMillion` `int32`	SamplingRatePerMillion is the number of samples to collect per million spans. Recommended is unset. If unset, sampler respects its parent span's sampling rate, but otherwise never samples.

AuthenticationConfiguration

AuthenticationConfiguration provides versioned configuration for authentication.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1beta1`
`kind` string	`AuthenticationConfiguration`

Field	Description
`jwt` **[Required]** [`[]JWTAuthenticator`](#)	jwt is a list of authenticator to authenticate Kubernetes users using JWT compliant tokens. The authenticator will attempt to parse a raw ID token, verify it's been signed by the configured issuer. The public key to verify the signature is discovered from the issuer's public endpoint using OIDC discovery. For an incoming token, each JWT authenticator will be attempted in the order in which it is specified in this list. Note however that other authenticators may run before or after the JWT authenticators. The specific position of JWT authenticators in relation to other authenticators is neither defined nor stable across releases. Since each JWT authenticator must have a unique issuer URL, at most one JWT authenticator will attempt to cryptographically validate the token. The minimum valid JWT payload must contain the following claims: { "iss": "https://issuer.example.com", "aud": ["audience"], "exp": 1234567890, "": "username" }
`anonymous` **[Required]** [`AnonymousAuthConfig`](#)	If present --anonymous-auth must not be set

AuthorizationConfiguration

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1beta1`
`kind` string	`AuthorizationConfiguration`
`authorizers` **[Required]** [`[]AuthorizerConfiguration`](#)	Authorizers is an ordered list of authorizers to authorize requests against. This is similar to the --authorization-modes kube-apiserver flag Must be at least one.

EgressSelectorConfiguration

EgressSelectorConfiguration provides versioned configuration for egress selector clients.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1beta1`
`kind` string	`EgressSelectorConfiguration`
`egressSelections` **[Required]** [`[]EgressSelection`](#)	connectionServices contains a list of egress selection client configurations

TracingConfiguration

TracingConfiguration provides versioned configuration for tracing clients.

Field	Description
`apiVersion` string	`apiserver.k8s.io/v1beta1`
`kind` string	`TracingConfiguration`
`TracingConfiguration` **[Required]** TracingConfiguration	(Members of `TracingConfiguration` are embedded into this type.) Embed the component config tracing configuration struct

AnonymousAuthCondition

Appears in:

- AnonymousAuthConfig

AnonymousAuthCondition describes the condition under which anonymous auth should be enabled.

Field	Description
`path` **[Required]** string	Path for which anonymous auth is enabled.

AnonymousAuthConfig

Appears in:

- AuthenticationConfiguration

AnonymousAuthConfig provides the configuration for the anonymous authenticator.

Field	Description
`enabled` **[Required]** bool	No description provided.
`conditions` **[Required]** []AnonymousAuthCondition	If set, anonymous auth is only allowed if the request meets one of the conditions.

AudienceMatchPolicyType

(Alias of `string`)

Appears in:

- Issuer

AudienceMatchPolicyType is a set of valid values for issuer.audienceMatchPolicy

AuthorizerConfiguration

Appears in:

- AuthorizationConfiguration

Field	Description
`type` **[Required]** `string`	Type refers to the type of the authorizer "Webhook" is supported in the generic API server Other API servers may support additional authorizer types like Node, RBAC, ABAC, etc.
`name` **[Required]** `string`	Name used to describe the webhook This is explicitly used in monitoring machinery for metrics Note: Names must be DNS1123 labels like `myauthorizername` or subdomains like `myauthorizer.example.domain` Required, with no default
`webhook` **[Required]** `WebhookConfiguration`	Webhook defines the configuration for a Webhook authorizer Must be defined when Type=Webhook Must not be defined when Type!=Webhook

ClaimMappings

Appears in:

- JWTAuthenticator

ClaimMappings provides the configuration for claim mapping

Field	Description

Field	Description
`username` **[Required]** [PrefixedClaimOrExpression](#)	username represents an option for the username attribute. The claim's value must be a singular string. Same as the --oidc-username-claim and --oidc-username-prefix flags. If username.expression is set, the expression must produce a string value. If username.expression uses 'claims.email', then 'claims.email_verified' must be used in username.expression or extra[*].valueExpression or claimValidationRules[*].expression. An example claim validation rule expression that matches the validation automatically applied when username.claim is set to 'email' is 'claims.?email_verified.orValue(true)'. In the flag based approach, the --oidc-username-claim and --oidc-username-prefix are optional. If --oidc-username-claim is not set, the default value is "sub". For the authentication config, there is no defaulting for claim or prefix. The claim and prefix must be set explicitly. For claim, if --oidc-username-claim was not set with legacy flag approach, configure username.claim="sub" in the authentication config. For prefix: (1) --oidc-username-prefix="-", no prefix was added to the username. For the same behavior using authentication config, set username.prefix="" (2) --oidc-username-prefix="" and --oidc-username-claim != "email", prefix was "<value of --oidc-issuer-url>#". For the same behavior using authentication config, set username.prefix="#" (3) --oidc-username-prefix="". For the same behavior using authentication config, set username.prefix=""
`groups` [PrefixedClaimOrExpression](#)	groups represents an option for the groups attribute. The claim's value must be a string or string array claim. If groups.claim is set, the prefix must be specified (and can be the empty string). If groups.expression is set, the expression must produce a string or string array value. "", [], and null values are treated as the group mapping not being present.
`uid` [ClaimOrExpression](#)	uid represents an option for the uid attribute. Claim must be a singular string claim. If uid.expression is set, the expression must produce a string value.

Field	Description
`extra` []ExtraMapping	extra represents an option for the extra attribute. expression must produce a string or string array value. If the value is empty, the extra mapping will not be present. hard-coded extra key/value • key: "foo" valueExpression: "'bar'" This will result in an extra attribute - foo: ["bar"] hard-coded key, value copying claim value • key: "foo" valueExpression: "claims.some_claim" This will result in an extra attribute - foo: [value of some_claim] hard-coded key, value derived from claim value • key: "admin" valueExpression: '(has(claims.is_admin) && claims.is_admin) ? "true":"" This will result in: • if is_admin claim is present and true, extra attribute - admin: ["true"] • if is_admin claim is present and false or is_admin claim is not present, no extra attribute will be added

ClaimOrExpression

Appears in:

- ClaimMappings

ClaimOrExpression provides the configuration for a single claim or expression.

Field	Description
`claim` `string`	claim is the JWT claim to use. Either claim or expression must be set. Mutually exclusive with expression.
`expression` `string`	expression represents the expression which will be evaluated by CEL. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/ Mutually exclusive with claim.

ClaimValidationRule

Appears in:

- JWTAuthenticator

ClaimValidationRule provides the configuration for a single claim validation rule.

Field	Description
`claim` `string`	claim is the name of a required claim. Same as --oidc-required-claim flag. Only string claim keys are supported. Mutually exclusive with expression and message.
`requiredValue` `string`	requiredValue is the value of a required claim. Same as --oidc-required-claim flag. Only string claim values are supported. If claim is set and requiredValue is not set, the claim must be present with a value set to the empty string. Mutually exclusive with expression and message.
`expression` `string`	expression represents the expression which will be evaluated by CEL. Must produce a boolean. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Must return true for the validation to pass. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/ Mutually exclusive with claim and requiredValue.
`message` `string`	message customizes the returned error message when expression returns false. message is a literal string. Mutually exclusive with claim and requiredValue.

Connection

Appears in:

- EgressSelection

Connection provides the configuration for a single egress selection client.

Field	Description
`proxyProtocol` **[Required]** ProtocolType	Protocol is the protocol used to connect from client to the konnectivity server.

Field	Description
`transport` [Transport](#)	Transport defines the transport configurations we use to dial to the konnectivity server. This is required if ProxyProtocol is HTTPConnect or GRPC.

EgressSelection

Appears in:

- EgressSelectorConfiguration

EgressSelection provides the configuration for a single egress selection client.

Field	Description
`name` **[Required]** `string`	name is the name of the egress selection. Currently supported values are "controlplane", "master", "etcd" and "cluster" The "master" egress selector is deprecated in favor of "controlplane"
`connection` **[Required]** [Connection](#)	connection is the exact information used to configure the egress selection

ExtraMapping

Appears in:

- ClaimMappings

ExtraMapping provides the configuration for a single extra mapping.

Field	Description
`key` **[Required]** `string`	key is a string to use as the extra attribute key. key must be a domain-prefix path (e.g. example.org/foo). All characters before the first "/" must be a valid subdomain as defined by RFC 1123. All characters trailing the first "/" must be valid HTTP Path characters as defined by RFC 3986. key must be lowercase. Required to be unique.

Field	Description
`valueExpression` **[Required]** `string`	valueExpression is a CEL expression to extract extra attribute value. valueExpression must produce a string or string array value. "", [], and null values are treated as the extra mapping not being present. Empty string values contained within a string array are filtered out. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/

`Issuer`

Appears in:

- JWTAuthenticator

Issuer provides the configuration for an external provider's specific settings.

Field	Description
`url` **[Required]** `string`	url points to the issuer URL in a format https://url or https://url/path. This must match the "iss" claim in the presented JWT, and the issuer returned from discovery. Same value as the --oidc-issuer-url flag. Discovery information is fetched from "{url}/.well-known/openid-configuration" unless overridden by discoveryURL. Required to be unique across all JWT authenticators. Note that egress selection configuration is not used for this network connection.

Field	Description
`discoveryURL` `string`	discoveryURL, if specified, overrides the URL used to fetch discovery information instead of using "{url}/.well-known/openid-configuration". The exact value specified is used, so "/.well-known/openid-configuration" must be included in discoveryURL if needed. The "issuer" field in the fetched discovery information must match the "issuer.url" field in the AuthenticationConfiguration and will be used to validate the "iss" claim in the presented JWT. This is for scenarios where the well-known and jwks endpoints are hosted at a different location than the issuer (such as locally in the cluster). Example: A discovery url that is exposed using kubernetes service 'oidc' in namespace 'oidc-namespace' and discovery information is available at '/.well-known/openid-configuration'. discoveryURL: "https://oidc.oidc-namespace/.well-known/openid-configuration" certificateAuthority is used to verify the TLS connection and the hostname on the leaf certificate must be set to 'oidc.oidc-namespace'. curl https://oidc.oidc-namespace/.well-known/openid-configuration (.discoveryURL field) { issuer: "https://oidc.example.com" (.url field) } discoveryURL must be different from url. Required to be unique across all JWT authenticators. Note that egress selection configuration is not used for this network connection.
`certificateAuthority` `string`	certificateAuthority contains PEM-encoded certificate authority certificates used to validate the connection when fetching discovery information. If unset, the system verifier is used. Same value as the content of the file referenced by the --oidc-ca-file flag.
`audiences` **[Required]** `[]string`	audiences is the set of acceptable audiences the JWT must be issued to. At least one of the entries must match the "aud" claim in presented JWTs. Same value as the --oidc-client-id flag (though this field supports an array). Required to be non-empty.

Field	Description
`audienceMatchPolicy` [AudienceMatchPolicyType]	audienceMatchPolicy defines how the "audiences" field is used to match the "aud" claim in the presented JWT. Allowed values are: 1. "MatchAny" when multiple audiences are specified and 2. empty (or unset) or "MatchAny" when a single audience is specified. - MatchAny: the "aud" claim in the presented JWT must match at least one of the entries in the "audiences" field. For example, if "audiences" is ["foo", "bar"], the "aud" claim in the presented JWT must contain either "foo" or "bar" (and may contain both). - "": The match policy can be empty (or unset) when a single audience is specified in the "audiences" field. The "aud" claim in the presented JWT must contain the single audience (and may contain others). For more nuanced audience validation, use claimValidationRules. example: claimValidationRule[].expression: 'sets.equivalent(claims.aud, ["bar", "foo", "baz"])' to require an exact match.

JWTAuthenticator

Appears in:

- AuthenticationConfiguration

JWTAuthenticator provides the configuration for a single JWT authenticator.

Field	Description
`issuer` **[Required]** Issuer	issuer contains the basic OIDC provider connection options.
`claimValidationRules` []ClaimValidationRule	claimValidationRules are rules that are applied to validate token claims to authenticate users.
`claimMappings` **[Required]** ClaimMappings	claimMappings points claims of a token to be treated as user attributes.
`userValidationRules` []UserValidationRule	userValidationRules are rules that are applied to final user before completing authentication. These allow invariants to be applied to incoming identities such as preventing the use of the system: prefix that is commonly used by Kubernetes components. The validation rules are logically ANDed together and must all return true for the validation to pass.

PrefixedClaimOrExpression

Appears in:

- ClaimMappings

PrefixedClaimOrExpression provides the configuration for a single prefixed claim or expression.

Field	Description
`claim` `string`	claim is the JWT claim to use. Mutually exclusive with expression.
`prefix` `string`	prefix is prepended to claim's value to prevent clashes with existing names. prefix needs to be set if claim is set and can be the empty string. Mutually exclusive with expression.
`expression` `string`	expression represents the expression which will be evaluated by CEL. CEL expressions have access to the contents of the token claims, organized into CEL variable: • 'claims' is a map of claim names to claim values. For example, a variable named 'sub' can be accessed as 'claims.sub'. Nested claims can be accessed using dot notation, e.g. 'claims.foo.bar'. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/ Mutually exclusive with claim and prefix.

ProtocolType

(Alias of `string`)

Appears in:

- Connection

ProtocolType is a set of valid values for Connection.ProtocolType

TCPTransport

Appears in:

- Transport

TCPTransport provides the information to connect to konnectivity server via TCP

Field	Description

Field	Description
`url` **[Required]** `string`	URL is the location of the konnectivity server to connect to. As an example it might be "https://127.0.0.1:8131"
`tlsConfig` [TLSConfig](#)	TLSConfig is the config needed to use TLS when connecting to konnectivity server

TLSConfig

Appears in:

- TCPTransport

TLSConfig provides the authentication information to connect to konnectivity server Only used with TCPTransport

Field	Description
`caBundle` `string`	caBundle is the file location of the CA to be used to determine trust with the konnectivity server. Must be absent/empty if TCPTransport.URL is prefixed with http:// If absent while TCPTransport.URL is prefixed with https://, default to system trust roots.
`clientKey` `string`	clientKey is the file location of the client key to be used in mtls handshakes with the konnectivity server. Must be absent/empty if TCPTransport.URL is prefixed with http:// Must be configured if TCPTransport.URL is prefixed with https://
`clientCert` `string`	clientCert is the file location of the client certificate to be used in mtls handshakes with the konnectivity server. Must be absent/empty if TCPTransport.URL is prefixed with http:// Must be configured if TCPTransport.URL is prefixed with https://

Transport

Appears in:

- Connection

Transport defines the transport configurations we use to dial to the konnectivity server

Field	Description
`tcp` [TCPTransport](#)	TCP is the TCP configuration for communicating with the konnectivity server via TCP ProxyProtocol of GRPC is not supported with TCP transport at the moment Requires at least one of TCP or UDS to be set

Field	Description
uds UDSTransport	UDS is the UDS configuration for communicating with the konnectivity server via UDS Requires at least one of TCP or UDS to be set

UDSTransport

Appears in:

- Transport

UDSTransport provides the information to connect to konnectivity server via UDS

Field	Description
udsName **[Required]** string	UDSName is the name of the unix domain socket to connect to konnectivity server This does not use a unix:// prefix. (Eg: /etc/srv/kubernetes/konnectivity-server/konnectivity-server.socket)

UserValidationRule

Appears in:

- JWTAuthenticator

UserValidationRule provides the configuration for a single user info validation rule.

Field	Description
expression **[Required]** string	expression represents the expression which will be evaluated by CEL. Must return true for the validation to pass. CEL expressions have access to the contents of UserInfo, organized into CEL variable: • 'user' - authentication.k8s.io/v1, Kind=UserInfo object Refer to https://github.com/kubernetes/api/blob/release-1.28/authentication/v1/types.go#L105-L122 for the definition. API documentation: https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.28/#userinfo-v1-authentication-k8s-io Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/
message string	message customizes the returned error message when rule returns false. message is a literal string.

WebhookConfiguration

Appears in:

- AuthorizerConfiguration

Field	Description
`authorizedTTL` **[Required]** meta/v1.Duration	The duration to cache 'authorized' responses from the webhook authoriz Same as setting `--authorization-webhook-cache-authorized-ttl` flag Default: 5m0s
`unauthorizedTTL` **[Required]** meta/v1.Duration	The duration to cache 'unauthorized' responses from the webhook authoriz Same as setting `--authorization-webhook-cache-unauthorized-ttl` fl Default: 30s
`timeout` **[Required]** meta/v1.Duration	Timeout for the webhook request Max allowed value is 30s. Required, no defa value.
`subjectAccessReviewVersion` **[Required]** `string`	The API version of the authorization.k8 SubjectAccessReview to send to and e from the webhook. Same as setting `--authorization-webhook-version` fla Valid values: v1beta1, v1 Required, no default value
`matchConditionSubjectAccessReviewVersion` **[Required]** `string`	MatchConditionSubjectAccessReviewV specifies the SubjectAccessReview vers the CEL expressions are evaluated aga Valid values: v1 Required, no default v
`failurePolicy` **[Required]** `string`	Controls the authorization decision wh webhook request fails to complete or returns a malformed response or erro evaluating matchConditions. Valid valu • NoOpinion: continue to subsequ authorizers to see if one of them allows the request • Deny: reject the request without consulting subsequent authorize Required, with no default.
`connectionInfo` **[Required]** WebhookConnectionInfo	ConnectionInfo defines how we talk to webhook

Field	Description
`matchConditions` **[Required]** []WebhookMatchCondition	matchConditions is a list of conditions must be met for a request to be sent t webhook. An empty list of matchCond matches all requests. There are a max of 64 match conditions allowed. The exact matching logic is (in order): 1. If at least one matchCondition evaluates to FALSE, then the web is skipped. 2. If ALL matchConditions evaluate TRUE, then the webhook is called 3. If at least one matchCondition evaluates to an error (but none a FALSE): ◦ If failurePolicy=Deny, then t webhook rejects the reques ◦ If failurePolicy=NoOpinion, the error is ignored and the webhook is skipped

WebhookConnectionInfo

Appears in:

- WebhookConfiguration

Field	Description
`type` **[Required]** `string`	Controls how the webhook should communicate with the server. Valid values: • KubeConfigFile: use the file specified in kubeConfigFile to locate the server. • InClusterConfig: use the in-cluster configuration to call the SubjectAccessReview API hosted by kube-apiserver. This mode is not allowed for kube-apiserver.

`kubeConfigFile`
[Required]
`string`
	Path to KubeConfigFile for connection info Required, if connectionInfo.Type is KubeConfig

WebhookMatchCondition

Appears in:

- WebhookConfiguration

Field	Description

Field	Description
`expression` **[Required]** `string`	expression represents the expression which will be evaluated by CEL. Must evaluate to bool. CEL expressions have access to the contents of the SubjectAccessReview in v1 version. If version specified by subjectAccessReviewVersion in the request variable is v1beta1, the contents would be converted to the v1 version before evaluating the CEL expression. Documentation on CEL: https://kubernetes.io/docs/reference/using-api/cel/

14.10 - kube-controller-manager Configuration (v1alpha1)

Resource Types

- CloudControllerManagerConfiguration
- LeaderMigrationConfiguration
- KubeControllerManagerConfiguration

NodeControllerConfiguration

Appears in:

- CloudControllerManagerConfiguration

NodeControllerConfiguration contains elements describing NodeController.

Field	Description
`ConcurrentNodeSyncs` **[Required]** `int32`	ConcurrentNodeSyncs is the number of workers concurrently synchronizing nodes

ServiceControllerConfiguration

Appears in:

- CloudControllerManagerConfiguration
- KubeControllerManagerConfiguration

ServiceControllerConfiguration contains elements describing ServiceController.

Field	Description
`ConcurrentServiceSyncs` **[Required]** `int32`	concurrentServiceSyncs is the number of services that are allowed to sync concurrently. Larger number = more responsive service management, but more CPU (and network) load.

CloudControllerManagerConfiguration

CloudControllerManagerConfiguration contains elements describing cloud-controller manager.

Field	Description

Field	Description
`apiVersion` string	`cloudcontrollermanager.config.k8s.i /v1alpha1`
`kind` string	`CloudControllerManagerConfiguratior`
`Generic` **[Required]** GenericControllerManagerConfiguration	Generic holds configuration for a generic controller-manager
`KubeCloudShared` **[Required]** KubeCloudSharedConfiguration	KubeCloudSharedConfiguration holds configuration for shared related features both in cloud controller manager and kub controller manager.
`NodeController` **[Required]** NodeControllerConfiguration	NodeController holds configuration for node controller related features.
`ServiceController` **[Required]** ServiceControllerConfiguration	ServiceControllerConfiguration holds configuration for ServiceController relatec features.
`NodeStatusUpdateFrequency` **[Required]** meta/v1.Duration	NodeStatusUpdateFrequency is the frequency at which the controller updates nodes' status
`Webhook` **[Required]** WebhookConfiguration	Webhook is the configuration for cloud-controller-manager hosted webhooks

CloudProviderConfiguration

Appears in:

- KubeCloudSharedConfiguration

CloudProviderConfiguration contains basically elements about cloud provider.

Field	Description
`Name` **[Required]** `string`	Name is the provider for cloud services.
`CloudConfigFile` **[Required]** `string`	cloudConfigFile is the path to the cloud provider configuration file.

KubeCloudSharedConfiguration

Appears in:

- CloudControllerManagerConfiguration

- KubeControllerManagerConfiguration

KubeCloudSharedConfiguration contains elements shared by both kube-controller manager and cloud-controller manager, but not genericconfig.

Field	Description
`CloudProvider` **[Required]** `CloudProviderConfiguration`	CloudProviderConfiguration holds configuration for CloudProvider related features.
`ExternalCloudVolumePlugin` **[Required]** `string`	externalCloudVolumePlugin specifies the plugin to use when cloudProvider is "external". It is currently used by the in repo cloud providers to handle node and volume control in the KCM.
`UseServiceAccountCredentials` **[Required]** `bool`	useServiceAccountCredentials indicates whether controllers should be run with individual service account credentials.
`AllowUntaggedCloud` **[Required]** `bool`	run with untagged cloud instances
`RouteReconciliationPeriod` **[Required]** `meta/v1.Duration`	routeReconciliationPeriod is the period for reconciling routes created for Nodes by cloud provider..
`NodeMonitorPeriod` **[Required]** `meta/v1.Duration`	nodeMonitorPeriod is the period for syncing NodeStatus in NodeController.
`ClusterName` **[Required]** `string`	clusterName is the instance prefix for the cluster.
`ClusterCIDR` **[Required]** `string`	clusterCIDR is CIDR Range for Pods in cluster.
`AllocateNodeCIDRs` **[Required]** `bool`	AllocateNodeCIDRs enables CIDRs for Pods to be allocated and, if ConfigureCloudRoutes is true, to be set on the cloud provider.
`CIDRAllocatorType` **[Required]** `string`	CIDRAllocatorType determines what kind of pod CIDR allocator will be used.
`ConfigureCloudRoutes` **[Required]** `bool`	configureCloudRoutes enables CIDRs allocated with allocateNodeCIDRs to be configured on the cloud provider.
`NodeSyncPeriod` **[Required]** `meta/v1.Duration`	nodeSyncPeriod is the period for syncing nodes from cloudprovider. Longer periods will result in fewer calls to cloud provider, but may delay addition of new nodes to cluster.

WebhookConfiguration

Appears in:

- CloudControllerManagerConfiguration

WebhookConfiguration contains configuration related to cloud-controller-manager hosted webhooks

Field	Description
`Webhooks` **[Required]** `[]string`	Webhooks is the list of webhooks to enable or disable '*' means "all enabled by default webhooks" 'foo' means "enable 'foo'" '-foo' means "disable 'foo'" first item for a particular name wins

LeaderMigrationConfiguration

Appears in:

- GenericControllerManagerConfiguration

LeaderMigrationConfiguration provides versioned configuration for all migrating leader locks.

Field	Description
`apiVersion` string	`controllermanager.config.k8s.io/v1alpha1`
`kind` string	`LeaderMigrationConfiguration`
`leaderName` **[Required]** `string`	LeaderName is the name of the leader election resource that protects the migration E.g. 1-20-KCM-to-1-21-CCM
`resourceLock` **[Required]** `string`	ResourceLock indicates the resource object type that will be used to lock Should be "leases" or "endpoints"
`controllerLeaders` **[Required]** `[]ControllerLeaderConfiguration`	ControllerLeaders contains a list of migrating leader lock configurations

ControllerLeaderConfiguration

Appears in:

- LeaderMigrationConfiguration

ControllerLeaderConfiguration provides the configuration for a migrating leader lock.

Field	Description

Field	Description
`name` **[Required]** `string`	Name is the name of the controller being migrated E.g. service-controller, route-controller, cloud-node-controller, etc
`component` **[Required]** `string`	Component is the name of the component in which the controller should be running. E.g. kube-controller-manager, cloud-controller-manager, etc Or '*' meaning the controller can be run under any component that participates in the migration

GenericControllerManagerConfiguration

Appears in:

- CloudControllerManagerConfiguration
- KubeControllerManagerConfiguration

GenericControllerManagerConfiguration holds configuration for a generic controller-manager.

Field	Description
`Port` **[Required]** `int32`	port is the port that the controller-manager's http service runs on.
`Address` **[Required]** `string`	address is the IP address to serve on (set to 0.0.0.0 for all interfaces).
`MinResyncPeriod` **[Required]** `meta/v1.Duration`	minResyncPeriod is the resync period in reflectors; will be random between minResyncPeriod and 2*minResyncPeriod.
`ClientConnection` **[Required]** `ClientConnectionConfiguration`	ClientConnection specifies the kubeconfig file and client connection settings for the proxy server to use when communicating with the apiserver.
`ControllerStartInterval` **[Required]** `meta/v1.Duration`	How long to wait between starting controller managers
`LeaderElection` **[Required]** `LeaderElectionConfiguration`	leaderElection defines the configuration of leader election client.
`Controllers` **[Required]** `[]string`	Controllers is the list of controllers to enable or disable '*' means "all enabled by default controllers" 'foo' means "enable 'foo'" '-foo' means "disable 'foo'" first item for a particular name wins

Field	Description
`Debugging` **[Required]** [DebuggingConfiguration](#)	DebuggingConfiguration holds configuration for Debugging related features.

`LeaderMigrationEnabled`
[Required]
`bool`
	LeaderMigrationEnabled indicates whether Leader Migration should be enabled for the controller manager.

`LeaderMigration` **[Required]** [LeaderMigrationConfiguration](#)	LeaderMigration holds the configuration for Leader Migration.

KubeControllerManagerConfiguration

KubeControllerManagerConfiguration contains elements describing kube-controller manager.

Field	Description
`apiVersion` string	`kubecontrollermana`
`kind` string	`KubeControllerMana`
`Generic` **[Required]** [GenericControllerManagerConfiguration](#)	Generic holds configu manager
`KubeCloudShared` **[Required]** [KubeCloudSharedConfiguration](#)	KubeCloudSharedCor shared related feature manager and kube-co
`AttachDetachController` **[Required]** [AttachDetachControllerConfiguration](#)	AttachDetachControll configuration for Atta features.
`CSRSigningController` **[Required]** [CSRSigningControllerConfiguration](#)	CSRSigningController for CSRSigningContro
`DaemonSetController` **[Required]** [DaemonSetControllerConfiguration](#)	DaemonSetController for DaemonSetContrc
`DeploymentController` **[Required]** [DeploymentControllerConfiguration](#)	DeploymentControlle for DeploymentContr
`StatefulSetController` **[Required]** [StatefulSetControllerConfiguration](#)	StatefulSetController for StatefulSetControl

Field	Description
`DeprecatedController` **[Required]** DeprecatedControllerConfiguration	DeprecatedController for some deprecated
`EndpointController` **[Required]** EndpointControllerConfiguration	EndpointControllerCo for EndpointControlle
`EndpointSliceController` **[Required]** EndpointSliceControllerConfiguration	EndpointSliceControll configuration for End features.
`EndpointSliceMirroringController` **[Required]** EndpointSliceMirroringControllerConfiguration	EndpointSliceMirrorin configuration for End related features.
`EphemeralVolumeController` **[Required]** EphemeralVolumeControllerConfiguration	EphemeralVolumeCor configuration for Eph features.
`GarbageCollectorController` **[Required]** GarbageCollectorControllerConfiguration	GarbageCollectorCon configuration for Garb features.
`HPAController` **[Required]** HPAControllerConfiguration	HPAControllerConfigu HPAController relatec
`JobController` **[Required]** JobControllerConfiguration	JobControllerConfigur JobController related
`CronJobController` **[Required]** CronJobControllerConfiguration	CronJobControllerCor CronJobController rel
`LegacySATokenCleaner` **[Required]** LegacySATokenCleanerConfiguration	LegacySATokenCleane configuration for Lega features.
`NamespaceController` **[Required]** NamespaceControllerConfiguration	NamespaceController for NamespaceContrc
`NodeIPAMController` **[Required]** NodeIPAMControllerConfiguration	NodeIPAMControllerC for NodeIPAMControl
`NodeLifecycleController` **[Required]** NodeLifecycleControllerConfiguration	NodeLifecycleControll configuration for Nod features.
`PersistentVolumeBinderController` **[Required]** PersistentVolumeBinderControllerConfiguration	PersistentVolumeBinc configuration for Pers related features.

Field	Description
`PodGCController` **[Required]** PodGCControllerConfiguration	PodGCControllerConf PodGCController relat
`ReplicaSetController` **[Required]** ReplicaSetControllerConfiguration	ReplicaSetControllerC for ReplicaSet related
`ReplicationController` **[Required]** ReplicationControllerConfiguration	ReplicationController(for ReplicationContro
`ResourceQuotaController` **[Required]** ResourceQuotaControllerConfiguration	ResourceQuotaContr configuration for Resc features.
`SAController` **[Required]** SAControllerConfiguration	SAControllerConfigur ServiceAccountContr
`ServiceController` **[Required]** ServiceControllerConfiguration	ServiceControllerCon ServiceController rela
`TTLAfterFinishedController` **[Required]** TTLAfterFinishedControllerConfiguration	TTLAfterFinishedCont configuration for TTLA features.
`ValidatingAdmissionPolicyStatusController` **[Required]** ValidatingAdmissionPolicyStatusControllerConfiguration	ValidatingAdmissionP holds configuration fc ValidatingAdmissionP features.

AttachDetachControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

AttachDetachControllerConfiguration contains elements describing AttachDetachController.

Field	Description

`DisableAttachDetachReconcilerSync`
[Required]
`bool`
	Reconciler runs a periodic loop to reconcile the desired state of the with the actual state of the world by triggering attach detach operations. This flag enables or disables reconcile. Is false by default, and thus enabled.

Field	Description
`ReconcilerSyncLoopPeriod` **[Required]** `meta/v1.Duration`	ReconcilerSyncLoopPeriod is the amount of time the reconciler sync states loop wait between successive executions. Is set to 60 sec by default.
`disableForceDetachOnTimeout` **[Required]** `bool`	DisableForceDetachOnTimeout disables force detach when the maximum unmount time is exceeded. Is false by default, and thus force detach on unmount is enabled.

CSRSigningConfiguration

Appears in:

- CSRSigningControllerConfiguration

CSRSigningConfiguration holds information about a particular CSR signer

Field	Description
`CertFile` **[Required]** `string`	certFile is the filename containing a PEM-encoded X509 CA certificate used to issue certificates
`KeyFile` **[Required]** `string`	keyFile is the filename containing a PEM-encoded RSA or ECDSA private key used to issue certificates

CSRSigningControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

CSRSigningControllerConfiguration contains elements describing CSRSigningController.

Field	Description
`ClusterSigningCertFile` **[Required]** `string`	clusterSigningCertFile is the filename containing a PEM-encoded X509 CA certificate used to issue cluster-scoped certificates
`ClusterSigningKeyFile` **[Required]** `string`	clusterSigningCertFile is the filename containing a PEM-encoded RSA or ECDSA private key used to issue cluster-scoped certificates
`KubeletServingSignerConfiguration` **[Required]** `CSRSigningConfiguration`	kubeletServingSignerConfiguration holds the certificate and key used to issue certificates for the kubernetes.io/kubelet-serving signer

Field	Description
`KubeletClientSignerConfiguration` **[Required]** [CSRSigningConfiguration](#)	kubeletClientSignerConfiguration holds the certificate and key used to issue certificates for the kubernetes.io/kube-apiserver-client-kubelet
`KubeAPIServerClientSignerConfiguration` **[Required]** [CSRSigningConfiguration](#)	kubeAPIServerClientSignerConfiguration holds the certificate and key used to issue certificates for the kubernetes.io/kube-apiserver-client
`LegacyUnknownSignerConfiguration` **[Required]** [CSRSigningConfiguration](#)	legacyUnknownSignerConfiguration holds the certificate and key used to issue certificates for the kubernetes.io/legacy-unknown
`ClusterSigningDuration` **[Required]** [meta/v1.Duration](#)	clusterSigningDuration is the max length of duration signed certificates will be given. Individual CSRs may request shorter certs by setting spec.expirationSeconds.

CronJobControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

CronJobControllerConfiguration contains elements describing CrongJob2Controller.

Field	Description
`ConcurrentCronJobSyncs` **[Required]** `int32`	concurrentCronJobSyncs is the number of job objects that are allowed to sync concurrently. Larger number = more responsive jobs, but more CPU (and network) load.

DaemonSetControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

DaemonSetControllerConfiguration contains elements describing DaemonSetController.

Field	Description
`ConcurrentDaemonSetSyncs` **[Required]** `int32`	concurrentDaemonSetSyncs is the number of daemonset objects that are allowed to sync concurrently. Larger number = more responsive daemonset, but more CPU (and network) load.

DeploymentControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

DeploymentControllerConfiguration contains elements describing DeploymentController.

Field	Description
`ConcurrentDeploymentSyncs` **[Required]** `int32`	concurrentDeploymentSyncs is the number of deployment objects that are allowed to sync concurrently. Larger number = more responsive deployments, but more CPU (and network) load.

DeprecatedControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

DeprecatedControllerConfiguration contains elements be deprecated.

EndpointControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

EndpointControllerConfiguration contains elements describing EndpointController.

Field	Description
`ConcurrentEndpointSyncs` **[Required]** `int32`	concurrentEndpointSyncs is the number of endpoint syncing operations that will be done concurrently. Larger number = faster endpoint updating, but more CPU (and network) load.
`EndpointUpdatesBatchPeriod` **[Required]** `meta/v1.Duration`	EndpointUpdatesBatchPeriod describes the length of endpoint updates batching period. Processing of pod changes will be delayed by this duration to join them with potential upcoming updates and reduce the overall number of endpoints updates.

EndpointSliceControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

EndpointSliceControllerConfiguration contains elements describing EndpointSliceController.

Field	Description
`ConcurrentServiceEndpointSyncs` **[Required]** `int32`	concurrentServiceEndpointSyncs is the number of service endpoint syncing operations that will be done concurrently. Larger number = faster endpoint slice updating, but more CPU (and network) load.
`MaxEndpointsPerSlice` **[Required]** `int32`	maxEndpointsPerSlice is the maximum number of endpoints that will be added to an EndpointSlice. More endpoints per slice will result in fewer and larger endpoint slices, but larger resources.
`EndpointUpdatesBatchPeriod` **[Required]** `meta/v1.Duration`	EndpointUpdatesBatchPeriod describes the length of endpoint updates batching period. Processing of pod changes will be delayed by this duration to join them with potential upcoming updates and reduce the overall number of endpoints updates.

`EndpointSliceMirroringControllerConfiguration`

Appears in:

- KubeControllerManagerConfiguration

EndpointSliceMirroringControllerConfiguration contains elements describing EndpointSliceMirroringController.

Field	Description
`MirroringConcurrentServiceEndpointSyncs` **[Required]** `int32`	mirroringConcurrentServiceEndpointSy is the number of service endpoint synci operations that will be done concurrent Larger number = faster endpoint slice updating, but more CPU (and network) load.
`MirroringMaxEndpointsPerSubset` **[Required]** `int32`	mirroringMaxEndpointsPerSubset is the maximum number of endpoints that wi be mirrored to an EndpointSlice for an EndpointSubset.

Field	Description
`MirroringEndpointUpdatesBatchPeriod` **[Required]** meta/v1.Duration	mirroringEndpointUpdatesBatchPeriod can be used to batch EndpointSlice updates. All updates triggered by EndpointSlice changes will be delayed b up to 'mirroringEndpointUpdatesBatchPeriod other addresses in the same Endpoints resource change in that period, they wil be batched to a single EndpointSlice update. Default 0 value means that eac Endpoints update triggers an EndpointSlice update.

EphemeralVolumeControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

EphemeralVolumeControllerConfiguration contains elements describing EphemeralVolumeController.

Field	Description

`ConcurrentEphemeralVolumeSyncs`
[Required]
`int32`
	ConcurrentEphemeralVolumeSyncseSyncs is the number of ephemeral volume syncing operations that will be done concurrently. Larger number = faster ephemeral volume updating, but more CPU (and network) load.

GarbageCollectorControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

GarbageCollectorControllerConfiguration contains elements describing GarbageCollectorController.

Field	Description

`EnableGarbageCollector`
[Required]
`bool`
	enables the generic garbage collector. MUST be synced with the corresponding flag of the kube-apiserver. WARNING: the generic garbage collector is an alpha feature.

`ConcurrentGCSyncs`
[Required]
`int32`
	concurrentGCSyncs is the number of garbage collector workers that are allowed to sync concurrently.

Field	Description
`GCIgnoredResources` **[Required]** []GroupResource	gcIgnoredResources is the list of GroupResources that garbage collection should ignore.

GroupResource

Appears in:

- GarbageCollectorControllerConfiguration

GroupResource describes an group resource.

Field	Description
`Group` **[Required]** `string`	group is the group portion of the GroupResource.
`Resource` **[Required]** `string`	resource is the resource portion of the GroupResource.

HPAControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

HPAControllerConfiguration contains elements describing HPAController.

Field	Description
`ConcurrentHorizontalPodAutoscalerSyncs` **[Required]** `int32`	ConcurrentHorizontalPoc number of HPA objects th concurrently. Larger num processing, but more CPU
`HorizontalPodAutoscalerSyncPeriod` **[Required]** meta/v1.Duration	HorizontalPodAutoscaler syncing the number of pc autoscaler.
`HorizontalPodAutoscalerDownscaleStabilizationWindow` **[Required]** meta/v1.Duration	HorizontalPodAutoscaler is a period for which auto and not scale down belov made during that period.
`HorizontalPodAutoscalerTolerance` **[Required]** `float64`	HorizontalPodAutoscaler when resource usage sug
`HorizontalPodAutoscalerCPUInitializationPeriod` **[Required]** meta/v1.Duration	HorizontalPodAutoscaler period after pod start wh skipped.

Field	Description
`HorizontalPodAutoscalerInitialReadinessDelay` **[Required]** `meta/v1.Duration`	HorizontalPodAutoscaler period after pod start dur are treated as readiness l The only effect of this is t samples from unready pc change during that perioc

JobControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

JobControllerConfiguration contains elements describing JobController.

Field	Description
`ConcurrentJobSyncs` **[Required]** `int32`	concurrentJobSyncs is the number of job objects that are allowed to sync concurrently. Larger number = more responsive jobs, but more CPU (and network) load.

LegacySATokenCleanerConfiguration

Appears in:

- KubeControllerManagerConfiguration

LegacySATokenCleanerConfiguration contains elements describing LegacySATokenCleaner

Field	Description
`CleanUpPeriod` **[Required]** `meta/v1.Duration`	CleanUpPeriod is the period of time since the last usage of an auto-generated service account token before it can be deleted.

NamespaceControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

NamespaceControllerConfiguration contains elements describing NamespaceController.

Field	Description
`NamespaceSyncPeriod` **[Required]** `meta/v1.Duration`	namespaceSyncPeriod is the period for syncing namespace life-cycle updates.

Field	Description
`ConcurrentNamespaceSyncs` **[Required]** `int32`	concurrentNamespaceSyncs is the number of namespace objects that are allowed to sync concurrently.

NodeIPAMControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

NodeIPAMControllerConfiguration contains elements describing NodeIpamController.

Field	Description
`ServiceCIDR` **[Required]** `string`	serviceCIDR is CIDR Range for Services in cluster.
`SecondaryServiceCIDR` **[Required]** `string`	secondaryServiceCIDR is CIDR Range for Services in cluster. This is used in dual stack clusters. SecondaryServiceCIDR must be of different IP family than ServiceCIDR
`NodeCIDRMaskSize` **[Required]** `int32`	NodeCIDRMaskSize is the mask size for node cidr in cluster.
`NodeCIDRMaskSizeIPv4` **[Required]** `int32`	NodeCIDRMaskSizeIPv4 is the mask size for node cidr in dual-stack cluster.
`NodeCIDRMaskSizeIPv6` **[Required]** `int32`	NodeCIDRMaskSizeIPv6 is the mask size for node cidr in dual-stack cluster.

NodeLifecycleControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

NodeLifecycleControllerConfiguration contains elements describing NodeLifecycleController.

Field	Description
`NodeEvictionRate` **[Required]** `float32`	nodeEvictionRate is the number of nodes per second on which pods are deleted in case of node failure when a zone is healthy

Field	Description
`SecondaryNodeEvictionRate` **[Required]** `float32`	secondaryNodeEvictionRate is the number of nodes per second on which pods are deleted in case of node failure when a zone is unhealthy
`NodeStartupGracePeriod` **[Required]** `meta/v1.Duration`	nodeStartupGracePeriod is the amount of time which we allow starting a node to be unresponsive before marking it unhealthy.
`NodeMonitorGracePeriod` **[Required]** `meta/v1.Duration`	nodeMontiorGracePeriod is the amount of time which we allow a running node to be unresponsive before marking it unhealthy. Must be N times more than kubelet's nodeStatusUpdateFrequency, where N means number of retries allowed for kubelet to post node status.
`PodEvictionTimeout` **[Required]** `meta/v1.Duration`	podEvictionTimeout is the grace period for deleting pods on failed nodes.
`LargeClusterSizeThreshold` **[Required]** `int32`	secondaryNodeEvictionRate is implicitly overridden to 0 for clusters smaller than or equal to largeClusterSizeThreshold
`UnhealthyZoneThreshold` **[Required]** `float32`	Zone is treated as unhealthy in nodeEvictionRate and secondaryNodeEvictionRate when at least unhealthyZoneThreshold (no less than 3) of Nodes in the zone are NotReady

PersistentVolumeBinderControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

PersistentVolumeBinderControllerConfiguration contains elements describing PersistentVolumeBinderController.

Field	Description
`PVClaimBinderSyncPeriod` **[Required]** `meta/v1.Duration`	pvClaimBinderSyncPeriod is the period for syncing persistent volumes and persistent volume claims.
`VolumeConfiguration` **[Required]** `VolumeConfiguration`	volumeConfiguration holds configuration for volume related features.

PersistentVolumeRecyclerConfiguration

Appears in:

- VolumeConfiguration

PersistentVolumeRecyclerConfiguration contains elements describing persistent volume plugins.

Field	Description
`MaximumRetry` **[Required]** `int32`	maximumRetry is number of retries the PV recycler will execute on failure to recycle PV.
`MinimumTimeoutNFS` **[Required]** `int32`	minimumTimeoutNFS is the minimum ActiveDeadlineSeconds to use for an NFS Recycler pod.
`PodTemplateFilePathNFS` **[Required]** `string`	podTemplateFilePathNFS is the file path to a pod definition used as a template for NFS persistent volume recycling
`IncrementTimeoutNFS` **[Required]** `int32`	incrementTimeoutNFS is the increment of time added per Gi to ActiveDeadlineSeconds for an NFS scrubber pod.
`PodTemplateFilePathHostPath` **[Required]** `string`	podTemplateFilePathHostPath is the file path to a pod definition used as a template for HostPath persistent volume recycling. This is for development and testing only and will not work in a multi-node cluster.
`MinimumTimeoutHostPath` **[Required]** `int32`	minimumTimeoutHostPath is the minimum ActiveDeadlineSeconds to use for a HostPath Recycler pod. This is for development and testing only and will not work in a multi-node cluster.
`IncrementTimeoutHostPath` **[Required]** `int32`	incrementTimeoutHostPath is the increment of time added per Gi to ActiveDeadlineSeconds for a HostPath scrubber pod. This is for development and testing only and will not work in a multi-node cluster.

PodGCControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

PodGCControllerConfiguration contains elements describing PodGCController.

Field	Description

Field	Description
`TerminatedPodGCThreshold` **[Required]** `int32`	terminatedPodGCThreshold is the number of terminated pods that can exist before the terminated pod garbage collector starts deleting terminated pods. If <= 0, the terminated pod garbage collector is disabled.

ReplicaSetControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

ReplicaSetControllerConfiguration contains elements describing ReplicaSetController.

Field	Description
`ConcurrentRSSyncs` **[Required]** `int32`	concurrentRSSyncs is the number of replica sets that are allowed to sync concurrently. Larger number = more responsive replica management, but more CPU (and network) load.

ReplicationControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

ReplicationControllerConfiguration contains elements describing ReplicationController.

Field	Description
`ConcurrentRCSyncs` **[Required]** `int32`	concurrentRCSyncs is the number of replication controllers that are allowed to sync concurrently. Larger number = more responsive replica management, but more CPU (and network) load.

ResourceQuotaControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

ResourceQuotaControllerConfiguration contains elements describing ResourceQuotaController.

Field	Description

Field	Description
`ResourceQuotaSyncPeriod` **[Required]** `meta/v1.Duration`	resourceQuotaSyncPeriod is the period for syncing quota usage status in the system.
`ConcurrentResourceQuotaSyncs` **[Required]** `int32`	concurrentResourceQuotaSyncs is the number of resource quotas that are allowed to sync concurrently. Larger number = more responsive quota management, but more CPU (and network) load.

SAControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

SAControllerConfiguration contains elements describing ServiceAccountController.

Field	Description
`ServiceAccountKeyFile` **[Required]** `string`	serviceAccountKeyFile is the filename containing a PEM-encoded private RSA key used to sign service account tokens.
`ConcurrentSATokenSyncs` **[Required]** `int32`	concurrentSATokenSyncs is the number of service account token syncing operations that will be done concurrently.
`RootCAFile` **[Required]** `string`	rootCAFile is the root certificate authority will be included in service account's token secret. This must be a valid PEM-encoded CA bundle.

StatefulSetControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

StatefulSetControllerConfiguration contains elements describing StatefulSetController.

Field	Description
`ConcurrentStatefulSetSyncs` **[Required]** `int32`	concurrentStatefulSetSyncs is the number of statefulset objects that are allowed to sync concurrently. Larger number = more responsive statefulsets, but more CPU (and network) load.

TTLAfterFinishedControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

TTLAfterFinishedControllerConfiguration contains elements describing TTLAfterFinishedController.

Field	Description
`ConcurrentTTLSyncs` **[Required]** `int32`	concurrentTTLSyncs is the number of TTL-after-finished collector workers that are allowed to sync concurrently.

ValidatingAdmissionPolicyStatusControllerConfiguration

Appears in:

- KubeControllerManagerConfiguration

ValidatingAdmissionPolicyStatusControllerConfiguration contains elements describing ValidatingAdmissionPolicyStatusController.

Field	Description
`ConcurrentPolicySyncs` **[Required]** `int32`	ConcurrentPolicySyncs is the number of policy objects that are allowed to sync concurrently. Larger number = quicker type checking, but more CPU (and network) load. The default value is 5.

VolumeConfiguration

Appears in:

- PersistentVolumeBinderControllerConfiguration

VolumeConfiguration contains *all* enumerated flags meant to configure all volume plugins. From this config, the controller-manager binary will create many instances of volume.VolumeConfig, each containing only the configuration needed for that plugin which are then passed to the appropriate plugin. The ControllerManager binary is the only part of the code which knows what plugins are supported and which flags correspond to each plugin.

Field	Description

Field	Description
`EnableHostPathProvisioning` **[Required]** `bool`	enableHostPathProvisioning enables HostPath PV provisioning when running without a cloud provider. This allows testing and development of provisioning features. HostPath provisioning is not supported in any way, won't work in a multi-node cluster, and should not be used for anything other than testing or development.
`EnableDynamicProvisioning` **[Required]** `bool`	enableDynamicProvisioning enables the provisioning of volumes when running within an environment that supports dynamic provisioning. Defaults to true.
`PersistentVolumeRecyclerConfiguration` **[Required]** PersistentVolumeRecyclerConfiguration	persistentVolumeRecyclerConfiguration holds configuration for persistent volume plugins.
`FlexVolumePluginDir` **[Required]** `string`	volumePluginDir is the full path of the directory in which the flex volume plugin should search for additional third party volume plugins

14.11 - kube-proxy Configuration (v1alpha1)

Resource Types

- KubeProxyConfiguration

`ClientConnectionConfiguration`

Appears in:

- KubeProxyConfiguration
- KubeSchedulerConfiguration
- GenericControllerManagerConfiguration

ClientConnectionConfiguration contains details for constructing a client.

Field	Description
`kubeconfig` **[Required]** `string`	kubeconfig is the path to a KubeConfig file.
`acceptContentTypes` **[Required]** `string`	acceptContentTypes defines the Accept header sent by clients when connecting to a server, overriding the default value of 'application/json'. This field will control all connections to the server used by a particular client.
`contentType` **[Required]** `string`	contentType is the content type used when sending data to the server from this client.
`qps` **[Required]** `float32`	qps controls the number of queries per second allowed for this connection.
`burst` **[Required]** `int32`	burst allows extra queries to accumulate when a client is exceeding its rate.

`DebuggingConfiguration`

Appears in:

- KubeSchedulerConfiguration
- GenericControllerManagerConfiguration

DebuggingConfiguration holds configuration for Debugging related features.

Field	Description

Field	Description
`enableProfiling` **[Required]** `bool`	enableProfiling enables profiling via web interface host:port/debug/pprof/
`enableContentionProfiling` **[Required]** `bool`	enableContentionProfiling enables block profiling, if enableProfiling is true.

LeaderElectionConfiguration

Appears in:

- KubeSchedulerConfiguration
- GenericControllerManagerConfiguration

LeaderElectionConfiguration defines the configuration of leader election clients for components that can run with leader election enabled.

Field	Description
`leaderElect` **[Required]** `bool`	leaderElect enables a leader election client to gain leadership before executing the main loop. Enable this when running replicated components for high availability.
`leaseDuration` **[Required]** `meta/v1.Duration`	leaseDuration is the duration that non-leader candidates will wait after observing a leadership renewal until attempting to acquire leadership of a led but unrenewed leader slot. This is effectively the maximum duration that a leader can be stopped before it is replaced by another candidate. This is only applicable if leader election is enabled.
`renewDeadline` **[Required]** `meta/v1.Duration`	renewDeadline is the interval between attempts by the acting master to renew a leadership slot before it stops leading. This must be less than or equal to the lease duration. This is only applicable if leader election is enabled.
`retryPeriod` **[Required]** `meta/v1.Duration`	retryPeriod is the duration the clients should wait between attempting acquisition and renewal of a leadership. This is only applicable if leader election is enabled.
`resourceLock` **[Required]** `string`	resourceLock indicates the resource object type that will be used to lock during leader election cycles.
`resourceName` **[Required]** `string`	resourceName indicates the name of resource object that will be used to lock during leader election cycles.
`resourceNamespace` **[Required]** `string`	resourceName indicates the namespace of resource object that will be used to lock during leader election cycles.

KubeProxyConfiguration

KubeProxyConfiguration contains everything necessary to configure the Kubernetes proxy server.

Field	Description
`apiVersion` string	`kubeproxy.config.k8s.io/v1alpha1`
`kind` string	`KubeProxyConfiguration`
`featureGates` **[Required]** `map[string]bool`	featureGates is a map of feature names to bools that enable or disable alpha/ experimental features.
`clientConnection` **[Required]** `ClientConnectionConfiguration`	clientConnection specifies the kubeconfig file and client connection settings for the proxy server to use when communicating with the apiserver.
`logging` **[Required]** `LoggingConfiguration`	logging specifies the options of logging. Refer to Logs Options for more information.
`hostnameOverride` **[Required]** `string`	hostnameOverride, if non-empty, will be used as the name of the Node that kube-proxy is running on. If unset, the node name is assumed to be the same as the node's hostname.
`bindAddress` **[Required]** `string`	bindAddress can be used to override kube-proxy's idea of what its node's primary IP is. Note that the name is a historical artifact, and kube-proxy does not actually bind any sockets to this IP.
`healthzBindAddress` **[Required]** `string`	healthzBindAddress is the IP address and port for the health check server to serve on, defaulting to "0.0.0.0:10256" (if bindAddress is unset or IPv4), or "[::]:10256" (if bindAddress is IPv6).
`metricsBindAddress` **[Required]** `string`	metricsBindAddress is the IP address and port for the metrics server to serve on, defaulting to "127.0.0.1:10249" (if bindAddress is unset or IPv4), or "[::1]:10249" (if bindAddress is IPv6). (Set to "0.0.0.0:10249" / "[::]:10249" to bind on all interfaces.)
`bindAddressHardFail` **[Required]** `bool`	bindAddressHardFail, if true, tells kube-proxy to treat failure to bind to a port as fatal and exit

Field	Description
`enableProfiling` **[Required]** `bool`	enableProfiling enables profiling via web interface on /debug/pprof handler. Profiling handlers will be handled by metrics server.
`showHiddenMetricsForVersion` **[Required]** `string`	showHiddenMetricsForVersion is the version for which you want to show hidden metrics.
`mode` **[Required]** ProxyMode	mode specifies which proxy mode to use.
`iptables` **[Required]** KubeProxyIPTablesConfiguration	iptables contains iptables-related configuration options.
`ipvs` **[Required]** KubeProxyIPVSConfiguration	ipvs contains ipvs-related configuration options.
`nftables` **[Required]** KubeProxyNFTablesConfiguration	nftables contains nftables-related configuration options.
`winkernel` **[Required]** KubeProxyWinkernelConfiguration	winkernel contains winkernel-related configuration options.
`detectLocalMode` **[Required]** LocalMode	detectLocalMode determines mode to use for detecting local traffic, defaults to ClusterCIDR
`detectLocal` **[Required]** DetectLocalConfiguration	detectLocal contains optional configuration settings related to DetectLocalMode.
`clusterCIDR` **[Required]** `string`	clusterCIDR is the CIDR range of the pods in the cluster. (For dual-stack clusters, this can be a comma-separated dual-stack pair of CIDR ranges.). When DetectLocalMode is set to ClusterCIDR, kube-proxy will consider traffic to be local if its source IP is in this range. (Otherwise it is not used.)
`nodePortAddresses` **[Required]** `[]string`	nodePortAddresses is a list of CIDR ranges that contain valid node IPs, or alternatively, the single string 'primary'. If set to a list of CIDRs, connections to NodePort services will only be accepted on node IPs in one of the indicated ranges. If set to 'primary', NodePort services will only be accepted on the node's primary IPv4 and/or IPv6 address according to the Node object. If unset, NodePort connections will be accepted on all local IPs.
`oomScoreAdj` **[Required]** `int32`	oomScoreAdj is the oom-score-adj value for kube-proxy process. Values must be within the range [-1000, 1000]

Field	Description
`conntrack` **[Required]** [KubeProxyConntrackConfiguration](#)	conntrack contains conntrack-related configuration options.
`configSyncPeriod` **[Required]** [meta/v1.Duration](#)	configSyncPeriod is how often configuration from the apiserver is refreshed. Must be greater than 0.
`portRange` **[Required]** `string`	portRange was previously used to configure the userspace proxy, but is now unused.
`windowsRunAsService` **[Required]** `bool`	windowsRunAsService, if true, enables Windows service control manager API integration.

DetectLocalConfiguration

Appears in:

- KubeProxyConfiguration

DetectLocalConfiguration contains optional settings related to DetectLocalMode option

Field	Description

`bridgeInterface`
[Required]
`string`
	bridgeInterface is a bridge interface name. When DetectLocalMode is set to LocalModeBridgeInterface, kube-proxy will consider traffic to be local if it originates from this bridge.

`interfaceNamePrefix`
[Required]
`string`
	interfaceNamePrefix is an interface name prefix. When DetectLocalMode is set to LocalModeInterfaceNamePrefix, kube-proxy will consider traffic to be local if it originates from any interface whose name begins with this prefix.

KubeProxyConntrackConfiguration

Appears in:

- KubeProxyConfiguration

KubeProxyConntrackConfiguration contains conntrack settings for the Kubernetes proxy server.

Field	Description
`maxPerCore` **[Required]** `int32`	maxPerCore is the maximum number of NAT connections to track per CPU core (0 to leave the limit as-is and ignore min).

Field	Description
`min` **[Required]** `int32`	min is the minimum value of connect-tracking records to allocate, regardless of maxPerCore (set maxPerCore=0 to leave the limit as-is).
`tcpEstablishedTimeout` **[Required]** meta/v1.Duration	tcpEstablishedTimeout is how long an idle TCP connection will be kept open (e.g. '2s'). Must be greater than 0 to set.
`tcpCloseWaitTimeout` **[Required]** meta/v1.Duration	tcpCloseWaitTimeout is how long an idle conntrack entry in CLOSE_WAIT state will remain in the conntrack table. (e.g. '60s'). Must be greater than 0 to set.
`tcpBeLiberal` **[Required]** `bool`	tcpBeLiberal, if true, kube-proxy will configure conntrack to run in liberal mode for TCP connections and packets with out-of-window sequence numbers won't be marked INVALID.
`udpTimeout` **[Required]** meta/v1.Duration	udpTimeout is how long an idle UDP conntrack entry in UNREPLIED state will remain in the conntrack table (e.g. '30s'). Must be greater than 0 to set.
`udpStreamTimeout` **[Required]** meta/v1.Duration	udpStreamTimeout is how long an idle UDP conntrack entry in ASSURED state will remain in the conntrack table (e.g. '300s'). Must be greater than 0 to set.

KubeProxyIPTablesConfiguration

Appears in:

- KubeProxyConfiguration

KubeProxyIPTablesConfiguration contains iptables-related configuration details for the Kubernetes proxy server.

Field	Description
`masqueradeBit` **[Required]** `int32`	masqueradeBit is the bit of the iptables fwmark space to use for SNAT if using the iptables or ipvs proxy mode. Values must be within the range [0, 31].
`masqueradeAll` **[Required]** `bool`	masqueradeAll tells kube-proxy to SNAT all traffic sent to Service cluster IPs, when using the iptables or ipvs proxy mode. This may be required with some CNI plugins.
`localhostNodePorts` **[Required]** `bool`	localhostNodePorts, if false, tells kube-proxy to disable the legacy behavior of allowing NodePort services to be accessed via localhost. (Applies only to iptables mode and IPv4; localhost NodePorts are never allowed with other proxy modes or with IPv6.)

Field	Description
`syncPeriod` **[Required]** [meta/v1.Duration](#)	syncPeriod is an interval (e.g. '5s', '1m', '2h22m') indicating how frequently various re-synchronizing and cleanup operations are performed. Must be greater than 0.
`minSyncPeriod` **[Required]** [meta/v1.Duration](#)	minSyncPeriod is the minimum period between iptables rule resyncs (e.g. '5s', '1m', '2h22m'). A value of 0 means every Service or EndpointSlice change will result in an immediate iptables resync.

KubeProxyIPVSConfiguration

Appears in:

- KubeProxyConfiguration

KubeProxyIPVSConfiguration contains ipvs-related configuration details for the Kubernetes proxy server.

Field	Description
`syncPeriod` **[Required]** [meta/v1.Duration](#)	syncPeriod is an interval (e.g. '5s', '1m', '2h22m') indicating how frequently various re-synchronizing and cleanup operations are performed. Must be greater than 0.
`minSyncPeriod` **[Required]** [meta/v1.Duration](#)	minSyncPeriod is the minimum period between IPVS rule resyncs (e.g. '5s', '1m', '2h22m'). A value of 0 means every Service or EndpointSlice change will result in an immediate IPVS resync.
`scheduler` **[Required]** `string`	scheduler is the IPVS scheduler to use
`excludeCIDRs` **[Required]** `[]string`	excludeCIDRs is a list of CIDRs which the ipvs proxier should not touch when cleaning up ipvs services.
`strictARP` **[Required]** `bool`	strictARP configures arp_ignore and arp_announce to avoid answering ARP queries from kube-ipvs0 interface
`tcpTimeout` **[Required]** [meta/v1.Duration](#)	tcpTimeout is the timeout value used for idle IPVS TCP sessions. The default value is 0, which preserves the current timeout value on the system.
`tcpFinTimeout` **[Required]** [meta/v1.Duration](#)	tcpFinTimeout is the timeout value used for IPVS TCP sessions after receiving a FIN. The default value is 0, which preserves the current timeout value on the system.
`udpTimeout` **[Required]** [meta/v1.Duration](#)	udpTimeout is the timeout value used for IPVS UDP packets. The default value is 0, which preserves the current timeout value on the system.

KubeProxyNFTablesConfiguration

Appears in:

- KubeProxyConfiguration

KubeProxyNFTablesConfiguration contains nftables-related configuration details for the Kubernetes proxy server.

Field	Description
`masqueradeBit` **[Required]** `int32`	masqueradeBit is the bit of the iptables fwmark space to use for SNAT if using the nftables proxy mode. Values must be within the range [0, 31].
`masqueradeAll` **[Required]** `bool`	masqueradeAll tells kube-proxy to SNAT all traffic sent to Service cluster IPs, when using the nftables mode. This may be required with some CNI plugins.
`syncPeriod` **[Required]** `meta/v1.Duration`	syncPeriod is an interval (e.g. '5s', '1m', '2h22m') indicating how frequently various re-synchronizing and cleanup operations are performed. Must be greater than 0.
`minSyncPeriod` **[Required]** `meta/v1.Duration`	minSyncPeriod is the minimum period between iptables rule resyncs (e.g. '5s', '1m', '2h22m'). A value of 0 means every Service or EndpointSlice change will result in an immediate iptables resync.

KubeProxyWinkernelConfiguration

Appears in:

- KubeProxyConfiguration

KubeProxyWinkernelConfiguration contains Windows/HNS settings for the Kubernetes proxy server.

Field	Description
`networkName` **[Required]** `string`	networkName is the name of the network kube-proxy will use to create endpoints and policies
`sourceVip` **[Required]** `string`	sourceVip is the IP address of the source VIP endpoint used for NAT when loadbalancing
`enableDSR` **[Required]** `bool`	enableDSR tells kube-proxy whether HNS policies should be created with DSR
`rootHnsEndpointName` **[Required]** `string`	rootHnsEndpointName is the name of hnsendpoint that is attached to l2bridge for root network namespace
`forwardHealthCheckVip` **[Required]** `bool`	forwardHealthCheckVip forwards service VIP for health check port on Windows

LocalMode

(Alias of `string`)

Appears in:

- KubeProxyConfiguration

LocalMode represents modes to detect local traffic from the node

ProxyMode

(Alias of `string`)

Appears in:

- KubeProxyConfiguration

ProxyMode represents modes used by the Kubernetes proxy server.

Currently, two modes of proxy are available on Linux platforms: 'iptables' and 'ipvs'. One mode of proxy is available on Windows platforms: 'kernelspace'.

If the proxy mode is unspecified, the best-available proxy mode will be used (currently this is `iptables` on Linux and `kernelspace` on Windows). If the selected proxy mode cannot be used (due to lack of kernel support, missing userspace components, etc) then kube-proxy will exit with an error.

14.12 - kube-scheduler Configuration (v1)

Resource Types

- DefaultPreemptionArgs
- InterPodAffinityArgs
- KubeSchedulerConfiguration
- NodeAffinityArgs
- NodeResourcesBalancedAllocationArgs
- NodeResourcesFitArgs
- PodTopologySpreadArgs
- VolumeBindingArgs

ClientConnectionConfiguration

Appears in:

- KubeSchedulerConfiguration

ClientConnectionConfiguration contains details for constructing a client.

Field	Description
`kubeconfig` **[Required]** `string`	kubeconfig is the path to a KubeConfig file.
`acceptContentTypes` **[Required]** `string`	acceptContentTypes defines the Accept header sent by clients when connecting to a server, overriding the default value of 'application/json'. This field will control all connections to the server used by a particular client.
`contentType` **[Required]** `string`	contentType is the content type used when sending data to the server from this client.
`qps` **[Required]** `float32`	qps controls the number of queries per second allowed for this connection.
`burst` **[Required]** `int32`	burst allows extra queries to accumulate when a client is exceeding its rate.

DebuggingConfiguration

Appears in:

- KubeSchedulerConfiguration

DebuggingConfiguration holds configuration for Debugging related features.

Field	Description
`enableProfiling` **[Required]** `bool`	enableProfiling enables profiling via web interface host:port/debug/pprof/
`enableContentionProfiling` **[Required]** `bool`	enableContentionProfiling enables block profiling, if enableProfiling is true.

LeaderElectionConfiguration

Appears in:

- KubeSchedulerConfiguration

LeaderElectionConfiguration defines the configuration of leader election clients for components that can run with leader election enabled.

Field	Description
`leaderElect` **[Required]** `bool`	leaderElect enables a leader election client to gain leadership before executing the main loop. Enable this when running replicated components for high availability.
`leaseDuration` **[Required]** `meta/v1.Duration`	leaseDuration is the duration that non-leader candidates will wait after observing a leadership renewal until attempting to acquire leadership of a led but unrenewed leader slot. This is effectively the maximum duration that a leader can be stopped before it is replaced by another candidate. This is only applicable if leader election is enabled.
`renewDeadline` **[Required]** `meta/v1.Duration`	renewDeadline is the interval between attempts by the acting master to renew a leadership slot before it stops leading. This must be less than or equal to the lease duration. This is only applicable if leader election is enabled.
`retryPeriod` **[Required]** `meta/v1.Duration`	retryPeriod is the duration the clients should wait between attempting acquisition and renewal of a leadership. This is only applicable if leader election is enabled.
`resourceLock` **[Required]** `string`	resourceLock indicates the resource object type that will be used to lock during leader election cycles.
`resourceName` **[Required]** `string`	resourceName indicates the name of resource object that will be used to lock during leader election cycles.
`resourceNamespace` **[Required]** `string`	resourceName indicates the namespace of resource object that will be used to lock during leader election cycles.

DefaultPreemptionArgs

DefaultPreemptionArgs holds arguments used to configure the DefaultPreemption plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`DefaultPreemptionArgs`
`minCandidateNodesPercentage` **[Required]** `int32`	MinCandidateNodesPercentage is the minimum number of candidates to shortlist when dry running preemption as a percentage of number of nodes. Must be in the range [0, 100]. Defaults to 10% of the cluster size if unspecified.
`minCandidateNodesAbsolute` **[Required]** `int32`	MinCandidateNodesAbsolute is the absolute minimum number of candidates to shortlist. The likely number of candidates enumerated for dry running preemption is given by the formula: numCandidates = max(numNodes * minCandidateNodesPercentage, minCandidateNodesAbsolute) We say "likely" because there are other factors such as PDB violations that play a role in the number of candidates shortlisted. Must be at least 0 nodes. Defaults to 100 nodes if unspecified.

InterPodAffinityArgs

InterPodAffinityArgs holds arguments used to configure the InterPodAffinity plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`InterPodAffinityArgs`
`hardPodAffinityWeight` **[Required]** `int32`	HardPodAffinityWeight is the scoring weight for existing pods with a matching hard affinity to the incoming pod.
`ignorePreferredTermsOfExistingPods` **[Required]** `bool`	IgnorePreferredTermsOfExistingPods configures the scheduler to ignore existing pods' preferred affinity rules when scoring candidate nodes, unless the incoming pod has inter-pod affinities.

KubeSchedulerConfiguration

KubeSchedulerConfiguration configures a scheduler

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`KubeSchedulerConfiguration`
`parallelism` **[Required]** `int32`	Parallelism defines the amount of parallelism in algorithms for scheduling a Pods. Must be greater t 0. Defaults to 16
`leaderElection` **[Required]** LeaderElectionConfiguration	LeaderElection defines the configuration of leader election client.
`clientConnection` **[Required]** ClientConnectionConfiguration	ClientConnection specifies the kubeconfig file and c connection settings for the proxy server to use whe communicating with the apiserver.
`DebuggingConfiguration` **[Required]** DebuggingConfiguration	(Members of `DebuggingConfiguration` are embe into this type.) DebuggingConfiguration holds configuration for Debugging related features TODO: We might wanna make this a substruct like Debugging componentbaseconfigv1alpha1.DebuggingConfigur
`percentageOfNodesToScore` **[Required]** `int32`	PercentageOfNodesToScore is the percentage of all nodes that once found feasible for running a pod, t scheduler stops its search for more feasible nodes i the cluster. This helps improve scheduler's performance. Scheduler always tries to find at least "minFeasibleNodesToFind" feasible nodes no matte what the value of this flag is. Example: if the cluster is 500 nodes and the value of this flag is 30, then scheduler stops finding further feasible nodes once finds 150 feasible ones. When the value is 0, default percentage (5%--50% based on the size of the cluste the nodes will be scored. It is overridden by profile PercentageOfNodesToScore.
`podInitialBackoffSeconds` **[Required]** `int64`	PodInitialBackoffSeconds is the initial backoff for unschedulable pods. If specified, it must be greater 0. If this value is null, the default value (1s) will be u
`podMaxBackoffSeconds` **[Required]** `int64`	PodMaxBackoffSeconds is the max backoff for unschedulable pods. If specified, it must be greater podInitialBackoffSeconds. If this value is null, the de value (10s) will be used.
`profiles` **[Required]** []KubeSchedulerProfile	Profiles are scheduling profiles that kube-scheduler supports. Pods can choose to be scheduled under a particular profile by setting its associated scheduler name. Pods that don't specify any scheduler name a scheduled with the "default-scheduler" profile, if present here.

Field	Description
`extenders` **[Required]** []Extender	Extenders are the list of scheduler extenders, each holding the values of how to communicate with the extender. These extenders are shared by all schedu profiles.

`delayCacheUntilActive`
[Required]
`bool`
	DelayCacheUntilActive specifies when to start cachi this is true and leader election is enabled, the schec will wait to fill informer caches until it is the leader. Doing so will have slower failover with the benefit o lower memory overhead while waiting to become leader. Defaults to false.

NodeAffinityArgs

NodeAffinityArgs holds arguments to configure the NodeAffinity plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`NodeAffinityArgs`
`addedAffinity` core/v1.NodeAffinity	AddedAffinity is applied to all Pods additionally to the NodeAffinity specified in the PodSpec. That is, Nodes need to satisfy AddedAffinity AND .spec.NodeAffinity. AddedAffinity is empty by default (all Nodes match). When AddedAffinity is used, some Pods with affinity requirements that match a specific Node (such as Daemonset Pods) might remain unschedulable.

NodeResourcesBalancedAllocationArgs

NodeResourcesBalancedAllocationArgs holds arguments used to configure NodeResourcesBalancedAllocation plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`NodeResourcesBalancedAllocationArgs`
`resources` **[Required]** []ResourceSpec	Resources to be managed, the default is "cpu" and "memory" if not specified.

NodeResourcesFitArgs

NodeResourcesFitArgs holds arguments used to configure the NodeResourcesFit plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`NodeResourcesFitArgs`
`ignoredResources` **[Required]** `[]string`	IgnoredResources is the list of resources that NodeResources fit filter should ignore. This doesn't apply to scoring.
`ignoredResourceGroups` **[Required]** `[]string`	IgnoredResourceGroups defines the list of resource groups that NodeResources fit filter should ignore. e.g. if group is ["example.com"], it will ignore all resource names that begin with "example.com", such as "example.com/aaa" and "example.com/bbb". A resource group name can't contain '/'. This doesn't apply to scoring.
`scoringStrategy` **[Required]** ScoringStrategy	ScoringStrategy selects the node resource scoring strategy. The default strategy is LeastAllocated with an equal "cpu" and "memory" weight.

PodTopologySpreadArgs

PodTopologySpreadArgs holds arguments used to configure the PodTopologySpread plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`PodTopologySpreadArgs`
`defaultConstraints` []core/v1.TopologySpreadConstraint	DefaultConstraints defines topology spre constraints to be applied to Pods that do define any in `pod.spec.topologySpreadConstraints.defaultConstraints[*].labelSelects` must be empty, as they are deduced f the Pod's membership to Services, ReplicationControllers, ReplicaSets or StatefulSets. When not empty, .defaultingType must be "List".

Field	Description
`defaultingType` [PodTopologySpreadConstraintsDefaulting](#)	DefaultingType determines how .defaultConstraints are deduced. Can be one of "System" or "List". • "System": Use kubernetes defined constraints that spread Pods amon Nodes and Zones. • "List": Use constraints defined in .defaultConstraints. Defaults to "System".

VolumeBindingArgs

VolumeBindingArgs holds arguments used to configure the VolumeBinding plugin.

Field	Description
`apiVersion` string	`kubescheduler.config.k8s.io/v1`
`kind` string	`VolumeBindingArgs`

`bindTimeoutSeconds`
[Required]
`int64`
	BindTimeoutSeconds is the timeout in seconds in volume binding operation. Value must be non-negative integer. The value zero indicates no waiting. If this value is nil, the default value (600) will be used.

`shape` []UtilizationShapePoint	Shape specifies the points defining the score function shape, which is used to score nodes based on the utilization of statically provisioned PVs. The utilization is calculated by dividing the total requested storage of the pod by the total capacity of feasible PVs on each node. Each point contains utilization (ranges from 0 to 100) and its associated score (ranges from 0 to 10). You can turn the priority by specifying different scores for different utilization numbers. The default shape points are: 1. 0 for 0 utilization 2. 10 for 100 utilization All points must be sorted in increasing order by utilization.

Extender

Appears in:

- KubeSchedulerConfiguration

Extender holds the parameters used to communicate with the extender. If a verb is unspecified/empty, it is assumed that the extender chose not to provide that extension.

Field	Description
`urlPrefix` **[Required]** `string`	URLPrefix at which the extender is available
`filterVerb` **[Required]** `string`	Verb for the filter call, empty if not supported. This verb is appended to the URLPrefix when issuing the filter call to extender.
`preemptVerb` **[Required]** `string`	Verb for the preempt call, empty if not supported. This verb is appended to the URLPrefix when issuing the preempt call to extender.
`prioritizeVerb` **[Required]** `string`	Verb for the prioritize call, empty if not supported. This verb is appended to the URLPrefix when issuing the prioritize call to extender.
`weight` **[Required]** `int64`	The numeric multiplier for the node scores that the prioritize call generates. The weight should be a positive integer
`bindVerb` **[Required]** `string`	Verb for the bind call, empty if not supported. This verb is appended to the URLPrefix when issuing the bind call to extender. If this method is implemented by the extender, it is the extender's responsibility to bind the pod to apiserver. Only one extender can implement this function.
`enableHTTPS` **[Required]** `bool`	EnableHTTPS specifies whether https should be used to communicate with the extender
`tlsConfig` **[Required]** `ExtenderTLSConfig`	TLSConfig specifies the transport layer security config
`httpTimeout` **[Required]** `meta/v1.Duration`	HTTPTimeout specifies the timeout duration for a call to the extender. Filter timeout fails the scheduling of the pod. Prioritize timeout is ignored, k8s/other extenders priorities are used to select the node.

`nodeCacheCapable`
[Required]
`bool`
	NodeCacheCapable specifies that the extender is capable of caching node information, so the scheduler should only send minimal information about the eligible nodes assuming that the extender already cached full details of all nodes in the cluster

Field	Description
`managedResources` [`[]ExtenderManagedResource`](#)	ManagedResources is a list of extended resources that are managed by this extender. • A pod will be sent to the extender on the Filter, Prioritize and Bind (if the extender is the binder) phases iff the pod requests at least one of the extended resources in this list. If empty or unspecified, all pods will be sent to this extender. • If IgnoredByScheduler is set to true for a resource, kube-scheduler will skip checking the resource in predicates.
`ignorable` **[Required]** `bool`	Ignorable specifies if the extender is ignorable, i.e. scheduling should not fail when the extender returns an error or is not reachable.

`ExtenderManagedResource`

Appears in:

- Extender

ExtenderManagedResource describes the arguments of extended resources managed by an extender.

Field	Description
`name` **[Required]** `string`	Name is the extended resource name.
`ignoredByScheduler` **[Required]** `bool`	IgnoredByScheduler indicates whether kube-scheduler should ignore this resource when applying predicates.

`ExtenderTLSConfig`

Appears in:

- Extender

ExtenderTLSConfig contains settings to enable TLS with extender

Field	Description
`insecure` **[Required]** `bool`	Server should be accessed without verifying the TLS certificate. For testing only.
`serverName` **[Required]** `string`	ServerName is passed to the server for SNI and is used in the client to check server certificates against. If ServerName is empty, the hostname used to contact the server is used.

Field	Description
`certFile` **[Required]** `string`	Server requires TLS client certificate authentication
`keyFile` **[Required]** `string`	Server requires TLS client certificate authentication
`caFile` **[Required]** `string`	Trusted root certificates for server
`certData` **[Required]** `[]byte`	CertData holds PEM-encoded bytes (typically read from a client certificate file). CertData takes precedence over CertFile
`keyData` **[Required]** `[]byte`	KeyData holds PEM-encoded bytes (typically read from a client certificate key file). KeyData takes precedence over KeyFile
`caData` **[Required]** `[]byte`	CAData holds PEM-encoded bytes (typically read from a root certificates bundle). CAData takes precedence over CAFile

KubeSchedulerProfile

Appears in:

- KubeSchedulerConfiguration

KubeSchedulerProfile is a scheduling profile.

Field	Description
`schedulerName` **[Required]** `string`	SchedulerName is the name of the scheduler associated to this profile. If SchedulerName matches with the pod's "spec.schedulerName", then the pod is scheduled with this profile.
`percentageOfNodesToScore` **[Required]** `int32`	PercentageOfNodesToScore is the percentage of all nodes that once found feasible for running a pod, the scheduler stops its search for more feasible nodes in the cluster. This helps improve scheduler's performance. Scheduler always tries to find at least "minFeasibleNodesToFind" feasible nodes no matter what the value of this flag is. Example: if the cluster size is 500 nodes and the value of this flag is 30, then scheduler stops finding further feasible nodes once it finds 150 feasible ones. When the value is 0, default percentage (5%--50% based on the size of the cluster) of the nodes will be scored. It will override global PercentageOfNodesToScore. If it is empty, global PercentageOfNodesToScore will be used.

Field	Description
`plugins` **[Required]** [Plugins](#)	Plugins specify the set of plugins that should be enabled or disabled. Enabled plugins are the ones that should be enabled in addition to the default plugins. Disabled plugins are any of the default plugins that should be disabled. When no enabled or disabled plugin is specified for an extension point, default plugins for that extension point will be used if there is any. If a QueueSort plugin is specified, the same QueueSort Plugin and PluginConfig must be specified for all profiles.
`pluginConfig` **[Required]** []PluginConfig	PluginConfig is an optional set of custom plugin arguments for each plugin. Omitting config args for a plugin is equivalent to using the default config for that plugin.

Plugin

Appears in:

- PluginSet

Plugin specifies a plugin name and its weight when applicable. Weight is used only for Score plugins.

Field	Description
`name` **[Required]** `string`	Name defines the name of plugin
`weight` **[Required]** `int32`	Weight defines the weight of plugin, only used for Score plugins.

PluginConfig

Appears in:

- KubeSchedulerProfile

PluginConfig specifies arguments that should be passed to a plugin at the time of initialization. A plugin that is invoked at multiple extension points is initialized once. Args can have arbitrary structure. It is up to the plugin to process these Args.

Field	Description
`name` **[Required]** `string`	Name defines the name of plugin being configured
`args` **[Required]** k8s.io/apimachinery/pkg/runtime.RawExtension	Args defines the arguments passed to the plugins at the time of initialization. Args can have arbitrary structure.

PluginSet

Appears in:

- Plugins

PluginSet specifies enabled and disabled plugins for an extension point. If an array is empty, missing, or nil, default plugins at that extension point will be used.

Field	Description
`enabled` **[Required]** [][]Plugin	Enabled specifies plugins that should be enabled in addition to default plugins. If the default plugin is also configured in the scheduler config file, the weight of plugin will be overridden accordingly. These are called after default plugins and in the same order specified here.
`disabled` **[Required]** [][]Plugin	Disabled specifies default plugins that should be disabled. When all default plugins need to be disabled, an array containing only one "*" should be provided.

Plugins

Appears in:

- KubeSchedulerProfile

Plugins include multiple extension points. When specified, the list of plugins for a particular extension point are the only ones enabled. If an extension point is omitted from the config, then the default set of plugins is used for that extension point. Enabled plugins are called in the order specified here, after default plugins. If they need to be invoked before default plugins, default plugins must be disabled and re-enabled here in desired order.

Field	Description
`preEnqueue` **[Required]** PluginSet	PreEnqueue is a list of plugins that should be invoked before adding pods to the scheduling queue.
`queueSort` **[Required]** PluginSet	QueueSort is a list of plugins that should be invoked when sorting pods in the scheduling queue.
`preFilter` **[Required]** PluginSet	PreFilter is a list of plugins that should be invoked at "PreFilter" extension point of the scheduling framework.
`filter` **[Required]** PluginSet	Filter is a list of plugins that should be invoked when filtering out nodes that cannot run the Pod.
`postFilter` **[Required]** PluginSet	PostFilter is a list of plugins that are invoked after filtering phase, but only when no feasible nodes were found for the pod.

Field	Description
`preScore` **[Required]** [PluginSet](#)	PreScore is a list of plugins that are invoked before scoring.
`score` **[Required]** [PluginSet](#)	Score is a list of plugins that should be invoked when ranking nodes that have passed the filtering phase.
`reserve` **[Required]** [PluginSet](#)	Reserve is a list of plugins invoked when reserving/ unreserving resources after a node is assigned to run the pod.
`permit` **[Required]** [PluginSet](#)	Permit is a list of plugins that control binding of a Pod. These plugins can prevent or delay binding of a Pod.
`preBind` **[Required]** [PluginSet](#)	PreBind is a list of plugins that should be invoked before a pod is bound.
`bind` **[Required]** [PluginSet](#)	Bind is a list of plugins that should be invoked at "Bind" extension point of the scheduling framework. The scheduler call these plugins in order. Scheduler skips the rest of these plugins as soon as one returns success.
`postBind` **[Required]** [PluginSet](#)	PostBind is a list of plugins that should be invoked after a pod is successfully bound.
`multiPoint` **[Required]** [PluginSet](#)	MultiPoint is a simplified config section to enable plugins for all valid extension points. Plugins enabled through MultiPoint will automatically register for every individual extension point the plugin has implemented. Disabling a plugin through MultiPoint disables that behavior. The same is true for disabling "*" through MultiPoint (no default plugins will be automatically registered). Plugins can still be disabled through their individual extension points. In terms of precedence, plugin config follows this basic hierarchy 1. Specific extension points 2. Explicitly configured MultiPoint plugins 3. The set of default plugins, as MultiPoint plugins This implies that a higher precedence plugin will run first and overwrite any settings within MultiPoint. Explicitly user-configured plugins also take a higher precedence over default plugins. Within this hierarchy, an Enabled setting takes precedence over Disabled. For example, if a plugin is set in both `multiPoint.Enabled` and `multiPoint.Disabled`, the plugin will be enabled. Similarly, including `multiPoint.Disabled = '*'` and `multiPoint.Enabled = pluginA` will still register that specific plugin through MultiPoint. This follows the same behavior as all other extension point configurations.

PodTopologySpreadConstraintsDefault

ing

(Alias of `string`)

Appears in:

- PodTopologySpreadArgs

PodTopologySpreadConstraintsDefaulting defines how to set default constraints for the PodTopologySpread plugin.

RequestedToCapacityRatioParam

Appears in:

- ScoringStrategy

RequestedToCapacityRatioParam define RequestedToCapacityRatio parameters

Field	Description
`shape` **[Required]** `[]UtilizationShapePoint`	Shape is a list of points defining the scoring function shape.

ResourceSpec

Appears in:

- NodeResourcesBalancedAllocationArgs
- ScoringStrategy

ResourceSpec represents a single resource.

Field	Description
`name` **[Required]** `string`	Name of the resource.
`weight` **[Required]** `int64`	Weight of the resource.

ScoringStrategy

Appears in:

- NodeResourcesFitArgs

ScoringStrategy define ScoringStrategyType for node resource plugin

Field	Description
`type` **[Required]** `ScoringStrategyType`	Type selects which strategy to run.

Field	Description
`resources` **[Required]** [`[]ResourceSpec`](#)	Resources to consider when scoring. The default resource set includes "cpu" and "memory" with an equal weight. Allowed weights go from 1 to 100. Weight defaults to 1 if not specified or explicitly set to 0.
`requestedToCapacityRatio` **[Required]** [`RequestedToCapacityRatioParam`](#)	Arguments specific to RequestedToCapacityRatio strategy.

ScoringStrategyType

(Alias of `string`)

Appears in:

- ScoringStrategy

ScoringStrategyType the type of scoring strategy used in NodeResourcesFit plugin.

UtilizationShapePoint

Appears in:

- VolumeBindingArgs
- RequestedToCapacityRatioParam

UtilizationShapePoint represents single point of priority function shape.

Field	Description
`utilization` **[Required]** `int32`	Utilization (x axis). Valid values are 0 to 100. Fully utilized node maps to 100.
`score` **[Required]** `int32`	Score assigned to given utilization (y axis). Valid values are 0 to 10.

14.13 - kubeadm Configuration (v1beta3)

Overview

Package v1beta3 defines the v1beta3 version of the kubeadm configuration file format. This version improves on the v1beta2 format by fixing some minor issues and adding a few new fields.

A list of changes since v1beta2:

- The deprecated "ClusterConfiguration.useHyperKubeImage" field has been removed. Kubeadm no longer supports the hyperkube image.
- The "ClusterConfiguration.dns.type" field has been removed since CoreDNS is the only supported DNS server type by kubeadm.
- Include "datapolicy" tags on the fields that hold secrets. This would result in the field values to be omitted when API structures are printed with klog.
- Add "InitConfiguration.skipPhases", "JoinConfiguration.skipPhases" to allow skipping a list of phases during kubeadm init/join command execution.
- Add "InitConfiguration.nodeRegistration.imagePullPolicy" and "JoinConfiguration.nodeRegistration.imagePullPolicy" to allow specifying the images pull policy during kubeadm "init" and "join". The value must be one of "Always", "Never" or "IfNotPresent". "IfNotPresent" is the default, which has been the existing behavior prior to this addition.
- Add "InitConfiguration.patches.directory", "JoinConfiguration.patches.directory" to allow the user to configure a directory from which to take patches for components deployed by kubeadm.
- Move the BootstrapToken* API and related utilities out of the "kubeadm" API group to a new group "bootstraptoken". The kubeadm API version v1beta3 no longer contains the BootstrapToken* structures.

Migration from old kubeadm config versions

- kubeadm v1.15.x and newer can be used to migrate from v1beta1 to v1beta2.
- kubeadm v1.22.x and newer no longer support v1beta1 and older APIs, but can be used to migrate v1beta2 to v1beta3.
- kubeadm v1.27.x and newer no longer support v1beta2 and older APIs,

Basics

The preferred way to configure kubeadm is to pass an YAML configuration file with the `--config` option. Some of the configuration options defined in the kubeadm config file are also available as command line flags, but only the most common/simple use case are supported with this approach.

A kubeadm config file could contain multiple configuration types separated using three dashes (`---`).

kubeadm supports the following configuration types:

```
apiVersion: kubeadm.k8s.io/v1beta3
kind: InitConfiguration

apiVersion: kubeadm.k8s.io/v1beta3
kind: ClusterConfiguration

apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration

apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration

apiVersion: kubeadm.k8s.io/v1beta3
kind: JoinConfiguration
```

To print the defaults for "init" and "join" actions use the following commands:

```
kubeadm config print init-defaults
kubeadm config print join-defaults
```

The list of configuration types that must be included in a configuration file depends by the action you are performing (`init` or `join`) and by the configuration options you are going to use (defaults or advanced customization).

If some configuration types are not provided, or provided only partially, kubeadm will use default values; defaults provided by kubeadm includes also enforcing consistency of values across components when required (e.g. `--cluster-cidr` flag on controller manager and `clusterCIDR` on kube-proxy).

Users are always allowed to override default values, with the only exception of a small subset of setting with relevance for security (e.g. enforce authorization-mode Node and RBAC on api server).

If the user provides a configuration types that is not expected for the action you are performing, kubeadm will ignore those types and print a warning.

Kubeadm init configuration types

When executing kubeadm init with the `--config` option, the following configuration types could be used: InitConfiguration, ClusterConfiguration, KubeProxyConfiguration, KubeletConfiguration, but only one between InitConfiguration and ClusterConfiguration is mandatory.

```
apiVersion: kubeadm.k8s.io/v1beta3
kind: InitConfiguration
bootstrapTokens:
  ...
nodeRegistration:
  ...
```

The InitConfiguration type should be used to configure runtime settings, that in case of kubeadm init are the configuration of the bootstrap token and all the setting which are specific to the node where kubeadm is executed, including:

- NodeRegistration, that holds fields that relate to registering the new node to the cluster; use it to customize the node name, the CRI

socket to use or any other settings that should apply to this node only (e.g. the node ip).

- LocalAPIEndpoint, that represents the endpoint of the instance of the API server to be deployed on this node; use it e.g. to customize the API server advertise address.

```
apiVersion: kubeadm.k8s.io/v1beta3
kind: ClusterConfiguration
networking:
  ...
etcd:
  ...
apiServer:
  extraArgs:
    ...
  extraVolumes:
    ...
...
```

The ClusterConfiguration type should be used to configure cluster-wide settings, including settings for:

- `networking` that holds configuration for the networking topology of the cluster; use it e.g. to customize Pod subnet or services subnet.
- `etcd` : use it e.g. to customize the local etcd or to configure the API server for using an external etcd cluster.
- kube-apiserver, kube-scheduler, kube-controller-manager configurations; use it to customize control-plane components by adding customized setting or overriding kubeadm default settings.

```
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
  ...
```

The KubeProxyConfiguration type should be used to change the configuration passed to kube-proxy instances deployed in the cluster. If this object is not provided or provided only partially, kubeadm applies defaults.

See https://kubernetes.io/docs/reference/command-line-tools-reference/kube-proxy/ or https://pkg.go.dev/k8s.io/kube-proxy/config/v1alpha1#KubeProxyConfiguration for kube-proxy official documentation.

```
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
  ...
```

The KubeletConfiguration type should be used to change the configurations that will be passed to all kubelet instances deployed in the cluster. If this object is not provided or provided only partially, kubeadm applies defaults.

See https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/ or https://pkg.go.dev/k8s.io/kubelet/config/v1beta1#KubeletConfiguration for kubelet official documentation.

Here is a fully populated example of a single YAML file containing multiple configuration types to be used during a `kubeadm init` run.

```yaml
apiVersion: kubeadm.k8s.io/v1beta3
kind: InitConfiguration
bootstrapTokens:
  - token: "9a08jv.c0izixklcxtmnze7"
    description: "kubeadm bootstrap token"
    ttl: "24h"
  - token: "783bde.3f89s0fje9f38fhf"
    description: "another bootstrap token"
    usages:
      - authentication
      - signing
    groups:
      - system:bootstrappers:kubeadm:default-node-token
nodeRegistration:
  name: "ec2-10-100-0-1"
  criSocket: "/var/run/dockershim.sock"
  taints:
    - key: "kubeadmNode"
      value: "someValue"
      effect: "NoSchedule"
  kubeletExtraArgs:
    v: 4
  ignorePreflightErrors:
    - IsPrivilegedUser
  imagePullPolicy: "IfNotPresent"
localAPIEndpoint:
  advertiseAddress: "10.100.0.1"
  bindPort: 6443
certificateKey:
"e6a2eb8581237ab72a4f494f30285ec12a9694d750b9785706a83bfcbbbd2204"
skipPhases:
  - addon/kube-proxy
---
apiVersion: kubeadm.k8s.io/v1beta3
kind: ClusterConfiguration
etcd:
  # one of local or external
  local:
    imageRepository: "registry.k8s.io"
    imageTag: "3.2.24"
    dataDir: "/var/lib/etcd"
    extraArgs:
      listen-client-urls: "http://10.100.0.1:2379"
    serverCertSANs:
      -  "ec2-10-100-0-1.compute-1.amazonaws.com"
    peerCertSANs:
      - "10.100.0.1"
  # external:
    # endpoints:
    # - "10.100.0.1:2379"
    # - "10.100.0.2:2379"
    # caFile: "/etcd/kubernetes/pki/etcd/etcd-ca.crt"
    # certFile: "/etcd/kubernetes/pki/etcd/etcd.crt"
    # keyFile: "/etcd/kubernetes/pki/etcd/etcd.key"
networking:
  serviceSubnet: "10.96.0.0/16"
  podSubnet: "10.244.0.0/24"
  dnsDomain: "cluster.local"
kubernetesVersion: "v1.21.0"
controlPlaneEndpoint: "10.100.0.1:6443"
apiServer:
  extraArgs:
    authorization-mode: "Node,RBAC"
  extraVolumes:
    - name: "some-volume"
      hostPath: "/etc/some-path"
      mountPath: "/etc/some-pod-path"
      readOnly: false
      pathType: File
  certSANs:
```

```
    - "10.100.1.1"
    - "ec2-10-100-0-1.compute-1.amazonaws.com"
  timeoutForControlPlane: 4m0s
controllerManager:
  extraArgs:
    "node-cidr-mask-size": "20"
  extraVolumes:
    - name: "some-volume"
      hostPath: "/etc/some-path"
      mountPath: "/etc/some-pod-path"
      readOnly: false
      pathType: File
scheduler:
  extraArgs:
    bind-address: "10.100.0.1"
  extraVolumes:
    - name: "some-volume"
      hostPath: "/etc/some-path"
      mountPath: "/etc/some-pod-path"
      readOnly: false
      pathType: File
certificatesDir: "/etc/kubernetes/pki"
imageRepository: "registry.k8s.io"
clusterName: "example-cluster"
---
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
# kubelet specific options here
---
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
# kube-proxy specific options here
```

Kubeadm join configuration types

When executing `kubeadm join` with the `--config` option, the JoinConfiguration type should be provided.

```
apiVersion: kubeadm.k8s.io/v1beta3
kind: JoinConfiguration
  ...
```

The JoinConfiguration type should be used to configure runtime settings, that in case of `kubeadm join` are the discovery method used for accessing the cluster info and all the setting which are specific to the node where kubeadm is executed, including:

- `nodeRegistration`, that holds fields that relate to registering the new node to the cluster; use it to customize the node name, the CRI socket to use or any other settings that should apply to this node only (e.g. the node ip).
- `apiEndpoint`, that represents the endpoint of the instance of the API server to be eventually deployed on this node.

Resource Types

- ClusterConfiguration
- InitConfiguration
- JoinConfiguration

BootstrapToken

Appears in:

- InitConfiguration

BootstrapToken describes one bootstrap token, stored as a Secret in the cluster

Field	Description
`token` **[Required]** BootstrapTokenString	`token` is used for establishing bidirectional trust between nodes and control-planes. Used for joining nodes in the cluster.
`description` string	`description` sets a human-friendly message why this token exists and what it's used for, so other administrators can know its purpose.
`ttl` meta/v1.Duration	`ttl` defines the time to live for this token. Defaults to `24h`. `expires` and `ttl` are mutually exclusive.
`expires` meta/v1.Time	`expires` specifies the timestamp when this token expires. Defaults to being set dynamically at runtime based on the `ttl`. `expires` and `ttl` are mutually exclusive.
`usages` []string	`usages` describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.
`groups` []string	`groups` specifies the extra groups that this token will authenticate as when/if used for authentication

BootstrapTokenString

Appears in:

- BootstrapToken

BootstrapTokenString is a token of the format `abcdef.abcdef0123456789` that is used for both validation of the practically of the API server from a joining node's point of view and as an authentication method for the node in the bootstrap phase of "kubeadm join". This token is and should be short-lived.

Field	Description
- **[Required]** string	No description provided.
- **[Required]** string	No description provided.

ClusterConfiguration

ClusterConfiguration contains cluster-wide configuration for a kubeadm cluster.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta3`
`kind` string	`ClusterConfiguration`
`etcd` [Etcd](#)	`etcd` holds the configuration for etcd.
`networking` [Networking](#)	`networking` holds configuration for the networking topology of the cluster.
`kubernetesVersion` string	`kubernetesVersion` is the target version of the control plane.
`controlPlaneEndpoint` string	`controlPlaneEndpoint` sets a stable IP address or DNS name for the control plane. It can be a valid IP address or a RFC-1123 DNS subdomain, both with optional TCP port. In case the `controlPlaneEndpoint` is not specified, the `advertiseAddress + bindPort` are used; in case the `controlPlaneEndpoint` is specified but without a TCP port, the `bindPort` is used. Possible usages are: • In a cluster with more than one control plane instances, this field should be assigned the address of the external load balancer in front of the control plane instances. • In environments with enforced node recycling, the `controlPlaneEndpoint` could be used for assigning a stable DNS to the control plane.
`apiServer` [APIServer](#)	`apiServer` contains extra settings for the API server.
`controllerManager` [ControlPlaneComponent](#)	`controllerManager` contains extra settings for the controller manager.
`scheduler` [ControlPlaneComponent](#)	`scheduler` contains extra settings for the scheduler.
`dns` [DNS](#)	`dns` defines the options for the DNS add-on installed in the cluster.
`certificatesDir` string	`certificatesDir` specifies where to store or look for all required certificates.

Field	Description
`imageRepository` `string`	`imageRepository` sets the container registry to pull images from. If empty, `registry.k8s.io` will be used by default. In case of kubernetes version is a CI build (kubernetes version starts with `ci/`) `gcr.io/k8s-staging-ci-images` will be used as a default for control plane components and for kube-proxy, while `registry.k8s.io` will be used for all the other images.
`featureGates` `map[string]bool`	`featureGates` contains the feature gates enabled by the user.
`clusterName` `string`	The cluster name.

InitConfiguration

InitConfiguration contains a list of elements that is specific "kubeadm init"-only runtime information. `kubeadm init` -only information. These fields are solely used the first time `kubeadm init` runs. After that, the information in the fields IS NOT uploaded to the `kubeadm-config` ConfigMap that is used by `kubeadm upgrade` for instance. These fields must be omitempty.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta3`
`kind` string	`InitConfiguration`
`bootstrapTokens` []BootstrapToken	`bootstrapTokens` is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature
`nodeRegistration` NodeRegistrationOptions	`nodeRegistration` holds fields that relate to registering the new control-plane node to the cluster.
`localAPIEndpoint` APIEndpoint	`localAPIEndpoint` represents the endpoint of the API server instance that's deployed on this control plane node. In HA setups, this differs from `ClusterConfiguration.controlPlaneEndpoint` in the sense that `controlPlaneEndpoint` is the global endpoint for the cluster, which then load-balances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.

Field	Description
`certificateKey` `string`	`certificateKey` sets the key with which certificates and keys are encrypted prior to being uploaded in a Secret in the cluster during the `uploadcerts init` phase. The certificate key is a hex encoded string that is an AES key of size 32 bytes.
`skipPhases` `[]string`	`skipPhases` is a list of phases to skip during command execution. The list of phases can be obtained with the `kubeadm init --help` command. The flag "--skip-phases" takes precedence over this field.
`patches` [Patches]	`patches` contains options related to applying patches to components deployed by kubeadm during `kubeadm init`.

JoinConfiguration

JoinConfiguration contains elements describing a particular node.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta3`
`kind` string	`JoinConfiguration`
`nodeRegistration` [NodeRegistrationOptions]	`nodeRegistration` holds fields that relate to registering the new control-plane node to the cluster.
`caCertPath` `string`	`caCertPath` is the path to the SSL certificate authority used to secure comunications between a node and the control-plane. Defaults to "/etc/kubernetes/pki/ca.crt".
`discovery` **[Required]** [Discovery]	`discovery` specifies the options for the kubelet to use during the TLS bootstrap process.
`controlPlane` [JoinControlPlane]	`controlPlane` defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.
`skipPhases` `[]string`	`skipPhases` is a list of phases to skip during command execution. The list of phases can be obtained with the `kubeadm join --help` command. The flag `--skip-phases` takes precedence over this field.
`patches` [Patches]	`patches` contains options related to applying patches to components deployed by kubeadm during `kubeadm join`.

APIEndpoint

Appears in:

- InitConfiguration
- JoinControlPlane

APIEndpoint struct contains elements of API server instance deployed on a node.

Field	Description
`advertiseAddress` `string`	`advertiseAddress` sets the IP address for the API server to advertise.
`bindPort` `int32`	`bindPort` sets the secure port for the API Server to bind to. Defaults to 6443.

APIServer

Appears in:

- ClusterConfiguration

APIServer holds settings necessary for API server deployments in the cluster

Field	Description
`ControlPlaneComponent` **[Required]** `ControlPlaneComponent`	(Members of `ControlPlaneComponent` are embedded into this type.) No description provided.
`certSANs` `[]string`	`certSANs` sets extra Subject Alternative Names (SANs) for the API Server signing certificate.
`timeoutForControlPlane` `meta/v1.Duration`	`timeoutForControlPlane` controls the timeout that we wait for API server to appear.

BootstrapTokenDiscovery

Appears in:

- Discovery

BootstrapTokenDiscovery is used to set the options for bootstrap token based discovery.

Field	Description
`token` **[Required]** `string`	`token` is a token used to validate cluster information fetched from the control-plane.

Field	Description
`apiServerEndpoint` `string`	`apiServerEndpoint` is an IP or domain name to the API server from which information will be fetched.
`caCertHashes` `[]string`	`caCertHashes` specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as `<type>:<value>`, where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be calculated using, for example, OpenSSL.
`unsafeSkipCAVerification` `bool`	`unsafeSkipCAVerification` allows token-based discovery without CA verification via `caCertHashes`. This can weaken the security of kubeadm since other nodes can impersonate the control-plane.

ControlPlaneComponent

Appears in:

- ClusterConfiguration
- APIServer

ControlPlaneComponent holds settings common to control plane component of the cluster

Field	Description
`extraArgs` `map[string]string`	`extraArgs` is an extra set of flags to pass to the control plane component. A key in this map is the flag name as it appears on the command line except without leading dash(es).
`extraVolumes` `[]HostPathMount`	`extraVolumes` is an extra set of host volumes, mounted to the control plane component.

DNS

Appears in:

- ClusterConfiguration

DNS defines the DNS addon that should be used in the cluster

Field	Description

Field	Description
`ImageMeta` **[Required]** [ImageMeta]	(Members of `ImageMeta` are embedded into this type.) `imageMeta` allows to customize the image used for the DNS component.

Discovery

Appears in:

- JoinConfiguration

Discovery specifies the options for the kubelet to use during the TLS Bootstrap process.

Field	Description
`bootstrapToken` [BootstrapTokenDiscovery]	`bootstrapToken` is used to set the options for bootstrap token based discovery. `bootstrapToken` and `file` are mutually exclusive.
`file` [FileDiscovery]	`file` is used to specify a file or URL to a kubeconfig file from which to load cluster information. `bootstrapToken` and `file` are mutually exclusive.
`tlsBootstrapToken` `string`	`tlsBootstrapToken` is a token used for TLS bootstrapping. If `bootstrapToken` is set, this field is defaulted to `.bootstrapToken.token`, but can be overridden. If `file` is set, this field **must be set** in case the KubeConfigFile does not contain any other authentication information
`timeout` [meta/v1.Duration]	`timeout` modifies the discovery timeout.

Etcd

Appears in:

- ClusterConfiguration

Etcd contains elements describing Etcd configuration.

Field	Description
`local` [LocalEtcd]	`local` provides configuration knobs for configuring the local etcd instance. `local` and `external` are mutually exclusive.
`external` [ExternalEtcd]	`external` describes how to connect to an external etcd cluster. `local` and `external` are mutually exclusive.

ExternalEtcd

Appears in:

- Etcd

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

Field	Description
`endpoints` **[Required]** `[]string`	`endpoints` contains the list of etcd members.
`caFile` **[Required]** `string`	`caFile` is an SSL Certificate Authority (CA) file used to secure etcd communication. Required if using a TLS connection.
`certFile` **[Required]** `string`	`certFile` is an SSL certification file used to secure etcd communication. Required if using a TLS connection.
`keyFile` **[Required]** `string`	`keyFile` is an SSL key file used to secure etcd communication. Required if using a TLS connection.

FileDiscovery

Appears in:

- Discovery

FileDiscovery is used to specify a file or URL to a kubeconfig file from which to load cluster information.

Field	Description
`kubeConfigPath` **[Required]** `string`	`kubeConfigPath` is used to specify the actual file path or URL to the kubeconfig file from which to load cluster information.

HostPathMount

Appears in:

- ControlPlaneComponent

HostPathMount contains elements describing volumes that are mounted from the host.

Field	Description
`name` **[Required]** `string`	`name` is the name of the volume inside the Pod template.
`hostPath` **[Required]** `string`	`hostPath` is the path in the host that will be mounted inside the Pod.

Field	Description
`mountPath` **[Required]** `string`	`mountPath` is the path inside the Pod where `hostPath` will be mounted.
`readOnly` `bool`	`readOnly` controls write access to the volume.
`pathType` [`core/v1.HostPathType`](#)	`pathType` is the type of the `hostPath`.

ImageMeta

Appears in:

- DNS
- LocalEtcd

ImageMeta allows to customize the image used for components that are not originated from the Kubernetes/Kubernetes release process

Field	Description
`imageRepository` `string`	`imageRepository` sets the container registry to pull images from. If not set, the `imageRepository` defined in ClusterConfiguration will be used instead.
`imageTag` `string`	`imageTag` allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.

JoinControlPlane

Appears in:

- JoinConfiguration

JoinControlPlane contains elements describing an additional control plane instance to be deployed on the joining node.

Field	Description
`localAPIEndpoint` [`APIEndpoint`](#)	`localAPIEndpoint` represents the endpoint of the API server instance to be deployed on this node.
`certificateKey` `string`	`certificateKey` is the key that is used for decryption of certificates after they are downloaded from the secret upon joining a new control plane node. The corresponding encryption key is in the InitConfiguration. The certificate key is a hex encoded string that is an AES key of size 32 bytes.

LocalEtcd

Appears in:

- Etcd

LocalEtcd describes that kubeadm should run an etcd cluster locally.

Field	Description
`ImageMeta` **[Required]** ImageMeta	(Members of `ImageMeta` are embedded into this type.) ImageMeta allows to customize the container used for etcd.
`dataDir` **[Required]** `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. A key in this map is the flag name as it appears on the command line except without leading dash(es).
`serverCertSANs` `[]string`	`serverCertSANs` sets extra Subject Alternative Names (SANs) for the etcd server signing certificate.
`peerCertSANs` `[]string`	`peerCertSANs` sets extra Subject Alternative Names (SANs) for the etcd peer signing certificate.

Networking

Appears in:

- ClusterConfiguration

Networking contains elements describing cluster's networking configuration.

Field	Description
`serviceSubnet` `string`	`serviceSubnet` is the subnet used by Kubernetes Services. Defaults to "10.96.0.0/12".
`podSubnet` `string`	`podSubnet` is the subnet used by Pods.
`dnsDomain` `string`	`dnsDomain` is the DNS domain used by Kubernetes Services. Defaults to "cluster.local".

NodeRegistrationOptions

Appears in:

- InitConfiguration
- JoinConfiguration

NodeRegistrationOptions holds fields that relate to registering a new control-plane or node to the cluster, either via `kubeadm init` or `kubeadm join`.

Field	Description
`name` `string`	`name` is the `.metadata.name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the `CommonName` field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`criSocket` `string`	`criSocket` is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use.
`taints` **[Required]** `[]core/v1.Taint`	`taints` specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, it will be defaulted with a control-plane taint for control-plane nodes. If you don't want to taint your control-plane node, set this field to an empty list, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.
`kubeletExtraArgs` `map[string]string`	`kubeletExtraArgs` passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the `kubelet-config` ConfigMap. Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on. A key in this map is the flag name as it appears on the command line except without leading dash(es).
`ignorePreflightErrors` `[]string`	`ignorePreflightErrors` provides a list of pre-flight errors to be ignored when the current node is registered, e.g. `IsPrevilegedUser,Swap`. Value `all` ignores errors from all checks.
`imagePullPolicy` `core/v1.PullPolicy`	`imagePullPolicy` specifies the policy for image pulling during kubeadm "init" and "join" operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". If this field is not set, kubeadm will default it to "IfNotPresent", or pull the required images if not present on the host.

Patches

Appears in:

- InitConfiguration
- JoinConfiguration

Patches contains options related to applying patches to components

deployed by kubeadm.

Field	Description
`directory` `string`	`directory` is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd". "patchtype" can be one of "strategic" "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically.

14.14 - kubeadm Configuration (v1beta4)

Overview

Package v1beta4 defines the v1beta4 version of the kubeadm configuration file format. This version improves on the v1beta3 format by fixing some minor issues and adding a few new fields.

A list of changes since v1beta3:

- TODO https://github.com/kubernetes/kubeadm/issues/2890
- Support custom environment variables in control plane components under `ClusterConfiguration`. Use `apiServer.extraEnvs`, `controllerManager.extraEnvs`, `scheduler.extraEnvs`, `etcd.local.extraEnvs`.
- The `ResetConfiguration` API type is now supported in v1beta4. Users are able to reset a node by passing a `--config` file to `kubeadm reset`.
- Dry run mode is now configureable in InitConfiguration and JoinConfiguration.
- Replace the existing string/string extra argument maps with structured extra arguments that support duplicates. The change applies to `ClusterConfiguration - apiServer.extraArgs`, `controllerManager.extraArgs`, `scheduler.extraArgs`, `etcd.local.extraArgs`. Also to `nodeRegistration.kubeletExtraArgs`.
- Add `ClusterConfiguration.encryptionAlgorithm` that can be used to set the asymmetric encryption algorithm used for this cluster's keys and certificates. Can be one of `"RSA-2048"` (default), `"RSA-3072"`, `"RSA-4096"` or `"ECDSA-P256"`.
- Add `ClusterConfiguration.dns.disabled` and `ClusterConfiguration.proxy.disabled` that can be used to disable the CoreDNS and kube-proxy addons during cluster initialization. Skipping the related addons phases, during cluster creation will set the same fields to `false`.
- Add the `nodeRegistration.imagePullSerial` field in `InitConfiguration` and `JoinConfiguration`, which can be used to control if kubeadm pulls images serially or in parallel.
- The `UpgradeConfiguration` kubeadm API is now supported in v1beta4 when passing `--config` to `kubeadm upgrade` subcommands. Usage of component configuration for `kubelet` and `kube-proxy`, `InitConfiguration` and `ClusterConfiguration` is deprecated and will be ignored when passing `--config` to `upgrade` subcommands.
- Add a `Timeouts` structure to `InitConfiguration`, `JoinConfiguration`, `ResetConfiguration` and `UpgradeConfiguration` that can be used to configure various timeouts.
- Add a `certificateValidityPeriod` and `caCertificateValidityPeriod` fields to `ClusterConfiguration`. These fields can be used to control the validity period of certificates generated by kubeadm during sub-commands such as `init`, `join`, `upgrade` and `certs`. Default values continue to be 1 year for non-CA certificates and 10 years for CA certificates. Only non-CA certificates continue to be renewable by `kubeadm certs renew`.

Migration from old kubeadm config versions

- kubeadm v1.15.x and newer can be used to migrate from v1beta1 to v1beta2.
- kubeadm v1.22.x and newer no longer support v1beta1 and older APIs, but can be used to migrate v1beta2 to v1beta3.
- kubeadm v1.27.x and newer no longer support v1beta2 and older APIs.
- TODO: https://github.com/kubernetes/kubeadm/issues/2890 add version that can be used to convert to v1beta4

Basics

The preferred way to configure kubeadm is to pass an YAML configuration file with the `--config“ option. Some of the configuration options defined in the kubeadm config file are also available as command line flags, but only the most common/simple use case are supported with this approach.

A kubeadm config file could contain multiple configuration types separated using three dashes (`---`).

kubeadm supports the following configuration types:

```
apiVersion: kubeadm.k8s.io/v1beta4
kind: InitConfiguration
apiVersion: kubeadm.k8s.io/v1beta4
kind: ClusterConfiguration


apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration


apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration


apiVersion: kubeadm.k8s.io/v1beta4
kind: JoinConfiguration


apiVersion: kubeadm.k8s.io/v1beta4
kind: ResetConfiguration


 apiVersion: kubeadm.k8s.io/v1beta4
kind: UpgradeConfiguration
```

To print the defaults for "init" and "join" actions use the following commands:

```
kubeadm config print init-defaults
kubeadm config print join-defaults
kubeadm config print reset-defaults
kubeadm config print upgrade-defaults
```

The list of configuration types that must be included in a configuration file depends by the action you are performing (`init or  join`) and by the configuration options you are going to use (defaults or advanced customization).

If some configuration types are not provided, or provided only partially, kubeadm will use default values; defaults provided by kubeadm includes also enforcing consistency of values across components when required

(e.g. `--cluster-cidr` flag on controller manager and `clusterCIDR` on kube-proxy).

Users are always allowed to override default values, with the only exception of a small subset of setting with relevance for security (e.g. enforce authorization-mode Node and RBAC on api server).

If the user provides a configuration types that is not expected for the action you are performing, kubeadm will ignore those types and print a warning.

Kubeadm init configuration types

When executing kubeadm init with the `--config“ option, the following configuration types could be used: InitConfiguration, ClusterConfiguration, KubeProxyConfiguration, KubeletConfiguration, but only one between InitConfiguration and ClusterConfiguration is mandatory.

```
apiVersion: kubeadm.k8s.io/v1beta4
kind: InitConfiguration
bootstrapTokens:

        ...

nodeRegistration:

        ...
```

The InitConfiguration type should be used to configure runtime settings, that in case of kubeadm init are the configuration of the bootstrap token and all the setting which are specific to the node where kubeadm is executed, including:

- NodeRegistration, that holds fields that relate to registering the new node to the cluster; use it to customize the node name, the CRI socket to use or any other settings that should apply to this node only (e.g. the node ip).
- LocalAPIEndpoint, that represents the endpoint of the instance of the API server to be deployed on this node; use it e.g. to customize the API server advertise address.

```
apiVersion: kubeadm.k8s.io/v1beta4
kind: ClusterConfiguration
networking:

        ...

etcd:

        ...

apiServer:

        extraArgs:
          ...
        extraVolumes:
          ...

...
```

The ClusterConfiguration type should be used to configure cluster-wide

settings, including settings for:

- `networking` that holds configuration for the networking topology of the cluster; use it e.g. to customize Pod subnet or services subnet.
- `etcd` : use it e.g. to customize the local etcd or to configure the API server for using an external etcd cluster.
- kube-apiserver, kube-scheduler, kube-controller-manager configurations; use it to customize control-plane components by adding customized setting or overriding kubeadm default settings.

```
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration

        ...
```

The KubeProxyConfiguration type should be used to change the configuration passed to kube-proxy instances deployed in the cluster. If this object is not provided or provided only partially, kubeadm applies defaults.

See https://kubernetes.io/docs/reference/command-line-tools-reference/kube-proxy/ or https://pkg.go.dev/k8s.io/kube-proxy/config/v1alpha1#KubeProxyConfiguration for kube-proxy official documentation.

```
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration

        ...
```

The KubeletConfiguration type should be used to change the configurations that will be passed to all kubelet instances deployed in the cluster. If this object is not provided or provided only partially, kubeadm applies defaults.

See https://kubernetes.io/docs/reference/command-line-tools-reference/kubelet/ or https://pkg.go.dev/k8s.io/kubelet/config/v1beta1#KubeletConfiguration for kubelet official documentation.

Here is a fully populated example of a single YAML file containing multiple configuration types to be used during a `kubeadm init` run.

```yaml
apiVersion: kubeadm.k8s.io/v1beta4
kind: InitConfiguration
bootstrapTokens:
  - token: "9a08jv.c0izixklcxtmnze7"
    description: "kubeadm bootstrap token"
    ttl: "24h"
  - token: "783bde.3f89s0fje9f38fhf"
    description: "another bootstrap token"
    usages:
  - authentication
  - signing
    groups:
  - system:bootstrappers:kubeadm:default-node-token

nodeRegistration:

                name: "ec2-10-100-0-1"
                criSocket: "unix:///var/run/containerd/
containerd.sock"
                taints:
                  - key: "kubeadmNode"
                    value: "someValue"
                    effect: "NoSchedule"
                kubeletExtraArgs:
                  - name: v
              value: "5"
                ignorePreflightErrors:
                  - IsPrivilegedUser
                imagePullPolicy: "IfNotPresent"
          imagePullSerial: true

localAPIEndpoint:

        advertiseAddress: "10.100.0.1"
        bindPort: 6443

certificateKey:
"e6a2eb8581237ab72a4f494f30285ec12a9694d750b9785706a83bfcbbbd2204"
skipPhases:
  - preflight

timeouts:

        controlPlaneComponentHealthCheck: "60s"
        kubenetesAPICall: "40s"

---
apiVersion: kubeadm.k8s.io/v1beta4
kind: ClusterConfiguration
etcd:

                # one of local or external
                local:
                  imageRepository: "registry.k8s.io"
                  imageTag: "3.2.24"
                  dataDir: "/var/lib/etcd"
                  extraArgs:
                    - name: listen-client-urls
                value: http://10.100.0.1:2379
            extraEnvs:
              - name: SOME_VAR
                value: SOME_VALUE
                   serverCertSANs:
                     -  "ec2-10-100-0-1.compute-1.amazonaws.com"
                   peerCertSANs:
                     - "10.100.0.1"
                # external:
                   # endpoints:
                   # - "10.100.0.1:2379"
                   # - "10.100.0.2:2379"
```

```
                # caFile: "/etc/kubernetes/pki/etcd/etcd-
ca.crt"
                # certFile: "/etc/kubernetes/pki/etcd/etcd.crt"
                # keyFile: "/etc/kubernetes/pki/etcd/etcd.key"

networking:

        serviceSubnet: "10.96.0.0/16"
        podSubnet: "10.244.0.0/24"
        dnsDomain: "cluster.local"

kubernetesVersion: "v1.21.0"
controlPlaneEndpoint: "10.100.0.1:6443"
apiServer:

               extraArgs:
                 - name: authorization-mode
            value: "Node,RBAC"
         extraEnvs:
           - name: SOME_VAR
             value: SOME_VALUE
               extraVolumes:
                 - name: "some-volume"
                   hostPath: "/etc/some-path"
                   mountPath: "/etc/some-pod-path"
                   readOnly: false
                   pathType: File
               certSANs:
                 - "10.100.1.1"
                 - "ec2-10-100-0-1.compute-1.amazonaws.com"

controllerManager:

               extraArgs:
                 - name: node-cidr-mask-size
            value: "20"
               extraVolumes:
                 - name: "some-volume"
                   hostPath: "/etc/some-path"
                   mountPath: "/etc/some-pod-path"
                   readOnly: false
                   pathType: File

scheduler:

               extraArgs:
                 - name: address
            value: "10.100.0.1"
               extraVolumes:
                 - name: "some-volume"
                   hostPath: "/etc/some-path"
                   mountPath: "/etc/some-pod-path"
                   readOnly: false
                   pathType: File

certificatesDir: "/etc/kubernetes/pki"
imageRepository: "registry.k8s.io"
clusterName: "example-cluster"
encryptionAlgorithm: ECDSA-P256
dns:

        disabled: true  # disable CoreDNS

proxy:

        diabled: true   # disable kube-proxy

---
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
# kubelet specific options here
```

```
---
apiVersion: kubeproxy.config.k8s.io/v1alpha1
kind: KubeProxyConfiguration
# kube-proxy specific options here
```

Kubeadm join configuration types

When executing kubeadm join with the --config option, the JoinConfiguration type should be provided.

```
apiVersion: kubeadm.k8s.io/v1beta4
kind: JoinConfiguration
discovery:

        bootstrapToken:
          apiServerEndpoint: some-address:6443
          token: abcdef.0123456789abcdef
          unsafeSkipCAVerification: true
        tlsBootstrapToken: abcdef.0123456789abcdef
```

The JoinConfiguration type should be used to configure runtime settings, that in case of kubeadm join are the discovery method used for accessing the cluster info and all the setting which are specific to the node where kubeadm is executed, including:

- `nodeRegistration`, that holds fields that relate to registering the new node to the cluster; use it to customize the node name, the CRI socket to use or any other settings that should apply to this node only (e.g. the node ip).
- `apiEndpoint`, that represents the endpoint of the instance of the API server to be eventually deployed on this node.

Kubeadm reset configuration types

When executing `kubeadm reset` with the `--config` option, the `ResetConfiguration` type should be provided.

```
apiVersion: kubeadm.k8s.io/v1beta4
kind: ResetConfiguration
...
```

Kubeadm upgrade configuration types

When executing `kubeadm upgrade` with the `--config` option, the `UpgradeConfiguration` type should be provided.

```
apiVersion: kubeadm.k8s.io/v1beta4
kind: UpgradeConfiguration
apply:

        ...

diff:

        ...

node:

        ...

plan:

        ...
```

The `UpgradeConfiguration` structure includes a few substructures that only apply to different subcommands of `kubeadm upgrade`. For example, the `apply` substructure will be used with the `kubeadm upgrade apply` subcommand and all other substructures will be ignored in such a case.

Resource Types

- ClusterConfiguration
- InitConfiguration
- JoinConfiguration
- ResetConfiguration
- UpgradeConfiguration

`BootstrapToken`

Appears in:

- InitConfiguration
- InitConfiguration

BootstrapToken describes one bootstrap token, stored as a Secret in the cluster

Field	Description
`token` **[Required]** `BootstrapTokenString`	`token` is used for establishing bidirectional trust between nodes and control-planes. Used for joining nodes in the cluster.
`description` `string`	`description` sets a human-friendly message why this token exists and what it's used for, so other administrators can know its purpose.
`ttl` `meta/v1.Duration`	`ttl` defines the time to live for this token. Defaults to `24h`. `expires` and `ttl` are mutually exclusive.

Field	Description
`expires` [meta/v1.Time](#)	`expires` specifies the timestamp when this token expires. Defaults to being set dynamically at runtime based on the `ttl`. `expires` and `ttl` are mutually exclusive.
`usages` `[]string`	`usages` describes the ways in which this token can be used. Can by default be used for establishing bidirectional trust, but that can be changed here.
`groups` `[]string`	`groups` specifies the extra groups that this token will authenticate as when/if used for authentication

BootstrapTokenString

Appears in:

- BootstrapToken

BootstrapTokenString is a token of the format `abcdef.abcdef0123456789` that is used for both validation of the practically of the API server from a joining node's point of view and as an authentication method for the node in the bootstrap phase of "kubeadm join". This token is and should be short-lived.

Field	Description
- **[Required]** `string`	No description provided.
- **[Required]** `string`	No description provided.

ClusterConfiguration

ClusterConfiguration contains cluster-wide configuration for a kubeadm cluster.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta4`
`kind` string	`ClusterConfiguration`
`etcd` [Etcd](#)	`etcd` holds the configuration for etcd.
`networking` [Networking](#)	`networking` holds configuration for the networking topology of the cluster.

Field	Description
`kubernetesVersion` `string`	`kubernetesVersion` is the target version of the control plane.
`controlPlaneEndpoint` `string`	`controlPlaneEndpoint` sets a stable IP address or DNS name for the control plane; It can be a valid IP address or a RFC-1123 DNS subdomain, both with optional TCP port. In case the `controlPlaneEndpoint` is not specified, the `advertiseAddress + bindPort` are used; in case the `controlPlaneEndpoint` is specified but without a TCP port, the `bindPort` is used. Possible usages are: • In a cluster with more than one control plane instances, this field should be assigned the address of the external load balancer in front of the control plane instances. • In environments with enforced node recycling, the `controlPlaneEndpoint` could be used for assigning a stable DNS to the control plane.
`apiServer` `APIServer`	`apiServer` contains extra settings for the API server.
`controllerManager` `ControlPlaneComponent`	`controllerManager` contains extra settings for the controller manager.
`scheduler` `ControlPlaneComponent`	`scheduler` contains extra settings for the scheduler.
`dns` `DNS`	`dns` defines the options for the DNS add-on installed in the cluster.
`proxy` **[Required]** `Proxy`	`proxy` defines the options for the proxy add-on installed in the cluster.
`certificatesDir` `string`	`certificatesDir` specifies where to store or look for all required certificates.
`imageRepository` `string`	`imageRepository` sets the container registry to pull images from. If empty, `registry.k8s.io` will be used by default. In case of kubernetes version is a CI build (kubernetes version starts with `ci/`) `gcr.io/k8s-staging-ci-images` will be used as a default for control plane components and for kube-proxy, while `registry.k8s.io` will be used for all the other images.
`featureGates` `map[string]bool`	`featureGates` contains the feature gates enabled by the user.

Field	Description
`clusterName` `string`	The cluster name.
`encryptionAlgorithm` [EncryptionAlgorithmType](#)	`encryptionAlgorithm` holds the type of asymmetric encryption algorithm used for keys and certificates. Can be `"RSA"` (default algorithm, key size is 2048) or `"ECDSA"` (uses the P-256 elliptic curve).
`certificateValidityPeriod` [meta/v1.Duration](#)	`certificateValidityPeriod` specifies the validity period for a non-CA certificate generated by kubeadm. Default value: `8760h`` (365 days * 24 hours = 1 year)
`caCertificateValidityPeriod` [meta/v1.Duration](#)	`caCertificateValidityPeriod` specifies the validity period for a CA certificate generated by kubeadm. Default value: `87600h` (365 days * 24 hours * 10 = 10 years)

InitConfiguration

InitConfiguration contains a list of elements that is specific "kubeadm init"-only runtime information. `kubeadm init` -only information. These fields are solely used the first time `kubeadm init` runs. After that, the information in the fields IS NOT uploaded to the `kubeadm-config` ConfigMap that is used by `kubeadm upgrade` for instance. These fields must be omitempty.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta4`
`kind` string	`InitConfiguration`
`bootstrapTokens` []BootstrapToken	`bootstrapTokens` is respected at `kubeadm init` time and describes a set of Bootstrap Tokens to create. This information IS NOT uploaded to the kubeadm cluster configmap, partly because of its sensitive nature
`dryRun` **[Required]** `bool`	`dryRun` tells if the dry run mode is enabled, don't apply any change in dry run mode, just out put what would be done.
`nodeRegistration` NodeRegistrationOptions	`nodeRegistration` holds fields that relate to registering the new control-plane node to the cluster.

Field	Description
`localAPIEndpoint` [APIEndpoint](#)	`localAPIEndpoint` represents the endpoint of the API server instance that's deployed on this control plane node. In HA setups, this differs from `ClusterConfiguration.controlPlaneEndpoint` in the sense that `controlPlaneEndpoint` is the global endpoint for the cluster, which then loadbalances the requests to each individual API server. This configuration object lets you customize what IP/DNS name and port the local API server advertises it's accessible on. By default, kubeadm tries to auto-detect the IP of the default interface and use that, but in case that process fails you may set the desired value here.
`certificateKey` `string`	`certificateKey` sets the key with which certificates and keys are encrypted prior to being uploaded in a Secret in the cluster during the `uploadcerts init` phase. The certificate key is a hex encoded string that is an AES key of size 32 bytes.
`skipPhases` `[]string`	`skipPhases` is a list of phases to skip during command execution. The list of phases can be obtained with the `kubeadm init --help` command. The flag `--skip-phases` takes precedence over this field.
`patches` [Patches](#)	`patches` contains options related to applying patches to components deployed by kubeadm during `kubeadm init` .
`timeouts` [Timeouts](#)	`timeouts` holds various timeouts that apply to kubeadm commands.

JoinConfiguration

JoinConfiguration contains elements describing a particular node.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta4`
`kind` string	`JoinConfiguration`
`dryRun` `bool`	`dryRun` tells if the dry run mode is enabled, don't apply any change if it is set, just output what would be done.
`nodeRegistration` [NodeRegistrationOptions](#)	`nodeRegistration` holds fields that relate to registering the new control-plane node to the cluster

Field	Description
`caCertPath` `string`	`caCertPath` is the path to the SSL certificate authority used to secure comunications between node and control-plane. Defaults to "/etc/kubernetes/pki/ca.crt".
`discovery` **[Required]** [Discovery](#)	`discovery` specifies the options for the kubelet to use during the TLS bootstrap process.
`controlPlane` [JoinControlPlane](#)	`controlPlane` defines the additional control plane instance to be deployed on the joining node. If nil, no additional control plane instance will be deployed.
`skipPhases` `[]string`	`skipPhases` is a list of phases to skip during command execution. The list of phases can be obtained with the `kubeadm join --help` command. The flag `--skip-phases` takes precedence over this field.
`patches` [Patches](#)	`patches` contains options related to applying patches to components deployed by kubeadm during `kubeadm join` .
`timeouts` [Timeouts](#)	`timeouts` holds various timeouts that apply to kubeadm commands.

ResetConfiguration

ResetConfiguration contains a list of fields that are specifically `kubeadm reset` -only runtime information.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta4`
`kind` string	`ResetConfiguration`
`cleanupTmpDir` `bool`	`cleanupTmpDir` specifies whether the "/etc/kubernetes/tmp" directory should be cleaned during the reset process.
`certificatesDir` `string`	`certificatesDir` specifies the directory where the certificates are stored. If specified, it will be cleaned during the reset process.
`criSocket` `string`	`criSocket` is used to retrieve container runtime inforomation and used for the removal of the containers. If `criSocket` is not specified by flag or config file, kubeadm will try to detect one valid CRI socket instead.

Field	Description
`dryRun` `bool`	`dryRun` tells if the dry run mode is enabled, don't apply any change if it is set and just output what would be done.
`force` `bool`	The `force` flag instructs kubeadm to reset the node without prompting for confirmation.
`ignorePreflightErrors` `[]string`	`ignorePreflightErrors` provides a list of pre-flight errors to be ignored during the reset process, e.g. `IsPrivilegedUser,Swap`. Value `all` ignores errors from all checks.
`skipPhases` `[]string`	`skipPhases` is a list of phases to skip during command execution. The list of phases can be obtained with the `kubeadm reset phase --help` command.
`unmountFlags` `[]string`	`unmountFlags` is a list of `unmount2()` syscall flags that kubeadm can use when unmounting directories during "reset". This flag can be one of: `"MNT_FORCE"`, `"MNT_DETACH"`, `"MNT_EXPIRE"`, `"UMOUNT_NOFOLLOW"`. By default this list is empty.
`timeouts` `Timeouts`	Timeouts holds various timeouts that apply to kubeadm commands.

UpgradeConfiguration

UpgradeConfiguration contains a list of options that are specific to `kubeadm upgrade` subcommands.

Field	Description
`apiVersion` string	`kubeadm.k8s.io/v1beta4`
`kind` string	`UpgradeConfiguration`
`apply` `UpgradeApplyConfiguration`	`apply` holds a list of options that are specific to the `kubeadm upgrade apply` command.
`diff` `UpgradeDiffConfiguration`	`diff` holds a list of options that are specific to the `kubeadm upgrade diff` command.
`node` `UpgradeNodeConfiguration`	`node` holds a list of options that are specific to the `kubeadm upgrade node` command.
`plan` `UpgradePlanConfiguration`	`plan` holds a list of options that are specific to the `kubeadm upgrade plan` command.

Field	Description
`timeouts` [Timeouts](#)	`timeouts` holds various timeouts that apply to kubeadm commands.

APIEndpoint

Appears in:

- InitConfiguration
- JoinControlPlane

APIEndpoint struct contains elements of API server instance deployed on a node.

Field	Description
`advertiseAddress` `string`	`dvertiseAddress` sets the IP address for the API server to advertise.
`bindPort` `int32`	`bindPort` sets the secure port for the API Server to bind to. Defaults to 6443.

APIServer

Appears in:

- ClusterConfiguration

APIServer holds settings necessary for API server deployments in the cluster

Field	Description
`ControlPlaneComponent` **[Required]** [ControlPlaneComponent](#)	(Members of `ControlPlaneComponent` are embedded into this type.) No description provided.
`certSANs` `[]string`	`certSANs` sets extra Subject Alternative Names (SANs) for the API Server signing certificate.

Arg

Appears in:

- ControlPlaneComponent
- LocalEtcd
- NodeRegistrationOptions

Arg represents an argument with a name and a value.

Field	Description
`name` **[Required]** `string`	The name of the argument.
`value` **[Required]** `string`	The value of the argument.

BootstrapTokenDiscovery

Appears in:

- Discovery

BootstrapTokenDiscovery is used to set the options for bootstrap token based discovery.

Field	Description
`token` **[Required]** `string`	`token` is a token used to validate cluster information fetched from the control-plane.
`apiServerEndpoint` `string`	`apiServerEndpoint` is an IP or domain name to the API server from which information will be fetched.
`caCertHashes` `[]string`	`caCertHashes` specifies a set of public key pins to verify when token-based discovery is used. The root CA found during discovery must match one of these values. Specifying an empty set disables root CA pinning, which can be unsafe. Each hash is specified as `<type>:<value>`, where the only currently supported type is "sha256". This is a hex-encoded SHA-256 hash of the Subject Public Key Info (SPKI) object in DER-encoded ASN.1. These hashes can be // calculated using, for example, OpenSSL.
`unsafeSkipCAVerification` `bool`	`unsafeSkipCAVerification` allows token-based discovery without CA verification via `caCertHashes`. This can weaken the security of kubeadm since other nodes can impersonate the control-plane.

ControlPlaneComponent

Appears in:

- ClusterConfiguration
- APIServer

ControlPlaneComponent holds settings common to control plane component of the cluster

Field	Description

Field	Description
`extraArgs` [`[]Arg`]	`extraArgs` is an extra set of flags to pass to the control plane component. An argument name in this list is the flag name as it appears on the command line except without leading dash(es). Extra arguments will override existing default arguments. Duplicate extra arguments are allowed.
`extraVolumes` [`[]HostPathMount`]	`extraVolumes` is an extra set of host volumes, mounted to the control plane component.
`extraEnvs` [`[]EnvVar`]	`extraEnvs` is an extra set of environment variables to pass to the control plane component. Environment variables passed using `extraEnvs` will override any existing environment variables, or `*_proxy` environment variables that kubeadm adds by default.

DNS

Appears in:

- ClusterConfiguration

DNS defines the DNS addon that should be used in the cluster

Field	Description
`ImageMeta` **[Required]** `ImageMeta`	(Members of `ImageMeta` are embedded into this type.) `imageMeta` allows to customize the image used for the DNS addon.
`disabled` **[Required]** `bool`	`disabled` specifies whether to disable this addon in the cluster.

Discovery

Appears in:

- JoinConfiguration

Discovery specifies the options for the kubelet to use during the TLS Bootstrap process

Field	Description
`bootstrapToken` `BootstrapTokenDiscovery`	`bootstrapToken` is used to set the options for bootstrap token based discovery. `bootstrapToken` and `file` are mutually exclusive.
`file` `FileDiscovery`	`file` is used to specify a file or URL to a kubeconfig file from which to load cluster information. `bootstrapToken` and `file` are mutually exclusive.

Field	Description
`tlsBootstrapToken` `string`	`tlsBootstrapToken` is a token used for TLS bootstrapping. If `bootstrapToken` is set, this field is defaulted to `bootstrapToken.token`, but can be overridden. If `file` is set, this field **must be set** in case the KubeConfigFile does not contain any other authentication information.

EncryptionAlgorithmType

(Alias of `string`)

Appears in:

- ClusterConfiguration

EncryptionAlgorithmType can define an asymmetric encryption algorithm type.

EnvVar

Appears in:

- ControlPlaneComponent
- LocalEtcd

EnvVar represents an environment variable present in a Container.

Field	Description
`EnvVar` **[Required]** `core/v1.EnvVar`	(Members of `EnvVar` are embedded into this type.) No description provided.

Etcd

Appears in:

- ClusterConfiguration

Etcd contains elements describing Etcd configuration.

Field	Description
`local` `LocalEtcd`	`local` provides configuration knobs for configuring the local etcd instance. `local` and `external` are mutually exclusive.
`external` `ExternalEtcd`	`external` describes how to connect to an external etcd cluster. `local` and `external` are mutually exclusive.

ExternalEtcd

Appears in:

- Etcd

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

Field	Description
`endpoints` **[Required]** `[]string`	`endpoints` contains the list of etcd members.
`caFile` **[Required]** `string`	`caFile` is an SSL Certificate Authority (CA) file used to secure etcd communication. Required if using a TLS connection.
`certFile` **[Required]** `string`	`certFile` is an SSL certification file used to secure etcd communication. Required if using a TLS connection.
`keyFile` **[Required]** `string`	`keyFile` is an SSL key file used to secure etcd communication. Required if using a TLS connection.

FileDiscovery

Appears in:

- Discovery

FileDiscovery is used to specify a file or URL to a kubeconfig file from which to load cluster information.

Field	Description
`kubeConfigPath` **[Required]** `string`	`kubeConfigPath` is used to specify the actual file path or URL to the kubeconfig file from which to load cluster information.

HostPathMount

Appears in:

- ControlPlaneComponent

HostPathMount contains elements describing volumes that are mounted from the host.

Field	Description
`name` **[Required]** `string`	`name` is the name of the volume inside the Pod template.
`hostPath` **[Required]** `string`	`hostPath` is the path in the host that will be mounted inside the Pod.

Field	Description
`mountPath` **[Required]** `string`	`mountPath` is the path inside the Pod where `hostPath` will be mounted.
`readOnly` `bool`	`readOnly` controls write access to the volume.
`pathType` [`core/v1.HostPathType`](#)	`pathType` is the type of the `hostPath`.

ImageMeta

Appears in:

- DNS
- LocalEtcd

ImageMeta allows to customize the image used for components that are not originated from the Kubernetes/Kubernetes release process

Field	Description
`imageRepository` `string`	`imageRepository` sets the container registry to pull images from. if not set, the `imageRepository` defined in ClusterConfiguration will be used instead.
`imageTag` `string`	`imageTag` allows to specify a tag for the image. In case this value is set, kubeadm does not change automatically the version of the above components during upgrades.

JoinControlPlane

Appears in:

- JoinConfiguration

JoinControlPlane contains elements describing an additional control plane instance to be deployed on the joining node.

Field	Description
`localAPIEndpoint` [`APIEndpoint`](#)	`localAPIEndpoint` represents the endpoint of the API server instance to be deployed on this node.
`certificateKey` `string`	`certificateKey` is the key that is used for decryption of certificates after they are downloaded from the Secret upon joining a new control plane node. The corresponding encryption key is in the InitConfiguration. The certificate key is a hex encoded string that is an AES key of size 32 bytes.

LocalEtcd

Appears in:

- Etcd

LocalEtcd describes that kubeadm should run an etcd cluster locally.

Field	Description
`ImageMeta` **[Required]** ImageMeta	(Members of `ImageMeta` are embedded into this type.) ImageMeta allows to customize the container used for etcd
`dataDir` **[Required]** `string`	`dataDir` is the directory etcd will place its data. Defaults to "/var/lib/etcd".
`extraArgs` **[Required]** []Arg	`extraArgs` are extra arguments provided to the etcd binary when run inside a static Pod. An argument name in this list is the flag name as it appears on the command line except without leading dash(es). Extra arguments will override existing default arguments. Duplicate extra arguments are allowed.
`extraEnvs` []EnvVar	`extraEnvs` is an extra set of environment variables to pass to the control plane component. Environment variables passed using `extraEnvs` will override any existing environment variables, or `*_proxy` environment variables that kubeadm adds by default.
`serverCertSANs` `[]string`	`serverCertSANs` sets extra Subject Alternative Names (SANs) for the etcd server signing certificate.
`peerCertSANs` `[]string`	`peerCertSANs` sets extra Subject Alternative Names (SANs) for the etcd peer signing certificate.

Networking

Appears in:

- ClusterConfiguration

Networking contains elements describing cluster's networking configuration.

Field	Description
`serviceSubnet` `string`	`serviceSubnet` is the subnet used by Kubernetes Services. Defaults to "10.96.0.0/12".
`podSubnet` `string`	`podSubnet` is the subnet used by Pods.
`dnsDomain` `string`	`dnsDomain` is the dns domain used by Kubernetes Services. Defaults to "cluster.local".

NodeRegistrationOptions

Appears in:

- InitConfiguration
- JoinConfiguration

NodeRegistrationOptions holds fields that relate to registering a new control-plane or node to the cluster, either via `kubeadm init` or `kubeadm join`.

Field	Description
`name` `string`	`name` is the `.Metadata.Name` field of the Node API object that will be created in this `kubeadm init` or `kubeadm join` operation. This field is also used in the `CommonName` field of the kubelet's client certificate to the API server. Defaults to the hostname of the node if not provided.
`criSocket` `string`	`criSocket` is used to retrieve container runtime info. This information will be annotated to the Node API object, for later re-use.
`taints` **[Required]** `[]core/v1.Taint`	`taints` specifies the taints the Node API object should be registered with. If this field is unset, i.e. nil, it will be defaulted with a control-plane taint for control-plane nodes. If you don't want to taint your control-plane node, set this field to an empty list, i.e. `taints: []` in the YAML file. This field is solely used for Node registration.
`kubeletExtraArgs` `[]Arg`	`kubeletExtraArgs` passes through extra arguments to the kubelet. The arguments here are passed to the kubelet command line via the environment file kubeadm writes at runtime for the kubelet to source. This overrides the generic base-level configuration in the `kubelet-config` ConfigMap. Flags have higher priority when parsing. These values are local and specific to the node kubeadm is executing on. An argument name in this list is the flag name as it appears on the command line except without leading dash(es). Extra arguments will override existing default arguments. Duplicate extra arguments are allowed.
`ignorePreflightErrors` `[]string`	`ignorePreflightErrors` provides a slice of pre-flight errors to be ignored when the current node is registered, e.g. 'IsPrivilegedUser,Swap'. Value 'all' ignores errors from all checks.
`imagePullPolicy` `core/v1.PullPolicy`	`imagePullPolicy` specifies the policy for image pulling during kubeadm `init` and `join` operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". If this field is unset kubeadm will default it to "IfNotPresent", or pull the required images if not present on the host.

Field	Description
`imagePullSerial` `bool`	`imagePullSerial` specifies if image pulling performed by kubeadm must be done serially or in parallel. Default: true

Patches

Appears in:

- InitConfiguration
- JoinConfiguration
- UpgradeApplyConfiguration
- UpgradeNodeConfiguration

Patches contains options related to applying patches to components deployed by kubeadm.

Field	Description
`directory` `string`	`directory` is a path to a directory that contains files named "target[suffix][+patchtype].extension". For example, "kube-apiserver0+merge.yaml" or just "etcd.json". "target" can be one of "kube-apiserver", "kube-controller-manager", "kube-scheduler", "etcd", "kubeletconfiguration", "corednsdeployment". "patchtype" can be one of "strategic", "merge" or "json" and they match the patch formats supported by kubectl. The default "patchtype" is "strategic". "extension" must be either "json" or "yaml". "suffix" is an optional string that can be used to determine which patches are applied first alpha-numerically.

Proxy

Appears in:

- ClusterConfiguration

Proxy defines the proxy addon that should be used in the cluster.

Field	Description
`disabled` **[Required]** `bool`	`disabled` specifies whether to disable this addon in the cluster.

Timeouts

Appears in:

- InitConfiguration
- JoinConfiguration
- ResetConfiguration

- UpgradeConfiguration

Timeouts holds various timeouts that apply to kubeadm commands.

Field	Description
`controlPlaneComponentHealthCheck` meta/v1.Duration	`controlPlaneComponentHealthCheck` is the amount of time to wait for a control plane component, such as the API server, to be healthy during `kubeadm init` and `kubeadm join`. Default: 4m
`kubeletHealthCheck` meta/v1.Duration	`kubeletHealthCheck` is the amount of time to wait for the kubelet to be healthy during `kubeadm init` and `kubeadm join`. Default: 4m
`kubernetesAPICall` meta/v1.Duration	`kubernetesAPICall` is the amount of time to wait for the kubeadm client to complete a request to the API server. This applies to all types of methods (GET, POST, etc). Default: 1m
`etcdAPICall` meta/v1.Duration	`etcdAPICall` is the amount of time to wait for the kubeadm etcd client to complete a request to the etcd cluster. Default: 2m
`tlsBootstrap` meta/v1.Duration	`tlsBootstrap` is the amount of time to wait for the kubelet to complete TLS bootstrap for a joining node. Default: 5m
`discovery` meta/v1.Duration	`discovery` is the amount of time to wait for kubeadm to validate the API server identity for a joining node. Default: 5m
`upgradeManifests` **[Required]** meta/v1.Duration	`upgradeManifests` is the timeout for upgrading static Pod manifests Default: 5m

UpgradeApplyConfiguration

Appears in:

- UpgradeConfiguration

UpgradeApplyConfiguration contains a list of configurable options which are specific to the "kubeadm upgrade apply" command.

Field	Description
`kubernetesVersion` string	`kubernetesVersion` is the target version of the control plane.

Field	Description
`allowExperimentalUpgrades` `bool`	`allowExperimentalUpgrades` instructs kubeadm to show unstable versions of Kubernetes as an upgrade alternative and allows upgrading to an alpha/beta/ release candidate version of Kubernetes. Default: false
`allowRCUpgrades` `bool`	Enable `allowRCUpgrades` will show release candidate versions of Kubernetes as an upgrade alternative and allows upgrading to a release candidate version of Kubernetes.
`certificateRenewal` `bool`	`certificateRenewal` instructs kubeadm to execute certificate renewal during upgrades. Defaults to true.
`dryRun` `bool`	`dryRun` tells if the dry run mode is enabled, don't apply any change if it is and just output what would be done.
`etcdUpgrade` `bool`	`etcdUpgrade` instructs kubeadm to execute etcd upgrade during upgrades. Defaults to true.
`forceUpgrade` `bool`	`forceUpgrade` flag instructs kubeadm to upgrade the cluster without prompting for confirmation.
`ignorePreflightErrors` `[]string`	`ignorePreflightErrors` provides a slice of pre-flight errors to be ignored during the upgrade process, e.g. `IsPrivilegedUser,Swap` . Value `all` ignores errors from all checks.
`patches` Patches	`patches` contains options related to applying patches to components deployed by kubeadm during `kubeadm upgrade` .
`printConfig` `bool`	`printConfig` specifies whether the configuration file that will be used in the upgrade should be printed or not.
`skipPhases` **[Required]** `[]string`	`skipPhases` is a list of phases to skip during command execution. NOTE: This field is currently ignored for `kubeadm upgrade apply` , but in the future it will be supported.
`imagePullPolicy` core/v1.PullPolicy	`imagePullPolicy` specifies the policy for image pulling during `kubeadm upgrade apply` operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". If this field is unset kubeadm will default it to "IfNotPresent", or pull the required images if not present on the host.

Field	Description
`imagePullSerial` `bool`	`imagePullSerial` specifies if image pulling performed by kubeadm must be done serially or in parallel. Default: true

UpgradeDiffConfiguration

Appears in:

- UpgradeConfiguration

UpgradeDiffConfiguration contains a list of configurable options which are specific to the `kubeadm upgrade diff` command.

Field	Description
`kubernetesVersion` `string`	`kubernetesVersion` is the target version of the control plane.
`contextLines` `int`	`diffContextLines` is the number of lines of context in the diff.

UpgradeNodeConfiguration

Appears in:

- UpgradeConfiguration

UpgradeNodeConfiguration contains a list of configurable options which are specific to the "kubeadm upgrade node" command.

Field	Description
`certificateRenewal` `bool`	`certificateRenewal` instructs kubeadm to execute certificate renewal during upgrades. Defaults to true.
`dryRun` `bool`	`dryRun` tells if the dry run mode is enabled, don't apply any change if it is and just output what would be done.
`etcdUpgrade` `bool`	`etcdUpgrade` instructs kubeadm to execute etcd upgrade during upgrades. Defaults to true.
`ignorePreflightErrors` `[]string`	`ignorePreflightErrors` provides a slice of pre-flight errors to be ignored during the upgrade process, e.g. 'IsPrivilegedUser,Swap'. Value 'all' ignores errors from all checks.
`skipPhases` `[]string`	`skipPhases` is a list of phases to skip during command execution. The list of phases can be obtained with the `kubeadm upgrade node phase --help` command.

Field	Description
`patches` [Patches](#)	`patches` contains options related to applying patches to components deployed by kubeadm during `kubeadm upgrade`.
`imagePullPolicy` `core/v1.PullPolicy`	`imagePullPolicy` specifies the policy for image pulling during `kubeadm upgrade node` operations. The value of this field must be one of "Always", "IfNotPresent" or "Never". If this field is unset kubeadm will default it to "IfNotPresent", or pull the required images if not present on the host.
`imagePullSerial` `bool`	`imagePullSerial` specifies if image pulling performed by kubeadm must be done serially or in parallel. Default: true

UpgradePlanConfiguration

Appears in:

- UpgradeConfiguration

UpgradePlanConfiguration contains a list of configurable options which are specific to the "kubeadm upgrade plan" command.

Field	Description

`kubernetesVersion`
[Required]
`string`
	`kubernetesVersion` is the target version of the control plane.

`allowExperimentalUpgrades` `bool`	`allowExperimentalUpgrades` instructs kubeadm to show unstable versions of Kubernetes as an upgrade alternative and allows upgrading to an alpha/beta/release candidate version of Kubernetes. Default: false
`allowRCUpgrades` `bool`	Enable `allowRCUpgrades` will show release candidate versions of Kubernetes as an upgrade alternative and allows upgrading to a release candidate version of Kubernetes.
`dryRun` `bool`	`dryRun` tells if the dry run mode is enabled, don't apply any change if it is and just output what would be done.
`ignorePreflightErrors` `[]string`	`ignorePreflightErrors` provides a slice of pre-flight errors to be ignored during the upgrade process, e.g. 'IsPrivilegedUser,Swap'. Value 'all' ignores errors from all checks.

Field	Description
`printConfig` `bool`	`printConfig` specifies whether the configuration file that will be used in the upgrade should be printed or not.

14.15 - kubeconfig (v1)

Resource Types

- Config

Config

Config holds the information needed to build connect to remote kubernetes clusters as a given user

Field	Description
`apiVersion` string	`/v1`
`kind` string	`Config`
`kind` `string`	Legacy field from pkg/api/types.go TypeMeta. TODO(jlowdermilk): remove this after eliminating downstream dependencies.
`apiVersion` `string`	Legacy field from pkg/api/types.go TypeMeta. TODO(jlowdermilk): remove this after eliminating downstream dependencies.
`preferences` **[Required]** Preferences	Preferences holds general information to be use for cli interactions
`clusters` **[Required]** []NamedCluster	Clusters is a map of referencable names to cluster configs
`users` **[Required]** []NamedAuthInfo	AuthInfos is a map of referencable names to user configs
`contexts` **[Required]** []NamedContext	Contexts is a map of referencable names to context configs
`current-context` **[Required]** `string`	CurrentContext is the name of the context that you would like to use by default
`extensions` []NamedExtension	Extensions holds additional information. This is useful for extenders so that reads and writes don't clobber unknown fields

AuthInfo

Appears in:

- NamedAuthInfo

AuthInfo contains information that describes identity information. This is use to tell the kubernetes cluster who you are.

Field	Description
`client-certificate` `string`	ClientCertificate is the path to a client cert file for TLS.
`client-certificate-data` `[]byte`	ClientCertificateData contains PEM-encoded data from a client cert file for TLS. Overrides ClientCertificate
`client-key` `string`	ClientKey is the path to a client key file for TLS.
`client-key-data` `[]byte`	ClientKeyData contains PEM-encoded data from a client key file for TLS. Overrides ClientKey
`token` `string`	Token is the bearer token for authentication to the kubernetes cluster.
`tokenFile` `string`	TokenFile is a pointer to a file that contains a bearer token (as described above). If both Token and TokenFile are present, Token takes precedence.
`as` `string`	Impersonate is the username to impersonate. The name matches the flag.
`as-uid` `string`	ImpersonateUID is the uid to impersonate.
`as-groups` `[]string`	ImpersonateGroups is the groups to impersonate.
`as-user-extra` `map[string][]string`	ImpersonateUserExtra contains additional information for impersonated user.
`username` `string`	Username is the username for basic authentication to the kubernetes cluster.
`password` `string`	Password is the password for basic authentication to the kubernetes cluster.
`auth-provider` `AuthProviderConfig`	AuthProvider specifies a custom authentication plugin for the kubernetes cluster.
`exec` `ExecConfig`	Exec specifies a custom exec-based authentication plugin for the kubernetes cluster.
`extensions` `[]NamedExtension`	Extensions holds additional information. This is useful for extenders so that reads and writes don't clobber unknown fields

AuthProviderConfig

Appears in:

- AuthInfo

AuthProviderConfig holds the configuration for a specified auth provider.

Field	Description
`name` **[Required]** `string`	No description provided.
`config` **[Required]** `map[string]string`	No description provided.

Cluster

Appears in:

- NamedCluster

Cluster contains information about how to communicate with a kubernetes cluster

Field	Description
`server` **[Required]** `string`	Server is the address of the kubernetes cluster (https://hostname:port).
`tls-server-name` `string`	TLSServerName is used to check server certificate. If TLSServerName is empty, the hostname used to contact the server is used.
`insecure-skip-tls-verify` `bool`	InsecureSkipTLSVerify skips the validity check for the server's certificate. This will make your HTTPS connections insecure.
`certificate-authority` `string`	CertificateAuthority is the path to a cert file for the certificate authority.
`certificate-authority-data` `[]byte`	CertificateAuthorityData contains PEM-encoded certificate authority certificates. Overrides CertificateAuthority
`proxy-url` `string`	ProxyURL is the URL to the proxy to be used for all requests made by this client. URLs with "http", "https", and "socks5" schemes are supported. If this configuration is not provided or the empty string, the client attempts to construct a proxy configuration from http_proxy and https_proxy environment variables. If these environment variables are not set, the client does not attempt to proxy requests. socks5 proxying does not currently support spdy streaming endpoints (exec, attach, port forward).

Field	Description
`disable-compression` `bool`	DisableCompression allows client to opt-out of response compression for all requests to the server. This is useful to speed up requests (specifically lists) when client-server network bandwidth is ample, by saving time on compression (server-side) and decompression (client-side): https://github.com/kubernetes/kubernetes/issues/112296.
`extensions` `[]NamedExtension`	Extensions holds additional information. This is useful for extenders so that reads and writes don't clobber unknown fields

Context

Appears in:

- NamedContext

Context is a tuple of references to a cluster (how do I communicate with a kubernetes cluster), a user (how do I identify myself), and a namespace (what subset of resources do I want to work with)

Field	Description
`cluster` **[Required]** `string`	Cluster is the name of the cluster for this context
`user` **[Required]** `string`	AuthInfo is the name of the authInfo for this context
`namespace` `string`	Namespace is the default namespace to use on unspecified requests
`extensions` `[]NamedExtension`	Extensions holds additional information. This is useful for extenders so that reads and writes don't clobber unknown fields

ExecConfig

Appears in:

- AuthInfo

ExecConfig specifies a command to provide client credentials. The command is exec'd and outputs structured stdout holding credentials.

See the client.authentication.k8s.io API group for specifications of the exact input and output format

Field	Description
`command` **[Required]** `string`	Command to execute.

Field	Description
`args` `[]string`	Arguments to pass to the command when executing it.
`env` []ExecEnvVar	Env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

`apiVersion`
[Required]
`string`
	Preferred input version of the ExecInfo. The returned ExecCredentials MUST use the same encoding version as the input.

`installHint`
[Required]
`string`
	This text is shown to the user when the executable doesn't seem to be present. For example, `brew install foo-cli` might be a good InstallHint for foo-cli on Mac OS systems.

`provideClusterInfo`
[Required]
`bool`
	ProvideClusterInfo determines whether or not to provide cluster information, which could potentially contain very large CA data, to this exec plugin as a part of the KUBERNETES_EXEC_INFO environment variable. By default, it is set to false. Package k8s.io/client-go/tools/auth/exec provides helper methods for reading this environment variable.

`interactiveMode` ExecInteractiveMode	InteractiveMode determines this plugin's relationship with standard input. Valid values are "Never" (this exec plugin never uses standard input), "IfAvailable" (this exec plugin wants to use standard input if it is available), or "Always" (this exec plugin requires standard input to function). See ExecInteractiveMode values for more details. If APIVersion is client.authentication.k8s.io/v1alpha1 or client.authentication.k8s.io/v1beta1, then this field is optional and defaults to "IfAvailable" when unset. Otherwise, this field is required.

ExecEnvVar

Appears in:

- ExecConfig

ExecEnvVar is used for setting environment variables when executing an exec-based credential plugin.

Field	Description
`name` **[Required]** `string`	No description provided.
`value` **[Required]** `string`	No description provided.

ExecInteractiveMode

(Alias of `string`)

Appears in:

- ExecConfig

ExecInteractiveMode is a string that describes an exec plugin's relationship with standard input.

NamedAuthInfo

Appears in:

- Config

NamedAuthInfo relates nicknames to auth information

Field	Description
`name` **[Required]** `string`	Name is the nickname for this AuthInfo
`user` **[Required]** AuthInfo	AuthInfo holds the auth information

NamedCluster

Appears in:

- Config

NamedCluster relates nicknames to cluster information

Field	Description
`name` **[Required]** `string`	Name is the nickname for this Cluster
`cluster` **[Required]** Cluster	Cluster holds the cluster information

NamedContext

Appears in:

- Config

NamedContext relates nicknames to context information

Field	Description
`name` **[Required]** `string`	Name is the nickname for this Context
`context` **[Required]** Context	Context holds the context information

NamedExtension

Appears in:

- Config
- AuthInfo
- Cluster
- Context
- Preferences

NamedExtension relates nicknames to extension information

Field	Description
`name` **[Required]** `string`	Name is the nickname for this Extension
`extension` **[Required]** `k8s.io/apimachinery/pkg/runtime.RawExtension`	Extension holds the extension information

Preferences

Appears in:

- Config

Field	Description
`colors` `bool`	No description provided.
`extensions` `[]NamedExtension`	Extensions holds additional information. This is useful for extenders so that reads and writes don't clobber unknown fields

14.16 - Kubelet Configuration (v1)

Resource Types

- CredentialProviderConfig

CredentialProviderConfig

CredentialProviderConfig is the configuration containing information about each exec credential provider. Kubelet reads this configuration from disk and enables each provider as specified by the CredentialProvider type.

Field	Description
apiVersion string	kubelet.config.k8s.io/v1
kind string	CredentialProviderConfig
providers **[Required]** []CredentialProvider	providers is a list of credential provider plugins that will be enabled by the kubelet. Multiple providers may match against a single image, in which case credentials from all providers will be returned to the kubelet. If multiple providers are called for a single image, the results are combined. If providers return overlapping auth keys, the value from the provider earlier in this list is used.

CredentialProvider

Appears in:

- CredentialProviderConfig

CredentialProvider represents an exec plugin to be invoked by the kubelet. The plugin is only invoked when an image being pulled matches the images handled by the plugin (see matchImages).

Field	Description
name **[Required]** string	name is the required name of the credential provider. It must match the name of the provider executable as seen by the kubelet. The executable must be in the kubelet's bin directory (set by the --image-credential-provider-bin-dir flag).

Field	Description
`matchImages` **[Required]** `[]string`	matchImages is a required list of strings used to match against images in order to determine if this provider should be invoked. If one of the strings matches the requested image from the kubelet, the plugin will be invoked and given a chance to provide credentials. Images are expected to contain the registry domain and URL path. Each entry in matchImages is a pattern which can optionally contain a port and a path. Globs can be used in the domain, but not in the port or the path. Globs are supported as subdomains like '*.k8s.io' or 'k8s.*.io', and top-level-domains such as 'k8s.*'. *Matching partial subdomains like 'app*.k8s.io' is also supported. Each glob can only match a single subdomain segment, so *.io does not match *.k8s.io. A match exists between an image and a matchImage when all of the below are true: • Both contain the same number of domain parts and each part matches. • The URL path of an imageMatch must be a prefix of the target image URL path. • If the imageMatch contains a port, then the port must match in the image as well. Example values of matchImages: • 123456789.dkr.ecr.us-east-1.amazonaws.com • *.azurecr.io • gcr.io • *.registry.io • registry.io:8080/path
`defaultCacheDuration` **[Required]** `meta/v1.Duration`	defaultCacheDuration is the default duration the plugin will cache credentials in-memory if a cache duration is not provided in the plugin response. This field is required.
`apiVersion` **[Required]** `string`	Required input version of the exec CredentialProviderRequest. The returned CredentialProviderResponse MUST use the same encoding version as the input. Current supported values are: • credentialprovider.kubelet.k8s.io/v1
`args` `[]string`	Arguments to pass to the command when executing it.
`env` `[]ExecEnvVar`	Env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

ExecEnvVar

Appears in:

- CredentialProvider

ExecEnvVar is used for setting environment variables when executing an exec-based credential plugin.

Field	Description
`name` **[Required]** `string`	No description provided.
`value` **[Required]** `string`	No description provided.

14.17 - Kubelet Configuration (v1alpha1)

Resource Types

- CredentialProviderConfig

CredentialProviderConfig

CredentialProviderConfig is the configuration containing information about each exec credential provider. Kubelet reads this configuration from disk and enables each provider as specified by the CredentialProvider type.

Field	Description
`apiVersion` string	`kubelet.config.k8s.io/v1alpha1`
`kind` string	`CredentialProviderConfig`
`providers` **[Required]** []CredentialProvider	providers is a list of credential provider plugins that will be enabled by the kubelet. Multiple providers may match against a single image, in which case credentials from all providers will be returned to the kubelet. If multiple providers are called for a single image, the results are combined. If providers return overlapping auth keys, the value from the provider earlier in this list is used.

CredentialProvider

Appears in:

- CredentialProviderConfig

CredentialProvider represents an exec plugin to be invoked by the kubelet. The plugin is only invoked when an image being pulled matches the images handled by the plugin (see matchImages).

Field	Description
`name` **[Required]** string	name is the required name of the credential provider. It must match the name of the provider executable as seen by the kubelet. The executable must be in the kubelet's bin directory (set by the --image-credential-provider-bin-dir flag).

Field	Description
`matchImages` **[Required]** `[]string`	matchImages is a required list of strings used to match against images in order to determine if this provider should be invoked. If one of the strings matches the requested image from the kubelet, the plugin will be invoked and given a chance to provide credentials. Images are expected to contain the registry domain and URL path. Each entry in matchImages is a pattern which can optionally contain a port and a path. Globs can be used in the domain, but not in the port or the path. Globs are supported as subdomains like `*.k8s.io` or `k8s.*.io`, and top-level-domains such as `k8s.*`. Matching partial subdomains like `app*.k8s.io` is also supported. Each glob can only match a single subdomain segment, so `*.io` does not match `*.k8s.io`. A match exists between an image and a matchImage when all of the below are true: • Both contain the same number of domain parts and each part matches. • The URL path of an imageMatch must be a prefix of the target image URL path. • If the imageMatch contains a port, then the port must match in the image as well. Example values of matchImages: • `123456789.dkr.ecr.us-east-1.amazonaws.com` • `*.azurecr.io` • `gcr.io` • `*.*.registry.io` • `registry.io:8080/path`
`defaultCacheDuration` **[Required]** `meta/v1.Duration`	defaultCacheDuration is the default duration the plugin will cache credentials in-memory if a cache duration is not provided in the plugin response. This field is required.
`apiVersion` **[Required]** `string`	Required input version of the exec CredentialProviderRequest. The returned CredentialProviderResponse MUST use the same encoding version as the input. Current supported values are: • credentialprovider.kubelet.k8s.io/v1alpha1
`args` `[]string`	Arguments to pass to the command when executing it.
`env` `[]ExecEnvVar`	Env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

ExecEnvVar

Appears in:

- CredentialProvider

ExecEnvVar is used for setting environment variables when executing an exec-based credential plugin.

Field	Description
`name` **[Required]** `string`	No description provided.
`value` **[Required]** `string`	No description provided.

14.18 - Kubelet Configuration (v1beta1)

Resource Types

- CredentialProviderConfig
- KubeletConfiguration
- SerializedNodeConfigSource

FormatOptions

Appears in:

- LoggingConfiguration

FormatOptions contains options for the different logging formats.

Field	Description
`text` **[Required]** TextOptions	[Alpha] Text contains options for logging format "text". Only available when the LoggingAlphaOptions feature gate is enabled.
`json` **[Required]** JSONOptions	[Alpha] JSON contains options for logging format "json". Only available when the LoggingAlphaOptions feature gate is enabled.

JSONOptions

Appears in:

- FormatOptions

JSONOptions contains options for logging format "json".

Field	Description
`OutputRoutingOptions` **[Required]** OutputRoutingOptions	(Members of `OutputRoutingOptions` are embedded into this type.) No description provided.

LogFormatFactory

LogFormatFactory provides support for a certain additional, non-default log format.

LoggingConfiguration

Appears in:

- KubeletConfiguration

LoggingConfiguration contains logging options.

Field	Description
`format` **[Required]** `string`	Format Flag specifies the structure of log messages. default value of format is `text`
`flushFrequency` **[Required]** `TimeOrMetaDuration`	Maximum time between log flushes. If a string, parsed as a duration (i.e. "1s") If an int, the maximum number of nanoseconds (i.e. 1s = 1000000000). Ignored if the selected logging backend writes log messages without buffering.
`verbosity` **[Required]** `VerbosityLevel`	Verbosity is the threshold that determines which log messages are logged. Default is zero which logs only the most important messages. Higher values enable additional messages. Error messages are always logged.
`vmodule` **[Required]** `VModuleConfiguration`	VModule overrides the verbosity threshold for individual files. Only supported for "text" log format.
`options` **[Required]** `FormatOptions`	[Alpha] Options holds additional parameters that are specific to the different logging formats. Only the options for the selected format get used, but all of them get validated. Only available when the LoggingAlphaOptions feature gate is enabled.

LoggingOptions

LoggingOptions can be used with ValidateAndApplyWithOptions to override certain global defaults.

Field	Description
`ErrorStream` **[Required]** `io.Writer`	ErrorStream can be used to override the os.Stderr default.
`InfoStream` **[Required]** `io.Writer`	InfoStream can be used to override the os.Stdout default.

OutputRoutingOptions

Appears in:

- JSONOptions
- TextOptions

OutputRoutingOptions contains options that are supported by both "text" and "json".

Field	Description

Field	Description
`splitStream` **[Required]** `bool`	[Alpha] SplitStream redirects error messages to stderr while info messages go to stdout, with buffering. The default is to write both to stdout, without buffering. Only available when the LoggingAlphaOptions feature gate is enabled.
`infoBufferSize` **[Required]** k8s.io/apimachinery/pkg/api/resource.QuantityValue	[Alpha] InfoBufferSize sets the size of the info stream when using split streams. The default is zero, which disables buffering. Only available when the LoggingAlphaOptions feature gate is enabled.

TextOptions

Appears in:

- FormatOptions

TextOptions contains options for logging format "text".

Field	Description
`OutputRoutingOptions` **[Required]** OutputRoutingOptions	(Members of `OutputRoutingOptions` are embedded into this type.) No description provided.

TimeOrMetaDuration

Appears in:

- LoggingConfiguration

TimeOrMetaDuration is present only for backwards compatibility for the flushFrequency field, and new fields should use metav1.Duration.

Field	Description
`Duration` **[Required]** meta/v1.Duration	Duration holds the duration
- **[Required]** `bool`	SerializeAsString controls whether the value is serialized as a string or an integer

TracingConfiguration

Appears in:

- KubeletConfiguration

TracingConfiguration provides versioned configuration for OpenTelemetry tracing clients.

Field	Description

Field	Description
`endpoint` `string`	Endpoint of the collector this component will report traces to. The connection is insecure, and does not currently support TLS. Recommended is unset, and endpoint is the otlp grpc default, localhost:4317.
`samplingRatePerMillion` `int32`	SamplingRatePerMillion is the number of samples to collect per million spans. Recommended is unset. If unset, sampler respects its parent span's sampling rate, but otherwise never samples.

VModuleConfiguration

(Alias of `[]k8s.io/component-base/logs/api/v1.VModuleItem`)

Appears in:

- LoggingConfiguration

VModuleConfiguration is a collection of individual file names or patterns and the corresponding verbosity threshold.

VerbosityLevel

(Alias of `uint32`)

Appears in:

- LoggingConfiguration

VerbosityLevel represents a klog or logr verbosity threshold.

CredentialProviderConfig

CredentialProviderConfig is the configuration containing information about each exec credential provider. Kubelet reads this configuration from disk and enables each provider as specified by the CredentialProvider type.

Field	Description
`apiVersion` string	`kubelet.config.k8s.io/v1beta1`
`kind` string	`CredentialProviderConfig`
`providers` **[Required]** []CredentialProvider	providers is a list of credential provider plugins that will be enabled by the kubelet. Multiple providers may match against a single image, in which case credentials from all providers will be returned to the kubelet. If multiple providers are called for a single image, the results are combined. If providers return overlapping auth keys, the value from the provider earlier in this list is used.

KubeletConfiguration

KubeletConfiguration contains the configuration for the Kubelet

Field	Description
`apiVersion` string	`kubelet.config.k8s.io/v1beta1`
`kind` string	`KubeletConfiguration`
`enableServer` **[Required]** `bool`	enableServer enables Kubelet's secur server. Note: Kubelet's insecure port controlled by the readOnlyPort optio Default: true
`staticPodPath` `string`	staticPodPath is the path to the direc containing local (static) pods to run, c path to a single static pod file. Defaul
`podLogsDir` `string`	podLogsDir is a custom root directory kubelet will use to place pod's log file Default: "/var/log/pods/" Note: it is nc recommended to use the temp folde directory as it may cause unexpected behavior in many places.
`syncFrequency` `meta/v1.Duration`	syncFrequency is the max period bet synchronizing running containers and Default: "1m"
`fileCheckFrequency` `meta/v1.Duration`	fileCheckFrequency is the duration be checking config files for new data. De "20s"
`httpCheckFrequency` `meta/v1.Duration`	httpCheckFrequency is the duration b checking http for new data. Default: "
`staticPodURL` `string`	staticPodURL is the URL for accessing pods to run. Default: ""
`staticPodURLHeader` `map[string][]string`	staticPodURLHeader is a map of slice HTTP headers to use when accessing podURL. Default: nil
`address` `string`	address is the IP address for the Kub serve on (set to 0.0.0.0 for all interfac Default: "0.0.0.0"
`port` `int32`	port is the port for the Kubelet to ser The port number must be between 1 65535, inclusive. Default: 10250

Field	Description
`readOnlyPort` `int32`	readOnlyPort is the read-only port fo Kubelet to serve on with no authentic authorization. The port number must between 1 and 65535, inclusive. Setti field to 0 disables the read-only servi Default: 0 (disabled)
`tlsCertFile` `string`	tlsCertFile is the file containing x509 Certificate for HTTPS. (CA cert, if any, concatenated after server cert). If tlsC and tlsPrivateKeyFile are not provide signed certificate and key are generat the public address and saved to the c passed to the Kubelet's --cert-dir flag. ""
`tlsPrivateKeyFile` `string`	tlsPrivateKeyFile is the file containing private key matching tlsCertFile. Defa
`tlsCipherSuites` `[]string`	tlsCipherSuites is the list of allowed c suites for the server. Note that TLS 1. ciphersuites are not configurable. Val from tls package constants (https:// golang.org/pkg/crypto/tls/#pkg-const Default: nil
`tlsMinVersion` `string`	tlsMinVersion is the minimum TLS ve supported. Values are from tls packag constants (https://golang.org/pkg/cry #pkg-constants). Default: ""
`rotateCertificates` `bool`	rotateCertificates enables client certi rotation. The Kubelet will request a n certificate from the certificates.k8s.io requires an approver to approve the certificate signing requests. Default: f
`serverTLSBootstrap` `bool`	serverTLSBootstrap enables server ce bootstrap. Instead of self signing a se certificate, the Kubelet will request a certificate from the 'certificates.k8s.ic This requires an approver to approve certificate signing requests (CSR). The RotateKubeletServerCertificate featu be enabled when setting this field. De false
`authentication` `KubeletAuthentication`	authentication specifies how request Kubelet's server are authenticated. D anonymous: enabled: false webhook: enabled: true cacheTTL: "2m"

Field	Description
`authorization` [KubeletAuthorization](#)	authorization specifies how requests Kubelet's server are authorized. Defa mode: Webhook webhook: cacheAuthorizedTTL: "5m" cacheUnauthorizedTTL: "30s"
`registryPullQPS` `int32`	registryPullQPS is the limit of registry second. The value must not be a nega number. Setting it to 0 means no limi Default: 5
`registryBurst` `int32`	registryBurst is the maximum size of pulls, temporarily allows pulls to burs number, while still not exceeding registryPullQPS. The value must not b negative number. Only used if registr is greater than 0. Default: 10
`eventRecordQPS` `int32`	eventRecordQPS is the maximum eve creations per second. If 0, there is no enforced. The value cannot be a nega number. Default: 50
`eventBurst` `int32`	eventBurst is the maximum size of a event creations, temporarily allows e creations to burst to this number, wh not exceeding eventRecordQPS. This canot be a negative number and it is used when eventRecordQPS > 0. Defa
`enableDebuggingHandlers` `bool`	enableDebuggingHandlers enables se endpoints for log access and local rur containers and commands, including exec, attach, logs, and portforward fe Default: true
`enableContentionProfiling` `bool`	enableContentionProfiling enables bl profiling, if enableDebuggingHandler Default: false
`healthzPort` `int32`	healthzPort is the port of the localhos healthz endpoint (set to 0 to disable). number is between 1 and 65535. Def 10248
`healthzBindAddress` `string`	healthzBindAddress is the IP address healthz server to serve on. Default: "127.0.0.1"
`oomScoreAdj` `int32`	oomScoreAdj is The oom-score-adj va kubelet process. Values must be with range [-1000, 1000]. Default: -999

Field	Description
`clusterDomain` `string`	clusterDomain is the DNS domain for cluster. If set, kubelet will configure a containers to search this domain in a to the host's search domains. Default
`clusterDNS` `[]string`	clusterDNS is a list of IP addresses fo cluster DNS server. If set, kubelet will configure all containers to use this fo resolution instead of the host's DNS s Default: nil
`streamingConnectionIdleTimeout` `meta/v1.Duration`	streamingConnectionIdleTimeout is t maximum time a streaming connecti be idle before the connection is auto closed. Default: "4h"
`nodeStatusUpdateFrequency` `meta/v1.Duration`	nodeStatusUpdateFrequency is the fr that kubelet computes node status. If lease feature is not enabled, it is also frequency that kubelet posts node st master. Note: When node lease featu enabled, be cautious when changing constant, it must work with nodeMonitorGracePeriod in nodecon Default: "10s"
`nodeStatusReportFrequency` `meta/v1.Duration`	nodeStatusReportFrequency is the fr that kubelet posts node status to ma node status does not change. Kubele ignore this frequency and post node immediately if any change is detected only used when node lease feature is enabled. nodeStatusReportFrequenc default value is 5m. But if nodeStatusUpdateFrequency is set ex nodeStatusReportFrequency's defaul will be set to nodeStatusUpdateFrequ backward compatibility. Default: "5m
`nodeLeaseDurationSeconds` `int32`	nodeLeaseDurationSeconds is the du the Kubelet will set on its correspond Lease. NodeLease provides an indica node health by having the Kubelet cr periodically renew a lease, named aft node, in the kube-node-lease namesp the lease expires, the node can be co unhealthy. The lease is currently rene every 10s, per KEP-0009. In the future lease renewal interval may be set bas the lease duration. The field value mu greater than 0. Default: 40

Field	Description
`imageMinimumGCAge` [meta/v1.Duration](#)	imageMinimumGCAge is the minimu an unused image before it is garbage collected. Default: "2m"
`imageMaximumGCAge` [meta/v1.Duration](#)	imageMaximumGCAge is the maximu an image can be unused before it is g collected. The default of this field is "(which disables this field--meaning im won't be garbage collected based on unused for too long. Default: "0s" (dis
`imageGCHighThresholdPercent` `int32`	imageGCHighThresholdPercent is the of disk usage after which image garba collection is always run. The percent i calculated by dividing this field value so this field must be between 0 and 1 inclusive. When specified, the value n greater than imageGCLowThresholdP Default: 85
`imageGCLowThresholdPercent` `int32`	imageGCLowThresholdPercent is the of disk usage before which image gar collection is never run. Lowest disk us garbage collect to. The percent is calc by dividing this field value by 100, so value must be between 0 and 100, inc When specified, the value must be les imageGCHighThresholdPercent. Defa
`volumeStatsAggPeriod` [meta/v1.Duration](#)	volumeStatsAggPeriod is the frequen calculating and caching volume disk u all pods. Default: "1m"
`kubeletCgroups` `string`	kubeletCgroups is the absolute name cgroups to isolate the kubelet in Defa
`systemCgroups` `string`	systemCgroups is absolute name of c in which to place all non-kernel proce are not already in a container. Empty container. Rolling back the flag requir reboot. The cgroupRoot must be spec this field is not empty. Default: ""
`cgroupRoot` `string`	cgroupRoot is the root cgroup to use pods. This is handled by the containe runtime on a best effort basis.
`cgroupsPerQOS` `bool`	cgroupsPerQOS enable QoS based C(hierarchy: top level CGroups for QoS and all Burstable and BestEffort Pods brought up under their specific top le CGroup. Default: true

Field	Description
`cgroupDriver` `string`	cgroupDriver is the driver kubelet use manipulate CGroups on the host (cgr systemd). Default: "cgroupfs"
`cpuManagerPolicy` `string`	cpuManagerPolicy is the name of the use. Requires the CPUManager featu to be enabled. Default: "None"
`cpuManagerPolicyOptions` `map[string]string`	cpuManagerPolicyOptions is a set of key=value which allows to set extra o fine tune the behaviour of the cpu m policies. Requires both the "CPUMan and "CPUManagerPolicyOptions" feat gates to be enabled. Default: nil
`cpuManagerReconcilePeriod` `meta/v1.Duration`	cpuManagerReconcilePeriod is the reconciliation period for the CPU Mar Requires the CPUManager feature ga enabled. Default: "10s"
`memoryManagerPolicy` `string`	memoryManagerPolicy is the name o policy to use by memory manager. Re the MemoryManager feature gate to enabled. Default: "none"
`topologyManagerPolicy` `string`	topologyManagerPolicy is the name c topology manager policy to use. Valid include: • `restricted` : kubelet only allo with optimal NUMA node alignn requested resources; • `best-effort` : kubelet will favc with NUMA alignment of CPU ar resources; • `none` : kubelet has no knowled NUMA alignment of a pod's CPU device resources. • `single-numa-node` : kubelet or pods with a single NUMA alignm CPU and device resources. Default: "none"
`topologyManagerScope` `string`	topologyManagerScope represents th of topology hint generation that topo manager requests and hint providers generate. Valid values include: • `container` : topology policy is on a per-container basis. • `pod` : topology policy is applied per-pod basis. Default: "container"

Field	Description
`topologyManagerPolicyOptions` `map[string]string`	TopologyManagerPolicyOptions is a s key=value which allows to set extra o fine tune the behaviour of the topolo manager policies. Requires both the "TopologyManager" and "TopologyManagerPolicyOptions" fea gates to be enabled. Default: nil
`qosReserved` `map[string]string`	qosReserved is a set of resource nam percentage pairs that specify the min percentage of a resource reserved fo exclusive use by the guaranteed QoS Currently supported resources: "men Requires the QOSReserved feature ga enabled. Default: nil
`runtimeRequestTimeout` `meta/v1.Duration`	runtimeRequestTimeout is the timeo runtime requests except long running requests - pull, logs, exec and attach. "2m"
`hairpinMode` `string`	hairpinMode specifies how the Kubel should configure the container bridge hairpin packets. Setting this flag allow endpoints in a Service to loadbalance themselves if they should try to acces own Service. Values: • "promiscuous-bridge": make the container bridge promiscuous. • "hairpin-veth": set the hairpin fla container veth interfaces. • "none": do nothing. Generally, one must set `--hairpin-mode=hairpin-veth to` achieve hair because promiscuous-bridge assume existence of a container bridge name Default: "promiscuous-bridge"
`maxPods` `int32`	maxPods is the maximum number of that can run on this Kubelet. The valu be a non-negative integer. Default: 11
`podCIDR` `string`	podCIDR is the CIDR to use for pod IP addresses, only used in standalone m cluster mode, this is obtained from th control plane. Default: ""
`podPidsLimit` `int64`	podPidsLimit is the maximum numbe in any pod. Default: -1

Field	Description
`resolvConf` `string`	resolvConf is the resolver configurati used as the basis for the container DI resolution configuration. If set to the string, will override the default and ef disable DNS lookups. Default: "/etc/ resolv.conf"
`runOnce` `bool`	runOnce causes the Kubelet to check server once for pods, run those in ad the pods specified by static pod files, Default: false
`cpuCFSQuota` `bool`	cpuCFSQuota enables CPU CFS quota enforcement for containers that spec limits. Default: true
`cpuCFSQuotaPeriod` `meta/v1.Duration`	cpuCFSQuotaPeriod is the CPU CFS q period value, `cpu.cfs_period_us` .T must be between 1 ms and 1 second, inclusive. Requires the CustomCPUCFSQuotaPeriod feature g be enabled. Default: "100ms"
`nodeStatusMaxImages` `int32`	nodeStatusMaxImages caps the num images reported in Node.status.imag value must be greater than -2. Note: I specified, no cap will be applied. If 0 i specified, no image is returned. Defau
`maxOpenFiles` `int64`	maxOpenFiles is Number of files that opened by Kubelet process. The valu be a non-negative number. Default: 1
`contentType` `string`	contentType is contentType of reques to apiserver. Default: "application/ vnd.kubernetes.protobuf"
`kubeAPIQPS` `int32`	kubeAPIQPS is the QPS to use while t with kubernetes apiserver. Default: 5
`kubeAPIBurst` `int32`	kubeAPIBurst is the burst to allow wh talking with kubernetes API server. Th cannot be a negative number. Defaul
`serializeImagePulls` `bool`	serializeImagePulls when enabled, te Kubelet to pull images one at a time. recommend *not* changing the default nodes that run docker daemon with v 1.9 or an Aufs storage backend. Issue has more details. Default: true

Field	Description
`maxParallelImagePulls` `int32`	MaxParallelImagePulls sets the maxir number of image pulls in parallel. Thi cannot be set if SerializeImagePulls is Setting it to nil means no limit. Defau
`evictionHard` `map[string]string`	evictionHard is a map of signal name quantities that defines hard eviction thresholds. For example: `{"memory.available": "300Mi"}` . explicitly disable, pass a 0% or 100% threshold on an arbitrary resource. D memory.available: "100Mi" nodefs.av "10%" nodefs.inodesFree: "5%" imagefs.available: "15%"
`evictionSoft` `map[string]string`	evictionSoft is a map of signal names quantities that defines soft eviction thresholds. For example: `{"memory.available": "300Mi"}` . nil
`evictionSoftGracePeriod` `map[string]string`	evictionSoftGracePeriod is a map of s names to quantities that defines grac periods for each soft eviction signal. F example: `{"memory.available": "3` Default: nil
`evictionPressureTransitionPeriod` `meta/v1.Duration`	evictionPressureTransitionPeriod is tl duration for which the kubelet has to before transitioning out of an eviction pressure condition. Default: "5m"
`evictionMaxPodGracePeriod` `int32`	evictionMaxPodGracePeriod is the m allowed grace period (in seconds) to u terminating pods in response to a sof eviction threshold being met. This val effectively caps the Pod's terminationGracePeriodSeconds valu soft evictions. Note: Due to issue #64 behavior has a bug where this value c just overrides the grace period during eviction, which can increase the grace from what is set on the Pod. This bug fixed in a future release. Default: 0
`evictionMinimumReclaim` `map[string]string`	evictionMinimumReclaim is a map of names to quantities that defines mini reclaims, which describe the minimur amount of a given resource the kube reclaim when performing a pod evicti that resource is under pressure. For e `{"imagefs.available": "2Gi"}` . D nil

Field	Description
`podsPerCore` `int32`	podsPerCore is the maximum numbe pods per core. Cannot exceed maxPc value must be a non-negative integer there is no limit on the number of Po Default: 0
`enableControllerAttachDetach` `bool`	enableControllerAttachDetach enable Attach/Detach controller to manage attachment/detachment of volumes scheduled to this node, and disables from executing any attach/detach op Note: attaching/detaching CSI volume supported by the kubelet, so this opt needs to be true for that use case. De true
`protectKernelDefaults` `bool`	protectKernelDefaults, if true, causes Kubelet to error if kernel flags are no expects. Otherwise the Kubelet will at modify kernel flags to match its expe Default: false
`makeIPTablesUtilChains` `bool`	makeIPTablesUtilChains, if true, caus Kubelet to create the KUBE-IPTABLES chain in iptables as a hint to other components about the configuration iptables on the system. Default: true
`iptablesMasqueradeBit` `int32`	iptablesMasqueradeBit formerly cont the creation of the KUBE-MARK-MASC Deprecated: no longer has any effect. 14
`iptablesDropBit` `int32`	iptablesDropBit formerly controlled t creation of the KUBE-MARK-DROP ch Deprecated: no longer has any effect. 15
`featureGates` `map[string]bool`	featureGates is a map of feature nam bools that enable or disable experime features. This field modifies piecemea built-in default values from "k8s.io/ kubernetes/pkg/features/kube_featu Default: nil
`failSwapOn` `bool`	failSwapOn tells the Kubelet to fail to swap is enabled on the node. Default
`memorySwap` `MemorySwapConfiguration`	memorySwap configures swap memo available to container workloads.

Field	Description
`containerLogMaxSize` `string`	containerLogMaxSize is a quantity de the maximum size of the container lo before it is rotated. For example: "5M "256Ki". Default: "10Mi"
`containerLogMaxFiles` `int32`	containerLogMaxFiles specifies the m number of container log files that car present for a container. Default: 5
`containerLogMaxWorkers` `int32`	ContainerLogMaxWorkers specifies tl maximum number of concurrent wor spawn for performing the log rotate operations. Set this count to 1 for dis the concurrent log rotation workflow Default: 1
`containerLogMonitorInterval` `meta/v1.Duration`	ContainerLogMonitorInterval specifie duration at which the container logs a monitored for performing the log rot operation. This defaults to 10 * time.S But can be customized to a smaller va based on the log generation rate and required to be rotated against Defaul
`configMapAndSecretChangeDetectionStrategy` `ResourceChangeDetectionStrategy`	configMapAndSecretChangeDetectio is a mode in which ConfigMap and Se managers are running. Valid values ir • `Get` : kubelet fetches necessary directly from the API server; • `Cache` : kubelet uses TTL cache object fetched from the API serv • `Watch` : kubelet uses watches t observe changes to objects that its interest. Default: "Watch"
`systemReserved` `map[string]string`	systemReserved is a set of ResourceName=ResourceQuantity (e cpu=200m,memory=150G) pairs that resources reserved for non-kubernet components. Currently only cpu and are supported. See http://kubernetes user-guide/compute-resources for m detail. Default: nil

Field	Description
`kubeReserved` `map[string]string`	kubeReserved is a set of ResourceName=ResourceQuantity (e cpu=200m,memory=150G) pairs that resources reserved for kubernetes sy components. Currently cpu, memory local storage for root file system are supported. See https://kubernetes.io/ concepts/configuration/manage-reso containers/ for more details. Default:
`reservedSystemCPUs` **[Required]** `string`	The reservedSystemCPUs option spe CPU list reserved for the host level sy threads and kubernetes related threa provide a "static" CPU list rather than "dynamic" list by systemReserved and kubeReserved. This option does not s systemReservedCgroup or kubeReservedCgroup.
`showHiddenMetricsForVersion` `string`	showHiddenMetricsForVersion is the version for which you want to show h metrics. Only the previous minor vers meaningful, other values will not be a The format is `<major>.<minor>`, e.g The purpose of this format is make su have the opportunity to notice if the r release hides additional metrics, rath being surprised when they are perma removed in the release after that. De
`systemReservedCgroup` `string`	systemReservedCgroup helps the kub identify absolute name of top level C used to enforce `systemReserved` co resource reservation for OS system d Refer to Node Allocatable doc for mo information. Default: ""
`kubeReservedCgroup` `string`	kubeReservedCgroup helps the kube identify absolute name of top level C used to enforce `KubeReserved` com resource reservation for Kubernetes system daemons. Refer to Node Alloc doc for more information. Default: ""

Field	Description
`enforceNodeAllocatable` `[]string`	This flag specifies the various Node Allocatable enforcements that Kubele to perform. This flag accepts a list of Acceptable options are `none` , `pods` `system-reserved` and `kube-reser` `none` is specified, no other options specified. When `system-reserved` i list, systemReservedCgroup must be specified. When `kube-reserved` is i kubeReservedCgroup must be specifi field is supported only when `cgroup` is set to true. Refer to Node Allocatab more information. Default: ["pods"]
`allowedUnsafeSysctls` `[]string`	A comma separated whitelist of unsa or sysctl patterns (ending in `*`). Uns groups are `kernel.shm*` , `kernel.m` `kernel.sem` , `fs.mqueue.*` , and `n` For example: " `kernel.msg*,net.ipv4.route.min` Default: []
`volumePluginDir` `string`	volumePluginDir is the full path of the directory in which to search for additi third party volume plugins. Default: " libexec/kubernetes/kubelet-plugins/v exec/"
`providerID` `string`	providerID, if set, sets the unique ID o instance that an external provider (i.e cloudprovider) can use to identify a s node. Default: ""
`kernelMemcgNotification` `bool`	kernelMemcgNotification, if set, instr kubelet to integrate with the kernel m notification for determining if memor eviction thresholds are exceeded rath polling. Default: false
`logging` **[Required]** `LoggingConfiguration`	logging specifies the options of loggir to Logs Options for more information Default: Format: text
`enableSystemLogHandler` `bool`	enableSystemLogHandler enables sys via web interface host:port/logs/ Defa
`enableSystemLogQuery` `bool`	enableSystemLogQuery enables the query feature on the /logs endpoint. EnableSystemLogHandler has to be e in addition for this feature to work. D false

Field	Description
`shutdownGracePeriod` [meta/v1.Duration](#)	shutdownGracePeriod specifies the t duration that the node should delay t shutdown and total grace period for p termination during a node shutdown "0s"
`shutdownGracePeriodCriticalPods` [meta/v1.Duration](#)	shutdownGracePeriodCriticalPods sp the duration used to terminate critica during a node shutdown. This should than shutdownGracePeriod. For exan shutdownGracePeriod=30s, and shutdownGracePeriodCriticalPods=1(during a node shutdown the first 20 s would be reserved for gracefully term normal pods, and the last 10 seconds be reserved for terminating critical pc Default: "0s"
`shutdownGracePeriodByPodPriority` [][ShutdownGracePeriodByPodPriority](#)	shutdownGracePeriodByPodPriority s the shutdown grace period for Pods b their associated priority class value. W shutdown request is received, the Ku initiate shutdown on all pods running node with a grace period that depenc priority of the pod, and then wait for to exit. Each entry in the array repres graceful shutdown time a pod with a class value that lies in the range of th and the next higher entry in the list w node is shutting down. For example, critical pods 10s to shutdown, priority>=10000 pods 20s to shutdow all remaining pods 30s to shutdown. shutdownGracePeriodByPodPriority: • priority: 2000000000 shutdownGracePeriodSeconds: • priority: 10000 shutdownGracePeriodSeconds: • priority: 0 shutdownGracePeriodSeconds: The time the Kubelet will wait before will at most be the maximum of all shutdownGracePeriodSeconds for ea priority class range represented on th When all pods have exited or reached grace periods, the Kubelet will release shutdown inhibit lock. Requires the GracefulNodeShutdown feature gate enabled. This configuration must be e either ShutdownGracePeriod or ShutdownGracePeriodCriticalPods is Default: nil

Field	Description
`reservedMemory` [`[]MemoryReservation`]	reservedMemory specifies a comma-separated list of memory reservation NUMA nodes. The parameter makes only in the context of the memory ma feature. The memory manager will nc allocate reserved memory for contair workloads. For example, if you have a with 10Gi of memory and the reservedMemory was specified to res of memory at NUMA0, the memory m will assume that only 9Gi is available allocation. You can specify a different of NUMA node and memory types. Yc omit this parameter at all, but you sh aware that the amount of reserved m from all NUMA nodes should be equa amount of memory specified by the r allocatable. If at least one node alloca parameter has a non-zero value, you to specify at least one NUMA node. A avoid specifying: 1. Duplicates, the same NUMA noc memory type, but with a differe 2. zero limits for any memory type 3. NUMAs nodes IDs that do not e under the machine. 4. memory types except for memc hugepages- Default: nil
`enableProfilingHandler` `bool`	enableProfilingHandler enables profi web interface host:port/debug/pprof/ true
`enableDebugFlagsHandler` `bool`	enableDebugFlagsHandler enables fl endpoint via web interface host:port/ flags/v Default: true
`seccompDefault` `bool`	SeccompDefault enables the use of `RuntimeDefault` as the default secc profile for all workloads. Default: fals
`memoryThrottlingFactor` `float64`	MemoryThrottlingFactor specifies the multiplied by the memory limit or no allocatable memory when setting the cgroupv2 memory.high value to enfo MemoryQoS. Decreasing this factor w lower high limit for container cgroups heavier reclaim pressure while increa put less reclaim pressure. See https:/ kep.k8s.io/2570 for more details. Def

Field	Description
`registerWithTaints` []core/v1.Taint	registerWithTaints are an array of tai add to a node object when the kubel registers itself. This only takes effect registerNode is true and upon the ini registration of the node. Default: nil
`registerNode` `bool`	registerNode enables automatic regis with the apiserver. Default: true
`tracing` TracingConfiguration	Tracing specifies the versioned config for OpenTelemetry tracing clients. Se kep.k8s.io/2832 for more details. Def
`localStorageCapacityIsolation` `bool`	LocalStorageCapacityIsolation enable ephemeral storage isolation feature. default setting is true. This feature all users to set request/limit for containe ephemeral storage and manage it in way as cpu and memory. It also allow sizeLimit for emptyDir volume, which trigger pod eviction if disk usage from volume exceeds the limit. This featur depends on the capability of detectin root file system disk usage. For certai systems, such as kind rootless, if this capability cannot be supported, the f LocalStorageCapacityIsolation should disabled. Once disabled, user should request/limit for container's epheme storage, or sizeLimit for emptyDir. De true
`containerRuntimeEndpoint` **[Required]** `string`	ContainerRuntimeEndpoint is the enc container runtime. Unix Domain Sock supported on Linux, while npipe and endpoints are supported on Windows Examples:'unix:///path/to/runtime.so 'npipe:////./pipe/runtime'
`imageServiceEndpoint` `string`	ImageServiceEndpoint is the endpoin container image service. Unix Domai are supported on Linux, while npipe endpoints are supported on Windows Examples:'unix:///path/to/runtime.so 'npipe:////./pipe/runtime'. If not speci value in containerRuntimeEndpoint is
`failCgroupV1` `bool`	FailCgroupV1 prevents the kubelet fr starting on hosts that use cgroup v1. default, this is set to 'false', meaning kubelet is allowed to start on cgroup unless this option is explicitly enable Default: false

SerializedNodeConfigSource

SerializedNodeConfigSource allows us to serialize v1.NodeConfigSource. This type is used internally by the Kubelet for tracking checkpointed dynamic configs. It exists in the kubeletconfig API group because it is classified as a versioned input to the Kubelet.

Field	Description
`apiVersion` string	`kubelet.config.k8s.io/v1beta1`
`kind` string	`SerializedNodeConfigSource`
`source` core/v1.NodeConfigSource	source is the source that we are serializing.

CredentialProvider

Appears in:

- CredentialProviderConfig

CredentialProvider represents an exec plugin to be invoked by the kubelet. The plugin is only invoked when an image being pulled matches the images handled by the plugin (see matchImages).

Field	Description
`name` **[Required]** `string`	name is the required name of the credential provider. It must match the name of the provider executable as seen by the kubelet. The executable must be in the kubelet's bin directory (set by the --image-credential-provider-bin-dir flag).

Field	Description
`matchImages` **[Required]** `[]string`	matchImages is a required list of strings used to match against images in order to determine if this provider should be invoked. If one of the strings matches the requested image from the kubelet, the plugin will be invoked and given a chance to provide credentials. Images are expected to contain the registry domain and URL path. Each entry in matchImages is a pattern which can optionally contain a port and a path. Globs can be used in the domain, but not in the port or the path. Globs are supported as subdomains like '*.k8s.io' or 'k8s.*.io', and top-level-domains such as 'k8s.*'. *Matching partial subdomains like 'app*.k8s.io' is also supported. Each glob can only match a single subdomain segment, so *.io does not match *.k8s.io. A match exists between an image and a matchImage when all of the below are true: • Both contain the same number of domain parts and each part matches. • The URL path of an imageMatch must be a prefix of the target image URL path. • If the imageMatch contains a port, then the port must match in the image as well. Example values of matchImages: • 123456789.dkr.ecr.us-east-1.amazonaws.com • *.azurecr.io • gcr.io • *.registry.io • registry.io:8080/path
`defaultCacheDuration` **[Required]** `meta/v1.Duration`	defaultCacheDuration is the default duration the plugin will cache credentials in-memory if a cache duration is not provided in the plugin response. This field is required.
`apiVersion` **[Required]** `string`	Required input version of the exec CredentialProviderRequest. The returned CredentialProviderResponse MUST use the same encoding version as the input. Current supported values are: • credentialprovider.kubelet.k8s.io/v1beta1
`args` `[]string`	Arguments to pass to the command when executing it.
`env` `[]ExecEnvVar`	Env defines additional environment variables to expose to the process. These are unioned with the host's environment, as well as variables client-go uses to pass argument to the plugin.

ExecEnvVar

Appears in:

- CredentialProvider

ExecEnvVar is used for setting environment variables when executing an exec-based credential plugin.

Field	Description
`name` **[Required]** `string`	No description provided.
`value` **[Required]** `string`	No description provided.

KubeletAnonymousAuthentication

Appears in:

- KubeletAuthentication

Field	Description
`enabled` `bool`	enabled allows anonymous requests to the kubelet server. Requests that are not rejected by another authentication method are treated as anonymous requests. Anonymous requests have a username of `system:anonymous`, and a group name of `system:unauthenticated`.

KubeletAuthentication

Appears in:

- KubeletConfiguration

Field	Description
`x509` `KubeletX509Authentication`	x509 contains settings related to x509 client certificate authentication.
`webhook` `KubeletWebhookAuthentication`	webhook contains settings related to webhook bearer token authentication.
`anonymous` `KubeletAnonymousAuthentication`	anonymous contains settings related to anonymous authentication.

KubeletAuthorization

Appears in:

- KubeletConfiguration

Field	Description

Field	Description
mode KubeletAuthorizationMode	mode is the authorization mode to apply to requests to the kubelet server. Valid values are `AlwaysAllow` and `Webhook`. Webhook mode uses the SubjectAccessReview API to determine authorization.
webhook KubeletWebhookAuthorization	webhook contains settings related to Webhook authorization.

KubeletAuthorizationMode

(Alias of `string`)

Appears in:

- KubeletAuthorization

KubeletWebhookAuthentication

Appears in:

- KubeletAuthentication

Field	Description
enabled bool	enabled allows bearer token authentication backed by the tokenreviews.authentication.k8s.io API.
cacheTTL meta/v1.Duration	cacheTTL enables caching of authentication results

KubeletWebhookAuthorization

Appears in:

- KubeletAuthorization

Field	Description
cacheAuthorizedTTL meta/v1.Duration	cacheAuthorizedTTL is the duration to cache 'authorized' responses from the webhook authorizer.
cacheUnauthorizedTTL meta/v1.Duration	cacheUnauthorizedTTL is the duration to cache 'unauthorized' responses from the webhook authorizer.

KubeletX509Authentication

Appears in:

- KubeletAuthentication

Field	Description
`clientCAFile` `string`	clientCAFile is the path to a PEM-encoded certificate bundle. If set, any request presenting a client certificate signed by one of the authorities in the bundle is authenticated with a username corresponding to the CommonName, and groups corresponding to the Organization in the client certificate.

MemoryReservation

Appears in:

- KubeletConfiguration

MemoryReservation specifies the memory reservation of different types for each NUMA node

Field	Description
`numaNode` **[Required]** `int32`	No description provided.
`limits` **[Required]** `core/v1.ResourceList`	No description provided.

MemorySwapConfiguration

Appears in:

- KubeletConfiguration

Field	Description
`swapBehavior` `string`	swapBehavior configures swap memory available to container workloads. May be one of "", "NoSwap": workloads can not use swap, default option. "LimitedSwap": workload swap usage is limited. The swap limit is proportionate to the container's memory request.

ResourceChangeDetectionStrategy

(Alias of `string`)

Appears in:

- KubeletConfiguration

ResourceChangeDetectionStrategy denotes a mode in which internal managers (secret, configmap) are discovering object changes.

ShutdownGracePeriodByPodPriority

Appears in:

- KubeletConfiguration

ShutdownGracePeriodByPodPriority specifies the shutdown grace period
for Pods based on their associated priority class value

Field	Description
`priority` **[Required]** `int32`	priority is the priority value associated with the shutdown grace period
`shutdownGracePeriodSeconds` **[Required]** `int64`	shutdownGracePeriodSeconds is the shutdown grace period in seconds

14.19 - Kubelet CredentialProvider (v1)

Resource Types

- CredentialProviderRequest
- CredentialProviderResponse

`CredentialProviderRequest`

CredentialProviderRequest includes the image that the kubelet requires authentication for. Kubelet will pass this request object to the plugin via stdin. In general, plugins should prefer responding with the same apiVersion they were sent.

Field	Description
`apiVersion` string	`credentialprovider.kubelet.k8s.io/v1`
`kind` string	`CredentialProviderRequest`
`image` **[Required]** `string`	image is the container image that is being pulled as part of the credential provider plugin request. Plugins may optionally parse the image to extract any information required to fetch credentials.

`CredentialProviderResponse`

CredentialProviderResponse holds credentials that the kubelet should use for the specified image provided in the original request. Kubelet will read the response from the plugin via stdout. This response should be set to the same apiVersion as CredentialProviderRequest.

Field	Description
`apiVersion` string	`credentialprovider.kubelet.k8s.io/v1`
`kind` string	`CredentialProviderResponse`
`cacheKeyType` **[Required]** `PluginCacheKeyType`	cacheKeyType indiciates the type of caching key to use based on the image provided in the request. There are three valid values for the cache key type: Image, Registry, and Global. If an invalid value is specified, the response will NOT be used by the kubelet.

Field	Description
`cacheDuration` [meta/v1.Duration](#)	cacheDuration indicates the duration the provided credentials should be cached for. The kubelet will use this field to set the in-memory cache duration for credentials in the AuthConfig. If null, the kubelet will use defaultCacheDuration provided in CredentialProviderConfig. If set to 0, the kubelet will not cache the provided AuthConfig.
`auth` [map[string]AuthConfig](#)	auth is a map containing authentication information passed into the kubelet. Each key is a match image string (more on this below). The corresponding authConfig value should be valid for all images that match against this key. A plugin should set this field to null if no valid credentials can be returned for the requested image. Each key in the map is a pattern which can optionally contain a port and a path. Globs can be used in the domain, but not in the port or the path. Globs are supported as subdomains like '*.k8s.io' or 'k8s.*.io', and top-level-domains such as 'k8s.*'. Matching partial subdomains like 'app*.k8s.io' is also supported. Each glob can only match a single subdomain segment, so *.io does not match *.k8s.io. The kubelet will match images against the key when all of the below are true: • Both contain the same number of domain parts and each part matches. • The URL path of an imageMatch must be a prefix of the target image URL path. • If the imageMatch contains a port, then the port must match in the image as well. When multiple keys are returned, the kubelet will traverse all keys in reverse order so that: • longer keys come before shorter keys with the same prefix • non-wildcard keys come before wildcard keys with the same prefix. For any given match, the kubelet will attempt an image pull with the provided credentials, stopping after the first successfully authenticated pull. Example keys: • 123456789.dkr.ecr.us-east-1.amazonaws.com • *.azurecr.io • gcr.io • *.registry.io • registry.io:8080/path

AuthConfig

Appears in:

- CredentialProviderResponse

AuthConfig contains authentication information for a container registry.
Only username/password based authentication is supported today, but
more authentication mechanisms may be added in the future.

Field	Description
`username` **[Required]** `string`	username is the username used for authenticating to the container registry An empty username is valid.
`password` **[Required]** `string`	password is the password used for authenticating to the container registry An empty password is valid.

PluginCacheKeyType

(Alias of `string`)

Appears in:

- CredentialProviderResponse

14.20 - WebhookAdmission Configuration (v1)

Package v1 is the v1 version of the API.

Resource Types

- WebhookAdmission

WebhookAdmission

WebhookAdmission provides configuration for the webhook admission controller.

Field	Description
`apiVersion` string	`apiserver.config.k8s.io/v1`
`kind` string	`WebhookAdmission`
`kubeConfigFile` **[Required]** `string`	KubeConfigFile is the path to the kubeconfig file.

15 - External APIs

15.1 - Kubernetes Custom Metrics (v1beta2)

Package v1beta2 is the v1beta2 version of the custom_metrics API.

Resource Types

- MetricListOptions
- MetricValue
- MetricValueList

`MetricListOptions`

MetricListOptions is used to select metrics by their label selectors

Field	Description
`apiVersion` string	`custom.metrics.k8s.io/v1beta2`
`kind` string	`MetricListOptions`
`labelSelector` `string`	A selector to restrict the list of returned objects by their labels. Defaults to everything.
`metricLabelSelector` `string`	A selector to restrict the list of returned metrics by their labels

`MetricValue`

Appears in:

- MetricValueList

MetricValue is the metric value for some object

Field	Description
`apiVersion` string	`custom.metrics.k8s.io/v1beta2`
`kind` string	`MetricValue`
`describedObject` **[Required]** `core/v1.ObjectReference`	a reference to the described object

Field	Description
`metric` **[Required]** [MetricIdentifier](#)	No description provided.
`timestamp` **[Required]** [meta/v1.Time](#)	indicates the time at which the metrics were produced
`windowSeconds` **[Required]** `int64`	indicates the window ([Timestamp-Window, Timestamp]) from which these metrics were calculated, when returning rate metrics calculated from cumulative metrics (or zero for non-calculated instantaneous metrics).
`value` **[Required]** [k8s.io/apimachinery/pkg/api/resource.Quantity](#)	the value of the metric for this

MetricValueList

MetricValueList is a list of values for a given metric for some set of objects

Field	Description
`apiVersion` string	`custom.metrics.k8s.io/v1beta2`
`kind` string	`MetricValueList`
`metadata` **[Required]** [meta/v1.ListMeta](#)	No description provided.
`items` **[Required]** [][MetricValue](#)	the value of the metric across the described objects

MetricIdentifier

Appears in:

- MetricValue

MetricIdentifier identifies a metric by name and, optionally, selector

Field	Description
`name` **[Required]** `string`	name is the name of the given metric
`selector` [meta/v1.LabelSelector](#)	selector represents the label selector that could be used to select this metric, and will generally just be the selector passed in to the query used to fetch this metric. When left blank, only the metric's Name will be used to gather metrics.

15.2 - Kubernetes External Metrics (v1beta1)

Package v1beta1 is the v1beta1 version of the external metrics API.

Resource Types

- ExternalMetricValue
- ExternalMetricValueList

`ExternalMetricValue`

Appears in:

- ExternalMetricValueList

ExternalMetricValue is a metric value for external metric A single metric value is identified by metric name and a set of string labels. For one metric there can be multiple values with different sets of labels.

Field	Description
`apiVersion` string	`external.metrics.k8s.io/v1beta1`
`kind` string	`ExternalMetricValue`
`metricName` **[Required]** `string`	the name of the metric
`metricLabels` **[Required]** `map[string]string`	a set of labels that identify a single time series for the metric
`timestamp` **[Required]** `meta/v1.Time`	indicates the time at which the metrics were produced
`window` **[Required]** `int64`	indicates the window ([Timestamp-Window, Timestamp]) from which these metrics were calculated, when returning rate metrics calculated from cumulative metrics (or zero for non-calculated instantaneous metrics).
`value` **[Required]** `k8s.io/apimachinery/pkg/api/resource.Quantity`	the value of the metric

`ExternalMetricValueList`

ExternalMetricValueList is a list of values for a given metric for some set labels

Field	Description
`apiVersion` string	`external.metrics.k8s.io/v1beta1`
`kind` string	`ExternalMetricValueList`
`metadata` **[Required]** meta/v1.ListMeta	No description provided.
`items` **[Required]** []ExternalMetricValue	value of the metric matching a given set of labels

15.3 - Kubernetes Metrics (v1beta1)

Package v1beta1 is the v1beta1 version of the metrics API.

Resource Types

- NodeMetrics
- NodeMetricsList
- PodMetrics
- PodMetricsList

NodeMetrics

Appears in:

- NodeMetricsList

NodeMetrics sets resource usage metrics of a node.

Field	Description
`apiVersion` string	`metrics.k8s.io/v1beta1`
`kind` string	`NodeMetrics`
`metadata` meta/v1.ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata Refer to the Kubernetes API documentation for the fields of the `metadata` field.
`timestamp` **[Required]** meta/v1.Time	The following fields define time interval from which metrics were collected from the interval [Timestamp-Window, Timestamp].
`window` **[Required]** meta/v1.Duration	No description provided.
`usage` **[Required]** core/v1.ResourceList	The memory usage is the memory working set.

NodeMetricsList

NodeMetricsList is a list of NodeMetrics.

Field	Description
`apiVersion` string	`metrics.k8s.io/v1beta1`

Field	Description
`kind` string	`NodeMetricsList`
`metadata` **[Required]** meta/v1.ListMeta	Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
`items` **[Required]** []NodeMetrics	List of node metrics.

PodMetrics

Appears in:

- PodMetricsList

PodMetrics sets resource usage metrics of a pod.

Field	Description
`apiVersion` string	`metrics.k8s.io/v1beta1`
`kind` string	`PodMetrics`
`metadata` meta/v1.ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata Refer to the Kubernetes API documentation for the fields of the `metadata` field.
`timestamp` **[Required]** meta/v1.Time	The following fields define time interval from which metrics were collected from the interval [Timestamp-Window, Timestamp].
`window` **[Required]** meta/v1.Duration	No description provided.
`containers` **[Required]** []ContainerMetrics	Metrics for all containers are collected within the same time window.

PodMetricsList

PodMetricsList is a list of PodMetrics.

Field	Description
`apiVersion` string	`metrics.k8s.io/v1beta1`

Field	Description
`kind` string	`PodMetricsList`
`metadata` **[Required]** meta/v1.ListMeta	Standard list metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
`items` **[Required]** []PodMetrics	List of pod metrics.

ContainerMetrics

Appears in:

- PodMetrics

ContainerMetrics sets resource usage metrics of a container.

Field	Description
`name` **[Required]** `string`	Container name corresponding to the one from pod.spec.containers.
`usage` **[Required]** core/v1.ResourceList	The memory usage is the memory working set.

16 - Scheduling

16.1 - Scheduler Configuration

ⓘ **FEATURE STATE:** `Kubernetes v1.25 [stable]`

You can customize the behavior of the `kube-scheduler` by writing a configuration file and passing its path as a command line argument.

A scheduling Profile allows you to configure the different stages of scheduling in the kube-scheduler. Each stage is exposed in an extension point. Plugins provide scheduling behaviors by implementing one or more of these extension points.

You can specify scheduling profiles by running `kube-scheduler --config <filename>`, using the KubeSchedulerConfiguration v1 struct.

A minimal configuration looks as follows:

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
clientConnection:
  kubeconfig: /etc/srv/kubernetes/kube-scheduler/kubeconfig
```

> **Note:**
> KubeSchedulerConfiguration v1beta3 is deprecated in v1.26 and is removed in v1.29. Please migrate KubeSchedulerConfiguration to v1.

Profiles

A scheduling Profile allows you to configure the different stages of scheduling in the kube-scheduler. Each stage is exposed in an extension point. Plugins provide scheduling behaviors by implementing one or more of these extension points.

You can configure a single instance of `kube-scheduler` to run multiple profiles.

Extension points

Scheduling happens in a series of stages that are exposed through the following extension points:

1. `queueSort`: These plugins provide an ordering function that is used to sort pending Pods in the scheduling queue. Exactly one queue sort plugin may be enabled at a time.
2. `preFilter`: These plugins are used to pre-process or check information about a Pod or the cluster before filtering. They can mark a pod as unschedulable.
3. `filter`: These plugins are the equivalent of Predicates in a scheduling Policy and are used to filter out nodes that can not run the Pod. Filters are called in the configured order. A pod is marked as unschedulable if no nodes pass all the filters.
4. `postFilter`: These plugins are called in their configured order when

no feasible nodes were found for the pod. If any `postFilter` plugin marks the Pod *schedulable*, the remaining plugins are not called.

5. `preScore` : This is an informational extension point that can be used for doing pre-scoring work.
6. `score` : These plugins provide a score to each node that has passed the filtering phase. The scheduler will then select the node with the highest weighted scores sum.
7. `reserve` : This is an informational extension point that notifies plugins when resources have been reserved for a given Pod. Plugins also implement an `Unreserve` call that gets called in the case of failure during or after `Reserve` .
8. `permit` : These plugins can prevent or delay the binding of a Pod.
9. `preBind` : These plugins perform any work required before a Pod is bound.
10. `bind` : The plugins bind a Pod to a Node. `bind` plugins are called in order and once one has done the binding, the remaining plugins are skipped. At least one bind plugin is required.
11. `postBind` : This is an informational extension point that is called after a Pod has been bound.
12. `multiPoint` : This is a config-only field that allows plugins to be enabled or disabled for all of their applicable extension points simultaneously.

For each extension point, you could disable specific default plugins or enable your own. For example:

```
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - plugins:
      score:
        disabled:
        - name: PodTopologySpread
        enabled:
        - name: MyCustomPluginA
          weight: 2
        - name: MyCustomPluginB
          weight: 1
```

You can use `*` as name in the disabled array to disable all default plugins for that extension point. This can also be used to rearrange plugins order, if desired.

Scheduling plugins

The following plugins, enabled by default, implement one or more of these extension points:

- `ImageLocality` : Favors nodes that already have the container images that the Pod runs. Extension points: `score` .
- `TaintToleration` : Implements taints and tolerations. Implements extension points: `filter` , `preScore` , `score` .
- `NodeName` : Checks if a Pod spec node name matches the current node. Extension points: `filter` .
- `NodePorts` : Checks if a node has free ports for the requested Pod ports. Extension points: `preFilter` , `filter` .
- `NodeAffinity` : Implements node selectors and node affinity. Extension points: `filter` , `score` .
- `PodTopologySpread` : Implements Pod topology spread. Extension

points: `preFilter`, `filter`, `preScore`, `score`.

- `NodeUnschedulable`: Filters out nodes that have `.spec.unschedulable` set to true. Extension points: `filter`.
- `NodeResourcesFit`: Checks if the node has all the resources that the Pod is requesting. The score can use one of three strategies: `LeastAllocated` (default), `MostAllocated` and `RequestedToCapacityRatio`. Extension points: `preFilter`, `filter`, `score`.
- `NodeResourcesBalancedAllocation`: Favors nodes that would obtain a more balanced resource usage if the Pod is scheduled there. Extension points: `score`.
- `VolumeBinding`: Checks if the node has or if it can bind the requested volumes. Extension points: `preFilter`, `filter`, `reserve`, `preBind`, `score`.

> **Note:**
> `score` extension point is enabled when `VolumeCapacityPriority` feature is enabled. It prioritizes the smallest PVs that can fit the requested volume size.

- `VolumeRestrictions`: Checks that volumes mounted in the node satisfy restrictions that are specific to the volume provider. Extension points: `filter`.
- `VolumeZone`: Checks that volumes requested satisfy any zone requirements they might have. Extension points: `filter`.
- `NodeVolumeLimits`: Checks that CSI volume limits can be satisfied for the node. Extension points: `filter`.
- `EBSLimits`: Checks that AWS EBS volume limits can be satisfied for the node. Extension points: `filter`.
- `GCEPDLimits`: Checks that GCP-PD volume limits can be satisfied for the node. Extension points: `filter`.
- `AzureDiskLimits`: Checks that Azure disk volume limits can be satisfied for the node. Extension points: `filter`.
- `InterPodAffinity`: Implements inter-Pod affinity and anti-affinity. Extension points: `preFilter`, `filter`, `preScore`, `score`.
- `PrioritySort`: Provides the default priority based sorting. Extension points: `queueSort`.
- `DefaultBinder`: Provides the default binding mechanism. Extension points: `bind`.
- `DefaultPreemption`: Provides the default preemption mechanism. Extension points: `postFilter`.

You can also enable the following plugins, through the component config APIs, that are not enabled by default:

- `CinderLimits`: Checks that OpenStack Cinder volume limits can be satisfied for the node. Extension points: `filter`.

Multiple profiles

You can configure `kube-scheduler` to run more than one profile. Each profile has an associated scheduler name and can have a different set of plugins configured in its extension points.

With the following sample configuration, the scheduler will run with two profiles: one with the default plugins and one with all scoring plugins disabled.

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: default-scheduler
  - schedulerName: no-scoring-scheduler
    plugins:
      preScore:
        disabled:
        - name: '*'
      score:
        disabled:
        - name: '*'
```

Pods that want to be scheduled according to a specific profile can include the corresponding scheduler name in its `.spec.schedulerName`.

By default, one profile with the scheduler name `default-scheduler` is created. This profile includes the default plugins described above. When declaring more than one profile, a unique scheduler name for each of them is required.

If a Pod doesn't specify a scheduler name, kube-apiserver will set it to `default-scheduler`. Therefore, a profile with this scheduler name should exist to get those pods scheduled.

> **Note:**
> Pod's scheduling events have `.spec.schedulerName` as their `reportingController`. Events for leader election use the scheduler name of the first profile in the list.
>
> For more information, please refer to the `reportingController` section under Event API Reference.

> **Note:**
> All profiles must use the same plugin in the `queueSort` extension point and have the same configuration parameters (if applicable). This is because the scheduler only has one pending pods queue.

Plugins that apply to multiple extension points

Starting from `kubescheduler.config.k8s.io/v1beta3`, there is an additional field in the profile config, `multiPoint`, which allows for easily enabling or disabling a plugin across several extension points. The intent of `multiPoint` config is to simplify the configuration needed for users and administrators when using custom profiles.

Consider a plugin, `MyPlugin`, which implements the `preScore`, `score`, `preFilter`, and `filter` extension points. To enable `MyPlugin` for all its available extension points, the profile config looks like:

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: multipoint-scheduler
    plugins:
      multiPoint:
        enabled:
        - name: MyPlugin
```

This would equate to manually enabling `MyPlugin` for all of its extension points, like so:

```
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: non-multipoint-scheduler
    plugins:
      preScore:
        enabled:
        - name: MyPlugin
      score:
        enabled:
        - name: MyPlugin
      preFilter:
        enabled:
        - name: MyPlugin
      filter:
        enabled:
        - name: MyPlugin
```

One benefit of using `multiPoint` here is that if `MyPlugin` implements another extension point in the future, the `multiPoint` config will automatically enable it for the new extension.

Specific extension points can be excluded from `MultiPoint` expansion using the `disabled` field for that extension point. This works with disabling default plugins, non-default plugins, or with the wildcard (`'*'`) to disable all plugins. An example of this, disabling `Score` and `PreScore`, would be:

```
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: non-multipoint-scheduler
    plugins:
      multiPoint:
        enabled:
        - name: 'MyPlugin'
      preScore:
        disabled:
        - name: '*'
      score:
        disabled:
        - name: '*'
```

Starting from `kubescheduler.config.k8s.io/v1beta3`, all default plugins are enabled internally through `MultiPoint`. However, individual extension points are still available to allow flexible reconfiguration of the default values (such as ordering and Score weights). For example, consider two Score plugins `DefaultScore1` and `DefaultScore2`, each with a weight of `1`. They can be reordered with different weights like so:

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: multipoint-scheduler
    plugins:
      score:
        enabled:
        - name: 'DefaultScore2'
          weight: 5
```

In this example, it's unnecessary to specify the plugins in `MultiPoint` explicitly because they are default plugins. And the only plugin specified in `Score` is `DefaultScore2`. This is because plugins set through specific extension points will always take precedence over `MultiPoint` plugins. So, this snippet essentially re-orders the two plugins without needing to specify both of them.

The general hierarchy for precedence when configuring `MultiPoint` plugins is as follows:

1. Specific extension points run first, and their settings override whatever is set elsewhere
2. Plugins manually configured through `MultiPoint` and their settings
3. Default plugins and their default settings

To demonstrate the above hierarchy, the following example is based on these plugins:

Plugin	Extension Points
`DefaultQueueSort`	`QueueSort`
`CustomQueueSort`	`QueueSort`
`DefaultPlugin1`	`Score`, `Filter`
`DefaultPlugin2`	`Score`
`CustomPlugin1`	`Score`, `Filter`
`CustomPlugin2`	`Score`, `Filter`

A valid sample configuration for these plugins would be:

```yaml
apiVersion: kubescheduler.config.k8s.io/v1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: multipoint-scheduler
    plugins:
      multiPoint:
        enabled:
        - name: 'CustomQueueSort'
        - name: 'CustomPlugin1'
          weight: 3
        - name: 'CustomPlugin2'
        disabled:
        - name: 'DefaultQueueSort'
      filter:
        disabled:
        - name: 'DefaultPlugin1'
      score:
        enabled:
        - name: 'DefaultPlugin2'
```

Note that there is no error for re-declaring a `MultiPoint` plugin in a specific extension point. The re-declaration is ignored (and logged), as specific extension points take precedence.

Besides keeping most of the config in one spot, this sample does a few things:

- Enables the custom `queueSort` plugin and disables the default one
- Enables `CustomPlugin1` and `CustomPlugin2`, which will run first for all of their extension points
- Disables `DefaultPlugin1`, but only for `filter`
- Reorders `DefaultPlugin2` to run first in `score` (even before the custom plugins)

In versions of the config before `v1beta3`, without `multiPoint`, the above snippet would equate to this:

```yaml
apiVersion: kubescheduler.config.k8s.io/v1beta2
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: multipoint-scheduler
    plugins:

      # Disable the default QueueSort plugin
      queueSort:
        enabled:
        - name: 'CustomQueueSort'
        disabled:
        - name: 'DefaultQueueSort'

      # Enable custom Filter plugins
      filter:
        enabled:
        - name: 'CustomPlugin1'
        - name: 'CustomPlugin2'
        - name: 'DefaultPlugin2'
        disabled:
        - name: 'DefaultPlugin1'

      # Enable and reorder custom score plugins
      score:
        enabled:
        - name: 'DefaultPlugin2'
          weight: 1
        - name: 'DefaultPlugin1'
          weight: 3
```

While this is a complicated example, it demonstrates the flexibility of `MultiPoint` config as well as its seamless integration with the existing methods for configuring extension points.

Scheduler configuration migrations

v1beta1 → v1beta2 | v1beta2 → v1beta3 | v1beta3 → v1

- With the v1beta2 configuration version, you can use a new score extension for the `NodeResourcesFit` plugin. The new extension combines the functionalities of the `NodeResourcesLeastAllocated`, `NodeResourcesMostAllocated` and `RequestedToCapacityRatio` plugins. For example, if you previously used the `NodeResourcesMostAllocated` plugin, you would instead use `NodeResourcesFit` (enabled by default) and add a `pluginConfig` with a `scoreStrategy` that is similar to:

  ```yaml
  apiVersion: kubescheduler.config.k8s.io/v1beta2
  kind: KubeSchedulerConfiguration
  profiles:
  - pluginConfig:
    - args:
        scoringStrategy:
          resources:
          - name: cpu
            weight: 1
          type: MostAllocated
      name: NodeResourcesFit
  ```

- The scheduler plugin `NodeLabel` is deprecated; instead, use the `NodeAffinity`

plugin (enabled by default) to achieve similar behavior.

- The scheduler plugin `ServiceAffinity` is deprecated; instead, use the `InterPodAffinity` plugin (enabled by default) to achieve similar behavior.
- The scheduler plugin `NodePreferAvoidPods` is deprecated; instead, use node taints to achieve similar behavior.
- A plugin enabled in a v1beta2 configuration file takes precedence over the default configuration for that plugin.
- Invalid `host` or `port` configured for scheduler healthz and metrics bind address will cause validation failure.

What's next

- Read the kube-scheduler reference
- Learn about scheduling
- Read the kube-scheduler configuration (v1) reference

16.2 - Scheduling Policies

In Kubernetes versions before v1.23, a scheduling policy can be used to specify the *predicates* and *priorities* process. For example, you can set a scheduling policy by running `kube-scheduler --policy-config-file <filename>` or `kube-scheduler --policy-configmap <ConfigMap>`.

This scheduling policy is not supported since Kubernetes v1.23. Associated flags `policy-config-file`, `policy-configmap`, `policy-configmap-namespace` and `use-legacy-policy-config` are also not supported. Instead, use the Scheduler Configuration to achieve similar behavior.

What's next

- Learn about scheduling
- Learn about kube-scheduler Configuration
- Read the kube-scheduler configuration reference (v1)

17 - Other Tools

Kubernetes contains several tools to help you work with the Kubernetes system.

crictl

`crictl` is a command-line interface for inspecting and debugging CRI-compatible container runtimes.

Dashboard

`Dashboard` , the web-based user interface of Kubernetes, allows you to deploy containerized applications to a Kubernetes cluster, troubleshoot them, and manage the cluster and its resources itself.

Helm

> ⃠ This item links to a third party project or product that is not part of Kubernetes itself. More information

Helm is a tool for managing packages of pre-configured Kubernetes resources. These packages are known as *Helm charts*.

Use Helm to:

- Find and use popular software packaged as Kubernetes charts
- Share your own applications as Kubernetes charts
- Create reproducible builds of your Kubernetes applications
- Intelligently manage your Kubernetes manifest files
- Manage releases of Helm packages

Kompose

`Kompose` is a tool to help Docker Compose users move to Kubernetes.

Use Kompose to:

- Translate a Docker Compose file into Kubernetes objects
- Go from local Docker development to managing your application via Kubernetes
- Convert v1 or v2 Docker Compose `yaml` files or Distributed Application Bundles

Kui

`Kui` is a GUI tool that takes your normal `kubectl` command line requests and responds with graphics.

Kui takes the normal `kubectl` command line requests and responds with graphics. Instead of ASCII tables, Kui provides a GUI rendering with tables that you can sort.

Kui lets you:

- Directly click on long, auto-generated resource names instead of copying and pasting
- Type in `kubectl` commands and see them execute, even sometimes faster than `kubectl` itself
- Query a Job and see its execution rendered as a waterfall diagram
- Click through resources in your cluster using a tabbed UI

Minikube

`minikube` is a tool that runs a single-node Kubernetes cluster locally on your workstation for development and testing purposes.

17.1 - Mapping from dockercli to crictl

Note:
This page is being directed to https://v1-24.docs.kubernetes.io/docs/reference/tools/map-crictl-dockercli/ because of the removal of dockershim from crictl in v1.24. As per our community policy, deprecated documents are not maintained beyond next three versions. The reason for deprecation is explained in Dockershim-FAQ.