

[Alauda DevOps Pipelines Docs](#) > [API Documentation](#) > [Kubernetes APIs](#) > [Operator](#)

Operator

[TektonConfig \[operator.tekton.dev\]](#) [TektonPipeline \[operator.tekton.dev\]](#) [TektonTrigger \[operator.tekton.dev\]](#)

[TektonChain \[operator.tekton.dev\]](#) [TektonHub \[operator.tekton.dev\]](#) [TektonResult \[operator.tekton.dev\]](#)

[TektonInstallerSet \[operator.tekton.dev\]](#) [OpenShift Pipelines as Code \[operator.tekton.dev\]](#) [TektonPruner \[operator.tekton.dev\]](#)

TektonConfig

[operator.tekton.dev/v1alpha1]

Description

TektonConfig is the Schema for the TektonConfigs API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	TektonConfigSpec defines the desired state of TektonConfig
<code>status</code>	<code>object</code>	TektonConfigStatus defines the observed state of TektonConfig

.spec

Description

TektonConfigSpec defines the desired state of TektonConfig

Type

`object`

Property	Type	Description
<code>addon</code>	<code>object</code>	Addon holds the addons config
<code>chain</code>	<code>object</code>	Chain holds the customizable option for chains component

Property	Type	Description
<code>config</code>	<code>object</code>	Config holds the configuration for resources created by TektonConfig
<code>dashboard</code>	<code>object</code>	Dashboard holds the customizable options for dashboards component
<code>hub</code>	<code>object</code>	Hub holds the hub config
<code>multiclusterProxyAAE</code>	<code>object</code>	MulticlusterProxyAAE holds the customizable options for the multicluster-proxy-aae component
<code>params</code>	<code>array</code>	Params is the list of params passed for all platforms
<code>pipeline</code>	<code>object</code>	Pipeline holds the customizable option for pipeline component
<code>platforms</code>	<code>object</code>	Platforms allows configuring platform specific configurations
<code>profile</code>	<code>string</code>	
<code>pruner</code>	<code>object</code>	Pruner holds the prune config

Property	Type	Description
<code>result</code>	<code>object</code>	Result holds the customize option for results component
<code>scheduler</code>	<code>object</code>	To enable Pipeline Scheduling on Single Cluster or Multiple Clusters
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed
<code>targetNamespaceMetadata</code>	<code>object</code>	holds target namespace metadata
<code>tektonpruner</code>	<code>object</code>	New EventBasedPruner which provides more granular control over TaskRun and PipelineRuns
<code>trigger</code>	<code>object</code>	Trigger holds the customizable option for triggers component

.spec.addon

Description

Addon holds the addons config

Type

`object`

Property	Type	Description
<code>enablePipelinesAsCode</code>	<code>boolean</code>	Deprecated, will be removed in further release EnablePAC field defines whether to install PAC
<code>params</code>	<code>array</code>	Params is the list of params passed for Addon customization

`.spec.addon.params`

Description

Params is the list of params passed for Addon customization

Type

`array`

`.spec.addon.params[]`

Description

Param declares an string value to use for the parameter called name.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	
<code>value</code>	<code>string</code>	

`.spec.chain`

Description

Chain holds the customizable option for chains component

Type

`object`

Required

`disabled`

Property	Type	Description
<code>artifacts.oci.format</code>	<code>string</code>	oci artifacts config
<code>artifacts.oci.signer</code>	<code>string</code>	
<code>artifacts.oci.storage</code>	<code>string</code>	
<code>artifacts.pipelinerun.enable-deep-inspection</code>	<code>string</code>	
<code>artifacts.pipelinerun.format</code>	<code>string</code>	pipelinerun artifacts config
<code>artifacts.pipelinerun.signer</code>	<code>string</code>	
<code>artifacts.pipelinerun.storage</code>	<code>string</code>	
<code>artifacts.taskrun.format</code>	<code>string</code>	taskrun artifacts config
<code>artifacts.taskrun.signer</code>	<code>string</code>	
<code>artifacts.taskrun.storage</code>	<code>string</code>	
<code>builddefinition.buildtype</code>	<code>string</code>	
<code>builder.id</code>	<code>string</code>	builder config

Property	Type	Description
<code>controllerEnvs</code>	<code>array</code>	
<code>disabled</code>	<code>boolean</code>	enable or disable chains feature
<code>generateSigningSecret</code>	<code>boolean</code>	generate signing key
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>performance</code>	<code>object</code>	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
<code>signers.kms.auth.address</code>	<code>string</code>	
<code>signers.kms.auth.oidc.path</code>	<code>string</code>	
<code>signers.kms.auth.oidc.role</code>	<code>string</code>	
<code>signers.kms.auth.spire.audience</code>	<code>string</code>	
<code>signers.kms.auth.spire.sock</code>	<code>string</code>	
<code>signers.kms.auth.token</code>	<code>string</code>	
<code>signers.kms.auth.token-path</code>	<code>string</code>	

Property	Type	Description
<code>signers.kms.kmsref</code>	<code>string</code>	kms signer config
<code>signers.x509.fulcio.address</code>	<code>string</code>	
<code>signers.x509.fulcio.enabled</code>	<code>boolean</code>	
<code>signers.x509.fulcio.issuer</code>	<code>string</code>	x509 signer config
<code>signers.x509.fulcio.provider</code>	<code>string</code>	
<code>signers.x509.identity.token.file</code>	<code>string</code>	
<code>signers.x509.tuf.mirror.url</code>	<code>string</code>	
<code>storage.docdb.mongo-server-url</code>	<code>string</code>	
<code>storage.docdb.mongo-server-url-dir</code>	<code>string</code>	storage configs
<code>storage.docdb.url</code>	<code>string</code>	
<code>storage.gcs.bucket</code>	<code>string</code>	
<code>storage.grafeas.notehint</code>	<code>string</code>	
<code>storage.grafeas.noteid</code>	<code>string</code>	
<code>storage.grafeas.projectid</code>	<code>string</code>	
<code>storage.oci.repository</code>	<code>string</code>	
<code>storage.oci.repository.insecure</code>	<code>boolean</code>	
<code>transparency.enabled</code>	<code>string</code>	

Property	Type	Description
<code>transparency.url</code>	<code>string</code>	

`.spec.chain.controllerEnvs`

Type

`array`

`.spec.chain.controllerEnvs[]`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. May consist of any printable ASCII characters except '='.
<code>value</code>	<code>string</code>	Variable references <code>\$(VAR_NAME)</code> are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code> , which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code> . Escaped references will never be

Property	Type	Description
		expanded, regardless of whether the variable exists or not. Defaults to "".
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.chain.controllerEnvs[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports metadata.name, metadata.namespace, <code>metadata.labels['<KEY>']</code> , <code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
<code>fileKeyRef</code>	<code>object</code>	FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Property	Type	Description
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.chain.controllerEnvs[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.chain.controllerEnvs[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

.spec.chain.controllerEnvs[].valueFrom.fileKeyRef

Description

FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Type

object

Required

key path volumeName

Property	Type	Description
key	string	The key within the env file. An invalid key will prevent the pod from starting. The keys defined within a source may consist of any printable ASCII characters except '='. During Alpha stage of the EnvFiles feature gate, the key size is limited to 128 characters.
optional	boolean	Specify whether the file or its key must be defined. If the file or key does not exist, then the env var is not published. If optional is set to true and the specified key does not exist, the environment variable will not be set in the Pod's containers. If optional is set to false and the specified key does not exist, an error will be returned during Pod creation.
path	string	The path within the volume from which to select the file. Must be relative and may not contain the '..' path or start with '..'.

Property	Type	Description
<code>volumeName</code>	<code>string</code>	The name of the volume mount containing the env file.

`.spec.chain.controllerEnvs[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>	<code>string</code>	Specifies the output format of the exposed resources, defaults to "1"
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.chain.controllerEnvs[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

object

Required

key

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

.spec.chain.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	

Property	Type	Description
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.chain.options.configMaps`

Type

`object`

`.spec.chain.options.deployments`

Type

`object`

`.spec.chain.options.horizontalPodAutoscalers`

Type

`object`

`.spec.chain.options.ingress`

Type

`object`

.spec.chain.options.statefulSets

Type

object

.spec.chain.options.webhookConfigurationOptions

Type

object

.spec.chain.performance

Description

PerformanceProperties defines the fields which are configurable to tune the performance of component controller

Type

object

Required

disable-ha

Property	Type	Description
buckets	integer	
disable-ha	boolean	if it is true, disables the HA feature
kube-api-burst	integer	
kube-api-qps	number	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5L87 defaults:

Property	Type	Description
		https://github.com/tektoncd/pipeline/blob/34618964300620dca4go/rest/config.go#L45-L46 ↗
<code>replicas</code>	<code>integer</code>	
<code>statefulset- ordinals</code>	<code>boolean</code>	if is true, enable StatefulsetOrdinals mode
<code>threads-per- controller</code>	<code>integer</code>	The number of workers to use when processing the component

.spec.config

Description

Config holds the configuration for resources created by TektonConfig

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

.spec.config.nodeSelector

Type

object

.spec.config.tolerations

Type

array

.spec.config.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric

Property	Type	Description
		comparisons (requires feature gate TaintTolerationComparisonOperators).
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.dashboard

Description

Dashboard holds the customizable options for dashboards component

Type

`object`

Required

`readonly`

Property	Type	Description
<code>external-logs</code>	<code>string</code>	

Property	Type	Description
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>readonly</code>	<code>boolean</code>	Readonly when set to true configures the Tekton dashboard in read-only mode

`.spec.dashboard.options`

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.dashboard.options.configMaps`

Type`object`**.spec.dashboard.options.deployments****Type**`object`**.spec.dashboard.options.horizontalPodAutoscalers****Type**`object`**.spec.dashboard.options.ingress****Type**`object`**.spec.dashboard.options.statefulSets****Type**`object`**.spec.dashboard.options.webhookConfigurationOptions****Type**`object`**.spec.hub****Description**

Hub holds the hub config

Type

object

Property	Type	Description
<code>options</code>	object	options holds additions fields and these fields will be updated on the manifests
<code>params</code>	array	Params is the list of params passed for Hub customization

.spec.hub.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
<code>configMaps</code>	object	
<code>deployments</code>	object	
<code>disabled</code>	boolean	
<code>horizontalPodAutoscalers</code>	object	
<code>ingress</code>	object	
<code>statefulSets</code>	object	
<code>webhookConfigurationOptions</code>	object	

.spec.hub.options.configMaps

Type

object

.spec.hub.options.deployments

Type

object

.spec.hub.options.horizontalPodAutoscalers

Type

object

.spec.hub.options.ingress

Type

object

.spec.hub.options.statefulSets

Type

object

.spec.hub.options.webhookConfigurationOptions

Type

object

.spec.hub.params

Description

Params is the list of params passed for Hub customization

Type

array

.spec.hub.params[]

Description

Param declares an string value to use for the parameter called name.

Type

object

Property	Type	Description
name	string	
value	string	

.spec.multiclusterProxyAAE

Description

MulticlusterProxyAAE holds the customizable options for the multicluster-proxy-aae component

Type

object

Property	Type	Description
options	object	options holds additional fields and these fields will be updated on the manifests

.spec.multiclusterProxyAAE.options

Description

options holds additional fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.multiclusterProxyAAE.options.configMaps

Type

object

.spec.multiclusterProxyAAE.options.deployments

Type

object

.spec.multiclusterProxyAAE.options.horizontalPodAutoscalers

Type

object

.spec.multiclusterProxyAAE.options.ingress

Type

object

.spec.multiclusterProxyAAE.options.statefulSets

Type

object

.spec.multiclusterProxyAAE.options.webhookConfigurationOptions

Type

object

.spec.params

Description

Params is the list of params passed for all platforms

Type

array

.spec.params[]

Description

Param declares an string value to use for the parameter called name.

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	
<code>value</code>	<code>string</code>	

.spec.pipeline

Description

Pipeline holds the customizable option for pipeline component

Type

`object`

Property	Type	Description
<code>await-sidecar-readiness</code>	<code>boolean</code>	
<code>bundles-resolver-config</code>	<code>object</code>	
<code>cluster-resolver-config</code>	<code>object</code>	
<code>coschedule</code>	<code>string</code>	
<code>default-affinity-assistant-pod-template</code>	<code>string</code>	
<code>default-cloud-events-sink</code>	<code>string</code>	
<code>default-forbidden-env</code>	<code>string</code>	
<code>default-managed-by-label-value</code>	<code>string</code>	
<code>default-max-matrix-combinations-count</code>	<code>string</code>	

Property	Type	Description
default-pod-template	string	
default-resolver-type	string	
default-service-account	string	
default-task-run-workspace-binding	string	
default-timeout-minutes	integer	
disable-affinity-assistant	boolean	Deprecated: DisableAffinityAssistant is deprecated and no longer used. This field is removed from pipeline component. Keeping here to maintain API compatibility during upgrades.
disable-creds-init	boolean	
disable-inline-spec	string	
embedded-status	string	
enable-api-fields	string	
enable-bundles-resolver	boolean	
enable-cel-in-whenexpression	boolean	
enable-cluster-resolver	boolean	
enable-custom-tasks	boolean	
enable-git-resolver	boolean	
enable-hub-resolver	boolean	

Property	Type	Description
<code>enable-param-enum</code>	<code>boolean</code>	
<code>enable-provenance-in-status</code>	<code>boolean</code>	
<code>enable-step-actions</code>	<code>boolean</code>	
<code>enable-tekton-oci-bundles</code>	<code>boolean</code>	not in use, see: https://github.com/tektoncd/pipeline/pull/778 ↗ this field is removed from pipeline component keeping here to maintain the AP compatibility
<code>enforce-nonfalsifiability</code>	<code>string</code>	
<code>git-resolver-config</code>	<code>object</code>	
<code>hub-resolver-config</code>	<code>object</code>	
<code>keep-pod-on-cancel</code>	<code>boolean</code>	
<code>max-result-size</code>	<code>integer</code>	
<code>metrics.count.enable-reason</code>	<code>boolean</code>	
<code>metrics.pipelinerun.duration-type</code>	<code>string</code>	
<code>metrics.pipelinerun.level</code>	<code>string</code>	
<code>metrics.taskrun.duration-type</code>	<code>string</code>	
<code>metrics.taskrun.level</code>	<code>string</code>	

Property	Type	Description
<code>options</code>	<code>object</code>	options holds additions fields and these field will be updated on the manifests
<code>params</code>	<code>array</code>	The params to customize different components of Pipelines
<code>performance</code>	<code>object</code>	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
<code>require-git-ssh-secret-known-hosts</code>	<code>boolean</code>	
<code>results-from</code>	<code>string</code>	
<code>running-in-environment-with-injected-sidecars</code>	<code>boolean</code>	
<code>scope-when-expressions-to-task</code>	<code>boolean</code>	ScopeWhenExpressionsToTask is deprecated and never used.
<code>send-cloudevents-for-runs</code>	<code>boolean</code>	Deprecated: send-cloudevents-for-runs is deprecated in tektoncd/pipeline v1.12.0 (https://github.com/tektoncd/pipeline/pull/97) and will be removed in a future release. CloudEvents for CustomRuns are now enabled by default when a sink is configured in the config-events ConfigMap. This field only affects CustomRun objects; it has no effect on TaskRuns or PipelineRuns. Set to

Property	Type	Description
		false only to suppress duplicate events whe a custom task controller already sends its own CloudEvents.
<code>set-security-context</code>	<code>boolean</code>	
<code>traces.credentialsSecret</code>	<code>string</code>	CredentialsSecret is the name of the secret containing credentials for the tracing endpoi
<code>traces.enabled</code>	<code>boolean</code>	Enabled controls whether tracing is enabled or not
<code>traces.endpoint</code>	<code>string</code>	Endpoint is the URL for the OpenTelemetry trace collector
<code>trusted-resources-verification-no-match-policy</code>	<code>string</code>	
<code>verification-mode</code>	<code>string</code>	

.spec.pipeline.bundles-resolver-config

Type

`object`

.spec.pipeline.cluster-resolver-config

Type

`object`

.spec.pipeline.git-resolver-config

Type

object

.spec.pipeline.hub-resolver-config

Type

object

.spec.pipeline.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.pipeline.options.configMaps

Type

object

.spec.pipeline.options.deployments

Type

object

.spec.pipeline.options.horizontalPodAutoscalers

Type

object

.spec.pipeline.options.ingress

Type

object

.spec.pipeline.options.statefulSets

Type

object

.spec.pipeline.options.webhookConfigurationOptions

Type

object

.spec.pipeline.params

Description

The params to customize different components of Pipelines

Type

array

.spec.pipeline.params[]

Description

Param declares an string value to use for the parameter called name.

Type

object

Property	Type	Description
name	string	
value	string	

.spec.pipeline.performance

Description

PerformanceProperties defines the fields which are configurable to tune the performance of component controller

Type

object

Required

disable-ha

Property	Type	Description
buckets	integer	
disable-ha	boolean	if it is true, disables the HA feature

Property	Type	Description
<code>kube-api-burst</code>	<code>integer</code>	
<code>kube-api-qps</code>	<code>number</code>	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5L87 ↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4go/rest/config.go#L45-L46 ↗
<code>replicas</code>	<code>integer</code>	
<code>statefulset-ordinals</code>	<code>boolean</code>	if is true, enable StatefulsetOrdinals mode
<code>threads-per-controller</code>	<code>integer</code>	The number of workers to use when processing the component

.spec.platforms

Description

Platforms allows configuring platform specific configurations

Type

`object`

Property	Type	Description
<code>kubernetes</code>	<code>object</code>	Kubernetes allows configuring kubernetes specific components and configurations

Property	Type	Description
<code>openshift</code>	<code>object</code>	OpenShift allows configuring openshift specific components and configurations

`.spec.platforms.kubernetes`

Description

Kubernetes allows configuring kubernetes specific components and configurations

Type

`object`

Property	Type	Description
<code>pipelinesAsCode</code>	<code>object</code>	PipelinesAsCode allows configuring PipelinesAsCode configurations

`.spec.platforms.kubernetes.pipelinesAsCode`

Description

PipelinesAsCode allows configuring PipelinesAsCode configurations

Type

`object`

Property	Type	Description
<code>additionalPACControllers</code>	<code>object</code>	AdditionalPACControllers allows to deploy additional PAC controller

Property	Type	Description
<code>enable</code>	<code>boolean</code>	Enable or disable pipelines as code by changing this bool
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>settings</code>	<code>object</code>	

`.spec.platforms.kubernetes.pipelinesAsCode.additionalPACControllers`

Description

AdditionalPACControllers allows to deploy additional PAC controller

Type

`object`

`.spec.platforms.kubernetes.pipelinesAsCode.options`

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	
<code>deployments</code>	<code>object</code>	

Property	Type	Description
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.platforms.kubernetes.pipelinesAsCode.options.configMaps`

Type

`object`

`.spec.platforms.kubernetes.pipelinesAsCode.options.deployments`

Type

`object`

`.spec.platforms.kubernetes.pipelinesAsCode.options.horizontalPodAutoscalers`

Type

`object`

`.spec.platforms.kubernetes.pipelinesAsCode.options.ingress`

Type

object

`.spec.platforms.kubernetes.pipelinesAsCode.options.stat efulSets`

Type

object

`.spec.platforms.kubernetes.pipelinesAsCode.options.web hookConfigurationOptions`

Type

object

`.spec.platforms.kubernetes.pipelinesAsCode.settings`

Type

object

`.spec.platforms.openshift`

Description

OpenShift allows configuring openshift specific components and configurations

Type

object

Property	Type	Description
<code>enableCentralTLSConfig</code>	boolean	EnableCentralTLSConfig controls TLS configuration inheritance from the cluster's APIServer TLS security profile. When enabled

Property	Type	Description
		(the default), TLS settings (minimum version, cipher suites, curve preferences) are automatically derived from the cluster-wide security policy and injected into Tekton component containers that support TLS configuration. Set to false to opt out and manage TLS settings manually. Default: true (opt-out)
<code>pipelinesAsCode</code>	<code>object</code>	PipelinesAsCode allows configuring PipelinesAsCode configurations
<code>scc</code>	<code>object</code>	SCC allows configuring security context constraints used by workloads

`.spec.platforms.openshift.pipelinesAsCode`

Description

PipelinesAsCode allows configuring PipelinesAsCode configurations

Type

`object`

Property	Type	Description
<code>additionalPACControllers</code>	<code>object</code>	AdditionalPACControllers allows to deploy additional PAC controller

Property	Type	Description
<code>enable</code>	<code>boolean</code>	Enable or disable pipelines as code by changing this bool
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>settings</code>	<code>object</code>	

`.spec.platforms.openshift.pipelinesAsCode.additionalPACControllers`

Description

AdditionalPACControllers allows to deploy additional PAC controller

Type

`object`

`.spec.platforms.openshift.pipelinesAsCode.options`

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	
<code>deployments</code>	<code>object</code>	

Property	Type	Description
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.platforms.openshift.pipelinesAsCode.options.config` Maps

Type

`object`

`.spec.platforms.openshift.pipelinesAsCode.options.deployments`

Type

`object`

`.spec.platforms.openshift.pipelinesAsCode.options.horizontalPodAutoscalers`

Type

`object`

`.spec.platforms.openshift.pipelinesAsCode.options.ingresses`

Type

object

.spec.platforms.openshift.pipelinesAsCode.options.statefulSets

Type

object

.spec.platforms.openshift.pipelinesAsCode.options.webhookConfigurationOptions

Type

object

.spec.platforms.openshift.pipelinesAsCode.settings

Type

object

.spec.platforms.openshift.scc

Description

SCC allows configuring security context constraints used by workloads

Type

object

Property	Type	Description
default	string	Default contains the default SCC that will be attached to the service account used for workloads (<code>pipeline</code> SA by default) and def

Property	Type	Description
		in PipelineProperties.OptionalPipelineProperties.DefaultServiceAc
maxAllowed	string	MaxAllowed specifies the highest SCC that can be requested fo namespace or in the Default field.

.spec.pruner

Description

Pruner holds the prune config

Type

object

Required

disabled

Property	Type	Description
disabled	boolean	enable or disable pruner feature
keep	integer	The number of resource to keep You dont want to delete all the pipelinerun/taskrun's by a cron
keep-since	integer	KeepSince keeps the resources younger than the specified value Its value is taken in minutes

Property	Type	Description
<code>prune-per-resource</code>	<code>boolean</code>	apply the prune job to the individual resources
<code>resources</code>	<code>array</code>	The resources which need to be pruned
<code>schedule</code>	<code>string</code>	How frequent pruning should happen
<code>startingDeadlineSeconds</code>	<code>integer</code>	Optional deadline in seconds for starting the job if it misses scheduled time for any reason. Missed jobs executions will be counted as failed ones.

`.spec.pruner.resources`

Description

The resources which need to be pruned

Type

`array`

`.spec.pruner.resources[]`

Type

`string`

`.spec.result`

Description

Result holds the customize option for results component

Type

object

Required

disabled

is_external_db

Property	Type	Description
auth_disable	boolean	
auth_impersonate	boolean	
db_enable_auto_migration	boolean	
db_host	string	
db_name	string	
db_port	integer	
db_secret_name	string	
db_secret_password_key	string	
db_secret_user_key	string	
db_sslmode	string	
db_sslrootcert	string	
disabled	boolean	enable or disable Result Component
gcs_bucket_name	string	
gcs_creds_secret_key	string	

Property	Type	Description
<code>gcs_creds_secret_name</code>	<code>string</code>	
<code>is_external_db</code>	<code>boolean</code>	
<code>log_level</code>	<code>string</code>	
<code>logging_plugin_api_url</code>	<code>string</code>	
<code>logging_plugin_ca_cert</code>	<code>string</code>	
<code>logging_plugin_forwarder_delay_duration</code>	<code>integer</code>	
<code>logging_plugin_multipart_regex</code>	<code>string</code>	
<code>logging_plugin_namespace_key</code>	<code>string</code>	
<code>logging_plugin_proxy_path</code>	<code>string</code>	
<code>logging_plugin_query_limit</code>	<code>integer</code>	
<code>logging_plugin_query_params</code>	<code>string</code>	
<code>logging_plugin_static_labels</code>	<code>string</code>	
<code>logging_plugin_tls_verification_disable</code>	<code>boolean</code>	
<code>logging_plugin_token_path</code>	<code>string</code>	
<code>logging_pvc_name</code>	<code>string</code>	
<code>logs_api</code>	<code>boolean</code>	
<code>logs_buffer_size</code>	<code>integer</code>	
<code>logs_path</code>	<code>string</code>	
<code>logs_type</code>	<code>string</code>	
<code>loki_stack_name</code>	<code>string</code>	

Property	Type	Description
loki_stack_namespace	string	
options	object	Options holds additions fields and these fields will be updated on the manifests
performance	object	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
prometheus_histogram	boolean	
prometheus_port	integer	
route_enabled	boolean	Route configuration for Results API service exposure
route_host	string	
route_path	string	
route_tls_termination	string	
secret_name	string	name of the secret used to get S3 credentials and pass it as environment variables to the "tekton-

Property	Type	Description
		results-api" deployment under "api" container
server_port	integer	
storage_emulator_host	string	
tls_hostname_override	string	

.spec.result.options

Description

Options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.result.options.configMaps

Type

object

.spec.result.options.deployments

Type

object

.spec.result.options.horizontalPodAutoscalers

Type

object

.spec.result.options.ingress

Type

object

.spec.result.options.statefulSets

Type

object

.spec.result.options.webhookConfigurationOptions

Type

object

.spec.result.performance

Description

PerformanceProperties defines the fields which are configurable to tune the performance of component controller

Type

`object`

Required

`disable-ha`

Property	Type	Description
<code>buckets</code>	<code>integer</code>	
<code>disable-ha</code>	<code>boolean</code>	if it is true, disables the HA feature
<code>kube-api-burst</code>	<code>integer</code>	
<code>kube-api-qps</code>	<code>number</code>	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5L87 ↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4go/rest/config.go#L45-L46 ↗
<code>replicas</code>	<code>integer</code>	
<code>statefulset-ordinals</code>	<code>boolean</code>	if is true, enable StatefulsetOrdinals mode
<code>threads-per-controller</code>	<code>integer</code>	The number of workers to use when processing the component

.spec.scheduler

Description

To enable Pipeline Scheduling on Single Cluster or Multiple Clusters

Type

`object`

Required

`config.yaml``disabled``multi-cluster-disabled``multi-cluster-role`

Property	Type	Description
<code>config.yaml</code>	<code>object</code>	This hold the config data from tekton-kueue. ConfigMap in tekton kueue is loaded as config.yaml so we need to match the key here
<code>disabled</code>	<code>boolean</code>	enable or disable TektonScheduler Component
<code>multi-cluster-disabled</code>	<code>boolean</code>	
<code>multi-cluster-role</code>	<code>string</code>	MultiClusterRole Define the role of current cluster in multi-cluster environment. The MultiClusterRole can be one of Hub or Spoke
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests

.spec.scheduler.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.scheduler.options.configMaps`

Type

`object`

`.spec.scheduler.options.deployments`

Type

`object`

`.spec.scheduler.options.horizontalPodAutoscalers`

Type

`object`

`.spec.scheduler.options.ingress`

Type

`object`

.spec.scheduler.options.statefulSets

Type

object

.spec.scheduler.options.webhookConfigurationOptions

Type

object

.spec.targetNamespaceMetadata

Description

holds target namespace metadata

Type

object

Property	Type	Description
annotations	object	
labels	object	

.spec.targetNamespaceMetadata.annotations

Type

object

.spec.targetNamespaceMetadata.labels

Type

object

.spec.tektonpruner

Description

New EventBasedPruner which provides more granular control over TaskRun and PipelineRuns

Type

object

Required

disabled

global-config

Property	Type	Description
<code>disabled</code>	<code>boolean</code>	enable or disable TektonPruner Component
<code>global-config</code>	<code>object</code>	
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests

.spec.tektonpruner.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
<code>configMaps</code>	<code>object</code>	

Property	Type	Description
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.tektonpruner.options.configMaps`

Type

`object`

`.spec.tektonpruner.options.deployments`

Type

`object`

`.spec.tektonpruner.options.horizontalPodAutoscalers`

Type

`object`

`.spec.tektonpruner.options.ingress`

Type

`object`

`.spec.tektonpruner.options.statefulSets`

Type

`object`

`.spec.tektonpruner.options.webhookConfigurationOptions`

Type

`object`

`.spec.trigger`

Description

Trigger holds the customizable option for triggers component

Type

`object`

Required

`disabled`

Property	Type	Description
<code>default-service-account</code>	<code>string</code>	
<code>disabled</code>	<code>boolean</code>	enable or disable Trigger Component
<code>enable-api-fields</code>	<code>string</code>	
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests

.spec.trigger.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.trigger.options.configMaps

Type

object

.spec.trigger.options.deployments

Type

object

.spec.trigger.options.horizontalPodAutoscalers

Type

object

.spec.trigger.options.ingress

Type

object

.spec.trigger.options.statefulSets

Type

object

.spec.trigger.options.webhookConfigurationOptions

Type

object

.status

Description

TektonConfigStatus defines the observed state of TektonConfig

Type

object

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>profile</code>	<code>string</code>	The profile installed
<code>tektonInstallerSets</code>	<code>object</code>	The current installer set name
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).

Property	Type	Description
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

`.status.tektonInstallerSets`

Description

The current installer set name

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonconfigs`
 - **DELETE** : delete collection of TektonConfig
 - **GET** : list objects of kind TektonConfig
 - **POST** : create a new TektonConfig
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonconfigs/{name}`
 - **DELETE** : delete the specified TektonConfig
 - **GET** : read the specified TektonConfig
 - **PATCH** : partially update the specified TektonConfig
 - **PUT** : replace the specified TektonConfig
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonconfigs/{name}/status`
 - **GET** : read status of the specified TektonConfig
 - **PATCH** : partially update status of the specified TektonConfig
 - **PUT** : replace status of the specified TektonConfig

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonconfigs`

HTTP method

DELETE

Description

delete collection of TektonConfig

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonConfig

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfigList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonConfig

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonConfig</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code> schema
201 - Created	<code>TektonConfig</code> schema
202 - Accepted	<code>TektonConfig</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonconfigs/{name}

HTTP method

DELETE

Description

delete the specified TektonConfig

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified TektonConfig

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code>  schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonConfig

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `TektonConfig`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonConfig</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code> schema

HTTP code	Response body
201 - Created	<code>TektonConfig</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonconfigs/{name}/status

HTTP method

`GET`

Description

read status of the specified TektonConfig

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonConfig

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonConfig

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonConfig</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonConfig</code> schema
201 - Created	<code>TektonConfig</code> schema
401 - Unauthorized	Empty

TektonPipeline

[operator.tekton.dev/v1alpha1]

Description

TektonPipeline is the Schema for the tektonpipelines API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	ObjectMeta ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	TektonPipelineSpec defines the desired state of TektonPipeline
<code>status</code>	<code>object</code>	TektonPipelineStatus defines the observed state of TektonPipeline

.spec

Description

TektonPipelineSpec defines the desired state of TektonPipeline

Type

`object`

Property	Type	Description
<code>await-sidecar-readiness</code>	<code>boolean</code>	
<code>bundles-resolver-config</code>	<code>object</code>	
<code>cluster-resolver-config</code>	<code>object</code>	

Property	Type	Description
<code>config</code>	<code>object</code>	Config holds the configuration for resources created by TektonPipeline
<code>coschedule</code>	<code>string</code>	
<code>default-affinity-assistant-pod-template</code>	<code>string</code>	
<code>default-cloud-events-sink</code>	<code>string</code>	
<code>default-forbidden-env</code>	<code>string</code>	
<code>default-managed-by-label-value</code>	<code>string</code>	
<code>default-max-matrix-combinations-count</code>	<code>string</code>	
<code>default-pod-template</code>	<code>string</code>	
<code>default-resolver-type</code>	<code>string</code>	
<code>default-service-account</code>	<code>string</code>	
<code>default-task-run-workspace-binding</code>	<code>string</code>	
<code>default-timeout-minutes</code>	<code>integer</code>	
<code>disable-affinity-assistant</code>	<code>boolean</code>	Deprecated: DisableAffinityAssistant is deprecated and no longer used. This field is removed from pipeline component. Keeping here to maintain API compatibility during upgrades.

Property	Type	Description
<code>disable-creds-init</code>	boolean	
<code>disable-inline-spec</code>	string	
<code>embedded-status</code>	string	
<code>enable-api-fields</code>	string	
<code>enable-bundles-resolver</code>	boolean	
<code>enable-cel-in-whenexpression</code>	boolean	
<code>enable-cluster-resolver</code>	boolean	
<code>enable-custom-tasks</code>	boolean	
<code>enable-git-resolver</code>	boolean	
<code>enable-hub-resolver</code>	boolean	
<code>enable-param-enum</code>	boolean	
<code>enable-provenance-in-status</code>	boolean	
<code>enable-step-actions</code>	boolean	
<code>enable-tekton-oci-bundles</code>	boolean	not in use, see: https://github.com/tektoncd/pipeline/pull/778 ↗ this field is removed from pipeline component keeping here to maintain the AP compatibility
<code>enforce-nonfalsifiability</code>	string	
<code>git-resolver-config</code>	object	
<code>hub-resolver-config</code>	object	

Property	Type	Description
<code>keep-pod-on-cancel</code>	<code>boolean</code>	
<code>max-result-size</code>	<code>integer</code>	
<code>metrics.count.enable-reason</code>	<code>boolean</code>	
<code>metrics.pipelinerun.duration-type</code>	<code>string</code>	
<code>metrics.pipelinerun.level</code>	<code>string</code>	
<code>metrics.taskrun.duration-type</code>	<code>string</code>	
<code>metrics.taskrun.level</code>	<code>string</code>	
<code>options</code>	<code>object</code>	options holds additions fields and these field will be updated on the manifests
<code>params</code>	<code>array</code>	The params to customize different components of Pipelines
<code>performance</code>	<code>object</code>	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
<code>require-git-ssh-secret-known-hosts</code>	<code>boolean</code>	
<code>results-from</code>	<code>string</code>	
<code>running-in-environment-with-injected-sidecars</code>	<code>boolean</code>	

Property	Type	Description
<code>scope-when-expressions-to-task</code>	<code>boolean</code>	ScopeWhenExpressionsToTask is deprecated and never used.
<code>send-cloudevents-for-runs</code>	<code>boolean</code>	Deprecated: <code>send-cloudevents-for-runs</code> is deprecated in tektoncd/pipeline v1.12.0 (https://github.com/tektoncd/pipeline/pull/97) and will be removed in a future release. CloudEvents for CustomRuns are now enabled by default when a sink is configured in the config-events ConfigMap. This field only affects CustomRun objects; it has no effect on TaskRuns or PipelineRuns. Set to false only to suppress duplicate events when a custom task controller already sends its own CloudEvents.
<code>set-security-context</code>	<code>boolean</code>	
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed
<code>traces.credentialsSecret</code>	<code>string</code>	CredentialsSecret is the name of the secret containing credentials for the tracing endpoint
<code>traces.enabled</code>	<code>boolean</code>	Enabled controls whether tracing is enabled or not

Property	Type	Description
<code>traces.endpoint</code>	<code>string</code>	Endpoint is the URL for the OpenTelemetry trace collector
<code>trusted-resources-verification-no-match-policy</code>	<code>string</code>	
<code>verification-mode</code>	<code>string</code>	

`.spec.bundles-resolver-config`

Type

`object`

`.spec.cluster-resolver-config`

Type

`object`

`.spec.config`

Description

Config holds the configuration for resources created by TektonPipeline

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	

Property	Type	Description
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

`.spec.config.nodeSelector`

Type

`object`

`.spec.config.tolerations`

Type

`array`

`.spec.config.tolerations[]`

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

`object`

Property	Type	Description
<code>effect</code>	<code>string</code>	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.

Property	Type	Description
<code>key</code>	<code>string</code>	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric comparisons (requires feature gate TaintTolerationComparisonOperators).
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.git-resolver-config`

Type

`object`

.spec.hub-resolver-config

Type

object

.spec.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.options.configMaps

Type

object

.spec.options.deployments

Type`object`**.spec.options.horizontalPodAutoscalers****Type**`object`**.spec.options.ingress****Type**`object`**.spec.options.statefulSets****Type**`object`**.spec.options.webhookConfigurationOptions****Type**`object`**.spec.params****Description**

The params to customize different components of Pipelines

Type`array`**.spec.params[]**

Description

Param declares an string value to use for the parameter called name.

Type

object

Property	Type	Description
name	string	
value	string	

.spec.performance

Description

PerformanceProperties defines the fields which are configurable to tune the performance of component controller

Type

object

Required

disable-ha

Property	Type	Description
buckets	integer	
disable-ha	boolean	if it is true, disables the HA feature
kube-api-burst	integer	
kube-api-qps	number	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5L87 defaults:

Property	Type	Description
		https://github.com/tektoncd/pipeline/blob/34618964300620dca4go/rest/config.go#L45-L46 ↗
<code>replicas</code>	<code>integer</code>	
<code>statefulset- ordinals</code>	<code>boolean</code>	if is true, enable StatefulsetOrdinals mode
<code>threads-per- controller</code>	<code>integer</code>	The number of workers to use when processing the component

.status

Description

TektonPipelineStatus defines the observed state of TektonPipeline

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.

Property	Type	Description
<code>extTektonInstallerSets</code>	<code>object</code>	The installer sets created for extension components
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>tektonInstallerSet</code>	<code>string</code>	The current installer set name for TektonPipeline
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

`.status.conditions[]`

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

`object`

Required

`status``type`

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.

Property	Type	Description
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

.status.extTektonInstallerSets

Description

The installer sets created for extension components

Type

`object`

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpipelines`
 - `DELETE` : delete collection of TektonPipeline
 - `GET` : list objects of kind TektonPipeline
 - `POST` : create a new TektonPipeline
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpipelines/{name}`
 - `DELETE` : delete the specified TektonPipeline
 - `GET` : read the specified TektonPipeline
 - `PATCH` : partially update the specified TektonPipeline
 - `PUT` : replace the specified TektonPipeline

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpipelines/{name}/status`
 - `GET` : read status of the specified TektonPipeline
 - `PATCH` : partially update status of the specified TektonPipeline
 - `PUT` : replace status of the specified TektonPipeline

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpipelines`

HTTP method

`DELETE`

Description

delete collection of TektonPipeline

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind TektonPipeline

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipelineList</code>  schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonPipeline

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonPipeline</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipeline</code> schema
201 - Created	<code>TektonPipeline</code> schema
202 - Accepted	<code>TektonPipeline</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpipelines/{name}

HTTP method

`DELETE`

Description

delete the specified TektonPipeline

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified TektonPipeline

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipeline</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified TektonPipeline

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipeline</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified TektonPipeline

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	TektonPipeline schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	TektonPipeline schema
201 - Created	TektonPipeline schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpipelines/{name}/status

HTTP method

GET

Description

read status of the specified TektonPipeline

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipeline</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonPipeline

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipeline</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified TektonPipeline

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonPipeline</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPipeline</code> schema
201 - Created	<code>TektonPipeline</code> schema
401 - Unauthorized	Empty

TektonTrigger

[operator.tekton.dev/v1alpha1]

Description

TektonTrigger is the Schema for the tektontriggers API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	TektonTriggerSpec defines the desired state of TektonTrigger
<code>status</code>	<code>object</code>	TektonTriggerStatus defines the observed state of TektonTrigger

.spec

Description

TektonTriggerSpec defines the desired state of TektonTrigger

Type

`object`

Required

`disabled`

Property	Type	Description
<code>config</code>	<code>object</code>	Config holds the configuration for resources created by TektonTrigger

Property	Type	Description
<code>default-service-account</code>	<code>string</code>	
<code>disabled</code>	<code>boolean</code>	enable or disable Trigger Component
<code>enable-api-fields</code>	<code>string</code>	
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed

.spec.config

Description

Config holds the configuration for resources created by TektonTrigger

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

.spec.config.nodeSelector

Type

object

.spec.config.tolerations

Type

array

.spec.config.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.

Property	Type	Description
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric comparisons (requires feature gate TaintTolerationComparisonOperators).
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	

Property	Type	Description
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.options.configMaps`

Type

`object`

`.spec.options.deployments`

Type

`object`

`.spec.options.horizontalPodAutoscalers`

Type

`object`

`.spec.options.ingress`

Type

`object`

.spec.options.statefulSets

Type

object

.spec.options.webhookConfigurationOptions

Type

object

.status

Description

TektonTriggerStatus defines the observed state of TektonTrigger

Type

object

Property	Type	Description
annotations	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
conditions	array	Conditions the latest available observations of a resource's current state.

Property	Type	Description
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>tektonInstallerSet</code>	<code>string</code>	The current installer set name
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektontriggers`
 - `DELETE` : delete collection of TektonTrigger
 - `GET` : list objects of kind TektonTrigger
 - `POST` : create a new TektonTrigger
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektontriggers/{name}`
 - `DELETE` : delete the specified TektonTrigger
 - `GET` : read the specified TektonTrigger
 - `PATCH` : partially update the specified TektonTrigger
 - `PUT` : replace the specified TektonTrigger
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektontriggers/{name}/status`
 - `GET` : read status of the specified TektonTrigger
 - `PATCH` : partially update status of the specified TektonTrigger
 - `PUT` : replace status of the specified TektonTrigger

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektontriggers`

HTTP method

DELETE

Description

delete collection of TektonTrigger

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonTrigger

HTTP responses

HTTP code	Response body
200 - OK	TektonTriggerList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonTrigger

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonTrigger</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
201 - Created	<code>TektonTrigger</code> schema
202 - Accepted	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektontriggers/{name}

HTTP method

DELETE

Description

delete the specified TektonTrigger

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified TektonTrigger

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonTrigger

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified TektonTrigger

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonTrigger</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
201 - Created	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektontriggers/{name}/status`

HTTP method

GET

Description

read status of the specified TektonTrigger

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified TektonTrigger

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified TektonTrigger

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonTrigger</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonTrigger</code> schema
201 - Created	<code>TektonTrigger</code> schema
401 - Unauthorized	Empty

TektonChain [operator.tekton.dev/v1alpha1]

Description

TektonChain is the Schema for the tektonchain API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	TektonChainSpec defines the desired state of TektonChain
<code>status</code>	<code>object</code>	TektonChainStatus defines the observed state of TektonChain

.spec

Description

TektonChainSpec defines the desired state of TektonChain

Type

`object`

Required

`disabled`

Property	Type	Description
<code>artifacts.oci.format</code>	<code>string</code>	oci artifacts config
<code>artifacts.oci.signer</code>	<code>string</code>	
<code>artifacts.oci.storage</code>	<code>string</code>	
<code>artifacts.pipelinerun.enable-deep-inspection</code>	<code>string</code>	
<code>artifacts.pipelinerun.format</code>	<code>string</code>	pipelinerun artifacts config
<code>artifacts.pipelinerun.signer</code>	<code>string</code>	
<code>artifacts.pipelinerun.storage</code>	<code>string</code>	
<code>artifacts.taskrun.format</code>	<code>string</code>	taskrun artifacts config
<code>artifacts.taskrun.signer</code>	<code>string</code>	
<code>artifacts.taskrun.storage</code>	<code>string</code>	
<code>builddefinition.buildtype</code>	<code>string</code>	
<code>builder.id</code>	<code>string</code>	builder config
<code>config</code>	<code>object</code>	Config holds the configuration for resources created by TektonChain
<code>controllerEnvs</code>	<code>array</code>	

Property	Type	Description
<code>disabled</code>	<code>boolean</code>	enable or disable chains feature
<code>generateSigningSecret</code>	<code>boolean</code>	generate signing key
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>performance</code>	<code>object</code>	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
<code>signers.kms.auth.address</code>	<code>string</code>	
<code>signers.kms.auth.oidc.path</code>	<code>string</code>	
<code>signers.kms.auth.oidc.role</code>	<code>string</code>	
<code>signers.kms.auth.spire.audience</code>	<code>string</code>	
<code>signers.kms.auth.spire.sock</code>	<code>string</code>	
<code>signers.kms.auth.token</code>	<code>string</code>	
<code>signers.kms.auth.token-path</code>	<code>string</code>	
<code>signers.kms.kmsref</code>	<code>string</code>	kms signer config

Property	Type	Description
<code>signers.x509.fulcio.address</code>	<code>string</code>	
<code>signers.x509.fulcio.enabled</code>	<code>boolean</code>	x509 signer config
<code>signers.x509.fulcio.issuer</code>	<code>string</code>	
<code>signers.x509.fulcio.provider</code>	<code>string</code>	
<code>signers.x509.identity.token.file</code>	<code>string</code>	
<code>signers.x509.tuf.mirror.url</code>	<code>string</code>	
<code>storage.docdb.mongo-server-url</code>	<code>string</code>	
<code>storage.docdb.mongo-server-url-dir</code>	<code>string</code>	
<code>storage.docdb.url</code>	<code>string</code>	
<code>storage.gcs.bucket</code>	<code>string</code>	storage configs
<code>storage.grafeas.notehint</code>	<code>string</code>	
<code>storage.grafeas.noteid</code>	<code>string</code>	
<code>storage.grafeas.projectid</code>	<code>string</code>	
<code>storage.oci.repository</code>	<code>string</code>	
<code>storage.oci.repository.insecure</code>	<code>boolean</code>	
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed

Property	Type	Description
<code>transparency.enabled</code>	<code>string</code>	
<code>transparency.url</code>	<code>string</code>	

`.spec.config`

Description

Config holds the configuration for resources created by TektonChain

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

`.spec.config.nodeSelector`

Type

`object`

`.spec.config.tolerations`

Type

`array`

.spec.config.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric comparisons (requires feature gate TaintTolerationComparisonOperators).
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By

Property	Type	Description
		default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

`.spec.controllerEnvs`

Type

`array`

`.spec.controllerEnvs[]`

Description

EnvVar represents an environment variable present in a Container.

Type

`object`

Required

`name`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the environment variable. May consist of any printable ASCII characters except '='.

Property	Type	Description
<code>value</code>	<code>string</code>	Variable references <code>\$(VAR_NAME)</code> are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$\$</code> are reduced to a single <code>\$</code> , which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>\$\$\$(VAR_NAME)</code> will produce the string literal <code>\$(VAR_NAME)</code> . Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to <code>""</code> .
<code>valueFrom</code>	<code>object</code>	Source for the environment variable's value. Cannot be used if value is not empty.

`.spec.controllerEnvs[].valueFrom`

Description

Source for the environment variable's value. Cannot be used if value is not empty.

Type

`object`

Property	Type	Description
<code>configMapKeyRef</code>	<code>object</code>	Selects a key of a ConfigMap.
<code>fieldRef</code>	<code>object</code>	Selects a field of the pod: supports <code>metadata.name</code> , <code>metadata.namespace</code> , <code>metadata.labels['<KEY>']</code> ,

Property	Type	Description
		<code>metadata.annotations['<KEY>']</code> , spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
<code>fileKeyRef</code>	<code>object</code>	FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.
<code>resourceFieldRef</code>	<code>object</code>	Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the pod's namespace

`.spec.controllerEnvs[].valueFrom.configMapKeyRef`

Description

Selects a key of a ConfigMap.

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key to select.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the ConfigMap or its key must be defined

`.spec.controllerEnvs[].valueFrom.fieldRef`

Description

Selects a field of the pod: supports `metadata.name`, `metadata.namespace`, ``metadata.labels['<KEY>']``, ``metadata.annotations['<KEY>']``, `spec.nodeName`, `spec.serviceAccountName`, `status.hostIP`, `status.podIP`, `status.podIPs`.

Type

`object`

Required

`fieldPath`

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
<code>fieldPath</code>	<code>string</code>	Path of the field to select in the specified API version.

`.spec.controllerEnvs[].valueFrom.fileKeyRef`

Description

FileKeyRef selects a key of the env file. Requires the EnvFiles feature gate to be enabled.

Type

`object`

Required

`key`

`path`

`volumeName`

Property	Type	Description
<code>key</code>	<code>string</code>	The key within the env file. An invalid key will prevent the pod from starting. The keys defined within a source may consist of any printable ASCII characters except '='. During Alpha stage of the EnvFiles feature gate, the key size is limited to 128 characters.
<code>optional</code>	<code>boolean</code>	Specify whether the file or its key must be defined. If the file or key does not exist, then the env var is not published. If optional is set to true and the specified key does not exist, the environment variable will not be set in the Pod's containers. If optional is set to false and the specified key does not exist, an error will be returned during Pod creation.

Property	Type	Description
<code>path</code>	<code>string</code>	The path within the volume from which to select the file. Must be relative and may not contain the '..' path or start with '..'.
<code>volumeName</code>	<code>string</code>	The name of the volume mount containing the env file.

`.spec.controllerEnvs[].valueFrom.resourceFieldRef`

Description

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

Type

`object`

Required

`resource`

Property	Type	Description
<code>containerName</code>	<code>string</code>	Container name: required for volumes, optional for env vars
<code>divisor</code>		Specifies the output format of the exposed resources, defaults to "1"

Property	Type	Description
<code>resource</code>	<code>string</code>	Required: resource to select

`.spec.controllerEnvs[].valueFrom.secretKeyRef`

Description

Selects a key of a secret in the pod's namespace

Type

`object`

Required

`key`

Property	Type	Description
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>name</code>	<code>string</code>	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

`.spec.options`

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.options.configMaps

Type

object

.spec.options.deployments

Type

object

.spec.options.horizontalPodAutoscalers

Type

object

.spec.options.ingress

Type

object

.spec.options.statefulSets

Type

object

.spec.options.webhookConfigurationOptions

Type

object

.spec.performance

Description

PerformanceProperties defines the fields which are configurable to tune the performance of component controller

Type

object

Required

disable-ha

Property	Type	Description
buckets	integer	
disable-ha	boolean	if it is true, disables the HA feature

Property	Type	Description
<code>kube-api-burst</code>	<code>integer</code>	
<code>kube-api-qps</code>	<code>number</code>	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5L87 ↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4go/rest/config.go#L45-L46 ↗
<code>replicas</code>	<code>integer</code>	
<code>statefulset-ordinals</code>	<code>boolean</code>	if is true, enable StatefulsetOrdinals mode
<code>threads-per-controller</code>	<code>integer</code>	The number of workers to use when processing the component

.status

Description

TektonChainStatus defines the observed state of TektonChain

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource,

Property	Type	Description
		just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>tektonInstallerSet</code>	<code>string</code>	The current installer set name for TektonChain
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

`.status.conditions[]`

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

`object`

Required

`status``type`

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.

Property	Type	Description
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonchains`
 - `DELETE` : delete collection of TektonChain
 - `GET` : list objects of kind TektonChain
 - `POST` : create a new TektonChain
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonchains/{name}`
 - `DELETE` : delete the specified TektonChain
 - `GET` : read the specified TektonChain
 - `PATCH` : partially update the specified TektonChain
 - `PUT` : replace the specified TektonChain
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonchains/{name}/status`
 - `GET` : read status of the specified TektonChain
 - `PATCH` : partially update status of the specified TektonChain
 - `PUT` : replace status of the specified TektonChain

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonchains

HTTP method

DELETE

Description

delete collection of TektonChain

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonChain

HTTP responses

HTTP code	Response body
200 - OK	TektonChainList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonChain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonChain</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonChain</code> schema
201 - Created	<code>TektonChain</code> schema

HTTP code	Response body
202 - Accepted	TektonChain schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonchains/{name}

HTTP method

[DELETE](#)

Description

delete the specified TektonChain

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

[GET](#)

Description

read the specified TektonChain

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonChain</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonChain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonChain</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TektonChain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	TektonChain schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	TektonChain schema
201 - Created	TektonChain schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonchains/{name}/status

HTTP method

GET

Description

read status of the specified TektonChain

HTTP responses

HTTP code	Response body
200 - OK	TektonChain schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified TektonChain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonChain</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonChain

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonChain</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonChain</code> schema
201 - Created	<code>TektonChain</code> schema
401 - Unauthorized	Empty



TektonHub [operator.tekton.dev/v1alpha1]

Description

TektonHub is the Schema for the tektonhub API

Type

object

Specification

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

Property	Type	Description
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
<code>metadata</code>	<code>ObjectMeta</code>	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	TektonHubStatus defines the observed state of TektonHub

.spec

Type

`object`

Property	Type	Description
<code>api</code>	<code>object</code>	
<code>catalogs</code>	<code>array</code>	
<code>categories</code>	<code>array</code>	
<code>customLogo</code>	<code>object</code>	The Base64 Encode data and mediaType of the Custom Logo

Property	Type	Description
db	object	
default	object	
options	object	options holds additions fields and these fields will be updated on the manifests
params	array	Params is the list of params passed for Hub customization
scopes	array	
targetNamespace	string	TargetNamespace is where resources will be installed

.spec.api

Type

object

Property	Type	Description
catalogRefreshInterval	string	
hubConfigUrl	string	Deprecated, will be removed in further release
routeHostUrl	string	
secret	string	

`.spec.catalogs`

Type

`array`

`.spec.catalogs[]`

Type

`object`

Property	Type	Description
<code>contextDir</code>	<code>string</code>	
<code>name</code>	<code>string</code>	
<code>org</code>	<code>string</code>	
<code>provider</code>	<code>string</code>	
<code>revision</code>	<code>string</code>	
<code>sshUrl</code>	<code>string</code>	
<code>type</code>	<code>string</code>	
<code>url</code>	<code>string</code>	

`.spec.categories`

Type

`array`

`.spec.categories[]`

Type

`string`

.spec.customLogo

Description

The Base64 Encode data and mediaType of the Custom Logo

Type

`object`

Property	Type	Description
<code>base64Data</code>	<code>string</code>	
<code>mediaType</code>	<code>string</code>	

.spec.db

Type

`object`

Property	Type	Description
<code>secret</code>	<code>string</code>	

.spec.default

Type

`object`

Property	Type	Description
<code>scopes</code>	<code>array</code>	

.spec.default.scopes

Type

array

.spec.default.scopes[]

Type

string

.spec.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

object

Property	Type	Description
configMaps	object	
deployments	object	
disabled	boolean	
horizontalPodAutoscalers	object	
ingress	object	
statefulSets	object	
webhookConfigurationOptions	object	

.spec.options.configMaps

Type`object`**.spec.options.deployments****Type**`object`**.spec.options.horizontalPodAutoscalers****Type**`object`**.spec.options.ingress****Type**`object`**.spec.options.statefulSets****Type**`object`**.spec.options.webhookConfigurationOptions****Type**`object`**.spec.params****Description**

Params is the list of params passed for Hub customization

Type

array

.spec.params[]

Description

Param declares an string value to use for the parameter called name.

Type

object

Property	Type	Description
name	string	
value	string	

.spec.scopes

Type

array

.spec.scopes[]

Type

object

Property	Type	Description
name	string	
users	array	

.spec.scopes[].users

Type

array

.spec.scopes[].users[]

Type

string

.status

Description

TektonHubStatus defines the observed state of TektonHub

Type

object

Property	Type	Description
annotations	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
apiUrl	string	The URL route for API which needs to be exposed
authUrl	string	The URL route for Auth server

Property	Type	Description
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>hubInstallerSets</code>	<code>object</code>	The current installer set name
<code>manifests</code>	<code>array</code>	The url links of the manifests, separated by comma
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>uiUrl</code>	<code>string</code>	The URL route for UI which needs to be exposed
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.

Property	Type	Description
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

.status.hubInstallerSets

Description

The current installer set name

Type

`object`

.status.manifests

Description

The url links of the manifests, separated by comma

Type

`array`

.status.manifests[]

Type

`string`

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonhubs`
 - `DELETE` : delete collection of TektonHub
 - `GET` : list objects of kind TektonHub
 - `POST` : create a new TektonHub
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonhubs/{name}`
 - `DELETE` : delete the specified TektonHub
 - `GET` : read the specified TektonHub
 - `PATCH` : partially update the specified TektonHub
 - `PUT` : replace the specified TektonHub
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonhubs/{name}/status`
 - `GET` : read status of the specified TektonHub
 - `PATCH` : partially update status of the specified TektonHub
 - `PUT` : replace status of the specified TektonHub

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonhubs`

HTTP method

`DELETE`

Description

delete collection of TektonHub

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonHub

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHubList</code> ↗ schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonHub

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

Parameter	Type	Description
		duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonHub</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
201 - Created	<code>TektonHub</code> schema
202 - Accepted	<code>TektonHub</code> schema
401 - Unauthorized	Empty

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonhubs/{name}`

HTTP method

DELETE

Description

delete the specified TektonHub

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified TektonHub

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonHub

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TektonHub

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonHub</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
201 - Created	<code>TektonHub</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonhubs/{name}/status

HTTP method

`GET`

Description

read status of the specified TektonHub

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonHub

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonHub

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonHub</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonHub</code> schema
201 - Created	<code>TektonHub</code> schema

HTTP code	Response body
401 - Unauthorized	Empty

TektonResult

[operator.tekton.dev/v1alpha1]

Description

TektonResult is the Schema for the tektonresults API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	TektonResultSpec defines the desired state of TektonResult
<code>status</code>	<code>object</code>	TektonResultStatus defines the observed state of TektonResult

.spec

Description

TektonResultSpec defines the desired state of TektonResult

Type

`object`

Required

`disabled`

`is_external_db`

Property	Type	Description
<code>auth_disable</code>	<code>boolean</code>	
<code>auth_impersonate</code>	<code>boolean</code>	

Property	Type	Description
<code>config</code>	<code>object</code>	Config holds the configuration for resources created by TektonResult
<code>db_enable_auto_migration</code>	<code>boolean</code>	
<code>db_host</code>	<code>string</code>	
<code>db_name</code>	<code>string</code>	
<code>db_port</code>	<code>integer</code>	
<code>db_secret_name</code>	<code>string</code>	
<code>db_secret_password_key</code>	<code>string</code>	
<code>db_secret_user_key</code>	<code>string</code>	
<code>db_sslmode</code>	<code>string</code>	
<code>db_sslrootcert</code>	<code>string</code>	
<code>disabled</code>	<code>boolean</code>	enable or disable Result Component
<code>gcs_bucket_name</code>	<code>string</code>	
<code>gcs_creds_secret_key</code>	<code>string</code>	
<code>gcs_creds_secret_name</code>	<code>string</code>	
<code>is_external_db</code>	<code>boolean</code>	
<code>log_level</code>	<code>string</code>	

Property	Type	Description
logging_plugin_api_url	string	
logging_plugin_ca_cert	string	
logging_plugin_forwarder_delay_duration	integer	
logging_plugin_multipart_regex	string	
logging_plugin_namespace_key	string	
logging_plugin_proxy_path	string	
logging_plugin_query_limit	integer	
logging_plugin_query_params	string	
logging_plugin_static_labels	string	
logging_plugin_tls_verification_disable	boolean	
logging_plugin_token_path	string	
logging_pvc_name	string	
logs_api	boolean	
logs_buffer_size	integer	
logs_path	string	
logs_type	string	
loki_stack_name	string	
loki_stack_namespace	string	
options	object	Options holds additions fields and these fields will

Property	Type	Description
		be updated on the manifests
performance	object	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
prometheus_histogram	boolean	
prometheus_port	integer	
route_enabled	boolean	Route configuration for Results API service exposure
route_host	string	
route_path	string	
route_tls_termination	string	
secret_name	string	name of the secret used to get S3 credentials and pass it as environment variables to the "tekton-results-api" deployment under "api" container
server_port	integer	

Property	Type	Description
<code>storage_emulator_host</code>	<code>string</code>	
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed
<code>tls_hostname_override</code>	<code>string</code>	

.spec.config

Description

Config holds the configuration for resources created by TektonResult

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

.spec.config.nodeSelector

Type

`object`

.spec.config.tolerations

Type

array

.spec.config.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric comparisons (requires feature gate TaintTolerationComparisonOperators).

Property	Type	Description
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.options

Description

Options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	

Property	Type	Description
<code>webhookConfigurationOptions</code>	<code>object</code>	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

Type

`object`

.spec.options.ingress

Type

`object`

.spec.options.statefulSets

Type

`object`

.spec.options.webhookConfigurationOptions

Type

`object`

.spec.performance

Description

PerformanceProperties defines the fields which are configurable to tune the performance of component controller

Type

`object`

Required

`disable-ha`

Property	Type	Description
<code>buckets</code>	<code>integer</code>	
<code>disable-ha</code>	<code>boolean</code>	if it is true, disables the HA feature
<code>kube-api-burst</code>	<code>integer</code>	
<code>kube-api-qps</code>	<code>number</code>	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5L87 ↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4go/rest/config.go#L45-L46 ↗
<code>replicas</code>	<code>integer</code>	
<code>statefulset-ordinals</code>	<code>boolean</code>	if is true, enable StatefulsetOrdinals mode

Property	Type	Description
<code>threads-per-controller</code>	<code>integer</code>	The number of workers to use when processing the component

.status

Description

TektonResultStatus defines the observed state of TektonResult

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>tektonInstallerSet</code>	<code>string</code>	The current installer set name for TektonResult

Property	Type	Description
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

`object`

Required

`status`

`type`

Property	Type	Description
<code>lastTransitionTime</code>	<code>string</code>	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
<code>message</code>	<code>string</code>	A human readable message indicating details about the transition.
<code>reason</code>	<code>string</code>	The reason for the condition's last transition.
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonresults`

- **DELETE** : delete collection of TektonResult
- **GET** : list objects of kind TektonResult
- **POST** : create a new TektonResult
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonresults/{name}`
 - **DELETE** : delete the specified TektonResult
 - **GET** : read the specified TektonResult
 - **PATCH** : partially update the specified TektonResult
 - **PUT** : replace the specified TektonResult
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonresults/{name}/status`
 - **GET** : read status of the specified TektonResult
 - **PATCH** : partially update status of the specified TektonResult
 - **PUT** : replace status of the specified TektonResult

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonresults`

HTTP method

DELETE

Description

delete collection of TektonResult

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonResult

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResultList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonResult

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the

Parameter	Type	Description
		request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	TektonResult schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	TektonResult schema
201 - Created	TektonResult schema
202 - Accepted	TektonResult schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonresults/{name}

HTTP method

DELETE

Description

delete the specified TektonResult

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

read the specified TektonResult

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResult</code>  schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonResult

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResult</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified `TektonResult`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonResult</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResult</code> schema

HTTP code	Response body
201 - Created	<code>TektonResult</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonresults/{name}/status

HTTP method

`GET`

Description

read status of the specified TektonResult

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResult</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonResult

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResult</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonResult

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun

Parameter	Type	Description
		directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonResult</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonResult</code> schema
201 - Created	<code>TektonResult</code> schema
401 - Unauthorized	Empty

TektonInstallerSet

[operator.tekton.dev/v1alpha1]

Description

TektonInstallerSet is the Schema for the TektonInstallerSet API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	ObjectMeta ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	TektonInstallerSetSpec defines the desired state of TektonInstallerSet
<code>status</code>	<code>object</code>	TektonInstallerSetStatus defines the observed state of TektonInstallerSet

.spec

Description

TektonInstallerSetSpec defines the desired state of TektonInstallerSet

Type

`object`

Property	Type	Description
<code>manifests</code>		

.status

Description

TektonInstallerSetStatus defines the observed state of TektonInstallerSet

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.

Property	Type	Description
<code>severity</code>	<code>string</code>	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
<code>status</code>	<code>string</code>	Status of the condition, one of True, False, Unknown.
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektoninstallersets`
 - `DELETE` : delete collection of TektonInstallerSet
 - `GET` : list objects of kind TektonInstallerSet
 - `POST` : create a new TektonInstallerSet
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektoninstallersets/{name}`
 - `DELETE` : delete the specified TektonInstallerSet
 - `GET` : read the specified TektonInstallerSet
 - `PATCH` : partially update the specified TektonInstallerSet
 - `PUT` : replace the specified TektonInstallerSet

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektoninstallersets/{name}/status`
 - **GET** : read status of the specified TektonInstallerSet
 - **PATCH** : partially update status of the specified TektonInstallerSet
 - **PUT** : replace status of the specified TektonInstallerSet

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektoninstallersets`

HTTP method

DELETE

Description

delete collection of TektonInstallerSet

HTTP responses

HTTP code	Response body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonInstallerSet

HTTP responses

HTTP code	Response body
200 - OK	TektonInstallerSetList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonInstallerSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonInstallerSet</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonInstallerSet</code> schema
201 - Created	<code>TektonInstallerSet</code> schema
202 - Accepted	<code>TektonInstallerSet</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektoninstallersets/{name}

HTTP method

`DELETE`

Description

delete the specified TektonInstallerSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	<code>Status</code>  schema
202 - Accepted	<code>Status</code>  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified TektonInstallerSet

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonInstallerSet</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update the specified TektonInstallerSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonInstallerSet</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified TektonInstallerSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	TektonInstallerSet schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	TektonInstallerSet schema
201 - Created	TektonInstallerSet schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektoninstallersets/{name}/status

HTTP method

GET

Description

read status of the specified TektonInstallerSet

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonInstallerSet</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonInstallerSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonInstallerSet</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified TektonInstallerSet

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonInstallerSet</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonInstallerSet</code> schema
201 - Created	<code>TektonInstallerSet</code> schema
401 - Unauthorized	Empty

OpenShift Pipelines as Code

[operator.tekton.dev/v1alpha1]

Description

OpenShiftPipelinesAsCode is the Schema for the OpenShiftPipelinesAsCode API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
metadata	ObjectMeta ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
spec	object	OpenShiftPipelinesAsCodeSpec defines the desired state of OpenShiftPipelinesAsCode
status	object	OpenShiftPipelinesAsCodeStatus defines the observed state of OpenShiftPipelinesAsCode

.spec

Description

OpenShiftPipelinesAsCodeSpec defines the desired state of OpenShiftPipelinesAsCode

Type

object

Property	Type	Description
additionalPACControllers	object	AdditionalPACControllers allows to deploy additional PAC controller
config	object	

Property	Type	Description
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>settings</code>	<code>object</code>	
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed

.spec.additionalPACControllers

Description

AdditionalPACControllers allows to deploy additional PAC controller

Type

`object`

.spec.config

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

.spec.config.nodeSelector

Type

object

.spec.config.tolerations

Type

array

.spec.config.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.

Property	Type	Description
<code>operator</code>	<code>string</code>	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric comparisons (requires feature gate TaintTolerationComparisonOperators).
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	

Property	Type	Description
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	
<code>webhookConfigurationOptions</code>	<code>object</code>	

`.spec.options.configMaps`

Type

`object`

`.spec.options.deployments`

Type

`object`

`.spec.options.horizontalPodAutoscalers`

Type

`object`

`.spec.options.ingress`

Type

`object`

.spec.options.statefulSets

Type

object

.spec.options.webhookConfigurationOptions

Type

object

.spec.settings

Type

object

.status

Description

OpenShiftPipelinesAsCodeStatus defines the observed state of OpenShiftPipelinesAsCode

Type

object

Property	Type	Description
annotations	object	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Property	Type	Description
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

`object`

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

`array`

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See:
<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.

Property	Type	Description
<code>type</code>	<code>string</code>	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/openshiftpipelinesascodes`
 - `DELETE` : delete collection of OpenShiftPipelinesAsCode
 - `GET` : list objects of kind OpenShiftPipelinesAsCode
 - `POST` : create a new OpenShiftPipelinesAsCode
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/openshiftpipelinesascodes/{name}`
 - `DELETE` : delete the specified OpenShiftPipelinesAsCode
 - `GET` : read the specified OpenShiftPipelinesAsCode
 - `PATCH` : partially update the specified OpenShiftPipelinesAsCode
 - `PUT` : replace the specified OpenShiftPipelinesAsCode
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/openshiftpipelinesascodes/{name}/status`
 - `GET` : read status of the specified OpenShiftPipelinesAsCode
 - `PATCH` : partially update status of the specified OpenShiftPipelinesAsCode
 - `PUT` : replace status of the specified OpenShiftPipelinesAsCode

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/openshiftpipelinesascodes`

HTTP method

DELETE

Description

delete collection of OpenShiftPipelinesAsCode

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind OpenShiftPipelinesAsCode

HTTP responses

HTTP code	Response body
200 - OK	OpenShiftPipelinesAsCodeList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new OpenShiftPipelinesAsCode

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>OpenShiftPipelinesAsCode</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>OpenShiftPipelinesAsCode</code> schema
201 - Created	<code>OpenShiftPipelinesAsCode</code> schema
202 - Accepted	<code>OpenShiftPipelinesAsCode</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/openshiftpipelinesascode/{name}

HTTP method

DELETE

Description

delete the specified OpenShiftPipelinesAsCode

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
202 - Accepted	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified OpenShiftPipelinesAsCode

HTTP responses

HTTP code	Response body
200 - OK	<code>OpenShiftPipelinesAsCode</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified `OpenShiftPipelinesAsCode`

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>OpenShiftPipelinesAsCode</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified OpenShiftPipelinesAsCode

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a <code>BadRequest</code> error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>OpenShiftPipelinesAsCode</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>OpenShiftPipelinesAsCode</code> schema
201 - Created	<code>OpenShiftPipelinesAsCode</code> schema
401 - Unauthorized	Empty

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/openshiftpipelinesascodes/{name}/status`

HTTP method

GET

Description

read status of the specified OpenShiftPipelinesAsCode

HTTP responses

HTTP code	Response body
200 - OK	<code>OpenShiftPipelinesAsCode</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified OpenShiftPipelinesAsCode

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>OpenShiftPipelinesAsCode</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified OpenShiftPipelinesAsCode

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	OpenShiftPipelinesAsCode schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	OpenShiftPipelinesAsCode schema
201 - Created	OpenShiftPipelinesAsCode schema
401 - Unauthorized	Empty

TektonPruner

[operator.tekton.dev/v1alpha1]

Description

TektonPruner is the Schema for the TektonPruner API

Type

object

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

Property	Type	Description
		https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds ↗
<code>metadata</code>	<code>ObjectMeta</code> ↗	ObjectMeta is metadata that all persisted resources must have, which includes all objects users must create.
<code>spec</code>	<code>object</code>	
<code>status</code>	<code>object</code>	TektonPrunerStatus defines the observed state of TektonPruner

.spec

Type

`object`

Required

`disabled``global-config`

Property	Type	Description
<code>config</code>	<code>object</code>	Config holds the configuration for resources created by TektonPruner
<code>disabled</code>	<code>boolean</code>	enable or disable TektonPruner Component
<code>global-config</code>	☐	

Property	Type	Description
<code>options</code>	<code>object</code>	options holds additions fields and these fields will be updated on the manifests
<code>targetNamespace</code>	<code>string</code>	TargetNamespace is where resources will be installed

`.spec.config`

Description

Config holds the configuration for resources created by TektonPruner

Type

`object`

Property	Type	Description
<code>nodeSelector</code>	<code>object</code>	
<code>priorityClassName</code>	<code>string</code>	PriorityClassName holds the priority class to be set to pod template
<code>tolerations</code>	<code>array</code>	

`.spec.config.nodeSelector`

Type

`object`

`.spec.config.tolerations`

Type

array

.spec.config.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
effect	string	Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
key	string	Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
operator	string	Operator represents a key's relationship to the value. Valid operators are Exists, Equal, Lt, and Gt. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category. Lt and Gt perform numeric comparisons (requires feature gate TaintTolerationComparisonOperators).

Property	Type	Description
<code>tolerationSeconds</code>	<code>integer</code>	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
<code>value</code>	<code>string</code>	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

.spec.options

Description

options holds additions fields and these fields will be updated on the manifests

Type

`object`

Property	Type	Description
<code>configMaps</code>	<code>object</code>	
<code>deployments</code>	<code>object</code>	
<code>disabled</code>	<code>boolean</code>	
<code>horizontalPodAutoscalers</code>	<code>object</code>	
<code>ingress</code>	<code>object</code>	
<code>statefulSets</code>	<code>object</code>	

Property	Type	Description
<code>webhookConfigurationOptions</code>	<code>object</code>	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

Type

`object`

.spec.options.ingress

Type

`object`

.spec.options.statefulSets

Type

`object`

.spec.options.webhookConfigurationOptions

Type

`object`

.status

Description

TektonPrunerStatus defines the observed state of TektonPruner

Type

`object`

Property	Type	Description
<code>annotations</code>	<code>object</code>	Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.
<code>conditions</code>	<code>array</code>	Conditions the latest available observations of a resource's current state.
<code>observedGeneration</code>	<code>integer</code>	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
<code>tektonInstallerSet</code>	<code>string</code>	The current installer set name for TektonPruner
<code>version</code>	<code>string</code>	The version of the installed release

.status.annotations

Description

Annotations is additional Status fields for the Resource to save some additional State as well as convey more information to the user. This is roughly akin to Annotations on any k8s resource, just the reconciler conveying richer information outwards.

Type

object

.status.conditions

Description

Conditions the latest available observations of a resource's current state.

Type

array

.status.conditions[]

Description

Condition defines a readiness condition for a Knative resource. See: <https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#typical-status-properties>

Type

object

Required

status

type

Property	Type	Description
lastTransitionTime	string	LastTransitionTime is the last time the condition transitioned from one status to another. We use VolatileTime in place of metav1.Time to exclude

Property	Type	Description
		this from creating equality.Semantic differences (all other things held constant).
message	string	A human readable message indicating details about the transition.
reason	string	The reason for the condition's last transition.
severity	string	Severity with which to treat failures of this type of condition. When this is not specified, it defaults to Error.
status	string	Status of the condition, one of True, False, Unknown.
type	string	Type of condition.

API Endpoints

The following API endpoints are available:

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpruners`
 - `DELETE` : delete collection of TektonPruner
 - `GET` : list objects of kind TektonPruner
 - `POST` : create a new TektonPruner

- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpruners/{name}`
 - **DELETE** : delete the specified TektonPruner
 - **GET** : read the specified TektonPruner
 - **PATCH** : partially update the specified TektonPruner
 - **PUT** : replace the specified TektonPruner
- `/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpruners/{name}/status`
 - **GET** : read status of the specified TektonPruner
 - **PATCH** : partially update status of the specified TektonPruner
 - **PUT** : replace status of the specified TektonPruner

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpruners`

HTTP method

DELETE

Description

delete collection of TektonPruner

HTTP responses

HTTP code	Response body
200 - OK	Status ↗ schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list objects of kind TektonPruner

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPrunerList</code> schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a new TektonPruner

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+. - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonPruner</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPruner</code> schema
201 - Created	<code>TektonPruner</code> schema
202 - Accepted	<code>TektonPruner</code> schema
401 - Unauthorized	Empty

/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpruners/{name}

HTTP method

DELETE

Description

delete the specified TektonPruner

Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

HTTP responses

HTTP code	Response body
200 - OK	Status  schema
202 - Accepted	Status  schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified TektonPruner

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPruner</code> schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonPruner

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPruner</code> schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified TektonPruner

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
<code>body</code>	<code>TektonPruner</code> schema	<code>application/json</code> formatted

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPruner</code> schema
201 - Created	<code>TektonPruner</code> schema
401 - Unauthorized	Empty

`/apis/operator.tekton.dev/v1alpha1/namespaces/{namespace}/tektonpruners/{name}/status`

HTTP method

GET

Description

read status of the specified TektonPruner

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPruner</code> schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified TektonPruner

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized <code>dryRun</code> directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	<code>fieldValidation</code> instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

Parameter	Type	Description
		This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP responses

HTTP code	Response body
200 - OK	<code>TektonPruner</code> schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonPruner

Query parameters

Parameter	Type	Description
<code>dryRun</code>	<code>string</code>	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
<code>fieldValidation</code>	<code>string</code>	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

Parameter	Type	Description
		and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Body parameters

Parameter	Type	Description
body	TektonPruner schema	application/json formatted

HTTP responses

HTTP code	Response body
200 - OK	TektonPruner schema
201 - Created	TektonPruner schema
401 - Unauthorized	Empty