Operator

TektonConfig [operator.tekton.c | TektonInstallerSet [operator.tek | TektonPipel

TektonTrigger [operator.tekton. | TektonChain [operator.tekton.d | TektonHub

TektonResult [operator.tekton.c | TektonInstallerSet [operator.tek | OpenShift P

TektonConfig [operator.tekton.dev/v1alpha1]

Description

TektonConfig is the Schema for the TektonConfigs 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 ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

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`	TektonConfigSpec defines the desired state of TektonConfig
`status`	`object`	TektonConfigStatus defines the observed state of TektonConfig

.spec

Description

TektonConfigSpec defines the desired state of TektonConfig

Type

`object`

Property	Type	Description
`addon`	`object`	Addon holds the addons config
`chain`	`object`	Chain holds the customizable option for chains component

Property	Type	Description
`config`	`object`	Config holds the configuration for resources created by TektonConfig
`dashboard`	`object`	Dashboard holds the customizable options for dashboards component
`hub`	`object`	Hub holds the hub config
`params`	`array`	Params is the list of params passed for all platforms
`pipeline`	`object`	Pipeline holds the customizable option for pipeline component
`platforms`	`object`	Platforms allows configuring platform specific configurations
`profile`	`string`	
`pruner`	`object`	Pruner holds the prune config
`result`	`object`	Result holds the customize option for results component

Property	Type	Description
`targetNamespace`	`string`	TargetNamespace is where resources will be installed
`targetNamespaceMetadata`	`object`	holds target namespace metadata
`tektonpruner`	`object`	New EventBasedPruner which provides more granular control over TaskRun and PipelineRuns
`trigger`	`object`	Trigger holds the customizable option for triggers component

.spec.addon

Description

Addon holds the addons config

Type

`object`

Property	Type	Description
`enablePipelinesAsCode`	`boolean`	Deprecated, will be removed in further release EnablePAC field defines whether to install PAC

Property	Type	Description
`params`	`array`	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
`name`	`string`	
`value`	`string`	

.spec.chain

Description

Chain holds the customizable option for chains component

Type

`object`

Required

`disabled` `options`

Property	Type	Description
`artifacts.oci.format`	`string`	oci artifacts config
`artifacts.oci.signer`	`string`	
`artifacts.oci.storage`	`string`	
`artifacts.pipelinerun.enable-deep-inspection`	`string`	
`artifacts.pipelinerun.format`	`string`	pipelinerun artifacts config
`artifacts.pipelinerun.signer`	`string`	
`artifacts.pipelinerun.storage`	`string`	
`artifacts.taskrun.format`	`string`	taskrun artifacts config
`artifacts.taskrun.signer`	`string`	
`artifacts.taskrun.storage`	`string`	
`builddefinition.buildtype`	`string`	
`builder.id`	`string`	builder config
`controllerEnvs`	`array`	

Property	Type	Description
`disabled`	`boolean`	enable or disable chains feature
`generateSigningSecret`	`boolean`	generate signing key
`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
`signers.kms.auth.address`	`string`	
`signers.kms.auth.oidc.path`	`string`	
`signers.kms.auth.oidc.role`	`string`	
`signers.kms.auth.spire.audience`	`string`	
`signers.kms.auth.spire.sock`	`string`	
`signers.kms.auth.token`	`string`	
`signers.kms.auth.token-path`	`string`	
`signers.kms.kmsref`	`string`	kms signer config

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

.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
`name`	`string`	Name of the environment variable. Must be a C_IDENTIFIER.
`value`	`string`	Variable references $(VAR_NAME) 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 $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".

Property	Type	Description
`valueFrom`	`object`	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
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.
`fieldRef`	`object`	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.
`resourceFieldRef`	`object`	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.

Property	Type	Description
`secretKeyRef`	`object`	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
`key`	`string`	The key to select.
`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 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
`apiVersion`	`string`	Version of the schema the FieldPath is written in terms of, defaults to "v1".
`fieldPath`	`string`	Path of the field to select in the specified API version.

.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
`containerName`	`string`	Container name: required for volumes, optional for env vars
`divisor`		Specifies the output format of the exposed resources, defaults to "1"
`resource`	`string`	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:

Property	Type	Description
		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`	
`deployments`	`object`	
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.chain.options.configMaps

Type

`object`

.spec.chain.options.deployments

Type

`object`

.spec.chain.options.horizontalPodAutoscalers

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/05d67e427c722a2a5 L87↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4 go/rest/config.go#L45-L46↗
`replicas`	`integer`	
`statefulset-ordinals`	`boolean`	if is true, enable StatefulsetOrdinals mode
`threads-per-controller`	`integer`	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
`nodeSelector`	`object`	
`priorityClassName`	`string`	PriorityClassName holds the priority class to be set to pod template
`tolerations`	`array`	

.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,

Property	Type	Description
		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 and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
`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 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.
`value`	`string`	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

`options` `readonly`

Property	Type	Description
`external-logs`	`string`	
`options`	`object`	options holds additions fields and these fields will be updated on the manifests
`readonly`	`boolean`	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
`configMaps`	`object`	
`deployments`	`object`	
`disabled`	`boolean`	

Property	Type	Description
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.dashboard.options.configMaps

Type

`object`

.spec.dashboard.options.deployments

Type

`object`

.spec.dashboard.options.horizontalPodAutoscalers

Type

`object`

.spec.dashboard.options.statefulSets

Type

`object`

.spec.dashboard.options.webhookConfigurationOptions

Type

`object`

.spec.hub

Description

Hub holds the hub config

Type

`object`

Required

`options`

Property	Type	Description
`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

.spec.hub.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`	

Property	Type	Description
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.hub.options.configMaps

Type

`object`

.spec.hub.options.deployments

Type

`object`

.spec.hub.options.horizontalPodAutoscalers

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.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
`name`	`string`	
`value`	`string`	

.spec.pipeline

Description

Pipeline holds the customizable option for pipeline component

Type

`object`

Required

`options`

Property	Type	Description
`await-sidecar-readiness`	`boolean`	
`bundles-resolver-config`	`object`	
`cluster-resolver-config`	`object`	
`coschedule`	`string`	
`default-affinity-assistant-pod-template`	`string`	
`default-cloud-events-sink`	`string`	
`default-forbidden-env`	`string`	
`default-managed-by-label-value`	`string`	

Property	Type	Description
`default-max-matrix-combinations-count`	`string`	
`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`	
`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`	
`enable-param-enum`	`boolean`	
`enable-provenance-in-status`	`boolean`	

Property	Type	Description
`enable-step-actions`	`boolean`	
`enable-tekton-oci-bundles`	`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
`enforce-nonfalsifiability`	`string`	
`git-resolver-config`	`object`	
`hub-resolver-config`	`object`	
`keep-pod-on-cancel`	`boolean`	
`max-result-size`	`integer`	
`metrics.count.enable-reason`	`boolean`	
`metrics.pipelinerun.duration-type`	`string`	
`metrics.pipelinerun.level`	`string`	
`metrics.taskrun.duration-type`	`string`	
`metrics.taskrun.level`	`string`	
`options`	`object`	options holds additions fields and these fields will be updated on the manifests

Property	Type	Description
`params`	`array`	The params to customize different components of Pipelines
`performance`	`object`	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
`require-git-ssh-secret-known-hosts`	`boolean`	
`results-from`	`string`	
`running-in-environment-with-injected-sidecars`	`boolean`	
`scope-when-expressions-to-task`	`boolean`	ScopeWhenExpressionsToTask is deprecated and never used.
`send-cloudevents-for-runs`	`boolean`	
`set-security-context`	`boolean`	
`trusted-resources-verification-no-match-policy`	`string`	
`verification-mode`	`string`	

.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`	
`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.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
`kube-api-burst`	`integer`	

Property	Type	Description
`kube-api-qps`	`number`	queries per second (QPS) and burst to the master from rest API https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a5 L87↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4 go/rest/config.go#L45-L46↗
`replicas`	`integer`	
`statefulset-ordinals`	`boolean`	if is true, enable StatefulsetOrdinals mode
`threads-per-controller`	`integer`	The number of workers to use when processing the component

.spec.platforms

Description

Platforms allows configuring platform specific configurations

Type

`object`

Property	Type	Description
`openshift`	`object`	OpenShift allows configuring openshift specific components and configurations

.spec.platforms.openshift

Description

OpenShift allows configuring openshift specific components and configurations

Type

`object`

Property	Type	Description
`pipelinesAsCode`	`object`	PipelinesAsCode allows configuring PipelinesAsCode configurations
`scc`	`object`	SCC allows configuring security context constraints used by workloads

.spec.platforms.openshift.pipelinesAsCode

Description

PipelinesAsCode allows configuring PipelinesAsCode configurations

Type

`object`

Required

`options`

Property	Type	Description
`additionalPACControllers`	`object`	AdditionalPACControllers allows to deploy additional PAC controller
`enable`	`boolean`	Enable or disable pipelines as code by changing this bool

Property	Type	Description
`options`	`object`	options holds additions fields and these fields will be updated on the manifests
`settings`	`object`	

.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
`configMaps`	`object`	
`deployments`	`object`	
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	

Property	Type	Description
`webhookConfigurationOptions`	`object`	

.spec.platforms.openshift.pipelinesAsCode.options.configMaps

Type

`object`

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

Type

`object`

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

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 ser account used for workloads (`pipeline` SA by default) and def 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
`prune-per-resource`	`boolean`	apply the prune job to the individual resources
`resources`	`array`	The resources which need to be pruned
`schedule`	`string`	How frequent pruning should happen

Property	Type	Description
`startingDeadlineSeconds`	`integer`	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` `options`

Property	Type	Description
`auth_disable`	`boolean`	

Property	Type	Description
`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`	
`gcs_creds_secret_name`	`string`	
`is_external_db`	`boolean`	
`log_level`	`string`	
`logging_plugin_api_url`	`string`	
`logging_plugin_ca_cert`	`string`	

Property	Type	Description
`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 be updated on the manifests

Property	Type	Description
`performance`	`object`	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
`prometheus_histogram`	`boolean`	
`prometheus_port`	`integer`	
`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`	
`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`	
`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.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
`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/05d67e427c722a2a5 L87↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4 go/rest/config.go#L45-L46↗
`replicas`	`integer`	

Property	Type	Description
`statefulset-ordinals`	`boolean`	if is true, enable StatefulsetOrdinals mode
`threads-per-controller`	`integer`	The number of workers to use when processing the component

.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` `options`

Property	Type	Description
`disabled`	`boolean`	enable or disable TektonPruner Component
`options`	`object`	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
`configMaps`	`object`	
`deployments`	`object`	
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	

Property	Type	Description
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.tektonpruner.options.configMaps

Type

`object`

.spec.tektonpruner.options.deployments

Type

`object`

.spec.tektonpruner.options.horizontalPodAutoscalers

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

`options`

Property	Type	Description
`default-service-account`	`string`	
`enable-api-fields`	`string`	
`options`	`object`	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`	

Property	Type	Description
`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.statefulSets

Type

`object`

.spec.trigger.options.webhookConfigurationOptions

Type

`object`

.status

Description

TektonConfigStatus defines the observed state of TektonConfig

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.
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`profile`	`string`	The profile installed
`tektonInstallerSets`	`object`	The current installer set name
`version`	`string`	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.
`type`	`string`	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	`TektonConfigList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new TektonConfig

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		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
`body`	`TektonConfig` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TektonConfig` schema
201 - Created	`TektonConfig` schema
202 - Accepted	`TektonConfig` 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
`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 TektonConfig

HTTP responses

HTTP code	Response body
200 - OK	`TektonConfig` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonConfig

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

HTTP responses

HTTP code	Response body
200 - OK	`TektonConfig` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TektonConfig

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

Body parameters

Parameter	Type	Description
`body`	`TektonConfig` schema	`application/json` formatted

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified TektonConfig

Query parameters

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

HTTP responses

HTTP code	Response body
200 - OK	`TektonConfig` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonConfig

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

Body parameters

Parameter	Type	Description
`body`	`TektonConfig` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TektonConfig` schema
201 - Created	`TektonConfig` schema

HTTP code	Response body
401 - Unauthorized	Empty

TektonInstallerSet [operator.tekton.dev/v1alpha1]

Description

TektonInstallerSet is the Schema for the TektonInstallerSet 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 ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

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`	TektonInstallerSetSpec defines the desired state of TektonInstallerSet
`status`	`object`	TektonInstallerSetStatus defines the observed state of TektonInstallerSet

.spec

Description

TektonInstallerSetSpec defines the desired state of TektonInstallerSet

Type

`object`

Property	Type	Description
`manifests`	`array`	Slice is a Source comprised of existing objects

.spec.manifests

Description

Slice is a Source comprised of existing objects

Type

`array`

.spec.manifests[]

Type

`object`

.status

Description

TektonInstallerSetStatus defines the observed state of TektonInstallerSet

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.
`observedGeneration`	`integer`	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

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

Parameter	Type	Description
		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
202 - Accepted	`TektonInstallerSet` 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
`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

Parameter	Type	Description
		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 TektonInstallerSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified TektonInstallerSet

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified TektonInstallerSet

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

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

HTTP code	Response body
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	`TektonInstallerSet` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonInstallerSet

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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	`TektonInstallerSet` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonInstallerSet

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

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

HTTP code	Response body
401 - Unauthorized	Empty

TektonPipeline [operator.tekton.dev/v1alpha1]

Description

TektonPipeline is the Schema for the tektonpipelines 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 ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

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`	TektonPipelineSpec defines the desired state of TektonPipeline
`status`	`object`	TektonPipelineStatus defines the observed state of TektonPipeline

.spec

Description

TektonPipelineSpec defines the desired state of TektonPipeline

Type

`object`

Required

`options`

Property	Type	Description
`await-sidecar-readiness`	`boolean`	
`bundles-resolver-config`	`object`	
`cluster-resolver-config`	`object`	

Property	Type	Description
`config`	`object`	Config holds the configuration for resources created by TektonPipeline
`coschedule`	`string`	
`default-affinity-assistant-pod-template`	`string`	
`default-cloud-events-sink`	`string`	
`default-forbidden-env`	`string`	
`default-managed-by-label-value`	`string`	
`default-max-matrix-combinations-count`	`string`	
`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`	
`disable-creds-init`	`boolean`	
`disable-inline-spec`	`string`	
`embedded-status`	`string`	

Property	Type	Description
`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`	
`enable-param-enum`	`boolean`	
`enable-provenance-in-status`	`boolean`	
`enable-step-actions`	`boolean`	
`enable-tekton-oci-bundles`	`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
`enforce-nonfalsifiability`	`string`	
`git-resolver-config`	`object`	
`hub-resolver-config`	`object`	
`keep-pod-on-cancel`	`boolean`	
`max-result-size`	`integer`	
`metrics.count.enable-reason`	`boolean`	

Property	Type	Description
`metrics.pipelinerun.duration-type`	`string`	
`metrics.pipelinerun.level`	`string`	
`metrics.taskrun.duration-type`	`string`	
`metrics.taskrun.level`	`string`	
`options`	`object`	options holds additions fields and these fields will be updated on the manifests
`params`	`array`	The params to customize different components of Pipelines
`performance`	`object`	PerformanceProperties defines the fields which are configurable to tune the performance of component controller
`require-git-ssh-secret-known-hosts`	`boolean`	
`results-from`	`string`	
`running-in-environment-with-injected-sidecars`	`boolean`	
`scope-when-expressions-to-task`	`boolean`	ScopeWhenExpressionsToTask is deprecated and never used.

Property	Type	Description
`send-cloudevents-for-runs`	`boolean`	
`set-security-context`	`boolean`	
`targetNamespace`	`string`	TargetNamespace is where resources will b installed
`trusted-resources-verification-no-match-policy`	`string`	
`verification-mode`	`string`	

.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
`nodeSelector`	`object`	
`priorityClassName`	`string`	PriorityClassName holds the priority class to be set to pod template
`tolerations`	`array`	

.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,

Property	Type	Description
		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 and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
`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 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.
`value`	`string`	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`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

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/05d67e427c722a2a5 L87↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4 go/rest/config.go#L45-L46↗
`replicas`	`integer`	

Property	Type	Description
`statefulset-ordinals`	`boolean`	if is true, enable StatefulsetOrdinals mode
`threads-per-controller`	`integer`	The number of workers to use when processing the component

.status

Description

TektonPipelineStatus defines the observed state of TektonPipeline

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.
`extTektonInstallerSets`	`object`	The installer sets created for extension components

Property	Type	Description
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`tektonInstallerSet`	`string`	The current installer set name for TektonPipeline
`version`	`string`	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
`type`	`string`	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	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind TektonPipeline

HTTP responses

HTTP code	Response body
200 - OK	`TektonPipelineList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new TektonPipeline

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

Body parameters

Parameter	Type	Description
`body`	`TektonPipeline` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TektonPipeline` schema

HTTP code	Response body
201 - Created	`TektonPipeline` schema
202 - Accepted	`TektonPipeline` 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
`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 TektonPipeline

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified TektonPipeline

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

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.

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified TektonPipeline

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The

Parameter	Type	Description
		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	`TektonPipeline` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonPipeline

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified TektonPipeline

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

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

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

TektonTrigger [operator.tekton.dev/v1alpha1]

Description

TektonTrigger is the Schema for the tektontriggers 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 ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

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`	TektonTriggerSpec defines the desired state of TektonTrigger
`status`	`object`	TektonTriggerStatus defines the observed state of TektonTrigger

.spec

Description

TektonTriggerSpec defines the desired state of TektonTrigger

Type

`object`

Required

`options`

Property	Type	Description
`config`	`object`	Config holds the configuration for resources created by TektonTrigger

Property	Type	Description
`default-service-account`	`string`	
`enable-api-fields`	`string`	
`options`	`object`	options holds additions fields and these fields will be updated on the manifests
`targetNamespace`	`string`	TargetNamespace is where resources will be installed

.spec.config

Description

Config holds the configuration for resources created by TektonTrigger

Type

`object`

Property	Type	Description
`nodeSelector`	`object`	
`priorityClassName`	`string`	PriorityClassName holds the priority class to be set to pod template
`tolerations`	`array`	

.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 and Equal. Defaults to Equal. Exists is equivalent to wildcard

Property	Type	Description
		for value, so that a pod can tolerate all taints of a particular category.
`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 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.
`value`	`string`	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
`configMaps`	`object`	
`deployments`	`object`	
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	

Property	Type	Description
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

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.
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`tektonInstallerSet`	`string`	The current installer set name
`version`	`string`	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).

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

Body parameters

Parameter	Type	Description
`body`	`TektonTrigger` schema	`application/json` formatted

HTTP responses

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

HTTP method

`PATCH`

Description

partially update 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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore:

Parameter	Type	Description
		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	`TektonTrigger` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace 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

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

Body parameters

Parameter	Type	Description
`body`	`TektonTrigger` schema	`application/json` formatted

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of 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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields.

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	`TektonTrigger` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of 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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

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`	`TektonTrigger` schema	`application/json` formatted

HTTP responses

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

.spec

Description

TektonChainSpec defines the desired state of TektonChain

Type

`object`

Required

`disabled` `options`

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

Property	Type	Description
`disabled`	`boolean`	enable or disable chains feature
`generateSigningSecret`	`boolean`	generate signing key
`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
`signers.kms.auth.address`	`string`	
`signers.kms.auth.oidc.path`	`string`	
`signers.kms.auth.oidc.role`	`string`	
`signers.kms.auth.spire.audience`	`string`	
`signers.kms.auth.spire.sock`	`string`	
`signers.kms.auth.token`	`string`	
`signers.kms.auth.token-path`	`string`	
`signers.kms.kmsref`	`string`	kms signer config

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

Property	Type	Description
`transparency.enabled`	`string`	
`transparency.url`	`string`	

.spec.config

Description

Config holds the configuration for resources created by TektonChain

Type

`object`

Property	Type	Description
`nodeSelector`	`object`	
`priorityClassName`	`string`	PriorityClassName holds the priority class to be set to pod template
`tolerations`	`array`	

.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 and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
`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 default, it is not set, which means tolerate the taint

Property	Type	Description
		forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
`value`	`string`	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
`name`	`string`	Name of the environment variable. Must be a C_IDENTIFIER.
`value`	`string`	Variable references $(VAR_NAME) are expanded using the previously defined environment variables in the

Property	Type	Description
		container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double $$ are reduced to a single $, which allows for escaping the $(VAR_NAME) syntax: i.e. "$$(VAR_NAME)" will produce the string literal "$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
`valueFrom`	`object`	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
`configMapKeyRef`	`object`	Selects a key of a ConfigMap.

Property	Type	Description
`fieldRef`	`object`	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.
`resourceFieldRef`	`object`	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.
`secretKeyRef`	`object`	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
`key`	`string`	The key to select.
`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 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
`apiVersion`	`string`	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
`fieldPath`	`string`	Path of the field to select in the specified API version.

.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
`containerName`	`string`	Container name: required for volumes, optional for env vars
`divisor`		Specifies the output format of the exposed resources, defaults to "1"
`resource`	`string`	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
`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.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
`deployments`	`object`	
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

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
`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/05d67e427c722a2a5 L87↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca4 go/rest/config.go#L45-L46↗
`replicas`	`integer`	
`statefulset-ordinals`	`boolean`	if is true, enable StatefulsetOrdinals mode

Property	Type	Description
`threads-per-controller`	`integer`	The number of workers to use when processing the component

.status

Description

TektonChainStatus defines the observed state of TektonChain

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.
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`tektonInstallerSet`	`string`	The current installer set name for TektonChain

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

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
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	`TektonChain` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

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

HTTP responses

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

HTTP method

`PUT`

Description

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

Body parameters

Parameter	Type	Description
`body`	`TektonChain` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TektonChain` schema

HTTP code	Response body
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
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of 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

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified TektonChain

Query parameters

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

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

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

.spec

Type

`object`

Required

`options`

Property	Type	Description
`api`	`object`	
`catalogs`	`array`	
`categories`	`array`	

Property	Type	Description
`customLogo`	`object`	The Base64 Encode data and mediaType of the Custom Logo
`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

Property	Type	Description
`routeHostUrl`	`string`	
`secret`	`string`	

.spec.catalogs

Type

`array`

.spec.catalogs[]

Type

`object`

Property	Type	Description
`contextDir`	`string`	
`name`	`string`	
`org`	`string`	
`provider`	`string`	
`revision`	`string`	
`sshUrl`	`string`	
`type`	`string`	
`url`	`string`	

.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
`base64Data`	`string`	
`mediaType`	`string`	

.spec.db

Type

`object`

Property	Type	Description
`secret`	`string`	

.spec.default

Type

`object`

Property	Type	Description
`scopes`	`array`	

.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`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

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

Property	Type	Description
`hubInstallerSets`	`object`	The current installer set name
`manifests`	`array`	The url links of the manifests, separated by comma
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`uiUrl`	`string`	The URL route for UI which needs to be exposed
`version`	`string`	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
`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.

.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	`Status↗` schema
401 - Unauthorized	Empty

HTTP method

`GET`

Description

list objects of kind TektonHub

HTTP responses

HTTP code	Response body
200 - OK	`TektonHubList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new TektonHub

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last

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
`body`	`TektonHub` schema	`application/json` formatted

HTTP responses

HTTP code	Response body
200 - OK	`TektonHub` schema
201 - Created	`TektonHub` schema
202 - Accepted	`TektonHub` 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
`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 TektonHub

HTTP responses

HTTP code	Response body
200 - OK	`TektonHub` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonHub

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

HTTP responses

HTTP code	Response body
200 - OK	`TektonHub` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TektonHub

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

Body parameters

Parameter	Type	Description
`body`	`TektonHub` schema	`application/json` formatted

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified TektonHub

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

HTTP responses

HTTP code	Response body
200 - OK	`TektonHub` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonHub

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

Body parameters

Parameter	Type	Description
`body`	`TektonHub` schema	`application/json` formatted

HTTP responses

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

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`	TektonResultSpec defines the desired state of TektonResult
`status`	`object`	TektonResultStatus defines the observed state of TektonResult

.spec

Description

TektonResultSpec defines the desired state of TektonResult

Type

`object`

Required

`disabled` `is_external_db` `options`

Property	Type	Description
`auth_disable`	`boolean`	
`auth_impersonate`	`boolean`	

Property	Type	Description
`config`	`object`	Config holds the configuration for resources created by TektonResult
`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`	
`gcs_creds_secret_name`	`string`	
`is_external_db`	`boolean`	
`log_level`	`string`	

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`	
`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`	
`storage_emulator_host`	`string`	
`targetNamespace`	`string`	TargetNamespace is where resources will be installed
`tls_hostname_override`	`string`	

.spec.config

Description

Config holds the configuration for resources created by TektonResult

Type

`object`

Property	Type	Description
`nodeSelector`	`object`	
`priorityClassName`	`string`	PriorityClassName holds the priority class to be set to pod template
`tolerations`	`array`	

.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 and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
`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 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.
`value`	`string`	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
`configMaps`	`object`	
`deployments`	`object`	
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

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
`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/05d67e427c722a2a5 L87↗ defaults:

Property	Type	Description
		https://github.com/tektoncd/pipeline/blob/34618964300620dca4 go/rest/config.go#L45-L46 ↗
`replicas`	`integer`	
`statefulset-ordinals`	`boolean`	if is true, enable StatefulsetOrdinals mode
`threads-per-controller`	`integer`	The number of workers to use when processing the component

.status

Description

TektonResultStatus defines the observed state of TektonResult

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
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`tektonInstallerSet`	`string`	The current installer set name for TektonResult
`version`	`string`	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
`type`	`string`	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	`TektonResultList` schema
401 - Unauthorized	Empty

HTTP method

`POST`

Description

create a new TektonResult

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

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
`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 TektonResult

HTTP responses

HTTP code	Response body
200 - OK	`TektonResult` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified TektonResult

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

HTTP responses

HTTP code	Response body
200 - OK	`TektonResult` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace the specified TektonResult

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

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

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
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	`TektonResult` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonResult

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

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	`TektonResult` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonResult

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object,

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`	`TektonResult` schema	`application/json` formatted

HTTP responses

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

TektonInstallerSet [operator.tekton.dev/v1alpha1]

Description

TektonInstallerSet is the Schema for the TektonInstallerSet 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 ↗
`kind`	`string`	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info:

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`	TektonInstallerSetSpec defines the desired state of TektonInstallerSet
`status`	`object`	TektonInstallerSetStatus defines the observed state of TektonInstallerSet

.spec

Description

TektonInstallerSetSpec defines the desired state of TektonInstallerSet

Type

`object`

Property	Type	Description
`manifests`	`array`	Slice is a Source comprised of existing objects

.spec.manifests

Description

Slice is a Source comprised of existing objects

Type

`array`

.spec.manifests[]

Type

`object`

.status

Description

TektonInstallerSetStatus defines the observed state of TektonInstallerSet

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.
`observedGeneration`	`integer`	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

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

Parameter	Type	Description
		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
202 - Accepted	`TektonInstallerSet` 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
`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

Parameter	Type	Description
		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 TektonInstallerSet

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified TektonInstallerSet

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

replace the specified TektonInstallerSet

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

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

HTTP code	Response body
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	`TektonInstallerSet` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of the specified TektonInstallerSet

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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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	`TektonInstallerSet` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

replace status of the specified TektonInstallerSet

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

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

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

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`

Required

`options`

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

Property	Type	Description
`options`	`object`	options holds additions fields and these fields will be updated on the manifests
`settings`	`object`	
`targetNamespace`	`string`	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
`nodeSelector`	`object`	
`priorityClassName`	`string`	PriorityClassName holds the priority class to be set to pod template
`tolerations`	`array`	

.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
`operator`	`string`	Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
`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 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.
`value`	`string`	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
`configMaps`	`object`	
`deployments`	`object`	

Property	Type	Description
`disabled`	`boolean`	
`horizontalPodAutoscalers`	`object`	
`statefulSets`	`object`	
`webhookConfigurationOptions`	`object`	

.spec.options.configMaps

Type

`object`

.spec.options.deployments

Type

`object`

.spec.options.horizontalPodAutoscalers

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.
`conditions`	`array`	Conditions the latest available observations of a resource's current state.
`observedGeneration`	`integer`	ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.
`version`	`string`	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}/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 processing of the request. Valid values are: - All: all dry run stages will be processed
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server

Parameter	Type	Description
		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
202 - Accepted	`OpenShiftPipelinesAsCode` schema
401 - Unauthorized	Empty

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

Parameter	Type	Description
		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	`OpenShiftPipelinesAsCode` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update the specified OpenShiftPipelinesAsCode

Query parameters

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

HTTP responses

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

HTTP method

`PUT`

Description

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

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

HTTP code	Response body
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	`OpenShiftPipelinesAsCode` schema
401 - Unauthorized	Empty

HTTP method

`PATCH`

Description

partially update status of 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
`fieldValidation`	`string`	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing

Parameter	Type	Description
		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	`OpenShiftPipelinesAsCode` schema
401 - Unauthorized	Empty

HTTP method

`PUT`

Description

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

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

HTTP code	Response body
401 - Unauthorized	Empty