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]

`operator.tekton.dev` group

TektonConfig is the Schema for the TektonConfigs API

`v1alpha1` version

▼ spec `object`

TektonConfigSpec defines the desired state of TektonConfig

▼ addon `object`

Addon holds the addons config

▼ enablePipelinesAsCode `boolean`

Deprecated, will be removed in further release EnablePAC field defines whether to install PAC

▼ params `[]object`

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

▼ name `string`

▼ value `string`

▼ chain `object`

Chain holds the customizable option for chains component

▼ 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 `[]object`

EnvVar represents an environment variable present in a Container.

▼ name `string` required

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 "".

▼ valueFrom `object`

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

▼ **configMapKeyRef** `object`

Selects a key of a ConfigMap.

▼ **key** `string` required

The key to select.

▼ **name** `string`

Name of the referent. 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

▼ **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.

▼ **apiVersion** `string`

Version of the schema the FieldPath is written in terms of, defaults to "v1".

▼ **fieldPath** `string` required

Path of the field to select in the specified API version.

▼ **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.

▼ **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

Required: resource to select

▼ **secretKeyRef** `object`

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

▼ **key** `string` required

The key of the secret to select from. Must be a valid secret key.

▼ **name** `string`

Name of the referent. 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

▼ **disabled** `boolean` required

enable or disable chains feature

▼ **generateSigningSecret** `boolean`

generate signing key

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **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

▼ 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`

▼ **config** `object`

Config holds the configuration for resources created by TektonConfig

▼ **nodeSelector** `object`

▼ priorityClassName `string`

PriorityClassName holds the priority class to be set to pod template

▼ tolerations `[]object`

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

▼ 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.

▼ **dashboard** `object`

Dashboard holds the customizable options for dashboards component

▼ **external-logs** `string`

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **readonly** `boolean` required

Readonly when set to true configures the Tekton dashboard in read-only mode

▼ **hub** `object`

Hub holds the hub config

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **params** `[]object`

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

▼ **name** `string`

▼ **value** `string`

▼ **params** `[]object`

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

▼ **name** `string`

▼ **value** `string`

▼ **pipeline** `object`

Pipeline holds the customizable option for pipeline component

▼ **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`

▼ **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`

▼ **enable-step-actions** `boolean`

▼ **enable-tekton-oci-bundles** `boolean`

not in use, see: https://github.com/tektoncd/pipeline/pull/7789 this field is removed from pipeline component keeping here to maintain the API 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` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ params `[]object`

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

▼ name `string`

▼ value `string`

▼ performance `object`

PipelinePerformanceProperties defines the fields which are configurable to tune the performance of pipelines controller

▼ buckets `integer`

▼ disable-ha `boolean` required

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 client actually the number multiplied by 2 https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a57e58328d7097e21429b7524/cmd/controller/main.go#L85-L87↗ defaults:

https://github.com/tektoncd/pipeline/blob/34618964300620dca44d10a595e4af84e9903a55/vendor/k8s.io/client-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 pipelines controller's work queue

▼ **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`

▼ **platforms** `object`

Platforms allows configuring platform specific configurations

▼ **openshift** `object`

OpenShift allows configuring openshift specific components and configurations

▼ **pipelinesAsCode** `object`

PipelinesAsCode allows configuring PipelinesAsCode configurations

▼ **additionalPACControllers** `object`

AdditionalPACControllers allows to deploy additional PAC controller

▼ **enable** `boolean`

Enable or disable pipelines as code by changing this bool

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **settings** `object`

▼ **scc** `object`

SCC allows configuring security context constraints used by workloads

▼ **default** `string`

Default contains the default SCC that will be attached to the service account used for workloads (`pipeline` SA by default) and defined in PipelineProperties.OptionalPipelineProperties.DefaultServiceAccount

▼ **maxAllowed** `string`

MaxAllowed specifies the highest SCC that can be requested for in a namespace or in the Default field.

▼ profile `string`

▼ pruner `object`

Pruner holds the prune config

▼ disabled `boolean` required

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 `[]string`

The resources which need to be pruned

▼ **schedule** `string`

How frequent pruning should happen

▼ **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.

▼ **targetNamespace** `string`

TargetNamespace is where resources will be installed

▼ **targetNamespaceMetadata** `object`

holds target namespace metadata

▼ **annotations** `object`

▼ **labels** `object`

▼ **trigger** `object`

Trigger holds the customizable option for triggers component

▼ **default-service-account** `string`

▼ **enable-api-fields** `string`

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **status** `object`

TektonConfigStatus defines the observed state of TektonConfig

▼ **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 `[]object`

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 ↗

▼ 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` required

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

▼ type `string` required

Type of condition.

▼ 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

TektonInstallerSet [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonInstallerSet is the Schema for the TektonInstallerSet API

`v1alpha1` version

▼ spec `object`

TektonInstallerSetSpec defines the desired state of TektonInstallerSet

▼ manifests `[]object`

▼ status `object`

TektonInstallerSetStatus defines the observed state of TektonInstallerSet

▼ 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 `[]object`

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 ↗

▼ **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` required

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

▼ **type** `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

TektonPipeline [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonPipeline is the Schema for the tektonpipelines API

`v1alpha1` version

▼ **spec** `object`

TektonPipelineSpec defines the desired state of TektonPipeline

▼ **await-sidecar-readiness** `boolean`

▼ **bundles-resolver-config** `object`

▼ **cluster-resolver-config** `object`

▼ **config** `object`

Config holds the configuration for resources created by TektonPipeline

▼ **nodeSelector** `object`

▼ **priorityClassName** `string`

PriorityClassName holds the priority class to be set to pod template

▼ tolerations `[]object`

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

▼ 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.

▼ **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`

▼ **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/7789 ↗ this field is removed from pipeline component keeping here to maintain the API 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` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **params** `[]object`

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

▼ **name** `string`

▼ **value** `string`

▼ **performance** `object`

PipelinePerformanceProperties defines the fields which are configurable to tune the performance of pipelines controller

▼ **buckets** `integer`

▼ **disable-ha** `boolean` required

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 client actually the number multiplied by 2 https://github.com/pierretasci/pipeline/blob/05d67e427c722a2a57e58328d7097e21429b7524/cmd/controller/main.go#L85-L87↗ defaults: https://github.com/tektoncd/pipeline/blob/34618964300620dca44d10a595e4af84e9903a55/vendor/k8s.io/client-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 pipelines controller's work queue

▼ **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`

▼ **targetNamespace** `string`

TargetNamespace is where resources will be installed

▼ **trusted-resources-verification-no-match-policy** `string`

▼ **verification-mode** `string`

▼ **status** `object`

TektonPipelineStatus defines the observed state of TektonPipeline

▼ **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** `[]object`

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 ↗

▼ **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` required

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

▼ type `string` required

Type of condition.

▼ extTektonInstallerSets `object`

The installer sets created for extension components

▼ 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

TektonTrigger [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonTrigger is the Schema for the tektontriggers API

`v1alpha1` version

▼ spec `object`

TektonTriggerSpec defines the desired state of TektonTrigger

▼ config `object`

Config holds the configuration for resources created by TektonTrigger

▼ nodeSelector `object`

▼ priorityClassName `string`

PriorityClassName holds the priority class to be set to pod template

▼ tolerations `[]object`

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

▼ 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.

▼ **default-service-account** `string`

▼ **enable-api-fields** `string`

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **targetNamespace** `string`

TargetNamespace is where resources will be installed

▼ **status** `object`

TektonTriggerStatus defines the observed state of TektonTrigger

▼ 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 `[]object`

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 ↗

▼ 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` required

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

▼ **type** `string` required

Type of condition.

▼ **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

TektonChain [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonChain is the Schema for the tektonchain API

`v1alpha1` version

▼ spec `object`

TektonChainSpec defines the desired state of TektonChain

▼ 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

▼ **nodeSelector** `object`

▼ **priorityClassName** `string`

PriorityClassName holds the priority class to be set to pod template

▼ tolerations `[]object`

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

▼ 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.

▼ controllerEnvs `[]object`

EnvVar represents an environment variable present in a Container.

▼ name `string` required

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 "".

▼ valueFrom `object`

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

▼ configMapKeyRef `object`

Selects a key of a ConfigMap.

▼ key `string` required

The key to select.

▼ **name** `string`

Name of the referent. 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

▼ **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.

▼ **apiVersion** `string`

Version of the schema the FieldPath is written in terms of, defaults to "v1".

▼ **fieldPath** `string` required

Path of the field to select in the specified API version.

▼ **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.

▼ **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

Required: resource to select

▼ **secretKeyRef** `object`

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

▼ **key** `string` required

The key of the secret to select from. Must be a valid secret key.

▼ **name** `string`

Name of the referent. 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

▼ **disabled** `boolean` required

enable or disable chains feature

▼ **generateSigningSecret** `boolean`

generate signing key

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **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

▼ **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

▼ **transparency.enabled** `string`

▼ **transparency.url** `string`

▼ **status** `object`

TektonChainStatus defines the observed state of TektonChain

▼ **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** `[]object`

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 ↗

▼ **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` required

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

▼ **type** `string` required

Type of condition.

▼ **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

▼ version `string`

The version of the installed release

TektonHub [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonHub is the Schema for the tektonhub API

`v1alpha1` version

▼ **spec** `object`

▼ **api** `object`

▼ **catalogRefreshInterval** `string`

▼ **hubConfigUrl** `string`

Deprecated, will be removed in further release

▼ **routeHostUrl** `string`

▼ **secret** `string`

▼ **catalogs** `[]object`

▼ **contextDir** `string`

▼ **name** `string`

▼ **org** `string`

▼ **provider** `string`

▼ **revision** `string`

▼ **sshUrl** `string`

▼ **type** `string`

▼ **url** `string`

▼ **categories** `[]string`

▼ **customLogo** `object`

The Base64 Encode data and mediaType of the Custom Logo

▼ **base64Data** `string`

▼ **mediaType** `string`

▼ **db** `object`

▼ **secret** `string`

▼ **default** `object`

▼ **scopes** `[]string`

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **params** `[]object`

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

▼ **name** `string`

▼ **value** `string`

▼ **scopes** `[]object`

▼ **name** `string`

▼ **users** `[]string`

▼ **targetNamespace** `string`

TargetNamespace is where resources will be installed

▼ **status** `object`

TektonHubStatus defines the observed state of TektonHub

▼ 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 `[]object`

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 ↗

▼ 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` required

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

▼ **type** `string` required

Type of condition.

▼ **hubInstallerSets** `object`

The current installer set name

▼ **manifests** `[]string`

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

TektonResult [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonResult is the Schema for the tektonresults API

`v1alpha1` version

▼ **spec** `object`

TektonResultSpec defines the desired state of TektonResult

▼ **auth_disable** `boolean`

▼ **auth_impersonate** `boolean`

▼ **db_enable_auto_migration** `boolean`

▼ **db_host** `string`

▼ **db_name** `string`

▼ **db_port** `integer`

▼ **db_sslmode** `string`

▼ **db_sslrootcert** `string`

▼ **gcs_bucket_name** `string`

▼ **gcs_creds_secret_key** `string`

▼ **gcs_creds_secret_name** `string`

▼ **is_external_db** `boolean` required

▼ **log_level** `string`

▼ **logging_plugin_api_url** `string`

▼ **logging_plugin_ca_cert** `string`

▼ **logging_plugin_forwarder_delay_duration** `integer`

▼ **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` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **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`

▼ **status** `object`

TektonResultStatus defines the observed state of TektonResult

▼ **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** `[]object`

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 ↗

▼ 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` required

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

▼ type `string` required

Type of condition.

▼ 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

TektonInstallerSet [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

TektonInstallerSet is the Schema for the TektonInstallerSet API

`v1alpha1` version

▼ spec `object`

TektonInstallerSetSpec defines the desired state of TektonInstallerSet

▼ manifests `[]object`

▼ status `object`

TektonInstallerSetStatus defines the observed state of TektonInstallerSet

▼ 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 `[]object`

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 ↗

▼ **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` required

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

▼ **type** `string` required

Type of condition.

▼ observedGeneration `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

OpenShift Pipelines as Code [operator.tekton.dev/v1alpha1]

`operator.tekton.dev` group

OpenShiftPipelinesAsCode is the Schema for the OpenShiftPipelinesAsCode API

`v1alpha1` version

▼ spec `object`

OpenShiftPipelinesAsCodeSpec defines the desired state of OpenShiftPipelinesAsCode

▼ additionalPACControllers `object`

AdditionalPACControllers allows to deploy additional PAC controller

▼ config `object`

▼ nodeSelector `object`

▼ priorityClassName `string`

PriorityClassName holds the priority class to be set to pod template

▼ tolerations `[]object`

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

▼ **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.

▼ **options** `object` required

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

▼ **configMaps** `object`

▼ **deployments** `object`

▼ **disabled** `boolean`

▼ **horizontalPodAutoscalers** `object`

▼ **statefulSets** `object`

▼ **webhookConfigurationOptions** `object`

▼ **settings** `object`

▼ **targetNamespace** `string`

TargetNamespace is where resources will be installed

▼ status `object`

OpenShiftPipelinesAsCodeStatus defines the observed state of OpenShiftPipelinesAsCode

▼ 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 `[]object`

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 ↗

▼ 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` required

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

▼ **type** `string` required

Type of condition.

▼ **observedGeneration** `integer`

ObservedGeneration is the 'Generation' of the Service that was last processed by the controller.

▼ **version** `string`

The version of the installed release