



Analytical Database APIs

[ClickHouseInstallations](#)

[ClickHouseInstallationTemplate](#)

[ClickHouse](#)



ClickHouseInstallations

Description

define a set of Kubernetes resources (StatefulSet, PVC, Service, ConfigMap) which describe behavior one or more ClickHouse clusters

Type

`object`

Required

`spec`

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint

Property	Type	Description
		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.
status	object	Current ClickHouseInstallation manifest status, contains many fields like a normalized configuration, clickhouse-operator version, current action and all applied action list, current taskID and all applied taskIDs and other
spec	object	Specification of the desired behavior of one or more ClickHouse clusters More info: https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md

.status

Description

Current ClickHouseInstallation manifest status, contains many fields like a normalized configuration, clickhouse-operator version, current action and all applied action list, current taskID and all applied taskIDs and other

Type

object

Property	Type	Description
chop-version	string	ClickHouse operator version
chop-commit	string	ClickHouse operator git commit SHA
chop-date	string	ClickHouse operator build date
chop-ip	string	IP address of the operator's pod which managed this CHI
clusters	integer	Clusters count
shards	integer	Shards count
replicas	integer	Replicas count
hosts	integer	Hosts count
status	string	Status
taskID	string	Current task id

Property	Type	Description
<code>taskIDsStarted</code>	<code>array</code>	Started task ids
<code>taskIDsCompleted</code>	<code>array</code>	Completed task ids
<code>action</code>	<code>string</code>	Action
<code>actions</code>	<code>array</code>	Actions
<code>error</code>	<code>string</code>	Last error
<code>errors</code>	<code>array</code>	Errors
<code>updated</code>	<code>integer</code>	Updated Hosts count
<code>added</code>	<code>integer</code>	Added Hosts count
<code>deleted</code>	<code>integer</code>	Deleted Hosts count
<code>delete</code>	<code>integer</code>	About to delete Hosts count

Property	Type	Description
pods	array	Pods
pod-ips	array	Pod IPs
fqdns	array	Pods FQDNs
endpoint	string	Endpoint
generation	integer	Generation
normalized	object	Normalized CHI requested
normalizedCompleted	object	Normalized CHI completed

.status.taskIDsStarted

Description

Started task ids

Type

array

.status.taskIDsStarted[]

Type

string

.status.taskIDsCompleted

Description

Completed task ids

Type

array

.status.taskIDsCompleted[]

Type

string

.status.actions

Description

Actions

Type

array

.status.actions[]

Type

string

.status.errors

Description

Errors

Type

array

.status.errors[]

Type

string

.status.pods

Description

Pods

Type

array

.status.pods[]

Type

string

.status.pod-ips

Description

Pod IPs

Type

array

.status.pod-ips[]

Type

string

.status.fqdns

Description

Pods FQDNs

Type

array

.status.fqdns[]

Type

string

.status.normalized

Description

Normalized CHI requested

Type

object

.status.normalizedCompleted

Description

Normalized CHI completed

Type

object

.spec

Description

Specification of the desired behavior of one or more ClickHouse clusters More info:
https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md

Type

object

Property	Type	Description
taskID	string	Allows to define custom taskID for named update o status of this update execution in .status.taskIDs fie update of chi manifest will generate random taskID
stop	string	Allow stop all ClickHouse clusters described in curr mechanism works as follows: - When stop is 1 then setup Replicas: 0 current chi StatefulSet resource, all Pods a resources will descstroy, but PVCs still live - When stop is 0 then Pods will created ag retained PVCs and Service also will created
restart	string	This is a 'soft restart' button. When set to 'RollingUp restart ClickHouse pods in a graceful way. Remove order to avoid unneeded restarts
troubleshoot	string	allows troubleshoot Pods during CrashLoopBack si wrong configuration, clickhouse-server wouldr
namespaceDomainPattern	string	custom domain suffix which will add to end of Ser use it when you use custom cluster domain in your

Property	Type	Description
templating	object	optional, define policy for auto applying ClickHouse inside ClickHouseInstallation
reconciling	object	optional, allows tuning reconciling cycle for Clickho clickhouse-operator side
defaults	object	define default behavior for whole ClickHouseInstall can be re-define on cluster, shard and replica level https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_expla
configuration	object	allows configure multiple aspects and behavior for <code>server</code> instance and also allows describe multiple <code>server</code> clusters inside one <code>chi</code> resource
templates	object	allows define templates which will use for render K8 like StatefulSet, ConfigMap, Service, PVC, by default operator have own templates, but you can override
useTemplates	array	list of <code>ClickHouseInstallationTemplate</code> (chit) r will merge with current <code>Chi</code> manifest during render resources to create related ClickHouse clusters

.spec.templating

Description

optional, define policy for auto applying ClickHouseInstallationTemplate inside ClickHouseInstallation

Type

object

Property	Type	Description
policy	string	when defined as <code>auto</code> inside ClickhouseInstallationTemplate, it will auto add into all ClickHouseInstallation, manual value is default

.spec.reconciling

Description

optional, allows tuning reconciling cycle for ClickhouseInstallation from clickhouse-operator side

Type

object

Property	Type	Description
policy	string	DEPRECATED
configMapPropagationTimeout	integer	timeout in seconds when <code>clickhouse-operator</code> applied <code>ConfigMap</code> during reconcile <code>ClickHouseInstallation</code> pods will updated from cache see details: https://kubernetes.io/docs/concepts/configuration/configmaps-are-updated-automatically

Property	Type	Description
<code>cleanup</code>	<code>object</code>	optional, define behavior for cleanup Kubernetes resources during reconcile cycle

`.spec.reconciling.cleanup`

Description

optional, define behavior for cleanup Kubernetes resources during reconcile cycle

Type

`object`

Property	Type	Description
<code>unknownObjects</code>	<code>object</code>	what clickhouse-operator shall do when found Kubernetes resources which should be managed with clickhouse-operator, but not have <code>ownerReference</code> to any currently managed <code>ClickHouseInstallation</code> resource, default behavior is <code>Delete</code>
<code>reconcileFailedObjects</code>	<code>object</code>	what clickhouse-operator shall do when reconciling Kubernetes resources are failed, default behavior is <code>Retain</code>

`.spec.reconciling.cleanup.unknownObjects`

Description

what clickhouse-operator shall do when found Kubernetes resources which should be managed with clickhouse-operator, but not have `ownerReference` to any currently

managed `ClickHouseInstallation` resource, default behavior is `Delete`

Type

object

Property	Type	Description
statefulSet	string	behavior policy for unknown StatefulSet, Delete by default
pvc	string	behavior policy for unknown PVC, Delete by default
configMap	string	behavior policy for unknown ConfigMap, Delete by default
service	string	behavior policy for unknown Service, Delete by default

.spec.reconciling.cleanup.reconcileFailedObjects

Description

what clickhouse-operator shall do when reconciling Kubernetes resources are failed, default behavior is `Retain`

Type

object

Property	Type	Description
statefulSet	string	behavior policy for failed StatefulSet reconciling, Retain by default

Property	Type	Description
<code>pvc</code>	<code>string</code>	behavior policy for failed PVC reconciling, Retain by default
<code>configMap</code>	<code>string</code>	behavior policy for failed ConfigMap reconciling, Retain by default
<code>service</code>	<code>string</code>	behavior policy for failed Service reconciling, Retain by default

.spec.defaults

Description

define default behavior for whole ClickHouseInstallation, some behavior can be re-define on cluster, shard and replica level More info: https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specdefaults

Type

`object`

Property	Type	Description
<code>replicasUseFQDN</code>	<code>string</code>	define should replicas be specified by FQDN in <code><host></host></code> . In case of "no" will use short hostname and clickhouse-server will use kubernetes default suffixes for DNS lookup "yes" by default

Property	Type	Description
distributedDDL	object	allows change <code><yandex><distributed_ddl></code> <code></distributed_ddl></yandex></code> settings More info: https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings-distributed_ddl ↗
storageManagement	object	default storage management options
templates	object	optional, configuration of the templates names which will use for generate Kubernetes resources according to one or more ClickHouse clusters described in current ClickHouseInstallation (chi) resource

.spec.defaults.distributedDDL

Description

allows change ``<yandex><distributed_ddl></distributed_ddl></yandex>`` settings More info:
https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings-distributed_ddl

Type

object

Property	Type	Description
profile	string	Settings from this profile will be used to execute DDL queries

.spec.defaults.storageManagement

Description

default storage management options

Type

object

Property	Type	Description
provisioner	string	defines <code>PVC</code> provisioner - be it StatefulSet or the Operator
reclaimPolicy	string	defines behavior of <code>PVC</code> deletion. <code>Delete</code> by default, if <code>Retain</code> specified then <code>PVC</code> will be kept when deleting StatefulSet

.spec.defaults.templates

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to one or more ClickHouse clusters described in current ClickHouseInstallation (chi) resource

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from chi.spec.templates.hostTemplates, which will apply to configure every <code>clickhouse-</code>

Property	Type	Description
		<code>server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource

Property	Type	Description
		which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
volumeClaimTemplate	string	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.configuration

Description

allows configure multiple aspects and behavior for `clickhouse-server` instance and also allows describe multiple `clickhouse-server` clusters inside one `chi` resource

Type

object

Property	Type	Description
zookeeper	object	allows configure .. section in each Pod during generate ConfigMap mounted in /etc/clickhouse-server/config.d/ clickhouse-operator doesn't manage Zookeeper, please install Zookeeper separately! https://github.com/Altinity/clickhouse-operator/tree/master/deploy currently, zookeeper (or clickhouse-keeper replacement) used for *ReplicatedMergeTree table engines and for distributed_ddl https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings_zookeeper
users	object	allows configure .. section in each Pod during generate ConfigMap mount in /etc/clickhouse-server/users.d/ you can configure hashed, authorization restrictions, database level security row filter https://clickhouse.tech/docs/en/operations/settings/settings-user

Property	Type	Description
		will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spe
profiles	object	allows configure .. section in each <code>Pod</code> during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/users.d/</code> you can configure settings profile More details: https://clickhouse.tech/docs/en/operations/settings/settings-profiles will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spe
quotas	object	allows configure .. section in each <code>Pod</code> during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/users.d/</code> you can configure resource quotas More details: https://clickhouse.tech/docs/en/operations/settings/settings-quotas Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spe
settings	object	allows configure <code>clickhouse-server</code> settings inside ... tag in generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/settings.d/</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings-configuration code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spe
files	object	allows define content of any setting file inside each <code>Pod</code> during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/settings.d/</code>

Property	Type	Description
		every key in this object is the file name every value in this object you can use <code>!!binary</code> and base64 for binary files, see details https://yaml.org/type/binary.html each key could contains prefixes COMMON, HOST or config.d, users.d, cond.d, wrong prefixes will also be ignored More details: https://github.com/Altinity/clickhouse-operator/blob/master/docs/chi-examples/05-settings-05-files-net
<code>clusters</code>	<code>array</code>	describes ClickHouse clusters layout and allows change settings: shard-level and replica-level every cluster is a set of StatefulSet contains only one Pod with <code>clickhouse-server</code> all Pods will run <code><remote_server></code> part of ClickHouse configs, mounted from <code>/etc/clickhouse-server/config.d/chop-generated-remote</code> Clusters will use for Distributed table engine, more details: https://clickhouse.tech/docs/en/engines/table-engines/special/distributed <code>cluster</code> contains zookeeper settings (could be inherited from <code>zookeeper</code> when you can create <code>*ReplicatedMergeTree</code> tables

.spec.configuration.zookeeper

Description

allows configure `<yandex><zookeeper>..</zookeeper></yandex>` section in each `Pod` during generate `ConfigMap` which will be mounted in `/etc/clickhouse-server/config.d/` `clickhouse-operator` itself doesn't manage Zookeeper, please install Zookeeper separately look examples on <https://github.com/Altinity/clickhouse-operator/tree/master/deploy/zookeeper/> currently, zookeeper (or clickhouse-keeper replacement) used for `*ReplicatedMergeTree` table engines and for `distributed_ddl` More details: https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings_zookeeper

Type

`object`

Property	Type	Description
nodes	array	describe every available zookeeper cluster node for interaction
session_timeout_ms	integer	session timeout during connect to Zookeeper
operation_timeout_ms	integer	one operation timeout during Zookeeper transactions
root	string	optional root znode path inside zookeeper to store ClickHouse related data (replication queue or distributed DDL)
identity	string	optional access credentials string with <code>user:password</code> format used when use digest authorization in Zookeeper

.spec.configuration.zookeeper.nodes

Description

describe every available zookeeper cluster node for interaction

Type

array

.spec.configuration.zookeeper.nodes[]

Type

object

Property	Type	Description
host	string	dns name or ip address for Zookeeper node
port	integer	TCP port which used to connect to Zookeeper node

.spec.configuration.users

Description

allows configure `<yandex><users>..</users></yandex>` section in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/users.d/` you can configure password hashed, authorization restrictions, database level security row filters etc. More details: <https://clickhouse.tech/docs/en/operations/settings/settings-users/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationusers

Type

object

.spec.configuration.profiles

Description

allows configure `<yandex><profiles>..</profiles></yandex>` section in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/users.d/` you can configure any aspect of settings profile More details: <https://clickhouse.tech/docs/en/operations/settings/settings-profiles/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationprofiles

Type

object

.spec.configuration.quotas

Description

allows configure `<yandex><quotas>..</quotas></yandex>` section in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/users.d/` you can configure any aspect of resource quotas More details:

<https://clickhouse.tech/docs/en/operations/quotas/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationquotas

Type

object

.spec.configuration.settings

Description

allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` More details: <https://clickhouse.tech/docs/en/operations/settings/settings/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationsettings

Type

object

.spec.configuration.files

Description

allows define content of any setting file inside each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or `/etc/clickhouse-server/conf.d/` or `/etc/clickhouse-server/users.d/` every key in this object is the file name every value in this object is the file content you can use `!!binary |` and base64 for binary files, see details here <https://yaml.org/type/binary.html> each key could contains prefix like USERS, COMMON, HOST or config.d, users.d, cond.d, wrong prefixes will ignored, subfolders also will ignored More details: <https://github.com/Altinity/clickhouse-operator/blob/master/docs/chi-examples/05-settings-05-files-nested.yaml>

Type

object

.spec.configuration.clusters

Description

describes ClickHouse clusters layout and allows change settings on cluster-level, shard-level and replica-level every cluster is a set of StatefulSet, one StatefulSet contains only one Pod with `clickhouse-server` all Pods will rendered in <remote_server> part of ClickHouse configs, mounted from ConfigMap as `etc/clickhouse-server/config.d/chop-generated-remote\_servers.xml` Clusters will use for Distributed table engine, more details: <https://clickhouse.tech/docs/en/engines/table-engines/special/distributed/> If `cluster` contains zookeeper settings (could be inherited from top `chi` level), when you can create *ReplicatedMergeTree tables

Type

array

.spec.configuration.clusters[]

Type

object

Property	Type	Description
name	string	cluster name, used to identify set of ClickHouse servers and wide used during generate names of related Kubernetes resources
zookeeper	object	optional, allows configure .. section in each Pod only in current ClickHouse cluster, during generate ConfigMap which will mounted in etc/clickhouse-

Property	Type	Description
		<code>server/config.d/</code> override top-level <code>chi.spec.configuration.zookeeper</code> settings
<code>settings</code>	<code>object</code>	optional, allows configure <code>clickhouse-server</code> settings inside ... tag in each <code>Pod</code> only in one cluster during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> override top-level <code>chi.spec.configuration.settings</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings/
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside each <code>Pod</code> on current cluster during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/users.d/</code> override top-level <code>chi.spec.configuration.files</code>
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will use for generate Kubernetes resources according to selected cluster override top-level <code>chi.spec.configuration.templates</code>
<code>schemaPolicy</code>	<code>object</code>	describes how schema is propagated within replicas and shards

Property	Type	Description
<code>secret</code>	<code>object</code>	optional, shared secret value to secure cluster communications
<code>layout</code>	<code>object</code>	describe current cluster layout, how much shards in cluster, how much replica in shard allows override settings on each shard and replica separately

.spec.configuration.clusters[].zookeeper

Description

optional, allows configure `<yandex><zookeeper>..</zookeeper></yandex>` section in each `Pod` only in current ClickHouse cluster, during generate `ConfigMap` which will mounted in `/etc/clickhouse-server/config.d/` override top-level `chi.spec.configuration.zookeeper` settings

Type

`object`

Property	Type	Description
<code>nodes</code>	<code>array</code>	describe every available zookeeper cluster node for interaction
<code>session_timeout_ms</code>	<code>integer</code>	session timeout during connect to Zookeeper
<code>operation_timeout_ms</code>	<code>integer</code>	one operation timeout during Zookeeper transactions

Property	Type	Description
<code>root</code>	<code>string</code>	optional root znode path inside zookeeper to store ClickHouse related data (replication queue or distributed DDL)
<code>identity</code>	<code>string</code>	optional access credentials string with <code>user:password</code> format used when use digest authorization in Zookeeper

.spec.configuration.clusters[].zookeeper.nodes

Description

describe every available zookeeper cluster node for interaction

Type

`array`

.spec.configuration.clusters[].zookeeper.nodes[]

Type

`object`

Property	Type	Description
<code>host</code>	<code>string</code>	dns name or ip address for Zookeeper node
<code>port</code>	<code>integer</code>	TCP port which used to connect to Zookeeper node

.spec.configuration.clusters[].settings

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in each `Pod` only in one cluster during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` override top-level `chi.spec.configuration.settings` More details: https://clickhouse.tech/docs/en/operations/settings/settings/`

Type

object

.spec.configuration.clusters[].files

Description

optional, allows define content of any setting file inside each `Pod` on current cluster during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or /etc/clickhouse-server/conf.d/` or /etc/clickhouse-server/users.d/` override top-level `chi.spec.configuration.files``

Type

object

.spec.configuration.clusters[].templates

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected cluster override top-level `chi.spec.configuration.templates`

Type

object

Property	Type	Description
<code>hostTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-</code>

Property	Type	Description
		<code>server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource

Property	Type	Description
		which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.configuration.clusters[].schemaPolicy

Description

describes how schema is propagated within replicas and shards

Type

`object`

Property	Type	Description
<code>replica</code>	<code>string</code>	how schema is propagated within a replica
<code>shard</code>	<code>string</code>	how schema is propagated between shards

.spec.configuration.clusters[].secret

Description

optional, shared secret value to secure cluster communications

Type

`object`

Property	Type	Description
auto	string	Auto-generate shared secret value to secure cluster communications
value	string	Cluster shared secret value in plain text
valueFrom	object	Cluster shared secret source

.spec.configuration.clusters[].secret.valueFrom

Description

Cluster shared secret source

Type

object

Property	Type	Description
secretKeyRef	object	Selects a key of a secret in the clickhouse installation namespace. Should not be used if value is not empty.

.spec.configuration.clusters[].secret.valueFrom.secretKeyRef

Description

Selects a key of a secret in the clickhouse installation namespace. Should not be used if value is not empty.

Type

object

Required

name

key

Property	Type	Description
name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
key	string	The key of the secret to select from. Must be a valid secret key.
optional	boolean	Specify whether the Secret or its key must be defined

.spec.configuration.clusters[].layout

Description

describe current cluster layout, how much shards in cluster, how much replica in shard
allows override settings on each shard and replica separately

Type

object

Property	Type	Description
type	string	DEPRECATED - to be removed soon
shardsCount	integer	how much shards for current ClickHouse cluster will run in Kubernetes, each shard contains shared-nothing part

Property	Type	Description
		of data and contains set of replicas, cluster contains 1 shard by default
replicasCount	integer	how much replicas in each shards for current ClickHouse cluster will run in Kubernetes, each replica is a separate <code>StatefulSet</code> which contains only one <code>Pod</code> with <code>clickhouse-server</code> instance, every shard contains 1 replica by default
shards	array	optional, allows override top-level <code>chi.spec.configuration</code> , cluster-level <code>chi.spec.configuration.clusters</code> settings for each shard separately, use it only if you fully understand what you do
replicas	array	optional, allows override top-level <code>chi.spec.configuration</code> and cluster-level <code>chi.spec.configuration.clusters</code> configuration for each replica and each shard relates to selected replica, use it only if you fully understand what you do

.spec.configuration.clusters[].layout.shards

Description

optional, allows override top-level ``chi.spec.configuration`` , cluster-level ``chi.spec.configuration.clusters`` settings for each shard separately, use it only if you fully understand what you do

Type

array

.spec.configuration.clusters[].layout.shards[]

Type

object

Property	Type	Description
name	string	optional, by default shard name is generated, but you can override it and setup custom name
definitionType	string	DEPRECATED - to be removed soon
weight	integer	optional, 1 by default, allows setup shard setting which during insert into tables with <code>Distributed</code> engine, with <remote_servers> inside ConfigMap which will mount in /etc/clickhouse-server/config.d/chop-generated-remote_servers.xml More details: https://clickhouse.tech/docs/en/engines/table-engines/special/distributed/
internalReplication	string	optional, <code>true</code> by default when <code>chi.spec.configuration.clusters[].layout.ReplicationFactor > 1</code> and 0 otherwise allows setup <internal_replication> which will use during insert into tables with <code>Distributed</code> engine for insert only in one live replica and other replicas download inserted data during replication, will apply in <remote_servers> inside ConfigMap which will mount in /etc/clickhouse-server/config.d/chop-generated-remote_servers.xml More details:

Property	Type	Description
		https://clickhouse.tech/docs/en/engines/table-engines/special/distributed/ ↗
settings	object	optional, allows configure <code>clickhouse-server</code> setting ... tag in each <code>Pod</code> only in one shard during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> override top-level <code>chi.spec.configuration.settings</code> and cluster-level <code>chi.spec.configuration.clusters.settings</code> More https://clickhouse.tech/docs/en/operations/settings/settings
files	object	optional, allows define content of any setting file inside <code>Pod</code> only in one shard during generate <code>ConfigMap</code> mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/users.d/</code> override top-level <code>chi.spec.configuration.files</code> and cluster-level <code>chi.spec.configuration.clusters.files</code>
templates	object	optional, configuration of the templates names which will generate Kubernetes resources according to selected override top-level <code>chi.spec.configuration.templates</code> cluster-level <code>chi.spec.configuration.clusters.templates</code>
replicasCount	integer	optional, how much replicas in selected shard for selected ClickHouse cluster will run in Kubernetes, each replica separate <code>StatefulSet</code> which contains only one <code>Pod</code> <code>clickhouse-server</code> instance, shard contains 1 replica

Property	Type	Description
		default override cluster-level <code>chi.spec.configuration.clusters.layout.replicas</code>
<code>replicas</code>	<code>array</code>	optional, allows override behavior for selected replicas cluster-level <code>chi.spec.configuration.clusters</code> or level <code>chi.spec.configuration.clusters.layout.sl</code>

`.spec.configuration.clusters[].layout.shards[].settings`

Description

optional, allows configure ``clickhouse-server`` settings inside `<yandex>...</yandex>` tag in each ``Pod`` only in one shard during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/config.d/`` override top-level ``chi.spec.configuration.settings`` and cluster-level ``chi.spec.configuration.clusters.settings`` More details:
<https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

`object`

`.spec.configuration.clusters[].layout.shards[].files`

Description

optional, allows define content of any setting file inside each ``Pod`` only in one shard during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/config.d/`` or ``/etc/clickhouse-server/conf.d/`` or ``/etc/clickhouse-server/users.d/`` override top-level ``chi.spec.configuration.files`` and cluster-level ``chi.spec.configuration.clusters.files``

Type

`object`

`.spec.configuration.clusters[].layout.shards[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected shard override top-level `chi.spec.configuration.templates` and cluster-level `chi.spec.configuration.clusters.templates`

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from chi.spec.templates.hostTemplates, which will apply to configure every clickhouse-server instance during render ConfigMap resources which will mount into Pod
podTemplate	string	optional, template name from chi.spec.templates.podTemplates, allows customization each Pod resource during render and reconcile each StatefulSet.spec resource described in chi.spec.configuration.clusters
dataVolumeClaimTemplate	string	optional, template name from chi.spec.templates.volumeClaimTemplates, allows customization each PVC which will mount for clickhouse data directory in each Pod during render and reconcile every StatefulSet.spec resource described in chi.spec.configuration.clusters

Property	Type	Description
logVolumeClaimTemplate	string	optional, template name from chi.spec.templates.volumeClaimTemplates, allows customization each PVC which will mount for clickhouse log directory in each Pod during render and reconcile every StatefulSet.spec resource described in chi.spec.configuration.clusters
serviceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for one Service resource which will created by clickhouse-operator which cover all clusters in whole chi resource
clusterServiceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for each Service resource which will created by clickhouse-operator which cover each clickhouse cluster described in chi.spec.configuration.clusters
shardServiceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for each Service resource which will created by clickhouse-operator which cover each shard inside

Property	Type	Description
		clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.configuration.clusters[].layout.shards[].replicas

Description

optional, allows override behavior for selected replicas from cluster-level ``chi.spec.configuration.clusters`` and shard-level ``chi.spec.configuration.clusters.layout.shards``

Type

`array`

.spec.configuration.clusters[].layout.shards[].replicas[]

Type

object

Property	Type	Description
name	string	optional, by default replica name is generated, but you and setup custom name
secure	boolean	optional, setup <code>secure</code> inside <code>clickhouse-server</code> each Pod where current template will apply if specified
tcpPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name selected replica, override <code>chi.spec.templates.hostTemplates.spec.tcpPort</code> connect to <code>clickhouse-server</code> via TCP Native protocol via kubernetes <code>Service</code>
httpPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name selected replica, override <code>chi.spec.templates.hostTemplates.spec.httpPort</code> connect to <code>clickhouse-server</code> via HTTP protocol via <code>Service</code>
interserverHTTPPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name <code>interserver</code> for selected replica, override <code>chi.spec.templates.hostTemplates.spec.interserverHTTPPort</code> allows connect between replicas inside same shard distributed replicated data parts HTTP protocol

Property	Type	Description
settings	object	optional, allows configure <code>clickhouse-server</code> setting in <code>Pod</code> only in one replica during generate <code>ConfigMap</code> mount in <code>/etc/clickhouse-server/conf.d/</code> overriding <code>chi.spec.configuration.settings</code>, cluster-level <code>chi.spec.configuration.clusters.settings</code> and <code>chi.spec.configuration.clusters.layout.shards</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings
files	object	optional, allows define content of any setting file inside one replica during generate <code>ConfigMap</code> which will mount <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/user.d/</code> top-level <code>chi.spec.configuration.files</code>, cluster-level <code>chi.spec.configuration.clusters.files</code> and shards <code>chi.spec.configuration.clusters.layout.shards</code>
templates	object	optional, configuration of the templates names which will generate Kubernetes resources according to selected top-level <code>chi.spec.configuration.templates</code>, cluster-level <code>chi.spec.configuration.clusters.templates</code> and shards <code>chi.spec.configuration.clusters.layout.shards</code>

`.spec.configuration.clusters[].layout.shards[].replicas[].settings`

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in `Pod` only in one replica during generate `ConfigMap` which will mount in `/etc/clickhouse-server/conf.d/` override top-level `chi.spec.configuration.settings``, cluster-level `chi.spec.configuration.clusters.settings`` and shard-level `chi.spec.configuration.clusters.layout.shards.settings`` More details: <https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

object

`.spec.configuration.clusters[].layout.shards[].replicas[].files`

Description

optional, allows define content of any setting file inside `Pod` only in one replica during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or `/etc/clickhouse-server/conf.d/` or `/etc/clickhouse-server/users.d/` override top-level `chi.spec.configuration.files``, cluster-level `chi.spec.configuration.clusters.files`` and shard-level `chi.spec.configuration.clusters.layout.shards.files``

Type

object

`.spec.configuration.clusters[].layout.shards[].replicas[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level `chi.spec.configuration.templates``, cluster-level `chi.spec.configuration.clusters.templates`` and shard-level `chi.spec.configuration.clusters.layout.shards.templates``

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
podTemplate	string	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
dataVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
logVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
serviceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
clusterServiceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
shardServiceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
replicaServiceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster

Property	Type	Description
		described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.configuration.clusters[].layout.replicas

Description

optional, allows override top-level ``chi.spec.configuration`` and cluster-level ``chi.spec.configuration.clusters`` configuration for each replica and each shard relates to selected replica, use it only if you fully understand what you do

Type

`array`

.spec.configuration.clusters[].layout.replicas[]

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	optional, by default replica name is generated, but you can override it and setup custom name
<code>settings</code>	<code>object</code>	optional, allows configure <code>clickhouse-server</code> settings inside tag in <code>Pod</code> only in one replica during generate <code>ConfigMap</code>

Property	Type	Description
		which will mount in <code>/etc/clickhouse-server/conf.d/</code> overri top-level <code>chi.spec.configuration.settings</code> , cluster-level <code>chi.spec.configuration.clusters.settings</code> and will ignor if shard-level <code>chi.spec.configuration.clusters.layout.shards</code> present More details: https://clickhouse.tech/docs/en/operations/settings/settings/ ↗
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside each <code>Pod</code> only in one replica during generate <code>ConfigMap</code> which wi mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/users.d/</code> override top-level <code>chi.spec.configuration.files</code> and cluster-level <code>chi.spec.configuration.clusters.files</code> , will ignore if <code>chi.spec.configuration.clusters.layout.shards</code> present
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level <code>chi.spec.configuration.templates</code> , cluster-level <code>chi.spec.configuration.clusters.templates</code>
<code>shardsCount</code>	<code>integer</code>	optional, count of shards related to current replica, you can override each shard behavior on low-level <code>chi.spec.configuration.clusters.layout.replicas.sharc</code>

Property	Type	Description
<code>shards</code>	<code>array</code>	optional, list of shards related to current replica, will ignore if <code>chi.spec.configuration.clusters.layout.shards</code> present

`.spec.configuration.clusters[].layout.replicas[].settings`

Description

optional, allows configure ``clickhouse-server`` settings inside `<yandex>...</yandex>` tag in ``Pod`` only in one replica during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/conf.d/`` override top-level ``chi.spec.configuration.settings``, cluster-level ``chi.spec.configuration.clusters.settings`` and will ignore if shard-level ``chi.spec.configuration.clusters.layout.shards`` present More details: <https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

`object`

`.spec.configuration.clusters[].layout.replicas[].files`

Description

optional, allows define content of any setting file inside each ``Pod`` only in one replica during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/config.d/`` or ``/etc/clickhouse-server/conf.d/`` or ``/etc/clickhouse-server/users.d/`` override top-level ``chi.spec.configuration.files`` and cluster-level ``chi.spec.configuration.clusters.files``, will ignore if ``chi.spec.configuration.clusters.layout.shards`` presents

Type

`object`

`.spec.configuration.clusters[].layout.replicas[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level

`chi.spec.configuration.templates`, cluster-level `chi.spec.configuration.clusters.templates`

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from chi.spec.templates.hostTemplates, which will apply to configure every clickhouse-server instance during render ConfigMap resources which will mount into Pod
podTemplate	string	optional, template name from chi.spec.templates.podTemplates, allows customization each Pod resource during render and reconcile each StatefulSet.spec resource described in chi.spec.configuration.clusters
dataVolumeClaimTemplate	string	optional, template name from chi.spec.templates.volumeClaimTemplates, allows customization each PVC which will mount for clickhouse data directory in each Pod during render and reconcile every StatefulSet.spec resource described in chi.spec.configuration.clusters
logVolumeClaimTemplate	string	optional, template name from chi.spec.templates.volumeClaimTemplates,

Property	Type	Description
		allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! <code>VolumeClaimTemplate</code> is deprecated in favor of <code>DataVolumeClaimTemplate</code> and <code>LogVolumeClaimTemplate</code>

`.spec.configuration.clusters[].layout.replicas[].shards`

Description

optional, list of shards related to current replica, will ignore if ``chi.spec.configuration.clusters.layout.shards`` presents

Type

`array`

`.spec.configuration.clusters[].layout.replicas[].shards[]`

Type

`object`

Property	Type	Description
name	string	optional, by default shard name is generated, but you can override it and setup custom name
secure	boolean	optional, setup <code>secure</code> inside <code>clickhouse-server</code> config, each Pod where current template will apply if specified
tcpPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name <code>tcp</code> for selected shard, override <code>chi.spec.templates.hostTemplates.spec.tcpPort</code> to connect to <code>clickhouse-server</code> via TCP Native protocol through kubernetes <code>Service</code>
httpPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name <code>http</code> for selected shard, override <code>chi.spec.templates.hostTemplates.spec.httpPort</code> to connect to <code>clickhouse-server</code> via HTTP protocol through kubernetes <code>Service</code>
interserverHTTPPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name <code>interserver</code> for selected shard, override <code>chi.spec.templates.hostTemplates.spec.interserverHTTPPort</code> to allow connect between replicas inside same shard through replicated data parts HTTP protocol
settings	object	optional, allows configure <code>clickhouse-server</code> settings in <code>Pod</code> only in one shard related to current replica due to ClickHouse distributed nature

Property	Type	Description
		<code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/conf.d/</code> override top-level <code>chi.spec.configuration.settings</code> , cluster-level <code>chi.spec.configuration.clusters.settings</code> and <code>chi.spec.configuration.clusters.layout.replic</code> More details: https://clickhouse.tech/docs/en/operations/settings/seti
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside only in one shard related to current replica during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/co</code> <code>/etc/clickhouse-server/users.d/</code> override top-le <code>chi.spec.configuration.files</code> and cluster-level <code>chi.spec.configuration.clusters.files</code> , will ign <code>chi.spec.configuration.clusters.layout.shards</code>
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which w generate Kubernetes resources according to selected top-level <code>chi.spec.configuration.templates</code> , clus <code>chi.spec.configuration.clusters.templates</code> , re <code>chi.spec.configuration.clusters.layout.replic</code>

.spec.configuration.clusters[].layout.replicas[].shards[].settings

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in `Pod` only in one shard related to current replica during generate `ConfigMap` which will

mount in `/etc/clickhouse-server/conf.d/` override top-level `chi.spec.configuration.settings``, cluster-level `chi.spec.configuration.clusters.settings`` and replica-level `chi.spec.configuration.clusters.layout.replicas.settings`` More details: <https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

object

`.spec.configuration.clusters[].layout.replicas[].shards[].files`

Description

optional, allows define content of any setting file inside each `Pod`` only in one shard related to current replica during generate `ConfigMap`` which will mount in `/etc/clickhouse-server/config.d/`` or `/etc/clickhouse-server/conf.d/`` or `/etc/clickhouse-server/users.d/`` override top-level `chi.spec.configuration.files`` and cluster-level `chi.spec.configuration.clusters.files``, will ignore if `chi.spec.configuration.clusters.layout.shards`` presents

Type

object

`.spec.configuration.clusters[].layout.replicas[].shards[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level `chi.spec.configuration.templates``, cluster-level `chi.spec.configuration.clusters.templates``, replica-level `chi.spec.configuration.clusters.layout.replicas.templates``

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
podTemplate	string	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
dataVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
logVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
serviceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
clusterServiceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
shardServiceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
replicaServiceTemplate	string	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster

Property	Type	Description
		described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.templates

Description

allows define templates which will use for render Kubernetes resources like StatefulSet, ConfigMap, Service, PVC, by default, clickhouse-operator have own templates, but you can override it

Type

`object`

Property	Type	Description
<code>hostTemplates</code>	<code>array</code>	hostTemplate will use during apply to generate <code>clickhouse-operator</code>
<code>podTemplates</code>	<code>array</code>	podTemplate will use during render <code>Pod</code> inside <code>StatefulSet</code> , rendered <code>Pod.spec</code> , pod scheduling distribution and https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explain

Property	Type	Description
volumeClaimTemplates	array	allows define template for rendering <code>PVC</code> kubernetes <code>Pod</code> for mount clickhouse <code>data</code> , clickhouse <code>logs</code>
serviceTemplates	array	allows define template for rendering <code>Service</code> which which scoped chi-wide, cluster-wide, shard-wide, repl

.spec.templates.hostTemplates

Description

hostTemplate will use during apply to generate `clickhose-server` config files

Type

array

.spec.templates.hostTemplates[]

Type

object

Property	Type	Description
name	string	template name, could use to link inside top-level <code>chi.spec.defaults.templates.hostTemplate</code> , cluster-level <code>chi.spec.configuration.clusters.templates.hostTe</code> <code>chi.spec.configuration.clusters.layout.shards.te</code> replica-level <code>chi.spec.configuration.clusters.layout.replicas.</code>

Property	Type	Description
portDistribution	array	define how will distribute numeric values of named ports in <code>Pod.spec.containers.ports</code> and clickhouse-server config
spec	object	

.spec.templates.hostTemplates[].portDistribution

Description

define how will distribute numeric values of named ports in ``Pod.spec.containers.ports`` and clickhouse-server configs

Type

array

.spec.templates.hostTemplates[].portDistribution[]

Type

object

Property	Type	Description
type	string	type of distribution, when <code>Unspecified</code> (default value) then all listen ports on clickhouse-server configuration in all Pods will have the same value, when <code>ClusterScopeIndex</code> then ports will increment to offset from base value depends on shard and replica index inside cluster with combination of <code>chi.spec.templates.podTemplates.spec.HostNetwork</code> it allows setup ClickHouse cluster inside Kubernetes and provide access via external network bypass Kubernetes internal network

.spec.templates.hostTemplates[].spec

Type

object

Property	Type	Description
name	string	by default, hostname will generate, but this allows defini <code>clickhuse-server</code>
secure	boolean	optional, setup <code>secure</code> inside <code>clickhouse-server</code> current template will apply if specified

Property	Type	Description
tcpPort	integer	optional, setup <code>tcp_port</code> inside <code>clickhouse-serve</code> current template will apply if specified, should have equ <code>chi.spec.templates.podTemplates.spec.containe</code> info: https://clickhouse.tech/docs/en/interfaces/tcp/ ↗
httpPort	integer	optional, setup <code>http_port</code> inside <code>clickhouse-serv</code> current template will apply if specified, should have equ <code>chi.spec.templates.podTemplates.spec.containe</code> info: https://clickhouse.tech/docs/en/interfaces/http/ ↗
interserverHTTPPort	integer	optional, setup <code>interserver_http_port</code> inside <code>cli</code> each Pod where current template will apply if specified <code>chi.spec.templates.podTemplates.spec.containe</code> More info: https://clickhouse.tech/docs/en/operations/sparameters/settings/#interserver-http-port ↗
settings	object	optional, allows configure <code>clickhouse-server</code> settin where this template will apply during generate <code>Config</code> <code>/etc/clickhouse-server/conf.d/</code> More details: https://clickhouse.tech/docs/en/operations/settings/seti
files	object	optional, allows define content of any setting file inside template will apply during generate <code>ConfigMap</code> which <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/cli</code> <code>/etc/clickhouse-server/users.d/</code>

Property	Type	Description
<code>templates</code>	<code>object</code>	be careful, this part of CRD allows override template in don't understand what you do

`.spec.templates.hostTemplates[].spec.settings`

Description

optional, allows configure ``clickhouse-server`` settings inside `<yandex>...</yandex>` tag in each ``Pod`` where this template will apply during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/conf.d`` More details:

<https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

`object`

`.spec.templates.hostTemplates[].spec.files`

Description

optional, allows define content of any setting file inside each ``Pod`` where this template will apply during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/config.d`` or ``/etc/clickhouse-server/conf.d`` or ``/etc/clickhouse-server/users.d``

Type

`object`

`.spec.templates.hostTemplates[].spec.templates`

Description

be careful, this part of CRD allows override template inside template, don't use it if you don't understand what you do

Type

`object`

Property	Type	Description
hostTemplate	string	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
podTemplate	string	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
dataVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
logVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
serviceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for one Service resource which will created by clickhouse-operator which cover all clusters in whole chi resource
clusterServiceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for each Service resource which will created by clickhouse-operator which cover each clickhouse cluster described in chi.spec.configuration.clusters
shardServiceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for each Service resource which will created by clickhouse-operator which cover each shard inside clickhouse cluster described in chi.spec.configuration.clusters
replicaServiceTemplate	string	optional, template name from chi.spec.templates.serviceTemplates, allows customization for each Service resource which will created by clickhouse-operator which cover each replica inside each shard inside each clickhouse cluster

Property	Type	Description
		described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.templates.podTemplates

Description

podTemplate will use during render `Pod` inside `StatefulSet.spec` and allows define rendered `Pod.spec`, pod scheduling distribution and pod zone More information: https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spectemplatespodtemplates

Type

`array`

.spec.templates.podTemplates[]

Type

`object`

Property	Type	Description
name	string	template name, could use to link inside top-level <code>chi.spec.defaults.templates.podTemplate</code> , cluster-level <code>chi.spec.configuration.clusters.templates.podTemplate</code> <code>chi.spec.configuration.clusters.layout.shards.template</code> level <code>chi.spec.configuration.clusters.layout.replicas</code>
generateName	string	allows define format for generated <code>Pod</code> name, look to https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md ↗ for details about available template variables
zone	object	allows define custom zone name and will separate ClickHouse shortcut for <code>chi.spec.templates.podTemplates.spec.affinity.pod</code>
distribution	string	DEPRECATED, shortcut for <code>chi.spec.templates.podTemplates.spec.affinity.pod</code>
podDistribution	array	define ClickHouse Pod distribution policy between Kubernetes Namespace, CHI, another ClickHouse cluster
spec	object	allows define whole Pod.spec inside StatefulSet.spec, look https://kubernetes.io/docs/concepts/workloads/pods/#pod-template-spec
metadata	ObjectMeta ↗	allows pass standard object's metadata from template to Pod https://git.k8s.io/community/contributors/devel/sig-architecture

Property	Type	Description
		conventions.md#metadata ↗

.spec.templates.podTemplates[].zone

Description

allows define custom zone name and will separate ClickHouse `Pods` between nodes, shortcut for `chi.spec.templates.podTemplates.spec.affinity.podAntiAffinity`

Type

object

Property	Type	Description
key	string	optional, if defined, allows select kubernetes nodes by label with <code>name</code> equal <code>key</code>
values	array	optional, if defined, allows select kubernetes nodes by label with <code>value</code> in <code>values</code>

.spec.templates.podTemplates[].zone.values

Description

optional, if defined, allows select kubernetes nodes by label with `value` in `values`

Type

array

.spec.templates.podTemplates[].zone.values[]

Type

string

.spec.templates.podTemplates[].podDistribution

Description

define ClickHouse Pod distibution policy between Kubernetes Nodes inside Shard, Replica, Namespace, CHI, another ClickHouse cluster

Type

array

.spec.templates.podTemplates[].podDistribution[]

Type

object

Property	Type	Description
type	string	you can define multiple affinity policy types
scope	string	scope for apply each podDistribution
number	integer	define, how much ClickHouse Pods could be inside selected scope
topologyKey	string	use for inter-pod affinity look to pod.spec.affinity.podAntiAffinity.preferredDuringSchedulingAndEviction More info: https://kubernetes.io/docs/concepts/scheduling-eviction/affinity/

.spec.templates.podTemplates[].spec

Description

allows define whole Pod.spec inside StaefulSet.spec, look to <https://kubernetes.io/docs/concepts/workloads/pods/#pod-templates> for details

Type

object

.spec.templates.volumeClaimTemplates

Description

allows define template for rendering `PVC` kubernetes resource, which would use inside `Pod` for mount clickhouse `data`, clickhouse `logs` or something else

Type

array

.spec.templates.volumeClaimTemplates[]

Type

object

Property	Type	Description
		template name, could use to link inside top-level
		<code>chi.spec.defaults.templates.dataVolumeClaimTemplate</code>
		<code>chi.spec.defaults.templates.logVolumeClaimTemplate</code>
		<code>chi.spec.configuration.clusters.templates.dataVolumeClaimTemplate</code>
		<code>chi.spec.configuration.clusters.templates.logVolumeClaimTemplate</code>
name	string	<code>chi.spec.configuration.clusters.layout.shards.templateName</code>
		<code>chi.spec.configuration.clusters.layout.shards.templateName</code>
		replica-level
		<code>chi.spec.configuration.clusters.layout.replicas.templateName</code>
		or
		<code>chi.spec.configuration.clusters.layout.replicas.templateName</code>

Property	Type	Description
<code>provisioner</code>	<code>string</code>	defines <code>PVC</code> provisioner - be it StatefulSet or the Operator
<code>reclaimPolicy</code>	<code>string</code>	defines behavior of <code>PVC</code> deletion. <code>Delete</code> by default, if <code>Retain</code> kept when deleting StatefulSet
<code>metadata</code>	<code>ObjectMeta</code> ↗	allows to pass standard object's metadata from template to PVC https://git.k8s.io/community/contributors/devel/sig-architecture
<code>spec</code>	<code>object</code>	allows define all aspects of <code>PVC</code> resource More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims

`.spec.templates.volumeClaimTemplates[].spec`

Description

allows define all aspects of `PVC` resource More info:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims>

Type

`object`

`.spec.templates.serviceTemplates`

Description

allows define template for rendering `Service` which would get endpoint from Pods which scoped chi-wide, cluster-wide, shard-wide, replica-wide level

Type

`array`

.spec.templates.serviceTemplates[]

Type

object

Property	Type	Description
name	string	template name, could use to link inside chi-level <code>chi.spec.defaults.templates.serviceTemplate</code> cluster-level <code>chi.spec.configuration.clusters.templates.clusterServiceTemplate</code> cluster-level <code>chi.spec.configuration.clusters.layout.shards.templateName</code> replica-level <code>chi.spec.configuration.clusters.layout.replicas.templateName</code> replica-level or <code>chi.spec.configuration.clusters.layout.shards.replicaTemplate</code>
generateName	string	allows define format for generated <code>Service</code> name, look to https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spec-templates details about available template variables
metadata	ObjectMeta ↗	allows pass standard object's metadata from template to Service specificly for Cloud Provider metadata which impact to behavior https://kubernetes.io/docs/concepts/services-networking/service/
spec	object	describe behavior of generated Service More info: https://kubernetes.io/docs/concepts/services-networking/service/ ↗

.spec.templates.serviceTemplates[].spec

Description

describe behavior of generated Service More info:
<https://kubernetes.io/docs/concepts/services-networking/service/>

Type

object

.spec.useTemplates

Description

list of `ClickHouseInstallationTemplate` (chit) resource names which will merge with current `Chi` manifest during render Kubernetes resources to create related ClickHouse clusters

Type

array

.spec.useTemplates[]

Type

object

Property	Type	Description
name	string	name of ClickHouseInstallationTemplate (chit) resource
namespace	string	Kubernetes namespace where need search chit resource, depending on watchNamespaces settings in clichouse-operator
useType	string	optional, current strategy is only merge, and current chi settings have more priority than merged template chit

API Endpoints

The following API endpoints are available:

- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallations`
 - **DELETE** : delete collection of ClickHouseInstallation
 - **GET** : list objects of kind ClickHouseInstallation
 - **POST** : create a new ClickHouseInstallation
- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallations/{name}`
 - **DELETE** : delete the specified ClickHouseInstallation
 - **GET** : read the specified ClickHouseInstallation
 - **PATCH** : partially update the specified ClickHouseInstallation
 - **PUT** : replace the specified ClickHouseInstallation
- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallations/{name}/status`
 - **GET** : read status of the specified ClickHouseInstallation
 - **PATCH** : partially update status of the specified ClickHouseInstallation
 - **PUT** : replace status of the specified ClickHouseInstallation

/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallations

HTTP method

DELETE

Description

delete collection of ClickHouseInstallation

HTTP responses

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

HTTP method

GET

Description

list objects of kind ClickHouseInstallation

HTTP responses

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

HTTP method

POST

Description

create a new ClickHouseInstallation

Query parameters

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

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

Body parameters

Parameter	Type	Description
body	ClickHouseInstallation schema	application/json formatted

HTTP responses

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

/apis/clickhouse.altinity.com/v1/namespaces/{namespace}
/clickhouseinstallations/{name}

HTTP method

DELETE

Description

delete the specified ClickHouseInstallation

Query parameters

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

HTTP responses

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

HTTP method

`GET`

Description

read the specified ClickHouseInstallation

HTTP responses

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

HTTP method

`PATCH`

Description

partially update the specified ClickHouseInstallation

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	ClickHouseInstallation schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace the specified ClickHouseInstallation

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	ClickHouseInstallation schema	application/json formatted

HTTP responses

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

/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallations/{name}/status

HTTP method

GET

Description

read status of the specified ClickHouseInstallation

HTTP responses

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

HTTP method

PATCH

Description

partially update status of the specified ClickHouseInstallation

Query parameters

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

Parameter	Type	Description
		processing of the request. Valid values are: - All: all dry run stages will be processed
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	ClickHouseInstallation schema
401 - Unauthorized	Empty

HTTP method

PUT

Description

replace status of the specified ClickHouseInstallation

Query parameters

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

Body parameters

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

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty



ClickHouseInstallationTemplates

Description

define a set of Kubernetes resources (StatefulSet, PVC, Service, ConfigMap) which describe behavior one or more ClickHouse clusters

Type

`object`

Required

`spec`

Specification

Property	Type	Description
<code>apiVersion</code>	<code>string</code>	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
<code>kind</code>	<code>string</code>	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint

Property	Type	Description
		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.
status	object	Current ClickHouseInstallation manifest status, contains many fields like a normalized configuration, clickhouse-operator version, current action and all applied action list, current taskID and all applied taskIDs and other
spec	object	Specification of the desired behavior of one or more ClickHouse clusters More info: https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md

.status

Description

Current ClickHouseInstallation manifest status, contains many fields like a normalized configuration, clickhouse-operator version, current action and all applied action list, current taskID and all applied taskIDs and other

Type

object

Property	Type	Description
chop-version	string	ClickHouse operator version
chop-commit	string	ClickHouse operator git commit SHA
chop-date	string	ClickHouse operator build date
chop-ip	string	IP address of the operator's pod which managed this CHI
clusters	integer	Clusters count
shards	integer	Shards count
replicas	integer	Replicas count
hosts	integer	Hosts count
status	string	Status
taskID	string	Current task id

Property	Type	Description
<code>taskIDsStarted</code>	<code>array</code>	Started task ids
<code>taskIDsCompleted</code>	<code>array</code>	Completed task ids
<code>action</code>	<code>string</code>	Action
<code>actions</code>	<code>array</code>	Actions
<code>error</code>	<code>string</code>	Last error
<code>errors</code>	<code>array</code>	Errors
<code>updated</code>	<code>integer</code>	Updated Hosts count
<code>added</code>	<code>integer</code>	Added Hosts count
<code>deleted</code>	<code>integer</code>	Deleted Hosts count
<code>delete</code>	<code>integer</code>	About to delete Hosts count

Property	Type	Description
pods	array	Pods
pod-ips	array	Pod IPs
fqdns	array	Pods FQDNs
endpoint	string	Endpoint
generation	integer	Generation
normalized	object	Normalized CHI requested
normalizedCompleted	object	Normalized CHI completed

.status.taskIDsStarted

Description

Started task ids

Type

array

.status.taskIDsStarted[]

Type`string`**.status.taskIDsCompleted****Description**

Completed task ids

Type`array`**.status.taskIDsCompleted[]****Type**`string`**.status.actions****Description**

Actions

Type`array`**.status.actions[]****Type**`string`**.status.errors****Description**

Errors

Type

`array`

.status.errors[]

Type

`string`

.status.pods

Description

Pods

Type

`array`

.status.pods[]

Type

`string`

.status.pod-ips

Description

Pod IPs

Type

`array`

.status.pod-ips[]

Type

`string`

.status.fqdns

Description

Pods FQDNs

Type

array

.status.fqdns[]

Type

string

.status.normalized

Description

Normalized CHI requested

Type

object

.status.normalizedCompleted

Description

Normalized CHI completed

Type

object

.spec

Description

Specification of the desired behavior of one or more ClickHouse clusters More info:
https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md

Type

object

Property	Type	Description
taskID	string	Allows to define custom taskID for named update o status of this update execution in .status.taskIDs fie update of chi manifest will generate random taskID
stop	string	Allow stop all ClickHouse clusters described in curr mechanism works as follows: - When <code>stop</code> is <code>1</code> then setup <code>Replicas: 0</code> current <code>chi</code> StatefulSet resource, all <code>Pods</code> a resources will descstroy, but PVCs still live - When <code>stop</code> is <code>0</code> then <code>Pods</code> will created ag retained PVCs and <code>Service</code> also will created
restart	string	This is a 'soft restart' button. When set to 'RollingUp restart ClickHouse pods in a graceful way. Remove order to avoid unneeded restarts
troubleshoot	string	allows troubleshoot Pods during CrashLoopBack si wrong configuration, <code>clickhouse-server</code> wouldr
namespaceDomainPattern	string	custom domain suffix which will add to end of <code>Ser</code> use it when you use custom cluster domain in your

Property	Type	Description
<code>templating</code>	<code>object</code>	optional, define policy for auto applying ClickHouse inside ClickHouseInstallation
<code>reconciling</code>	<code>object</code>	optional, allows tuning reconciling cycle for Clickho clickhouse-operator side
<code>defaults</code>	<code>object</code>	define default behavior for whole ClickHouseInstall can be re-define on cluster, shard and replica level https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_expla
<code>configuration</code>	<code>object</code>	allows configure multiple aspects and behavior for <code>server</code> instance and also allows describe multiple <code>server</code> clusters inside one <code>chi</code> resource
<code>templates</code>	<code>object</code>	allows define templates which will use for render K8 like StatefulSet, ConfigMap, Service, PVC, by default operator have own templates, but you can override
<code>useTemplates</code>	<code>array</code>	list of <code>ClickHouseInstallationTemplate</code> (chit) r will merge with current <code>Chi</code> manifest during render resources to create related ClickHouse clusters

.spec.templating

Description

optional, define policy for auto applying ClickHouseInstallationTemplate inside ClickHouseInstallation

Type

object

Property	Type	Description
policy	string	when defined as <code>auto</code> inside ClickhouseInstallationTemplate, it will auto add into all ClickHouseInstallation, manual value is default

.spec.reconciling

Description

optional, allows tuning reconciling cycle for ClickhouseInstallation from clickhouse-operator side

Type

object

Property	Type	Description
policy	string	DEPRECATED
configMapPropagationTimeout	integer	timeout in seconds when <code>clickhouse-operator</code> applied <code>ConfigMap</code> during reconcile <code>ClickHouseInstallation</code> pods will updated from cache see details: https://kubernetes.io/docs/concepts/configuration/configmaps-are-updated-automatically/

Property	Type	Description
<code>cleanup</code>	<code>object</code>	optional, define behavior for cleanup Kubernetes reconcile cycle

`.spec.reconciling.cleanup`

Description

optional, define behavior for cleanup Kubernetes resources during reconcile cycle

Type

`object`

Property	Type	Description
<code>unknownObjects</code>	<code>object</code>	what clickhouse-operator shall do when found Kubernetes resources which should be managed with clickhouse-operator, but not have <code>ownerReference</code> to any currently managed <code>ClickHouseInstallation</code> resource, default behavior is <code>Delete</code>
<code>reconcileFailedObjects</code>	<code>object</code>	what clickhouse-operator shall do when reconciling Kubernetes resources are failed, default behavior is <code>Retain</code>

`.spec.reconciling.cleanup.unknownObjects`

Description

what clickhouse-operator shall do when found Kubernetes resources which should be managed with clickhouse-operator, but not have `ownerReference` to any currently

managed `ClickHouseInstallation` resource, default behavior is `Delete`

Type

object

Property	Type	Description
<code>statefulSet</code>	<code>string</code>	behavior policy for unknown StatefulSet, Delete by default
<code>pvc</code>	<code>string</code>	behavior policy for unknown PVC, Delete by default
<code>configMap</code>	<code>string</code>	behavior policy for unknown ConfigMap, Delete by default
<code>service</code>	<code>string</code>	behavior policy for unknown Service, Delete by default

.spec.reconciling.cleanup.reconcileFailedObjects

Description

what clickhouse-operator shall do when reconciling Kubernetes resources are failed, default behavior is `Retain`

Type

object

Property	Type	Description
<code>statefulSet</code>	<code>string</code>	behavior policy for failed StatefulSet reconciling, Retain by default

Property	Type	Description
<code>pvc</code>	<code>string</code>	behavior policy for failed PVC reconciling, Retain by default
<code>configMap</code>	<code>string</code>	behavior policy for failed ConfigMap reconciling, Retain by default
<code>service</code>	<code>string</code>	behavior policy for failed Service reconciling, Retain by default

.spec.defaults

Description

define default behavior for whole ClickHouseInstallation, some behavior can be re-define on cluster, shard and replica level More info: https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specdefaults

Type

`object`

Property	Type	Description
<code>replicasUseFQDN</code>	<code>string</code>	define should replicas be specified by FQDN in <code><host></host></code> . In case of "no" will use short hostname and clickhouse-server will use kubernetes default suffixes for DNS lookup "yes" by default

Property	Type	Description
<code>distributedDDL</code>	<code>object</code>	allows change <code><yandex><distributed_ddl></code> <code></distributed_ddl></yandex></code> settings More info: https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings-distributed_ddl ↗
<code>storageManagement</code>	<code>object</code>	default storage management options
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will use for generate Kubernetes resources according to one or more ClickHouse clusters described in current ClickHouseInstallation (chi) resource

.spec.defaults.distributedDDL

Description

allows change ``<yandex><distributed_ddl></distributed_ddl></yandex>`` settings More info:
https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings-distributed_ddl

Type

`object`

Property	Type	Description
<code>profile</code>	<code>string</code>	Settings from this profile will be used to execute DDL queries

.spec.defaults.storageManagement

Description

default storage management options

Type

object

Property	Type	Description
<code>provisioner</code>	<code>string</code>	defines <code>PVC</code> provisioner - be it StatefulSet or the Operator
<code>reclaimPolicy</code>	<code>string</code>	defines behavior of <code>PVC</code> deletion. <code>Delete</code> by default, if <code>Retain</code> specified then <code>PVC</code> will be kept when deleting StatefulSet

.spec.defaults.templates

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to one or more ClickHouse clusters described in current ClickHouseInstallation (chi) resource

Type

object

Property	Type	Description
<code>hostTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-</code>

Property	Type	Description
		<code>server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource

Property	Type	Description
		which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.configuration

Description

allows configure multiple aspects and behavior for `clickhouse-server` instance and also allows describe multiple `clickhouse-server` clusters inside one `chi` resource

Type

`object`

Property	Type	Description
<code>zookeeper</code>	<code>object</code>	allows configure <code>..</code> section in each <code>Pod</code> during generate <code>Conf:</code> mounted in <code>/etc/clickhouse-server/config.d/</code> <code>clickhou:</code> doesn't manage Zookeeper, please install Zookeeper separatel! https://github.com/Altinity/clickhouse-operator/tree/master/deplo currently, zookeeper (or clickhouse-keeper replacement) used fr <code>*ReplicatedMergeTree</code> table engines and for <code>distributed_ddl</code> https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings_zookeeper ↗
<code>users</code>	<code>object</code>	allows configure <code>..</code> section in each <code>Pod</code> during generate <code>Conf:</code> mount in <code>/etc/clickhouse-server/users.d/</code> you can config hashed, authorization restrictions, database level security row fil https://clickhouse.tech/docs/en/operations/settings/settings-user

Property	Type	Description
		will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specifying-clickhouse-server-configuration
profiles	object	allows configure .. section in each <code>Pod</code> during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/users.d/</code> you can configure settings profile More details: https://clickhouse.tech/docs/en/operations/settings/settings-profiles will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specifying-clickhouse-server-configuration
quotas	object	allows configure .. section in each <code>Pod</code> during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/users.d/</code> you can configure resource quotas More details: https://clickhouse.tech/docs/en/operations/settings/settings-quotas Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specifying-clickhouse-server-configuration
settings	object	allows configure <code>clickhouse-server</code> settings inside ... tag in generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/settings.d/</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings-configuration code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specifying-clickhouse-server-configuration
files	object	allows define content of any setting file inside each <code>Pod</code> during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/settings.d/</code>

Property	Type	Description
		every key in this object is the file name every value in this object you can use <code>!!binary</code> and base64 for binary files, see details https://yaml.org/type/binary.html each key could contains prefix COMMON, HOST or config.d, users.d, cond.d, wrong prefixes will also will ignored More details: https://github.com/Altinity/clickhouse-operator/blob/master/docs/chi-examples/05-settings-05-files-ne
<code>clusters</code>	<code>array</code>	describes ClickHouse clusters layout and allows change settings: shard-level and replica-level every cluster is a set of StatefulSet contains only one Pod with <code>clickhouse-server</code> all Pods will run <code><remote_server></code> part of ClickHouse configs, mounted from <code>/etc/clickhouse-server/config.d/chop-generated-remote</code> Clusters will use for Distributed table engine, more details: https://clickhouse.tech/docs/en/engines/table-engines/special/distributed <code>cluster</code> contains zookeeper settings (could be inherited from <code>zookeeper</code> when you can create <code>*ReplicatedMergeTree</code> tables

.spec.configuration.zookeeper

Description

allows configure `<yandex><zookeeper>..</zookeeper></yandex>` section in each `Pod` during generate `ConfigMap` which will mounted in `/etc/clickhouse-server/config.d/` `clickhouse-operator` itself doesn't manage Zookeeper, please install Zookeeper separately look examples on https://github.com/Altinity/clickhouse-operator/tree/master/deploy/zookeeper/ currently, zookeeper (or clickhouse-keeper replacement) used for *ReplicatedMergeTree table engines and for `distributed_ddl` More details: https://clickhouse.tech/docs/en/operations/server-configuration-parameters/settings/#server-settings_zookeeper`

Type

`object`

Property	Type	Description
<code>nodes</code>	<code>array</code>	describe every available zookeeper cluster node for interaction
<code>session_timeout_ms</code>	<code>integer</code>	session timeout during connect to Zookeeper
<code>operation_timeout_ms</code>	<code>integer</code>	one operation timeout during Zookeeper transactions
<code>root</code>	<code>string</code>	optional root znode path inside zookeeper to store ClickHouse related data (replication queue or distributed DDL)
<code>identity</code>	<code>string</code>	optional access credentials string with <code>user:password</code> format used when use digest authorization in Zookeeper

`.spec.configuration.zookeeper.nodes`

Description

describe every available zookeeper cluster node for interaction

Type

`array`

`.spec.configuration.zookeeper.nodes[]`

Type

object

Property	Type	Description
host	string	dns name or ip address for Zookeeper node
port	integer	TCP port which used to connect to Zookeeper node

.spec.configuration.users

Description

allows configure `<yandex><users>..</users></yandex>` section in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/users.d/` you can configure password hashed, authorization restrictions, database level security row filters etc. More details: <https://clickhouse.tech/docs/en/operations/settings/settings-users/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationusers

Type

object

.spec.configuration.profiles

Description

allows configure `<yandex><profiles>..</profiles></yandex>` section in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/users.d/` you can configure any aspect of settings profile More details: <https://clickhouse.tech/docs/en/operations/settings/settings-profiles/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationprofiles

Type

object

.spec.configuration.quotas

Description

allows configure `<yandex><quotas>..</quotas></yandex>` section in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/users.d/` you can configure any aspect of resource quotas More details:

<https://clickhouse.tech/docs/en/operations/quotas/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationquotas

Type

object

.spec.configuration.settings

Description

allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` More details: <https://clickhouse.tech/docs/en/operations/settings/settings/> Your yaml code will convert to XML, see examples https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#specconfigurationsettings

Type

object

.spec.configuration.files

Description

allows define content of any setting file inside each `Pod` during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or `/etc/clickhouse-server/conf.d/` or `/etc/clickhouse-server/users.d/` every key in this object is the file name every value in this object is the file content you can use `!!binary |` and base64 for binary files, see details here <https://yaml.org/type/binary.html> each key could contains prefix like USERS, COMMON, HOST or config.d, users.d, cond.d, wrong prefixes will ignored, subfolders also will ignored More details: <https://github.com/Altinity/clickhouse-operator/blob/master/docs/chi-examples/05-settings-05-files-nested.yaml>

Type

object

.spec.configuration.clusters

Description

describes ClickHouse clusters layout and allows change settings on cluster-level, shard-level and replica-level every cluster is a set of StatefulSet, one StatefulSet contains only one Pod with `clickhouse-server` all Pods will rendered in <remote_server> part of ClickHouse configs, mounted from ConfigMap as `etc/clickhouse-server/config.d/chop-generated-remote\_servers.xml` Clusters will use for Distributed table engine, more details: <https://clickhouse.tech/docs/en/engines/table-engines/special/distributed/> If `cluster` contains zookeeper settings (could be inherited from top `chi` level), when you can create *ReplicatedMergeTree tables

Type

array

.spec.configuration.clusters[]

Type

object

Property	Type	Description
name	string	cluster name, used to identify set of ClickHouse servers and wide used during generate names of related Kubernetes resources
zookeeper	object	optional, allows configure .. section in each Pod only in current ClickHouse cluster, during generate ConfigMap which will mounted in etc/clickhouse-

Property	Type	Description
		<code>server/config.d/</code> override top-level <code>chi.spec.configuration.zookeeper</code> settings
<code>settings</code>	<code>object</code>	optional, allows configure <code>clickhouse-server</code> settings inside ... tag in each <code>Pod</code> only in one cluster during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> override top-level <code>chi.spec.configuration.settings</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings/
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside each <code>Pod</code> on current cluster during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/users.d/</code> override top-level <code>chi.spec.configuration.files</code>
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will use for generate Kubernetes resources according to selected cluster override top-level <code>chi.spec.configuration.templates</code>
<code>schemaPolicy</code>	<code>object</code>	describes how schema is propagated within replicas and shards

Property	Type	Description
<code>secret</code>	<code>object</code>	optional, shared secret value to secure cluster communications
<code>layout</code>	<code>object</code>	describe current cluster layout, how much shards in cluster, how much replica in shard allows override settings on each shard and replica separately

`.spec.configuration.clusters[].zookeeper`

Description

optional, allows configure `<yandex><zookeeper>..</zookeeper></yandex>` section in each `Pod` only in current ClickHouse cluster, during generate `ConfigMap` which will mounted in `/etc/clickhouse-server/config.d/` override top-level `chi.spec.configuration.zookeeper` settings

Type

`object`

Property	Type	Description
<code>nodes</code>	<code>array</code>	describe every available zookeeper cluster node for interaction
<code>session_timeout_ms</code>	<code>integer</code>	session timeout during connect to Zookeeper
<code>operation_timeout_ms</code>	<code>integer</code>	one operation timeout during Zookeeper transactions

Property	Type	Description
<code>root</code>	<code>string</code>	optional root znode path inside zookeeper to store ClickHouse related data (replication queue or distributed DDL)
<code>identity</code>	<code>string</code>	optional access credentials string with <code>user:password</code> format used when use digest authorization in Zookeeper

`.spec.configuration.clusters[].zookeeper.nodes`

Description

describe every available zookeeper cluster node for interaction

Type

`array`

`.spec.configuration.clusters[].zookeeper.nodes[]`

Type

`object`

Property	Type	Description
<code>host</code>	<code>string</code>	dns name or ip address for Zookeeper node
<code>port</code>	<code>integer</code>	TCP port which used to connect to Zookeeper node

`.spec.configuration.clusters[].settings`

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in each `Pod` only in one cluster during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` override top-level `chi.spec.configuration.settings` More details: <https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

`object`

`.spec.configuration.clusters[].files`

Description

optional, allows define content of any setting file inside each `Pod` on current cluster during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or `/etc/clickhouse-server/conf.d/` or `/etc/clickhouse-server/users.d/` override top-level `chi.spec.configuration.files`

Type

`object`

`.spec.configuration.clusters[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected cluster override top-level `chi.spec.configuration.templates`

Type

`object`

Property	Type	Description
<code>hostTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-</code>

Property	Type	Description
		<code>server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource

Property	Type	Description
		which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

`.spec.configuration.clusters[].schemaPolicy`

Description

describes how schema is propagated within replicas and shards

Type

`object`

Property	Type	Description
<code>replica</code>	<code>string</code>	how schema is propagated within a replica
<code>shard</code>	<code>string</code>	how schema is propagated between shards

`.spec.configuration.clusters[].secret`

Description

optional, shared secret value to secure cluster communications

Type

`object`

Property	Type	Description
<code>auto</code>	<code>string</code>	Auto-generate shared secret value to secure cluster communications
<code>value</code>	<code>string</code>	Cluster shared secret value in plain text
<code>valueFrom</code>	<code>object</code>	Cluster shared secret source

`.spec.configuration.clusters[].secret.valueFrom`

Description

Cluster shared secret source

Type

`object`

Property	Type	Description
<code>secretKeyRef</code>	<code>object</code>	Selects a key of a secret in the clickhouse installation namespace. Should not be used if value is not empty.

`.spec.configuration.clusters[].secret.valueFrom.secretKeyRef`

Description

Selects a key of a secret in the clickhouse installation namespace. Should not be used if value is not empty.

Type

`object`

Required

`name``key`

Property	Type	Description
<code>name</code>	<code>string</code>	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names ↗
<code>key</code>	<code>string</code>	The key of the secret to select from. Must be a valid secret key.
<code>optional</code>	<code>boolean</code>	Specify whether the Secret or its key must be defined

.spec.configuration.clusters[].layout

Description

describe current cluster layout, how much shards in cluster, how much replica in shard
allows override settings on each shard and replica separately

Type

`object`

Property	Type	Description
<code>type</code>	<code>string</code>	DEPRECATED - to be removed soon
<code>shardsCount</code>	<code>integer</code>	how much shards for current ClickHouse cluster will run in Kubernetes, each shard contains shared-nothing part

Property	Type	Description
		of data and contains set of replicas, cluster contains 1 shard by default
<code>replicasCount</code>	<code>integer</code>	how much replicas in each shards for current ClickHouse cluster will run in Kubernetes, each replica is a separate <code>StatefulSet</code> which contains only one <code>Pod</code> with <code>clickhouse-server</code> instance, every shard contains 1 replica by default
<code>shards</code>	<code>array</code>	optional, allows override top-level <code>chi.spec.configuration</code> , cluster-level <code>chi.spec.configuration.clusters</code> settings for each shard separately, use it only if you fully understand what you do
<code>replicas</code>	<code>array</code>	optional, allows override top-level <code>chi.spec.configuration</code> and cluster-level <code>chi.spec.configuration.clusters</code> configuration for each replica and each shard relates to selected replica, use it only if you fully understand what you do

`.spec.configuration.clusters[].layout.shards`

Description

optional, allows override top-level ``chi.spec.configuration``, cluster-level

``chi.spec.configuration.clusters`` settings for each shard separately, use it only if you fully understand what you do

Type

array

.spec.configuration.clusters[].layout.shards[]

Type

object

Property	Type	Description
name	string	optional, by default shard name is generated, but you can override it and setup custom name
definitionType	string	DEPRECATED - to be removed soon
weight	integer	optional, 1 by default, allows setup shard setting which during insert into tables with <code>Distributed</code> engine, with <code><remote_servers></code> inside ConfigMap which will mount in <code>/etc/clickhouse-server/config.d/chop-generated-remote_servers.xml</code> More details: https://clickhouse.tech/docs/en/engines/table-engines/special/distributed/
internalReplication	string	optional, <code>true</code> by default when <code>chi.spec.configuration.clusters[].layout.ReplicationFactor > 1</code> and 0 otherwise allows setup <code><internal_replication></code> which will use during insert into tables with <code>Distributed</code> engine for insert only in one live replica and other replicas download inserted data during replication, will apply in <code><remote_servers></code> inside ConfigMap which will mount in <code>/etc/clickhouse-server/config.d/chop-generated-remote_servers.xml</code> More details:

Property	Type	Description
		https://clickhouse.tech/docs/en/engines/table-engines/special/distributed/ ↗
settings	object	optional, allows configure <code>clickhouse-server</code> setting ... tag in each <code>Pod</code> only in one shard during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> override top-level <code>chi.spec.configuration.settings</code> and cluster-level <code>chi.spec.configuration.clusters.settings</code> More https://clickhouse.tech/docs/en/operations/settings/settings/
files	object	optional, allows define content of any setting file inside <code>Pod</code> only in one shard during generate <code>ConfigMap</code> mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/users.d/</code> override top-level <code>chi.spec.configuration.files</code> and cluster-level <code>chi.spec.configuration.clusters.files</code>
templates	object	optional, configuration of the templates names which will generate Kubernetes resources according to selected override top-level <code>chi.spec.configuration.templates</code> cluster-level <code>chi.spec.configuration.clusters.templates</code>
replicasCount	integer	optional, how much replicas in selected shard for selected ClickHouse cluster will run in Kubernetes, each replica separate <code>StatefulSet</code> which contains only one <code>Pod</code> <code>clickhouse-server</code> instance, shard contains 1 replica

Property	Type	Description
		default override cluster-level <code>chi.spec.configuration.clusters.layout.replicas</code>
<code>replicas</code>	<code>array</code>	optional, allows override behavior for selected replicas cluster-level <code>chi.spec.configuration.clusters</code> or level <code>chi.spec.configuration.clusters.layout.sl</code>

`.spec.configuration.clusters[].layout.shards[].settings`

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in each `Pod` only in one shard during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` override top-level `chi.spec.configuration.settings` and cluster-level `chi.spec.configuration.clusters.settings` More details: https://clickhouse.tech/docs/en/operations/settings/settings/`

Type

`object`

`.spec.configuration.clusters[].layout.shards[].files`

Description

optional, allows define content of any setting file inside each `Pod` only in one shard during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or /etc/clickhouse-server/conf.d/` or /etc/clickhouse-server/users.d/` override top-level `chi.spec.configuration.files` and cluster-level `chi.spec.configuration.clusters.files``

Type

`object`

`.spec.configuration.clusters[].layout.shards[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected shard override top-level `chi.spec.configuration.templates`` and cluster-level `chi.spec.configuration.clusters.templates``

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
podTemplate	string	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
dataVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside

Property	Type	Description
		clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-</code> <code>operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

`.spec.configuration.clusters[].layout.shards[].replicas`

Description

optional, allows override behavior for selected replicas from cluster-level
``chi.spec.configuration.clusters`` and shard-level
``chi.spec.configuration.clusters.layout.shards``

Type

`array`

`.spec.configuration.clusters[].layout.shards[].replicas[]`

Type

object

Property	Type	Description
name	string	optional, by default replica name is generated, but you and setup custom name
secure	boolean	optional, setup <code>secure</code> inside <code>clickhouse-server</code> each Pod where current template will apply if specified
tcpPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name selected replica, override <code>chi.spec.templates.hostTemplates.spec.tcpPort</code> connect to <code>clickhouse-server</code> via TCP Native protocol via kubernetes <code>Service</code>
httpPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name selected replica, override <code>chi.spec.templates.hostTemplates.spec.httpPort</code> connect to <code>clickhouse-server</code> via HTTP protocol via <code>Service</code>
interserverHTTPPort	integer	optional, setup <code>Pod.spec.containers.ports</code> with name <code>interserver</code> for selected replica, override <code>chi.spec.templates.hostTemplates.spec.interserverHTTPPort</code> allows connect between replicas inside same shard distributed replicated data parts HTTP protocol

Property	Type	Description
<code>settings</code>	<code>object</code>	optional, allows configure <code>clickhouse-server</code> setting in <code>Pod</code> only in one replica during generate <code>ConfigMap</code> mount in <code>/etc/clickhouse-server/conf.d/</code> overriding <code>chi.spec.configuration.settings</code>, cluster-level <code>chi.spec.configuration.clusters.settings</code> and <code>chi.spec.configuration.clusters.layout.shards</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside one replica during generate <code>ConfigMap</code> which will mount <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse-server/user</code> top-level <code>chi.spec.configuration.files</code>, cluster-level <code>chi.spec.configuration.clusters.files</code> and shards <code>chi.spec.configuration.clusters.layout.shards</code>
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will generate Kubernetes resources according to selected top-level <code>chi.spec.configuration.templates</code>, cluster-level <code>chi.spec.configuration.clusters.templates</code> and shards <code>chi.spec.configuration.clusters.layout.shards</code>

`.spec.configuration.clusters[].layout.shards[].replicas[].settings`

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in `Pod` only in one replica during generate `ConfigMap` which will mount in `/etc/clickhouse-server/conf.d/` override top-level `chi.spec.configuration.settings`, cluster-level `chi.spec.configuration.clusters.settings` and shard-level `chi.spec.configuration.clusters.layout.shards.settings` More details: https://clickhouse.tech/docs/en/operations/settings/settings/`

Type

object

`.spec.configuration.clusters[].layout.shards[].replicas[].files`

Description

optional, allows define content of any setting file inside `Pod` only in one replica during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or /etc/clickhouse-server/conf.d/` or /etc/clickhouse-server/users.d/` override top-level `chi.spec.configuration.files`, cluster-level `chi.spec.configuration.clusters.files` and shard-level `chi.spec.configuration.clusters.layout.shards.files``

Type

object

`.spec.configuration.clusters[].layout.shards[].replicas[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level `chi.spec.configuration.templates`, cluster-level `chi.spec.configuration.clusters.templates` and shard-level `chi.spec.configuration.clusters.layout.shards.templates`

Type

object

Property	Type	Description
<code>hostTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster

Property	Type	Description
		described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

`.spec.configuration.clusters[].layout.replicas`

Description

optional, allows override top-level ``chi.spec.configuration`` and cluster-level ``chi.spec.configuration.clusters`` configuration for each replica and each shard relates to selected replica, use it only if you fully understand what you do

Type

`array`

`.spec.configuration.clusters[].layout.replicas[]`

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	optional, by default replica name is generated, but you can override it and setup custom name
<code>settings</code>	<code>object</code>	optional, allows configure <code>clickhouse-server</code> settings inside tag in <code>Pod</code> only in one replica during generate <code>ConfigMap</code>

Property	Type	Description
		which will mount in <code>/etc/clickhouse-server/conf.d/</code> overri top-level <code>chi.spec.configuration.settings</code> , cluster-level <code>chi.spec.configuration.clusters.settings</code> and will ignor if shard-level <code>chi.spec.configuration.clusters.layout.shards</code> present More details: https://clickhouse.tech/docs/en/operations/settings/settings/ ↗
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside each <code>Pod</code> only in one replica during generate <code>ConfigMap</code> which wi mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/conf.d/</code> or <code>/etc/clickhouse- server/users.d/</code> override top-level <code>chi.spec.configuration.files</code> and cluster-level <code>chi.spec.configuration.clusters.files</code> , will ignore if <code>chi.spec.configuration.clusters.layout.shards</code> present
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level <code>chi.spec.configuration.templates</code> , cluster-level <code>chi.spec.configuration.clusters.templates</code>
<code>shardsCount</code>	<code>integer</code>	optional, count of shards related to current replica, you can override each shard behavior on low-level <code>chi.spec.configuration.clusters.layout.replicas.sharc</code>

Property	Type	Description
<code>shards</code>	<code>array</code>	optional, list of shards related to current replica, will ignore if <code>chi.spec.configuration.clusters.layout.shards</code> present

`.spec.configuration.clusters[].layout.replicas[].settings`

Description

optional, allows configure ``clickhouse-server`` settings inside `<yandex>...</yandex>` tag in ``Pod`` only in one replica during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/conf.d`` override top-level ``chi.spec.configuration.settings``, cluster-level ``chi.spec.configuration.clusters.settings`` and will ignore if shard-level ``chi.spec.configuration.clusters.layout.shards`` present More details: <https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

`object`

`.spec.configuration.clusters[].layout.replicas[].files`

Description

optional, allows define content of any setting file inside each ``Pod`` only in one replica during generate ``ConfigMap`` which will mount in ``/etc/clickhouse-server/config.d`` or ``/etc/clickhouse-server/conf.d`` or ``/etc/clickhouse-server/users.d`` override top-level ``chi.spec.configuration.files`` and cluster-level ``chi.spec.configuration.clusters.files``, will ignore if ``chi.spec.configuration.clusters.layout.shards`` presents

Type

`object`

`.spec.configuration.clusters[].layout.replicas[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level

`chi.spec.configuration.templates`, cluster-level `chi.spec.configuration.clusters.templates`

Type

object

Property	Type	Description
hostTemplate	string	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
podTemplate	string	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
dataVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
logVolumeClaimTemplate	string	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> ,

Property	Type	Description
		allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! <code>VolumeClaimTemplate</code> is deprecated in favor of <code>DataVolumeClaimTemplate</code> and <code>LogVolumeClaimTemplate</code>

`.spec.configuration.clusters[].layout.replicas[].shards`

Description

optional, list of shards related to current replica, will ignore if ``chi.spec.configuration.clusters.layout.shards`` presents

Type

`array`

`.spec.configuration.clusters[].layout.replicas[].shards[]`

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	optional, by default shard name is generated, but you can override it and setup custom name
<code>secure</code>	<code>boolean</code>	optional, setup <code>secure</code> inside <code>clickhouse-server</code> container, each Pod where current template will apply if specified
<code>tcpPort</code>	<code>integer</code>	optional, setup <code>Pod.spec.containers.ports</code> with name <code>tcp</code> for selected shard, override <code>chi.spec.templates.hostTemplates.spec.tcpPort</code> to connect to <code>clickhouse-server</code> via TCP Native protocol through kubernetes <code>Service</code>
<code>httpPort</code>	<code>integer</code>	optional, setup <code>Pod.spec.containers.ports</code> with name <code>http</code> for selected shard, override <code>chi.spec.templates.hostTemplates.spec.httpPort</code> to connect to <code>clickhouse-server</code> via HTTP protocol through kubernetes <code>Service</code>
<code>interserverHTTPPort</code>	<code>integer</code>	optional, setup <code>Pod.spec.containers.ports</code> with name <code>interserver</code> for selected shard, override <code>chi.spec.templates.hostTemplates.spec.interserverHTTPPort</code> to allow connect between replicas inside same shard through replicated data parts HTTP protocol
<code>settings</code>	<code>object</code>	optional, allows configure <code>clickhouse-server</code> settings in <code>Pod</code> only in one shard related to current replica due to distributed nature of ClickHouse

Property	Type	Description
		<code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/conf.d/</code> override top-level <code>chi.spec.configuration.settings</code> , cluster-level <code>chi.spec.configuration.clusters.settings</code> and <code>chi.spec.configuration.clusters.layout.replicas</code> More details: https://clickhouse.tech/docs/en/operations/settings/settings
<code>files</code>	<code>object</code>	optional, allows define content of any setting file inside only in one shard related to current replica during generate <code>ConfigMap</code> which will mount in <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/clickhouse-server/config.d/</code> override top-level <code>chi.spec.configuration.files</code> and cluster-level <code>chi.spec.configuration.clusters.files</code> , will ignore <code>chi.spec.configuration.clusters.layout.shards</code>
<code>templates</code>	<code>object</code>	optional, configuration of the templates names which will generate Kubernetes resources according to selected top-level <code>chi.spec.configuration.templates</code> , cluster-level <code>chi.spec.configuration.clusters.templates</code> , replica-level <code>chi.spec.configuration.clusters.layout.replicas</code>

`chi.spec.configuration.clusters[].layout.replicas[].shards[].settings`

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in `Pod` only in one shard related to current replica during generate `ConfigMap` which will

mount in `/etc/clickhouse-server/conf.d/` override top-level `chi.spec.configuration.settings``, cluster-level `chi.spec.configuration.clusters.settings`` and replica-level `chi.spec.configuration.clusters.layout.replicas.settings`` More details: <https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

object

`.spec.configuration.clusters[].layout.replicas[].shards[].files`

Description

optional, allows define content of any setting file inside each `Pod`` only in one shard related to current replica during generate `ConfigMap`` which will mount in `/etc/clickhouse-server/config.d/`` or `/etc/clickhouse-server/conf.d/`` or `/etc/clickhouse-server/users.d/`` override top-level `chi.spec.configuration.files`` and cluster-level `chi.spec.configuration.clusters.files``, will ignore if `chi.spec.configuration.clusters.layout.shards`` presents

Type

object

`.spec.configuration.clusters[].layout.replicas[].shards[].templates`

Description

optional, configuration of the templates names which will use for generate Kubernetes resources according to selected replica override top-level `chi.spec.configuration.templates``, cluster-level `chi.spec.configuration.clusters.templates``, replica-level `chi.spec.configuration.clusters.layout.replicas.templates``

Type

object

Property	Type	Description
<code>hostTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster

Property	Type	Description
		described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

.spec.templates

Description

allows define templates which will use for render Kubernetes resources like StatefulSet, ConfigMap, Service, PVC, by default, clickhouse-operator have own templates, but you can override it

Type

`object`

Property	Type	Description
<code>hostTemplates</code>	<code>array</code>	hostTemplate will use during apply to generate <code>clickhouse-operator</code>
<code>podTemplates</code>	<code>array</code>	podTemplate will use during render <code>Pod</code> inside <code>StatefulSet</code> , rendered <code>Pod.spec</code> , pod scheduling distribution and https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explain

Property	Type	Description
<code>volumeClaimTemplates</code>	<code>array</code>	allows define template for rendering <code>PVC</code> kubernetes: <code>Pod</code> for mount clickhouse <code>data</code> , clickhouse <code>logs</code>
<code>serviceTemplates</code>	<code>array</code>	allows define template for rendering <code>Service</code> which which scoped chi-wide, cluster-wide, shard-wide, repl

`.spec.templates.hostTemplates`

Description

hostTemplate will use during apply to generate `clickhose-server` config files

Type

`array`

`.spec.templates.hostTemplates[]`

Type

`object`

Property	Type	Description
<code>name</code>	<code>string</code>	template name, could use to link inside top-level <code>chi.spec.defaults.templates.hostTemplate</code> , cluster- <code>chi.spec.configuration.clusters.templates.hostTe</code> <code>chi.spec.configuration.clusters.layout.shards.te</code> replica-level <code>chi.spec.configuration.clusters.layout.replicas.</code>

Property	Type	Description
<code>portDistribution</code>	<code>array</code>	define how will distribute numeric values of named ports in <code>Pod.spec.containers.ports</code> and clickhouse-server configs
<code>spec</code>	<code>object</code>	

`.spec.templates.hostTemplates[].portDistribution`

Description

define how will distribute numeric values of named ports in `Pod.spec.containers.ports` and clickhouse-server configs

Type

`array`

`.spec.templates.hostTemplates[].portDistribution[]`

Type

`object`

Property	Type	Description
type	string	type of distribution, when <code>Unspecified</code> (default value) then all listen ports on clickhouse-server configuration in all Pods will have the same value, when <code>ClusterScopeIndex</code> then ports will increment to offset from base value depends on shard and replica index inside cluster with combination of <code>chi.spec.templates.podTemplates.spec.HostNetwork</code> it allows setup ClickHouse cluster inside Kubernetes and provide access via external network bypass Kubernetes internal network

.spec.templates.hostTemplates[].spec

Type

object

Property	Type	Description
name	string	by default, hostname will generate, but this allows defini <code>clickhuse-server</code>
secure	boolean	optional, setup <code>secure</code> inside <code>clickhouse-server</code> current template will apply if specified

Property	Type	Description
tcpPort	integer	optional, setup <code>tcp_port</code> inside <code>clickhouse-serve</code> current template will apply if specified, should have equ <code>chi.spec.templates.podTemplates.spec.containe</code> info: https://clickhouse.tech/docs/en/interfaces/tcp/ ↗
httpPort	integer	optional, setup <code>http_port</code> inside <code>clickhouse-serv</code> current template will apply if specified, should have equ <code>chi.spec.templates.podTemplates.spec.containe</code> info: https://clickhouse.tech/docs/en/interfaces/http/ ↗
interserverHTTPPort	integer	optional, setup <code>interserver_http_port</code> inside <code>cli</code> each Pod where current template will apply if specified <code>chi.spec.templates.podTemplates.spec.containe</code> More info: https://clickhouse.tech/docs/en/operations/s/parameters/settings/#interserver-http-port ↗
settings	object	optional, allows configure <code>clickhouse-server</code> settin where this template will apply during generate <code>Config</code> <code>/etc/clickhouse-server/conf.d/</code> More details: https://clickhouse.tech/docs/en/operations/settings/seti
files	object	optional, allows define content of any setting file inside template will apply during generate <code>ConfigMap</code> which <code>/etc/clickhouse-server/config.d/</code> or <code>/etc/cli</code> <code>/etc/clickhouse-server/users.d/</code>

Property	Type	Description
<code>templates</code>	<code>object</code>	be careful, this part of CRD allows override template in don't understand what you do

`.spec.templates.hostTemplates[].spec.settings`

Description

optional, allows configure `clickhouse-server` settings inside `<yandex>...</yandex>` tag in each `Pod` where this template will apply during generate `ConfigMap` which will mount in `/etc/clickhouse-server/conf.d/` More details:`

<https://clickhouse.tech/docs/en/operations/settings/settings/>

Type

`object`

`.spec.templates.hostTemplates[].spec.files`

Description

optional, allows define content of any setting file inside each `Pod` where this template will apply during generate `ConfigMap` which will mount in `/etc/clickhouse-server/config.d/` or /etc/clickhouse-server/conf.d/` or /etc/clickhouse-server/users.d/``

Type

`object`

`.spec.templates.hostTemplates[].spec.templates`

Description

be careful, this part of CRD allows override template inside template, don't use it if you don't understand what you do

Type

`object`

Property	Type	Description
<code>hostTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.hostTemplates</code> , which will apply to configure every <code>clickhouse-server</code> instance during render ConfigMap resources which will mount into <code>Pod</code>
<code>podTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.podTemplates</code> , allows customization each <code>Pod</code> resource during render and reconcile each <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>dataVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse data directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>
<code>logVolumeClaimTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.volumeClaimTemplates</code> , allows customization each <code>PVC</code> which will mount for clickhouse log directory in each <code>Pod</code> during render and reconcile every <code>StatefulSet.spec</code> resource described in <code>chi.spec.configuration.clusters</code>

Property	Type	Description
<code>serviceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for one <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover all clusters in whole <code>chi</code> resource
<code>clusterServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>shardServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each shard inside clickhouse cluster described in <code>chi.spec.configuration.clusters</code>
<code>replicaServiceTemplate</code>	<code>string</code>	optional, template name from <code>chi.spec.templates.serviceTemplates</code> , allows customization for each <code>Service</code> resource which will created by <code>clickhouse-operator</code> which cover each replica inside each shard inside each clickhouse cluster

Property	Type	Description
		described in <code>chi.spec.configuration.clusters</code>
<code>volumeClaimTemplate</code>	<code>string</code>	DEPRECATED! VolumeClaimTemplate is deprecated in favor of DataVolumeClaimTemplate and LogVolumeClaimTemplate

`.spec.templates.podTemplates`

Description

podTemplate will use during render `Pod` inside `StatefulSet.spec` and allows define rendered `Pod.spec`, pod scheduling distribution and pod zone More information: https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spectemplatespodtemplates

Type

`array`

`.spec.templates.podTemplates[]`

Type

`object`

Property	Type	Description
name	string	template name, could use to link inside top-level <code>chi.spec.defaults.templates.podTemplate</code> , cluster-level <code>chi.spec.configuration.clusters.templates.podTemplate</code> <code>chi.spec.configuration.clusters.layout.shards.template</code> level <code>chi.spec.configuration.clusters.layout.replicas</code>
generateName	string	allows define format for generated <code>Pod</code> name, look to https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md ↗ for details about available template variables
zone	object	allows define custom zone name and will separate ClickHouse shortcut for <code>chi.spec.templates.podTemplates.spec.affinity.pod</code>
distribution	string	DEPRECATED, shortcut for <code>chi.spec.templates.podTemplates.spec.affinity.pod</code>
podDistribution	array	define ClickHouse Pod distribution policy between Kubernetes Namespace, CHI, another ClickHouse cluster
spec	object	allows define whole Pod.spec inside StatefulSet.spec, look https://kubernetes.io/docs/concepts/workloads/pods/#pod-template-spec
metadata	ObjectMeta ↗	allows pass standard object's metadata from template to Pod https://git.k8s.io/community/contributors/devel/sig-architecture

Property	Type	Description
		conventions.md#metadata ↗

`.spec.templates.podTemplates[].zone`

Description

allows define custom zone name and will separate ClickHouse `Pods` between nodes, shortcut for ``chi.spec.templates.podTemplates.spec.affinity.podAntiAffinity``

Type

object

Property	Type	Description
<code>key</code>	<code>string</code>	optional, if defined, allows select kubernetes nodes by label with <code>name</code> equal <code>key</code>
<code>values</code>	<code>array</code>	optional, if defined, allows select kubernetes nodes by label with <code>value</code> in <code>values</code>

`.spec.templates.podTemplates[].zone.values`

Description

optional, if defined, allows select kubernetes nodes by label with ``value`` in ``values``

Type

array

`.spec.templates.podTemplates[].zone.values[]`

Type

`string`

`.spec.templates.podTemplates[].podDistribution`

Description

define ClickHouse Pod distibution policy between Kubernetes Nodes inside Shard, Replica, Namespace, CHI, another ClickHouse cluster

Type

`array`

`.spec.templates.podTemplates[].podDistribution[]`

Type

`object`

Property	Type	Description
<code>type</code>	<code>string</code>	you can define multiple affinity policy types
<code>scope</code>	<code>string</code>	scope for apply each podDistribution
<code>number</code>	<code>integer</code>	define, how much ClickHouse Pods could be inside selected scope
<code>topologyKey</code>	<code>string</code>	use for inter-pod affinity look to <code>pod.spec.affinity.podAntiAffinity.preferredDuringSchedulingEvictionPolicy</code> More info: https://kubernetes.io/docs/concepts/scheduling-eviction/inter-pod-anti-affinity/

`.spec.templates.podTemplates[].spec`

Description

allows define whole Pod.spec inside StaefulSet.spec, look to

<https://kubernetes.io/docs/concepts/workloads/pods/#pod-templates> for details

Type

object

.spec.templates.volumeClaimTemplates

Description

allows define template for rendering `PVC` kubernetes resource, which would use inside

`Pod` for mount clickhouse `data`, clickhouse `logs` or something else

Type

array

.spec.templates.volumeClaimTemplates[]

Type

object

Property	Type	Description
		template name, could use to link inside top-level
		<code>chi.spec.defaults.templates.dataVolumeClaimTemplate</code>
		<code>chi.spec.defaults.templates.logVolumeClaimTemplate</code>
		<code>chi.spec.configuration.clusters.templates.dataVolumeClaimTemplate</code>
		<code>chi.spec.configuration.clusters.templates.logVolumeClaimTemplate</code>
name	string	<code>chi.spec.configuration.clusters.layout.shards.templateName</code>
		<code>chi.spec.configuration.clusters.layout.shards.templateName</code>
		replica-level
		<code>chi.spec.configuration.clusters.layout.replicas.templateName</code>
		or
		<code>chi.spec.configuration.clusters.layout.replicas.templateName</code>

Property	Type	Description
<code>provisioner</code>	<code>string</code>	defines <code>PVC</code> provisioner - be it StatefulSet or the Operator
<code>reclaimPolicy</code>	<code>string</code>	defines behavior of <code>PVC</code> deletion. <code>Delete</code> by default, if <code>Retain</code> kept when deleting StatefulSet
<code>metadata</code>	<code>ObjectMeta</code> ↗	allows to pass standard object's metadata from template to PVC https://git.k8s.io/community/contributors/devel/sig-architecture
<code>spec</code>	<code>object</code>	allows define all aspects of <code>PVC</code> resource More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims

`.spec.templates.volumeClaimTemplates[].spec`

Description

allows define all aspects of `PVC` resource More info:

<https://kubernetes.io/docs/concepts/storage/persistent-volumes/#persistentvolumeclaims>

Type

`object`

`.spec.templates.serviceTemplates`

Description

allows define template for rendering `Service` which would get endpoint from Pods which scoped chi-wide, cluster-wide, shard-wide, replica-wide level

Type

`array`

.spec.templates.serviceTemplates[]

Type

object

Property	Type	Description
name	string	template name, could use to link inside chi-level <code>chi.spec.defaults.templates.serviceTemplate</code> cluster-level <code>chi.spec.configuration.clusters.templates.clusterServiceTemplate</code> cluster-level <code>chi.spec.configuration.clusters.layout.shards.template</code> replica-level <code>chi.spec.configuration.clusters.layout.replicas.template</code> replica-level or <code>chi.spec.configuration.clusters.layout.shards.replicaTemplate</code> replica-level
generateName	string	allows define format for generated <code>Service</code> name, look to https://github.com/Altinity/clickhouse-operator/blob/master/docs/custom_resource_explained.md#spec-templates details about available template variables
metadata	ObjectMeta ↗	allows pass standard object's metadata from template to Service specificly for Cloud Provider metadata which impact to behavior https://kubernetes.io/docs/concepts/services-networking/service/
spec	object	describe behavior of generated Service More info: https://kubernetes.io/docs/concepts/services-networking/service/ ↗

.spec.templates.serviceTemplates[].spec

Description

describe behavior of generated Service More info:

<https://kubernetes.io/docs/concepts/services-networking/service/>

Type

object

.spec.useTemplates

Description

list of `ClickHouseInstallationTemplate` (chit) resource names which will merge with current `Chi` manifest during render Kubernetes resources to create related ClickHouse clusters

Type

array

.spec.useTemplates[]

Type

object

Property	Type	Description
name	string	name of <code>ClickHouseInstallationTemplate</code> (chit) resource
namespace	string	Kubernetes namespace where need search <code>chit</code> resource, depending on <code>watchNamespaces</code> settings in <code>clickhouse-operator</code>
useType	string	optional, current strategy is only merge, and current <code>chi</code> settings have more priority than merged template <code>chit</code>

API Endpoints

The following API endpoints are available:

- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallati`
`ontemplates`
 - **DELETE** : delete collection of ClickHouseInstallationTemplate
 - **GET** : list objects of kind ClickHouseInstallationTemplate
 - **POST** : create a new ClickHouseInstallationTemplate
- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallati`
`ontemplates/{name}`
 - **DELETE** : delete the specified ClickHouseInstallationTemplate
 - **GET** : read the specified ClickHouseInstallationTemplate
 - **PATCH** : partially update the specified ClickHouseInstallationTemplate
 - **PUT** : replace the specified ClickHouseInstallationTemplate
- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallati`
`ontemplates/{name}/status`
 - **GET** : read status of the specified ClickHouseInstallationTemplate
 - **PATCH** : partially update status of the specified ClickHouseInstallationTemplate
 - **PUT** : replace status of the specified ClickHouseInstallationTemplate

/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallationtemplates

HTTP method

DELETE

Description

delete collection of ClickHouseInstallationTemplate

HTTP responses

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

HTTP method

GET

Description

list objects of kind ClickHouseInstallationTemplate

HTTP responses

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

HTTP method

POST

Description

create a new ClickHouseInstallationTemplate

Query parameters

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

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

Body parameters

Parameter	Type	Description
body	ClickHouseInstallationTemplate schema	application/json formatted

HTTP responses

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

**/apis/clickhouse.altinity.com/v1/namespaces/{namespace}
/clickhouseinstallationtemplates/{name}**

HTTP method

DELETE

Description

delete the specified ClickHouseInstallationTemplate

Query parameters

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

HTTP responses

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

HTTP method

GET

Description

read the specified ClickHouseInstallationTemplate

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ClickHouseInstallationTemplate

Query parameters

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

HTTP responses

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

HTTP method

PUT

Description

replace the specified ClickHouseInstallationTemplate

Query parameters

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

Body parameters

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

HTTP responses

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

/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseinstallationtemplates/{name}/status

HTTP method

`GET`

Description

read status of the specified ClickHouseInstallationTemplate

HTTP responses

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

HTTP method

`PATCH`

Description

partially update status of the specified ClickHouseInstallationTemplate

Query parameters

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

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

HTTP responses

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

HTTP method

`PUT`

Description

replace status of the specified ClickHouseInstallationTemplate

Query parameters

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

Body parameters

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

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty



ClickHouseOperatorConfigurations

Description

allows customize `clickhouse-operator` settings, need restart clickhouse-operator pod after adding, more details https://github.com/Altinity/clickhouse-operator/blob/master/docs/operator_configuration.md

Type

object

Specification

Property	Type	Description
<code>status</code>	object	
<code>spec</code>	object	Allows to define settings of the clickhouse-operator. More info: https://github.com/Altinity/clickhouse-operator/blob/master/config/config.yaml ↗ Check into etc-clickhouse-operator* ConfigMaps if you need more control

.status

Type

object

.spec

Description

Allows to define settings of the clickhouse-operator. More info:

<https://github.com/Altinity/clickhouse-operator/blob/master/config/config.yaml> Check into etc-clickhouse-operator* ConfigMaps if you need more control

Type

object

Property	Type	Description
<code>watch</code>	object	Parameters for watch kubernetes resources which used by clickhouse-operator deployment
<code>clickhouse</code>	object	Clickhouse related parameters used by clickhouse-operator
<code>template</code>	object	Parameters which are used if you want to generate ClickHouseInstallationTemplate custom resources from files which are stored inside clickhouse-operator deployment
<code>reconcile</code>	object	allow tuning reconciling process
<code>annotation</code>	object	defines which metadata.annotations items will include or exclude during render StatefulSet, Pod, PVC resources

Property	Type	Description
<code>label</code>	<code>object</code>	defines which metadata.labels will include or exclude during render StatefulSet, Pod, PVC resources
<code>statefulSet</code>	<code>object</code>	define StatefulSet-specific parameters
<code>pod</code>	<code>object</code>	define pod specific parameters
<code>logger</code>	<code>object</code>	allow setup clickhouse-operator logger behavior

`.spec.watch`

Description

Parameters for watch kubernetes resources which used by clickhouse-operator deployment

Type

`object`

Property	Type	Description
<code>namespaces</code>	<code>array</code>	List of namespaces where clickhouse-operator watches for events.

`.spec.watch.namespaces`

Description

List of namespaces where clickhouse-operator watches for events.

Type`array`**.spec.watch.namespaces[]****Type**`string`**.spec.clickhouse****Description**

Clickhouse related parameters used by clickhouse-operator

Type`object`

Property	Type	Description
<code>configuration</code>	<code>object</code>	
<code>access</code>	<code>object</code>	parameters which use for connect to clickhouse from clickhouse-operator deployment

.spec.clickhouse.configuration**Type**`object`

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

Property	Type	Description
<code>user</code>	<code>object</code>	Default parameters for any user which will create
<code>network</code>	<code>object</code>	Default network parameters for any user which will create

`.spec.clickhouse.configuration.file`

Type

`object`

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

`.spec.clickhouse.configuration.file.path`

Type

`object`

Property	Type	Description
<code>common</code>	<code>string</code>	Path to the folder where ClickHouse configuration files common for all instances within a CHI are located. Default - config.d
<code>host</code>	<code>string</code>	Path to the folder where ClickHouse configuration files unique for each instance (host) within a CHI are located. Default - conf.d

Property	Type	Description
<code>user</code>	<code>string</code>	Path to the folder where ClickHouse configuration files with users settings are located. Files are common for all instances within a CHI. Default - users.d

`.spec.clickhouse.configuration.user`

Description

Default parameters for any user which will create

Type

`object`

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

`.spec.clickhouse.configuration.user.default`

Type

`object`

Property	Type	Description
<code>profile</code>	<code>string</code>	ClickHouse server configuration <code><profile>...</profile></code> for any
<code>quota</code>	<code>string</code>	ClickHouse server configuration <code><quota>...</quota></code> for any

Property	Type	Description
<code>networksIP</code>	<code>array</code>	ClickHouse server configuration <code><networks><ip>...</ip></networks></code> for any
<code>password</code>	<code>string</code>	ClickHouse server configuration <code><password>...</password></code> for any

`.spec.clickhouse.configuration.user.default.networksIP`

Description

ClickHouse server configuration ``<networks><ip>...</ip></networks>`` for any `<user>`

Type

`array`

`.spec.clickhouse.configuration.user.default.networksIP[]`

Type

`string`

`.spec.clickhouse.configuration.network`

Description

Default network parameters for any user which will create

Type

`object`

Property	Type	Description
<code>hostRegexpTemplate</code>	<code>string</code>	ClickHouse server configuration <code><host_regexp>...</host_regexp></code> for any

.spec.clickhouse.access

Description

parameters which use for connect to clickhouse from clickhouse-operator deployment

Type

`object`

Property	Type	Description
<code>scheme</code>	<code>string</code>	The scheme to user for connecting to ClickHouse. One of http or https
<code>username</code>	<code>string</code>	ClickHouse username to be used by operator to connect to ClickHouse instances, deprecated, use <code>chCredentialsSecretName</code>
<code>password</code>	<code>string</code>	ClickHouse password to be used by operator to connect to ClickHouse instances, deprecated, use <code>chCredentialsSecretName</code>
<code>rootCA</code>	<code>string</code>	Root certificate authority that clients use when verifying server certificates. Used for https connection to ClickHouse

Property	Type	Description
<code>secret</code>	<code>object</code>	
<code>port</code>	<code>integer</code>	Port to be used by operator to connect to ClickHouse instances
<code>timeouts</code>	<code>object</code>	Timeouts used to limit connection and queries from the operator to ClickHouse instances, In seconds

`.spec.clickhouse.access.secret`

Type

`object`

Property	Type	Description
<code>namespace</code>	<code>string</code>	Location of k8s Secret with username and password to be used by operator to connect to ClickHouse instances
<code>name</code>	<code>string</code>	Name of k8s Secret with username and password to be used by operator to connect to ClickHouse instances

`.spec.clickhouse.access.timeouts`

Description

Timeouts used to limit connection and queries from the operator to ClickHouse instances, In seconds

Type

`object`

Property	Type	Description
<code>connect</code>	<code>integer</code>	Connect timeout. In seconds.
<code>query</code>	<code>integer</code>	Query timeout. In seconds.

.spec.template

Description

Parameters which are used if you want to generate ClickHouseInstallationTemplate custom resources from files which are stored inside clickhouse-operator deployment

Type

`object`

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

.spec.template.chi

Type

`object`

Property	Type	Description
<code>path</code>	<code>string</code>	Path to folder where ClickHouseInstallationTemplate .yaml manifests are located.

.spec.reconcile

Description

allow tuning reconciling process

Type

object

Property	Type	Description
<code>runtime</code>	object	runtime parameters for clickhouse-operator process which use during reconciling
<code>statefulSet</code>	object	Allow change default behavior for reconciling StatefulSet which generated by clickhouse-operator
<code>host</code>	object	allow define how to wait host include to system.cluster behavior during scale up and scale down cluster operations

.spec.reconcile.runtime

Description

runtime parameters for clickhouse-operator process which use during reconciling

Type

object

Property	Type	Description
<code>threadsNumber</code>	<code>integer</code>	How many goroutines will be used to reconcile in parallel, 10 by default

`.spec.reconcile.statefulSet`

Description

Allow change default behavior for reconciling StatefulSet which generated by clickhouse-operator

Type

`object`

Property	Type	Description
<code>create</code>	<code>object</code>	Behavior during create StatefulSet
<code>update</code>	<code>object</code>	Behavior during update StatefulSet

`.spec.reconcile.statefulSet.create`

Description

Behavior during create StatefulSet

Type

`object`

Property	Type	Description
		What to do in case created StatefulSet is not in Ready after <code>statefulSetUpdateTimeout</code> seconds Possible options:
<code>onFailure</code>	<code>string</code>	1. abort - do nothing, just break the process and wait for admin. 2. delete - delete newly created problematic StatefulSet. 3. ignore (default) - ignore error, pretend nothing happened and move on to the next StatefulSet.

.spec.reconcile.statefulSet.update

Description

Behavior during update StatefulSet

Type

`object`

Property	Type	Description
<code>timeout</code>	<code>integer</code>	How many seconds to wait for created/updated StatefulSet to be Ready
<code>pollInterval</code>	<code>integer</code>	How many seconds to wait between checks for created/updated StatefulSet status
<code>onFailure</code>	<code>string</code>	What to do in case updated StatefulSet is not in Ready after <code>statefulSetUpdateTimeout</code> seconds Possible options:

Property	Type	Description
		1. abort - do nothing, just break the process and wait for admin. 2. rollback (default) - delete Pod and rollback StatefulSet to previous Generation. Pod would be recreated by StatefulSet based on rollback-ed configuration. 3. ignore - ignore error, pretend nothing happened and move on to the next StatefulSet.

.spec.reconcile.host

Description

allow define how to wait host include to system.cluster behavior during scale up and scale down cluster operations

Type

object

Property	Type	Description
wait	object	

.spec.reconcile.host.wait

Type

object

Property	Type	Description
exclude	string	wait when a pod will be removed from the cluster

Property	Type	Description
<code>include</code>	<code>string</code>	wait when a pod will be added to the cluster

`.spec.annotation`

Description

defines which `metadata.annotations` items will include or exclude during render `StatefulSet`, `Pod`, `PVC` resources

Type

`object`

Property	Type	Description
<code>include</code>	<code>array</code>	When propagating labels from the chi's <code>metadata.annotations</code> section to child objects' <code>metadata.annotations</code>, include annotations with names from the following list
<code>exclude</code>	<code>array</code>	When propagating labels from the chi's <code>metadata.annotations</code> section to child objects' <code>metadata.annotations</code>, exclude annotations with names from the following list

`.spec.annotation.include`

Description

When propagating labels from the chi's `metadata.annotations` section to child objects' `metadata.annotations`, include annotations with names from the following list

Type

array

.spec.annotation.include[]

Type

string

.spec.annotation.exclude

Description

When propagating labels from the chi's `metadata.annotations` section to child objects' `metadata.annotations`, exclude annotations with names from the following list

Type

array

.spec.annotation.exclude[]

Type

string

.spec.label

Description

defines which metadata.labels will include or exclude during render StatefulSet, Pod, PVC resources

Type

object

Property	Type	Description
include	array	When propagating labels from the chi's <code>metadata.labels</code> section to child objects' <code>metadata.labels</code> , include labels from the following list
exclude	array	When propagating labels from the chi's <code>metadata.labels</code> section to child objects' <code>metadata.labels</code> , exclude labels from the following list
appendScope	string	Whether to append <i>Scope</i> labels to StatefulSet and Pod "LabelShardScopeIndex" "LabelReplicaScopeIndex" "LabelCHIScopeIndex" "LabelCHIScopeCycleSize" "LabelCHIScopeCycleIndex" "LabelCHIScopeCycleOffset" "LabelClusterScopeIndex" "LabelClusterScopeCycleSize" "LabelClusterScopeCycleIndex" "LabelClusterScopeCycleOffset"

.spec.label.include

Description

When propagating labels from the chi's ``metadata.labels`` section to child objects' ``metadata.labels`` , include labels from the following list

Type

`array`

.spec.label.include[]

Type

`string`

.spec.label.exclude

Description

When propagating labels from the chi's `metadata.labels` section to child objects' `metadata.labels`, exclude labels from the following list

Type

`array`

.spec.label.exclude[]

Type

`string`

.spec.statefulSet

Description

define StatefulSet-specific parameters

Type

`object`

Property	Type	Description
<code>revisionHistoryLimit</code>	<code>integer</code>	revisionHistoryLimit is the maximum number of revisions that will be maintained in the StatefulSet's revision history.

Property	Type	Description
		Look details in <code>statefulset.spec.revisionHistoryLimit</code>

.spec.pod

Description

define pod specific parameters

Type

`object`

Property	Type	Description
<code>terminationGracePeriod</code>	<code>integer</code>	Optional duration in seconds the pod needs to terminate gracefully. Look details in <code>pod.spec.terminationGracePeriodSeconds</code>

.spec.logger

Description

allow setup clickhouse-operator logger behavior

Type

`object`

Property	Type	Description
<code>logtostderr</code>	<code>string</code>	boolean, allows logs to stderr

Property	Type	Description
<code>alsologtostderr</code>	<code>string</code>	boolean allows logs to stderr and files both
<code>v</code>	<code>string</code>	verbosity level of clickhouse-operator log, default - 1 max - 9
<code>stderrthreshold</code>	<code>string</code>	
<code>vmodule</code>	<code>string</code>	Comma-separated list of filename=N, where filename (can be a pattern) must have no .go ext, and N is a V level. Ex.: file*=2 sets the 'V' to 2 in all files with names like file*.
<code>log_backtrace_at</code>	<code>string</code>	It can be set to a file and line number with a logging line. Ex.: file.go:123 Each time when this line is being executed, a stack trace will be written to the Info log.

API Endpoints

The following API endpoints are available:

- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseoperatorconfigurations`
 - `DELETE` : delete collection of ClickHouseOperatorConfiguration
 - `GET` : list objects of kind ClickHouseOperatorConfiguration
 - `POST` : create a new ClickHouseOperatorConfiguration

- `/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseoperatorconfigurations/{name}`
 - **DELETE** : delete the specified ClickHouseOperatorConfiguration
 - **GET** : read the specified ClickHouseOperatorConfiguration
 - **PATCH** : partially update the specified ClickHouseOperatorConfiguration
 - **PUT** : replace the specified ClickHouseOperatorConfiguration

`/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseoperatorconfigurations`

HTTP method

DELETE

Description

delete collection of ClickHouseOperatorConfiguration

HTTP responses

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

HTTP method

GET

Description

list objects of kind ClickHouseOperatorConfiguration

HTTP responses

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

HTTP method

POST

Description

create a new ClickHouseOperatorConfiguration

Query parameters

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

Body parameters

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

HTTP responses

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

/apis/clickhouse.altinity.com/v1/namespaces/{namespace}/clickhouseoperatorconfigurations/{name}

HTTP method

`DELETE`

Description

delete the specified ClickHouseOperatorConfiguration

Query parameters

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

HTTP responses

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

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified ClickHouseOperatorConfiguration

HTTP responses

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

HTTP method

PATCH

Description

partially update the specified ClickHouseOperatorConfiguration

Query parameters

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

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

HTTP responses

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

HTTP method

PUT

Description

replace the specified ClickHouseOperatorConfiguration

Query parameters

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

Parameter	Type	Description
		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	ClickHouseOperatorConfiguration schema	application/json formatted

HTTP responses

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