

NiFi Config Reference

Date published: 2024-06-11

Date modified: 2024-06-11



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

NiFi config reference.....	4
Nifi.....	4
NifiSpec.....	4
NiFiStatus.....	6
ImageSpec.....	6
NifiZookeeperSpec.....	7
NifiConfigurationSpec.....	7
NifiPersistenceSpec.....	8
Assets.....	8
StatefulsetSpec.....	9
NifiPodDisruptionBudget.....	9
NifiSecuritySpec.....	9
NifiKerberosSpec.....	10
NifiResourceSpec.....	10
JVMDiagnosticSpec.....	11
StateManagementSpec.....	11
ProxyConfigSpec.....	11

NiFi config reference

The API types documented are Kubernetes CustomResourceDefinitions. They follow the standard [API-Conventions](#). The main objects monitored are the NiFi and NiFiRegistry, which have the standard Spec and Status.

Nifi

Nifi is the Schema for the nifis API.

Spec NifiSpec

Desired State of NiFi.

Status NifiStatus

Observed State of NiFi.

NifiSpec

NifiSpec defines the desired state of NiFi.

nifiVersion string

NiFi version to create a cluster for. Must have full pattern of Major.Minor.Patch. For example: 1.2.3, or 2.1.1

Replicas int32

Replicas defines the desired quantity of NiFi pods.

Image ImageSpec

Image for the NiFi container .

tiniImage ImageSpec

Tini image for sidecars / init containers.

ingressAccessType IngressRoutingSpec

ingressAccessType is deprecated, use uiConnection instead.

Zookeeper NifiZookeeperSpec

ZooKeeper instance specification.

configOverride NifiConfigurationSpec (Optional)

ConfigOverride specifies user provided input overriding default configurations.

persistenceNifiPersistenceSpec (Optional)

Persistence for the NiFi container.

Assets Assets (Optional)

Flow assets that the NiFi processor needs to access in the form of local file system.

Statefulset StatefulsetSpec (Optional)

Statefulset parameters for the underlying NiFi statefulset.

podDisruptionBudget NifiPodDisruptionBudget(Optional)

PodDisruptionBudget parameters for the underlying NiFi pod disruption budget.

Security NifiSecuritySpec (Optional)

Security specifies the security related configurations (HTTPS, LDAP, Kerberos...)

Resources NifiResourceSpec (Optional)

Resources specifies container limits by containers

S2s S2SSpec (Optional)

S2s specifies the site-to-site related configurations.

Debug JVMDebugSpec (Optional)

Debug specifies the details for JVM remote debugging.

stateManagement StateManagementSpec (Optional)

StateManagement specifies NiFi state management details. For more information, see [State management](#) in NiFi documentation.

sensitivePropsKeySecretName string (Optional)

SensitivePropsKeySecretName is the name of the secret which specifies the value for the `nifi.sensitive.props.key` in `nifi.properties`. If a secret with such name pre-exists, then the operator will use it. When left empty, the operator will create a secret with the name: `{cr-name}-sensitive-pr ops-key` and a randomly generated password as a value for `'nifi.sensitive.props.key'`.

proxyConfig ProxyConfigSpec (Optional)

proxyConfig contains details about a reverse proxy in front of the ingress controller. When set, it is used in the ingress object to set `'X-Proxy(Scheme|Host|Port)'` headers, which are used by NiFi to construct URIs which work properly with the proxy. For more information, see [Proxy Configuration](#) in NiFi documentation.

inboundConnection InboundConfig

inboundConnection defines the configuration for allowing inbound connection to listening processors in a NiFi flow.

hostName string (Optional)

hostName specifies the host name with which NiFi is reachable. The host name is used to configure the host of the Ingress/Route for uiConnection. When omitted, any arbitrary host header is feasible, for example. IP address. The host name should match the ingress certificate in case manual ingress certificate is set. HostName is required for “host named based virtual access” mode, but can be omitted in “fanout” mode (See `IngressAccessType` property).

additionalAttachments []Kubernetes core/v1.TypedLocalObjectReference (Optional)

AdditionalAttachments specifies a list of object references to be attached to all NiFi pods for general usage. Objects with Kind “Secret” and “ConfigMap” are accepted. The base path for attached volumes is `“/home/nifi/additional/{configmap|secret}”`, depending on its Kind. Attachments appear as a directory. Name of the attachment is the same as the name of the Kubernetes object, only dashes ‘-’ are replaced with underscores ‘_’. For example, a ConfigMap with name “my-config” will be mounted to `“/home/nifi/additional/configmap/my_config”`.

additionalPodLabels map[string]string

AdditionalPodLabels specifies key/value pairs that will be assigned to each NiFi pod resource as additional labels.

additionalPodAnnotations map[string]string

additionalPodAnnotations specifies key/value pairs that will be assigned to each NiFi pod resource as additional annotations.

Suspend bool (Optional)

Suspend reconciliation of this NiFi cluster

suspendCluster bool (Optional)

Suspend this NiFi cluster's operation by scaling replicas to zero without offloading data or deleting PVCs.

Tolerations []Kubernetes core/v1.Toleration

Tolerations sets the tolerations on each NiFi pod.

Connections []Connection

Connections configures Ingresses, Routes, and Services for connecting to NiFi.

uiConnection Connection (Optional)

uiConnection configures the Ingress, Route, or Service used for accessing NiFi's WebUI.

NiFiStatus

NiFiStatus defines the observed state of NiFi.

replicas int32

replicas defines the current quantity of NiFi pods.

lastCleanedPodOrdinal int32 (Optional)

LastCleanedPodOrdinal is the pod ordinal of last cleaned NiFi node. Used in downscaling: PersistentVolumeClaim, PersistentVolume and Certificate will be deleted, if the pod has already been deleted. Before the first scaling operation, this attribute is hidden. In normal case, LastCleanedPodOrdinal = Replicas. During downscaling, LastCleanedPodOrdinal > Replicas, until all pods are cleaned up.

selector string (Optional)

selector is used by Horizontal Pod Autoscaling to match.

conditions []Kubernetes meta/v1.Condition

conditions are a list of conditions defined by the operator.

statefulsetUpdateRevision string

statefulsetUpdateRevision holds the last observed updateRevision of the underlying statefulSet resource.

minOffloadOrdinal int32 (Optional)

minOffloadOrdinal is the lower boundary of the pod ordinal whose data needs to be offloaded during a downscaling.

ImageSpec

Generic Image spec used to specify image repository, tag, pull policy, and pull secret for a container.

repository string

Image repository from which to pull.

tag string

Image tag to pull from repository.

pullPolicy string (Optional)

ImagePullPolicy for this image. Defaults to IfNotPresent. Valid values are IfNotPresent, Always, and Never.

pullSecret string (Optional)

Name of the secret which contains docker-registry credentials for the repository.

NifiZookeeperSpec

NifiZookeeperSpec defines the ZooKeeper options for NiFi.

host string

host is the comma separated list of hosts of the ZooKeeper nodes, or the name of the headless service on top of them.

port int

port is the client port number of the ZooKeeper instance(s), the default value is: 2181.

secure bool (Optional)

secure determines whether two-way TLS should be used when communicating with ZooKeeper.

NifiConfigurationSpec

NifiConfigurationSpec contains user provided values which override the default configuration values that are shipped with NiFi.

nifiProperties ConfigOperation (Optional)

nifiProperties holds configuration file overrides for the nifi.properties configuration file. If set, this content will override the default nifi.properties configuration file.

bootstrapConf ConfigOperation (Optional)

bootstrapConf holds configuration file overrides for the bootstrap.conf configuration file. If set, this content will override the default bootstrap.conf configuration file.

authorizers string (Optional)

authorizers holds configuration file overrides for the authorizers.xml configuration file. If set, this content will override the default authorizers.xml configuration file.

logback LogbackOverrideSpec (Optional)

logback holds configuration file overrides for logback.xml configuration file. It can either specify the content of the logback.xml as string, or a list of LogbackSpec.

loginIdentityProviders string (Optional)

loginIdentityProviders holds configuration file overrides for the login-identity-providers.xml configuration file. If set, this content will override the default login-identity-providers.xml file.

authorizersRef Kubernetes core/v1.ObjectReference (Optional)

authorizersObjectReferences holds details about a custom configmap or a secret for the authorizers.xml file.

bootstrapConfRef Kubernetes core/v1.ObjectReference (Optional)

bootstrapConfRef holds details about a custom configmap or a secret for the bootstrap.conf configuration file.

nifiPropertiesRef Kubernetes core/v1.ObjectReference (Optional)

nifiPropertiesRef holds details about a custom configmap or a secret for the nifi.properties configuration file.

logbackRef Kubernetes core/v1.ObjectReference (Optional)

logbackRef holds details about a custom configmap or a secret for the logback.xml configuration file.

loginIdentityProvidersRef Kubernetes core/v1.ObjectReference (Optional)

loginIdentityProvidersRef holds details about a custom configmap or a secret for the login-identity-providers.xml configuration file.

stateManagementRef Kubernetes core/v1.ObjectReference (Optional)

stateManagementRef holds details about a custom configmap or a secret for the state-management.xml configuration file.

NifiPersistenceSpec

NifiPersistenceSpec defines the storage specification for NiFi.

storageClass string (Optional)

storageClass is the name of the storage class used in the persistent volume. If set to “-”, then sets the storage class name “”, which disables dynamic provisioning. If undefined (the default) or set to null then no storage class name is set, choosing the default provisioner (gp2 on AWS, standard on GKE, AWS, and OpenStack).

size string (Optional)

Size of the persistent volume. The default value is 5Gi

state PersistenceSpec (Optional)

state describes the storage specification for the ‘state’ directory, which is used to hold information about the node’s current state. If provided, it overrides the default storage specification.

data PersistenceSpec (Optional)

data describes the storage specification for the ‘data’ directory, which is used to hold items such as the flow.xml.gz, configuration, state, and so on. If provided, it overrides the default storage specification.

flowfileRepo PersistenceSpec (Optional)

flowfileRepo describes the storage specification for the flowfile repository. If provided, it overrides the default storage specification.

contentRepo MultiplePersistenceSpec (Optional)

contentRepo describes the storage specification for the content repository. If provided, it overrides the default storage specification. This parameter supports multiple repositories as values.

provenanceRepo MultiplePersistenceSpec (Optional)

provenanceRepo describes the storage specification for the provenance repository. If provided, it overrides the default storage specification. This parameter supports multiple repositories as values.

Assets

Assets represent a Persistent Volume (PV) shared across all Pods to serve sharable assets. Such a PV must support ReadWriteMany or ReadWriteMany access mode, such as NFS.

mountPath string

The mount path in a container inside a Pod where you can access shareable assets.

persistentVolumeClaim Kubernetes core/v1.LocalObjectReference

The PersistentVolumeClaim bound to the asset PersistentVolume

StatefulsetSpec

StatefulsetSpec contains the parameters for the underlying statefulset.

podManagementPolicy string (Optional)

Defines the pod management policy of the underlying statefulset. The default value is Parallel.

antiAffinity string (Optional)

Defines the anti-affinity policy for the pods. The default value is Soft.

livenessProbe ProbeSpec (Optional)

livenessProbe is a periodic probe of container liveness. If the probe fails, the container is restarted. This parameter cannot be updated.

readinessProbe ProbeSpec (Optional)

readinessProbe is a periodic probe of container service readiness. If the probe fails, the container is removed from service endpoints. This parameter cannot be updated.

terminationGracePeriodSeconds int64(Optional)

terminationGracePeriodSeconds is the time allotted for a graceful termination of a pod's containers before a SIGKILL signal is sent.

NifiPodDisruptionBudget

NifiPodDisruptionBudget contains the parameters for the NiFi Pod Disruption Budget.

maxUnavailable k8s.io/apimachinery/pkg/util/intstr.IntOrString (Optional)

maxUnavailable sets the maximum number of pods from the stateful set that can be unavailable after an eviction. You can either specify an absolute number or a percentage.

NifiSecuritySpec

NifiSecuritySpec contains the security configuration for the custom resource.

initialAdminIdentity string (Optional)

initialAdminIdentity specifies the initial admin user which has full access.

ingressCertGen IngressCertSpec (Optional)

ingressCertGen specifies ingress certificate generation related information.

s2sCertGen S2SCertSpec (Optional)

s2sCertGen specifies ingress certificate generation related information.

nodeCertGen NodeCertSpec (Optional)

nodeCertGen specifies node and proxy certificate generation related information.

additionalCABundles AdditionalCACertSpec (Optional)

additionalCABundles contains additional CA certificates to be used.

userCertAuth UserCertSpec (Optional)

userCertAuth specifies user certificate authentication related configurations.

samlSingleSignOnAuth SamlSingleSignOnAuthSpec (Optional)

samlSingleSignOnAuth specifies SAML-based single sign-on authentication related configurations. SAML authentication is enabled if SamlAuth is specified.

ldap LdapSpec (Optional)

ldap specifies the various LDAP authentication related configurations.

kerberos NifiKerberosSpec (Optional)

kerberos specifies configs related to user authentication through Kerberos.

krb5confSecret Kubernetes core/v1.LocalObjectReference (Optional)

krb5confSecret specifies the location of a Kerberos configuration file which will be added to the nifi.properties file. It does not enable Kerberos based authentication, but it can be used to integrate a NiFi cluster with services using Kerberos. The specified secret has to have a field with key “krb5.conf” and the value has to be the base64 encoded contents of the configuration file.

openIDAuth OpenIDSpec (Optional)

openIDAuth specifies OpenID authentication related configurations.

customAuthorizer CustomAuthorizerSpec (Optional)

customAuthorizer specifies a custom UserGroupProvider. The authorizer has to be present in the NiFi image being used, as this property only sets the authorizers.xml file accordingly. If not set, a default provider is used, which provides users for NiFi nodes and proxy.

NifiKerberosSpec

NifiKerberosSpec contains configuration for Kerberized environments.

realm string

realm is the Kerberos realm (for example: EXAMPLE.COM)

kdc string

KDC is the Key Distribution Center address

spnegoPrincipal string

SPNEGOPrincipal is the service principal used for SPNEGO access

spnegoKeytabName string

SPNEGOKeytabName is the keytab name for the SPNEGO service principal, for example, http.keytab.

authExpiration string (Optional)

authExpiration is the duration of how long the user authentication is valid for. Even if the user never logs out, they will be required to log back in following this duration.

servicePrincipal string

servicePrincipal is the kerberos service principal.

servicePrincipalKeytabName string

servicePrincipalKeytabName is the keytab location for the service principal's keytab.

NifiResourceSpec

NifiResourceSpec collects resource requests and limits for nifi, log and site-to-site proxy containers.

nifi ResourcesSpec (Optional)

nifi specifies how much CPU and memory (RAM) each Nifi container needs / can have.

log ResourcesSpec(Optional)

log specifies how much CPU and memory (RAM) each sidecar log container needs / can have.

s2s ResourcesSpec(Optional)

s2s specifies how much CPU and memory (RAM) the site-to-site proxy deployment pods need / can have.

JVMDebugSpec

JVMDebugSpec specifies the debug configuration to be applied to the target JVM.

suspend bool (Optional)

suspend, if set to true, instructs the target JVM to suspend its boot process until a debug client connects. Consider, that delaying a JVM start can cause the underlying pod's liveness-probe to kick in, and terminate the pod. To avoid that, for debugging purposes, you could specify: statefulset: livenessProbe: failureThreshold: to an arbitrary big number.

port int32

port defines the debug port on which the target JVM listens.

StateManagementSpec

StateManagement specifies NiFi state management details.

localProvider StateProviderSpec (Optional)

localProvider specifies the local-provider configurations.

clusterProvider StateProviderSpec (Optional)

clusterProvider specifies the cluster-provider configurations.

ProxyConfigSpec

ProxyConfigSpec specifies the proxy details which may be in front of the ingress controller.

Values set here are passed as X-Proxy... headers to NiFi, which will generate URIs based on those values.

scheme string (Optional)

scheme defines the scheme of the outermost reverse-proxy which is addressed by the client directly. Scheme can be either http or https. kubebuilder:validation:Enum=http,https

host string (Optional)

host is the name of the outermost reverse-proxy which is directly addressed by the client directly.

port int (Optional)

port defines the port of the outermost reverse-proxy which is addressed by the client directly.