

The zeebe.cfg.toml file

The following snippet represents the default Zeebe configuration, which is shipped with the distribution. It can be found inside the `config` folder (`config/zeebe.cfg.toml`) and can be used to adjust Zeebe to your needs.

[Source on github](#)



```
# Zeebe broker configuration file

# Overview -----

# This file contains a complete list of available configuration options.

# Default values:
#
# When the default value is used for a configuration option, the option is
# commented out. You can learn the default value from this file

# Conventions:
#
# Byte sizes
# For buffers and others must be specified as strings and follow the following
# format: "10U" where U (unit) must be replaced with K = Kilobytes, M =
# Megabytes or G = Gigabytes.
# If unit is omitted then the default unit is simply bytes.
# Example:
# sendBufferSize = "16M" (creates a buffer of 16 Megabytes)
#
# Time units
# Timeouts, intervals, and the likes, must be specified as strings and follow
# the following
# format: "VU", where:
#   - V is a numerical value (e.g. 1, 1.2, 3.56, etc.)
#   - U is the unit, one of: ms = Millis, s = Seconds, m = Minutes, or h = Hours
#
# Paths:
# Relative paths are resolved relative to the installation directory of the
# broker.

# -----

[gateway]
# Enable the embedded gateway to start on broker startup.
# This setting can also be overridden using the environment variable
# ZEEBE_EMBED_GATEWAY.
# enable = true

[gateway.network]
# Sets the host the embedded gateway binds to.
# This setting can be specified using the following precedence:
# 1. setting the environment variable ZEEBE_GATEWAY_HOST
# 2. setting gateway.network.host property in this file
# 3. setting the environment variable ZEEBE__HOST
# 4. setting network.host property in this file
# host = "0.0.0.0"

# Sets the port the embedded gateway binds to.
# This setting can also be overridden using the environment variable
# ZEEBE_GATEWAY_PORT.
# port = 26500

[gateway.cluster]
# Sets the broker the gateway should initial contact.
```

```
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_CONTACT_POINT.
# contactPoint = "127.0.0.1:26501"

# Sets size of the transport buffer to send and received messages between
gateway and broker cluster.
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_TRANSPORT_BUFFER.
# transportBuffer = "128M"

# Sets the timeout of requests send to the broker cluster
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_REQUEST_TIMEOUT.
# requestTimeout = "15s"

[gateway.threads]
# Sets the number of threads the gateway will use to communicate with the broker
cluster
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_MANAGEMENT_THREADS.
# managementThreads = 1

[gateway.monitoring]
# Enables the metrics collection in the gateway
# enabled = false

[gateway.security]
# Enables TLS authentication between clients and the gateway
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_SECURITY_ENABLED.
# enabled = false
#
# Sets the path to the certificate chain file
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_CERTIFICATE_PATH.
# certificateChainPath = ""
#
# Sets the path to the private key file location
# This setting can also be overridden using the environment variable
ZEEBE_GATEWAY_PRIVATE_KEY_PATH.
# privateKeyPath = ""

[network]

# This section contains the network configuration. Particularly, it allows to
# configure the hosts and ports the broker should bind to. The broker exposes
two sockets:
# 1. command: the socket which is used for gateway-to-broker communication
# 2. internal: the socket which is used for broker-to-broker communication
# 3. monitoring: the socket which is used to monitor the broker

# Controls the default host the broker should bind to. Can be overwritten on a
# per binding basis for client, management and replication
#
# This setting can also be overridden using the environment variable ZEEBE_HOST.
# host = "0.0.0.0"

# Controls the advertised host; if omitted defaults to the host. This is
```

```

particularly useful if your
# broker stands behind a proxy.
# advertisedHost = "0.0.0.0"

# If a port offset is set it will be added to all ports specified in the config
# or the default values. This is a shortcut to not always specifying every port.
#
# The offset will be added to the second last position of the port, as Zeebe
# requires multiple ports. As example a portOffset of 5 will increment all ports
# by 50, i.e. 26500 will become 26550 and so on.
#
# This setting can also be overridden using the environment variable
ZEEBE_PORT_OFFSET.
# portOffset = 0

# Sets the maximum size of the incoming and outgoing messages (i.e. commands and
# events).
# maxMessageSize = "4M"

[network.commandApi]
# Overrides the host used for gateway-to-broker communication
# host = "localhost"

# Sets the port used for gateway-to-broker communication
# port = 26501

[network.internalApi]
# Overrides the host used for internal broker-to-broker communication
# host = "localhost"

# Sets the port used for internal broker-to-broker communication
# port = 26502

[network.monitoringApi]
# Overrides the host used for exposing monitoring information
# host = "localhost"

# Sets the port used for exposing monitoring information
# port = 9600

[data]

# This section allows to configure Zeebe's data storage. Data is stored in
# "partition folders". A partition folder has the following structure:
#
# partition-0                                (root partition folder)
# └─ partition.json                          (metadata about the partition)
# └─ segments                               (the actual data as segment files)
#   └─ 00.data
#   └─ 01.data
# └─ state                                  (stream processor state and snapshots)
#   └─ stream-processor
#       └─ runtime
#       └─ snapshots

# Specify a list of directories in which data is stored. Using multiple
# directories makes sense in case the machine which is running Zeebe has

```

```
# multiple disks which are used in a JBOD (just a bunch of disks) manner. This
# allows to get greater throughput in combination with a higher io thread count
# since writes to different disks can potentially be done in parallel.
#
# This setting can also be overridden using the environment variable
ZEEBE_DIRECTORIES.
# directories = [ "data" ]

# The size of data log segment files.
# logSegmentSize = "512M"

# How often we take snapshots of streams (time unit)
# snapshotPeriod = "15m"

# The maximum number of snapshots kept (must be a positive integer). When this
# limit is passed the oldest snapshot is deleted.
# maxSnapshots = "3"
```

[cluster]

```
# This section contains all cluster related configurations, to setup an zeebe
cluster

# Specifies the unique id of this broker node in a cluster.
# The id should be between 0 and number of nodes in the cluster (exclusive).
#
# This setting can also be overridden using the environment variable
ZEEBE_NODE_ID.
# nodeId = 0

# Controls the number of partitions, which should exist in the cluster.
#
# This can also be overridden using the environment variable
ZEEBE_PARTITIONS_COUNT.
# partitionsCount = 1

# Controls the replication factor, which defines the count of replicas per
partition.
# The replication factor cannot be greater than the number of nodes in the
cluster.
#
# This can also be overridden using the environment variable
ZEEBE_REPLICATION_FACTOR.
# replicationFactor = 1

# Specifies the zeebe cluster size. This value is used to determine which broker
# is responsible for which partition.
#
# This can also be overridden using the environment variable ZEEBE_CLUSTER_SIZE.
# clusterSize = 1

# Allows to specify a list of known other nodes to connect to on startup
# The contact points of the internal network configuration must be specified.
# The format is [HOST:PORT]
# Example:
# initialContactPoints = [ "192.168.1.22:26502", "192.168.1.32:26502" ]
#
# This setting can also be overridden using the environment variable
```

```
ZEEBE_CONTACT_POINTS
# specifying a comma-separated list of contact points.
#
# To guarantee the cluster can survive network partitions, all nodes must be
# specified
# as initial contact points.
#
# Default is empty list:
# initialContactPoints = []
```

```
# Allows to specify a name for the cluster
# This setting can also be overridden using the environment variable
ZEEBE_CLUSTER_NAME
# Example:
# clusterName = "zeebe-cluster"
```

[threads]

```
# Controls the number of non-blocking CPU threads to be used. WARNING: You
# should never specify a value that is larger than the number of physical cores
# available. Good practice is to leave 1-2 cores for ioThreads and the operating
# system (it has to run somewhere). For example, when running Zeebe on a machine
# which has 4 cores, a good value would be 2.
#
# The default value is 2.
#cpuThreadCount = 2
```

```
# Controls the number of io threads to be used. These threads are used for
# workloads that write data to disk. While writing, these threads are blocked
# which means that they yield the CPU.
#
# The default value is 2.
#ioThreadCount = 2
```

```
# Configure exporters below; note that configuration parsing conventions do not
# apply to exporter
# arguments, which will be parsed as normal TOML.
#
# Each exporter should be configured following this template:
#
# id:
#   property should be unique in this configuration file, as it will server as
#   the exporter
#   ID for loading/unloading.
# jarPath:
#   path to the JAR file containing the exporter class. JARs are only loaded
#   once, so you can define
#   two exporters that point to the same JAR, with the same class or a different
#   one, and use args
#   to parametrize its instantiation.
# className:
#   entry point of the exporter, a class which must extend the
#   io.zeebe.exporter.Exporter
#   interface.
#
# A nested table as [exporters.args] will allow you to inject arbitrary
# arguments into your
# class through the use of annotations.
```

```
#
# Enable the following debug exporter to log the exported records to console
# This exporter can also be enabled using the environment variable ZEEBE_DEBUG,
# the pretty print
# option will be enabled if the variable is set to "pretty".
#
# [[exporters]]
# id = "debug-log"
# className = "io.zeebe.broker.exporter.debug.DebugLogExporter"
# [exporters.args]
#   logLevel = "debug"
#   prettyPrint = false
#
# Enable the following debug exporter to start a http server to inspect the
# exported records
#
# [[exporters]]
# id = "debug-http"
# className = "io.zeebe.broker.exporter.debug.DebugHttpExporter"
# [exporters.args]
#   port = 8000
#   limit = 1024
#
#
# An example configuration for the elasticsearch exporter:
#
# [[exporters]]
# id = "elasticsearch"
# className = "io.zeebe.exporter.ElasticsearchExporter"
#
# [exporters.args]
# url = "http://localhost:9200"
#
# [exporters.args.bulk]
# delay = 5
# size = 1_000
#
# [exporters.args.authentication]
# username = elastic
# password = changeme
#
# [exporters.args.index]
# prefix = "zeebe-record"
# createTemplate = true
#
# command = false
# event = true
# rejection = false
#
# deployment = true
# error = true
# incident = true
# job = true
# jobBatch = false
# message = false
# messageSubscription = false
# variable = true
# variableDocument = false
```

```
# workflowInstance = true  
# workflowInstanceCreation = false  
# workflowInstanceSubscription = false
```