

HAZELCAST PLATFORM DATASHEET

The Real-Time Stream Processing Platform

Hazelcast is a software technology for building real-time applications that act on data immediately. hazelcast.com/platform

Feature Summary

- ◆ **Client programming languages.** Java, .NET, Node.js, C++, Go, Python, Scala.
- ◆ **Connectors.** Apache Kafka, Apache Pulsar, AWS Kinesis, MongoDB, JDBC, and more; see official documentation for the up-to-date list; can use any Kafka Connect module.
- ◆ **JVM versions.** JDK 8, 11 - 17.
- ◆ **Supported JVM vendors.** Oracle JDK, OpenJDK, IBM JDK, Azul Zing & Zulu.
- ◆ **Integrations.** Spring, Apache Tomcat, Apache Spark, Apache Kafka, and more; see the [Hazelcast Connector Hub](#) for the up-to-date list.
- ◆ **SQL querying.** Query/update data using industry-standard SQL.
- ◆ **Stream Processing Engine.** Read and process continuous, incoming data flows, to act upon the data as soon as it is created.
- ◆ **Low-Latency Data Store.** Fast, scalable, cloud-native (Kubernetes + containers), fault-tolerant, object and JSON store with parallel application execution and strong data consistency mode.
- ◆ **Management Center.** Administration UI for per-node statistics, JMX APIs, REST APIs. Limited to 3 nodes for the open source version.

The Hazelcast Platform integrates a powerful stream processing engine with low-latency storage.

The Hazelcast Platform lets you build real-time applications with the unique integration of stream processing and low-latency storage. Hazelcast is an open source software platform, but is also available in a commercial edition (“Enterprise”) that adds more business-critical capabilities, and in the Hazelcast Viridian cloud-managed services.

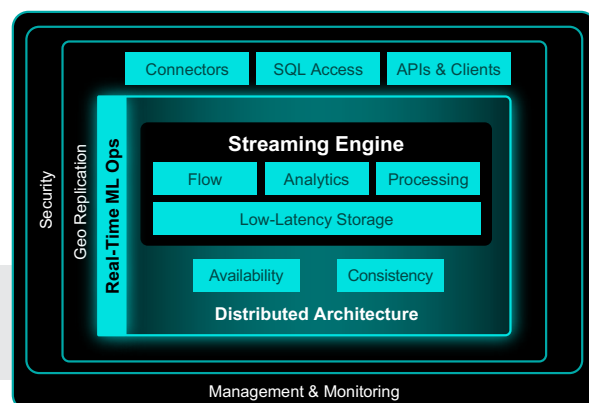
Why Use the Hazelcast Platform

The Hazelcast Platform helps you easily add real-time capabilities to your existing infrastructure. It also boosts performance to let you take on new projects that you previously thought were not possible or practical. It is architected for zero downtime deployments to run your 24/7 operations. The Hazelcast Platform provides high-performance, real-time processing via:

- ◆ **Fast access to data.** The low-latency data store enables fast reads and writes.
- ◆ **Parallel execution of application code.** Submit applications that are run in a distributed architecture to take advantage of the collective CPU power of your cluster.
- ◆ **Stream processing and streaming analytics.** Use Hazelcast with messaging technologies like Apache Kafka, Apache Pulsar, and AWS Kinesis, to analyze and automatically act upon new trends and patterns in data.

Hazelcast is designed for simplicity, and is packaged as a linkable library. If you run multiple instances of your application in a network, each instance will broadcast to identify the other instances and will automatically form a cluster for you. You can also run Hazelcast in client-server mode and submit application code as a job, which will get distributed to all nodes and then run in parallel.

Hazelcast is also very reliable with its high availability and disaster recovery (HA/DR) features. It replicates data across the cluster to safeguard against data loss upon hardware failure. Its WAN Replication (Enterprise version only) efficiently copies data to a remote cluster for disaster recovery (as well as geo-distribution).



Feature Summary for Enterprise Version

- ✦ **Built-in security.** LDAP, Kerberos, x.509 certificates, role-based access controls, TLS/SSL, x.509, data-at-rest encryption.
- ✦ **WAN Replication.** Efficiently replicate data to a remote cluster.
- ✦ **Automatic Failover.** Clients automatically switch to the secondary cluster if the primary cluster is down.
- ✦ **Tiered Storage.** Store larger volumes of reference data within the single integrated platform by complementing RAM with disk storage.
- ✦ **High-Density Memory Store.** Manage more RAM (up to 200 GB) by using off-heap memory to avoid garbage collection pauses. Support for Intel Optane as a cost-effective alternative to DRAM.
- ✦ **Persistence.** Stores in-memory data to disk which can be used to quickly repopulate the data store after node shutdowns.
- ✦ **Rolling Upgrades.** Allows upgrade of Hazelcast software in a cluster one node at a time without downtime.
- ✦ **Blue/Green Deployment Support.** Enables easy client switchover in a dual-cluster upgrade strategy.
- ✦ **PaaS Support.** OpenShift, VMware Tanzu (formerly Pivotal Cloud Foundry).

Example Use Cases

- ✦ **Digital integration hub / high-performance data layer.**
- ✦ **Payment processing.**
- ✦ **E-commerce transaction processing.**
- ✦ **Real-time machine learning inference.**
- ✦ **Internet of Things analytics and edge processing.**
- ✦ **Large-scale risk assessment calculations.**

Architecture

Hazelcast was built from the ground up as a cloud-native, real-time stream processing platform to support high performance, scale, elasticity, and fault tolerance. It runs in containerized environments orchestrated by Kubernetes. It features an API that is familiar to many application developers.

- ✦ **Performance at scale.** Hazelcast has run numerous benchmarks over the years to ensure the performance that customers need. Published benchmarks are available [here](#).
- ✦ **SQL support.** Use industry-standard SQL to query and update data or join it with streaming data. Leverage indexes to rapidly query on attributes in addition to keys. Streaming SQL lets you build real-time stream processing applications in a familiar paradigm.
- ✦ **Built-in security (Enterprise version only).** Hazelcast offers a comprehensive security framework that includes support for Java Authentication and Authorization Service (JAAS), pluggable authentication (including LDAP and Kerberos out-of-the-box), role-based access controls, encryption of data-in-motion via TLS/SSL with minimal performance impact, x.509 certificates for client identity or mutual authentication, encryption for persisted data-at-rest; key rotation, and additional security APIs for advanced security controls.
- ✦ **WAN Replication (Enterprise version only).** Hazelcast efficiently replicates data to a remote cluster for disaster recovery (DR), geo-distribution, or multi-cloud strategies; supports active-active or active-passive topologies. Automatic Failover lets clients automatically switch to the secondary cluster should the first cluster become inaccessible.
- ✦ **Tiered Storage (Enterprise version only).** Store larger volumes of data using disk-based storage to complement RAM storage.
- ✦ **User-defined functions.** Hazelcast lets you easily plug in external Python, C++, or Java code as part of a stream processing pipeline. When using Python, Hazelcast starts up Python VMs in the same nodes as the processing tasks to reduce network latency, and maintains the VMs to ensure high availability.
- ✦ **Exactly-once stream processing.** Hazelcast handles streaming transactions with exactly-once processing, even upon failures. It supports any replayable data source (e.g., Kafka, Pulsar, Hazelcast IMap journal) and any idempotent data sink (i.e., with upsert capabilities). It also supports two-phase commit logic to allow more sources (e.g., JMS) and sinks (e.g., Kafka, Pulsar, files) that are not replayable/idempotent.

To get started with Hazelcast, visit <https://hazelcast.com/get-started>.