



## 1. Apache Ignite In-Memory Data Fabric

---

Ignite In-Memory Data Fabric is designed to deliver uncompromised performance for the widest array of in-memory computing use cases.

Following main components are included in the fabric:

- **Advanced Clustering** - support for auto-discovery of cluster nodes in any environment, including, public clouds (e.g. AWS), private clouds, or hybrid clouds.
- **Compute Grid** - includes distributed clustering, messaging, events, and computational features.
- **Data Grid** - partitioned in-memory key-value store with support for ACID transactions, off-heap memory, SQL, and more.
- **Service Grid** - support for managed user-defined services, like cluster singletons, node-singletons, and services with custom deployment topology.
- **Distributed Data Structures** - support for common distributed data structures, like maps, sets, queues, atomics, etc.
- **Streaming & CEP** - supports event workflow, rolling data windows and indexing, continuous querying, and more.
- **Distributed Filesystem** - distributed Hadoop-compliant in-memory file system.
- **Distributed Messaging** - support for topic-based and point-to-point message exchange between cluster nodes.
- **Distributed Events** - support for cluster-wide event notifications.

## 2. Apache Ignite Installation

---

Ignite distribution comes in a ZIP file that simply needs to be unzipped, and `IGNITE_HOME` environment variable can optionally be set to point to it.

There are no additional steps required for Ignite installation in such multi machine setup.

Installation requirements:

1. Windows, Linux, or MacOS environment.
2. Java 7 or 8 (latest update is advisable).
3. Point `JAVA_HOME` environment variable to your JDK or JRE installation.
4. Optional: point `IGNITE_HOME` environment variable to the Ignite installation folder.

### 2.1 Check Ignite Installation

To verify Ignite installation, you can execute the Ignite startup script.

The following command will startup Ignite with default configuration using Multicast node discovery.

```
bin/ignite.{sh|bat}
```

The following command will startup Ignite with example configuration.

```
bin/ignite.{sh|bat} examples/config/example-compute.xml
```

If Ignite was installed successfully, the output from above commands should produce no exceptions or errors. Note that you may see some warnings during startup, but this is OK as they are meant to inform that certain functionality is turned on or off by default.

You can execute the above commands multiple times on the same machine and make sure that nodes discover each other. Here is an example of log printout when 2 nodes join topology:

```
... Topology snapshot [nodes=2, CPUs=8, hash=0xD551B245]
```

You can also start Ignite Management Console, called Visor, and observe started nodes. To startup Visor, you should execute the following script:

```
/bin/ignitevisorcmd.{sh|bat}
```

## 2.2 Running Ignite Examples

Ignite comes with many well documented examples. All examples have documentation about how they should be started and what the expected outcome should be.

Use provided pom.xml to import examples into IDE of your choice.

## 3. Maven

---

Apache Ignite hosts its Maven artifacts in Apache maven repository as well as in Maven Central.

### 3.1 Maven Artifacts

You can use maven to add Ignite artifacts to your project. Ignite has one main artifact called `ignite-fabric`. You can also import individual maven artifacts a la carte to bring in more optional dependencies, like `ignite-aws` for AWS integration, for example. All optional maven dependencies are also available in the Ignite installation under `libs/optional` folder.

### 3.2 Maven Example

```
<dependency>
  <groupId>org.apache.ignite</groupId>
  <artifactId>ignite-fabric</artifactId>
  <version>${ignite.version}</version>
  <type>pom</type>
</dependency>
```

## 4. Starting Ignite Nodes

---

Ignite nodes can be started by executing `bin/ignite.{sh|bat}` script and passing a relative path to Ignite configuration file. If no file is passed, then grid nodes are started with default configuration using Multicast discovery protocol.

Here is an example of how to start Ignite node with non-default configuration:

```
`bin/ignite.sh examples/config/example-cache.xml`
```

## 5. Management & Monitoring with Visor

---

Ignite comes with CLI (command) based DevOps Managements Console, called Visor, delivering advance set of management and monitoring capabilities.

To start Visor in console mode you should execute the following command:

```
`bin/ignitevisorcmd.sh`
```

On Windows, run the same commands with `.bat` extension.

## 6. Scala Integration

---

Ignite provides a very nice and easy to use DSL for Scala users called `Scalar`. If you like Scala, take a look at Scalar examples located under `examples/src/main/scala` folder.

## 7. Javadoc & Scaladoc

---

All documentation is shipped with it and you can find it under `docs/javadoc` and `docs/scaladoc` sub-folder respectively.