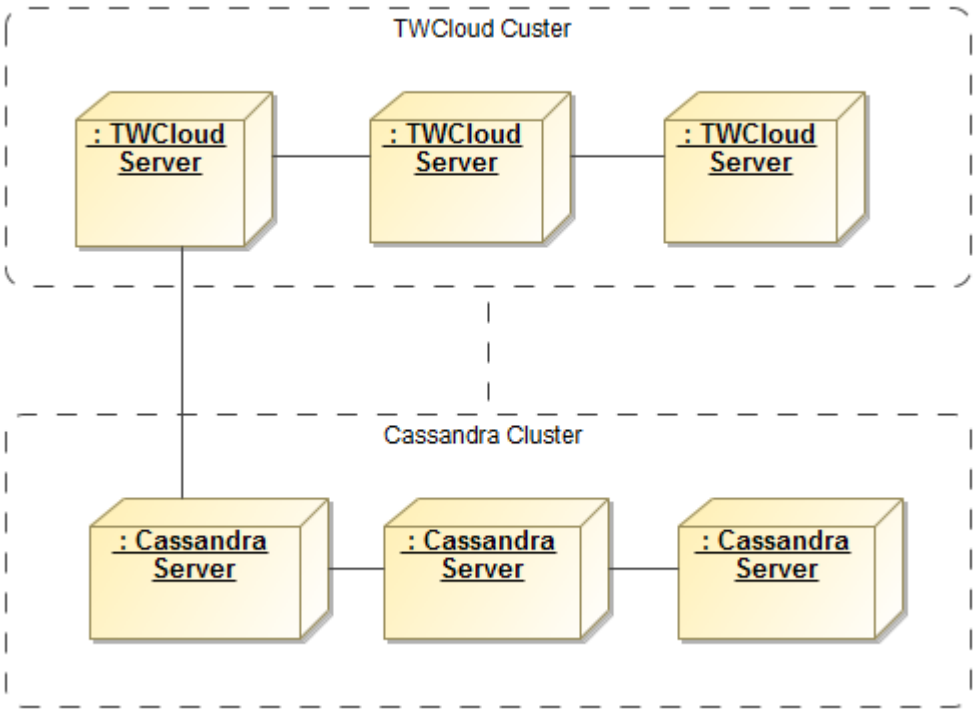
TWCloud Cluster Setup

On this page:

This page provides instructions to set up a Teamwork Cloud (TWCloud) Cluster on your system. A TWCloud Cluster is composed of two clustering layers, TWCloud and Cassandra.

To set up a TWCloud Cluster, you need to perform two separate tasks:

1. Set up a Cassandra Cluster
2. Set up a TWCloud Cluster



An illustration of TWCloud Cluster and Cassandra Cluster nodes within TWCloud Cluster.

Setting up a Cassandra Cluster

Prior to establishing a Cassandra cluster, you need to determine the following.

- The initial number of nodes in the cluster.
- The IP address of each node.
- Determine which node will be the seed.

When setting up a Cassandra node, you need to configure a seed that indicates an initial contact point for the new node. So, you can configure the seed using the IP of any existing active node in the cluster. Use the node's own IP when configuring the first node. For example, if you are configuring a three-node cluster and the nodes' IPs are *10.1.1.101*, *10.1.1.102,* and *10.1.1.103*. Select one, for example *10.1.1.101*, as the **seed**. While you are configuring **cassandra.yaml**, specify *10.1.1.101* as the seed value for all of the three nodes.

Now that you understand what a seed is and how to configure it, follow the instructions to install and configure Cassandra on Windows or Linux Operating System.

- Windows instructions
- Linux instructions

Please note that **listen_address** and **broadcast_rpc_address** are still their machine.

Machine IP	10.1.1.101	10.1.1.102	10.1.1.103
seeds	**10.1.1.101**	**10.1.1.101**	**10.1.1.101**
listen_address	10.1.1.101	10.1.1.102	10.1.1.103
broadcast_rpc_address	10.1.1.101	10.1.1.102	10.1.1.103

After installing and configuring all nodes, start the seed node. Verify that it is operational by issuing the command "nodetool status".

Once the seed node is operational, you can proceed to start the other nodes in the cluster. You must allow the node to join the cluster before proceeding to the next node. After all nodes have started, the "nodetool status" command will display results as shown below.

```
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--   Address     Load      Tokens    Owns (effective)    Host ID                               Rack
UN   10.1.1.101  6.7 GB    256       35.1%               b33c603a-95c7-426d-9f7b-ebad2375086a  rack1
UN   10.1.1.102  6.34 GB   256       32.5%               16f40503-4a65-45fe-9ee7-5d942506aa87  rack1
UN   10.1.1.103  6.15 GB   256       32.4%               2d90c119-08a4-4799-ac73-0440215d0b18  rack1
```

Node Consistency

In order to ensure that all nodes are consistent, please run the "nodetool repair" command nightly. This can be scheduled as a crontab job or via Windows Task Scheduler. This only needs to be executed on a single node.

Setting up a TWCloud Cluster

Before setting up a TWCloud Cluster, you need to determine the following.

- The initial number of nodes in the cluster.
- The IP address of each TWC node.
- Prepare a list of IPs of Cassandra nodes.

The instructions below use the following sample environment for ease of understanding.

- 3 nodes TWC cluster.
- TWC node IP addresses are *10.1.1.111, 10.1.1.112*, and *10.1.1.113.*
- All nodes will be used as seed nodes.
- IP addresses of Cassandra nodes are *10.1.1.101, 10.1.1.102*, *10.1.1.103.*

Setting up the TWCloud Cluster involves 2 groups of parameters in **application.conf**, and 1 group of parameters in **authserver.properties.**

application.conf changes

TWCloud clustering parameter

- **akka.cluster.seed-nodes**

This parameter indicates the initial contact points for the cluster.

If you install TWCloud using the installer file, you will be asked to provide the seed node IP during installation process and the value will be configured in **application.conf**.

If you install using the zip file, you will need to manually configure the parameter in **application.conf**. Search for the following:

application.conf

```
seed-nodes = ["akka.tcp://twcloud@${seed-node.ip}:2552"]
```

Replace *${seed-node.ip}* with the IP addresses of the seed nodes, so it should look similar to the following:

application.conf

```
seed-nodes = ["akka.tcp://twcloud@10.1.1.111:2552","akka.tcp://twcloud@10.1.1.112:2552","akka.
tcp://twcloud@10.1.1.113:2552"]
```

Cassandra configuration parameters

- **esi.persistence.cassandra.connection.seeds**

The value of this parameter is a list of Cassandra node IP addresses. You can find this parameter in **application.conf** by looking for the following.

application.conf

```
# List of comma delimited host.
# Setting the value as localhost will be resolved from InetAddress.getLocalHost().getHostAddress()
# Ex. seeds = ["10.1.1.123", "10.1.1.124", "10.1.1.125"]
```

- **seeds = ["localhost"]**

As you can see, the default value is **["localhost"]**, which is only suitable for a single node server where both TWCloud and Cassandra are deployed on the same machine. According to our sample environment, you should change it to the following.

application.conf

```
seeds = ["10.1.1.101", "10.1.1.102", "10.1.1.103"]
```

- **esi.persistence.cassandra.keyspace.replication-factor**

This parameter defines the Cassandra replication factor for the "**esi**" keyspace used by TWCloud. The replication factor describes how many copies of your data will be written by Cassandra. For example, replication factor 2 means your data will be written to 2 nodes.

For a three-node cluster, if you would like the cluster to be able to survive 1 node loss, you will need to set the replication factor to 3.

```
persistence {
         cassandra {
                keyspace
{
                       replication-factor = 3
                }
```

Please note that this configuration will be used only for the first time TWCloud connects to Cassandra and creates a new esi keyspace. Changing the replication factor after the keyspace has already been created is rather a complex task. Read this document if you need to change it.

authserver.properties changes

Cassandra configuration parameters

authserver.properties

```
cassandra.contactPoints=10.1.1.101,10.1.1.102,10.1.1.103
cassandra.keyspace.replication.factor=3
```

Please make sure that there are no spaces after the commas in the list of **cassandra.contactPoints**.

TWCloud uses QUORUM for both write and read consistency levels.

Click here for a detailed explanation of data consistency.

To start up the TWCloud cluster, start the server on the seed machine and wait until you see a message similar to the following in the **server.log**.

```
INFO  2017-02-15 10:57:08.409 TWCloud Cluster with 1 node(s) : [10.1.1.111] [com.nomagic.esi.server.core.
actor.ClusterHealthActor, twcloud-esi.actor.other-dispatcher-31]
```

Then you can start the server on the remaining machines. You should see the following messages in the **server.log** which shows all 3 nodes are forming the cluster.

```
INFO  2017-02-15 10:58:23.956 TWCloud Cluster with 2 node(s) : [10.1.1.111, 10.1.1.112] [com.nomagic.esi.
server.core.actor.ClusterHealthActor, twcloud-esi.actor.other-dispatcher-18]
INFO  2017-02-15 10:58:25.963 TWCloud Cluster with 3 node(s) : [10.1.1.111, 10.1.1.112, 10.1.1.113] [com.
nomagic.esi.server.core.actor .ClusterHealthActor, twcloud-esi.actor.other-dispatcher-18]
```

Related pages

- Installing and configuring Cassandra on Windows
- Installing and configuring Cassandra on Linux