要连接到集群，首先要告诉集群：你是谁，你有什么特征。在es中体现为实例化节点。

es通过org.elasticsearch.node.NodeBuilder的build()或者node()方法实例化节点，build()创建节点而不启动，而node()方法等价于build().start()，即创建并启动。

首先实例化NodeBuilder，有两种方式，第一种是new，如下：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = **new** NodeBuilder();

第二种是使用NodeBuilder的工厂方法，NodeBuilder中有以下代码：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. **public** **static** NodeBuilder nodeBuilder() {
2. **return** **new** NodeBuilder();
3. }

所以可以直接通过以下代码实例化NodeBuilder：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder();

实例化后便可设置该node的特性，NodeBuilder本身自带了一些方法：

1. loadConfigSettings，是否加载配置文件。

你可以在classpath下创建一个elasticsearch.yml文件，然后在其中设置一些属性（参见[分布式搜索Elasticsearch——配置](http://blog.csdn.net/geloin/article/details/8444972)），再把loadConfigSettings的值设置为true即可，代码如下：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().loadConfigSettings(**true**);

loadConfigSettings默认为true。

2. 是否只作为客户端，即不存储索引数据，默认值为false，代码如下所示：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().client(**true**);

3. 是否持有索引数据，默认值为true，代码如下所示：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().data(**true**);

client(boolean)和data(boolean)会有一定的关联，当client的值为true时，data的值默认为false，且不能改为true，否则会报错，即如下代码是不允许的：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().data(**true**).client(**true**);

当client的值为false时，data的值默认为true，而且改为false也不起作用，也即以下两行代码是等价的：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().data(**true**).client(**false**);
2. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().data(**false**).client(**false**);

4. 是否为本地节点，本地节点是指在JVM级别中的同级，当多个节点使用同一个JVM时，这些节点可以组合成一个集群，而非同一个JVM下的节点则不处于集群中，默认值为false：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().local(**true**);

5. 设置集群名，即该节点位于哪个集群下，默认值为elasticsearch：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. NodeBuilder nodeBuilder = NodeBuilder.nodeBuilder().clusterName("your name");

若多个节点设置了同一个clusterName，则说明这几个节点位于同一个集群下。

6. 以Map的方式设置属性，NodeBuilder中有以下两个方法：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. /\*\*
2. \* Explicit node settings to set.
3. \*/
4. **public** NodeBuilder settings(Settings.Builder settings) {
5. **return** settings(settings.build());
6. }
8. /\*\*
9. \* Explicit node settings to set.
10. \*/
11. **public** NodeBuilder settings(Settings settings) {
12. **this**.settings.put(settings);
13. **return** **this**;
14. }

这两个方法允许你通过Map的方法设置属性，如以下代码所示：

**[java]** [view plain](http://blog.csdn.net/geloin/article/details/8446895) [copy](http://blog.csdn.net/geloin/article/details/8446895)

1. Map<String, String> settingMap = **new** HashMap<String, String>();
2. settingMap.put("node.client", "false");
3. settingMap.put("node.data", "true");
4. settingMap.put("node.local", "true");
5. settingMap.put("cluster.name", "clasterName");
6. settingMap.put("node.name", "geloin");
8. Settings settings = ImmutableSettings.settingsBuilder().put(settingMap).build();
9. Node node = NodeBuilder.nodeBuilder().settings(settings).node();

org.elasticsearch.common.settings.Settings是一个接口，org.elasticsearch.common.settings.ImmutableSettings才是它的实现类。

上一篇文章[分布式搜索Elasticsearch——配置](http://blog.csdn.net/geloin/article/details/8444972)说了一些es的配置，你可以都设置在上述代码中的map中。

综上所述，我们对设置Node的属性有以下结论：

1. 使用配置文件方式，即在classpath下建立一个elasticsearch.yml文件，然后把[分布式搜索Elasticsearch——配置](http://blog.csdn.net/geloin/article/details/8444972)里面提到的配置加入到此文件中，再保持loadConfigSettings为true即可；

2. 使用Map方式，然后用NodeBuilder.nodeBuilder().settings(settings)；

3. 使用NodeBuilder自带了client、data、local、clusterName方法，不过NodeBuilder自带的方法有限，所以可能还需要配合以上提到的两种方式才能达到你的要求。  
至于如何选择，那就看你自己了。