

同一台机器上的一个 Jetty 实例做负载均衡器, 另外两个 Jetty 实例做为 slave(s), 还有另一台机器上的 Jetty 做 slave.

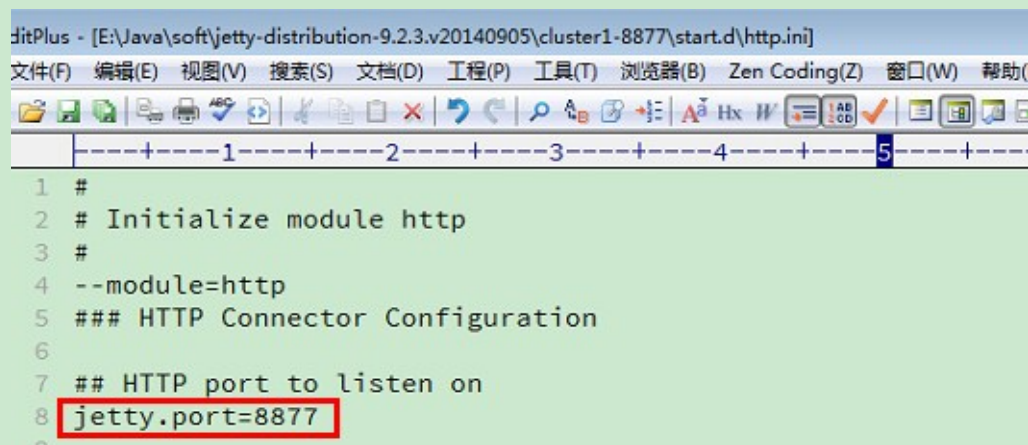
首先创建 **Balancer**, 创建一个 Java 应用程序即可, 最主要的就是设置以 **balancerMember.** 开头的初始参数. 它的作用就是将请求转发到其它的 Jetty 实例.

```
1.  public class BalancerServer {
2.      private boolean stickySessions;
3.      private Server balancer;
4.      private static final String CONTEXT_PATH = "/*";
5.      private static final String SERVLET_PATH = "";
6.
7.      public static void main(String[] args) throws Exception {
8.          BalancerServer server = new BalancerServer();
9.
10.         server.startBalancer();
11.     }
12.
13.     private void startBalancer() throws Exception {
14.         ServletHolder balancerServletHolder = new ServletHolder(BalancerServlet.class);
15.         //balancerServletHolder.setInitParameter("stickySessions", String.valueOf(stickySessions));
16.         //balancerServletHolder.setInitParameter("proxyPassReverse", "true");
17.         // 配置多个节点
18.         balancerServletHolder.setInitParameter("balancerMember." + "cluster1" + ".proxyTo",
19. "http://localhost:8877");
20.         balancerServletHolder.setInitParameter("balancerMember." + "cluster2" + ".proxyTo",
21. "http://localhost:8888");
22.         balancerServletHolder.setInitParameter("balancerMember." + "cluster3" + ".proxyTo",
23. "http://192.168.119.134:8899");
24.
25.         balancer = createServer(balancerServletHolder, 8080);
26.         balancer.start();
27.         balancer.join();
28.     }
29.
30.     private Server createServer(ServletHolder servletHolder, int port) {
31.         Server server = new Server(port);
32.         ServerConnector connector = new ServerConnector(server);
33.         server.addConnector(connector);
34.
35.         ServletContextHandler context = new ServletContextHandler(server, CONTEXT_PATH, ServletContextHandler.SESSIONS);
36.         context.addServlet(servletHolder, SERVLET_PATH + "/*");
37.
38.         return server;
39.     }
40. }
```

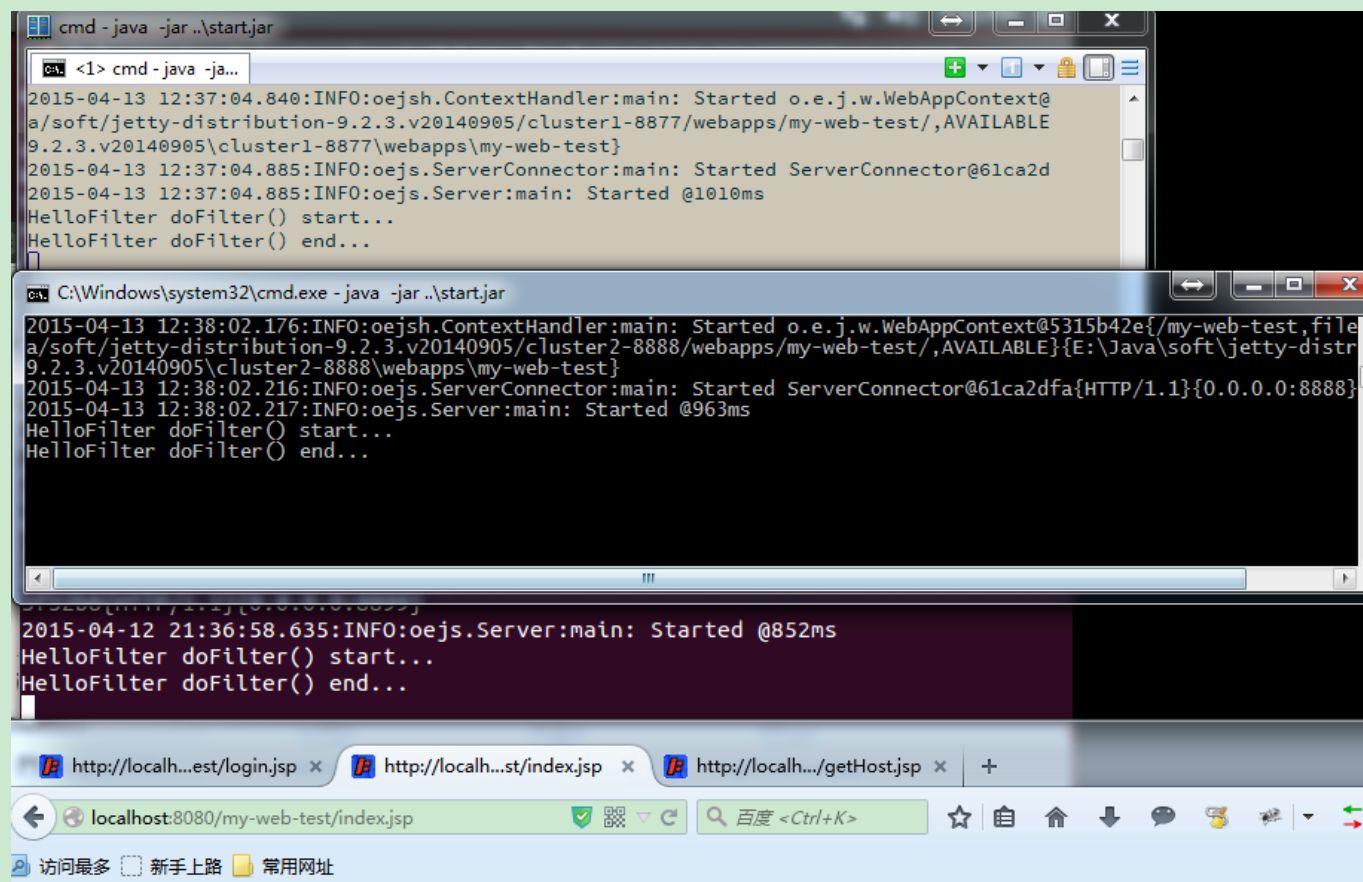
从上面的代码中可以看到我们配置了两个 **slave** 节点. 创建两个目录, 如下:



里面的内容一样.修改各自的 `http.ini` 文件中的端口配置即可.



另外还有一台 Ubuntu 机器上的配置与此类似.接着启动这 4 个 Jetty 实例.然后用浏览器发出请求.这里我们都是用 `http://localhost:8080`.可以看到请求都平均到了每个 Jetty 实例.



Hello World!

更具体的信息可以参考 `org.eclipse.jetty.proxy.BalancerServletTest` 或阅读 `org.eclipse.jetty.proxy.BalancerServlet` 类相关的源码.