

Hardware, Database, App, and Java Monitoring

Below are the instructions to deploy Monitors on each of them below.

- Hardware Monitoring
- Database Monitoring
- WepApp Monitoring
- Java Monitoring

Hardware Monitoring

A file called APCMonitor.zip will be provided. It needs to be unzipped to a folder, and required permissions need to be set to start/stop scripts (in the case of the Unix environment). Please check APC Monitors: How to monitor your app or other servers?

The command to start APC Monitor is StartMonitor.bat or StartMonitor.sh depending on the environment that is been worked on and needs a configuration file called apcmonitor.config.

The command to stop APC Monitor is StopMonitor.bat or StopMonitor.sh depending on the environment that is been worked.

An example apcmonitor.config file for Hardware monitoring looks like the below:

```
{
  server:{
    port:51111
  },
  configuration:{
    timedelays:1000
  },
  listeners:[
    {class:"com.appvance.apcmonitor.listeners.${os}.CPUListener",configuration:{}},
    {class:"com.appvance.apcmonitor.listeners.${os}.MEMListener",configuration:{}},
    {class:"com.appvance.apcmonitor.listeners.${os}.NETListener",configuration:{
      "net":"default"
    }}
  ]
}
```

```
}}  
}  
}
```

The listening port which connects AIQ with AIQ Monitor is defined by `server.port`. The example shown is 51111.

The `${os}` parses automatically to the correct operating system environment.

Database Monitoring

The database monitoring deployment is the same as Hardware monitoring. The change is the configuration file.

A single APC Monitor can store hardware information along with the Database information. APC Monitor does not need to be in the same box as the database.

In order to use APC Monitor to store the database information, it requires database credentials. The required fields are the user name, password, database URL, database type, default database (if applicable).

An example `acpmonitor.config` file for Database monitoring looks like the below:

```
{  
  "server":{  
    port:51111  
  },  
  "configuration":{  
    "timedelays":1000  
  },  
  "configurations":[  
    {  
      name: MySQLConfiguration  
      values:{  
        username: "XXX",  
        password: "XXX",  
        url:"XXX"  
      }  
    }  
  ]  
}
```

```
],
listeners:[
{
class:"com. appvance .apcmonitor.listeners.mysql.CPUUsageListener",
configuration:"MySQLConfiguration"
},
{
class:"com.appvance.apcmonitor.listeners.mysql.WorkloadListener",
configuration:"MySQLConfiguration"
},
{
class:"com.appvance.apcmonitor.listeners.mysql.IOListener",
configuration:"MySQLConfiguration"
}
}
```

WebApp Monitoring

In the case of the WebApps, provided APCMonitor.war needs to be deployed in the WebApplication server

Below is the standard configuration for WebApps:

```
{
configuration:{
timedelays:1000
},
listeners:[
{class:"com.appvance.apcmonitor.listeners.jvm.ThreadListener",configuration:{}},
{class:"com.appvance.apcmonitor.listeners.jvm.GCListener",configuration:{}},
{class:"com.appvance.apcmonitor.listeners.jvm.HeapListener",configuration:{}}
]
}
```

The file will be inside the war file. If the configuration needs to be changed the WebApps need to be redeployed.

Java Monitoring

In the case of the Java environment, the jar files need to be added to the classpath. The APC Monitor needs to be configured and started manually using the next method from `com.appvance.apcmonitor.APCMonitor`

```
public void configure(JSONObject configuration) throws Exception ;
```

The configuration file is the same as the WebApp monitoring.

Revision #1

Created 2026-07-31 14:12:54 UTC by Admin

Updated 2026-07-31 14:12:54 UTC by Admin