

Monitoring Scenarios

Appvance Monitor is an agent-based service that observes CPU, network, and memory utilization. Appvance Monitor logs observed values to the repository for analysis in the results charts. The Monitoring tab identifies the location of Appvance Monitor services running on the Test Nodes and Hosts. A Host is a server running your application server or database server.

- [Monitoring Scenarios](#)
- [Monitoring Your App On Other Servers](#)
- [Hardware, Database, App, and Java Monitoring](#)

Monitoring Scenarios

Appvance Monitor is an agent-based service that observes CPU, network, and memory utilization. Appvance Monitor logs observed values to the repository for analysis in the results charts. The Monitoring tab identifies the location of Appvance Monitor services running on the Test Nodes and Hosts. A Host is a server running your application server or database server.

Use the Monitoring Tab to identify the location of Glassbox service agents. Glassbox observes memory leaks, thread deadlocks, slow database queries, and other aspects of performance for Java Enterprise Applications. Read the Glassbox chapter for details.

APC Monitor is a java based agent which can record multiple resources on different environments (Hardware, Database, Webapps, and Java) according to the configuration. It listens to the resource usage and once the test is done, it sends the information to AIQ to correlate resources with transactions. To achieve that the AIQ Console needs to be able to communicate with APC Monitor.

Appvance Monitor works as a small footprint service that doesn't require a restart after a test is complete. It can also connect to multiple AIQ Tests at the same time.

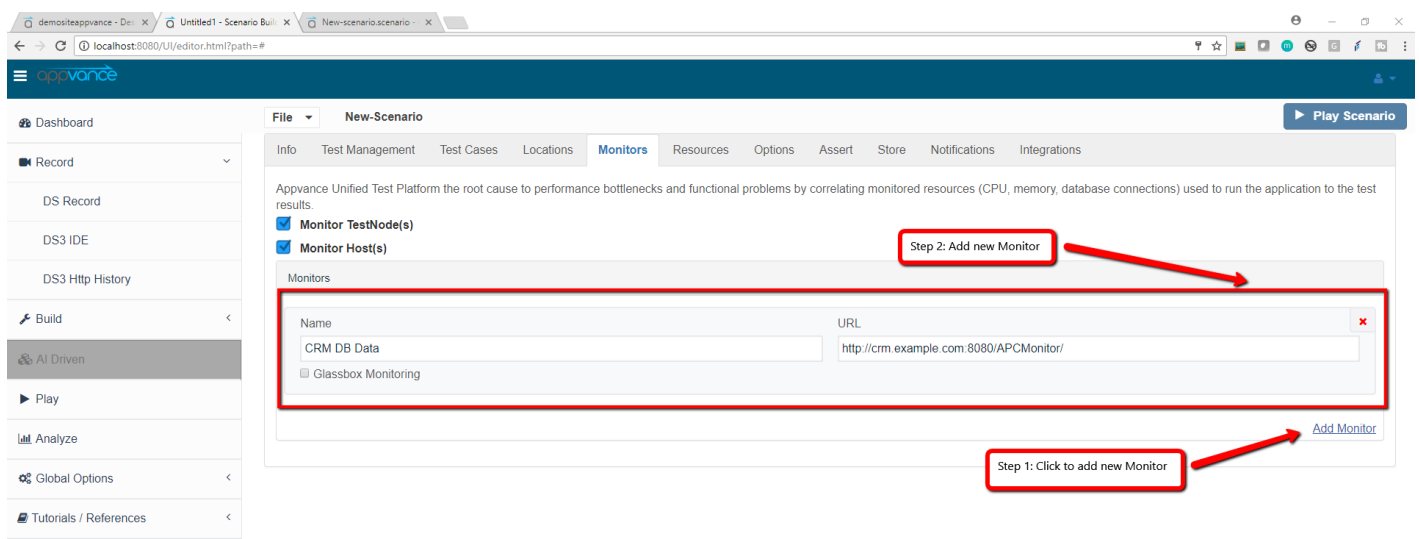
Monitoring Your App On Other Servers

APC Monitors: How to monitor your app or other servers?

APCMonitor can be deployed on 'Application Under Test (AUT)' in two ways:

1. APCMonitor is to run as a standalone service on the AUT machine (Most Popular)
2. Ensure the latest JRE is installed.
3. Download: wget https://s3.amazonaws.com/apc-monitor/APC_Monitor_New.zip
4. Unzip the contents to a folder
5. Give full read/write permissions to the folder: sudo chmod -R a+x APCMonitor_New (Linux)
6. cd APCMonitor_New
7. Run startup.bat (Windows) or sudo sh startup.sh (Linux)
8. Make sure the Monitor launch script startup.bat/.sh is run even if your servers are rebooted.
9. In the Scenario Editor, navigate to the Monitors section.
10. Click Add Host Monitors
11. Add the server name in the name text field and below URL as value:

`http://<AUT Machine IP Address>:8080/APCMonitor`



- APCMonitor is run as a web appliance inside an application server container, such as WebLogic, WebSphere, Tomcat, etc.
1. When deployed as an appliance, APCMonitor will provide system resource metrics such as CPU, RAM; In addition, JVM performance metrics like heap usage can also be gathered in real-time.
 1. <http://appvance.s3.amazonaws.com/APCMonitor/APCMonitor.war>
 2. WAR file of the APCMonitor can be downloaded from here:

NOTE: Please make sure there is no "/" at the end of the URL while saving the Scenario, else Monitor might not work.

For Ex: `http://<AUT Machine IP Address>:8080/APCMonitor` this will work however `http://<AUT Machine IP Address>:8080/APCMonitor/` wont work.

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"
}
}
```

WepApp 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.