Application Performance Management 2.0

Getting Started

Issue	01
Date	2023-07-20

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

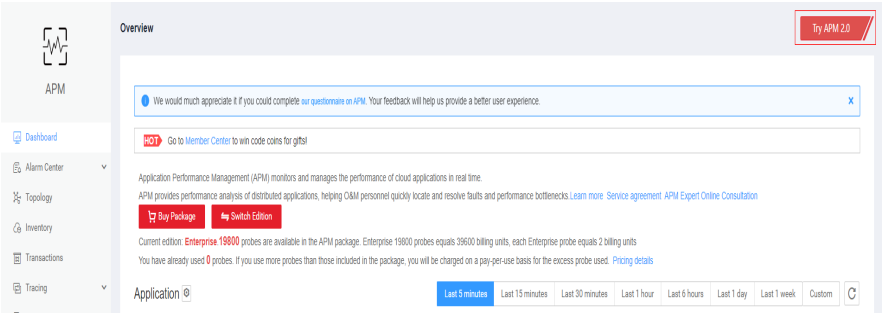
The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Enabling APM 2.0

1. Register a **Huawei Cloud account** and complete **real-name authentication**.
2. Enable APM 2.0
   a. Log in to the Huawei Cloud APM console.
   b. Click ☰ on the left and choose **Application** > **Application Performance Management**.
   c. Click **Try APM 2.0** in the upper right corner. The APM 2.0 page is displayed.

      Figure 1-1 Experiencing APM 2.0

   d. Click **Experience APM for free**.
3. Obtain the access key that is automatically created.

   APM 2.0 uses AK/SK for signature verification. Only authorized accounts can report data.

   a. Log in to the APM 2.0 console.
   b. In the navigation pane, choose **System Management** > **Access Keys**.
   c. On the **Access Keys** page, view the access key that has been automatically created.

2 Monitoring Java Applications

2.1 Connecting Agents

Prerequisite

The network between your host and APM is normal.

You can run the **Telnet** command to check the network. For example, to check the network connectivity in the **CN-Hong Kong** region, log in to the host where the application is deployed and run the **telnet 100.125.6.106:41333** command. For details about the access addresses of other regions, see **Access Addresses**.

Procedure

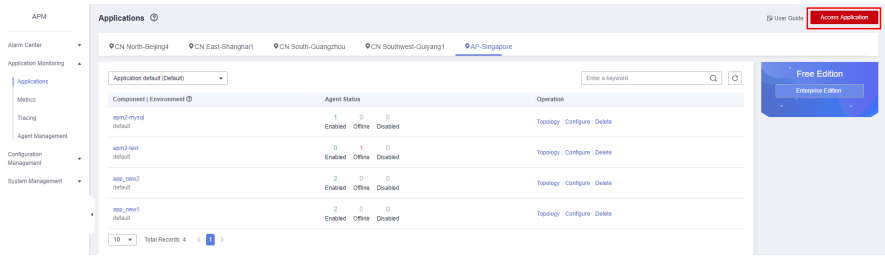
Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Application** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Applications**.

Step 4 On the displayed page, click **Access Application**.

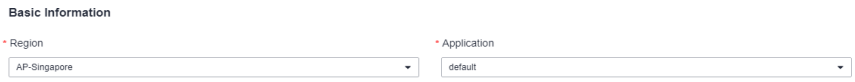
Figure 2-1 Connecting an application



Step 5 For **Code Source**, select **Enhanced probe**.

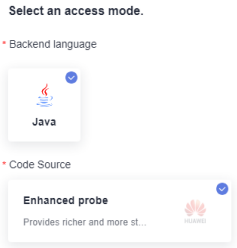
Step 6 Select a region and application.

Figure 2-2 Basic information



Step 7 For **Backend language**, select **Java**.

Figure 2-3 Access mode



Step 8 Select an access mode based on the application type, and access data by following the instructions.

Figure 2-4 Data access

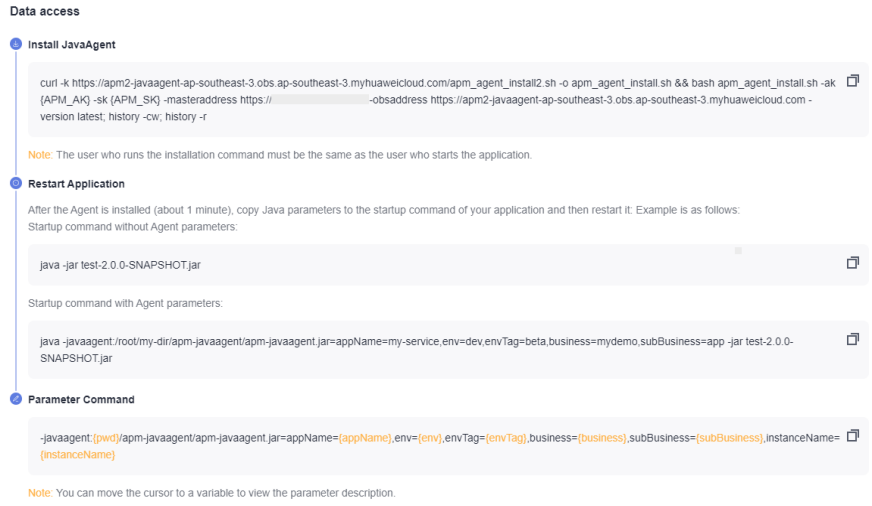


Table 2-1 Parameter description

Parameter	Description	Mandatory
pwd	Path where the **apm-javaagent.jar** package is located.	Yes
appName	Component name, which must start with a character. A component can contain multiple environments. The names of components under an application must be unique. If there are duplicate names, use **instanceName** to distinguish them.	Yes

Parameter	Description	Mandatory
env	Name of an environment where an application is deployed. A program can be deployed in different environments (such as the test or live network environment). Each environment is deployed in one region and has a unique region attribute. If this parameter is blank, the default environment will be used.	No
envTag	Environment tag for filtering environments. Different environments may have the same tag. This parameter can be left blank.	No
business	Name of an application that already exists (a global concept). If this parameter is left blank, the automatically created application will be used.	No
subBusiness	Name of a sub-application (a global concept), which is similar to a folder. If it is left blank, resources will be mounted to the root application. There can be up to three layers of sub-applications. For example, for **a/b/c**, **a**, **b**, and **c** respectively represents a layer.	No
instanceName	Name of an instance, which is left blank by default. If an application has multiple instances deployed on a host, use this parameter to distinguish them. Generally, Java instances deployed on a host belong to different applications. An application rarely has identical instances.	No

----End

Dynamically Configuring master.address and AK/SK

APM supports dynamic configuration of the master.address and AK/SK.

You can set the **APM_MASTER_ADDRESS**, **APM_ACCESS_KEY** (**apm-ak**), and **APM_SECRET_KEY** (**apm-sk**) environment variables. For details about how to obtain the AK/SK, see **Manually Installing Agents for Java Applications**. For details about how to obtain the master.address, see **Access Address (master.address)**.

NOTE

- If you set the AK/SK and master.address in both the JavaAgent configuration file and environment variables, your environment variable settings will take precedence.
- Agents later than 2.3.19 support dynamic configuration of master.address and AK/SK.

2.2 Manually Installing Agents for Java Applications

Prerequisites

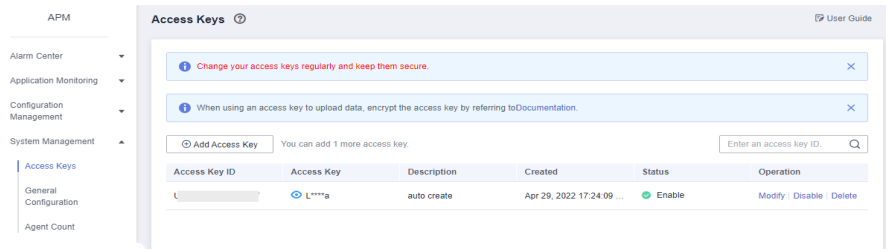
- The network between your host and APM is normal.

  You can run the **Telnet** command to check the network.

  For example, to check the network connectivity in the **CN-Hong Kong** region, log in to the host where the application is deployed and run the **telnet 100.125.6.106:41333** command. For details about the access addresses of other regions, see **Access Addresses**.

- The AK/SK required for accessing JavaAgents have been obtained. To obtain them, log in to the APM console and choose **System Management** > **Access Keys** in the navigation pane.

  Figure 2-5 Obtaining an AK/SK



Procedure

Step 1 Download **apm-javaagent-x.x.x.zip** to any directory of your host. For the download address, see **JavaAgent Download Addresses**.

Example command:

curl -O https://xxx/apm-javaagent-x.x.x.tar

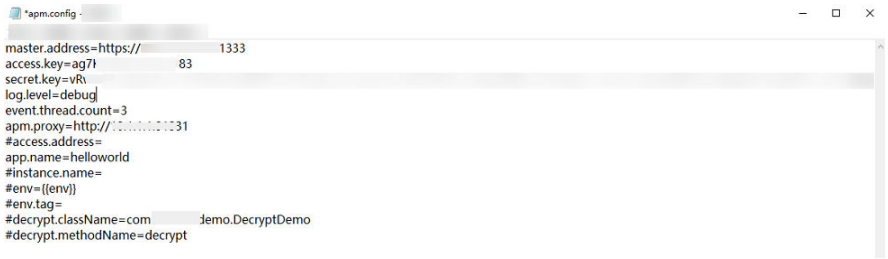
Step 2 Run the **tar** command to decompress the JavaAgent package.

Example command:

tar -xvf apm-javaagent-x.x.x.tar

Step 3 Modify the **apm.config** file in the JavaAgent package. Configure **master.address** by referring to **Access Address (master.address)**, and add the AK/SK to the configuration file, as shown in the following figure.

Figure 2-6 Adding the AK/SK



Step 4 Modify the startup script of the Java process.

Add the path of the **apm-javaagent.jar** package and the component name of the Java process to the end of the Java command in the service startup script.

Example of adding **-javaagent** parameters:

java **-javaagent:/xxx/apm-javaagent/apm-javaagent.jar=appName={appName}**

If your enterprise has a large number of services, you can add more complex configurations. For example:

java **-javaagent:/xxx/apm-javaagent/apm-javaagent.jar=appName=myApp,env=myEnv,envTag=myTag,business=myBusiness,subBusiness=mySub**

NOTE

- The preceding parameters are built-in CMDB information of APM. For details, see **CMDB Management**.
- Due to historical reasons, the metadata of APM startup parameters conflicts with some CMDB concepts. The following shows the details.

  Generally, the startup parameter is set to **-javaagent:D:\javaagent-package\apm-javaagent\apm-javaagent.jar=appName=xxx,env=yyy,business=zzz,subBusiness=sss,envTag=xxx**. **appName** indicates a component, **business** indicates an application, **subBusiness** indicates a sub-application, and **envTag** indicates an environment tag.

  If **business** is not set on the web page, the system reports an error when the JavaAgent is started. If other parameters (**subBusiness**, **appName**, **env**, and **envTag**) are not set, the system automatically creates them when the JavaAgent is started.

  Component names are unique under an application.

Step 5 Redeploy the application.

----End

2.3 Installing Agents for the Java Applications Deployed in CCE Containers

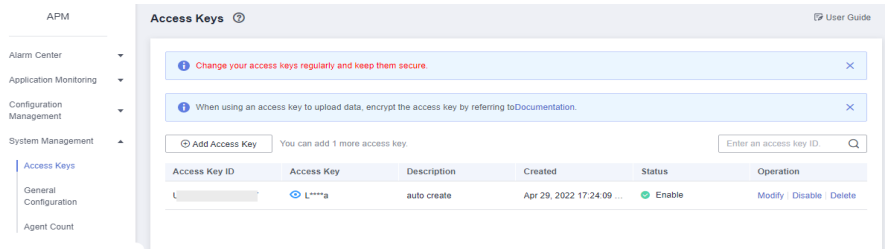
Prerequisites

- The network between your host and APM is normal.

  You can run the **Telnet** command to check the network.

  For example, to check the network connectivity in the **CN-Hong Kong** region, log in to the host where the application is deployed and run the **telnet 100.125.6.106:41333** command. For details about the access addresses of other regions, see **Access Addresses**.
- For details, see **Regions and Endpoints**.
- The AK/SK required for accessing JavaAgents have been obtained. To obtain them, log in to the APM console and choose **System Management** > **Access Keys** in the navigation pane.

Figure 2-7 Obtaining an AK/SK



Usage Instruction

APM only supports Java applications deployed on CCE. **Table 2-2** describes the parameters.

Table 2-2 Parameters for configuring performance management

Name	Description
Probe	Select a target probe. Options: **Disable**, **APM 1.0**, and **APM 2.0**.
Probe Version	Version of the probe.
Probe Upgrade Policy	Policy for the probe upgrade. The default value is **Auto upgrade upon restart**. • **Automatic upgrade upon restart**: The system downloads the probe image each time the pod is restarted. • **Manual upgrade**: If a local image is available, it will be used. If no local image is available, the system downloads the probe image.
APM Environment	Enter an APM environment name. This parameter is optional.
APM App	Select an existing APM application.
Sub-app	Enter an APM sub-application. This parameter is optional.
Access Key	The system automatically obtains the APM key. For details, see **Prerequisites**.

Step 1 Log in to the CCE console. In the navigation pane, choose **Workloads** > **Deployments** or **StatefulSets**, and click **Create Deployment** or **Create StatefulSet**.

Step 2 In the **APM Settings** area on the **Configure Advanced Settings** page, select **Java probe**. The APM service will be enabled and a probe will be installed on the node.

NOTE

Currently, both APM 1.0 and APM 2.0 probes are supported. You are advised to select the APM 2.0 probe.

Probes provide traces, topologies, SQL analysis, and stack tracing for Java workloads. A small number of resources will be consumed when you run probes.

Step 3 Set probe-related parameters.

- **Monitoring Group**: Enter a monitoring group name, for example, **testapp**. Select a group from the drop-down list if there are any.
- **Probe Version**: Select a probe version.
- **Probe Upgrade Policy**: By default, **Automatic upgrade upon restart** is selected.
  - **Automatic upgrade upon restart**: The system downloads the probe image each time the pod is restarted.
  - **Manual upgrade**: If a local image is available, it will be used. If no local image is available, the system downloads the probe image.

Step 4 After the application is started, wait for about 3 minutes. Then, the application data is displayed on the APM console. You can log in to the APM console and optimize the application performance through topology and tracing.

----End

For how to install Agents for Java applications deployed on the new CCE console, see **CCE User Guide**.

2.4 Installing Agents on Applications Deployed Using CodeArts Deploy

Prerequisite

The network between your host and APM is normal.

You can run the **Telnet** command to check the network. For example, to check the network connectivity in the **CN-Hong Kong** region, log in to the host where the application is deployed and run the **telnet 100.125.6.106:41333** command. For details about the access addresses of other regions, see **Access Addresses**.

Procedure

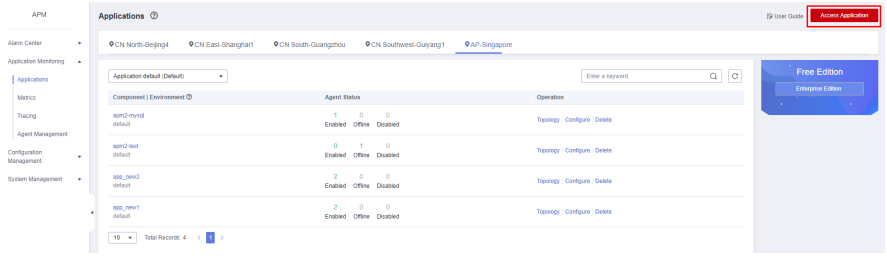
Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Application** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Applications**.

Step 4 On the displayed page, click **Access Application**.

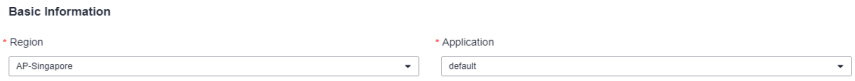
Figure 2-8 Connecting an application



Step 5 For **Code Source**, select **Enhanced probe**.

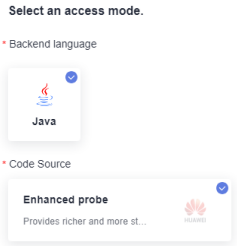
Step 6 Select a region and application.

Figure 2-9 Basic information



Step 7 For **Backend language**, select **Java**.

Figure 2-10 Access mode



Step 8 Select an access mode based on the application type, and access data by following the instructions.

Figure 2-11 Data access

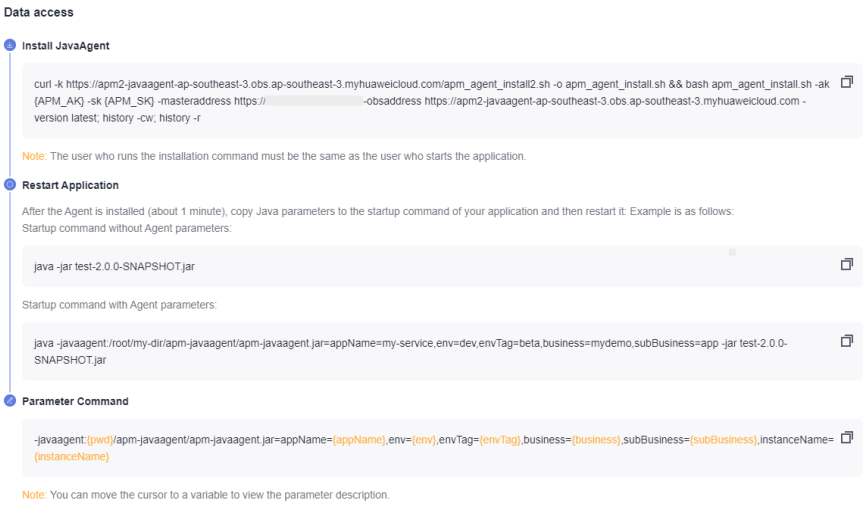


Table 2-3 Parameter description

Parameter	Description	Mandatory
pwd	Path where the **apm-javaagent.jar** package is located.	Yes
appName	Component name, which must start with a character. A component can contain multiple environments. The names of components under an application must be unique. If there are duplicate names, use **instanceName** to distinguish them.	Yes
env	Name of an environment where an application is deployed. A program can be deployed in different environments (such as the test or live network environment). Each environment is deployed in one region and has a unique region attribute. If this parameter is blank, the default environment will be used.	No
envTag	Environment tag for filtering environments. Different environments may have the same tag. This parameter can be left blank.	No
business	Name of an application that already exists (a global concept). If this parameter is left blank, the automatically created application will be used.	No
subBusiness	Name of a sub-application (a global concept), which is similar to a folder. If it is left blank, resources will be mounted to the root application. There can be up to three layers of sub-applications. For example, for **a/b/c**, **a**, **b**, and **c** respectively represents a layer.	No
instanceName	Name of an instance, which is left blank by default. If an application has multiple instances deployed on a host, use this parameter to distinguish them. Generally, Java instances deployed on a host belong to different applications. An application rarely has identical instances.	No

Step 9 Access the CodeArts Deploy deployment task, edit the deployment action, add the step of running the **shell** command, and then add the **copied command**.

Step 10 Modify the deployment procedure, copy the startup parameter, and add the parameter to the Java command of the service startup script.

Step 11 Redeploy the application.

----End

2.5 Installing Agents on Docker Applications

Prerequisite

The network between your host and APM is normal.

You can run the **Telnet** command to check the network. For example, to check the network connectivity in the **CN-Hong Kong** region, log in to the host where the application is deployed and run the **telnet 100.125.6.106:41333** command. For details about the access addresses of other regions, see **Access Addresses**.

Procedure

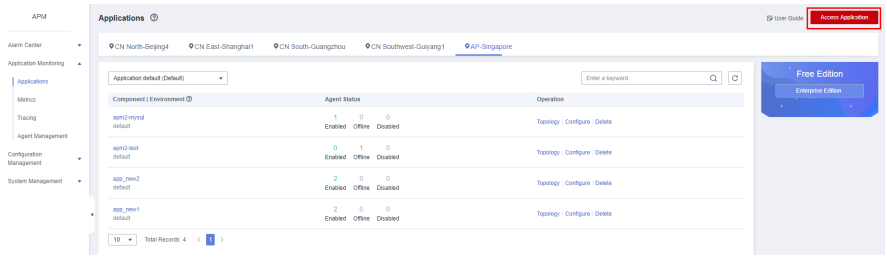
Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Application** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Applications**.

Step 4 On the displayed page, click **Access Application**.

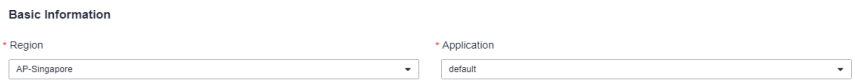
Figure 2-12 Connecting an application



Step 5 For **Code Source**, select **Enhanced probe**.

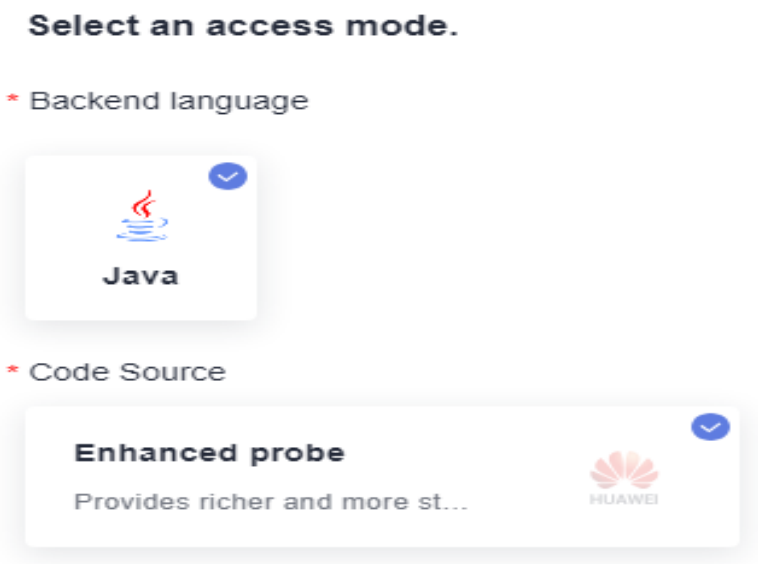
Step 6 Select a region and application.

Figure 2-13 Basic information



Step 7 For **Backend language**, select **Java**.

Figure 2-14 Access mode



Step 8 Select an access mode based on the application type and access data by following the instructions.

Figure 2-15 Data access

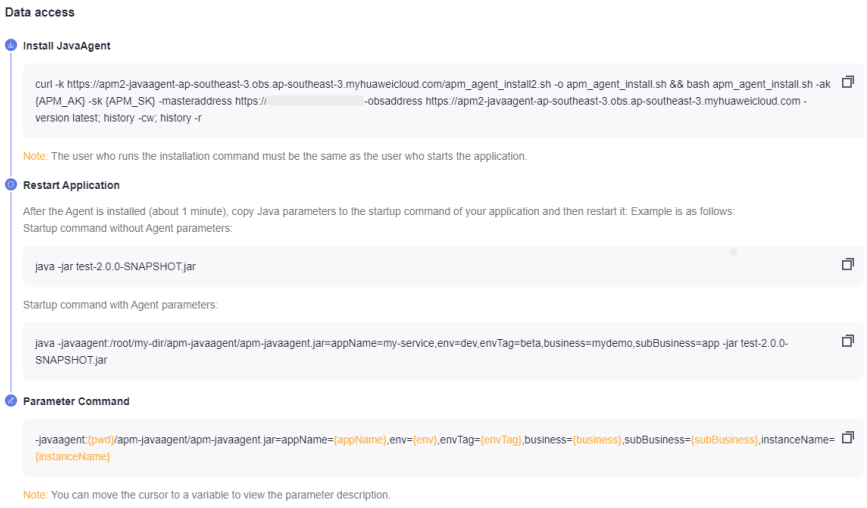


Table 2-4 Parameters

Parameter	Description	Mandatory
pwd	Path where the **apm-javaagent.jar** package is located.	Yes
appName	Component name, which must start with a character. A component can contain multiple environments. The names of components under an application must be unique. If there are duplicate names, use **instanceName** to distinguish them.	Yes

Parameter	Description	Mandatory
env	Name of an environment where an application is deployed. A program can be deployed in different environments (such as the test or live network environment). Each environment is deployed in one region and has a unique region attribute. If this parameter is blank, the default environment will be used.	No
envTag	Environment tag for filtering environments. Different environments may have the same tag. This parameter can be left blank.	No
business	Name of an application that already exists (a global concept). If this parameter is left blank, the automatically created application will be used.	No
subBusiness	Name of a sub-application (a global concept), which is similar to a folder. If it is left blank, resources will be mounted to the root application. There can be up to three layers of sub-applications. For example, for **a/b/c**, **a**, **b**, and **c** respectively represents a layer.	No
instanceName	Name of an instance, which is left blank by default. If an application has multiple instances deployed on a host, use this parameter to distinguish them. Generally, Java instances deployed on a host belong to different applications. An application rarely has identical instances.	No

Step 9 Add the copied command to the Dockerfile file and add the **JAVA_TOOL_OPTIONS** environment variable.

```
RUN curl -k https://javaagent.obs.xx-xxx-x.ulanqab.huawei.com/apm_agent_install2.sh -o apm_agent_install.sh && bash apm_agent_install.sh -ak {AK}-sk {SK} -masteraddress https://xxx.xx.xx.xx:xxx -obsaddress https://javaagent.obs.xx-xxx-x.ulanqab.huawei.com -version latest

ENV JAVA_TOOL_OPTIONS=-javaagent:${PROJECT_DIR}/apm-javaagent/apm-javaagent.jar=appName=${APP_NAME}
```

Step 10 Rebuild an image. The following is an example of the Dockerfile file.

```
FROM openjdk
RUN mkdir /opt/cloud
ENV PROJECT_DIR=/opt/cloud
ENV APP_NAME=hello
WORKDIR $PROJECT_DIR
ADD ${APP_NAME}.jar ${APP_NAME}.jar
RUN curl -k https://javaagent.obs.xx-xxx-x.ulanqab.huawei.com/apm_agent_install2.sh -o apm_agent_install.sh && bash apm_agent_install.sh -ak {AK}-sk {SK} -masteraddress https://xxx.xx.xx.xx:xxx -obsaddress https://javaagent.obs.xx-xxx-x.ulanqab.huawei.com -version latest
ENV JAVA_TOOL_OPTIONS=-javaagent:${PROJECT_DIR}/apm-javaagent/apm-javaagent.jar=appName=${APP_NAME}
CMD ["bash","-c","java -jar ${APP_NAME}.jar" ]
```

Step 11 Redeploy the application.

----End

3 Getting Started with Common Practices

After learning about how to use APM to monitor applications, implement common practices as required.

This section describes common practices of APM, helping you better use it.

Table 3-1 Common practices

Practice	Description
Locating the Causes of Request Errors	• Routine inspection of application metrics such as latency, throughput, and number of errors • Quick locating of error calls

4 Change History

Table 4-1 Change history

Released On	Description
2023-07-20	This issue is the first official release.

A JavaAgent Download Addresses

Region	Latest Version	Earlier Version			
CN-Hong Kong	2.4.5 sha256:37b155626f46f3a8b19772e3fd6597ece92bb255fdffa21e3e9e1467cc009392	2.4.4 sha256:f60b55d646fe592c427143f1c8d8e8f277a647a3dd73dcdca43d7d37655ec573	2.4.3 sha256:73de7b49e148b102b74a6075c3d8b25e76f43d6144ec5aafe6ac190ab3c1ef0b	-	-	-
AP-Singapore	2.4.5 sha256:37b155626f46f3a8b19772e3fd6597ece92bb255fdffa21e3e9e1467cc009392	2.4.4 sha256:f60b55d646fe592c427143f1c8d8e8f277a647a3dd73dcdca43d7d37655ec573	2.4.3 sha256:73de7b49e148b102b74a6075c3d8b25e76f43d6144ec5aafe6ac190ab3c1ef0b	2.4.1 sha256:c31cd55ead0b2172eb694fe402242b09caeb67d7059aaf40602586 33d9ade9f8	2.3.19 sha256:8755abcd541797ebf900f8f1767d62678b3cd57a6d8e045dcfb2f8c24885bfce	2.3.17 sha256:8893e89cd3174879232704828027fd636471525d6e5772ff77e3313f79a4b6b5

Region	Latest Version	Earlier Version				
LA-Sao Paulo1	2.4.3 sha256:73de7b49e148b102b74a6075c3d8b25e76f43d6144ec5aafe6ac190ab3c1ef0b	-	-	-	-	-

B Access Address (master.address)

Region	Access Address (master.address)
LA-Sao Paulo1	https://100.125.11.27:41333
CN-Hong Kong	https://100.125.6.106:41333
AP-Singapore	https://100.125.4.25:41333

