

**CodeArts PerfTest**

# Getting Started

**Issue**            01  
**Date**             2024-10-21



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI 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

---

<b>1 Introduction.....</b>	<b>1</b>
<b>2 Interactive Walkthroughs.....</b>	<b>3</b>
<b>3 Preparing Environment Resources.....</b>	<b>4</b>
<b>4 Creating a Test Project.....</b>	<b>7</b>
<b>5 Creating a Test Case.....</b>	<b>8</b>
<b>6 Creating a Test Task.....</b>	<b>10</b>
<b>7 Viewing a Test Report.....</b>	<b>12</b>
<b>8 Getting Started with Common Practices.....</b>	<b>13</b>

# 1 Introduction

CodeArts PerfTest provides performance test services for cloud applications that are built based on HTTP, HTTPS, TCP, UDP, HLS, RTMP, WebSocket, MQTT, or HTTP-FLV. CodeArts PerfTest quickly simulates service peaks with a large number of concurrent users. It allows you to define the contents and time sequences of packets and supports complex combinations of multiple transactions. After tests are complete, CodeArts PerfTest provides professional test reports to evaluate your service quality.

You can complete a performance test in four steps listed as in the following figure.

**Figure 1-1** Performance test procedure



You can quickly get familiar with the CodeArts PerfTest usage through the [interactive walkthroughs](#).

## Basic Concepts

- **Test Project:** Test projects are classified into PerfTest projects and JMeter projects. PerfTest projects provide project management capabilities, allowing you to share and reuse the contents of transactions, pressure test tasks, and test reports within a test project, and create different test projects for different test programs. JMeter projects are used to import JMeter scripts to CodeArts PerfTest.
- **Transaction:** A transaction is a user-defined operation model that consists of HTTP/HTTPS/TCP/UDP/WebSocket packet, think time, response extraction, checkpoint, and HLS/RTMP/HTTP-FLV/MQTT packet.
- **Packet:** Packets are data blocks transmitted between applications such as HTTP. These data blocks start with text metadata that describes the packet content and meaning. The text metadata is followed by optional data. Packets are transmitted among clients, servers, and agents.
- **Think Time:** To better simulate user behavior, insert a waiting time between different operations. For example, when a user receives data from the server, the user may wait several seconds before viewing data and providing responses. This period of time is called think time.

- **Response Extraction:** If a transaction contains multiple packets, the output of the previous packet, which is extracted by a regular expression or JSON, is used as the input of the next packet.
- **Checkpoint:** Checkpoints are where you define the verification information to determine whether the contents returned by the server are correct.
- **Number of Concurrent Users:** It refers to the number of users performing operations on a system at the same time. In CodeArts PerfTest, it is the number of virtual users you define when you configure a test phase.
- **Response Time:** Response time indicates the duration from the time when a client sends a request to the time when the client receives a response from the server.

# 2 Interactive Walkthroughs

---

## Scenarios

You can quickly get familiar with the CodeArts PerfTest usage through the interactive walkthroughs.

The interactive walkthroughs provide the following four experience wizards:

- Using the testing service: helps you get familiar with basic CodeArts PerfTest operations, including deploying a test project, adding a test task, and generating a test report.
- Variable debugging: guides you how to add global and local variables, quickly define a pressure-testing model, and check whether the configuration is correct when debugging.
- E-commerce solution: supports full-link pressure testing in scenarios with large numbers of concurrent users and multiple transactions. This wizard also guides you how to test E-commerce websites and solve problems such as application breakdown and capacity expansion.
- All-in-one system template: a pressure test template built for all-in-one systems, featuring ease of use, scenario simulation, and various pressure configurations.

## Procedure

**Step 1** Log in to the CodeArts PerfTest console.

**Step 2** In the navigation pane, choose **Interactive Walkthroughs**.

**Step 3** Choose a wizard and click **Try Now**.

Complete the experience as prompted.

----End

# 3 Preparing Environment Resources

---

## Test Resource Groups and Their Constraints

- Test resource groups are classified into shared resource groups and private resource groups. Shared resource groups are provided by the system by default, and private resource groups need to be created.
- Execution nodes of the shared resource group have been bound with an elastic IP address (EIP). When the tested application has network access restrictions, use a private resource group.
- A shared resource group supports a maximum of 1,000 concurrent users and 100 Mbit/s bandwidth. If higher concurrency or bandwidth is required, use a private resource group.
- JMeter test tasks can use only private resource groups.

## Creating a Private Resource Group

### NOTE

- When creating test resources for the first time, you need to be authorized by your account. An agency is automatically created so that CodeArts PerfTest can perform operations on Cloud Container Engine (CCE).
- To use CodeArts PerfTest, you must have the **CodeArts PerfTest Administrator** or **CodeArts PerfTest Developer** permission (you can only view the projects you created).
- To manage private resource groups, you must have the **CodeArts PerfTest Administrator** permission or both **CodeArts PerfTest Developer** and **CodeArts PerfTest Resource Administrator** permissions.
- To use private resource groups, you must have the **CodeArts PerfTest Administrator** permission or both **CodeArts PerfTest Developer** and **CodeArts PerfTest Resource Developer** permissions.
- For details about the permissions required for using CodeArts PerfTest and their application scenarios, see [What Are the Permissions Required for Using CodeArts PerfTest?](#).

**Step 1** Log in to the CodeArts PerfTest console, choose **Resource Groups** in the navigation pane, and click **Create Private Resource Group**.

**Step 2** (Optional) If this is the first time you create a private resource group, grant CodeArts PerfTest permissions necessary for creating private resource groups.

**Step 3** If you do not have a CCE cluster, [create a cluster](#) and then [create a resource group](#). If a CCE cluster is available, [create a resource group](#).

**Step 4** Create a cluster.

Click **Create Cluster**. The page for buying CCE clusters is displayed. For details about how to create a cluster, see [Buying a CCE Cluster](#).

 **NOTE**

- You are advised to use an independent CCE cluster for pressure tests to avoid configuration conflicts with CCE clusters in test or production environments.  
When selecting add-ons, retain the default settings for the test executor. For example, deselect unnecessary add-ons, such as NodeLocal DNSCache and Cloud Native Cluster Monitoring, to prevent the add-ons from occupying executor resources.
- The cluster management scale is related to the number of execution nodes. Create nodes of the corresponding specifications based on the number of concurrent users for pressure testing. For example, if 20 execution nodes are required, you can set the cluster scale to 50 nodes.
- You are advised to select **Tunnel network** as the network model. **Container CIDR Block** and **Service CIDR Block** must be the same as those of the tested object.
- CentOS is prone to IPVS- or Contrack-related stability issues on a heavy-load network. Therefore, do not use CentOS as the OS of cluster nodes when selecting IPVS. When both **IPVS** and **CentOS** are selected, network connection reuse may time out.

Click **Next: Select Add-on**. On the page displayed, select the add-ons to be installed during cluster creation.

Click **Next: Add-on Configuration**. Retain the default settings.

Click **Next: Confirm configuration**. After confirming that the cluster configuration is correct, select **I have read and understood the preceding instructions** and click **Submit**. It takes about 6 to 10 minutes to create a cluster.

After the cluster is created, return to the cluster management page and click **Create Node**. For details about how to create a node, see [Creating a Node](#).

 **NOTE**

- A node must have at least four vCPUs and 8 GB memory.
- Select EulerOS as the operating system.
- At least two nodes (one debugging node and one execution node) are required. The number of nodes depends on the specifications of the pressure test object. For example, 21 execution nodes (one debugging node and 20 execution nodes) are required for 100,000 concurrent users and 4 vCPUs | 8 GB memory.
- If the CCE cluster node and the application to be tested are not in the same VPC network, bind an EIP to the CCE cluster node. You can use existing EIPs. If no EIP is available, click **Auto create** to create one. When EIPs are automatically created, you are advised to pay by traffic and set the bandwidth to a large value to avoid affecting the pressure test. Additionally, an EIP with specified configurations is automatically assigned to each node. If the number of EIPs is less than the number of nodes, the EIPs are randomly bound to the nodes.
- In the **(Optional) Advanced Settings** area, set **Kubernetes Node Name** to **Node private IP**. If you select **Cloud server name**, the node cannot be managed.

Click **Next: Confirm**. After confirming that the node configuration is correct, select **I have read and understood the preceding instructions** and click **Submit**. After the node is created, return to the CodeArts PerfTest console.

**Step 5** Create a resource group.



Choose **Resource Groups** in the navigation pane, and click **Create Private Resource Group**.

Set the parameters listed in [Table 3-1](#).

**Table 3-1** Creating a private resource group

Parameter	Description
Resource Group Name	Name of the private resource group, for example, <b>Web-test-demo</b>
Node Cluster	Select a CCE cluster from the drop-down list.
Debugging Node	Node used for debugging. The debugging node cannot be changed after the resource group is created.
Execution Node	Target machine that a pressure test will be performed on and can provide performance data during testing

Click **Create**.

----End

# 4 Creating a Test Project

---

## Procedure

- Step 1** Log in to the CodeArts PerfTest console, choose **PerfTest Projects** in the navigation pane, and click **Create Test Project**.
- Step 2** In the displayed dialog box, enter a test project name (for example, **Web-test**) and description, and click **OK**.

----End

# 5 Creating a Test Case

---

## Procedure

**Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the navigation pane.

**Step 2** Locate the row that contains the target PerfTest project, and click the test project name, for example, **Web-test**. The test project details page is displayed. On the **Cases** tab, you can view the default directory and sample cases that are automatically generated.

**Step 3** In the default directory, choose **Sample Case**. Click **Case Script** and select **Pre-Steps** or **Steps**.

**Pre-Steps** are executed only once in each execution thread. This configuration is optional.

**Steps** are executor of a test case, which needs to be configured.

**Step 4** In the case, click **Add Request**. Enter a request name and add requests.

- Packet

(Mandatory) Packets are data blocks sent between HTTP applications.

Click the **Packet** tab, and set packet parameters based on the actual service to be tested.

- Think time

(Optional) To better simulate user behavior, it is necessary to simulate the waiting time between different operations. For example, when you receive data from a server, you may wait for several seconds to view the data before responding. This latency is called **Think Time**.

Set **Think Time** based on requirements. For more effective testing, do not set a long think time. You are advised to test the server for the worst case scenario.

Enable **Think Time** and set **Duration** to **1000** ms.

- Response extraction

(Optional) If multiple packets exist in the same case, use regular expressions or JSON to extract the output of the previous packet for the input of the next packet.

Enable **Response Extraction** and set parameters.

- **Checkpoint**  
(Optional) When enabled, you can check whether the content returned by a server is correct through customized verification information.

Enable **Checkpoint** and set parameters.

**Step 5** Select **Pressure Stage**.

You can select different pressure modes and add multiple phases. Each phase can simulate different numbers of concurrent users.

**Step 6** Click **Save**.

**Step 7** Click **Debug** in the upper right corner of the page. Select the target test resource group as the executor and click **Start**.

**Step 8** Click **View log** to view the test case debugging details.

If an error was reported in the debugging result, edit the case based on the log information and debug it again.

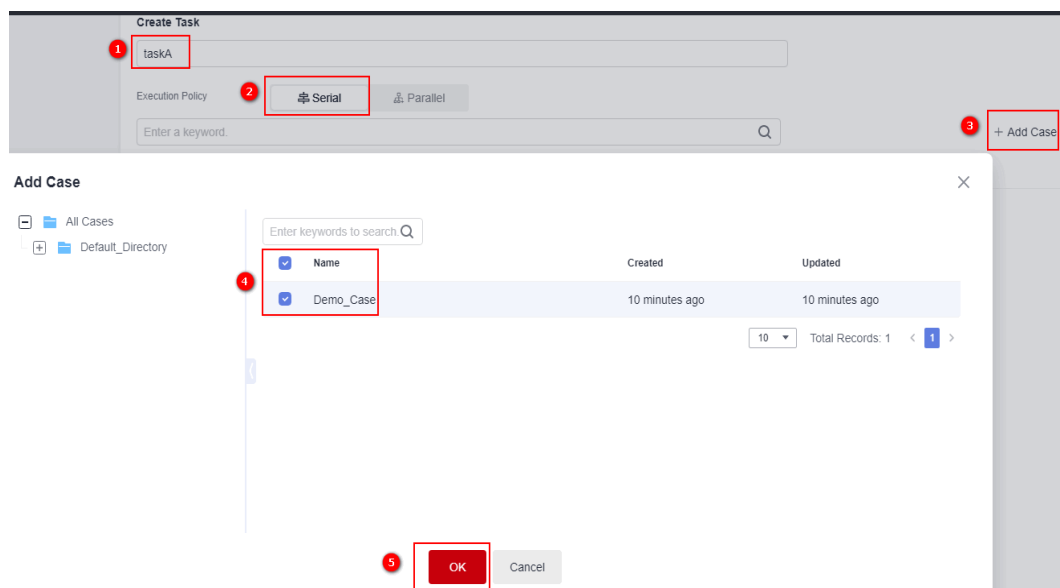
**Step 9** On the **Debug log** tab page, you can view the debugging history.

----End

# 6 Creating a Test Task

## Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the navigation pane.
- Step 2** Locate the row that contains the target PerfTest project, and click the test project name, for example, **Web-test**. The test project details page is displayed.
- Step 3** On the **Test Tasks** tab, click **Create Task** on the upper right of the page.
- Step 4** Enter the task name, for example, **taskA**, and select an execution mode.  
There are two execution policies:
  - **Serial**: Test cases in the test task are executed in sequence.
  - **Parallel**: Test cases in the test task are executed concurrently.
- Step 5** Click **Add Case**. In the displayed dialog box, select the created cases and click **OK**.




**Step 6** Click **Save**.


----End

# 7 Viewing a Test Report

---

## Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the navigation pane.
- Step 2** Locate the row that contains the target PerfTest project, and click the test project name, for example, **Web-test**. The test project details page is displayed.
- Step 3** On the **Test Tasks** tab, select a test task, for example, **taskA**, and click  in the **Operation** column.
- Step 4** Select the enterprise project and resource group type, and click **Start** to start the test task.
- Step 5** Click **View Report** to go to the real-time report page.

You can also click  in the **Operation** column of **taskA** after the pressure test task is complete to view the offline report.

On the **Overview** page, click **Download Report** in the upper right corner to obtain a PDF file. Contact experts to analyze the performance bottleneck of the current system and provide improvement suggestions.

----End

# 8 Getting Started with Common Practices

After preparing environment resources, creating test projects, creating test cases, and creating test tasks, you can use commonly used CodeArts PerfTest practices based on service requirements.

**Table 8-1** Common practices

Practice	Description
<b>Performance Tests of All-in-One System for Government Services</b>	Systems related to national economy and people's livelihood, such as all-in-one systems, respond slowly due to sudden increase of access traffic. To prevent system breakdown during peak hours, CodeArts PerfTest simulates actual scenarios and quickly constructs pressure models to detect service performance bottlenecks in different pressure models.
<b>Native Performance Pressure Test of JMeter Test Projects</b>	CodeArts PerfTest supports the native JMeter engine. You can import JMeter scripts into JMeter projects in CodeArts PerfTest to quickly initiate high-concurrency performance tests with JMeter, and view complete performance test reports.
<b>Process of Using Global Variables</b>	Global variables are used to build data sets and enrich test data. If a global variable is used in a packet of a request, variable values in the packet will be replaced with specified values during a pressure test.