



云插件 Cloud Toolkit 相关工具

文档版本: 20210415



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。 如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	⚠ 危险 重置操作将丢失用户配置数据。
▲ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.PelicanDT (05
1.1. 使用文档	05
1.1.1. 停止应用	05
1.1.2. 拉高服务器内存占用率	06
1.1.3. 拉高服务器 CPU 占用率	07
1.1.4. 中断服务器网络	09
1.1.5. 延时服务器网络	11
1.1.6. 实现 Dubbo 断网	13
1.1.7. 在远程服务器执行本地 shell 脚本	14
1.2. 快速入门	16
1.3. 工具介绍	17

1.PelicanDT

1.1. 使用文档

1.1.1.停止应用

若您想停止在远程服务器上运行的应用,通过 PelicanDT 只需要一行 Java 代码即可轻松完成。

前提条件

- 您已安装 PelicanDT SDK, 请参见安装 PelicanDT SDK。
- 您已在 Linux 远程服务器上运行应用。

操作步骤

在您的工程中添加以下代码,填入远程服务器 IP、用户名和登录密码等信息并执行。

```
import com.alibaba.pelican.chaos.client.RemoteCmdClientConfig;
import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient;
import com.alibaba.pelican.chaos.client.utils.CpuUtils;
/**
* @author moyun@middleware
*/
public class KillPID {
 public static void main(String[] args) {
   //远程服务器IP
   String ip = "";
   //远程服务器用户名
   String userName = "";
   //远程服务器登录密码
   String password = "";
   RemoteCmdClientConfig config = new RemoteCmdClientConfig();
   config.setlp(ip);
   config.setUserName(userName);
   config.setPassword(password);
   RemoteCmdClient client = new RemoteCmdClient(config);
       //应用停止
   client.killProcess("vmstat");
 }
}
```

结果验证

登录远程服务器执行命令 vmstat 1 , 可在服务器命令窗口看到该进程已被停止。

[ro	[root@iz2ze0kv2rqck9wpheu5vxz ~]# vmstat 1															
procsmemorycpu																
r	b	swpd	free	buff	cache	si	so	bi	bo	in	CS U	us s	sy i	.d wo	a st	t
1	0	0	2782856	64908	317800	0	0	286	9	181	320	0	0	99	0	0
0	0	0	2782840	64908	317804	0	0	0	64	1554	2941	1	0	99	0	0
0	0	0	2782840	64908	317804	0	0	0	0	1444	2812	1	1	99	0	0
0	0	0	2782840	64908	317804	0	0	0	4	1467	2849	0	0	100	0	0
0	0	0	2782840	64908	317804	0	0	0	0	1424	2803	1	0	100	0	0
0	0	0	2782840	64908	317804	0	0	0	356	1441	2813	0	1	99	1	0
0	0	0	2781288	64912	317756	0	0	0	88	1820	3230	2	2	96	0	0
Kil	led															

1.1.2. 拉高服务器内存占用率

若需要测试应用在内存占用率高的情况下是否能正常运行,您可以通过 PelicanDT 轻松地模拟服务器内存占 用率高的测试环境。本文将介绍使用 PelicanDT 拉高服务器内存占用率的方法。

前提条件

- 您已安装 PelicanDT SDK, 请参见安装 PelicanDT SDK。
- 您已在 Linux 远程服务器上运行应用。

操作步骤

使用 PelicanDT 模拟服务器内存占用率为 75%、持续时间 1 分钟的测试环境,具体操作步骤如下:

- 1. 将下方代码粘贴至您的工程,并更改配置。
 - i. 在代码中填入您的服务器 IP、用户名和登录密码等信息。
 - ii. 设置内存占用值和持续时间。例如: 服务器内存为 8G, 设置内存占用 6144 M, 内存占用率为 75%, 持续时间 1 分钟。

```
import com.alibaba.pelican.chaos.client.RemoteCmdClientConfig;
import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient;
import com.alibaba.pelican.chaos.client.utils.CpuUtils;
import com.alibaba.pelican.chaos.client.utils.MemUtils;
/**
* @author moyun@middleware
*/
public class DemoMem {
 public static void main(String[] args) {
   //服务器IP
   String ip = "";
   //服务器用户名
   String userName = "";
   //服务器登录密码
   String password = "";
   RemoteCmdClientConfig config = new RemoteCmdClientConfig();
   config.setIp(ip);
   config.setUserName(userName);
   config.setPassword(password);
   RemoteCmdClient client = new RemoteCmdClient(config);
   //内存占用单位为M
   int percent = 6144;
   //持续时间单位分钟
   int delayMinutes = 1;
   //内存占用率拉高
   MemUtils.adjustMemUsage(client, percent, delayMinutes);
 }
}
```

2. 运行您的工程。

结果验证

在 ECS 控制台查看内存使用率,请参见在实例详情页查看实例信息。在下图中可以看到内存使用率高达 75%,持续时间为1分钟。



1.1.3. 拉高服务器 CPU 占用率

若您需要测试应用在 CPU 占用率高的情况下是否能正常运行,通过 PelicanDT 可轻松地模拟服务器 CPU 占用率高的测试环境。本文将介绍使用 PelicanDT 拉高服务器 CPU 占用率的方法。##前提条件

前提条件

- 您已安装 PelicanDT SDK, 请参见安装 PelicanDT SDK。
- 您已在 Linux 远程服务器上运行应用。

操作步骤

在本示例中,将模拟 CPU 占用率为 75%,持续时间 3 分钟的环境。

- 1. 将下方代码添加至您的工程,并进行以下配置:
 - i. 在代码中填入您的远程服务器 IP、用户名和登录密码等信息。
 - ii. 设置 CPU 占用值和持续时间。例如设置 CPU 占用率为 75%, 持续时间为 3 分钟。

```
import com.alibaba.pelican.chaos.client.RemoteCmdClientConfig;
import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient;
import com.alibaba.pelican.chaos.client.utils.CpuUtils;
/**
* @author moyun@middleware
*/
public class DemoCPU {
 public static void main(String[] args) {
   //远程服务器IP
   String ip = "";
   //远程服务器用户名
   String userName = "";
   //远程服务器登录密码
   String password = "";
   RemoteCmdClientConfig config = new RemoteCmdClientConfig();
   config.setIp(ip);
   config.setUserName(userName);
   config.setPassword(password);
   RemoteCmdClient client = new RemoteCmdClient(config);
   //设置CPU占用比例
   int percent = 70;
   //设置持续时间
   int delayMinutes = 3;
   //CPU占用率拉高
   CpuUtils.adjustCpuUsage(client, percent, delayMinutes);
 }
}
```

```
2. 运行您的工程。
```

结果验证

在 ECS 控制台上查看 CPU 监控信息,请参见在实例详情页查看实例信息。在下图中可以看到, CPU 占用率 高达 75%,持续时间为 3 分钟。



1.1.4. 中断服务器网络

若您需要测试应用在服务器网络中断的情况下是否能正常运行,通过 PelicanDT 可轻松地模拟服务器网络中断的测试环境。本文将介绍使用 PelicanDT 中断服务器网络的方法。

前提条件

- 您已安装 PelicanDT SDK , 请参见安装 PelicanDT SDK。
- 您已在 Linux 远程服务器上运行应用。

操作步骤

本示例将模拟服务器网络中断 30s 的环境,具体操作步骤如下:

- 1. 将下方代码添加至您的工程,并进行以下配置:
 - i. 在代码中填入您的远程服务器的 IP、用户名和登录密码等信息。
 - ii. 设置 blockIP, 即需要设置网络中断的服务器 IP。
 - iii. 设置中断时间,本示例中为 30 秒。

```
import com.alibaba.pelican.chaos.client.RemoteCmdClientConfig;
import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient;
import com.alibaba.pelican.chaos.client.utils.NetAccessUtils;
/**
* @author moyun@middleware
*/
public class DemoNetAccess {
 public static void main(String[] args) {
   //远程服务器IP
   String ip = "";
   //远程服务器用户名
   String userName = "";
   //远程服务器登录密码
   String password = "";
   RemoteCmdClientConfig config = new RemoteCmdClientConfig();
   config.setIp(ip);
   config.setUserName(userName);
   config.setPassword(password);
   RemoteCmdClient client = new RemoteCmdClient(config);
   //设置blockIP
   String blockIP = "";
   //持续时间s
   int delaySecond = 30;
   //网络中断
   NetAccessUtils.blockIPInput(client, blockIP, delaySecond);
 }
}
```

2. 运行您的工程。

验证结果

A 为远程服务器, B 为设置网络中断的服务器。执行示例程序前, 登录 B 服务器, 通过 B 服务器向 A 服务器 执行 ping 命令,可访问成功。执行示例程序后, 网络被中断, 访问失败, 30 秒后网络恢复, 可继续访问。

[ro	oot@sel	lf-bui	ild-k8s-cluster	004 ~]# ping
PI	NG		1.148 (DBUB).127	bytes of data.
64	bytes	from	20, 06, 277, 140;	<pre>icmp_seq=1 ttl=56 time=2.14 ms</pre>
64	bytes	from	20.00.177.100:	<pre>icmp_seq=2 ttl=56 time=2.16 ms</pre>
64	bytes	from	20, 96, 97, 146;	<pre>icmp_seq=3 ttl=56 time=2.13 ms</pre>
64	bytes	from	an. No. 177, 1889;	<pre>icmp_seq=4 ttl=56 time=2.13 ms</pre>
64	bytes	from	20.00.002.000:	icmp_seq=5 ttl=56 time=2.15 ms
64	bytes	from	10. 00. 170. Last:	icmp_seq=6 ttl=56 time=2.16 ms
64	bytes	from	20, 00, 177, 140;	<pre>icmp_seq=7 ttl=56 time=2.16 ms</pre>
64	bytes	from	PR. 66. 177. 189:	<pre>icmp_seq=8 ttl=56 time=2.12 ms</pre>
64	bytes	from	20, 26, 272, 249;	icmp_seq=9 ttl=56 time=2.13 ms
64	bytes	from	20. 00. 277. Lab:	icmp_seq=10 ttl=56 time=2.10 ms
64	bytes	from	10. NO. 177, 189;	icmp_seq=11 ttl=56 time=2.14 ms
~ 4	hud en	C	100 July 1000 1 100	i
64	bytes	from		icmp_seq=72 ttl=56 time=2.15 ms
64	bytes	from		icmp_seq=73 ttl=56 time=2.16 ms
64	bytes	from	20.00 NO. 144	icmp_seq=74 ttl=56 time=2.13 ms
64	bytes	from	20. (C. 17. 199)	<pre>icmp_seq=75 ttl=56 time=2.16 ms</pre>

1.1.5. 延时服务器网络

若您需要测试应用在服务器网络延时的情况下是否能正常运行,通过 PelicanDT 可轻松模拟服务器网络延时 的测试环境。本文将介绍使用 PelicanDT 模拟服务器网络延时环境的方法。

前提条件

- 您已安装 PelicanDT SDK, 请参见安装 PelicanDT SDK。
- 您已在 Linux 远程服务器上运行应用。

操作步骤

本示例将模拟服务器网络延时为 1000 ms 且持续 10s 的测试环境,具体操作步骤如下:

- 1. 将下方代码添加至您的工程,并进行以下配置:
 - i. 在代码中填入您的远程服务器的 IP、用户名和登录密码等信息。
 - ii. 设置延时和持续时间,本示例中设置延时为 1000 ms,持续时间为 10s。

import com.alibaba.pelican.chaos.client.RemoteCmdClientConfig; import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient; import com.alibaba.pelican.chaos.client.utils.MemUtils; import com.alibaba.pelican.chaos.client.utils.NetTrafficUtils; import com.alibaba.pelican.deployment.element.impl.AbstractElement; import java.util.Map; /** * @author moyun@middleware */ public class DemoNetTraffic { public static void main(String[] args) { //远程服务器IP String ip = ""; //远程服务器用户名 String userName = ""; //远程服务器登录密码 String password = ""; RemoteCmdClientConfig config = new RemoteCmdClientConfig(); config.setIp(ip); config.setUserName(userName); config.setPassword(password); RemoteCmdClient client = new RemoteCmdClient(config); //延时时间 ms int delayTime = 1000; //持续时间s int delaySecond = 10; //网络延时 NetTrafficUtils.setNetworkDelay(client, delayTime, delaySecond); } }

2. 运行您的工程。

结果验证

A 为远程服务器, B 为设置网络延时的服务器。执行示例程序前, 登录 B 服务器, 通过 B 服务器向 A 服务器 执行 ping 命令, 网络延时为 2.12 ms; 执行示例程序后, 网络延时为 1002 ms, 10s 后恢复为 2.12 ms。

[ro	ot@sel	.f-bui	.ld-	-k8s-clust	04 ~]# ping			
PIN	IG			e geologia	and the second	bytes	of data.	
64	bytes	from	1	94.112.144	icmp_seq=1	ttl=56	time=2.12	ms
64	bytes	from	t,	66. TO 1.6	icmp_seq=2	ttl=56	time=2.14	ms
64	bytes	from	E.	56. T T	icmp_seq=3	ttl=56	time=2.14	ms
64	bytes	from	н.	M. 17. La	icmp_seq=4	ttl=56	time=2.14	ms
64	bytes	from	1	100, 111, 100	icmp_seq=5	ttl=56	time=1002	ms
64	bytes	from	E.	96. FO La	icmp_seq=6	ttl=56	time=1002	ms
64	bytes	from	5	98.27.2 Ber	icmp_seq=7	ttl=56	time=1002	ms
64	bytes	from	1	M. 17. La	icmp_seq=8	ttl=56	time=1002	ms
64	bytes	from		100 T T T T T	icmp_seq=9	ttl=56	time=1002	ms
64	bytes	from	1	M. 173 Lat	icmp_seq=10) ttl=56	time=1002	2 ms
64	bytes	from		96.171.24	<pre>icmp_seq=11</pre>	. ttl=56	time=1002	2 ms
64	bytes	from	E.	M. 177 Lat	icmp_seq=12	2 ttl=56	time=1002	2 ms
64	bytes	from	ц.	96.110.5M	icmp_seq=13	8 ttl=56	time=1002	2 ms
64	bytes	from	1	N. 172 Los	icmp_seq=15	5 ttl=56	time=2.12	2 ms
64	bytes	from		56. TT 56	<pre>icmp_seq=16</pre>	5 ttl=56	time=2.14	l ms
64	bytes	from	E.	M. 172 MM	icmp_seq=17	'ttl=56	time=2.13	3 ms
64	bytes	from			icmp_seq=18	8 ttl=56	time=2.14	1 ms
64	bytes	from	1	8,12,14	icmp_seq=19) ttl=56	time=2.13	3 ms

1.1.6. 实现 Dubbo 断网

若您需要在 Dubbo 断网的环境下测试应用是否能正常运行,使用 PelicanDT 可以轻松地模拟 Dubbo 断网的 测试环境。本文将介绍使用 PelicanDT 实现 Dubbo 断网的方法。

前提条件

- 已安装 PelicanDT SDK, 请参见安装 PelicanDT SDK。
- 已准备好 Linux 远程服务器。

操作步骤

本示例将模拟 Dubbo 环境下 8085 端口断网 20s 的测试环境。具体操作步骤如下:

- 1. 载 Dubbo-example 工程(用于禁止端口网络访问的示例)。
- 2. 在路径 Dubbo-example/src/test/resources/env/func 下打开 Dubbo.properties 配置文件,并填入您 的服务器 IP、用户名和登录密码等信息。
- 3. 在路径 Dubbo-example/src/test/java/com/alibaba/pelican/rocketmq 下打开 TestDubboNetwork.java 文件并运行单元测试。

结果验证

执行程序后,日志输出内容如下:

```
2019-02-01 17:20:30 [INFO] [main] c.a.p.c.client.utils.NetAccessUtils - Block port 8085 protcol TCP, dalay time
20 seconds.
2019-02-01 17:20:45 [INFO] [main] c.a.p.rocketmq.TestDubboNetwork - Operation timed out (Connection ti
med out)
Hello 123123, response form provider: 10.66.204.25:20880
```

• 通过第1行日志可以看出, 8085端口断网;

- 通过第 2 行日志可以看出, 在 8085 端口断网的情况下, 接口访问超时;
- 通过第3行日志可以看出,端口网络恢复后,接口访问成功。

1.1.7. 在远程服务器执行本地 shell 脚本

若您想在远程服务器执行本地 shell 脚本,使用传统方法需要先上传脚本,再登录远程服务器执行脚本,步骤繁多。使用 PelicanDT,可以通过一行 Java 代码实现此操作。

前提条件

- 已安装 PelicanDT SDK。请参见安装 PelicanDT SDK。
- 已在 Linux 远程服务器上运行应用。

背景信息

本文使用 demo 工程来进行说明。

操作步骤

- 1. 在本地创建 shell 脚本。
- 2. 在您的工程中执行以下代码。

⑦ 说明 请在代码中填入您的远程服务器 IP、用户名、登录密码和脚本地址等信息。

package com.alibaba.pelican.demo; import com.alibaba.pelican.chaos.client.RemoteCmdClientConfig; import com.alibaba.pelican.chaos.client.RemoteCmdResult; import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient; import com.alibaba.pelican.chaos.client.utils.CpuUtils; import org.apache.commons.io.IOUtils; /** * @author moyun@middleware */ public class DemoExecScript { public static void main(String[] args) { //服务器IP String ip = ""; //服务器用户名 String userName = ""; //服务器登录密码 String password = ""; RemoteCmdClientConfig config = new RemoteCmdClientConfig(); config.setIp(ip); config.setUserName(userName); config.setPassword(password); RemoteCmdClient client = new RemoteCmdClient(config); //脚本地址 String scriptPath = "demo.sh"; RemoteCmdResult result = client.scpAndExecScript(); System.out.println(result.getStdInfo()); } }

执行结果

以 demo 工程为例,执行程序后,日志输出内容如下:

```
[root@iZ2zejdbo0lni6dwgifbhqZ ~]$cd /root/scripts/
[root@iZ2zejdbo0lni6dwgifbhqZ ~/scripts]$sh demo.sh
total 56
dr-xr-x---. 6 root root 4096 Jan 25 15:32.
dr-xr-xr-x. 18 root root 4096 Jan 24 14:16 ..
-rw------ 1 root root 227 Jan 24 21:01 .bash_history
-rw-r--r-. 1 root root 18 Dec 29 2013 .bash_logout
-rw-r--r-. 1 root root 176 Dec 29 2013 .bash_profile
-rw-r--r-. 1 root root 176 Dec 29 2013 .bashrc
drwx----- 3 root root 4096 Oct 15 2017 .cache
-rw-r--r-. 1 root root 100 Dec 29 2013 .cshrc
drwxr-xr-x 2 root root 4096 Oct 15 2017.pip
-rw-r--r-- 1 root root 64 Oct 15 2017 .pydistutils.cfg
drwxr-xr-x 2 root root 4096 Jan 25 15:32 scripts
drwx----- 2 root root 4096 Jan 24 14:16 .ssh
-rw-r--r-. 1 root root 129 Dec 29 2013 .tcshrc
-rw------ 1 root root 596 Jan 25 15:32 .viminfo
[root@iZ2zejdbo0lni6dwgifbhqZ ~/scripts]$export HISTFILE=/dev/null
[root@iZ2zejdbo0lni6dwgifbhqZ ~/scripts]$exit
logout
```

1.2. 快速入门

若您需要测试应用在特殊环境下是否能正常运行,可以通过 PelicanDT 来模拟停止应用、CPU 占用率过高、 CPU 内存过高、网络中断、网络流量延时等异常环境以进行测试。本文以向服务器注入查询当前路径的命令 为例,介绍 PelicanDT 的使用方法。

安装 PelicanDT SDK

将 PelicanDT SDK 添加到项目中有两种方式:

- 下载 SDK 源码包,并添加到您的工作目录中。
- 在 Maven 项目的 pom.xml 文件中添加以下代码,引入 PelicanDT SDK 依赖。

```
<dependency>
<groupId>com.alibaba.pelican</groupId>
<artifactId>PelicanDT</artifactId>
<version>1.0.9</version>
</dependency>
```

注入命令

在您的工程中执行以下代码,向远程服务器注入查询当前路径的命令。

⑦ 说明 您需要在代码中填入您的 ECS 公网 IP、ECS 用户名和登录密码等信息。

```
import com.alibaba.pelican.chaos.client.impl.RemoteCmdClient;
import lombok.extern.slf4j.Slf4j;
import org.junit.Test;
/**
 * @author moyun@middleware
*/
@Slf4j
public class TestRemoteCmdClient {
 @Test
 public void testRemoteCmdClient() {
   //ECS 公网 IP
   String ip = "";
   //ECS 用户名,一般为 root
   String userName = "";
   //ECS 登录密码
   String password = "";
   //创建并初始化 RemoteCmdClient 实例
   RemoteCmdClientConfig remoteCmdClientConfig = new RemoteCmdClientConfig();
   remoteCmdClientConfig.setIp(ip);
   remoteCmdClientConfig.setUserName(userName);
   remoteCmdClientConfig.setPassword(password);
   RemoteCmdClient client = new RemoteCmdClient(remoteCmdClientConfig);
   //执行pwd命令
   RemoteCmdResult resultInfo = client.execCmdWithPTY(new RemoteCmd("pwd"));
   log.info(resultInfo.getStdInfo());
 }
}
```

结果验证

执行程序后,日志输出内容如下:

```
[root@iz2ze0kv2rqck9wpheu5vxz ~]$pwd
/root
[root@iz2ze0kv2rqck9wpheu5vxz ~]$export HISTFILE=/dev/null
[root@iz2ze0kv2rqck9wpheu5vxz ~]$exit
logout
```

通过第 2 行内容可以看出, 命令执行后, 屏幕打印出当前目录为 /root。

1.3. 工具介绍

PelicanDT(Pelican Distributed Test),是阿里云提供的一款针对 Linux 系统的测试环境模拟工具,是主要 针对分布式应用提供的集成测试解决方案,用于帮助开发者简单、高效地测试分布式应用。PelicanDT 现 已<mark>开源</mark>。

PelicanDT 具有以下特点:

- 使用 Java 语言与 Linux 系统交互。
- 本地控制远程服务器上的 Linux 系统执行命令。

● 通过简单的操作对服务器注入异常来模拟测试环境。例如:停止应用、CPU占用率过高、CPU内存过高、 网络中断、网络流量延时等测试环境。