ALIBABA CLOUD

阿里云

Web应用托管服务

CLI

文档版本：20210908

阿里云

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。

1. 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。

2. 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。

3. 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。	注意 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。	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 list --instanceid` *Instance_ID*
[] 或者 [a|b]	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 {active|stand}`

目录

1.安装升级Web+命令行界面（wpctl） 05
2.CLI命令 06
3.部署环境配置文件wpfile配置项说明 15
4.配置命令行自动补全 58
5.初始化CLI 59
6.创建应用 62
7.创建部署环境 63
8.更新环境 66
9.切换应用和环境 67
10.扩缩环境 68
11.删除环境和应用 69
12.查看环境和应用 70
13.环境配置模板 73
14.收集报错信息 74
15.Host操作 75

1.安装升级Web+命令行界面（wpctl）

Web+命令行界面 (wpctl) 是一个您可以用来创建、配置和管理Web+部署环境的命令行客户端。wpctl用作开发和测试的一部分，可以取代Web+控制台。

在Linux上安装wpctl

安装

运行以下命令：

```
eval "$(curl -s -L https://webplus-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/cli/install.sh)"
```

升级

如果您使用的CLI版本低于wpctl兼容最低版本号，在使用wpctl时会提示：The CLI version is too old. Please upgrade the CLI，请执行**wpctl upgrade**来升级CLI。

卸载

直接执行：`rm /usr/local/bin/wpctl`。

在macOS上安装wpctl

安装

运行以下命令：

```
eval "$(curl -s -L https://webplus-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/cli/install.sh)"
```

升级

如果您使用的CLI版本低于wpctl兼容最低版本号，在使用wpctl时会提示：The CLI version is too old. Please upgrade the CLI，请执行**wpctl upgrade**来升级CLI。

卸载

直接执行：`rm /usr/local/bin/wpctl`。

2.CLI命令

Web+命令行界面（wpctl）使您可以直接从终端轻松创建和管理Web+，下面罗列了Web+的核心CLI命令解释说明。您还可以使用 `wpctl --help` 、 `wpctl help` 或 `wpctl h` 在终端中查看所有的CLI命令。

应用相关

查询应用

查询本账号下的应用的列表。

```
USAGE:
 wpctl app:list
EXAMPLES:
 wpctl app:list
```

切换应用

查询应用后切换到目标应用。

```
USAGE:
 wpctl app:use ID/Name
EXAMPLES:
 wpctl app:use app-demo
```

参数	Flag
应用ID或应用Name	无

删除应用

删除应用需要释放应用下的所有环境。

```
USAGE:
 wpctl app:delete ID/Name
EXAMPLES:
 wpctl app:delete app-demo
```

参数	Flag
应用ID或应用Name	无

环境相关

apply命令

`apply` 命令可以创建应用、创建环境和更新环境， `apply --help` 命令可以自动获取工作目录的数据。

```
USAGE:
   wpctl env:apply [Wpfile] [*.zip] [flags]
FLAGS:
  --package-id  set the package id
  --package,-p  set the package url
  --label,-l   set the label of package
  --template,-t set the template id(only) to apply
  --type      set the default env type, 'HighAvailability' [HA] or 'StandAlone' [SA]
  --app,-a     set the application
  --env,-e     set the env
  --category   set the category of the new application('Tomcat', or 'Java')
  --stack-id   set the stackId of the new application
  --quiet,-q   run quietly
  --json,-j    result return as json
  --create-on-absent,-C create application or environment on absent，combine with --app/env flags
  --help      print help
`wpctl apply
  --app target_app (--create-on-absent to create when it doesn't exist)
  --env target_env (--create-on-absent to create when it doesn't exist)
  --package https://*** --label v1 (or just using existed package with --pkgId)`
`wpctl apply --template template***
  --app app_template_belongs_to
  --env new_env_name --create-on-absent
```

apply可不接参数，也可使用Wpfile类文件作为参数，也可使用zip包作为参数。

Flags	说明
--package-id	设置复用的包版本。
--package或-p	设置部署包。
--label或-l	设置包标识。
--template或-t	设置部署的模板ID。
--type	设置使用的配置类型： • 高可用：HighAvailability或HA。 • 低成本：LowCost或LC。
--app或-a	设置应用。
--env或-e	设置环境。
--category	设置新建应用的平台类型：Tomcat或Java。

Flags	说明
--stack-id	设置新建应用的技术栈ID: • Tomcat 8.5 / OpenJDK 8 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823983 • Tomcat 8.5 / Dragonwell 8 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c482398d • OpenJDK 8 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823984 • Dragonwell 8 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c482398c • Node.js 8.16.0 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823985 • Node.js 10.16.0 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823986 • Native / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823987 • Go 1.12.7 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823988 • PHP 7.3 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823989 • Python 3.7.4 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c482398a • Python 2.7.16 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c482398b • ASP.NET Core 2.2 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c482398e • ASP.NET Core 3.0 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823990 • ASP.NET Core 3.1 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c4823991 • Ruby 2.6.3 / Aliyun Linux 2.1903：ws-6c937c98a9c0296d0c482398f
--quiet或-q	静默执行命令，不会输出过程信息和返回信息，如需要信息输出请使用 `--json` 。
--json或-j	以json字串格式输出结果信息
--create-on-absent或-C	与 `--app` 或 `--env` 结合使用，用于在指定的app或者env不存在时创建应用或环境。

初始化环境

初始化环境的工作目录，包括区域，默认应用，默认环境。

```
USAGE:
 wpctl init
```

查看环境列表

查看当前应用下的环境列表。

```
USAGE:
 wpctl env:list [flags]
FLAGS:
 --app,-a set the application
 --help  print help
EXAMPLES:
 wpctl env:list --app appName
```

切换环境

切换应用内的部署环境。

```
USAGE:
 wpctl env:use id/name
FLAGS:
 --app,-a set the application
 --help  print help
```

显示环境信息

```
USAGE:
 wpctl env:info [flags]
FLAGS:
 --env,-e set the env
 --app,-a set the application
 --json,-j result return as json
 --help  print help
EXAMPLES:
 wpctl env:info --env envName
```

启动环境

```
USAGE:
 wpctl env:start [flags]
FLAGS:
 --app,-a set the application
 --env,-e set the env
 --help  print help
EXAMPLES:
 wpctl env:start --env envId/name
```

停止环境

```
USAGE:
 wpctl env:stop [flags]
FLAGS:
 --app,-a set the application
 --env,-e set the env
 --help  print help
EXAMPLES:
 wpctl env:stop --env envId/name
```

释放环境

```
USAGE:
 wpctl env:terminate [flags]
FLAGS:
 --app,-a  set the application
 --env,-e  set the env
 --help    print help
EXAMPLES:
 wbnx env:terminate
```

扩缩环境

```
USAGE:
 wpctl env:scale instanceNum [flags]
FLAGS:
 --app,-a set the application
 --env,-e set the env
 --help  print help
EXAMPLES:
 wbnx env:scale 1
```

删除环境

```
USAGE:
 wpctl env:delete ID/Name [flags]
FLAGS:
 --app,-a set the application
 --help  print help
EXAMPLES:
 wpctl env:delete env-demo
```

生成环境模板

```
USAGE:
 wpctl env:save [flags]
FLAGS:
 --name,-n set the name
 --app,-a  set the application
 --env,-e  set the env
 --help    print help
EXAMPLES:
 wpctl env:save --name templateName
```

下载环境

```
USAGE:
 wpctl env:dump [flags]
FLAGS:
 --env,-e set the env
 --app,-a set the application
 --help  print help
EXAMPLES:
 wpctl env:dump --env envName
```

检查环境健康

```
USAGE:
 wpctl env:health [flags]
FLAGS:
 --env,-e set the env
 --app,-a set the application
 --help   print help
EXAMPLES:
 wpctl env:health --env envName --app appName
```

监控环境

```
USAGE:
 wpctl env:top [flags]
FLAGS:
 --env,-e set the env
 --app,-a set the application
 --help   print help
EXAMPLES:
 wpctl env:top --env envName
```

查看环境事件

```
USAGE:
 wpctl env:events [flags]
FLAGS:
 --env,-e   set the env
 --app,-a   set the application
 --change,-c set the changeId
 --help     print help
EXAMPLES:
 wpctl env:events --env envName
```

更新环境部署包

```
USAGE:
 wpctl env:deploy package [flags]
FLAGS:
 --label,-l set the label of package
 --app,-a   set the application
 --env,-e   set the env
 --help     print help
EXAMPLES:
 wpctl env:deploy *.war --label v2
```

访问环境

```
USAGE:
 wpctl env:open [flags]
FLAGS:
 --app,-a set the application
 --env,-e set the env
 --help   print help
EXAMPLES:
 wpctl env:open --env envId/name
```

实例操作相关

登录实例

通过SSH登录环境下的实例。

```
USAGE:
 wpctl env:ssh envName/ID [flags]
FLAGS:
 --app,-a set the application
 --help   print help
EXAMPLES:
 wpctl env:ssh envName/ID --app appName/ID
```

在环境实例上执行命令

```
USAGE:
 wpctl env:exec <selector> <shell_command>[flags]
 :selector - `envName/id`
 The command will be execute by `root`
FLAGS:
 --quiet,-q   run quietly
 --timeout,-t execution timeout (in seconds, max is 86400)
 --json,-j    output as json (result will be base64 encoded)
 --force,-f   force to run command on all available servers
 --help       print help
EXAMPLES:
 wpctl env:exec envName 'echo hello'
```

模板相关

展示模板

```
USAGE:
 wpctl template:list [flags]
FLAGS:
 --app,-a set the application
 --help   print hel
EXAMPLES:
 wpctl template:list
```

删除模板

```
USAGE:
 wpctl template:delete id/name [flags]
FLAGS:
 --app,-a set the application
 --help   print help
EXAMPLES:
 wpctl template:delete templateName/ID
```

启动模板

```
USAGE:
 wpctl template:launch
FLAGS:
 --app,-a set the application
 --env,-e set the env
 --create-on-absent,-C create application or environment on absent，combine with --app/env flags
 --help   print help
EXAMPLES:
 wpctl template:launch templateName/id
```

其他命令

展示部署包列表

```
USAGE:
 wpctl pkg:list [flags]
FLAGS:
 --app,-a set the application
 --help   print help
EXAMPLES:
 wpctl pkg:list --app appName
```

配置账号

```
USAGE:
 wpctl configure --mode <AuthenticateMode> --profile <profileName>
COMMANDS:
 set   set config in non interactive mode
 list   list all config profile
 delete delete config profile
FLAGS:
  --language       use `--language [en|zh]` to assign language
 --region         use `--region <regionId>` to assign region
 --access-key-id    use `--access-key-id <AccessKeyId>` to assign AccessKeyId, required in AK/StsToken/RamRol
eArn mode
 --access-key-secret use `--access-key-secret <AccessKeySecret>` to assign AccessKeySecret
 --help      print help
```

自动补全命令

脚本安装时，当前终端会自动补全，但是切换了shell 的tab则失效。

```
USAGE:
 wpctl auto-completion [flags]
FLAGS:
 --uninstall uninstall auto completion
 --help     print help
```

查看CLI版本

```
USAGE:
 wpctl version
EXAMPLE:
 wpctl version
```

更新CLI版本

```
USAGE:
 wpctl upgrade
EXAMPLE:
 wpctl upgrade
```

收集CLI信息

```
USAGE:
 wpctl collect [flags]
FLAGS:
 --all  collect all log
 --help print help
EXAMPLE:
 wpctl collect
```

检测产品开通、依赖和授权服务

```
USAGE:
 wpctl doctor
EXAMPLE:
 wpctl doctor
```

3.部署环境配置文件wpfile配置项说明

部署环境配置项包含了创建一个部署环境所需的全部信息。配置是可重复的，环境独占的资源比如导入的ECS实例不属于重复配置。

平台

配置路径	配置项	配置名	默认值	值域
proxy	type	反向代理类型	nginx	[nginx,apache]

配置路径	配置项	配置名	默认值	值域
proxy.nginx	version	Nginx版本	1.14.2	1.14.2

配置路径	配置项	配置名	默认值	值域
proxy.apache	version	Apache版本	2.4.6	2.4.6

资源

resources.network：网络配置

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
vpcOption	网络选项	{"vpcId":"","vSwitches":[]}	

resources.ecs.autoScaling：实例配置

配置项	配置名	默认值	值域
instanceType	实例规格	["ecs.g5.large"]	

配置项	配置名	默认值	值域
instanceChargeType	实例付费模式	PrePaid 、PostPaid	PostPaid
periodUnit	时长	Week、Month	Month
period	实例时长	• 当periodUnit为Week时，period取值为：1, 2, 3, 4 • 当periodUnit为Month时，period取值为： 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, 36, 48, 60	1

配置项	配置名	默认值	值域
instanceNum	实例数量	1	[0,100]
securityGroupIds	安全组	[]	

配置项	配置名	默认值	值域
keyPairName	密钥对		
systemDiskSize	系统磁盘		

配置项	配置名	默认值	值域
enableInternet	启用公网IP	true	[true,false]

配置项	配置名	默认值	值域
instanceName	实例名称		
scalingPolicy	主机回收模式	recycle	[release,recycle]

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
userData	实例自定义数据		

配置项	配置名	默认值	值域
systemDiskCategory	系统盘的磁盘类型	cloud_efficiency	[cloud,cloud_efficiency,cloud_ssd,ephemeral_ssd]

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
multiAzPolicy	多可用区伸缩组 ECS 实例扩缩容策略	BALANCE	[PRIORITY,BALANCE]

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
internetChargeType	网络计费类型	PayByTraffic	[PayByBandwidth,PayByTraffic]

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
loginCredential	登录凭证	Password、Keypair、None	None
password	密码	字符串	无

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
internetMaxBandwidthIn	公网入带宽最大值	100	[1,200]

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
internetMaxBandwidthOut	公网出带宽最大值	50	[0,100]
ioOptimized	是否为 I/O 优化实例	true	[true,false]

resources.slb.internet：公网SLB配置

配置项	配置名	默认值	值域
enable	启用公网SLB	true	[true,false]

配置项	配置名	默认值	值域
slbId	公网SLB		
instanceChargeType	实例付费模式	PrePaid 、PostPaid	PostPaid
periodUnit	时长	Week、Month	Month

配置项	配置名	默认值	值域
period	实例时长	• 当periodUnit为Week时，period取值为：1, 2, 3 • 当periodUnit为Month时，period取值为： 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, 36	1
listenerPort	公网SLB监听端口	80	[1,65535]
protocol	公网SLB协议	http	[http,tcp]

配置项	配置名	默认值	值域
forwardingRule	公网SLB转发规则		
bandwidth	监听的带宽峰值	-1	[-1,5120]
loadBalancerSpec	负载均衡实例的规格	slb.s1.small	[slb.s1.small,slb.s2.small,slb.s2.medium,slb.s3.small,slb.s3.medium,slb.s3.large]

配置项	配置名	默认值	值域
internetChargeType	公网类型实例的付费方式	paybytraffic	[paybybandwidth,paybytraffic]
healthCheckUrl	健康检查URl	/	

resources.slb.intranet：内网SLB配置

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
enable	启用内网SLB	true	[true,false]

配置项	配置名	默认值	值域
slbId	内网SLB		
instanceChargeType	实例付费模式	PrePaid 、PostPaid	PostPaid
periodUnit	时长	Week、Month	Month

配置项	配置名	默认值	值域
period	实例时长	• 当periodUnit为Week时，period取值为：1, 2, 3 • 当periodUnit为Month时，period取值为：1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, 36	1
listenerPort	内网SLB监听端口	80	[1,65535]
protocol	内网SLB协议	http	[http,tcp]

配置项	配置名	默认值	值域
forwardingRule	内网SLB转发规则		
bandwidth	监听的带宽峰值	-1	[-1,5120]
bandwidth	监听的带宽峰值	-1	[-1,5120]
loadBalancerSpec	负载均衡实例的规格	slb.s1.small	[slb.s1.small,slb.s2.small,slb.s2.medium,slb.s3.small,slb.s3.medium,slb.s3.large]

配置项	配置名	默认值	值域
internetChargeType	公网类型实例的付费方式	paybytraffic	[paybybandwidth,paybytraffic]

resources.rds：RDS配置

配置项	配置名	有效值	默认值	描述
enable	是否开启	true / false	false	是否启用RDS。
imported	是否为导入	true / false	false	是否使用已有RDS。

配置项	配置名	有效值	默认值	描述
rdsId	RDS ID	有效的RDS实例ID	无	RDS实例ID。在导入已有RDS场景下，需要提供此配置项来指明导入的RDS。
instanceChargeType	实例付费模式	PrePaid 、PostPaid	PostPaid	代购实例的付费模式
periodUnit	时长	Week、Month	Month	包年包月实例时长单位
period	实例时长	• 当periodUnit为Week时，period取值为：1, 2, 3 • 当periodUnit为Month时，period取值为： 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 24, 36	1	包年包月实例时长
zoneId	可用区	有效的RDS可用区ID	无	RDS可用区ID。可通过RDS的DescribeRegions接口来获取有效的RDS可用区。
VSwitches	交换机列表	交换机ID	无	指定RDS所在可用区的交换机。
engine	数据库类型	数据库类型，取值：MySQL ； SQLServer ； PostgreSQL ； PPAS ； MariaDB 。	无	数据库类型。
engineVersion	数据库版本	数据库版本，取值：MySQL：5.5/5.6/5.7/8.0 ；SQL Server：2008r2/2012/2012_ent_ha/2012_std_ha/2012_web/2016_ent_ha/2016_std_ha/2016_web/2017_ent ；PostgreSQL：9.4/10.0 ； PPAS：9.3/10.0 ；MariaDB： 10.3 。	无	数据库版本。

配置项	配置名	有效值	默认值	描述
storageType	存储类型	实例存储类型，取值：local_ssd / ephemeral_ssd：本地SSD盘（推荐）；cloud_ssd：SSD云盘；cloud_essd：ESSD云盘。	无	存储类型。
storageSize	存储空间大小	整数	100	存储大小，单位G。
dbInstanceClass	实例规格	有效的实例规格	无	实例规格。
databaseName	库名	字符串	webplus	数据库名。
characterSetName	库字符集	字符集，取值：MySQL/MariaDB实例： utf8、gbk、latin1、utf8mb4；SQL Server实例：Chinese_PRC_CI_AS、Chinese_PRC_CS_AS、SQL_Latin1_General_CP1_CI_AS、SQL_Latin1_General_CP1_CS_AS、Chinese_PRC_BIN。	MySQL/MariaDB实例：utf8mb4；SQL Server实例：Chinese_PRC_CI_AS	字符集。
accountName	账号名	字符串	webplus	数据库账号。
accountPassword	账号密码	字符串	无	数据库账号密码。
category	类型	实例系列，取值： • Basic：基础版 • HighAvailability：高可用版 • AlwaysOn：集群版 • Finance：金融版（仅中国站支持）	无	数据库实例系列。

应用

application.option：服务端口

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
port	服务端口	8080	[1024,65535]

application.healthCheck：健康检查配置

配置项	配置名	默认值	值域

配置项	配置名	默认值	值域
type	健康检查类型	http	[http, tcp]

配置项	配置名	默认值	值域
retryCount	健康检查重试次数	3	[1,30]
intervalSeconds	健康检查间隔	3	[1,60]

配置项	配置名	默认值	值域
timeoutSeconds	健康检查超时时间	3	[1,60]

application.commands：操作命令

配置项	配置名
start	启动命令
stop	停止命令

hooks：生命周期挂钩

配置项	配置名

配置项	配置名
prestart	启动前置命令
poststart	启动后置命令
prestop	停止前置命令

配置项	配置名
poststop	停止后置命令
postinit	初始化后置命令

application.jvmOpts： Java虚拟机参数

配置项	配置名

配置项	配置名
value	JVM参数

application.enviromentVariables：环境变量

配置项	配置名
value	环境变量

4.配置命令行自动补全

按照安装命令安装CLI工具时，默认会为当前终端tab执行自动补全命令，但是切换了终端后该自动补全功能将会失效，因此建议您主动执行命令行自动补全操作。执行auto-completion后需要source配置文件。

语法

auto-completion

输出示例和描述

```
$ wpctl auto-completion
$ source ~/.bashrc
$ wpctl <tab>
app:delete    collect    env:apply    env:events    init    env:save    env:stop    pkg:list    template:list
app:list    configure    env:delete    env:exec    env:list    env:scale    env:terminate  select    upgrade
app:use    console    env:deploy    env:health    env:start    env:top    template:delete  version    aut
o-completion
doctor    env:dump    env:info    env:open
```

描述

其中source的对象文件可能为 `.bashrc` ， `.bash_profile` ， `.bash_login` ， `.profile` 。如某文件内容中有 `complete -C /usr/local/bin/wpctl wpctl` ， 即代表source的对象是该文件。

5.初始化CLI

在使用CLI前需要进行全局配置和检查：包括使用 `wpctl configure` 设置账号AK、SK、语言等全局信息和使用 `wpctl doctor` 校验此账号权限等，以及使用 `wpctl init` 初始化当前工作目录下的默认地域、应用和环境等信息。

步骤一：配置全局信息

使用 `wpctl configure` 设置AK、SK和语言等全局信息。

方法1：执行命令并创建一个名叫test的profile。（Region ID信息请参见Web+目前支持的地域。）

```
$ wpctl configure --profile test
Configuring profile 'test' in '' authenticate mode...
Access Key Id []: yourAk
Access Key Secret []: yourSk
Default Region Id []: cn-shenzhen
Default Output Format [json]: json (Only support json))
Default Language [zh|en] en: en
Saving profile[test] ...Done.
Configure Done!!!
$ wpctl configure list
Profile   | Credential       | Valid   | Region        | Language
--------- | ---------------- | ------- | ------------- | --------
test *    | AK:***rAk        | Valid   | cn-shenzhen   | en
```

方法2：使用一条完整配置命令进行配置。

```
$ wpctl configure -p test1 --access-key-id yourAk --access-key-secret yourSk --region cn-shenzhen --language e
n
Configuring profile 'test1' in '' authenticate mode...
Saving profile[test1] ...Done.
Configure Done!!!
$ wpctl configure list
Profile   | Credential       | Valid   | Region        | Language
--------- | ---------------- | ------- | ------------- | --------
default   | AK:***zGQ        | Valid   | cn-shenzhen   | en
test1 *   | AK:***zGQ        | Valid   | cn-shenzhen   | en
```

方法3：使用环境变量配置，环境变量罗列如下。

```
ALICLOUD_ACCESS_KEY
ALICLOUD_SECRET_KEY
ALICLOUD_REGION
```

切换profile并修改地域等参数。

```
$ wpctl configure set --profile default --region cn-hangzhou
$ wpctl configure list
Profile  | Credential      | Valid  | Region        | Language
--------- | ----------------- | ------- | ---------------- | --------
default * | AK:***zGQ       | Valid  | cn-hangzhou    | en
jungle   | AK:***3M1       | Valid  | cn-shenzhen    | en
test     | AK:***rAk       | Valid  | cn-shenzhen    | en
```

步骤二：校验账号权限

使用 `wpctl doctor` 命令可对当前账号的权限等进行校检。首先校检是否开通Web+业务，然后校检是否开通关联产品，最后校检Web+是否完成需要角色的授权，校检通过才可以使用其他功能。

```
$ wpctl doctor
[OK] Describe aliyun authority (1s)
[Products related checked]
+----+------+-----------------------+-------------------------------+--------+
| NO | NAME |      TITLE            |          DESC                 | STATUS |
+----+------+-----------------------+-------------------------------+--------+
| 1 | ESS  | Auto Scaling          | Elastic scaling (Auto ...     | OK     |
| 2 | OSS  | Object Storage Service | Massive, secure, ...          | OK     |
+----+------+-----------------------+-------------------------------+--------+
[OK] Describe role status (1s)
[Role related checked]
+----+---------+--------------------------------+------------+
| NO | SERVICE |          ROLE                  | AUTHORIZED |
+----+---------+--------------------------------+------------+
| 1 | WebPlus | AliyunWebPlusDefaultRole       | true       |
| 2 | ECS     | AliyunECSInstanceForWebPlusRole | true       |
+----+---------+--------------------------------+------------+
[doctor check ok, have fun]
```

步骤三：选择应用和环境

使用 `wpctl init` 初始化当前工作目录。

选择的地域没有应用时会要求创建一个新的应用。同时在CLI中所有的交互都将显示为绿色，您一般只需要选择序号或者按提示输入即可。

```
$ wpctl init
1) cn-hangzhou
2) cn-shenzhen
Select a default region, default region is [2:cn-shenzhen]: 1
No application get
0) new a application
Select a default application, default is create a application [0:new a application]:0
You are going to create a application, please enter application name: demo-test
...
[OK] Create application demo-test
No environment to select
...
[wpctl init successed]
```

当选择的地域已有应用时，选择默认环境后将保存初始化的结果到工作目录 `.webplus` 中。

```
$ wpctl init
...
0) new a application
1) demo-test2
2) demo-test1
Select a default application, default Application name [1:demo-test2]: 1
...
1) demo-test-env
Select a default environment, default environment [demo-test-env]: 1
Saving config to wpconfig:[/Users/***/Documents/webp/webp-cli-demo/.webplus/wpconfig.yaml
```

6.创建应用

您有两个方式创建应用：

- 在 `wpctl init`

```
$ wpctl init
...
No application get
0) new a application
Select a default application, default is create a application [0:new a application]:0
You are going to create a application, please enter application name: demo-test
...
[OK] Create application demo-test
...
```

- `webxctl apply` 时指定不存在的应用，添加 `--create-on-absent` flag来促使创建操作发生，此时应用和环境都会被创建。

```
wpctl env:apply
 --app target_app (此应用并不存在)
 --env target_env (此环境也不存在)
 --package https://*** --label v1 (or just using existed package with --pkgId)
 --create-on-absent (保证会进行创建操作)
```

7.创建部署环境

用户创建部署环境一般需要已创建了应用，特殊情况下可以应用和部署环境一同创建（见创建应用的第二种方式）。

创建部署环境简介

创建部署环境时，需要理解工作目录和Wpfile的意义。

工作目录：设置了当前目录下的所有操作的默认应用对象和默认部署环境对象，例如执行wpctl list将会读取当前工作目录的默认应用，列出它的部署环境列表。通过 `wpctl info` 可以查看当前工作目录下默认应用和默认部署环境。

Wpfile：保存了创建或者更新部署环境的配置信息，Wpfile与apply中的 `--type` flag互斥。Wpfile内配置项说明请参见部署环境配置文件wpfile配置项说明。

apply目前有两种使用方式：

- 直接使用wpctl apply
- wpctl apply Your_Wpfile

直接使用wpctl apply

这时候将读取当前工作目录的设置情况，同时如果当前工作目录有Wpfile文件时（名称必须为Wpfile），将读取Wpfile文件内容作为apply的配置。详细情况参考**apply flags**控制。

```
$ pwd
/Users/***/Documents/webp/webp-cli-demo
$ ls
<空目录>
$ wpctl info //查看当前工作目录是否有默认环境
No env in this app:demo-test
$ wpctl env:apply --env demo-test-env --create-on-absent
please input new package source : file:///Users/jungle/go/src/github.com/aliyun/aliyun-cli/hello.war
[No envId to do configSetting diff, you are going to create a env]
...
[OK] Create environment name demo-test-env (2s)
...
```

wpctl apply Your_Wpfile

用户指定了Wpfile时，（名称不固定，可以任意命名，例如jenkins、zookeeper等）cli会忽略工作目录的配置，读取指定的Wpfile内容，同样可以通过flags控制创建或者更新部署环境， 详细见**apply flags**控制。指定方式有：

- file://
- https://
- http://
- 本地文件路径方式（相对路径、绝对路径）

```
$ wpctl env:apply your_Wpfile --app your_app --env your_env --create-on-absent
...
[No envId to do configSetting diff, you are going to create a env]
...
[Adding config settings below]
+----+----------------------------------+-------------------------+-------------------------------+
| NO |          PATH                    |        NAME             |          VALUE                |
+----+----------------------------------+-------------------------+-------------------------------+
| 1 | platform                          | category                | Tomcat                        |
| 2 | platform                          | stackName               | Aliyun Linux 2.1903 - Java 8 - |
|  | ...                                |                         | Tomcat 8.5                    |
...
[OK] Create environment name your_env (2s)
...
```

控制apply flags

控制apply操作的执行动作的主要是三类因素：

- `--app` 、 `--env` 指定操作的应用和部署环境。
- 工作目录中的默认应用和部署环境。
- `--create-on-absent` 指定1中flag对象不存在时进行创建。 控制部署环境获取的flag

部署环境获取	--env	工作目录中默认环境	--create-on-absent	执行动作
1	有	NA	有	更新env指定环境，不存在则创建环境
2	有	NA	NA	更新env指定环境，不存在则报错
3	无	有	NA	更新部署环境
4	无	无	NA	报错

应用获取的flag控制

应用获取	--app	工作目录中默认应用	--create-on-absent	执行动作
1	有	NA	有	操作app指定应用，不存在则创建应用
2	有	NA	无	操作app指定应用，不存在则报错
3	无	有	NA	操作应用
4	无	无	NA	报错

根据模板创建部署环境

用户可以根据此应用下已保存的模板在此应用下创建一个环境，使用时直接使用 `--template` 指定模板ID即可，此时CLI将排斥除 `--env` 、 `--app` 、 `--create-on-absent` 以外的其他flags，使用方法与部署环境创建过程相同。

保存部署环境相关操作

创建了部署环境后，如果需要对配置进行保存操作，有两种方式：

方式1：保存到本地生成Wpfile文件。

```
$ wpctl env:dump
Saving Wpfile to /Users/***/Documents/webp/webp-cli-demo/Wpfile
```

方式2：直接保存成模板。

```
$ wpctl env:save --name demo-template
[OK] Start to create configTemplate name:demo-template (1s)
Template demo-template created, Id is wct-***
```

8.更新环境

在应用内创建了部署环境后，可以通过CLI来更改部署环境的配置。

您可以更新部署环境的配置和部署包版本：

- 您可以使用 `apply` 命令来更新Wpfile的配置，例如执行命令 `apply your_Wpfile` ，将读取Wpfile的配置项更新到线上环境中。
- 更新部署包的方式如下所示：
  - Wpfile中包含 `application.package: url` 配置时，将会自动更新环境部署包。
  - 工作目录中有指定的默认应用和环境，执行 `apply` 命令时可以指定 `--package` 和 `--label` ，执行 `--package-id` 可以使用已有的部署包来更新部署包。

    > **注意** 部署包的大小不能超过5G。

  - 可以在工作目录中指定的默认应用和环境，执行 `wpctl env:deploy` 命令可更新部署包版本。

执行 `wpctl env:apply --help` 会提示您怎样使用CLI来增加Flag和参数，同时还会显示您目前的工作目录。如果您未执行 `wpctl init` 来指定默认应用和环境，您可以执行 `apply` 来创建应用和部署环境。

```
$ wpctl env:apply --help
...
`wpctl env:apply
 --app target_app (--create-on-absent to create when it doesn't exist)
 --env target_env (--create-on-absent to create when it doesn't exist)
 --package https://*** --label v1 (or just using existed package with --package-id)`
`wpctl env:apply --template template***
 --app app_template_belongs_to
 --env new_env_name --create-on-absent`
```

当用户工作目录下有默认应用和默认环境时，执行 `wpctl env:apply --help` 会提示您当前的默认应用和部署环境，您可以通过执行 `--package` 重新部署应用，或在Wpfile中更新部署环境的配置。

```
$ wpctl env:apply --help
...
`wpctl env:apply
 (application benchmark-consumer loaded)
 (environment consumer loaded)
 --package https://*** --label v1 (or just using existed package with --package-id)`
 (Wpfile in this directory will be read as settings)
`wpctl env:apply --template template***
 --env new_env_name --create-on-absent`
```

9.切换应用和环境

当您有多个应用和环境时，您可以根据需要切换当前的默认环境和应用。

```
$ wpctl env:list
1) Web3-env
2) Web2-env
```

`wpctl env:use + 环境名称/id` ，可以切换默认环境： `wpctl env:use Webxfile3-env` 。

```
$ wpctl env:use Web***
$ wpctl env:info
EnvName:  Web***
EnvId:   we-***
AppId:   wa-***
AppName:  Webxfile2
Status:  RUNNING
Stack:   Aliyun Linux 2.1903 running Tomcat 8.5 Java 8
CreateUser:  ***
CreateTime:  2019-04-20 03:34:21 PM
UpdateUser:  ***
UpdateTime:  2019-04-20 03:34:21 PM
```

`wpctl app:use + 应用名称/id` ，可以切换默认应用。

```
$ wpctl app:list
1) app2
2) app1
$ webxctl app:use app2
Application app2 setted, no environment chosen
```

或者一步到位， `env:use` 时使用 `--app` 一起切换应用和环境

```
$ wpctl env:use Webxfile3-env --app Webxfile2
...
```

10.扩缩环境

扩缩环境即为通过改变实例的数量来增加或减少应用的计算容量。您可以在应用的实例负载过高时在环境中添加新实例，并在不再需要这些实例时终止它们。

用户可以根据需要对环境进行扩缩容动作，使用scale + 目的实例数，当用户实例数从1更新到n时，默认会开通一个slb，但是从n到m时，则不会再对slb进行开通处理，从n到0时，并不会删除slb。

```
wpctl env:scale 2
[OK] Update environment name Web*** (1s)
...
```

使用env:info 即可看到扩缩后的实例数

11.删除环境和应用

当应用不需要再运行时，您可以删除应用下面的环境和删除应用。

删除环境

在对环境进行删除操作时需要先终止环境。

```
$ wpctl env:terminate
[OK] Terminate environment name demo-test-env (1s)
-- Check events --
2019-05-05 08:16:36 PM: Start to terminate deployment, change Id is wc-5cced4243b331c3b5592551a
2019-05-05 08:16:37 PM: Start to terminate application on instance i-wz98cfp8hf0jct9pyfxy
2019-05-05 08:16:43 PM: Success to terminate application on instance i-wz98cfp8hf0jct9pyfxy
2019-05-05 08:16:44 PM: Successfully deleted monitor group
2019-05-05 08:16:44 PM: Success to terminate platform services
2019-05-05 08:16:46 PM: Start to reduce 1 ECS instances
2019-05-05 08:17:06 PM: Successfully deleted ECS instance i-wz98cfp8hf0jct9pyfxy
2019-05-05 08:17:08 PM: Untied VSwitches vsw-wz9bflpiecxeotsgetdfl successfully
2019-05-05 08:17:08 PM: Untied VPC vpc-1kgkcouha successfully
2019-05-05 08:17:08 PM: webx.wam.change.terminate.success
Change finished, apply success
$ wpctl env:delete we-5cced1e63b331c3b55925475
 (1s)
[OK] Delete environment name demo-test-env (1s)
$ wpctl env:info
No env in this app:demo-test
```

删除应用

应用删除前需要确定应用下的所有环境都已经处于释放状态。

```
$ wpctl app:delete --help
delete the app
Usage:
  wpctl app:delete Id/name
Sample:
  wpctl app:delete app-demo
Use `app:delete --help` for more information.
```

12.查看环境和应用

在创建完应用和部署环境后，您可以执行下面操作查看应用和部署环境的配置信息。

查看应用列表

您可以通过下面命令查看当前账号下的所有应用列表，可以查看应用的Name，ID，Category和关联的环境状况。

```
$ wpctl app:list
[OK] Query applications (2s)
+----+-------------------+---------------------------+-------------+----------+----------------+--------------+
| NO |     NAME     |        ID        | DESCRIPTION | CATEGORY | ENV(S) RUNNING | ENV(S) TOTAL |
+----+-------------------+---------------------------+-------------+----------+----------------+--------------+
| 1 | ****************** | wa-************************ |        | Java   |        0 |       1 |
| 2 | ****************** | wa-************************ |        | Java   |        0 |       3 |
| 3 | ****************** | wa-************************ |        | Tomcat |        0 |       1 |
+----+-------------------+---------------------------+-------------+----------+----------------+--------------+
```

查看应用的部署环境列表

您可以通过下面命令查看某一应用下面的环境列表。

```
$ wpctl env:list --app ***
[OK] Query environments (1s)
Environment(s) in application ***
+----+----------+---------------------------+-------------+----------------------------+---------------+------+----------+
| NO |  NAME  |        ID        | DESCRIPTION |      STACK ID      | PACKAGE LABEL | TYPE | STATUS |
+----+----------+---------------------------+-------------+----------------------------+---------------+------+----------+
| 1 | ***    | we-************************ |        | ws-************************ |  ********** | web | ABNORMAL |
+----+----------+---------------------------+-------------+----------------------------+---------------+------+----------+
```

查看事件列表

您可以随时查询环境发生的所有事件，执行 `e/b` 可切换到第1和最后1页，执行 `r` 可刷新事件列表。

```
wpctl env:events
-- Check events, ctl + c to end --
2019-04-19 09:09:14 PM: 应用环境变更，变更Id为wc-****89c9
2019-04-19 09:09:14 PM: 环境we-****8981成功绑定vpc:vpc-****ouha
2019-04-19 09:09:15 PM: 环境we-***8981绑定vswitch vsw-****tdfl成功
2019-04-19 09:09:23 PM: 开始创建实例，数量1
```

查看默认环境信息

使用 `wpctl env:info` 可查看当前目录下的环境信息。

```
$ wpctl env:info
 (1s)
EnvName:  demo-test-env
EnvId:   we-****
AppId:   wa-****
AppName:  demo-test
Status:   RUNNING
Stack:   Aliyun Linux 2.1903 - Java 8 - Tomcat 8.5
CreateUser:   *****
CreateTime:   2019-05-05 08:25:29 PM
UpdateUser:   *****
UpdateTime:   2019-05-05 08:25:30 PM
Resources:
You can access the application via http://*****
```

查看部署包版本信息

使用 `wpctl pkg:list` 可查看应用的部署包的版本信息。

```
$ wpctl pkg:list
[OK] Query pkgVersion appId:wa-******** (1s)
Package version(s) in application benchmark-provider
+----+-------------------------+------------+-------------+-------------------------------------------------------------------------+
| NO |        ID               |  LABEL     | DESCRIPTION |                    PACKAGESOURCE                                        |
+----+-------------------------+------------+-------------+-------------------------------------------------------------------------+
| 1 | wp-****2aeb | 1558353894 |          | resources/benchmark-provider/versions/1558353894/service-provider.ja
r |
| 2 | wp-****897b | 1558353887 |          | resources/benchmark-provider/versions/1558353887/service-provider.ja
r |
| 3 | wp-****5e0b | 1558353867 |          | resources/benchmark-provider/versions/1558353867/service-provider.ja
r |
+----+-----------------------------+------------+-------------+-------------------------------------------------------------------------+
```

查看环境的健康状态

使用 `wpctl env:health` 可查看环境的健康状态。

```
$ wpctl env:health
[OK] Describe environment health name demo-test-env (1s)
EnvId:we-5cced6393b331c3b55925679   EnvName:demo-test-env
+----+------------------------+-------------+-----------+------------------+
| NO |     INSTANCEID         | AGENTSTATUS | APPSTATUS | DISCONNECTEDTIME |
+----+------------------------+-------------+-----------+------------------+
| 1 | i-wz91tmwedluum4mldj9z | CONNECTED   | HEALTHY   |                0 |
| 2 | i-wz91tmwedluum4mldj9x | CONNECTED   | HEALTHY   |                0 |
| 3 | i-wz91tmwedluum4mldj9y | CONNECTED   | HEALTHY   |                0 |
| 4 | i-wz91tmwedluum4mldja0 | CONNECTED   | HEALTHY   |                0 |
+----+------------------------+-------------+-----------+------------------+
```

查看并访问环境URL

执行 `wpctl env:open env Name/ID` 可以查看并访问此环境的URL。

> **说明** 执行 `wpctl env:open env Name/ID` 命令时需要系统内已安装浏览器。应用绑定了多个SLB或应用内有多个ECS实例时，可以选择需要打开的网址。

```
$ webxctl open
+----+----------------+
| NO |     IP     |
+----+----------------+
| 1 | 39.108.253.169 |
| 2 | 39.108.248.175 |
| 3 | 39.108.255.172 |
+----+----------------+
Please choose IP to open[default:1]:
```

打开Web+控制台

在系统内有安装浏览器的情况下，执行 `$ wpctl env:console` 将打开Web+的控制台。

> **说明** 执行 `wpctl env:open env Name/ID` 命令时需要系统内已安装浏览器。应用绑定了多个SLB或应用内有多个ECS实例时，可以选择需要打开的网址。

```
$ wpctl env:console
```

13.环境配置模板

任何一个环境都可以保存为一个模板，用户可以通过这个模板随时拉起一个相同应用下的新环境。

- template:list：列出当前应用下的所有模板。
- template:delete：删除指定的模板。
- template:launch：由模板拉起一个应用。

```
$ wpctl template:launch demo-test
[OK] Start to query configTemplate, appId:wa-5cd3f66e9bf7932a32814ce5 (1s)
Enter env name to create: demo-test-1
[OK] (1s)
create env from template(id:wct-5cd4317edc509c5f154e1e36)
[OK] Create environment name demo-test-1 (2s)
Saving config to webxconfig:/home/jungle/test/.webx/webxconfig.yaml
[-- Check events --]
2019-05-09 09:57:07 PM: Start to apply deployment change, change Id is wc-5cd431b2dc509c5f154e1ec2
2019-05-09 09:57:08 PM: Successfully binded VPC(vpc-1kgkcouha)
2019-05-09 09:57:09 PM: Successfully binded VSwitch(vsw-wz9bflpiecxeotsgetdfl)
```

14.收集报错信息

当操作过程中出现未知错误，请执行 `wpctl collect` 收集报错信息。

1. 确定操作前进行过doctor诊断服务开通和依赖开通和授权。
2. 检查报错信息，是否参数输入问题或者网络问题。
3. 保持工作目录不要切换和操作，执行 `wpctl collect` 收集信息，将当前目录下的zip包发送给相关人员。
4. 如果在出错之后还进行了未知操作，执行 `wpctl collect --all` 收集全部信息，请注意查看操作时间段，操作命令，报错信息，出错信息中requestID。

```
$ wpctl collect
information collected, file path is /Users/***/Documents/webp/webp-cli-test/collect.zip
$ ls
collect.zip
```

15.Host操作

Host操作目前只支持SSH登录目标实例，exec在目标机器上执行命令。

- wpctl ssh：需要用户在环境所在的安全组里有一台跳板机，用户可以选择其中一台ECS，或者单独创建一台broker。

```
$ wpctl configure
Configuring profile 'default' in '' authenticate mode...
Access Key Id [*************zGQ]:
Access Key Secret [****************************N91]:
Default Region Id [cn-shenzhen]:
Default Output Format [json]: json (Only support json))
Default Language [zh|en] zh:
Configuring broker options...
Broker Address [120.79.86.15]: 47.112.30.3
Broker Port [22]:
Broker User [root]:
Broker Password [********4]:
Broker Identity File []:
Saving profile[default] ...Done.
Configure Done!!!
$ webxctl ssh demo-test-1
[\] Finding servers of demo-test-1] (1s)
[OK] Finding servers of demo-test-1 (2s)
Found 2 servers:
+----+------------------------+----------------+---------------+---------------+---------+
| NO |        ID        |  PRIVATE IP  |  HOSTNAME   |  PUBLIC IP  | STATUS |
+----+------------------------+----------------+---------------+---------------+---------+
| 1 | i-wz9i4zv4ov7fktngxefv | 172.18.207.215 | document-test | 47.112.30.3  | Running |
| 2 | i-wz9i6yrcr8jgkurozbzi | 172.18.207.217 | document-test | 39.108.254.90 | Running |
+----+------------------------+----------------+---------------+---------------+---------+
Please choose server to operate [1]: 2
[OK] Testing connection bewteen broker and server (0s)
[OK] Adding auth key to 172.18.207.217 through CA (4s)
[OK] Start forwarding on: [ 127.0.0.1:58424 -> 172.18.207.217:22 ] (0s)
****************************************
* Logged on 172.18.207.217 as root    *
****************************************
Last login: Tue Apr 23 15:14:52 2019 from 106.11.235.191
Welcome to Alibaba Cloud Elastic Compute Service !
[root@iZwz9i6yrcr8jgkurozbziZ ~]#
```

- wpctl exec：选择环境中的机器执行命令，多个实例存在时需要选择机器，此功能需要安装云助手。

```
$ wpctl env:exec demo-test-1 'echo hello'
[OK] (1s)ng servers of demo-test-1
[OK] Finding servers of demo-test-1 (2s)
Found 2 servers:
+----+-----------------------+---------------+---------------+---------------+---------+
| NO |        ID             |  PRIVATE IP   |   HOSTNAME    |   PUBLIC IP   | STATUS  |
+----+-----------------------+---------------+---------------+---------------+---------+
|  1 | i-wz9i4zv4ov7fktngxefv | 172.18.207.215 | document-test | 47.112.30.3   | Running |
|  2 | i-wz9i6yrcr8jgkurozbzi | 172.18.207.217 | document-test | 39.108.254.90 | Running |
+----+-----------------------+---------------+---------------+---------------+---------+
Please choose server to operate [default:1] [0 for all instances]:
[OK] Checking Cloud Assistant status of 1 servers (1s)
[OK] Executing Commands on 1 servers (3s)
>>>>> 172.18.207.215 [i-wz9i4zv4ov7fktngxefv] status:Finished >>>>>
hello
<<<<< -----
```