

Alibaba Cloud Web App Service

CLI

Issue: 20200319

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.









1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent

ial, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document

.

Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands.	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
{} or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Contents

Legal disclaimer.....	I
Document conventions.....	I
1 Install and upgrade the Web+ CLI (wpctl).....	1
2 Use CLI commands.....	3
3 Configure the deployment environment configuration file Wpfile.....	18
4 Automatically complete commands.....	45
5 Initialize the CLI.....	46
6 Create an application.....	49
7 Create an environment.....	50
8 Update an environment.....	55
9 Switch to an application and an environment.....	57
10 Resize an environment.....	58
11 Delete an application and its environments.....	59
12 Query applications and environments.....	60
13 Use an environment configuration template.....	64
14 Collect error information.....	65
15 Perform operations on an ECS instance.....	66

1 Install and upgrade the Web+ CLI (wpctl)

As the command-line interface (CLI) of Web App Service (Web+), `wpctl` is a client on which you can run commands to create, configure, and manage Web+ applications and deployment environments. You can use `wpctl` to replace the Web+ console for development and testing.

Install `wpctl` in Linux

Install `wpctl`

Run the following command:

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

Upgrade the CLI

If the CLI version you are using is earlier than the earliest version compatible with `wpctl`, the system displays the message `The CLI version is too old. Please upgrade the CLI when running wpctl. Run the wpctl upgrade command to upgrade the CLI.`

Uninstall `wpctl`

Run the `rm /usr/local/bin/wpctl` command.

Install `wpctl` in macOS

Install `wpctl`

Run the following command:

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

Upgrade the CLI

If the CLI version you are using is earlier than the earliest version compatible with `wpctl`, the system displays the message `The CLI version is too old. Please upgrade the CLI when running wpctl. Run the wpctl upgrade command to upgrade the CLI.`

Uninstall `wpctl`

Run the `rm /usr/local/bin/wpctl` command.

If the CLI version you are using is earlier than the earliest version compatible with wpctl, the system displays the message The CLI version is too old. Please upgrade the CLI **when running wpctl. Run the wpctl upgrade command to upgrade the CLI.**

2 Use CLI commands

As the command-line interface (CLI) of Web App Service (Web+), `wpctl` enables you to easily create and manage Web+ applications and deployment environments on a terminal. This topic describes the core CLI commands of Web+. You can view all CLI commands by running the `wpctl --help`, `wpctl help`, or `wpctl h` command on the terminal.

Application-related commands

Query applications

You can run this command to query applications under the current account.

```
Usage:
  wpctl app:list
```

```
Example:
  wpctl app:list
```

Switch to an application

After querying applications, you can run this command to switch to an application.

```
Usage:
  wpctl app:use ID/Name
```

```
Example:
  wpctl app:use app-demo
```

Parameter	Flag
ID/Name: the ID or name of the application to switch to.	None

Delete an application

Before deleting an application, you need to release all deployment environments of the application.

```
Usage:
  wpctl app:delete ID/Name
```

```
Example:
```

```
wpctl app:delete app-demo
```

Parameter	Flag
ID/Name: the ID or name of the application to delete.	None

Environment-related commands

apply command

You can run the `apply` command to create an application or deployment environment or update a deployment environment. You can run the `apply --help` command to view the information about the working directory.

```
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
```

You can directly run the `apply` command without specifying parameters. You can also use a `Wpfile` file or a `ZIP` file to specify parameters.

Flag	Description
<code>--package-id</code>	Specifies the version of the existing deployment package.
<code>--package</code> or <code>-p</code>	Specifies the deployment package.
<code>--label</code> or <code>-l</code>	Specifies the name of the deployment package.

Flag	Description
--template or -t	Specifies the ID of the deployment environment configuration template.
--type	Specifies the mode of the deployment environment configuration. Valid values: <ul style="list-style-type: none">• HighAvailability or HA• LowCost or LC
--app or -a	Specifies the application.
--env or -e	Specifies the deployment environment.
--category	Specifies the platform type of the application. Valid values: Tomcat or Java.

Flag	Description
--stack-id	<p>Specifies the technology stack ID of the application. Valid values:</p> <ul style="list-style-type: none"> • 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 or -q	<p>Specifies that the command is run quietly, without any process messages or command output. If you need to display the command output, use the <code>--json</code> flag.</p>
--json or -j	<p>Displays the command output in JSON format.</p>

Flag	Description
<code>--create-on-absent</code> or <code>-C</code>	Specifies that Web+ creates an application or deployment environment if the specified application or deployment environment does not exist. You must use this flag together with the <code>--app</code> or <code>--env</code> flag.

Initialize a deployment environment

You can run this command to initialize the working directory of a deployment environment, including the region, default application, and default deployment environment.

```
Usage:
  wpctl init
```

Query environments

You can run this command to query the deployment environments of the current application.

```
Usage:
  wpctl env:list [flags]

Flags:
  --app,-a set the application
  --help   print help

Example:
  wpctl env:list --app appName
```

Switch to a deployment environment

After querying the deployment environments of the current application, you can run this command to switch to a deployment environment.

```
Usage:
  wpctl env:use id/name

Flags:
  --app,-a set the application
  --help   print help
```

Parameter	Flag
id/name: the ID or name of the application to query.	<code>--app</code> or <code>-a</code> : specifies the ID or name of the application to which the target deployment environment belongs.

Display environment information

```
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

Example:
  wpctl env:info --env envName
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs. • --json or -j: displays information in JSON format.

Start a deployment environment

```
Usage:
  wpctl env:start [flags]

Flags:
  --app,-a set the application
  --env,-e set the env
  --help    print help

Example:
  wpctl env:start --env envId/name
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Stop a deployment environment

```
Usage:
  wpctl env:stop [flags]

Flags:
  --app,-a set the application
```



```

--env,-e set the env
--help   print help

Example:
wpctl env:stop --env envId/name
    
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Release a deployment environment

```

Usage:
wpctl env:terminate [flags]

Flags:
--app,-a set the application
--env,-e set the env
--help   print help

Example:
wbnx env:terminate
    
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Resize a deployment environment

```

Usage:
wpctl env:scale instanceNum [flags]

Flags:
--app,-a set the application
--env,-e set the env
--help   print help

Example:
    
```

```
wbnx env:scale 1
```

Parameter	Flag
instanceNum: the target number of Elastic Compute Service (ECS) instances . Valid values: 0 to 100.	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Delete a deployment environment

```
Usage:
  wpctl env:delete ID/Name [flags]

Flags:
  --app,-a set the application
  --help   print help

Example:
  wpctl env:delete env-demo
```

Parameter	Flag
ID/Name: the ID or name of the deployment environment to delete.	--app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Generate an environment configuration template

```
Usage:
  wpctl env:save [flags]

Flags:
  --name,-n set the name
  --app,-a  set the application
  --env,-e  set the env
  --help    print help

Example:
  wpctl env:save --name templateName
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs. • --name or -n: specifies the name of the template to be generated.

Download the deployment environment configuration

```
Usage:
  wpctl env:dump [flags]

Flags:
  --env,-e set the env
  --app,-a set the application
  --help   print help

Example:
  wpctl env:dump --env envName
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Check environment health

```
Usage:
  wpctl env:health [flags]

Flags:
  --env,-e set the env
  --app,-a set the application
  --help   print help

Example:
  wpctl env:health --env envName --app appName
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Monitor a deployment environment

```
Usage:
  wpctl env:top [flags]

Flags:
  --env,-e set the env
  --app,-a set the application
  --help   print help

Example:
```

```
wpctl env:top --env envName
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

View environment events

```
Usage:
  wpctl env:events [flags]

Flags:
  --env,-e    set the env
  --app,-a    set the application
  --change,-c set the changeId
  --help      print help

Example:
  wpctl env:events --env envName
```

Parameter	Flag
None	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs. • --change or -c: specifies the event of a change to be viewed. The ID of the change must be specified.

Update the environment deployment package

```
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

Example:
```

```
wpctl env:deploy *.war --label v2
```

Parameter	Flag
<p>package: the deployment package used for the update.</p>	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs. • --label or -l: specifies the name of the deployment package to upload.

Access a deployment environment

```
Usage:
  wpctl env:open [flags]

Flags:
  --app,-a set the application
  --env,-e set the env
  --help   print help

Example:
  wpctl env:open --env envId/name
```

Parameter	Flag
<p>None</p>	<ul style="list-style-type: none"> • --env or -e: specifies the ID or name of the target deployment environment. • --app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Instance-related commands

Log on to an instance

You can run this command to log on to an ECS instance in a deployment environment through Secure Shell (SSH).

```
Usage:
  wpctl env:ssh envName/ID [flags]

Flags:
  --app,-a set the application
  --help   print help

Example:
```

```
wpctl env:ssh envName/ID --app appName/ID
```

Parameter	Flag
envName/ID: the ID or name of the deployment environment to log on to.	--app or -a: specifies the ID or name of the application to which the target deployment environment belongs.

Run commands on an instance in a deployment environment

```
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

Example:
wpctl env:exec envName 'echo hello'
```

Template-related commands

Query templates

```
Usage:
wpctl template:list [flags]

Flags:
--app,-a set the application
--help   print hel

Example:
wpctl template:list
```

Parameter	Flag
None	--app or -a: specifies the ID or name of the application to query.

Delete a template

```
Usage:
wpctl template:delete id/name [flags]

Flags:
--app,-a set the application
--help   print help

Example:
```

```
wpctl template:delete templateName/ID
```

Parameter	Flag
id/name: the ID or name of the template to delete.	--app or -a: specifies the ID or name of the application to which the target template belongs.

Start a template

```
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

Example:
  wpctl template:launch templateName/id
```

Parameter	Flag
templateName/id: the ID or name of the template to start.	<ul style="list-style-type: none"> • --env or -e: specifies the name of the deployment environment to create and start by using the template. • --app or -a: specifies the ID or name of the application to which the target template belongs. • --create-on-absent or -C: creates an application or deployment environment if the specified application or deployment environment does not exist. You must use this flag together with the --app or --env flag.

Other commands

Query deployment packages

```
Usage:
  wpctl pkg:list [flags]

Flags:
  --app,-a set the application
  --help    print help

Example:
```

```
wpctl pkg:list --app appName
```

Parameter	Flag
None	--app or -a: specifies the ID or name of the application to query.

Configure an account

```
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/RamRoleArn mode
  --access-key-secret use `--access-key-secret <AccessKeySecret>` to
assign AccessKeySecret
  --help              print help
```

Automatically complete commands

When you run a script to install the CLI on a terminal, you enable the Tab key of the terminal to automatically complete commands. However, if you switch to another terminal, this feature becomes invalid. In this case, you can run this command to automatically complete commands.

```
Usage:
  wpctl auto-completion [flags]

Flags:
  --uninstall  uninstall auto completion
  --help      print help
```

Parameter	Flag
None	--uninstall: disables the automatic command completion feature.

Query the CLI version

```
Usage:
  wpctl version

Example:
```



```
wpctl version
```

Update the CLI version

Usage:
wpctl upgrade

Example:
wpctl upgrade

Collect CLI information

Usage:
wpctl collect [flags]

Flags:
--all collect all log
--help print help

Example:
wpctl collect

Parameter	Flag
None	--all: collects all log information.

Verify the activation of Web+ and associated services, and required permissions

Usage:
wpctl doctor

Example:
wpctl doctor

3 Configure the deployment environment configuration file Wpfile

Deployment environment configuration items contain all the information required to create a deployment environment. You can duplicate configuration items, excluding resources exclusive to the deployment environment. For example, you cannot duplicate Elastic Compute Service (ECS) instances.

Platform

Path	Parameter	Configuration item	Default value	Valid values	Description
proxy	type	Reverse proxy type	nginx	nginx and apache	The type of the reverse proxy to use. After the reverse proxy is enabled, the system listens on port 80 of your application. Ensure that this port is not occupied by other programs. The reverse proxy forwards received HTTP requests to the service port of the application.
proxy.nginx	version	Nginx version	1.14.2	1.14.2	The Nginx version to use. This parameter takes effect only when the type parameter is set to nginx.
proxy.apache	version	Apache version	2.4.6	2.4.6	The Apache version to use. This parameter takes effect only when the type parameter is set to apache.

Resources

resources.network: Network configuration

Parameter	Configurat ion item	Default value	Valid values	Description
vpcOption	Network options	{"vpcId ":""," vSwitches ":[]}		The Virtual Private Cloud (VPC) and VSwitch to use in the deployment environment. If multiple VSwitches are selected, Web App Service (Web+) allocates Elastic Compute Service (ECS) instances to the selected VSwitches in a balanced manner. You can create a VPC and a VSwitch in the VPC console.

resources.ecs.autoScaling: ECS configuration

Parameter	Configurat ion item	Default value	Valid values	Description
instanceType	Instance type	["ecs.g5. large"]		The type of the ECS instance to use in the deployment environment. Web+ creates ECS instances of the selected instance type. You can select a maximum of 10 instance types. When resources are insufficient for the selected instance types, Web+ tries to create ECS instances according to the sequence of selected instance types.
instanceNum	Instance quantity	1	[0, 100]	The number of ECS instances to use. Based on this parameter, Web+ resizes the deployment environment.
securityGroupIds	Security group	[]		The security group of the ECS instance. You can select a maximum of five security groups.
keyPairName	Key pair			The key pair to use for Secure Shell (SSH) logon. For more information about how to use the SSH key pair, see Use an SSH key pair.
systemDiskSize	System disk size			The size of the system disk. Unit: GB.

Parameter	Configurat ion item	Default value	Valid values	Description
enableInternet	Whether to enable the public IP address	true	true and false	Specifies whether to enable the public IP address. If the public IP address is enabled, all ECS instances in the deployment environment can be accessed from the public network. Pay attention to the security configuration. The services provided by ECS instances on the public network may incur charges. For more information, see the billing of Internet bandwidth.
instanceName	Instance name			The name of the ECS instance to create.
scalingPolicy	Instance deallocation policy	recycle	release and recycle	The deallocation policy to use for the ECS instance that is shut down . If this parameter is set to release, the system releases all resources of the ECS instance. If this parameter is set to recycle, the system retains the disks of the ECS instance, which may incur charges.
userData	Custom instance data			The custom data on the ECS instance. The data must be Base64-encoded. The maximum size of the raw data is 16 KB.
systemDiskCategory	System disk type	cloud_efficiency	cloud, cloud_efficiency, cloud_ssd, and ephemeral_ssd	The type of the system disk. A value of cloud indicates the basic disk. A value of cloud_efficiency indicates the ultra disk. A value of cloud_ssd indicates the SSD. A value of ephemeral_ssd indicates the local SSD. When Generation I and non I/O-optimized instance types are selected, the default value of this parameter is cloud. Otherwise, the default value is cloud_efficiency .

Parameter	Configuration item	Default value	Valid values	Description
multiAzPolicy	Multi-zone scaling policy	BALANCE	PRIORITY and BALANCE	The scaling policy for ECS instances in a multi-zone scaling group. If this parameter is set to PRIORITY, the system scales out or in ECS instances based on your VSwitches specified by the VSwitchIds.N parameter. This policy allows the system to create ECS instances in the zone of the secondary VSwitch when the system fails to create ECS instances in the zone of the primary VSwitch. If this parameter is set to BALANCE, the system allocates ECS instances in a balanced manner among multiple zones specified by the scaling group. If ECS instances become imbalanced among multiple zones due to a lack of instance resources, you can call the RebalanceInstances operation of Auto Scaling to balance the allocation of resources.
internetChargeType	Internet billing type	PayByTraffic	PayByBandwidth and PayByTraffic	The billing method on the public network. If this parameter is set to PayByBandwidth, the network service is billed by fixed bandwidth. In this case, the fixed bandwidth is the maximum available outbound bandwidth specified by the internetMaxBandwidthOut parameter. If this parameter is set to PayByTraffic, the network service is billed by traffic usage. In this case, the internetMaxBandwidthOut parameter specifies only the upper limit of available bandwidth.

Parameter	Configurat ion item	Default value	Valid values	Description
internetMaxBandwidthIn	Maximum inbound bandwidth from the public network	100	[1, 200]	The maximum inbound bandwidth from the public network. Unit: Mbit/s. If you do not specify this parameter, it is set to 200 by default. This parameter is not used for billing because the inbound traffic to ECS instances is free of charge.
internetMaxBandwidthOut	Maximum outbound bandwidth to the public network	50	[0, 100]	The maximum outbound bandwidth to the public network. Unit: Mbit/s. If the internetChargeType parameter is set to PayByBandwidth but you do not specify this parameter, it is set to 0 by default. If the internetChargeType parameter is set to PayByTraffic but you do not specify this parameter, the system reports an error.
ioOptimized	I/O optimized instance	true	true and false	Specifies whether the ECS instance is an I/O optimized instance.

resources.slb.internet: Public network SLB configuration

Parameter	Configuration item	Default value	Valid values	Description
enable	Whether to enable public network SLB	true	true and false	Specifies whether to enable public network SLB. If multiple ECS instances exist in the deployment environment, you can create or configure an SLB instance in the SLB console for load balancing.

Paramet	Configur	Default	Valid	Descriptio
	ion	value	values	n
	item			
slbId	SLB instance ID			The ID of the public network SLB instance . You can use an existing SLB instance or enable Web + to purchase an SLB instance .
listenerPort	SLB listener port	80	[1, 65535]	The listener port of the public network SLB instance . This port enables you to access the application from the public network .

Parameter	Configuration item	Default value	Valid values	Description
protocol	SLB protocol	http	http and tcp	The protocol type of the public network SLB instance.
forwarding Rule	SLB routing method			The routing method of the public network SLB instance.
bandwidth	Peak bandwidth on the listener port	-1	[-1, 5120]	The peak bandwidth on the listener port.

Parameter	Configuration item	Default value	Valid values	Description
loadBalancerSpec	SLB instance type	slb.s1.small	slb.s1.small, slb.s2.small, slb.s2.medium, slb.s3.small, slb.s3.medium, and slb.s3.large	The type of the public network SLB instance. The supported instance types vary with the region.

Parameter	Configuration item	Default value	Valid values	Description
internetChargeType	Billing method of the SLB instance	paybytraffic	paybybandwidth and paybytraffic	The billing method of the public network SLB instance. If this parameter is set to paybybandwidth, the SLB instance is billed by bandwidth. If this parameter is set to paybytraffic, the SLB instance is billed by traffic.
healthCheckUrl	URL for health check	N/A		

resources.slb.intranet: Internal network SLB configuration

Paramet	Configur	Default	Valid	Descriptio
	ion	value	values	n
	item			
enable	Whether to enable internal network SLB	true	true and false	Specifies whether to enable internal network SLB. If multiple ECS instances exist in the deployment environment, you can create or configure an SLB instance in the SLB console for load balancing.

Paramet	Configur ion item	Default value	Valid values	Descriptio n
slbId	SLB instance ID			<p>The ID of the internal network SLB instance. You can use an existing SLB instance or enable Web + to purchase an SLB instance.</p>

Parameter	Configuration item	Default value	Valid values	Description
listenerPort	SLB listener port	80	[1, 65535]	The listener port of the internal network SLB instance. This port enables you to access the application from the internal network.
protocol	SLB protocol	http	http and tcp	The protocol type of the internal network SLB instance.

Parameter	Configuration item	Default value	Valid values	Description
forwardingRule	SLB routing method			The routing method of the internal network SLB instance.
bandwidth	Peak bandwidth on the listener port	-1	[-1, 5120]	The peak bandwidth on the listener port.
bandwidth	Peak bandwidth on the listener port	-1	[-1, 5120]	The peak bandwidth on the listener port.

Parameter	Configuration item	Default value	Valid values	Description
loadBalancerSpec	SLB instance type	slb.s1.small	slb.s1.small, slb.s2.small, slb.s2.medium, slb.s3.small, slb.s3.medium, and slb.s3.large	The type of the internal network SLB instance. The supported instance types vary with the region.

Parameter	Configuration item	Default value	Valid values	Description
internetChargeType	Billing method of the SLB instance	paybytraffic	paybybandwidth and paybytraffic	The billing method of the internal network SLB instance. If this parameter is set to paybybandwidth, the SLB instance is billed by bandwidth. If this parameter is set to paybytraffic, the SLB instance is billed by traffic.

resources.rds: RDS configuration

Configurat ion item	Valid values	Default value	Descriptio n
Whether to enable RDS	true and false	false	Specifies whether to enable Relational Database Service (RDS).
Whether to use an existing RDS instance	true and false	false	Specifies whether to use an existing RDS instance .
RDS instance ID	Valid RDS instance ID	None	The ID of the RDS instance to use . To use an existing RDS instance , you must specify this parameter .

Configurat ion item	Valid values	Default value	Descriptio n
ZoneId	Valid zone ID	None	The ID of the zone where the RDS instance resides . You can call the DescribeRe gions operation of RDS to query available zones.
VSwitches list	VSwitch ID	None	The ID of the VSwitch in the zone where the RDS instance resides .
Database type	MySQL, SQLServer , PostgreSQL , PPAS , and MariaDB	None	The database type of the RDS instance .

Configuration item	Valid values	Default value	Description
Database version	MySQL: 5.5, 5.6, 5.7, and 8.0. SQL Server : 2008r2, 2012, 2012_ent_h a, 2012_std_h a, 2012_web , 2016_ent_h a, 2016_std_h a, 2016_web , and 2017_ent . PostgreSQL : 9.4 and 10.0. PPAS: 9.3 and 10.0. MariaDB : 10.3.	None	The database version of the RDS instance .

Configurat ion item	Valid values	Default value	Descriptio n
StorageType	local_ssd / ephemeral_ssd, cloud_ssd , and cloud_essd	None	<p>The storage type of the RDS instance. A value of local_ssd or ephemeral_ssd indicates the local SSD. A value of cloud_ssd indicates the SSD. A value of cloud_essd indicates the ESSD. Local SSDs are recommended.</p>

Configurat ion item	Valid values	Default value	Descriptio n
StorageSiz eapacity	Integer	100	The storage capacity of the RDS instance . Unit: GB.
Instance Class	Valid instance type	None	The type of the RDS instance . For more informatio n, see the instance types in the ApsaraDB for RDS documentat ion.
DatabaseNa me	A string	webplus	The database name of the RDS instance .

Configurat ion item	Valid values	Default value	Descriptio n
CharacterSet Name	MySQL or MariaDB : utf8 , gbk, latin1 , and utf8mb4 . SQL Server: Chinese_PR C_CI_AS , Chinese_PR C_CS_AS , SQL_Latin1 _General_C P1_CI_AS , SQL_Latin1 _General_C P1_CS_AS , and Chinese_PR C_BIN.	MySQL or MariaDB: utf8mb4. SQL Server: Chinese_PRC_CI_AS.	The character set of the RDS instance .
AccountName	A string	webplus	The database account name of the RDS instance .

Configurat ion item	Valid values	Default value	Descriptio n
AccountPas sword	A string	None	The database account password of the RDS instance .
Category	Valid values: <ul style="list-style-type: none"> • Basic : basic edition • HighAvaila bility : high availabili ty edition • AlwaysOn : cluster edition • Finance : financial edition (supported only in China) 	None	The family of the RDS instance .

Application

application.option: Service port

Parameter	Configuration item	Default value	Valid values	Description
port	Service port	8080	[1024, 65535]	The service port of the application. After the application is started, it uses this port to provide HTTP services. The port number ranges from 1024 to 65535.

application.healthCheck: Health check configuration

Parameter	Configurat ion item	Default value	Valid values	Description
type	Health check type	http	http and tcp	The protocol type to use for health check. Currently , TCP and HTTP are supported. Web+ periodically sends health check requests to the service port of the application to check whether the application is healthy.
retryCount	Number of health check retries	3	[1, 30]	The maximum number of health check retries. If the number of health check retries exceeds this upper limit, the health check fails.
intervalSe conds	Health check interval	3	[1, 60]	The interval between a failed health check and the next health check.
timeoutSec onds	Health check timeout duration	3	[1, 60]	The timeout duration of a health check.

application.commands: Commands

Parameter	Configuration item	Description
start	Start command	The command used to start the application.
stop	Stop command	The command used to stop the application.

hooks: Lifecycle hook

Parameter	Configuration item	Description
prestart	Pre-start command	The command to run before the application is started.
poststart	Post-start command	The command to run after the application is started.
prestop	Pre-stop command	The command to run before the application is stopped.
poststop	Post-stop command	The command to run after the application is stopped.
postinit	Post-initialization command	The command to run after the application is initialized.

application.jvmOpts: JVM configuration

Parameter	Configuration item	Description
value	JVM parameters	The Java virtual machine (JVM) parameters used to start the application.

application.environmentVariables: Environment variables

Parameter	Configuration item	Description
value	Environment variables	The environment variables used to start the application.

4 Automatically complete commands

When you run a command to install the command-line interface (CLI) on a terminal, you enable the Tab key of the terminal to automatically complete commands. However, if you switch to another terminal, this feature becomes invalid. In this case, we recommend that you run the auto-completion command to automatically complete commands. After running this command, you must run the source command on a configuration file.

Syntax

auto-completion

Command output example and description

```
$ 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
auto-completion
doctor            env:dump         env:info          env:open
```

Description

The configuration file may be in `.bashrc`, `.bash_profile`, `.bash_login`, or `.profile` format. You must run the source command on a configuration file that contains `complete -C /usr/local/bin/wpctl wpctl`.

5 Initialize the CLI

Before using the command-line interface (CLI), you must configure and check global information. For example, run the `wpctl configure` command to configure global information such as the AccessKey ID and AccessKey secret of your account and the language, run the `wpctl doctor` command to verify the permissions of your account, and run the `wpctl init` command to initialize the default region, application, and environment information in the current working directory.

Step 1: Configure global information

Run the `wpctl configure` command to configure global information such as the AccessKey ID and AccessKey secret of your account and the language.

Method 1: Run the following command to configure global information and create a profile named `test`. For more information about the Region ID, see [#unique_8](#).

```
$ 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

Method 2: Run the following command with full configuration:

```
$ wpctl configure -p test1 --access-key-id yourAk --access-key-secret
yourSk --region cn-shenzhen --language en
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

Method 3: Configure the following environment variables:

```
ALICLOUD_ACCESS_KEY
ALICLOUD_SECRET_KEY
```

```
ALICLOUD_REGION
```

Switch the profile and modify parameters such as Region.

```
$ 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

Step 2: Verify the permissions of your account

Run the `wpctl doctor` command to verify the permissions of the current account.

Check in sequence whether Web App Service (Web+) is activated, whether associated services are activated, and whether permissions to access your Alibaba Cloud resources are granted to roles related to Web+. You can use Web+ features only after your account passes the verification.

```
$ wpctl doctor
[OK] Describe aliyun authority (1s)
[Products related checked]
```

NO	NAME	TITLE	DESC
1	ESS	Auto Scaling	Elastic scaling (Auto ...
2	OSS	Object Storage Service	Massive, secure, ...

```
[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]
```

Step 3: Select an application and a deployment environment

Run the `wpctl init` command to initialize the current working directory.

If no application is available in the selected region, you are prompted to create an application. The CLI displays all interactions in green. Generally, you only need to select a number or enter information as prompted.

```
$ 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 succeeded]
```

If an application already exists in the selected region, select the default deployment environment. The system saves the initialization result in the `.webplus` folder of the working directory.

```
$ 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 Create an application

You can create an application in one of the following ways:

- **Run the `wpctl init` command:**

```
$ 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
...
```

- **Run the `webxctl apply` command: Specify an application and a deployment environment that do not exist. Add the `--create-on-absent` flag to enable the system to create the nonexistent application and deployment environment.**

```
wpctl env:apply
--app target_app (this application does not exist)
--env target_env (this environment does not exist)
--package https://*** --label v1 (or just using existed package
with --pkgId)
--create-on-absent (ensure that the system creates the specified
application and environment)
```

7 Create an environment

Generally, you must create an application before creating an environment. In special cases, you can create an application together with an environment. For more information, see the `webxctl apply` command for creating an application.

Overview

Before creating an environment, familiarize yourself with the working directory and the `Wpfile` file.

Working directory: sets the default application and default environment for all operations in the current directory. For example, you can run the `wpctl list` command to read the default application in the current working directory, and list all the environments of the application. You can also run the `wpctl info` command to view the information about the default application and default environment in the current working directory.

Wpfile: stores the configuration information for creating or updating an environment. The `Wpfile` file and the `--type` flag in the `apply` command are mutually exclusive. For more information about the configuration items in the `Wpfile` file, see [Configure the deployment environment configuration file Wpfile](#).

You can run the `apply` command in one of the following ways:

- [Run the `wpctl apply` command](#)
- [Run the `wpctl apply your_Wpfile` command](#)

Run the `wpctl apply` command

After you run the `wpctl apply` command, the system reads the settings of the current working directory. If the `Wpfile` file exists in the current working directory, the system reads the configuration items of the `Wpfile` file and uses them as the settings of the `apply` command. The name of the `Wpfile` file cannot be changed. For more information, see [Use flags to control the `apply` command](#) in this topic.

```
$ pwd
/Users/***/Documents/webp/webp-cli-demo
$ ls
<Empty directory>
$ wpctl info // Check whether the default environment is specified in
the current working directory.
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)
...
```

Run the `wpctl apply your_Wpfile` command

You can specify a Wpfile file, the name of which can be changed, such as `jenkins` or `zookeeper`. When you run the `wpctl apply your_Wpfile` command, the system ignores the settings of the working directory and reads the configuration items of the specified Wpfile file. In this case, you can also specify flags to create or update an environment. For more information, see [Use flags to control the apply command](#) in this topic. The methods for specifying your Wpfile file are as follows:

- **file://**
- **https://**
- **http://**
- **Local file path, which can be a relative path or absolute path**

```
$ 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 |          VALUE          | PATH |          NAME          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|  1 | platform                |      | category              |
Tomcat |
+-----+-----+-----+-----+
|  2 | platform                |      | stackName             |
Aliyun Linux 2.1903 - Java 8 - |
Tomcat | ...                      |      |
Tomcat 8.5 |
+-----+-----+-----+-----+
...
[OK] Create environment name your_env (2s)
...
```

Use flags to control the apply command

You can use the following three factors to control the action of the apply command:

- **`--app` and `--env`: specify an application and an environment, respectively.**
- **Default application and default environment in the working directory.**

- **--create-on-absent: creates an application or environment if the object specified by the --app or --env flag does not exist.**

Table 7-1: Control the action on the environment

No.	--env	Default environment in the working directory	--create-on-absent	Action
1	Used	N/A	Used	Updates the environment specified by the --env flag, or creates an environment if the specified one does not exist.
2	Used	N/A	N/A	Updates the environment specified by the --env flag, or reports an error if the specified environment does not exist.
3	Not used	Used	N/A	Updates the default environment.

No.	--env	Default environment in the working directory	--create-on-absent	Action
4	Not used	Not used	N/A	Reports an error.

Table 7-2: Control the action on the application

No.	--app	Default application in the working directory	--create-on-absent	Action
1	Used	N/A	Used	Performs operations on the application specified by the --app flag, or creates an application if the specified one does not exist.
2	Used	N/A	Not used	Performs operations on the application specified by the --app flag, or reports an error if the specified application does not exist.
3	Not used	Used	N/A	Performs operations on the default application.
4	Not used	Not used	N/A	Reports an error.

Create an environment based on a configuration template

You can create an environment for an application based on a saved configuration template of the application. When running the apply command, use the `--template` flag to specify the template ID. Then, the system ignores all flags other than `--env`, `--app`, and `--create-on-absent`.

Save the environment configuration

After creating an environment, you can save the configuration in one of the following ways:

Method 1: Save as a Wpfile file on your local computer.

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

Method 2: Save as a configuration template.

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

8 Update an environment

After creating an environment for an application, you can update the environment.

You can update the environment configuration as follows:

- When you run the `apply` command and the `Wpfile` file exists in the working directory, or when you run the `apply your_Wpfile` command, the system reads the configuration items of the `Wpfile` file and updates them to the online environment.

You can update the deployment package version in one of the following ways:

- If the preceding `Wpfile` file contains the configuration item `application.package : url`, the system automatically updates the environment deployment package.
- If the default application and default environment are specified in the working directory, you can specify the `--package` and `--label` flags when running the `apply` command, or directly use the `--package-id` flag to specify a deployment package version to update the environment deployment package. Currently, the package size cannot exceed 5 GB.
- If the default application and default environment are specified in the working directory, you can run the `wpctl env:deploy` command to update the deployment package.

To know how to run the `apply` command, you can run the `wpctl env:apply --help` command. The command-line interface (CLI) will inform you of the information about the working directory and how to add flags and set parameters for the `apply` command. For example, if you have not run the `wpctl init` command to specify the default application and default environment in the working directory, you can run the `apply --help` command. The CLI will prompt you that you can run the `apply` command to create an application or environment. When creating an application or environment, you can set relevant attributes by using flags such as `--category`.

```
$ 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`
```

If the default application and default environment are specified in your working directory, you can run the `apply --help` command. The CLI will prompt you that the default application and default environment are loaded. In this case, you can use the `--package` flag to update the environment deployment package, or use the `Wpfile` file to update the environment configuration.

```
$ 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 Switch to an application and an environment

When multiple applications and environments are available, you can switch from the current application or environment to another application or environment.

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

Run the `wpctl env:use + Environment name or ID` command, for example, `wpctl env:use Webxfile3-env`, to switch from the current environment to the specified environment.

```
$ wpctl env:use Web***
$ wpctl env:info
EnvName: Web***
EnvId: we-***
AppId: wa-***
AppName: Webxfile2
Status: RUNNING
Stack: Alibaba Cloud 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
```

Run the `wpctl app:use + Application name or ID` command to switch from the current application to the specified application.

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

Alternatively, when running the `env:use` command, you can specify the `--app` flag to switch to the specified application and environment.

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

10 Resize an environment

To extend or shrink the computing capacity of an application, you can resize an environment of the application by scaling out or in Elastic Compute Service (ECS) instances. You can add ECS instances to the environment when the load is too high for the existing ECS instances of the application, and delete some ECS instances when they are no longer required.

You can run the `env:scale` command and specify the target number of ECS instances as required to resize the environment. When the number of ECS instances increases from 1 to *n*, a Server Load Balancer (SLB) instance is created by default. However, when the number continues to increase from *n* to *m*, no SLB instance is added. When the number decreases from *n* to 0, the SLB instance is not deleted.

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

You can run the `env:info` command to query the number of ECS instances after the deployment environment is resized.

11 Delete an application and its environments

If an application is no longer required, you can delete the environments of the application and delete the application.

Delete an environment

You must stop an environment before deleting it.

```
$ 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
```

Delete an application

Before deleting an application, ensure that all environments of the application are released.

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

12 Query applications and environments

After creating applications and environments, you can run commands to view their configuration information.

Query applications

Run the following command to query applications under the current account. You can view the name, ID, category, and associated environments of each application.

```
$ wpctl app:list
[OK] Query applications (2s)
```

NO	NAME	ENV(S) RUNNING	ENV(S) TOTAL	ID	DESCRIPTION
1	Java	0	1	wa-*****	
2	Java	0	3	wa-*****	
3	Tomcat	0	1	wa-*****	

Query the environments of an application

Run the following command to query the deployment environments of an application:

```
$ wpctl env:list --app ***
[OK] Query environments (1s)
Environment(s) in application ***
```

NO	NAME	ID	DESCRIPTION
STACK ID	PACKAGE LABEL	TYPE	STATUS
1	***	we-*****	web ABNORMAL

Query events

You can query all events in a deployment environment at any time. After the `env :events` command is run and returns an event list, you can press **E** or **B** on your

keyboard to jump to the first or last page, respectively, or press R to refresh the event list.

```
wpctl env:events

-- Check events, ctl + c to end --
2019-04-19 09:09:14 PM: The application environment changes to that
whose ID is wc-****89c9.
2019-04-19 09:09:14 PM: Environment we-****8981 is bound to VPC vpc-
****ouha.
2019-04-19 09:09:15 PM: Environment we-****8981 is bound to VSwitch vsw
-****tdfl.
2019-04-19 09:09:23 PM: Start to create instances, quantity: 1.
```

Query the default deployment environment

Run the `wpctl env:info` command to query the environment information in the current working directory.

```
$ 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://****
```

Query deployment package versions

Run the `wpctl pkg:list` command to query the deployment package versions of an application.

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

```
+-----+-----+-----+-----+
+-----+
+
```

Query the health status of a deployment environment

Run the `wpctl env:health` command to query the health status of an environment.

```
$ wpctl env:health
[OK] Describe environment health name demo-test-env (1s)

EnvId:we-5cced6393b331c3b55925679   EnvName:demo-test-env
+-----+-----+-----+-----+
+-----+
| NO |          INSTANCEID          | AGENTSTATUS | APPSTATUS | DISCONNECT
EDTIME |
+-----+-----+-----+-----+
+-----+
| 1  | i-wz91tmwedluum4mldj9z | CONNECTED  | HEALTHY   |
0  |
| 2  | i-wz91tmwedluum4mldj9x | CONNECTED  | HEALTHY   |
0  |
| 3  | i-wz91tmwedluum4mldj9y | CONNECTED  | HEALTHY   |
0  |
| 4  | i-wz91tmwedluum4mldja0 | CONNECTED  | HEALTHY   |
0  |
+-----+-----+-----+-----+
+-----+
```

Query and access the URL of an environment

Run the `wpctl env:open env Name/ID` command to query and access the URL of an environment.



Note:

Before running the `wpctl env:open env Name/ID` command, ensure that a browser is installed in your operating system. When an application is bound to multiple Server Load Balancer (SLB) instances or created on multiple Elastic Compute Service (ECS) instances, you can select a URL to open.

```
$ 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]:
```

Open the Web+ console

If a browser is installed in your operating system, you can run the `$ wpctl env:console` command to open the Web+ console.



Note:

Before running the `wpctl env:open env Name/ID` command, ensure that a browser is installed in your operating system. When an application is bound to multiple Server Load Balancer (SLB) instances or created on multiple Elastic Compute Service (ECS) instances, you can select a URL to open.

```
$ wpctl env:console
```

13 Use an environment configuration template

You can save any environment as a configuration template. Then, you can use this template to create and start an environment for the same application at any time.

- **template:list:** lists all templates of the current application.
- **template:delete:** deletes a specified template.
- **template:launch:** creates and starts an environment for an application by using the template.

```
$ 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-wz9bflpiec
xeotsgetdfl)
```


14 Collect error information

When an unknown error occurs during an operation, you can run the `wpctl collect` command to collect error information and resolve the error as follows:

1. Run the doctor command to check whether Web App Service (Web+) and associated services are activated, and whether required permissions are granted to roles related to Web+.
2. Analyze error information and determine whether the error is caused by incorrect parameter settings or network disconnection.
3. Do not switch or perform operations on the working directory. After running the `wpctl collect` command to collect information, send the generated ZIP file in the working directory to the relevant technical support staff.
4. If an unknown operation is performed after the error occurs, run the `wpctl collect --all` command to collect all information. Pay attention to the operation period, operation commands, error messages, and the request ID in each error message.

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

15 Perform operations on an ECS instance

To perform operations on an Elastic Compute Service (ECS) instance in an environment, you can only run the `ssh` command to log on to the ECS instance through Secure Shell (SSH), and then run the `exec` command on the ECS instance.

- **wpctl ssh:** You need to specify a jump server in the security group where the environment resides. You can select an ECS instance or create a 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-wz9i6yr-cr8jgkurozbzi | 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@iZwz9i6yr-cr8jgkurozbzi ~]#
```

- **wpctl exec: After logging on to the target ECS instance in the deployment environment, you can run commands on the ECS instance. In this case, you need to install Cloud Assistant on the ECS instance.**

```
$ 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-wz9i6yr-cr8jgkurozbzi | 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
<<<<< -----
```