

Microsoft Azure Cross Platform Command Line

9,331 commits

4 branches

59 releases

136 contributors

Branch: devNew pull request

Find file

Clone or download

amarzavery committed on GitHub Merge pull request #3047 from AuxMon/dev Latest commit 7fdcf96 2 hours ago

.github	Support blob snapshots and blob/container lease commands	2 months ago
Documentation	Update Writing-Cmd.md	27 days ago
bin	Several bug fixes mainly to alerts and autoscale actions and webhooks...	4 months ago
features	Add contributors.txt and update copyrights	3 years ago
lib	Merge pull request #3047 from AuxMon/dev	2 hours ago
scripts	Updated logging and interaction.	a month ago
test	Merge pull request #3047 from AuxMon/dev	2 hours ago
tools	Bumped up version to 0.10.2 and update ChangeLog.md	a month ago
.dockerignore	Added dockerignore file	21 days ago
.gitattributes	Bring back deployment scripts	4 years ago
.gitignore	Add custom action to propagate env change	5 months ago
.jshintrc	'network dns record-set list' output improved, #1779 fixed	6 months ago
.npmignore	updated npmignore file	19 days ago
.travis.yml	Reverted "testing for NodeV6 on CI".	a month ago
CONTRIBUTORS.txt	Add contributors.txt and update copyrights	3 years ago
ChangeLog.md	Bumped up version to 0.10.2 and update ChangeLog.md	a month ago
Dockerfile	docker build use latest source (#3001)	21 days ago
LICENSE.txt	0.5.0 release.	5 years ago
README.md	Add Batch to README	2 months ago
cli.njsproj	Support blob snapshots and blob/container lease commands	2 months ago
cli.sln	add default vs nodejs project	a year ago
compile-for-publish.sh	correct naming for testlist-arm.txt	4 months ago
package.json	Merge pull request #3047 from AuxMon/dev	2 hours ago
xcli-cloned.sln	Changed networking quotas/usage command 'azure network usage list <re...	2 months ago

README.md

Microsoft Azure Xplat-CLI for Windows, Mac and Linux

npm package0.10.2buildpassing

This project provides a cross-platform command line interface for developers and IT administrators to develop, deploy and manage Microsoft Azure applications.

Features

- Accounts Management
- Azure Service Management
 - Storage
 - Websites
 - Virtual machines
 - Network
 - Mobile Services
 - Service Bus
 - SQL Database
- Azure Resource Management
 - Generic resource groups and deployments management
 - Role based access control
 - Cmdlets for individual resource providers, including compute, storage, network, redis cache, insights, etc.
 - Datalake
 - CDN
 - HDInsight (Note: The HDInsight commands in ASM mode are deprecated and will be removed by Jan. 2017)
 - Key Vault
 - Batch

Note: The list of features may not be up-to-date. For accurate command details, type `azure |` `azure -h` | `azure --help` to navigate through the help system. Also, use `azure config mode asm|arm` to switch between service management (Version V1) and resource management (Version V2) of the Azure REST API.

Endpoints for Azure

The CLI targets "Public Azure Cloud" by default. You can get more information about the endpoints supported in different environments from [here](#).

Non-Interactive Authentication

If you need to create an automation account for non interactive or scripting scenarios then please take a look at the documentation over [here](#).

Installation

Installation from a particular branch of this repository

- Uninstall the previously installed CLI
 - If you installed via MSI, then uninstall the windows MSI. For mac installer `sudo azure-uninstall -g`
 - If you installed via npm then execute: `npm uninstall azure-cli -g`
- Clear the global cache: `npm cache clear -g`
- Delete the `.streamline` folder from the User's home profile `C:\Users\<username>\.streamline` | `~/streamline`, if present.
- Download the tarball from this link: <https://github.com/Azure/azure-xplat-cli/archive/<branch-name>.tar.gz>
- Install from the tarball: `npm install -g <path to the downloaded tarball>`

And you should be good to go :). Note: You may require admin prompt or sudo access to install via npm.

Installation from npm

You can install the azure-cli npm package directly.

```
npm install -g azure-cli
```

Installing the latest version of node.js on different linux flavors

This [document](#) provides simple steps to install the latest version of node.js on a linux system. After successful installation of node.js, you can install "azure-cli" via npm as described above.

Installation on a Docker Host

In a Docker host, run:

```
sudo docker run -it microsoft/azure-cli
```

You can clone the repo and use the "Dockerfile" from master branch. It should install the last released version of azure-cli.

Pre-compiled installers

- [Windows](#)
- [Mac](#)
- [Linux](#)

Configure auto-complete

Auto-complete is supported for Mac and Linux.

To enable it in zsh, run:

```
echo '. <(azure --completion)' >> .zshrc
```

To enable it in bash, run:

```
azure --completion >> ~/azure.completion.sh  
echo 'source ~/azure.completion.sh' >> ~/.bash_profile
```

Get Started

- First, get authenticated with Microsoft Azure. For details, read [this article](#).
 - Option 1: Login with your Microsoft account, such as live-id, or organizational account, or service principals.
 - Option 2: Download and import a publish settings file which contains a management certificate.

If you use both mechanisms on the same subscription, Azure Active Directory authentication will be used by default. If you want to go back to management certificate authentication, please use 'azure logout', which will remove the Azure Active Directory information and bring management certificate authentication back in.

Login directly from xplat-cli (Azure Active Directory authentication) - works with ARM & ASM (Version V2 & V1) of Azure API

```
# This will output an url and a device code for you to use browser to login  
azure login  
  
# This will prompt for your password in the console  
azure login -u <your organizational ID email address>  
  
# This will login in using a service principal  
azure login -u "<service-principal-id>" -p "<key>" --service-principal --tenant "<tenant-id>"
```

Use publish settings file (Management certificate authentication) - works only with ASM (Version V1) of Azure API

```
# Download a file which contains the publish settings information of your subscription.  
# This will open a browser window and ask you to log in to get the file.  
azure account download  
  
# Import the file you just downloaded.  
# Notice that the file contains credential of your subscription so you don't want to make it public  
# (like check in to source control, etc.).  
azure account import <file location>
```

```
# Use the commands to manage your services/applications
azure site create --location "West US" mywebsite
```

azure cli with China Cloud

```
# This will log you into the China Cloud environment.
# You can use same set of commands to manage your service/applications
azure login -u <your organizational ID email address> -e AzureChinaCloud
```

azure cli on Ubuntu

If you want to run xplat cli on Ubuntu, then you should install **nodejs-legacy** instead of **nodejs**. For more information please check the following links:

- [why there is a problem with nodejs installation on ubuntu](#)
- [how to solve the nodejs installation problem on ubuntu](#)

Please perform the installation steps in following order:

```
sudo apt-get install nodejs-legacy
sudo apt-get install npm
sudo npm install -g azure-cli
```

2 Modes

Starting from 0.8.0, we are adding a separate mode for Resource Manager. You can use the following command to switch between the

- Service management: commands using the Azure service management API
- Resource manager: commands using the Azure Resource Manager API

They are not designed to work together.

```
azure config mode asm # service management
azure config mode arm # resource manager
```

For more details on the commands, please see the [command line tool reference](#) and this [How to Guide](#)

Docker

Usage is the same as `vm create` command:

```
azure vm docker create [options] <dns-name> <image> <user-name> [password]
```

This command only supports Ubuntu 14.04+ and CoreOS based images. Docker is configured on the VM using HTTPS as described here: <https://docs.docker.com/articles/https/> By default, generated TLS certificates are placed in the `~/.docker` directory, and Docker is configured to run on port 2376. These can be configured using new options:

<code>-dp, --docker-port [port]</code>	Port to use for docker [2376]
<code>-dc, --docker-cert-dir [dir]</code>	Directory containing docker certs [~/.docker/]

After the VM is created. It can be used as a Docker host with the `-H` option or `DOCKER_HOST` environment variable.

```
docker --tls -H tcp://<my-host>.cloudapp.net:2376 info
```

Note: To run docker commands on windows make sure ssl agent is installed.

Error Diagnostic

use the `-vv` option to see the actual REST requests on the console.

```
azure site create --location "West US" mytestsite -vv
```

Use web debugging proxy

Say, use 'Fiddler', setup the following environment variables before execute commands.

```
set NODE_TLS_REJECT_UNAUTHORIZED=0
set HTTPS_PROXY=http://127.0.0.1:8888
```

Running Tests

See [this page for instructions](#) that describe how to run the test suite.

Learn More

For documentation on how to host Node.js applications on Microsoft Azure, please see the [Microsoft Azure Node.js Developer Center](#).

Contribute Code or Provide Feedback

If you would like to become an active contributor to this project please follow the instructions provided in [Microsoft Azure Projects Contribution Guidelines](#).

Please send pull requests only to the **Dev branch**. Please make sure that you have checked in tests and recorded them live for your contribution. **Pull requests without sufficient tests will not be accepted.**

If you encounter any bugs with the library please file an issue in the [Issues](#) section of the project.

