Contents

- Azure CLI
  - Install
    - macOS
    - Windows
    - Linux via apt
    - Linux via yum
    - Linux via zypper
    - Linux
    - Run in Docker
  - Get started
  - Cloud shell
  - Sign in
    - Create a service principal
  - Interactive mode
  - Deployment templates
    - Use deployment templates
    - Export deployment templates
    - Deploy from private templates
  - Manage subscriptions
  - Work with multiple clouds
  - Output formats
  - Query
  - Extensions
    - Available extensions
    - Alias extension
  - Configuration settings
  - Release notes
  - CLI Versioning
- Tutorials

- Create virtual machines
- Sample Scripts
  - Linux VMs
  - Windows VMs
  - Web Apps
  - Functions
  - SQL Database
  - PostgreSQL
  - MySQL
  - Cosmos DB
  - Batch

Azure Command-Line Interface (CLI)

2/19/2019 • 2 minutes to read • Edit Online

The Azure command-line interface (CLI) is Microsoft's cross-platform command-line experience for managing Azure resources. Use it in your browser with Azure Cloud Shell, or install it on macOS, Linux, or Windows and run it from the command line.

The Azure CLI is easy to get started with, and best used for building automation scripts that work with the Azure Resource Manager. Using the Azure CLI, you can create VMs within Azure as easily as typing the following command:

```
az vm create -n MyLinuxVM -g MyResourceGroup --image UbuntuLTS
```

NOTE

In scripts and on the Microsoft documentation site, Azure CLI examples are written for the `bash` shell. One-line examples will run on any platform. Longer examples which include line continuations (`\`) or variable assignment need to be modified to work on other shells, including PowerShell.

Run or Install

You can install the CLI locally, run it in the browser with Azure Cloud Shell, or run in a Docker container. To get the current version of the CLI, run `az --version`.

- To run in your browser with Azure Cloud Shell, see Quickstart for Bash in Azure Cloud Shell or Quickstart for PowerShell in Azure Cloud Shell.
- To install the CLI, see Install the Azure CLI.
- To run as a Docker container, see Run Azure CLI in a Docker Container

Build your skills with Microsoft Learn

- Manage virtual machines with the Azure CLI
- Control Azure services with the CLI
- More interactive learning...

Get started

Read the Get Started article to learn the CLI basics. The following samples demonstrate some common uses cases:

- Linux Virtual Machines
- Windows Virtual Machines
- Web Apps
- SQL Database

A detailed reference is also available that documents how to use each individual Azure CLI command.

NOTE

If you use the previous version of the CLI (Azure classic CLI), you can continue to use it. However, we recommend updating to use the latest version of the Azure CLI for the best experience. If you use both CLIs, remember that `azure` is the classic CLI and that `az` is the most recent CLI.

Install the Azure CLI

2/19/2019 • 2 minutes to read • Edit Online

The Azure CLI is a command-line tool providing a great experience for managing Azure resources. The CLI is designed to make scripting easy, query data, support long-running operations, and more. Try it today and find out what the CLI has to offer!

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

- Install on Windows
- Install on macOS
- Install on Linux or Windows Subsystem for Linux (WSL)
  - Install with apt on Debian or Ubuntu
  - Install with yum on RHEL, Fedora, or CentOS
  - Install with zypper on openSUSE or SLE
  - Install from script
- Run in Docker container

> **NOTE**
> If you're using the Azure classic deployment model, install the Azure classic CLI.

Install Azure CLI on macOS

2/19/2019 • 2 minutes to read • Edit Online

For the macOS platform, you can install the Azure CLI with homebrew package manager. Homebrew makes it easy to keep your installation of the CLI update to date. The CLI package has been tested on macOS versions 10.9 and later.

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

Install

Homebrew is the easiest way to manage your CLI install. It provides convenient ways to install, update, and uninstall. If you don't have homebrew available on your system, install homebrew before continuing.

You can install the CLI by updating your brew repository information, and then running the `install` command:

```
brew update && brew install azure-cli
```

IMPORTANT

The Azure CLI has a dependency on the `python3` package in Homebrew, and will install it on your system, even if Python 2 is available. The Azure CLI is guaranteed to be compatible with the latest version of `python3` published on Homebrew.

You can then run the Azure CLI with the `az` command. To sign in, use az login command.

1. Run the `login` command.

   ```
   az login
   ```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with Azure CLI.

Troubleshooting

If you encounter a problem when installing the CLI through Homebrew, here are some common errors. If you experience a problem not covered here, file an issue on github.

Unable to find Python or installed packages

There may be a minor version mismatch or other issue during homebrew installation. The CLI doesn't use a Python virtual environment, so it relies on finding the installed Python version. A possible fix is to install and relink the `python3` dependency from Homebrew.

```
brew update && brew install python3 && brew upgrade python3
brew link --overwrite python3
```

CLI version 1.x is installed

If an out-of-date version was installed, it could be because of a stale homebrew cache. Follow the update instructions.

Update

The CLI is regularly updated with bug fixes, improvements, new features, and preview functionality. A new release is available roughly every two weeks. Update your local repository information and then upgrade the `azure-cli` package.

```
brew update && brew upgrade azure-cli
```

Uninstall

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

Use homebrew to uninstall the `azure-cli` package.

```
brew uninstall azure-cli
```

Other installation methods

If you can't use homebrew to install the Azure CLI in your environment, it's possible to use the manual instructions for Linux. Note that this process is not officially maintained to be compatible with macOS. Using a package manager such as Homebrew is always recommended. Only use the manual installation method if you have no other option available.

For the manual installation instructions, see Install Azure CLI on Linux manually.

Next Steps

Now that you've installed the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Install Azure CLI on Windows

2/19/2019 • 2 minutes to read • Edit Online

For Windows the Azure CLI is installed via an MSI, which gives you access to the CLI through the Windows Command Prompt (CMD) or PowerShell. When installing for Windows Subsystem for Linux (WSL), packages are available for your Linux distribution. See the main install page for the list of supported package managers or how to install manually under WSL.

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

Install or update

The MSI distributable is used for installing, updating, and uninstalling the `az` command on Windows.

Download the MSI installer

When the installer asks if it can make changes to your computer, click the "Yes" box.

You can now run the Azure CLI with the `az` command from either Windows Command Prompt or PowerShell. PowerShell offers some tab completion features not available from Windows Command Prompt. To sign in, run the az login command.

1. Run the `login` command.

   ```
   az login
   ```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with Azure CLI.

Uninstall

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

Uninstalling can be done by running the MSI again, and choosing the "Uninstall" option.

Download the MSI installer

Next Steps

Now that you've installed the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Install Azure CLI with apt

3/21/2019 • 4 minutes to read • Edit Online

If you are running a distribution that comes with `apt`, such as Ubuntu or Debian, there's an x86_64 package available for the Azure CLI. This package has been tested with:

- Ubuntu trusty, xenial, artful, and bionic
- Debian wheezy, jessie, and stretch

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

> **NOTE**
> The package for Azure CLI installs its own Python interpreter, and does not use the system Python.

Install

1. Get packages needed for the install process:

```
sudo apt-get update
sudo apt-get install curl apt-transport-https lsb-release gpg
```

2. Download and install the Microsoft signing key:

```
curl -sL https://packages.microsoft.com/keys/microsoft.asc | \
    gpg --dearmor | \
    sudo tee /etc/apt/trusted.gpg.d/microsoft.asc.gpg > /dev/null
```

3. Add the Azure CLI software repository:

```
AZ_REPO=$(lsb_release -cs)
echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO main" | \
    sudo tee /etc/apt/sources.list.d/azure-cli.list
```

4. Update repository information and install the `azure-cli` package:

```
sudo apt-get update
sudo apt-get install azure-cli
```

Run the Azure CLI with the `az` command. To sign in, use the az login command.

1. Run the `login` command.

```
az login
```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an

authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with Azure CLI.

Troubleshooting

Here are some common problems seen when installing with `apt`. If you experience a problem not covered here, file an issue on github.

lsb_release does not return the correct base distribution version

Some Ubuntu- or Debian-derived distributions such as Linux Mint may not return the correct version name from `lsb_release`. This value is used in the install process to determine the package to install. If you know the code name of the Ubuntu or Debian version your distribution is derived from, you can set the `AZ_REPO` value manually when adding the repository. Otherwise, look up information for your distribution on how to determine the base distribution code name and set `AZ_REPO` to the correct value.

No package for your distribution

Sometimes it may be a while after a distribution is released before there's an Azure CLI package available for it. The Azure CLI designed to be resilient with regards to future versions of dependencies and rely on as few of them as possible. If there's no package available for your base distribution, try a package for an earlier distribution.

To do this, set the value of `AZ_REPO` manually when adding the repository. For Ubuntu distributions use the `bionic` repository, and for Debian distributions use `stretch`. Distributions released before Ubuntu Trusty and Debian Wheezy are not supported.

CLI fails to install or run on Windows Subsystem for Linux

Since Windows Subsystem for Linux (WSL) is a system call translation layer on top of the Windows platform, you might experience an error when trying to install or run the Azure CLI. The CLI relies on some features that may have a bug in WSL. If you experience an error no matter how you install the CLI, there's a good chance it's an issue with WSL and not with the CLI install process.

To troubleshoot your WSL installation and possibly resolve issues:

- If you can, run an identical install process on a Linux machine or VM to see if it succeeds. If it does, your issue is almost certainly related to WSL. To start a Linux VM in Azure, see the create a Linux VM in the Azure Portal documentation.
- Make sure that you're running the latest version of WSL. To get the latest version, update your Windows 10 installation.
- Check for any open issues with WSL which might address your problem. Often there will be suggestions on how to work around the problem, or information about a release where the issue will be fixed.
- If there are no existing issues for your problem, file a new issue with WSL and make sure that you include as much information as possible.

If you continue to have issues installing or running on WSL, consider installing the CLI for Windows.

Update

Use `apt-get upgrade` to update the CLI package.

```
sudo apt-get update && sudo apt-get upgrade
```

NOTE

This command upgrades all of the installed packages on your system that have not had a dependency change. To upgrade the CLI only, use `apt-get install`.

```
sudo apt-get update && sudo apt-get install --only-upgrade -y azure-cli
```

Uninstall

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

1. Uninstall with `apt-get remove`:

   ```
   sudo apt-get remove -y azure-cli
   ```

2. If you don't plan to reinstall the CLI, remove the Azure CLI repository information:

   ```
   sudo rm /etc/apt/sources.list.d/azure-cli.list
   ```

3. Remove the signing key:

   ```
   sudo rm /etc/apt/trusted.gpg.d/microsoft.asc.gpg
   ```

4. Remove any unneeded packages:

   ```
   sudo apt autoremove
   ```

Next Steps

Now that you've installed the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Install Azure CLI with yum

2/19/2019 • 3 minutes to read • Edit Online

For Linux distributions with `yum` such as RHEL, Fedora, or CentOS, there's a package for the Azure CLI. This package has been tested with RHEL 7, Fedora 19 and higher, and CentOS 7.

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

> **IMPORTANT**
>
> The RPM package of the Azure CLI depends on the `python` package. On your system, this may be a Python version which predates the requirement of Python 2.7.x. If this affects you, find a replacement `python` package or follow the manual install instructions.
>
> Be aware that Python 2 is being end-of-lifed on January 1, 2020, and will no longer receive updates. For this reason, upgrade to Python 3 when possible. The Azure CLI is compatible with Python 3.6 and higher.

Install

1. Import the Microsoft repository key.

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

2. Create local `azure-cli` repository information.

```
sudo sh -c 'echo -e "[azure-cli]\nname=Azure CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
```

3. Install with the `yum install` command.

```
sudo yum install azure-cli
```

You can then run the Azure CLI with the `az` command. To sign in, use az login command.

1. Run the `login` command.

```
az login
```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with Azure CLI.

Troubleshooting

Here are some common problems seen when installing with `yum`. If you experience a problem not covered here, file an issue on github.

CLI fails to install or run on Windows Subsystem for Linux

Since Windows Subsystem for Linux (WSL) is a system call translation layer on top of the Windows platform, you might experience an error when trying to install or run the Azure CLI. The CLI relies on some features that may have a bug in WSL. If you experience an error no matter how you install the CLI, there's a good chance it's an issue with WSL and not with the CLI install process.

To troubleshoot your WSL installation and possibly resolve issues:

- If you can, run an identical install process on a Linux machine or VM to see if it succeeds. If it does, your issue is almost certainly related to WSL. To start a Linux VM in Azure, see the create a Linux VM in the Azure Portal documentation.
- Make sure that you're running the latest version of WSL. To get the latest version, update your Windows 10 installation.
- Check for any open issues with WSL which might address your problem. Often there will be suggestions on how to work around the problem, or information about a release where the issue will be fixed.
- If there are no existing issues for your problem, file a new issue with WSL and make sure that you include as much information as possible.

If you continue to have issues installing or running on WSL, consider installing the CLI for Windows.

Update

Update the Azure CLI with the `yum update` command.

```
sudo yum update azure-cli
```

Uninstall

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

1. Remove the package from your system.

   ```
   sudo yum remove azure-cli
   ```

2. If you don't plan to reinstall the CLI, remove the repository information.

   ```
   sudo rm /etc/yum.repos.d/azure-cli.repo
   ```

3. If you removed the repository information, also remove the Microsoft GPG signature key.

   ```
   MSFT_KEY=`rpm -qa gpg-pubkey /* --qf "%{version}-%{release} %{summary}\n" | grep Microsoft | awk '{print
   $1}'`
   sudo rpm -e --allmatches gpg-pubkey-$MSFT_KEY
   ```

Next Steps

Now that you've installed the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Install Azure CLI with zypper

2/19/2019 • 3 minutes to read • Edit Online

For Linux distributions with `zypper`, such as openSUSE or SLES, there's a package available for the Azure CLI. This package has been tested with openSUSE 42.2 and SLES 12 SP 2.

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

IMPORTANT

The RPM package of the Azure CLI depends on the `python` package. On your system, this may be a Python version which predates the requirement of Python 2.7.x. If this affects you, find a replacement `python` package or follow the manual install instructions.

Be aware that Python 2 is being end-of-lifed on January 1, 2020, and will no longer receive updates. For this reason, upgrade to Python 3 when possible. The Azure CLI is compatible with Python 3.6 and higher.

Install

1. Install `curl`:

```
sudo zypper install -y curl
```

2. Import the Microsoft repository key:

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

3. Create local `azure-cli` repository information:

```
sudo zypper addrepo --name 'Azure CLI' --check https://packages.microsoft.com/yumrepos/azure-cli azure-cli
```

4. Update the `zypper` package index and install:

```
sudo zypper install --from azure-cli -y azure-cli
```

You can then run the Azure CLI with the `az` command. To sign in, use az login command.

1. Run the `login` command.

```
az login
```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with Azure CLI.

Troubleshooting

Here are some common problems seen when installing with `zypper`. If you experience a problem not covered here, file an issue on github.

CLI fails to install or run on Windows Subsystem for Linux

Since Windows Subsystem for Linux (WSL) is a system call translation layer on top of the Windows platform, you might experience an error when trying to install or run the Azure CLI. The CLI relies on some features that may have a bug in WSL. If you experience an error no matter how you install the CLI, there's a good chance it's an issue with WSL and not with the CLI install process.

To troubleshoot your WSL installation and possibly resolve issues:

- If you can, run an identical install process on a Linux machine or VM to see if it succeeds. If it does, your issue is almost certainly related to WSL. To start a Linux VM in Azure, see the create a Linux VM in the Azure Portal documentation.
- Make sure that you're running the latest version of WSL. To get the latest version, update your Windows 10 installation.
- Check for any open issues with WSL which might address your problem. Often there will be suggestions on how to work around the problem, or information about a release where the issue will be fixed.
- If there are no existing issues for your problem, file a new issue with WSL and make sure that you include as much information as possible.

If you continue to have issues installing or running on WSL, consider installing the CLI for Windows.

Update

You can update the package with the `zypper update` command.

```
sudo zypper refresh
sudo zypper update azure-cli
```

Uninstall

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

1. Remove the package from your system.

   ```
   sudo zypper remove -y azure-cli
   ```

2. If you don't plan to reinstall the CLI, remove the repository information.

   ```
   sudo zypper removerepo azure-cli
   ```

3. If you removed the repository information, also remove the Microsoft GPG signature key.

```
MSFT_KEY=`rpm -qa gpg-pubkey /* --qf "%{version}-%{release} %{summary}\n" | grep Microsoft | awk '{print
$1}'`
sudo rpm -e --allmatches gpg-pubkey-$MSFT_KEY
```

Next Steps

Now that you've installed the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Install Azure CLI on Linux manually

2/19/2019 • 3 minutes to read • Edit Online

If there's no package for the Azure CLI for you your distribution, install the CLI manually by running a script.

The current version of the CLI is **2.0.60**. For information about the latest release, see the release notes. To find the version running on your system to see if you need to update, run `az --version`.

> **NOTE**
> It's strongly recommend to install the CLI with a package manager. A package manager makes sure you always get the latest updates, and guarantees the stability of CLI components. Check and see if there is a package for your distribution before installing manually.

Prerequisites

The CLI requires the following software:

- Python 3.6.x or 3.7.x.
- libffi
- OpenSSL 1.0.2

> **IMPORTANT**
> The CLI is also compatible with Python 2.7.x, which is being end-of-lifed on January 1, 2020. For this reason we recommend that you install Python 3 to run the CLI.

Install or update

Both installing and updating the CLI requires re-running the install script. Install the CLI by running `curl`.

```
curl -L https://aka.ms/InstallAzureCli | bash
```

The script can also be downloaded and run locally. You may have to restart your shell in order for changes to take effect.

You can then run the Azure CLI with the `az` command. To sign in, use az login command.

1. Run the `login` command.

   ```
   az login
   ```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with Azure CLI.

Troubleshooting

Here are some common problems seen during a manual installation. If you experience a problem not covered here, file an issue on GitHub.

curl "Object Moved" error

If you get an error from `curl` related to the `-L` parameter, or an error message including the text "Object Moved", try using the full URL instead of the `aka.ms` redirect:

```
curl https://azurecliprod.blob.core.windows.net/install | bash
```

`az` command not found

If you can't run the command after installation and using `bash` or `zsh`, clear your shell's command hash cache. Run

```
hash -r
```

and check if the problem is resolved.

The issue can also occur if you didn't restart your shell after installation. Make sure that the location of the `az` command is in your `$PATH`. The location of the `az` command is

```
<install path>/bin
```

CLI fails to install or run on Windows Subsystem for Linux

Since Windows Subsystem for Linux (WSL) is a system call translation layer on top of the Windows platform, you might experience an error when trying to install or run the Azure CLI. The CLI relies on some features that may have a bug in WSL. If you experience an error no matter how you install the CLI, there's a good chance it's an issue with WSL and not with the CLI install process.

To troubleshoot your WSL installation and possibly resolve issues:

- If you can, run an identical install process on a Linux machine or VM to see if it succeeds. If it does, your issue is almost certainly related to WSL. To start a Linux VM in Azure, see the create a Linux VM in the Azure Portal documentation.
- Make sure that you're running the latest version of WSL. To get the latest version, update your Windows 10 installation.
- Check for any open issues with WSL which might address your problem. Often there will be suggestions on how to work around the problem, or information about a release where the issue will be fixed.
- If there are no existing issues for your problem, file a new issue with WSL and make sure that you include as much information as possible.

If you continue to have issues installing or running on WSL, consider installing the CLI for Windows.

Uninstall

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

Uninstall the CLI by directly deleting the files from the location chosen at the time of installation. The default install location is `$HOME`.

1. Remove the installed CLI files.

```
rm -r <install location>/lib/azure-cli
rm <install location>/bin/az
```

2. Modify your `$HOME/.bash_profile` file to remove the following line:

```
<install location>/lib/azure-cli/az.completion
```

3. If using `bash` or `zsh`, reload your shell's command cache.

```
hash -r
```

Next Steps

Now that you've installed the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Run Azure CLI in a Docker container

1/17/2019 • 2 minutes to read • Edit Online

You can use Docker to run a standalone Linux container with the Azure CLI pre-installed. Docker gets you started quickly with an isolated environment to run the CLI in. The image can also be used as a base for your own deployments.

Run in a Docker container

Install the CLI using `docker run`.

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

NOTE

If you want to pick up the SSH keys from your user environment, use `-v ${HOME}/.ssh:/root/.ssh` to mount your SSH keys in the environment.

```
docker run -it -v ${HOME}/.ssh:/root/.ssh microsoft/azure-cli
```

The CLI is installed on the image as the `az` command in `/usr/local/bin`. To sign in, run the az login command.

1. Run the `login` command.

   ```
   az login
   ```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

To learn more about different authentication methods, see Sign in with the Azure CLI.

Update Docker image

Updating with Docker requires both pulling the new image and re-creating any existing containers. For this reason, you should try to avoid using a container that hosts the CLI as a data store.

Update your local image with `docker pull`.

```
docker pull microsoft/azure-cli
```

Uninstall Docker image

If you decide to uninstall the Azure CLI, we're sorry to see you go. Before you uninstall, use the `az feedback` command to let us know what could be improved or fixed. Our goal is to make the Azure CLI bug-free and user-

friendly. If you found a bug, we'd appreciate it if you file a GitHub issue.

After halting any containers running the CLI image, remove it.

```
docker rmi microsoft/azure-cli
```

Next Steps

Now that you're ready to use the Azure CLI, take a short tour of its features and common commands.

Get started with the Azure CLI

Get started with Azure CLI

2/27/2019 • 3 minutes to read • Edit Online

Welcome to the Azure CLI! The CLI is a tool designed to get you working quickly and efficiently with Azure services, with an emphasis on automation. This article introduces features of the CLI and links out to resources that help you be productive.

Install or run in Azure Cloud Shell

The easiest way to get started with the Azure CLI is by running it in an Azure Cloud Shell environment through your browser. To learn about Cloud Shell, see Quickstart for Bash in Azure Cloud Shell.

When you're ready to install the CLI, see the installation instructions.

After installing the CLI for the first time, check that it's installed and you've got the correct version by running `az --version`.

Sign in

Before using any CLI commands with a local install, you need to sign in with az login.

1. Run the `login` command.

   ```
   az login
   ```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

After logging in, you see a list of subscriptions associated with your Azure account. The subscription information with `isDefault: true` is the currently activated subscription after logging in. To select another subscription, use the az account set command with the subscription ID to switch to. For more information about subscription selection, see Use multiple Azure subscriptions.

There are ways to sign in non-interactively, which are covered in detail in Sign in with Azure CLI.

Common commands

This table lists some common commands used in the CLI and links to their reference documentation.

RESOURCE TYPE	AZURE CLI COMMAND GROUP
Resource group	az group
Virtual machines	az vm
Storage accounts	az storage account

RESOURCE TYPE	AZURE CLI COMMAND GROUP
Key Vault	az keyvault
Web applications	az webapp
SQL databases	az sql server
CosmosDB	az cosmosdb

Finding commands

Commands in the CLI are organized as *commands* of *groups*. Each group represents an Azure service, and commands operate on that service.

To search for commands, use az find. For example, to search for command names containing `secret`, use the following command:

```
az find -q secret
```

Use the `--help` argument to get a complete list of commands and subgroups of a group. For example, to find the CLI commands for working with Network Security Groups (NSGs):

```
az network nsg --help
```

The CLI has full tab completion for commands under the bash shell.

Globally available arguments

There are some arguments that are available for every command.

- `--help` prints CLI reference information about commands and their arguments and lists available subgroups and commands.
- `--output` changes the output format. The available output formats are `json`, `jsonc` (colorized JSON), `tsv` (Tab-Separated Values), `table` (human-readable ASCII tables), and `yaml`. By default the CLI outputs `json`. To learn more about the available output formats, see Output formats for Azure CLI.
- `--query` uses the JMESPath query language to filter the output returned from Azure services. To learn more about queries, see Query command results with Azure CLI and the JMESPath tutorial.
- `--verbose` prints information about resources created in Azure during an operation, and other useful information.
- `--debug` prints even more information about CLI operations, used for debugging purposes. If you find a bug, provide output generated with the `--debug` flag on when submitting a bug report.

Interactive mode

The CLI offers an interactive mode that automatically displays help information and makes it easier to select subcommands. You enter interactive mode with the az interactive command.

```
az interactive
```

For more information on interactive mode, see Azure CLI Interactive Mode.

There's also a Visual Studio Code plugin that offers an interactive experience, including autocomplete and mouse-over documentation.

Learn CLI basics with quickstarts and tutorials

To get you started with the Azure CLI, try an in-depth tutorial for setting up virtual machines and using the power of the CLI to query Azure resources.

Create virtual machines with the Azure CLI tutorial

There are also quickstarts for other popular services.

- Create a storage account using the Azure CLI
- Transfer objects to/from Azure Blob storage using the CLI
- Create a single Azure SQL database using the Azure CLI
- Create an Azure Database for MySQL server using the Azure CLI
- Create an Azure Database for PostgreSQL using the Azure CLI
- Create a Python web app in Azure
- Run a custom Docker Hub image in Azure Web Apps for Containers

Give feedback

We welcome your feedback for the CLI to help us make improvements and resolve bugs. You can file an issue on GitHub or use the built-in features of the CLI to leave general feedback with the az feedback command.

```
az feedback
```

Sign in with Azure CLI

2/22/2019 • 3 minutes to read • Edit Online

There are several authentication types for the Azure CLI. The easiest way to get started is with Azure Cloud Shell, which automatically logs you in. Locally, you can sign in interactively through your browser with the az login command. When writing scripts, the recommended approach is to use service principals. By granting just the appropriate permissions needed to a service principal, you can keep your automation secure.

None of your sign-in information is stored by the CLI. Instead, an authentication refresh token is generated by Azure and stored. As of August 2018 this token is revoked after 90 days of inactivity, but this value can be changed by Microsoft or your tenant administrator. Once the token is revoked you get a message from the CLI saying you need to sign in again.

After signing in, CLI commands are run against your default subscription. If you have multiple subscriptions, you can change your default subscription.

Sign in interactively

The Azure CLI's default authentication method uses a web browser and access token to sign in.

1. Run the `login` command.

   ```
   az login
   ```

   If the CLI can open your default browser, it will do so and load a sign-in page.

   Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to https://aka.ms/devicelogin in your browser.

2. Sign in with your account credentials in the browser.

Sign in with credentials on the command line

Provide your Azure user credentials on the command line.

> **NOTE**
> This approach doesn't work with Microsoft accounts or accounts that have two-factor authentication enabled.

```
az login -u <username> -p <password>
```

IMPORTANT

If you want to avoid displaying your password on console and are using `az login` interactively, use the `read -s` command under `bash`.

```
read -sp "Azure password: " AZ_PASS && echo && az login -u <username> -p $AZ_PASS
```

Under PowerShell, use the `Get-Credential` cmdlet.

```
$AzCred = Get-Credential -UserName <username>
az login -u $AzCred.UserName -p $AzCred.GetNetworkCredential().Password
```

Sign in with a service principal

Service principals are accounts not tied to any particular user, which can have permissions on them assigned through pre-defined roles. Authenticating with a service principal is the best way to write secure scripts or programs, allowing you to apply both permissions restrictions and locally stored static credential information. To learn more about service principals, see Create an Azure service principal with the Azure CLI.

To sign in with a service principal, you need:

- The URL or name associated with the service principal
- The service principal password, or the X509 certificate used to create the service principal in PEM format
- The tenant associated with the service principal, as either an `.onmicrosoft.com` domain or Azure object ID

IMPORTANT

If your service principal uses a certificate that is stored in Key Vault, that certificate's private key must be available without signing in to Azure. To retrieve a private key for use offline, use az keyvault secret show.

```
az login --service-principal -u <app-url> -p <password-or-cert> --tenant <tenant>
```

IMPORTANT

If you want to avoid displaying your password on console and are using `az login` interactively, use the `read -s` command under `bash`.

```
read -sp "Azure password: " AZ_PASS && echo && az login --service-principal -u <app-url> -p $AZ_PASS --
tenant <tenant>
```

Under PowerShell, use the `Get-Credential` cmdlet.

```
$AzCred = Get-Credential -UserName <app-url>
az login -u $AzCred.UserName -p $AzCred.GetNetworkCredential().Password --tenant <tenant>
```

Sign in with a different tenant

You can select a tenant to sign in under with the `--tenant` argument. The value of this argument can either be an `.onmicrosoft.com` domain or the Azure object ID for the tenant. Both interactive and command-line sign in methods work with `--tenant`.

```
az login --tenant <tenant>
```

Sign in with a managed identity

On resources configured for managed identities for Azure resources, you can sign in using the managed identity. Signing in with the resource's identity is done through the `--identity` flag.

```
az login --identity
```

To learn more about managed identities for Azure resources, see Configure managed identities for Azure resources and Use managed identities for Azure resources for sign in.

Create an Azure service principal with Azure CLI

2/22/2019 • 6 minutes to read • Edit Online

Automated tools that use Azure services should always have restricted permissions. Instead of having applications sign in as a fully privileged user, Azure offers service principals.

An Azure service principal is an identity created for use with applications, hosted services, and automated tools to access Azure resources. This access is restricted by the roles assigned to the service principal, giving you control over which resources can be accessed and at which level. For security reasons, it's always recommended to use service principals with automated tools rather than allowing them to log in with a user identity.

This article shows you the steps for creating, getting information about, and resetting a service principal with the Azure CLI.

Create a service principal

Create a service principal with the az ad sp create-for-rbac command. When creating a service principal, you choose the type of sign-in authentication it uses.

> **NOTE**
> If your account doesn't have permission to create a service principal, `az ad sp create-for-rbac` will return an error message containing "Insufficient privileges to complete the operation." Contact your Azure Active Directory admin to create a service principal.

There are two types of authentication available for service principals: Password-based authentication, and certificate-based authentication.

Password-based authentication

Without any authentication parameters, password-based authentication is used with a random password created for you. If you want password-based authentication, this method is recommended.

```
az ad sp create-for-rbac --name ServicePrincipalName
```

For a user-supplied password, use the `--password` argument. When creating a password, make sure you follow the Azure Active Directory password rules and restrictions. Don't use a weak password or reuse a password.

```
az ad sp create-for-rbac --name ServicePrincipalName --password <Choose a strong password>
```

> **IMPORTANT**
> For security reasons, the `--password` argument for service principal creation will be deprecated in a future release. If you want to use password-based authentication, avoid `--password` and let the CLI generate a secure password for you.

The output for a service principal with password authentication includes the `password` key. **Make sure** you copy this value - it can't be retrieved. If you forget the password, reset the service principal credentials.

The `appId` and `tenant` keys appear in the output of `az ad sp create-for-rbac` and are used in service principal authentication. Record their values, but they can be retrieved at any point with az ad sp list.

Certificate-based authentication

For certificate-based authentication, use the `--cert` argument. This argument requires that you hold an existing certificate. Make sure any tool that uses this service principal has access to the certificate's private key. Certificates should be in an ASCII format such as PEM, CER, or DER. Pass the certificate as a string, or use the `@path` format to load the certificate from a file.

```
az ad sp create-for-rbac --name ServicePrincipalName --cert "-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----"
```

```
az ad sp create-for-rbac --name ServicePrincipalName --cert @/path/to/cert.pem
```

The `--keyvault` argument can be added to use a certificate in Azure Key Vault. In this case, the `--cert` value is the name of the certificate.

```
az ad sp create-for-rbac --name ServicePrincipalName --cert CertName --keyvault VaultName
```

To create a *self-signed* certificate for authentication, use the `--create-cert` argument:

```
az ad sp create-for-rbac --name ServicePrincipalName --create-cert
```

The `--keyvault` argument can be added to store the certificate in Azure Key Vault. When using `--keyvault`, the `--cert` argument is **required**.

```
az ad sp create-for-rbac --name ServicePrincipalName --create-cert --cert CertName --keyvault VaultName
```

Unless you store the certificate in Key Vault, the output includes the `fileWithCertAndPrivateKey` key. This key's value tells you where the generated certificate is stored. **Make sure** that you copy the certificate to a secure location, or you can't sign in with this service principal.

For certificates stored in Key Vault, retrieve the certificate's private key with az keyvault secret show. In Key Vault, the name of the certificate's secret is the same as the certificate name. If you lose access to a certificate's private key, reset the service principal credentials.

The `appId` and `tenant` keys appear in the output of `az ad sp create-for-rbac` and are used in service principal authentication. Record their values, but they can be retrieved at any point with az ad sp list.

Get an existing service principal

A list of the service principals in a tenant can be retrieved with az ad sp list. By default this command returns the first 100 service principals for your tenant. To get all of a tenant's service principals, use the `--all` argument. Getting this list can take a long time, so it's recommended that you filter the list with one of the following arguments:

- `--display-name` requests service principals that have a *prefix* that match the provided name. The display name of a service principal is the value set with the `--name` parameter during creation. If you didn't set `--name` during service principal creation, the name prefix is `azure-cli-`.
- `--spn` filters on exact service principal name matching. The service principal name always starts with `https://`. if the value you used for `--name` wasn't a URI, this value is `https://` followed by the display name.
- `--show-mine` requests only service principals created by the signed-in user.

- `--filter` takes an OData filter, and performs *server-side* filtering. This method is recommended over filtering client-side with the CLI's `--query` argument. To learn about OData filters, see OData expression syntax for filters.

The information returned for service principal objects is verbose. To get only the information necessary for sign-in, use the query string `[].{"id":"appId", "tenant":"appOwnerTenantId"}`. For example, to get the sign-in information for all service principals created by the currently logged in user:

```
az ad sp list --show-mine --query '[].{"id":"appId", "tenant":"appOwnerTenantId"}'
```

> **IMPORTANT**
>
> `az ad sp list` or az ad sp show get the user and tenant, but not any authentication secrets *or* the authentication method. Secrets for certificates in Key Vault can be retrieved with az keyvault secret show, but no other secrets are stored by default. If you forget an authentication method or secret, reset the service principal credentials.

Manage service principal roles

The Azure CLI has the following commands to manage role assignments:

- az role assignment list
- az role assignment create
- az role assignment delete

The default role for a service principal is **Contributor**. This role has full permissions to read and write to an Azure account. The **Reader** role is more restrictive, with read-only access. For more information on Role-Based Access Control (RBAC) and roles, see RBAC: Built-in roles.

This example adds the **Reader** role and removes the **Contributor** one:

```
az role assignment create --assignee APP_ID --role Reader
az role assignment delete --assignee APP_ID --role Contributor
```

> **NOTE**
>
> If your account doesn't have permission to assign a role, you see an error message that your account "does not have authorization to perform action 'Microsoft.Authorization/roleAssignments/write'." Contact your Azure Active Directory admin to manage roles.

Adding a role *doesn't* restrict previously assigned permissions. When restricting a service principal's permissions, the **Contributor** role should be removed.

The changes can be verified by listing the assigned roles:

```
az role assignment list --assignee APP_ID
```

Sign in using a service principal

Test the new service principal's credentials and permissions by signing in. To sign in with a service prinicpal, you need the `appId`, `tenant`, and credentials.

To sign in with a service principal using a password:

```
az login --service-principal --username APP_ID --password PASSWORD --tenant TENANT_ID
```

To sign in with a certificate, it must be available locally as a PEM or DER file, in ASCII format:

```
az login --service-principal --username APP_ID --tenant TENANT_ID --password /path/to/cert
```

To learn more about signing in with a service principal, see Sign in with the Azure CLI.

Reset credentials

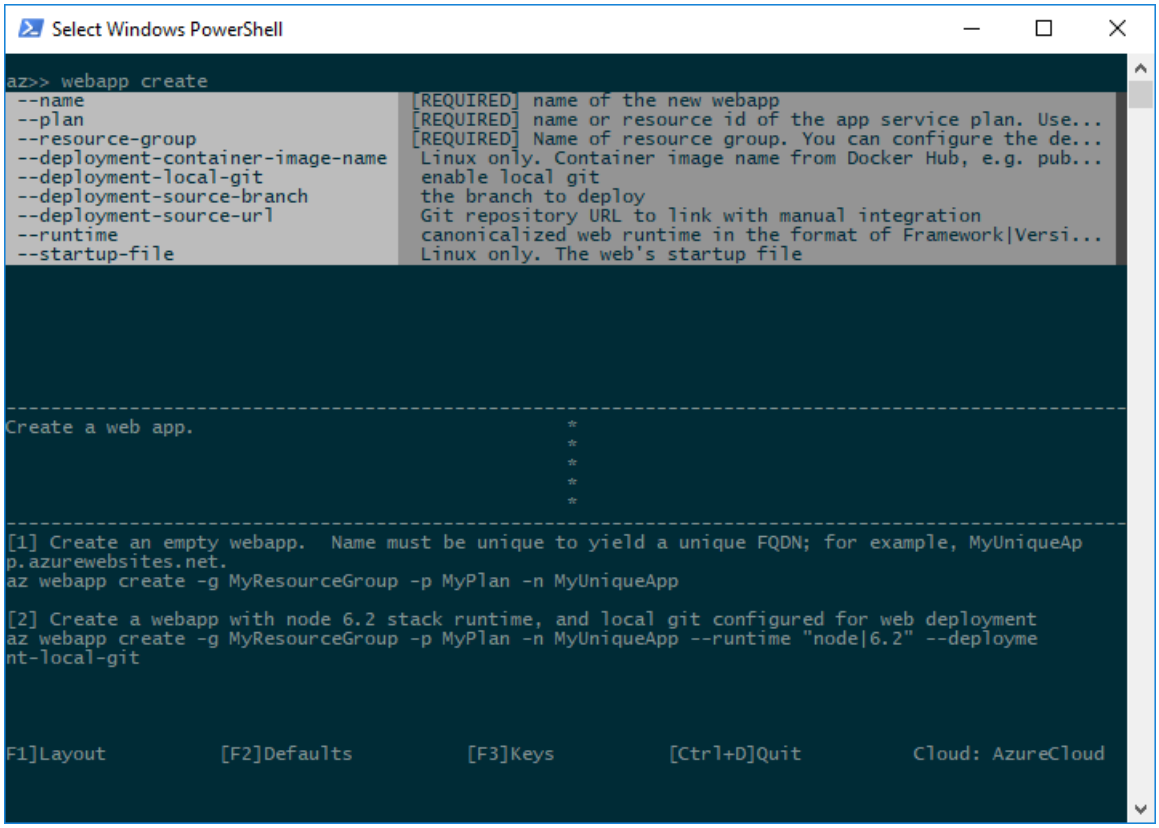
If you forget the credentials for a service principal, use az ad sp credential reset. The reset command takes the same arguments as `az ad sp create-for-rbac`.

```
az ad sp credential reset --name APP_ID
```

Azure CLI interactive mode

1/17/2019 • 2 minutes to read • Edit Online

You can use Azure CLI in interactive mode by running the `az interactive` command. This mode places you in an interactive shell with auto-completion, command descriptions, and examples.

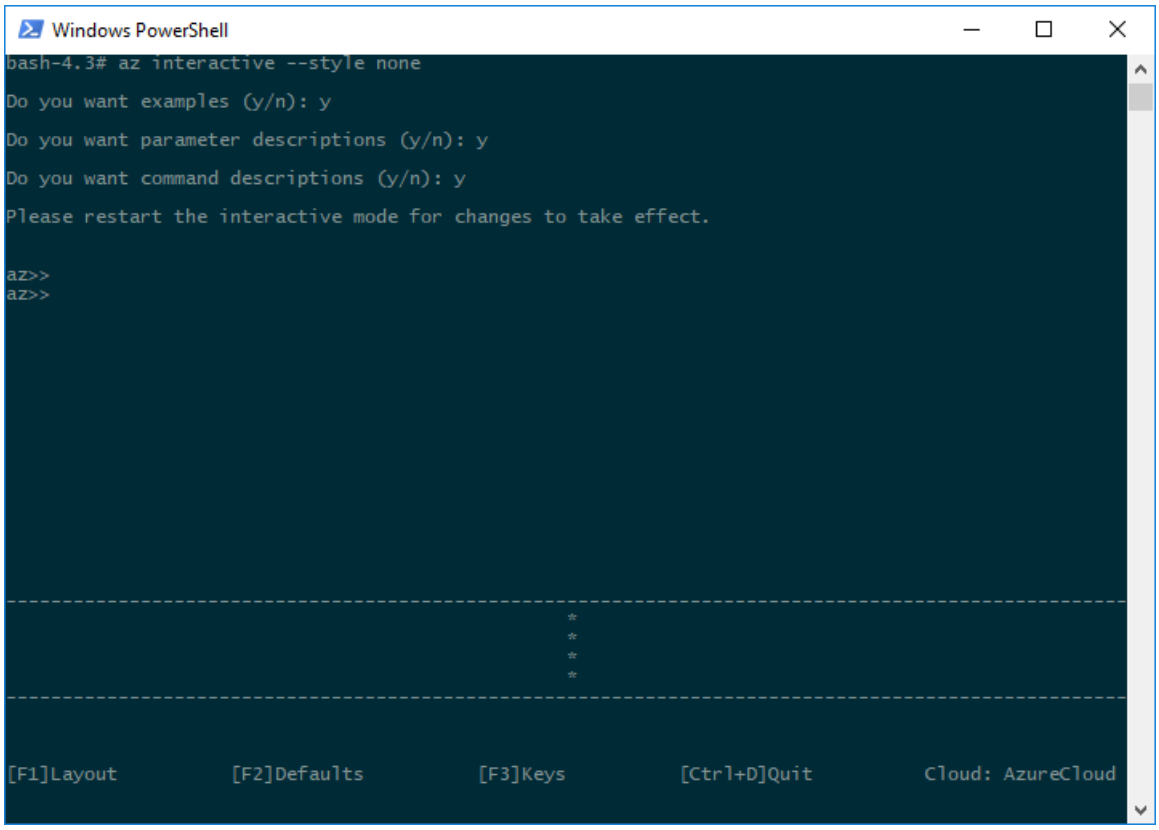


> **NOTE**
> We're not using the default style here, which doesn't read as well on a black background.

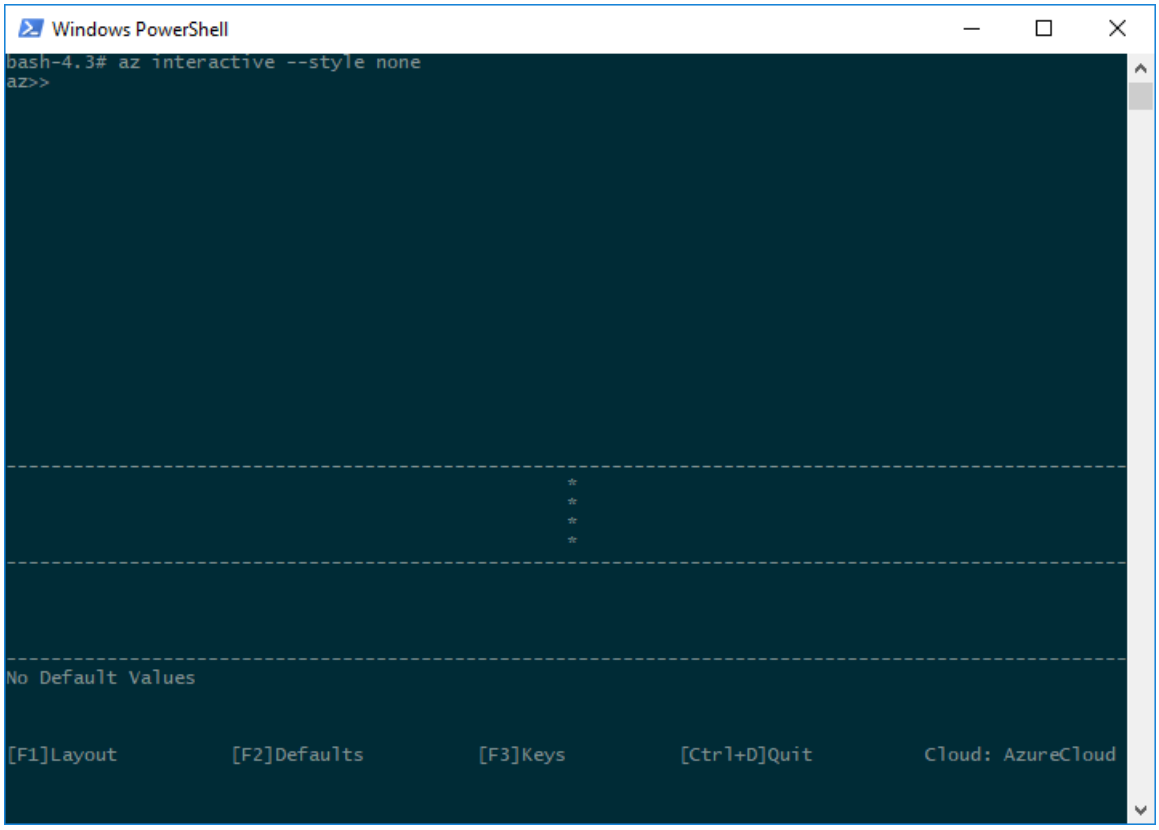
If you're not already signed in to your account, use the `login` command.

Configure

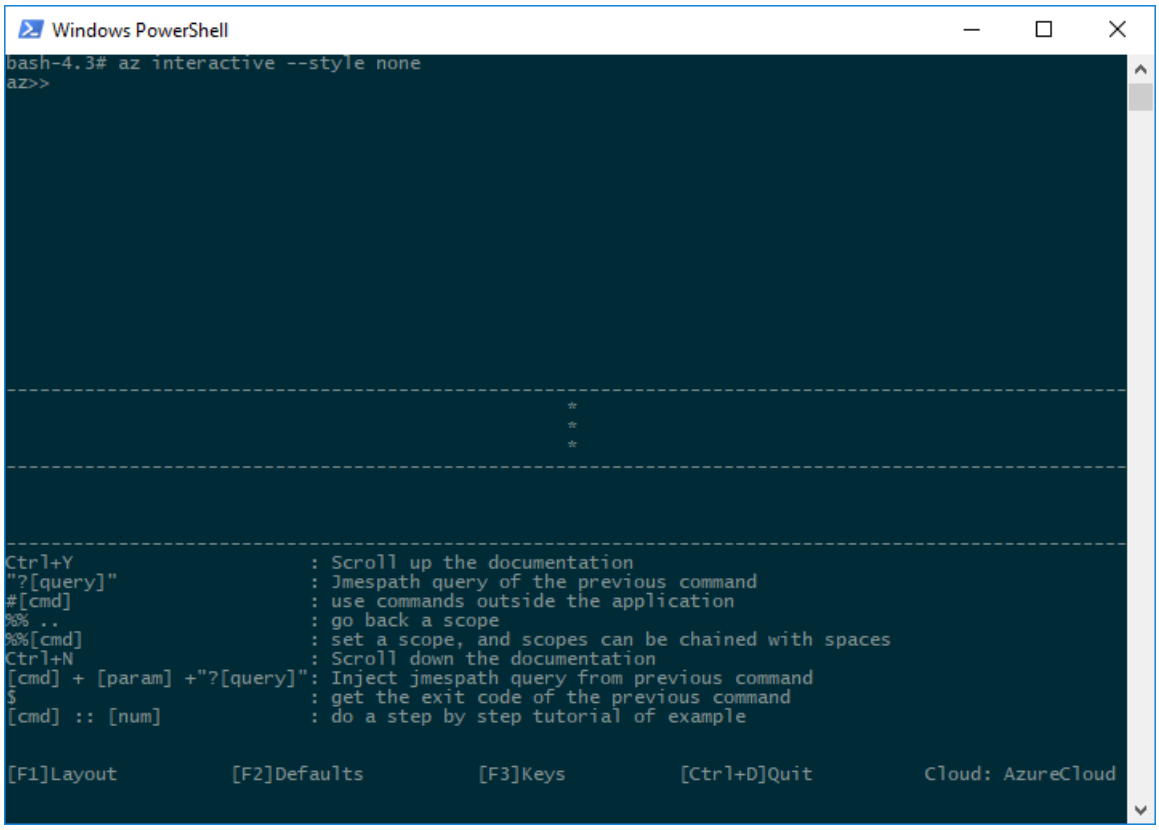
Interactive mode optionally displays command descriptions, parameter descriptions, and command examples. Turn descriptions and examples on or off using `F1`.



You can turn the display of parameter defaults on or off using `F2`.



`F3` toggles the display of some key gestures.



Scope

You can scope your interactive mode to a specific command group like `vm` or `vm image`. When you do, all commands are interpreted in that scope. It's a great shorthand if you're doing all your work in that command group.

Instead of typing these commands:

```
az>> vm create -n myVM -g myRG --image UbuntuLTS
az>> vm list -o table
```

You can scope to the vm command group and type these commands:

```
az>> %%vm
az vm>> create -n myVM -g myRG --image UbuntuLTS
az vm>>list -o table
```

You can scope to lower-level command groups as well. You could scope to `vm image` using `%%vm image`. In this case, since we're already scoped to `vm`, we would use `%%image`.

```
az vm>> %%image
az vm image>>
```

At that point, we can pop the scope back up to `vm` using `%%..`, or we can scope to the root with just `%%`.

```
az vm image>> %%
az>>
```

Query

You can execute a JMESPath query on the results of the last command that you executed. For example, after you create a VM, you can make sure it has fully provisioned.

```
az>> vm create --name myVM --resource-group myRG --image UbuntuLTS --no-wait -o json
az>> ? [*].provisioningState
```

```
[
  "Creating"
]
```

To learn more about querying the results of your commands, see Query command results with the Azure CLI.

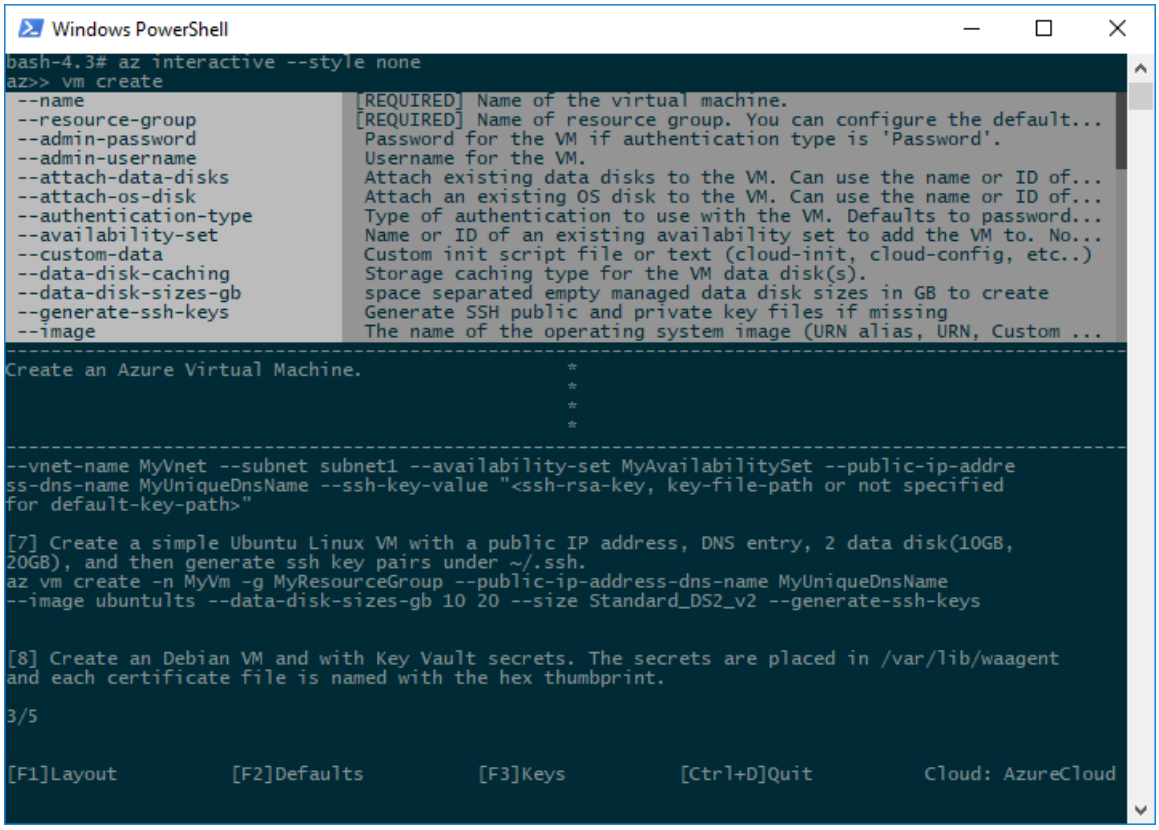
Bash commands

You can run shell commands without leaving interactive mode using `#[cmd]`.

```
az>> #dir
```

Examples

Some commands have lots of examples. You can scroll to the next page of examples using `CTRL-N` and the previous page using `CTRL-Y`.



You can also look at a specific example using `::#`.

```
az>> vm create ::8
```

Use multiple Azure subscriptions

1/23/2019 • 2 minutes to read • Edit Online

Most Azure users will only ever have a single subscription. However, if you are part of more than one organization or your organization has divided up access to certain resources across groupings, you may have multiple subscriptions within Azure. The CLI supports selecting a subscription both globally and per command.

Tenants, users, and subscriptions

You might have some confusion over the difference between tenants, users, and subscriptions within Azure. A *tenant* is the Azure Active Directory entity that encompasses a whole organization. This tenant has at least one *subscription* and *user*. A user is an individual and is associated with only one tenant, the organization that they belong to. Users are those accounts that sign in to Azure to create, manage, and use resources. A user may have access to multiple *subscriptions*, which are the agreements with Microsoft to use cloud services, including Azure. Every resource is associated with a subscription.

To learn more about the differences between tenants, users, and subscriptions, see the Azure cloud terminology dictionary. To learn how to add a new subscription to your Azure Active Directory tenant, see How to add an Azure subscription to Azure Active Directory. To learn how to sign in to a specific tenant, see Sign in with Azure CLI.

Change the active subscription

To access the resources for a subscription, switch your active subscription or use the `--subscription` argument. Switching your subscription for all commands is done with az account set.

To switch your active subscription:

1. Get a list of your subscriptions with the az account list command:

```
az account list --output table
```

2. Use `az account set` with the subscription ID or name you want to switch to.

```
az account set --subscription "My Demos"
```

To run only a single command with a different subscription, use the `--subscription` argument. This argument takes either a subscription ID or subscription name:

```
az vm create --subscription "My Demos" --resource-group MyGroup --name NewVM --image Ubuntu
```

Select clouds with the Azure CLI

2/20/2019 • 2 minutes to read • Edit Online

If you work across different regions or use Azure Stack, you may need to use more than one cloud. Microsoft provides clouds for compliance with regional laws, which are available for your use. This article shows you how to get information on clouds, change the current cloud, and register or unregister new clouds.

List available clouds

You can list available clouds with the az cloud list command. This command shows which cloud is currently active, what its current profile is, and information on regional suffixes and host names.

To get the active cloud and a list of all the available clouds:

```
az cloud list --output table
```

```
IsActive    Name                Profile
----------  ------------------  ---------
True        AzureCloud          latest
            AzureChinaCloud     latest
            AzureUSGovernment   latest
            AzureGermanCloud    latest
```

The currently active cloud has `True` in the `IsActive` column. Only one cloud can be active at any time. To get more detailed information on a cloud, including the endpoints that it uses for Azure services, use the `cloud show` command:

```
az cloud show --name AzureChinaCloud --output json
```

```
{
  "endpoints": {
    "activeDirectory": "https://login.chinacloudapi.cn",
    "activeDirectoryDataLakeResourceId": null,
    "activeDirectoryGraphResourceId": "https://graph.chinacloudapi.cn/",
    "activeDirectoryResourceId": "https://management.core.chinacloudapi.cn/",
    "batchResourceId": "https://batch.chinacloudapi.cn/",
    "gallery": "https://gallery.chinacloudapi.cn/",
    "management": "https://management.core.chinacloudapi.cn/",
    "resourceManager": "https://management.chinacloudapi.cn",
    "sqlManagement": "https://management.core.chinacloudapi.cn:8443/",
    "vmImageAliasDoc": "https://raw.githubusercontent.com/Azure/azure-rest-api-specs/master/arm-
compute/quickstart-templates/aliases.json"
  },
  "isActive": false,
  "name": "AzureChinaCloud",
  "profile": "latest",
  "suffixes": {
    "azureDatalakeAnalyticsCatalogAndJobEndpoint": null,
    "azureDatalakeStoreFileSystemEndpoint": null,
    "keyvaultDns": ".vault.azure.cn",
    "sqlServerHostname": ".database.chinacloudapi.cn",
    "storageEndpoint": "core.chinacloudapi.cn"
  }
}
```


Switch the active cloud

To switch the currently active cloud, run the az cloud set command. This command takes one required argument, the name of the cloud.

```
az cloud set --name AzureChinaCloud
```

IMPORTANT

If your authentication for the activated cloud has expired, you need to re-authenticate before performing any other CLI tasks. If this is your first time switching to the new cloud, you also need to set the active subscription. For instructions on authenticating, see Sign in with Azure CLI. For information on subscription management, see Manage Azure subscriptions with Azure CLI

Register a new cloud

Register a new cloud if you have your own endpoints for Azure Stack. Creating a cloud is done with the az cloud register command. This command requires a name and a set of service endpoints. To learn how to register a cloud for use with Azure Stack, see Use API version profiles with Azure CLI in Azure Stack.

You don't need to register information for the China, US Government, or German regions. These clouds are managed by Microsoft and available by default. For more information on all of the available endpoint settings, see the documentation for `az cloud register`.

Registering a cloud doesn't automatically switch to it. Use the `az cloud set` command to select the newly created cloud.

Update an existing cloud

If you have permissions, you can also update an existing cloud. Updating a cloud switches to a different Azure services profile or modifies the connection endpoints. Update a cloud with the az cloud update command, which

takes the same arguments as `az cloud register`.

Unregister a cloud

If you no longer need a created cloud, it can be unregistered with the az cloud unregister command:

```
az cloud unregister --name MyCloud
```

Output formats for Azure CLI commands

2/19/2019 • 4 minutes to read • Edit Online

The Azure CLI uses JSON as its default output format, but offers other formats. Use the `--output` (`--out` or `-o`) parameter to format CLI output. The argument values and types of output are:

--OUTPUT	DESCRIPTION
`json`	JSON string. This setting is the default.
`jsonc`	Colorized JSON.
`yaml`	YAML, a machine-readable alternative to JSON.
`table`	ASCII table with keys as column headings.
`tsv`	Tab-separated values, with no keys

JSON output format

The following example displays the list of virtual machines in your subscriptions in the default JSON format.

```
az vm list --output json
```

The following output has some fields omitted for brevity, and identifying information replaced.

```
[
  {
    "availabilitySet": null,
    "diagnosticsProfile": null,
    "hardwareProfile": {
      "vmSize": "Standard_DS1"
    },
    "id": "/subscriptions/.../resourceGroups/DEMORG1/providers/Microsoft.Compute/virtualMachines/DemoVM010",
    "instanceView": null,
    "licenseType": null,
    "location": "westus",
    "name": "DemoVM010",
    "networkProfile": {
      "networkInterfaces": [
        {
          "id":
"/subscriptions/.../resourceGroups/demorg1/providers/Microsoft.Network/networkInterfaces/DemoVM010VMNic",
          "primary": null,
          "resourceGroup": "demorg1"
        }
      ]
    },
          ...
          ...
          ...
]
```

YAML output format

The `yaml` format prints output as YAML, a plain-text data serialization format. YAML tends to be easier to read than JSON, and easily maps to that format. Some applications and CLI commands take YAML as configuration input, instead of JSON.

```
az vm list --out yaml
```

The following output has some fields omitted for brevity, and identifying information replaced.

```
- availabilitySet: null
  diagnosticsProfile: null
  hardwareProfile:
    vmSize: Standard_DS1_v2
  id: /subscriptions/.../resourceGroups/DEMORG1/providers/Microsoft.Compute/virtualMachines/DemoVM010
  identity: null
  instanceView: null
  licenseType: null
  location: westus
  name: ExampleVM1
  networkProfile:
    networkInterfaces:
    - id: /subscriptions/.../resourceGroups/DemoRG1/providers/Microsoft.Network/networkInterfaces/DemoVM010Nic
      primary: null
      resourceGroup: DemoRG1
  ...
...
```

Table output format

The `table` format prints output as an ASCII table, making it easy to read and scan. Nested objects aren't included in table output, but can still be filtered as part of a query. Some fields aren't included in the table, so this format is best when you want a quick, human-searchable overview of data.

```
az vm list --out table
```

```
Name         ResourceGroup    Location
-----------  ---------------  ----------
DemoVM010    DEMORG1          westus
demovm212    DEMORG1          westus
demovm213    DEMORG1          westus
KBDemo001VM  RGDEMO001        westus
KBDemo020    RGDEMO001        westus
```

You can use the `--query` parameter to customize the properties and columns you want to show in the list output. The following example shows how to select just the VM Name and the Resource Group Name in the `list` command.

```
az vm list --query "[].{resource:resourceGroup, name:name}" -o table
```

```
Resource    Name
----------  -----------
DEMORG1     DemoVM010
DEMORG1     demovm212
DEMORG1     demovm213
RGDEMO001   KBDemo001VM
RGDEMO001   KBDemo020
```

NOTE

Some keys are not printed in the table view by default. These are `id`, `type`, and `etag`. If you need to see these in your output, you can use the JMESPath re-keying feature to change the key name and avoid filtering.

```
az vm list --query "[].{objectID:id}" -o table
```

For more about using queries to filter data, see Use JMESPath queries with Azure CLI.

TSV output format

The `tsv` output format returns tab- and newline-separated values without additional formatting, keys, or other symbols. This format makes it easy to consume the output into other commands and tools that need to process the text in some form. Like the `table` format, `tsv` doesn't print nested objects.

Using the preceding example with the `tsv` option outputs the tab-separated result.

```
az vm list --out tsv
```

```
None    None
/subscriptions/.../resourceGroups/DEMORG1/providers/Microsoft.Compute/virtualMachines/DemoVM010    None
None    westus    DemoVM010            None    Succeeded    DEMORG1    None
Microsoft.Compute/virtualMachines    cbd56d9b-9340-44bc-a722-25f15b578444
None    None
/subscriptions/.../resourceGroups/DEMORG1/providers/Microsoft.Compute/virtualMachines/demovm212    None
None    westus    demovm212            None    Succeeded    DEMORG1    None
Microsoft.Compute/virtualMachines    4bdac85d-c2f7-410f-9907-ca7921d930b4
None    None
/subscriptions/.../resourceGroups/DEMORG1/providers/Microsoft.Compute/virtualMachines/demovm213    None
None    westus    demovm213            None    Succeeded    DEMORG1    None
Microsoft.Compute/virtualMachines    2131c664-221a-4b7f-9653-f6d542fbfa34
None    None
/subscriptions/.../resourceGroups/RGDEMO001/providers/Microsoft.Compute/virtualMachines/KBDemo001VM    None
None    westus    KBDemo001VM            None    Succeeded    RGDEMO001    None
Microsoft.Compute/virtualMachines    14e74761-c17e-4530-a7be-9e4ff06ea74b
None    None
/subscriptions/.../resourceGroups/RGDEMO001/providers/Microsoft.Compute/virtualMachines/KBDemo02None    None
westus    KBDemo020            None    Succeeded    RGDEMO001    None
Microsoft.Compute/virtualMachines    36baa9-9b80-48a8-b4a9-854c7a858ece
```

The next example shows how `tsv` output can be piped to other commands in bash. `grep` selects items that have text "RGD" in them, then the `cut` command selects the eighth field to show the name of the VM in output.

```
az vm list --out tsv | grep RGD | cut -f8
```

```
KBDemo001VM
KBDemo020
```

For the purposes of processing tab-separated fields, the values are in the same order that they appear in the printed JSON object. This order is guaranteed to be consistent between runs of the command.

Set the default output format

Use the interactive `az configure` command to set up your environment and establish default settings for output formats. The default output format is `json`.

```
az configure
```

```
Welcome to the Azure CLI! This command will guide you through logging in and setting some default values.

Your settings can be found at /home/defaultuser/.azure/config
Your current configuration is as follows:

  ...

Do you wish to change your settings? (y/N): y

What default output format would you like?
 [1] json - JSON formatted output that most closely matches API responses.
 [2] jsonc - Colored JSON formatted output that most closely matches API responses.
 [3] table - Human-readable output format.
 [4] tsv - Tab- and Newline-delimited. Great for GREP, AWK, etc.
 [5] yaml - YAML formatted output. An alternative to JSON. Great for configuration files.
 [6] none - No output, except for errors and warnings.
Please enter a choice [1]:
```

To learn more about configuring your environment, see Azure CLI configuration.

Query Azure CLI command output

1/23/2019 • 6 minutes to read • Edit Online

The Azure CLI uses the `--query` argument to execute a JMESPath query on the results of commands. JMESPath is a query language for JSON, giving you the ability to select and modify data from CLI output. Queries are executed on the JSON output before any display formatting.

The `--query` argument is supported by all commands in the Azure CLI. This article covers how to use the features of JMESPath with a series of small, simple examples.

Dictionary and list CLI results

Even when using an output format other than JSON, CLI command results are first treated as JSON for queries. CLI results are either a JSON array or dictionary. Arrays are sequences of objects that can be indexed, and dictionaries are unordered objects accessed with keys. Commands that *could* return more than one object return an array, and commands that *always* return *only* a single object return a dictionary.

Get properties in a dictionary

Working with dictionary results, you can access properties from the top level with just the key. The `.` (**subexpression**) character is used to access properties of nested dictionaries. Before introducing queries, take a look at the unmodified output of the `az vm show` command:

```
az vm show -g QueryDemo -n TestVM -o json
```

The command will output a dictionary. Some content has been omitted.

```
{
  "additionalCapabilities": null,
  "availabilitySet": null,
  "diagnosticsProfile": {
    "bootDiagnostics": {
      "enabled": true,
      "storageUri": "https://xxxxxx.blob.core.windows.net/"
    }
  },
  ...
  "osProfile": {
    "adminPassword": null,
    "adminUsername": "azureuser",
    "allowExtensionOperations": true,
    "computerName": "TestVM",
    "customData": null,
    "linuxConfiguration": {
      "disablePasswordAuthentication": true,
      "provisionVmAgent": true,
      "ssh": {
        "publicKeys": [
          {
            "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMobZNJTqgjWn/IB5xlilvE4Y+BMYpqkDnGRUcA0g9BYPgrGSQquCES37v2e3JmpfDPHFsaR+CPKlVr2
GoVJMMHeRcMJhj50ZWq0hAnkJBhlZVWy8S7dwdGAqPyPmWM2iJDCVMVrLITAJCno47O4Ees7RCH6ku7kU86b1NOanvrNwqTHr14wtnLhgZ0gQ5
GV1oLWvMEVg1YFMIgPRkTsSQKWCG5lLqQ45aU/4NMJoUxGyJTL9i8YxMavaB1Z2npfTQDQo9+womZ7SXzHaIWC858gWNl9e5UFyHDnTEDc14hK
kf1CqnGJVcCJkmSfmrrHk/CkmF0ZT3whTHO1DhJTtV stramer@contoso",
            "path": "/home/azureuser/.ssh/authorized_keys"
          }
        ]
      }
    },
    "secrets": [],
    "windowsConfiguration": null
  },
  ....
}
```

The following command gets the SSH public keys authorized to connect to the VM by adding a query:

```
az vm show -g QueryDemo -n TestVM --query osProfile.linuxConfiguration.ssh.publicKeys -o json
```

```
[
  {
    "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMobZNJTqgjWn/IB5xlilvE4Y+BMYpqkDnGRUcA0g9BYPgrGSQquCES37v2e3JmpfDPHFsaR+CPKlVr2
GoVJMMHeRcMJhj50ZWq0hAnkJBhlZVWy8S7dwdGAqPyPmWM2iJDCVMVrLITAJCno47O4Ees7RCH6ku7kU86b1NOanvrNwqTHr14wtnLhgZ0gQ5
GV1oLWvMEVg1YFMIgPRkTsSQKWCG5lLqQ45aU/4NMJoUxGyJTL9i8YxMavaB1Z2npfTQDQo9+womZ7SXzHaIWC858gWNl9e5UFyHDnTEDc14hK
kf1CqnGJVcCJkmSfmrrHk/CkmF0ZT3whTHO1DhJTtV stramer@contoso",
    "path": "/home/azureuser/.ssh/authorized_keys"
  }
]
```

To get more than one property, put expressions in square brackets `[ ]` (a **multiselect list**) as a comma-separated list. To get the VM name, admin user, and SSH key all at once use the command:

```
az vm show -g QueryDemo -n TestVM --query '[name, osProfile.adminUsername,
osProfile.linuxConfiguration.ssh.publicKeys[0].keyData]' -o json
```

```
[
  "TestVM",
  "azureuser",
  "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMobZNJTqgjWn/IB5xlilvE4Y+BMYpqkDnGRUcA0g9BYPgrGSQquCES37v2e3JmpfDPHFsaR+CPKlVr2
GoVJMMHeRcMJhj50ZWq0hAnkJBhlZVWy8S7dwdGAqPyPmWM2iJDCVMVrLITAJCno47O4Ees7RCH6ku7kU86b1NOanvrNwqTHr14wtnLhgZ0gQ5
GV1oLWvMEVg1YFMIgPRkTsSQKWCG5lLqQ45aU/4NMJoUxGyJTL9i8YxMavaB1Z2npfTQDQo9+womZ7SXzHaIWC858gWNl9e5UFyHDnTEDc14hK
kf1CqnGJVcCJkmSfmrrHk/CkmF0ZT3whTHO1DhJTtV stramer@contoso"
]
```

These values are listed in the result array in the order they were given in the query. Since the result is an array, there are no keys associated with the results.

Rename properties in a query

To get a dictionary instead of an array when querying for multiple values, use the `{ }` (**multiselect hash**) operator. The format for a multiselect hash is `{displayName:JMESPathExpression, ...}`. `displayName` will be the string shown in output, and `JMESPathExpression` is the JMESPath expression to evaluate. Modifying the example from the last section by changing the multiselect list to a hash:

```
az vm show -g QueryDemo -n TestVM --query '{VMName:name, admin:osProfile.adminUsername,
sshKey:osProfile.linuxConfiguration.ssh.publicKeys[0].keyData }' -o json
```

```
{
  "VMName": "TestVM",
  "admin": "azureuser",
  "ssh-key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMobZNJTqgjWn/IB5xlilvE4Y+BMYpqkDnGRUcA0g9BYPgrGSQquCES37v2e3JmpfDPHFsaR+CPKlVr2
GoVJMMHeRcMJhj50ZWq0hAnkJBhlZVWy8S7dwdGAqPyPmWM2iJDCVMVrLITAJCno47O4Ees7RCH6ku7kU86b1NOanvrNwqTHr14wtnLhgZ0gQ5
GV1oLWvMEVg1YFMIgPRkTsSQKWCG5lLqQ45aU/4NMJoUxGyJTL9i8YxMavaB1Z2npfTQDQo9+womZ7SXzHaIWC858gWNl9e5UFyHDnTEDc14hK
kf1CqnGJVcCJkmSfmrrHk/CkmF0ZT3whTHO1DhJTtV stramer@contoso"
}
```

Get properties in an array

An array has no properties of its own, but it can be indexed. This feature is shown in the last example with the expression `publicKeys[0]`, which gets the first element of the `publicKeys` array. There's no guarantee CLI output is ordered, so avoid using indexing unless you're sure of the order or don't care what element you get. To access the properties of elements in an array, you do one of two operations: *flattening* and *filtering*. This section covers how to flatten an array.

Flattening an array is done with the `[]` JMESPath operator. All expressions after the `[]` operator are applied to each element in the current array. If `[]` appears at the start of the query, it flattens the CLI command result. The results of `az vm list` can be inspected with this feature. To get the name, OS, and administrator name for each VM in a resource group:

```
az vm list -g QueryDemo --query '[].{Name:name, OS:storageProfile.osDisk.osType,
admin:osProfile.adminUsername}' -o json
```

```
[
  {
    "Name": "Test-2",
    "OS": "Linux",
    "admin": "sttramer"
  },
  {
    "Name": "TestVM",
    "OS": "Linux",
    "admin": "azureuser"
  },
  {
    "Name": "WinTest",
    "OS": "Windows",
    "admin": "winadmin"
  }
]
```

When combined with the `--output table` output format, the column names match up with the `displayKey` value of the multiselect hash:

```
az vm list -g QueryDemo --query '[].{Name:name, OS:storageProfile.osDisk.osType,
Admin:osProfile.adminUsername}' --output table
```

```
Name     OS       Admin
-------  -------  ---------
Test-2   Linux    sttramer
TestVM   Linux    azureuser
WinTest  Windows  winadmin
```

NOTE

Certain keys are filtered out and not printed in the table view. These keys are `id`, `type`, and `etag`. To see these values, you can change the key name in a multiselect hash.

```
az vm show -g QueryDemo -n TestVM --query "{objectID:id}" -o table
```

Any array can be flattened, not just the top-level result returned by the command. In the last section, the expression `osProfile.linuxConfiguration.ssh.publicKeys[0].keyData` was used to get the SSH public key for sign-in. To get *every* SSH public key, the expression could instead be written as `osProfile.linuxConfiguration.ssh.publicKeys[].keyData`. This query expression flattens the `osProfile.linuxConfiguration.ssh.publicKeys` array, and then runs the `keyData` expression on each element:

```
az vm show -g QueryDemo -n TestVM --query '{VMName:name, admin:osProfile.adminUsername,
sshKeys:osProfile.linuxConfiguration.ssh.publicKeys[].keyData }' -o json
```

```
{
  "VMName": "TestVM",
  "admin": "azureuser",
  "sshKeys": [
    "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMobZNJTqgjWn/IB5xlilvE4Y+BMYpqkDnGRUcA0g9BYPgrGSQquCES37v2e3JmpfDPHFsaR+CPKlVr2
GoVJMMHeRcMJhj50ZWq0hAnkJBhlZVWy8S7dwdGAqPyPmWM2iJDCVMVrLITAJCno47O4Ees7RCH6ku7kU86b1NOanvrNwqTHr14wtnLhgZ0gQ5
GV1oLWvMEVg1YFMIgPRkTsSQKWCG5lLqQ45aU/4NMJoUxGyJTL9i8YxMavaB1Z2npfTQDQo9+womZ7SXzHaIWC858gWNl9e5UFyHDnTEDc14hK
kf1CqnGJVcCJkmSfmrrHk/CkmF0ZT3whTHO1DhJTtV stramer@contoso\n"
  ]
}
```

Filter arrays

The other operation used to get data from an array is *filtering*. Filtering is done with the `[?...]` JMESPath operator. This operator takes a predicate as its contents. A predicate is any statement that can be evaluated to either `true` or `false`. Expressions where the predicate evaluates to `true` are included in the output.

JMESPath offers the standard comparison and logical operators. These include `<`, `<=`, `>`, `>=`, `==`, and `!=`. JMESPath also supports logical and (`&&`), or (`||`), and not (`!`). Expressions can be grouped within parenthesis, allowing for more complex predicate expressions. For the full details on predicates and logical operations, see the JMESPath specification.

In the last section, we flattened an array to get the complete list of all VMs in a resource group. Using filters, this output can be restricted to only Linux VMs:

```
az vm list -g QueryDemo --query "[?storageProfile.osDisk.osType=='Linux'].{Name:name,
admin:osProfile.adminUsername}" --output table
```

```
Name    Admin
------  ---------
Test-2  sttramer
TestVM  azureuser
```

IMPORTANT

In JMESPath, strings are always surrounded by single quotes (`'`). If you use double quotes as part of a string in a filter predicate, you'll get empty output.

JMESPath also has built-in functions that can help with filtering. One such function is `contains(string, substring)`, which checks to see if a string contains a substring. Expressions are evaluated before calling the function, so the first argument can be a full JMESPath expression. The next example finds all VMs using SSD storage for their OS disk:

```
az vm list -g QueryDemo --query "[?contains(storageProfile.osDisk.managedDisk.storageAccountType,'SSD')].
{Name:name, Storage:storageProfile.osDisk.managedDisk.storageAccountType}" -o json
```

```
[
  {
    "Name": "TestVM",
    "Storage": "StandardSSD_LRS"
  },
  {
    "Name": "WinTest",
    "Storage": "StandardSSD_LRS"
  }
]
```

This query is a little long. The `storageProfile.osDisk.managedDisk.storageAccountType` key is mentioned twice, and rekeyed in the output. One way to shorten it is to apply the filter after flattening and selecting data.

```
az vm list -g QueryDemo --query "[].{Name:name, Storage:storageProfile.osDisk.managedDisk.storageAccountType}
[?contains(Storage,'SSD')]" -o json
```

```
[
  {
    "Name": "TestVM",
    "Storage": "StandardSSD_LRS"
  },
  {
    "Name": "WinTest",
    "Storage": "StandardSSD_LRS"
  }
]
```

For large arrays, it may be faster to apply the filter before selecting data.

See the JMESPath specification - Built-in Functions for the full list of functions.

Change output

JMESPath functions also have another purpose, which is to operate on the results of a query. Any function that returns a non-boolean value changes the result of an expression. For example, you can sort data by a property value with `sort_by(array, &sort_expression)`. JMESPath uses a special operator, `&`, for expressions that should be evaluated later as part of a function. The next example shows how to sort a VM list by OS disk size:

```
az vm list -g QueryDemo --query "sort_by([].{Name:name, Size:storageProfile.osDisk.diskSizeGb}, &Size)" --
output table
```

```
Name     Size
-------  ------
TestVM   30
Test-2   32
WinTest  127
```

See the JMESPath specification - Built-in Functions for the full list of functions.

Experiment with queries interactively

To start experimenting with JMESPath, the JMESPath-terminal Python package offers an interactive environment to work with queries. Data is piped as input, and then queries are written and run in the editor.

```
pip install jmespath-terminal
az vm list --output json | jpterm
```

Use extensions with Azure CLI

1/17/2019 • 2 minutes to read • Edit Online

The Azure CLI offers the capability to load extensions. Extensions are Python wheels that aren't shipped as part of the CLI but run as CLI commands. With extensions, you gain access to experimental and pre-release commands along with the ability to write your own CLI interfaces. This article covers how to manage extensions and answers common questions about their use.

Find extensions

To see the extensions provided and maintained by Microsoft, use the az extension list-available command.

```
az extension list-available --output table
```

We also host a list of extensions on the documentation site.

Install extensions

Once you have found an extension to install, use az extension add to get it. If the extension is listed in `az extension list-available`, you can install the extension by name.

```
az extension add --name <extension-name>
```

If the extension is from an external resource or you have a direct link to it, provide the source URL or local path. The extension *must* be a compiled Python wheel file.

```
az extension add --source <URL-or-path>
```

Once an extension is installed, it's found under the value of the `$AZURE_EXTENSION_DIR` shell variable. If this variable is unset, by default the value is `$HOME/.azure/cliextensions` on Linux and macOS, and `%USERPROFILE%\.azure\cliextensions` on Windows.

Update extensions

If an extension was installed by name, update it using az extension update.

```
az extension update --name <extension-name>
```

Otherwise, an extension can be updated from source by following the Install extensions instructions.

If an extension name can't be resolved by the CLI, uninstall it and attempt to reinstall. The extension could also have become part of the base CLI. Try updating the CLI as described in Install the Azure CLI and see if the extension's commands were added.

Uninstall extensions

If you no longer need an extension, remove it with az extension remove.

```
az extension remove --name <extension-name>
```

You can also remove an extension manually by deleting it from the location where it was installed. The `$AZURE_EXTENSION_DIR` shell variable defines where modules are installed. If this variable is unset, by default the value is `$HOME/.azure/cliextensions` on Linux and macOS, and `%USERPROFILE%\.azure\cliextensions` on Windows.

```
rm -rf $AZURE_EXTENSION_DIR/<extension-name>
```

FAQ

Here are some answers to other common questions about CLI extensions.

What file formats are allowed for installation?

Currently, only compiled Python wheels can be installed as extensions.

Can extensions replace existing commands?

Yes. Extensions may replace existing commands, but before running a command that has been replaced the CLI will issue a warning.

How can I tell if an extension is in pre-release?

An extension's documentation and versioning will show if it's in pre-release. Microsoft often releases preview commands as CLI extensions, with the option of moving them into the main CLI product later. When commands are moved out of extensions, the old extension should be uninstalled.

Can extensions depend upon each other?

No. Since the CLI doesn't guarantee a load order, dependencies might not be satisfied. Removing an extension won't affect any others.

Are extensions updated along with the CLI?

No. Extensions must be updated separately, as described in Update extensions.

Available extensions for the Azure CLI

3/25/2019 • 2 minutes to read • Edit Online

This article is a complete list of the available extensions for the Azure CLI which are supported by Microsoft.

The list of extensions is also available from the CLI. To get it, run az extension list-available:

```
az extension list-available --output table
```

NAME	VERSION	SUMMARY	PREVIEW
aem	0.1.1	Manage Azure Enhanced Monitoring Extensions for SAP	
aks-preview	0.2.3	Provides a preview for upcoming AKS features	Yes
alias	0.5.2	Support for command aliases	Yes
anf-preview	0.1.0	Provides a preview for upcoming Azure NetApp Files (ANF) features.	Yes
appconfig	0.4.0	Provides a preview for upcoming App Configuration features.	Yes
application-insights	0.1.0	Support for querying Azure Application Insights metrics, events, and logs.	Yes
azure-batch-cli-extensions	3.0.3	Additional commands for working with Azure Batch service	
azure-cli-iot-ext	0.7.0	Provides the data plane command layer for Azure IoT Hub, IoT Edge and IoT Device Provisioning Service	
azure-devops	0.3.0	Tools for managing Azure DevOps.	Yes
azure-firewall	0.1.1	Manage Azure Firewall resources.	Yes
botservice	0.4.3	Bug fixes for issues in the native botservice cli command module.	Yes

NAME	VERSION	SUMMARY	PREVIEW
db-up	0.1.10	Additional commands to simplify Azure Database workflows.	Yes
dev-spaces-preview	0.1.6	Dev Spaces provides a rapid, iterative Kubernetes development experience for teams.	Yes
dms-preview	0.8.0	Support for new Database Migration Service scenarios.	Yes
dns	0.0.2	An Azure CLI Extension for DNS zones	
eventgrid	0.4.0	Support for Azure EventGrid 2018-09-15-preview features	Yes
express-route	0.1.3	Manage ExpressRoutes with preview features.	Yes
express-route-cross-connection	0.1.0	Manage customer ExpressRoute circuits using an ExpressRoute cross-connection.	
find	0.3.0	Intelligent querying for CLI information.	Yes
front-door	0.1.3	Manage networking Front Doors.	Yes
image-copy-extension	0.0.9	Support for copying managed vm images between regions	
interactive	0.4.1	Microsoft Azure Command-Line Interactive Shell	Yes
keyvault-preview	0.1.3	Preview Azure Key Vault commands.	Yes
log-analytics	0.1.3	Support for Azure Log Analytics query capabilities.	Yes
managementgroups	0.1.0	An Azure CLI Extension for Management Groups	
managementpartner	0.1.2	Support for Management Partner preview	
mesh	0.10.4	Support for Microsoft Azure Service Fabric Mesh - Public Preview	Yes

NAME	VERSION	SUMMARY	PREVIEW
mixed-reality	0.0.1	Mixed Reality Azure CLI Extension.	
privatedns	0.1.0	Commands to manage Private DNS Zones	Yes
rdbms-vnet	10.0.0	Support for Virtual Network rules in Azure MySQL and Azure PostgreSQL resources	
resource-graph	0.1.8	Support for querying Azure resources with Resource Graph.	Yes
sap-hana	0.3.4	Additional commands for working with SAP HanaOnAzure instances.	
signalr	0.1.0	Support for signalr management preview.	Yes
sqlvm-preview	0.1.0	Tools for managing SQL virtual machines, groups and availability group listeners.	Yes
storage-preview	0.2.3	Provides a preview for upcoming storage features.	Yes
subscription	0.1.1	Support for subscription management preview.	
virtual-network-tap	0.1.0	Manage virtual network taps (VTAP).	Yes
virtual-wan	0.1.0	Manage virtual WAN, hubs, VPN gateways and VPN sites.	Yes
webapp	0.2.16	Additional commands for Azure AppService.	Yes

The Azure CLI alias extension

1/17/2019 • 4 minutes to read • Edit Online

The alias extension allows users to define custom commands for the Azure CLI by using existing commands. Aliases help keep your workflow simple by allowing shortcuts. Since aliases are powered by the Jinja2 template engine, they even offer advanced argument processing.

> **NOTE**
> The Alias Extension is in public preview. The features and configuration file format may change.

Install the alias extension

The minimum required Azure CLI version to use the alias extension is **2.0.28**. To check your CLI version, run `az --version`. If you need to update your installation, follow the instructions in Install the Azure CLI.

Install the extension with the az extension add command.

```
az extension add --name alias
```

Verify the installation of the extension with az extension list. If the alias extension was installed properly, it's listed in the command output.

```
az extension list --output table --query '[].{Name:name}'
```

```
Name
------
alias
```

Keep the extension up-to-date

The alias extension is under active development and new versions are released regularly. New versions aren't installed when you update the CLI. Install the updates for the extension with az extension update.

```
az extension update --name alias
```

Manage aliases for the Azure CLI

The alias extension lets you create and manage aliases for other CLI commands. To view all the available commands and parameter details, run the alias command with `--help`.

```
az alias --help
```

Create simple alias commands

One use of aliases is for shortening existing command groups or command names. For example, you can shorten

the `group` command group to `rg` and the `list` command to `ls`.

```
az alias create --name rg --command group
az alias create --name ls --command list
```

These newly defined aliases can now be used anywhere that their definition would be.

```
az rg list
az rg ls
az vm ls
```

Do not include `az` as part of the command.

Aliases can also be shortcuts for complete commands. The next example lists available resource groups and their locations in table output:

```
az alias create --name ls-groups --command "group list --query '[].{Name:name, Location:location}' --output
table"
```

Now `ls-groups` can be run like any other CLI command.

```
az ls-groups
```

Create an alias command with arguments

You can also add positional arguments to an alias command by including them as `{{ arg_name }}` in the alias name. The whitespace inside the braces is required.

```
az alias create --name "alias_name {{ arg1 }} {{ arg2 }} ..." --command "invoke_including_args"
```

The next example alias shows how to use positional arguments to get the public IP address for a VM.

```
az alias create \
    --name "get-vm-ip {{ resourceGroup }} {{ vmName }}" \
    --command "vm list-ip-addresses --resource-group {{ resourceGroup }} --name {{ vmName }}
        --query [0].virtualMachine.network.publicIpAddresses[0].ipAddress"
```

When running this command, you give values to the positional arguments.

```
az get-vm-ip MyResourceGroup MyVM
```

You can also use environment variables in aliased commands, which are evaluated at runtime. The next example adds the `create-rg` alias, which creates a resource group in `eastus` and adds an `owner` tag. This tag is assigned the value of the local environment variable `USER`.

```
az alias create \
    --name "create-rg {{ groupName }}" \
    --command "group create --name {{ groupName }} --location eastus --tags owner=\$USER"
```

To register the environment variables inside the command of the alias, the dollar sign `$` must be escaped.

Process arguments using Jinja2 templates

Argument substitution in the alias extension is performed by Jinja2. Jinja2 templates allow for manipulating the arguments.

With Jinja2 templates, you can write aliases that take different types of arguments than the underlying command. For example, you can make an alias that takes a storage URL. Then this URL is parsed to pass the account and container names to the storage command.

```
az alias create \
    --name 'storage-ls {{ url }}' \
    --command "storage blob list
        --account-name {{ url.replace('https://', '').split('.')[0] }}
        --container-name {{ url.replace('https://', '').split('/')[1] }}"
```

To learn about the Jinja2 template engine, see the Jinja2 documentation.

Alias configuration file

Another way to create and modify aliases is to alter the alias configuration file. Alias command definitions are written into a configuration file, located at `$AZURE_USER_CONFIG/alias`. The default value of `AZURE_USER_CONFIG` is `$HOME/.azure` on macOS and Linux, and `%USERPROFILE%\.azure` on Windows. The alias configuration file is written in the INI configuration file format. The format for alias commands is:

```
[alias_name]
command = invoked_commands
```

For aliases that have positional arguments, the format for alias commands is:

```
[alias_name {{ arg1 }} {{ arg2 }} ...]
command = invoked_commands_including_args
```

Create an alias command with arguments via the alias configuration file

The next example shows an alias for a command with arguments. This command gets the public IP address for a VM. Aliased commands must all be on a single line, and use all of the arguments in the alias name.

```
[get-vm-ip {{ resourceGroup }} {{ vmName }}]
command = vm list-ip-addresses --resource-group {{ resourceGroup }} --name {{ vmName }} --query
[0].virtualMachine.network.publicIpAddresses[0].ipAddress
```

Uninstall the alias extension

To uninstall the extension, use the az extension remove command.

```
az extension remove --name alias
```

If you uninstalled because a bug or other problem with the extension, file a GitHub issue so that we can provide a fix.

Azure CLI configuration

1/17/2019 • 3 minutes to read • Edit Online

The Azure CLI allows for user configuration for settings such as logging, data collection, and default argument values. The CLI offers a convenience command for managing some defaults, `az configure`. Other values can be set in a configuration file or with environment variables.

Configuration values used by the CLI are evaluated in the following precedence, with items higher on the list taking priority.

1. Command-line parameters
2. Environment variables
3. Values in the configuration file or set with `az configure`

CLI configuration with az configure

You set defaults for the CLI with the az configure command. This command takes one argument, `--defaults`, which is a space-separated list of `key=value` pairs. The provided values are used by the CLI in place of required arguments.

The following table contains a list of available configuration keys.

NAME	DESCRIPTION
group	The default resource group to use for all commands.
location	The default location to use for all commands.
web	The default app name to use for `az webapp` commands.
vm	The default VM name to use for `az vm` commands.
vmss	The default virtual machine scale set (VMSS) name to use for `az vmss` commands.
acr	The default container registry name to use for `az acr` commands.

As an example, here's how you would set the default resource group and location for all commands.

```
az configure --defaults location=westus2 group=MyResourceGroup
```

CLI configuration file

The CLI configuration file contains other settings that are used for managing CLI behavior. The configuration file itself is located at `$AZURE_CONFIG_DIR/config`. The default value of `AZURE_CONFIG_DIR` is `$HOME/.azure` on Linux and macOS, and `%USERPROFILE%\.azure` on Windows.

Configuration files are written in the INI file format. This file format is defined by section headers, followed by a list of key-value entries.

- Section headers are written as `[section-name]`. Section names are case-sensitive.
- Entries are written as `key=value`. Key names are not case-sensitive.
- Comments are any line that begins with a `#` or `;`. Inline comments are not allowed.

Booleans are case-insensitive, and are represented by the following values.

- **True**: 1, yes, true, on
- **False**: 0, no, false, off

Here's an example of a CLI configuration file that disables any confirmation prompts and sets up logging to the `/var/log/azure` directory.

```
[core]
disable_confirm_prompt=Yes

[logging]
enable_log_file=yes
log_dir=/var/log/azure
```

See the next section for details on all of the available configuration values and what they mean. For the full details on the INI file format, see the Python documentation on INI.

CLI configuration values and environment variables

The following table contains all of the sections and option names that can be placed in a configuration file. Their corresponding environment variables are set as `AZURE_{section}_{name}`, in all caps. For example, the `storage_account` default for `batchai` is set in the `AZURE_BATCHAI_STORAGE_ACCOUNT` variable.

When you provide a default value, that argument is no longer required by any command. Instead, the default value is used.

SECTION	NAME	TYPE	DESCRIPTION
core	output	string	The default output format. Can be one of `json`, `jsonc`, `tsv`, or `table`.
	disable_confirm_prompt	boolean	Turn confirmation prompts on/off.
	collect_telemetry	boolean	Allow Microsoft to collect anonymous data on the usage of the CLI. For privacy information, see the Azure CLI Terms of Use.
logging	enable_log_file	boolean	Turn logging on/off.
	log_dir	string	The directory to write logs to. By default this value is `${AZURE_CONFIG_DIR}/logs`.
storage	connection_string	string	The default connection string to use for `az storage` commands.

SECTION	NAME	TYPE	DESCRIPTION
	account	string	The default account name to use for `az storage` commands.
	key	string	The default account key to use for `az storage` commands.
	sas_token	string	The default SAS token to use for `az storage` commands.
batchai	storage_account	string	The default storage account to use for `az batchai` commands.
	storage_key	string	The default storage key to use for `az batchai` commands.
batch	account	string	The default Azure Batch account name to use for `az batch` commands.
	access_key	string	The default access key to use for `az batch` commands. Only used with `aad` authorization.
	endpoint	string	The default endpoint to connect to for `az batch` commands.
	auth_mode	string	The authorization mode to use for `az batch` commands. Can be `shared_key` or `aad`.

NOTE

You may see other values in your configuration file, but these are managed directly through CLI commands, including `az configure`. The ones listed in the table above are the only values you should change yourself.

Azure CLI release notes

3/18/2019 • 77 minutes to read • Edit Online

March 12, 2019

Version 2.0.60

Core

- Fixed an incorrect error in `cloud set` about subscription not found

ACR

- Fixed redundant sources in image import

ACS

- Changed to ignore `--listen-address` argument to `aks browse` if `kubectl` doesn't support it

AppService

- Added `[webapp|functionapp] deployment list-publishing-credentials` to get the Kudu publishing url and its credentials
- Removed erroneous print statement for `webapp auth update`
- Fixed `functionapp` to set the correct image for runtime in Linux App Service plans
- Removed preview tag for `webapp up` and added improvements to the command

Botservice

- Added `SCM_DO_BUILD_DURING_DEPLOYMENT` to ARM template's Application Settings for v4 Web App Bots
- Added `Microsoft-BotFramework-AppId` and `Microsoft-BotFramework-AppPassword` to ARM template's Application Settings for v4 Web App Bots
- Removed single quotes from `bot publish` command output at end of `bot create`
- Changed `bot publish` to be asynchronous

Container

- Added `--no-wait` argument to `container [start|restart]`

EventHub

- Added `--skip-empty-archives` flag to `eventhub create|update` to support empty archives in capture

Find

- Major functionality update

HDInsight

- Added the `--storage-account-managed-identity` parameter to `hdinsight create` to support ADLS Gen2 MSI

Network

- Fixed issue with `vpn-connection update` where updating a VPN connection between gateways in different subscriptions would fail

Rdbms

- Minor fixes to get default location from resource group when not provided for creating servers and add validation for retention days

Role

- Fixed `role definition update` to use ID to resolve definition correctly
- Changed `ad app credential reset` to remove the assumption that app's service principal always exists

Service Fabric

- Fixed issue with `sf cluster list` was not iterable

February 26, 2019

Version 2.0.59

Core

- Fixed issue where in some instances using `--subscription NAME` would throw an exception

ACR

- Added `--target` parameter for `acr build`, `acr task create` and `acr task update` commands
- Improved error handling for runtime commands when not logged into Azure

ACS

- Added `--listen-address` option to `aks port-forward`

AppService

- Added `functionapp devops-build` command

Batch

- [BREAKING CHANGE] Removed the `batch pool upgrade os` command
- [BREAKING CHANGE] Removed the `Pacakges` property from `Application` responses
- Added the `batch application package list` command to list packages of an application
- [BREAKING CHANGE] Changed `--application-id` to `--application-name` in all `batch application` commands,
- Added the `--json-file` argument to commands for requesting the raw API response
- Updated validation to automatically include `https://` in all endpoints if missing

CosmosDB

- Added `network-rule` subgroup with commands `add`, `remove`, and `list` for managing VNET rules of a Cosmos DB account

Kusto

- [BREAKING CHANGE] Changed `hot_cache_period` and `soft_delete_period` types for database to ISO8601 duration format

Network

- Added `--express-route-gateway-bypass` argument to `vpn-connection [create|update]`
- Added command groups from `express-route` extensions
- Added `express-route gateway` and `express-route port` command groups
- Added argument `--legacy-mode` to `express-route peering [create|update]`
- Added arguments `--allow-classic-operations` and `--express-route-port` to `express-route [create|update]`
- Added `--gateway-default-site` argument to `vnet-gateway [create|update]`
- Added `ipsec-policy` commands to `vnet-gateway`

Resource

- Fixed issue with `deployment create` where type field was case-sensitive

- Added support for URI-based parameters file to `policy assignment create`
- Added support for URI-based parameters and definitions to `policy set-definition update`
- Fixed handling of parameters and rules for `policy definition update`
- Fixed issue with `resource show/update/delete/tag/invoke-action` where cross-subscription IDs did not properly honor the subscription ID

Role

- Added support for app roles to `ad app [create|update]`

VM

- Fixed issue with `vm create where` --accelerated-networking` was not enabled by default for Ubuntu 18.0

February 12, 2019

Version 2.0.58

Core

- `az --version` now displays a notification if you have packages that can be updated
- Fixed regression where `--ids` could no longer be used with JSON output

ACR

- [BREAKING CHANGE] Removed `acr build-task` command group
- [BREAKING CHANGE] Removed `--tag` and `--manifest` options from from `acr repository delete`

ACS

- Added support for case-insensitive names to `aks [enable-addons|disable-addons]`
- Added support for Azure Active Directory updating operation using `aks update-credentials --reset-aad`
- Added clarification that `--output` is ignored for `aks get-credentials`

AMS

- Added `ams streaming-endpoint [start | stop | create | update] wait` commands
- Added `ams live-event [create | start | stop | reset] wait` commands

Appservice

- Added ability to create and configure functions using ACR containers
- Added support for updating webapp configurations through json
- Improved help for `appservice-plan-update`
- Added support for app insights on functionapp create
- Fixed issues with webapp SSH

Botservice

- Improved UX for `bot publish`
- Added warning for timeouts when running `npm install` during `az bot publish`
- Removed invalid char `.` from `--name` in `az bot create`
- Changed to stop randomizing resource names when creating Azure Storage, App Service Plan, Function/Web App and Application Insights
- [DEPRECATED] Deprecated `--proj-name` argument in favor of `--proj-file-path`
- Changed `az bot publish` to remove fetched IIS Node.js deployment files if they did not already exist
- Added `--keep-node-modules` argument to `az bot publish` to not delete `node_modules` folder on App Service
- Added `"publishCommand"` key-value pair to output from `az bot create` when creating an Azure Function or Web App bot

    - The value of `"publishCommand"` is an `az bot publish` command prepopulated with the required parameters to publish the newly created bot
- Updated `"WEBSITE_NODE_DEFAULT_VERSION"` in ARM template for v4 SDK bots to use 10.14.1 instead of 8.9.4

Key Vault

- Fixed issue with `keyvault secret backup` where some users received an `unexpected_keyword` error when using `--id`

Monitor

- Changed `monitor metrics alert [create|update]` to allow dimension value `*`

Network

- Changed `dns zone export` to ensure exported CNAMEs are FQDNs
- Added `--gateway-name` parameter to `nic ip-config address-pool [add|remove]` to support application gateway backend address pools
- Added `--traffic-analytics` and `--workspace` arguments to `network watcher flow-log configure` to support traffic analytics through a Log Analytics workspace
- Added `--idle-timeout` and `--floating-ip` to `lb inbound-nat-pool [create|update]`

Policy Insights

- Added `policy remediation` commands to support resource policy remediation features

RDBMS

- Improved help message and command parameters

Redis

- Added commands for managing firewall-rules (create, update, delete, show, list)
- Added commands for managing server-link (create, delete, show, list)
- Added commands for managing patch-schedule (create, update, delete, show)
- Added support for Availability Zones and Minimum TLS Version to `redis create
- [BREAKING CHANGE] Removed `redis update-settings` and `redis list-all` commands
- [BREAKING CHANGE] Parameter for `redis create`: 'tenant settings' is not accepted in key[=value] format
- [DEPRECATED] Added warning message for deprecating `redis import-method` command

Role

- [BREAKING CHANGE] Moved `az identity` command here from `vm` commands

SQL VM

- [DEPRECATED] Deprecated `--boostrap-acc-pwd` argument due to typo

VM

- Changed `vm list-skus` to allow use of `--all` in place of `--all true`
- Added `vmss run-command [invoke | list | show]`
- Fixed bug where `vmss encryption enable` would fail if run previously
- [BREAKING CHANGE] Moved `az identity` command to `role` commands

January 31, 2019

Version 2.0.57

Core

- Hot Fix for issue 8399.

January 28, 2019

Version 2.0.56

ACR

- Added support for VNet/IP rules

ACS

- Added Virtual Nodes Preview
- Added Managed OpenShift commands
- Added support for service principal updates operation with `aks update-credentials -reset-service-principal`

AMS

- [BREAKING CHANGE] Renamed `ams asset get-streaming-locators` to `ams asset list-streaming-locators`
- [BREAKING CHANGE] Renamed `ams streaming-locator get-content-keys` to `ams streaming-locator list-content-keys`

Appservice

- Added support for app insights on `functionapp create`
- Added support for app service plan creation (including Elastic Premium) to Function Apps
- Fixed app setting issues with Elastic Premium plans

Container

- Added `container start` command
- Changed to allow using decimal values for CPU during container creation

EventGrid

- Added `--deadletter-endpoint` parameter to `event-subscription [create|update]`
- Added storagequeue and hybridconnection as new values for 'event-subscription [create|update] --endpoint-type`
- Added `--max-delivery-attempts` and `--event-ttl` parameters to `event-subscription create` to specify the retry policy for events
- Added a warning message to `event-subscription [create|update]` when webhook as destination is used for an event subscription
- Added source-resource-id parameter for all event subscription related commands and mark all other source resource related parameters as deprecated

HDInsight

- [BREAKING CHANGE] Removed the `--virtual-network` and `--subnet-name` parameters from `hdinsight [application] create`
- [BREAKING CHANGE] Changed `hdinsight create --storage-account` to accept name or id of storage account instead of blob endpoints
- Added `--vnet-name` and `--subnet-name` parameters to `hdinsight create`
- Added support for Enterprise Security Package and disk encryption to `hdinsight create`
- Added `hdinsight rotate-disk-encryption-key` command
- Added `hdinsight update` command

IoT

- Added encoding format to routing-endpoint command

Kusto

- Preview release

Monitor

- Changed ID comparison to be case insensitive

Profile

- Enable tenant level account for managed service identity for `login`

Network

- Fixed issue with `express-route update`: where `--bandwidth` argument was ignored
- Fixed issue with `ddos-protection update` where set comprehension caused stack trace

Resource

- Added support for URI parameters file to `group deployment create`
- Added support for managed identity to `policy assignment [create|list|show]`

SQL Virtual Machine

- Preview release

Storage

- Changed fix to update only properties that are changed on the same object
- Fixed #8021, binary data is encoded in base 64 when returned

VM

- Changed `vm encryption enable` to validate disk encryption keyvault and that key encryption keyvault exists
- Added `--force` flag to `vm encryption enable`

January 15, 2019

Version 2.0.55

ACR

- Changed to allow force push a helm chart that doesn't exist
- changed to allow runtime operations without ARM requests
- [DEPRECATED] Deprecated `--resource-group` parameter in the commands:
  - `acr login`
  - `acr repository`
  - `acr helm`

ACS

- Added support for new ACI regions

Appservice

- Fixed issue with uploading certificates for apps that are hosted on an ASE, where the ASE RG & App RG are different
- Changed `webapp up` to use SKU P1V1 as default for Linux
- Fixed `[webapp|functionapp] deployment source config-zip` to show the right error message when a deployment fails
- Added `webapp ssh` command

Botservice

- Added deployment status updates to `bot create`

Configure

- Added `none` as a configurable output format

CosmosDB

- Added support for creating database with shared throughput

HDInsight

- Added commands for managing applications
- Added commands for managing script actions
- Added commands for managing Operations Management Suite (OMS)
- Added support to list regional usage to `hdinsight list-usage`
- [BREAKING CHANGE] Removed default cluster type from `hdinsight create`

Network

- Added `--custom-headers` and `--status-code-ranges` arguments to `traffic-manager profile [create|update]`
- Added new routing types: Subnet and Multivalue
- Added `--custom-headers` and `--subnets` arguments to `traffic-manager endpoint [create|update]`
- Fixed issue where supplying `--vnets ""` to `ddos-protection update` caused an error

Role

- [DEPRECATED] Deprecated `--password` argument for `create-for-rbac`. Use secure passwords generated by the CLI instead

Security

- Initial Release

Storage

- [BREAKING CHANGE] Changed `storage [blob|file|container|share] list` default number of results to be 5,000. Use `--num-results *` for original behavior of returning all results
- Added `--marker` parameter to `storage [blob|file|container|share] list`
- Added log marker for next page to STDERR for `storage [blob|file|container|share] list`
- Added `storage blob service-properties update` command with support for static websites

VM

- Changed `vm [disk|unmanaged-disk]` and `vmss disk` to have more consistent parameters
- Added support for cross tenant image referencing to `[vm|vmss] create`
- Fixed bug with default configuration in `vm diagnostics get-default-config --windows-os`
- Added argument `--provision-after-extensions` to `vmss extension set` to define what extensions must be provisioned before the extension being set
- Added argument `--replica-count` to `sig image-version update` for setting the default replication count
- Fixed bug with `image create --source` where source os disk is mistaken for a VM with the same name, even if the full resource ID is provided

December 20, 2018

Version 2.0.54

Appservice

- Fixed issue where `webapp up` would fail to redeploy
- Added support for listing and restoring webapp snapshots
- Added support for `--runtime` flag to Windows function apps

IoTCentral

- Fixed update command API call

Role

- [BREAKING CHANGE] Changed `ad [app|sp] list` to only list the first 100 objects by default

SQL

- Added support for custom collation on managed instances

VM

- Added `---os-type` parameter to `disk create`

December 18, 2018

Version 2.0.53

ACR

- Added support for image import from external Container Registries
- Condensed the table layout for task list
- Added support for Azure DevOps URLs

ACS

- Added Virtual Nodes Preview
- Removed "(PREVIEW)" from AAD arguments to `aks create`
- [DEPRECATED] Deprecated `az acs` commands. The ACS service will retire on January 31, 2020
- Added support of Network Policy when creating new AKS clusters
- Removed requirement of `--nodepool-name` argument for `aks scale` if there's only one nodepool

Appservice

- Fixed issue where `webapp config container` did not honor `--slot` parameter

Botservice

- Added support for `.bot` file parsing when calling `bot show`
- Fixed AppInsights provisioning bug
- Fixed whitespace bug when dealing with file paths
- Reduced Kudu network calls
- General command UX improvements

Consumption

- Fixed bugs for budget API to show notifications

CosmosDB

- Added support for updating account from multi-master to single-master

Maps

- Added support for the S1 SKU to `maps account [create|update]`

Network

- Added support for `--format` and `--log-version` to `watcher flow-log configure`
- Fixed issue with `dns zone update` where using "" to clear resolution and registration VNets didn't work

Resource

- Fixed handling of scope parameter for management groups in `policy assignment [create|list|delete|show|update]`

- Added new command `resource wait`

Storage

- Added ability to update log schema version for storage services in `storage logging update`

VM

- Fixed crash in `vm identity remove` when the specified vm has no assigned managed service identities

December 4, 2018

Version 2.0.52

Core

- Added support for cross tenant resource provisioning for multi-tenant service principal
- Fixed bug where ids piped from a command with tsv output was improperly parsed

Appservice

- [PREVIEW] Added `webapp up` command that helps in creating & deploying contents to app
- Fixed a bug on container based windows app due to backend change

Network

- Added `--exclusion` argument to `application-gateway waf-config set` to support WAF exclusions

Role

- Added support for custom identifiers for password credential

VM

- [DEPRECATED] Deprecated `vm extension [show|wait] --expand` parameter
- Added `--force` parameter to `vm restart` to redeploy unresponsive VMs
- Changed `[vm|vmss] create --authentication-type` to accept "all" to create a VM with both password and ssh authentication
- Added `image create --os-disk-caching` parameter to set os disk caching for an image

November 20, 2018

Version 2.0.51

Core

- Changed MSI login to not reuse subscription name in identity

ACR

- Added context token to task step
- Added support for setting secrets in acr run to mirror acr task
- Improved support for `--top` and `--orderby` for `show-tags` and `show-manifests` commands

Appservice

- Changed zip deployment default timeout to poll for the status increased to 5 mins, also adding a timeout property to customize this value
- Updated the default `node_version`. Resetting slot swap action, during a two phase swap preserves all the appsettings & connection strings
- Removed client-side SKU check for Linux app service plan create
- Fixed error when trying to get zipdeploy status

IotCentral

- Added subdomain availability check when creating an IoT Central application

KeyVault

- Fixed bug where errors may have been ignored

Network

- Added `root-cert` subcommands to `application-gateway` to handle trusted root certifcates
- Added `--min-capacity` and `--custom-error-pages` options to `application-gateway [create|update]`:
- Added `--zones` for availability zone support to `application-gateway create`
- Added arguments `--file-upload-limit`, `--max-request-body-size` and `--request-body-check` to `application-gateway waf-config set`

Rdbms

- Added mariadb vnet commands

Rbac

- Fixed an issue with attempting to update immutable credentials in `ad app update`
- Added output warnings to communicate breaking changes in the near future for `ad [app|sp] list`

Storage

- Improved handling of corner cases for storage copy commands
- Fixed issue with `storage blob copy start-batch` not using login credentials when the destination and source accounts are the same
- Fixed bug with `storage [blob|file] url` where `sas_token` wasn't incorporated into URL
- Added breaking change warning to `[blob|container] list`: will soon output only first 5000 results by default

VM

- Added support to `[vm|vmss] create --storage-sku` to specify the storage account SKU for managed OS and data disks separately
- Changed version name parameters to `sig image-version` to be `--image-version -e`
- Deprecated `sig image-version` argument `--image-version-name`, replaced by `--image-version`
- Added support to use local OS disk to `[vm|vmss] create --ephemeral-os-disk`
- Added support for `--no-wait` to `snapshot create/update`
- Added `snapshot wait` command
- Added support for using instance name with `[vm|vmss] extension set --extension-instance-name`

November 6, 2018

Version 2.0.50

Core

- Added support for service principal sn+issuer auth

ACR

- Added support for commit and pull request git events for Task source trigger
- Changed to use default Dockerfile if it's not specified in build command

ACS

- [BREAKING CHANGE] Removed `enable_cloud_console_aks_browse` to enable 'az aks browse' by default

Advisor

- GA release

AMS

- Added new command groups:
  - `ams account-filter`
  - `ams asset-filter`
  - `ams content-key-policy`
  - `ams live-event`
  - `ams live-output`
  - `ams streaming-endpoint`
  - `ams mru`
- Added new commands:
  - `ams account check-name`
  - `ams job update`
  - `ams asset get-encryption-key`
  - `ams asset get-streaming-locators`
  - `ams streaming-locator get-content-keys`
- Added encryption parameters support to `ams streaming-policy create`
- Added support to `ams transform output remove` now can be performed by passing the output index to remove
- Added `--correlation-data` and `--label` arguments to `ams job` command group
- Added `--storage-account` and `--container` arguments to `ams asset` command group
- Added default values for expiry time (Now+23h) and permissions (Read) in `ams asset get-sas-url` command
- [BREAKING CHANGE] Replaced `ams streaming locator` command with `ams streaming-locator`
- [BREAKING CHANGE] Updated `--content-keys` argument of `ams streaming locator`
- [BREAKING CHANGE] Renamed `--content-policy-name` to `--content-key-policy-name` in `ams streaming locator` command
- [BREAKING CHANGE] Replaced `ams streaming policy` command with `ams streaming-policy`
- [BREAKING CHANGE] Replaced `--preset-names` argument with `--preset` in `ams transform` command group. Now you can only set 1 output/preset at a time (to add more you have to run `ams transform output add`). Also, you can set custom StandardEncoderPreset by passing the path to your custom JSON
- [BREAKING CHANGE] Renamed `--output-asset-names` to `--output-assets` in `ams job start` command. Now it accepts a space-separated list of assets in 'assetName=label' format. An asset without label can be sent like this: 'assetName='

AppService

- Fixed a bug in `az webapp config backup update` that prevents setting a backup schedule if one is not already set

Configure

- Added YAML to output format options

Container

- Changed to show identity when exporting a container group to yaml

EventHub

- Added `--enable-kafka` flag to support Kafka in `eventhub namespace [create|update]`

Interactive

- Interactive now installs the `interactive` extension, which will allow for faster updates and support

Monitor

- Added support for metric names which include characters forward-slash (/) and period (.) to `--condition` in `monitor metrics alert [create|update]`

Network

- Deprecated `network interface-endpoint` command names in favor of `network private-endpoint`
- Fixed issue with where `--peer-circuit` argument in `express-route peering connection create` would not accept an ID
- Fixed issue where `--ip-tags` did not work correctly with `public-ip create`

Profile

- Added `--use-cert-sn-issuer` to `az login` for service principal login with cert auto-rolls

RDBMS

- Added mysql replica commands

Resource

- Added support for management groups and subscriptions to `policy definition|set-definition` commands

Role

- Added support for API permission management, signed-in-user, and application password & certificate credential management
- Changed `ad sp create-for-rbac` to clarify the confusion between displayName and service principal name
- Added support to grant permissions to AAD apps

Storage

- Added support to connect to storage services only with SAS and endpoints (without an account name or a key) as described in
  `Configure Azure Storage connection strings <https://docs.microsoft.com/azure/storage/common/storage-configure-connection-string>`

VM

- Added `storage-sku` argument to `image create` for setting the image's default storage account type
- Fixed bug with `vm resize` where `--no-wait` option causes command to crash
- Changed `vm encryption show` table output format to show status
- Changed `vm secret format` to require json/jsonc output. Warns user and defaults to json output if an undesired output format is selected
- Improved argument validation for `vm create --image`

October 23, 2018

Version 2.0.49

Core

- Fixed issue with `--ids` where `--subscription` would take precedence over the subscription in `--ids`
- Added explicit warnings when parameters would be ignored by use of `--ids`

ACR

- Fixed an ACR Build encoding issue in Python2

CDN

- [BREAKING CHANGE] Changed `cdn endpoint create`'s default query string caching behaviour to no longer defaults to "IgnoreQueryString". It is now set by the service

Container

- Added `Private` as a valid type to pass to '--ip-address'
- Changed to allow using only subnet ID to setup a virtual network for the container group
- Changed to allow using vnet name or resource id to enable using vnets from different resource groups
- Added `--assign-identity` for adding a MSI identity to a container group
- Added `--scope` to create a role assignment for the system assigned MSI identity
- Added a warning when creating a container group with an image without a long running process
- Fixed table output issues for `list` and `show` commands

CosmosDB

- Added `--enable-multiple-write-locations` support to `cosmosdb create`

Interactive

- Changed to ensure global subscription parameter appears in parameters

IoT Central

- Added template and display name options for IoT Central Application creation
- [BREAKING CHANGE] Removed support for the F1 SKU; Use S1 SKU instead

Monitor

- Changes to `monitor activity-log list`:
  - Added support for listing all events at the subscription level
  - Added `--offset` parameter to more easily create time queries
  - Improved validation for `--start-time` and `--end-time` to use wider range of ISO8601 formats and more user-friendly datetime formats
  - Added `--namespace` as alias for deprecated option `--resource-provider`
  - Deprecated `--filters` because no values other than those with strongly-typed options are supported by the service
- Changes to `monitor metrics list`:
  - Added `--offset` parameter to more easily create time queries
  - Improved validation for `--start-time` and `--end-time` to use wider range of ISO8601 formats and more user-friendly datetime formats
- Improved validation for `--event-hub` and `--event-hub-rule` arguments to `monitor diagnostic-settings create`

Network

- Added `--app-gateway-address-pools` and `--gateway-name` arguments to `nic create`, to support adding application gateway backend address pools to a NIC
- Added `--app-gateway-address-pools` and `--gateway-name` arguments to `nic ip-config create/update`, to support adding application gateway backend address pools to a NIC

ServiceBus

- Added Read-Only `migration_state` to MigrationConfigProperties to show current Service Bus Standard to Premium namespace migration state

SQL

- Fixed `sql failover-group create` and `sql failover-group update` to work with Manual failover policy

Storage

- Fixed `az storage cors list` output formatting, all items show correct "Service" key
- Added `--bypass-immutability-policy` parameter for immutability-policy blocked container deletion

VM

- Enforce disk caching mode be `None` for Lv/Lv2 series of machines in `[vm|vmss] create`
- Updated supported size list supporting networking accelerator for `vm create`
- Added strong typed arguments for ultrassd iops and mbps configs for `disk create`

October 16, 2018

Version 2.0.48

VM

- Fixed SDK issue that caused Homebrew instllation to fail

October 9, 2018

Version 2.0.47

Core

- Improved error handling for "Bad Request" errors

ACR

- Added support for similar table format as helm client

ACS

- Added `aks [create|scale] --nodepool-name` to configure nodepool name, truncated to 12 characters, default - nodepool1
- Fixed to fall back to 'scp' when Parimiko fails
- Changed `aks create` to no longer require `--aad-tenant-id`
- Improved merging of Kubernetes credentials when duplicate entries are present

Container

- Changed `functionapp create` to support creating a Linux consumption plan type with a specific runtime
- [PREVIEW] Added support for hosting webapps on Windows containers

Event Hub

- Fixed `eventhub update` command
- [BREAKING CHANGE] Changed `list` commands to handle errors for resource(s) NotFound(404) in the typical way instead of showing empty list

Extensions

- Fixed issue with attempting to add an extension that is already installed

HDInsight

- Initial release

IoT

- Added extension installation comand to first-run banner

KeyVault

- Changed to restrict keyvault storage commmands to the latest API profile

Network

- Fixed `network dns zone create`: Command succeeds even if the user has configured a default location. See #6052

- Deprecated `--remote-vnet-id` for `network vnet peering create`
- Added `--remote-vnet` to `network vnet peering create` which accepts a name or ID
- Added support for multiple subnet prefixes to `network vnet create` with `--subnet-prefixes`
- Added support for multiple subnet prefixes to `network vnet subnet [create|update]` with `--address-prefixes`
- Fixed issue with `network application-gateway create` that prevented creating gateways with `WAF_v2` or `Standard_v2` SKU
- Added `--service-endpoint-policy` convenience argument to `network vnet subnet update`

Role

- Added support for listing Azure AD app owners to `ad app owner`
- Added support for listing Azure AD service principal owners to `ad sp owner`
- Changed to ensure role definition create & update commands accept multiple permission configurations
- Changed `ad sp create-for-rbac` to ensure home page URI is always "https"

Service Bus

- [BREAKING CHANGE] Changed `list` commands to handle errors for resource(s) NotFound(404) in the typical way instead of showing empty list

VM

- Fixed empty `accessSas` field in `disk grant-access`
- Changed `vmss create` to reserve large enough frontend port range to handle overprovisioning
- Fixed update commands for `sig`
- Added `--no-wait` support for managing image versions in `sig`
- Changed `vm list-ip-addresses` to show availability zone of public IP addresses
- Changed `[vm|vmss] disk attach` to set disk's default lun to the first available spot

September 21, 2018

Version 2.0.46

ACR

- Added ACR Task commands
- Added quick run command
- Deprecated `build-task` command group
- Added `helm` command group to support managing helm charts with ACR
- Added support for idempotent create for managed registry
- Added a no-format flag for displaying build logs

ACS

- Changed the `install-connector` command to set the AKS Master FQDN
- Fixed creating role assignment for vnet-subnet-id when not specifying service principal and skip-role-assignemnt

AppService

- Added support for webjobs (continuous and triggered) operations management
- az webapp config set supports --fts-state propertyAlso added support fot az functionapp config set & show
- Added support for bring your own storage for webapps
- Added support for listing and restoring deleted webapps

Batch

- Changed adding tasks through `--json-file` to support AddTaskCollectionParameter syntax
- Updated documentation of accepted `--json-file` formats
- Added `--max-tasks-per-node-option` to `batch pool create`
- Changed behavior of `batch account` to show currently logged in account if no options are specified

Batch AI

- Fixed auto storage account creation failure in `batchai cluster create` command

Cognitive Services

- Added completer for `--sku`, `--kind`, `--location` arguments
- Added command `cognitiveservices account list-usage`
- Added command `cognitiveservices account list-kinds`
- Added command `cognitiveservices account list`
- Deprecated `cognitiveservices list`
- Changed `--name` to be optional for `cognitiveservices account list-skus`

Container

- Added ability to restart and stop a running container group
- Added `--network-profile` for passing in a network profile
- Added `--subnet`, `--vnet_name`, to allow creating container groups in a VNET
- Changed table output to show the status of the container group

Datalake

- Added commands for virtual network rules

Interactive Shell

- Fixed error on Windows where commands fail to run properly
- Fixed command loading problem in interactive that was caused by deprecated objects

IoT

- Added support for routing IoT Hubs

Key Vault

- Fixed Key Vault key import for RSA keys

Network

- Add `network public-ip prefix` commands to support public IP prefixes features
- Add `network service-endpoint` commands to support service endpoint policy features
- Add `network lb outbound-rule` commands to support creation of Standard Load Balancer outbound rules
- Add `--public-ip-prefix` to `network lb frontend-ip create/update` to support frontend IP configurations using public IP prefixes
- Add `--enable-tcp-reset` to `network lb rule/inbound-nat-rule/inbound-nat-pool create/update`
- Add `--disable-outbound-snat` to `network lb rule create/update`
- Allow `network watcher flow-log show/configure` to be used with classic NSGs
- Add `network watcher run-configuration-diagnostic` command
- Fix `network watcher test-connectivity` command and add `--method`, `--valid-status-codes` and `--headers` properties
- `network express-route create/update`: Add `--allow-global-reach` flag
- `network vnet subnet create/update`: Add support for `--delegation`
- Added `network vnet subnet list-available-delegations` command

- `network traffic-manager profile create/update`: Added support for `--interval`, `--timeout` and `--max-failures` for Monitor configuration Deprecated options `--monitor-path`, `--monitor-port` and `--monitor-protocol` in favor of `--path`, `--port`, `--protocol`
- `network lb frontend-ip create/update`: Fixed the logic for setting private IP allocation methodIf a private IP address is provided, the allocation will be staticIf no private IP address is provided, or empty string is provided for private IP address, allocation is dynamic.
- `dns record-set * create/update`: Add support for `--target-resource`
- Add `network interface-endpoint` commands to query interface endpoint objects
- Add `network profile show/list/delete` for partial management of network profiles
- Add `network express-route peering connection` commands to manage peering connections between ExpressRoutes

RDBMS

- Added support for MariaDB service

Reservation

- Added CosmosDb in the reserved resource enum type
- Added name property in Patch model

Manage App

- Fixed bug in `managedapp create --kind MarketPlace` causing instance creation of a Marketplace managed app to crash
- Changed `feature` commands to be restricted to supported profiles

Role

- Added support for listing user's group memberships

SignalR

- First release

Storage

- Added `--auth-mode login` parameter for use of user's login credentials for blob and queue authorization
- Added `storage container immutability-policy/legal-hold` to manage immutable storage

VM

- Fixed issue where `vm create --generate-ssh-keys` overwrites private key file if public key file is missing (#4725, #6780)
- Added support for shared image gallery through `az sig`

August 28, 2018

Version 2.0.45

Core

- Fixed issue of loading empty configuration file
- Added support for profile `2018-03-01-hybrid` for Azure Stack

ACR

- Added a workaround for runtime operations without ARM requests
- Changed to exclude version control files (eg, .git, .gitignore) from uploaded tar by default in `build` command

ACS

- Changed `aks create` to defaults to `Standard_DS2_v2` VMs
- Changed `aks get-credentials` to now call new apis to get cluster credential

AppService

- Added support for CORS on functionapp & webapp
- Added ARM tag support on create commands
- Changed `[webapp|functionapp] identity show` to exit with code 3 upon a missing resource

Backup

- Changed `backup vault backup-properties show` to exit with code 3 upon a missing resource

Bot Service

- Initial Bot Service CLI Release

Cognitive Services

- Added new parameter `--api-properties,` which is required for creating some of the services

IoT

- Fixed issue with associating linked hubs

Monitor

- Added `monitor metrics alert` commands for near-realtime metric alerts
- Deprecated `monitor alert` commands

Network

- Changed `network application-gateway ssl-policy predefined show` to exit with code 3 upon a missing resource

Resource

- Changed `provider operation show` to exit with code 3 upon a missing resource

Storage

- Changed `storage share policy show` to exit with code 3 upon a missing resource

VM

- Changed `vm/vmss identity show` to exit with code 3 upon a missing resource
- Deprecated `--storage-caching` for `vm create`

Auguest 14, 2018

Version 2.0.44

Core

- Fixed numeric display in `table` output
- Added YAML output format

Telemetry

- Improved telemetry reporting

ACR

- Added `content-trust policy` commands
- Fixed issue where `.dockerignore` was not handled properly

ACS

- Changed `az acs/aks install-cli` to install under `%USERPROFILE%\.azure-kubectl` on Windows

- Changed `az aks install-connector` to detect if the cluster has RBAC and configure ACI Connector appropriately
- Changed to role assignment to the subnet when it's provided
- Added new option to "skip role assignment" for subnet when it's provided
- Changed to skip role assignment to subnet when assignment already exists

AppService

- Fixed a bug that prevent from creating a function-app using storage accounts in external resource groups
- Fixed a crash on zip deployment

BatchAI

- Changed logger output for auto-storage account creation to specifies "resource *group*".

Container

- Added `--secure-environment-variables` for passing secure environment variables to a container

IoT

- [BREAKING CHANGE] Removed deprecated commands which have moved to the iot extension
- Updated elements to not assume `azure-devices.net` domain

Iot Central

- Initial release of IoT Central module

KeyVault

- Added commands for managing storage accounts and sas-definitions
- Added commands for network-rules
- Added `--id` parameter to secret, key, and certificate operations
- Added support for KV mgmt multi-api version
- Added support for KV data plane multi-api version

Relay

- Initial release

Sql

- Added `sql failover-group` commands

Storage

- [BREAKING CHANGE] Changed `storage account show-usage` to require `--location` parameter and will list by region
- Changed `--resource-group` parameter to be optional for `storage account` commands
- Removed 'Failed precondition' warnings for individual failures in batch commands for single aggregated message
- Changed `[blob|file] delete-batch` commands to no longer output array of nulls
- Changed `blob [download|upload|delete-batch]` commands to read sas-token from container url

VM

- Added common filters to `vm list-skus` for ease of use

July 31, 2018

Version 2.0.43

ACR

- Added `--with-secure-properties` flag to `acr build-task show` command
- Added `acr build-task update-build` command

ACS

- Changed to return return 0 (success) when ending `az aks browse` by pressing [Ctrl+C]

Batch

- Fixed bug when showing AAD token in cloudshell

Container

- Removed requirement for `--log-analytics-workspace-key` for name or ID when in set subscription

Network

- Added dns support to 2017-03-09-profile for Azure Stack

Resource

- Added `--rollback-on-error` to `group deployment create` to execute a known-good deployment on error
- Fixed issue where `--parameters {}` with `group deployment create` resulted in an error

Role

- Added support for stack profile 2017-03-09-profile
- Fixed issue where generic update parameters to `app update` would not work correctly

Search

- Added commands for Azure Search service

Service Bus

- Added migration command group to migrate a namespace from Service Bus Standard to Premium
- Added new optional properties to Service Bus queue and Subscription
  - `--enable-batched-operations` and `--enable-dead-lettering-on-message-expiration` in `queue`
  - `--dead-letter-on-filter-exceptions` in `subscriptions`

Storage

- Added support for download of large files using a single connection
- Converted `show` commands that were missed from failing with exit code 3 upon a missing resource

VM

- Added support to list availability sets by subscription
- Added support for `StandardSSD_LRS`
- Added support for application security group on creating VM scale set
- [BREAKING CHANGE] Changed `[vm|vmss] create`, `[vm|vmss] identity assign`, and `[vm|vmss] identity remove` to output user assigned identities in dictionary format

July 18, 2018

Version 2.0.42

Core

- Added support for browser-based login in WSL bash window
- Added `--force-string` flag to all generic update commands
- [BREAKING CHANGE] Changed 'show' commands to log error message and fail with exit code of 3 upon a missing resource

ACR

- [BREAKING CHANGE] Updated '--no-push' to a pure flag in 'acr build' command
- Added `show` and `update` commands under `acr repository` group
- Added `--detail` flag for `show-manifests` and `show-tags` to show more detailed information
- Added `--image` parameter to support get build details or logs by an image

ACS

- Changed `az aks create` to error out if `--max-pods` is less than 5

AppService

- Added support for PremiumV2 skus

Batch

- Fixed bug on using token credential on cloud shell mode
- Changed JSON input to be case-insensitive

Batch AI

- Fixed `az batchai job exec` command

Container

- Removed the requirement for username and password for non dockerhub registries
- Fixed error when creating container groups from yaml file

Network

- Added `--no-wait` support to `network nic [create|update|delete]`
- Added `network nic wait`
- Deprecated `--ids` argument for `network vnet [subnet|peering] list`
- Added `--include-default` flag to include default security rules in the output of `network nsg rule list`

Resource

- Added `--no-wait` support to `group deployment delete`
- Added `--no-wait` support to `deployment delete`
- Added `deployment wait` command
- Fixed issue where the subscription-level `az deployment` commands erroneously appeared for profile 2017-03-09-profile

SQL

- Fixed 'The provided resource group name ... did not match the name in the Url' error when specifying elastic pool name for `sql db copy` and `sql db replica create` commands
- Allow configuring default sql server by executing `az configure --defaults sql-server=<name>`
- Implemented table formatters for `sql server`, `sql server firewall-rule`, `sql list-usages`, and `sql show-usage` commands

Storage

- Added `pageRanges` property to `storage blob show` output that will be populated for page blobs

VM

- [BREAKING CHANGE] Changed `vmss create` to use `Standard_DS1_v2` as the default instance size
- Added `--no-wait` support to `vm extension [set|delete]` and `vmss extension [set|delete]`
- Added `vm extension wait`

July 3, 2018

Version 2.0.41

AKS

- Changed monitoring to use subscription ID

July 3, 2018

Version 2.0.40

Core

- Added a new authorization code flow for interactive login

ACR

- Added polling build status
- Added support for case-insensitive enum values
- Added `--top` and `--orderby` parameters for `show-manifests`

ACS

- [BREAKING CHANGE] Enable Kubernetes role-based access control by default
- Added `--disable-rbac` argument and deprecated `--enable-rbac` since it's the default now
- Updated options for `aks browse` command. Added `--listen-port` support
- Updated the default helm chart package for `aks install-connector` command. Use virtual-kubelet-for-aks-latest.tgz
- Added `aks enable-addons` and `aks disable-addons` commands to update an existing cluster

AppService

- Added support for disabling identity via `webapp identity remove`
- Removed `preview` tag for Identity feature

Backup

- Updated module definition

BatchAI

- Fixed table output for `batchai cluster node list` and `batchai job node list` commands

Cloud

- Added `acr login` server suffix to cloud config

Container

- Changed `container create` to default to long running operation
- Added Log Analytics parameters `--log-analytics-workspace` and `--log-analytics-workspace-key`
- Added `--protocol` parameter to specify which network protocol to use

Extension

- Changed `extension list-available` to only show extensions compatible with CLI version

Network

- Fixed issue where record types were case-sensitive (#6602)

Rdbms

- Added `[postgres|myql] server vnet-rule` commands

Resource

- Added new operation group `deployment`

VM

- Added support for removing system assigned identity

June 25, 2018

Version 2.0.39

CLI

- Updated file trimming in MSI installer to fix extension installation issue

June 19, 2018

Version 2.0.38

Core

- Added global support for `--subscription` to most commands

ACR

- Added `azure-storage-blob` as dependency
- Changed default CPU configuration with `acr build-task create` to use 2 cores

ACS

- Updated options of `aks use-dev-spaces` command. Added `--update` support
- Changed `aks get-credentials --admin` to not eplace the user context in `$HOME/.kube/config`
- Exposed read-only `nodeResourceGroup` property on managed clusters
- Fixed `acs browse` command error
- Made `--connector-name` optional for `aks install-connector`, `aks upgrade-connector` and `aks remove-connector`
- Added new Azure Container Instance regions for `aks install-connector`
- Added the normalized location into the helm release name and node name to `aks install-connector`

AppService

- Added support for newer versions of urllib
- Added support to `functionapp create` to use appservice plan from external resource groups

Batch

- Removed `azure-batch-extensions` dependency

Batch AI

- Added support for workspaces. Workspaces allow to group clusters, file-servers and experiments in groups removing limitation on number of resources can be created
- Added support for experiments. Experiments allow to group jobs in collections removing limitation on number of created jobs
- Added support to configure `/dev/shm` for jobs running in a docker container
- Added `batchai cluster node exec` and `batchai job node exec` commands. These commands allow to execute any commands directly on nodes and provide functionality for port forwarding.
- Added support for `--ids` to `batchai` commands
- [BREAKING CHANGE] All clusters and fileservers must be created under workspaces
- [BREAKING CHANGE] Jobs must be created under experiments

- [BREAKING CHANGE] Removed `--nfs-resource-group` from `cluster create` and `job create` commands. To mount an NFS belonging to a different workspace/resource group provide file server's ARM ID via `--nfs` option
- [BREAKING CHANGE] Removed `--cluster-resource-group` from `job create` command. To submit a job on a cluster belonging to a different workspace/resource group provide cluster's ARM ID via `--cluster` option
- [BREAKING CHANGE] Removed `location` attribute from jobs, cluster and file servers. Location now is an attribute of a workspace.
- [BREAKING CHANGE] Removed `--location` from `job create`, `cluster create` and `file-server create` commands
- [BREAKING CHANGE] Changed names of short options to make interface more consistent:
  - Renamed [`--config` , `-c`] to [`--config-file` , `-f`]
  - Renamed [`--cluster` , `-r`] to [`--cluster` , `-c`]
  - Renamed [`--cluster` , `-n`] to [`--cluster` , `-c`]
  - Renamed [`--job` , `-n`] to [`--job` , `-j`]

Maps

- [BREAKING CHANGE] Changed `maps account create` to require accepting Terms of Service either by interactive prompt or `--accept-tos` flag

Network

- Added support for `https` to `network lb probe create` #6571
- Fixed issue where `--endpoint-status` was case sensitive. #6502

Reservations

- [BREAKING CHANGE] Added required parameter `ReservedResourceType` to `reservations catalog show`
- Added parameter `Location` to `reservations catalog show`
- [BREAKING CHANGE] Removed `kind` from `ReservationProperties`
- [BREAKING CHANGE] Renamed `capabilities` to `sku_properties` in `Catalog`
- [BREAKING CHANGE] Removed `size` and `tier` properties from `Catalog`
- Added parameter `InstanceFlexibility` to `reservations reservation update`

Role

- Improved error handling

SQL

- Fixed confusing error when running `az sql db list-editions` for a location that is not available to your subscription

Storage

- Changed table output for `storage blob download` to be more readable

VM

- Improved refine vm size check for accelerated networking support in `vm create`
- Added warning for `vmss create` that the default vm size will be switched from `Standard_D1_v2` to `Standard_DS1_v2`
- Added `--force-update` to `[vm|vmss] extension set` to update the extension even when the configuration has not changed

June 13, 2018

Version 2.0.37

Core

- Improved interactive telemetry

June 13, 2018

Version 2.0.36

AKS

- Added advanced networking options to `aks create`
- Added arguments to `aks create` to enable monitoring and HTTP routing
- Added `--no-ssh-key` argument to `aks create`
- Added `--enable-rbac` argument to `aks create`
- [PREVIEW] Added support for Azure Active Directory authentication to `aks create`

AppService

- Fixed an issue with incompatible urllib versions

June 5, 2018

Version 2.0.35

Interactive

- Added limits to the dependencies of interactive mode

June 5, 2018

Version 2.0.34

Core

- Added support for cross tenant resource referencing
- Improved telemetry upload reliability

ACR

- Added support for VSTS as a remote source location
- Added `acr import` command

AKS

- Changed `aks get-credentials` to create the kube config file with more secure filesystem permissions

Batch

- Fixed bug in Pool list table formatting [Issue #4378]

IOT

- Added support for creating Basic Tier IoT Hubs

Network

- Improved `network vnet peering`

Policy Insights

- Initial Release

ARM

- Added `account management-group` commands.

SQL

- Added new managed instance commands:
  - `sql mi create`
  - `sql mi show`
  - `sql mi list`
  - `sql mi update`
  - `sql mi delete`
- Added new managed database commands:
  - `sql midb create`
  - `sql midb show`
  - `sql midb list`
  - `sql midb restore`
  - `sql midb delete`

Storage

- Added extra mimetypes for json and javascript to be inferred from file extensions

VM

- Changed `vm list-skus` to use fixed columns and add warning that `Tier` and `Size` will be removed
- Added `--accelerated-networking` option to `vm create`
- Added `--tags` to `identity create`

May 22, 2018

Version 2.0.33

Core

- Added support for expanding `@` in file names

ACS

- Added new Dev-Spaces commands `aks use-dev-spaces` and `aks remove-dev-spaces`
- Fixed typo in help message

AppService

- Improved generic update commands
- Added async support for `webapp deployment source config-zip`

Container

- Added support for exporting a container group in yaml format
- Added support for using a yaml file to create / update a container group

Extension

- Improved removal of extensions

Interactive

- Changed logging to mute parser for completions
- Improved handling of bad help caches

KeyVault

- Fixed keyvault commands to work in cloud shell or VMs with identity

Network

- Fix issue where `network watcher show-topology` would not work with vnet and/or subnet name #6326
- Fix issue where some `network watcher` commands would claim Network Watcher is not enabled for regions when it actually is #6264

SQL

- [BREAKING CHANGE] Changed response objects returned from `db` and `dw` commands:
  - Renamed `serviceLevelObjective` property to `currentServiceObjectiveName`
  - Removed `currentServiceObjectiveId` and `requestedServiceObjectiveId` properties
  - Changed `maxSizeBytes` property to be an integer value instead of a string
- [BREAKING CHANGE] Changed the following `db` and `dw` properties to be read-only:
  - `requestedServiceObjectiveName`. To update, use the `--service-objective` parameter or set the `sku.name` property
  - `edition`. To update, use the `--edition` parameter or set the `sku.tier` property
  - `elasticPoolName`. To update, use the `--elastic-pool` parameter or set the `elasticPoolId` property
- [BREAKING CHANGE] Changed the following `elastic-pool` properties to be read-only:
  - `edition`. To update, use the `--edition` parameter
  - `dtu`. To update, use the `--capacity` parameter
  - `databaseDtuMin`. To update, use the `--db-min-capacity` parameter
  - `databaseDtuMax`. To update, use the `--db-max-capacity` parameter
- Added `--family` and `--capacity` parameters to `db`, `dw`, and `elastic-pool` commands.
- Added table formatters to `db`, `dw`, and `elastic-pool` commands.

Storage

- Added completer for `--account-name` argument
- Fixed problem with `storage entity query`

VM

- [BREAKING CHANGE] Removed `--write-accelerator` from `vm create`. The same support can be accessed through `vm update` or `vm disk attach`
- Fixed extension image matching in `[vm|vmss] extension`
- Added `--boot-diagnostics-storage` to `vm create` to capture boot log
- Added `--license-type` to `[vm|vmss] update`

May 7, 2018

Version 2.0.32

Core

- Fixed an unhandled exception when retrieving secrets from a service principal account with cert
- Added limited support for positional arguments
- Fix issue where `--query` could not be used with `--ids`. #5591
- Improved piping scenarios from commands when using `--ids`. Supports `-o tsv` with a query specified or `-o json` without specifying a query
- Added command suggestions on error if users have typo in their commands
- Improved error when users type `az ''`
- Added support custom resource types for command modules and extensions

ACR

- Added ACR Build commands

- Improved resource not found error messages
- Improved resource creation performance and error handling
- Improved acr login in non-standard consoles and WSL
- Improved repository commands error messages
- Updated table columns and ordering

ACS

- Added warning that `az aks` is a preview service
- Fixed the permission issue in `aks install-connector` when `--aci-resource-group` is not specified

AMS

- Initial release - Manage Azure Media Services resources

Appservice

- Fixed a bug in `webapp delete` when `--slot` is provided
- Removed `--runtime-version` from `webapp auth update`
- Added support for min_tls_version & https2.0
- Added support for multicontainers

Batch AI

- Changed `batchai create cluster` to respect vm priority configured in the cluster's configuration file

Cognitive Services

- Fixed typo in example for `cognitiveservices account create` #5603

Consumption

- Added new commands for budget API

Container

- Removed requirement for `--registry-server` for `container create` when a registry server is included in the image name

Cosmos DB

- Introducing VNET support for Azure CLI - Cosmos DB

DMS

- Initial release - Adds support for the SQL to Azure SQL migration scenario

Extension

- Fixed bug where extension metadata stopped being shown

Interactive

- Allow interactive completers to function with positional arguments
- More user-friendly output when users type ''
- Fixed completions for parameters with no help
- Fixed descriptions for command-groups

Lab

- Fixed regressions from knack conversion

Network

- [BREAKING CHANGE] Removed the `--ids` parameter for:
  - `express-route auth list`

- `express-route peering list`
- `nic ip-config list`
- `nsg rule list`
- `route-filter rule list`
- `route-table route list`
- `traffic-manager endpoint list`

Profile

- Fixed `disk create` source detection
- [BREAKING CHANGE] Removed `--msi-port` and `--identity-port` as they are no longer used
- Fixed typo in `account get-access-token` short summary

Redis

- Deprecated `redis patch-schedule patch-schedule show` in favor of `redis patch-schedule show`
- Deprecated `redis list-all`. This functionality has been folded into `redis list`
- Deprecated `redis import-method` in favor of `redis import`
- Added support for `--ids` to various commands

Role

- [BREAKING CHANGE] Removed deprecated `ad sp reset-credentials`

Storage

- Allow destination sas-token to apply to source for blob copy if source sas and account key are unspecified
- Exposed --socket-timeout for blob uploads and downloads
- Treat blob names that start with path separators as relative paths
- Allow `storage blob copy --source-sas` with starting query char, '?'
- Fixed `storage entity query --marker` to accept list of key=values

VM

- Fixed an invalid detection logic on unmanaged blob uri
- Added support disk encryption w/o user provided service principals
- [BREAKING CHANGE] Do not use VM 'ManagedIdentityExtension' for MSI support
- Added support for eviction policy to `vmss`
- [BREAKING CHANGE] Removed `--ids` from:
  - `vm extension list`
  - `vm secret list`
  - `vm unmanaged-disk list`
  - `vmss nic list`
- Added write accelerator support
- Added `vmss perform-maintenance`
- Fixed `vm diagnostics set` to detect VM's OS type reliably
- Changed `vm resize` to check if the requested size is different than currently set and update only on change

April 10, 2018

Version 2.0.31

ACR

- Improved error handling of wincred fallback

ACS

- Changed aks created SPNs to be valid for 5 years

Appservice

- [BREAKING CHANGE]: Removed `assign-identity`
- Fixed uncaught exception for nonexistant webapp plans

BatchAI

- Added support for 2018-03-01 API
  - Job level mounting
  - Environment variables with secret values
  - Performance counters settings
  - Reporting of job specific path segment
  - Support for subfolders in list files api
  - Usage and limits reporting
  - Allow to specify caching type for NFS servers
  - Support for custom images
  - Added pyTorch toolkit support
- Added `job wait` command which allows to wait for the job completion and reports job exit code
- Added `usage show` command to list current Batch AI resources usage and limits for different regions
- National clouds are supported
- Added job command line arguments to mount filesystems on the job level in addition to config files
- Added more options to customize clusters - vm priority, subnet, initial nodes count for auto-scale clusters, specifying custom image
- Added command line option to specify caching type for Batch AI managed NFS
- Simplified specifying mount filesystem in config files. Now you can omit credentials for Azure File Share and Azure Blob Containers - CLI will populate missing credentials using storage account key provided via command line parameters or specified via environment variable or will query the key from Azure Storage (if the storage account belongs to the current subscription)
- Job file stream command now auto-completes when the job is completed (succeeded, failed, terminated or deleted)
- Improved `table` output for `show` operations
- Added `--use-auto-storage` option for cluster creation. This option make it simpler to manage storage accounts and mount Azure File Share and Azure Blob Containers to clusters
- Added `--generate-ssh-keys` option to `cluster create` and `file-server create`
- Added ability to provide node setup task via command line
- [BREAKING CHANGE] Moved `job stream-file` and `job list-files` commands under `job file` group
- [BREAKING CHANGE] Renamed `--admin-user-name` to `--user-name` in `file-server create` command to be consistent with `cluster create` command

Billing

- Added enrollment account commands

Consumption

- Added `marketplace` commands
- [BREAKING CHANGE] Renamed `reservations summaries` to `reservation summary`
- [BREAKING CHANGE] Renamed `reservations details` to `reservation detail`
- [BREAKING CHANGE] Removed `--reservation-order-id` and `--reservation-id` short options for `reservation` commands
- [BREAKING CHANGE] Removed `--grain` short options for `reservation summary` commands
- [BREAKING CHANGE] Removed `--include-meter-details` short options for `pricesheet` commands

Container

- Added git repo volume mount parameters `--gitrepo-url` `--gitrepo-dir` `--gitrepo-revision` and `--gitrepo-mount-path`
- Fixed #5926: `az container exec` failing when --container-name specified

Extension

- Changed distribution check message to be debug-level

Interactive

- Changed to stop completions upon unrecognized commands
- Added event hooks before and after command subtree is created
- Added completion for `--ids` parameters

Network

- Fixed #5936: `application-gateway create` tags could not bet set
- Added argument `--auth-certs` to attach authentication certificates for `application-gateway http-settings [create|update]`. #4910
- Added `ddos-protection` commands to create DDoS protection plans
- Added support for `--ddos-protection-plan` to `vnet [create|update]` to associate a VNet to a DDoS protection plan
- Fixed issue with `--disable-bgp-route-propagation` flag in `network route-table [create|update]`
- Removed dummy arguments `--public-ip-address-type` and `--subnet-type` for `network lb [create|update]`
- Added support for TXT records with RFC 1035 escape sequences to `network dns zone [import|export]` and `network dns record-set txt add-record`

Profile

- Added support for Azure Classic accounts in `account list`
- [BREAKING CHANGE] Removed `--msi` & `--msi-port` arguments

RDBMS

- Added `georestore` command
- Removed storage size restriction from `create` command

Resource

- Added support for `--metadata` to `policy definition create`
- Added support for `--metadata`, `--set`, `--add`, `--remove` to `policy definition update`

SQL

- Added `sql elastic-pool op list` and `sql elastic-pool op cancel`

Storage

- Improved error messages for malformed connection strings

VM

- Added support to configure platform fault domain count to `vmss create`
- Changed `vmss create` to default to Standard LB for zonal, large or single-placement-group disabled scale-set
- [BREAKING CHANGE]: Removed `vm assign-identity`, `vm remove-identity and` vm format-secret`
- Added support for Public-IP SKU to `vm create`
- Added `--keyvault` and `--resource-group` arguments to `vm secret format` to support scenarios where the command is unable to resolve the vault ID. #5718
- Better errors for `[vm|vmss create]` when a resource group's location has no zone support

March 27, 2018

Version 2.0.30

Core

- Show message for extensions marked as preview in help

ACS

- Fix SSL certificate verification error for `aks install-cli` in Cloud Shell

Appservice

- Added HTTPS-only support to `webapp update`
- Added support for slots to `az webapp identity [assign|show]` and `az functionapp identity [assign|show]`

Backup

- Added new command `az backup protection isenabled-for-vm`. This command can be used to check if a VM is backed up by any vault in the subscription
- Enabled Azure object IDs for `--resource-group` and `--vault-name` parameters for the following commands:
  - `backup container show`
  - `backup item set-policy`
  - `backup item show`
  - `backup job show`
  - `backup job stop`
  - `backup job wait`
  - `backup policy delete`
  - `backup policy get-default-for-vm`
  - `backup policy list-associated-items`
  - `backup policy set`
  - `backup policy show`
  - `backup protection backup-now`
  - `backup protection disable`
  - `backup protection enable-for-vm`
  - `backup recoverypoint show`
  - `backup restore files mount-rp`
  - `backup restore files unmount-rp`
  - `backup restore restore-disks`
  - `backup vault delete`
  - `backup vault show`
- Changed `--name` parameters to accept the output format from `backup ... show` commands

Container

- Added `container exec` command. Executes commands in a container for a running container group
- Allow table output for creating and updating a container group

Extension

- Added message for `extension add` if extension is in preview
- Changed `extension list-available` to show full extension data with `--show-details`
- [BREAKING CHANGE] Changed `extension list-available` to show simplified extension data by default

Interactive

- Changed completions to activate as soon as command table loading is done
- Fixed bug with using `--style` parameter
- Interactive lexer instantiated after command table dump if missing
- Improved completer support

Lab

- Fixed bugs with `create environment` command

Monitor

- Added support for `--top`, `--orderby` and `--namespace` to `metrics list` #5785
- Fixed #4529: `metrics list` Accepts a space-separated list of metrics to retrieve
- Added support for `--namespace` to `metrics list-definitions` #5785

Network

- Added support for Private DNS zones

Profile

- Added warning for `--identity-port` and `--msi-port` to `login`

RDBMS

- Added business model GA API version 2017-12-01

Resource

- [BREAKING CHANGE]: Changed `provider operation [list|show]` to not require `--api-version`

Role

- Added support for required access configurations and native clients to `az ad app create`
- Changed `rbac` commands to return less than 1000 IDs on object resolution
- Added credential management commands `ad sp credential [reset|list|delete]`
- [BREAKING CHANGE] Removed 'properties' from `az role assignment [list|show]` output
- Added support for `dataActions` and `notDataActions` permissions to `role definition`

Storage

- Fixed issue when uploading file with size between 195GB and 200GB
- Fixed #4049: Problems with append blob uploads ignoring condition parameters

VM

- Added warning to `vmss create` for upcoming breaking changes for sets with 100+ instances
- Added zone resilient support to `vm [snapshot|image]`
- Changed disk instance view to report better encryption status
- [BREAKING CHANGE] Changed `vm extension delete` to no longer return output

March 13, 2018

Version 2.0.29

ACR

- Added support for `--image` parameter to `repository delete`
- Deprecated `--manifest` and `--tag` parameters of the `repository delete` command
- Added `repository untag` command to remove a tag without deleting data

ACS

- Added `aks upgrade-connector` command to upgrade an existing connector
- Changed `kubectl` config files to use a more readable block-style YAML

Advisor

- [BREAKING CHANGE] Renamed `advisor configuration get` to `advisor configuration list`
- [BREAKING CHANGE] Renamed `advisor configuration set` to `advisor configuration update`
- [BREAKING CHANGE] Removed `advisor recommendation generate`
- Added `--refresh` parameter to `advisor recommendation list`
- Added `advisor recommendation show` command

Appservice

- Deprecated `[webapp|functionapp] assign-identity`
- Added managed identity commands `webapp identity [assign|show]` and `functionapp identity [assign|show]`

Eventhubs

- Initial release

Extension

- Added check to warn user if used distro is different then the one stored in package source file, as this may lead into errors

Interactive

- Fixed #5625: Persist history across different sessions
- Fixed #3016: History not recorded while in scope
- Fixed #5688: Completions did not appear if command table loading encountered an exception
- Fixed progress meter for long running operations

Monitor

- Deprecated the `monitor autoscale-settings` commands
- Added `monitor autoscale` commands
- Added `monitor autoscale profile` commands
- Added `monitor autoscale rule` commands

Network

- [BREAKING CHANGE] Removed `--tags` parameter from `route-filter rule create`
- Removed some erroneous default values for the following commands:
  - `network express-route update`
  - `network nsg rule update`
  - `network public-ip update`
  - `traffic-manager profile update`
  - `network vnet-gateway update`

- Added `network watcher connection-monitor` commands`
- Added `--vnet` and `--subnet` parameters to `network watcher show-topology`

Profile

- Deprecated `--msi` parameter for `az login`
- Added `--identity` parameter for `az login` to replace `--msi`

RDBMS

- [PREVIEW] Changed to use the API 2017-12-01-preview

Service Bus

- Initial release

Storage

- Fixed #4971: `storage blob copy` now supports other Azure clouds
- Fixed #5286: Batch commands `storage blob [delete-batch|download-batch|upload-batch]` no longer throw an error upon precondition failures

VM

- Added support to `[vm|vmss] create` to attach unmanaged data disks and configure caching
- Deprecated `[vm|vmss] assign-identity` and `[vm|vmss] remove-identity`
- Added `vm identity [assign|remove|show]` and `vmss identity [assign|remove|show]` commands to replace deprecated commands
- Changed default priority in `vmss create` to None

February 27, 2018

Version 2.0.28

Core

- Fixed #5184: Homebrew install issue
- Added support for extension telemetry with custom keys
- Added HTTP logging to `--debug`

ACS

- Changed to use the the `virtual-kubelet-for-aks` Helm chart for `aks install-connector` by default
- Fixed issue: Insuffient permission for service principals to create ACI container group issue
- Added `--aci-container-group`, `--location`, and `--image-tag` parameters to `aks install-connector`
- Removed deprecation notice from `aks get-versions`

Appservice

- Updates for new SDK version (azure-mgmt-web 0.35.0)
- Fixed #5538: `Free` reported as invalid SKU

Cognitive Services

- Updated the 'notice' when creating a new Cognitive Services account

Consumption

- Added new commands for pricesheet API
- Updated the existing Usage Details and Reservation Details formats

Container

- Added `--secrets` and `--secrets-mount-path` arguments to `container create` to use secrets in ACI

Network

- Fixed #5559: Missing client in `network vnet-gateway vpn-client generate`

Resource

- Changed `group deployment export` to display a partial template and errors on failure

Role

- Added `role assignment list-changelogs` to allow auditing of service principal roles

SQL

- Added zone redundancy support for databases and elastic pools on creation and update

Storage

- Enabled specifying destination-path/prefix for `storage blob [upload-batch|download-batch]`

VM

- Added suport for attaching/detatching disks on a single VMSS instance

February 13, 2018

Version 2.0.27

Core

- Changed authentication to key on both subscription ID and name on MSI login

ACS

- [BREAKING CHANGE] Renamed `aks get-versions` to `aks get-upgrades` in the interest of accuracy
- Changed `aks get-versions` to show Kubernetes versions available for `aks create`
- Changed `aks create` defaults to letting the server choose the version of Kubernetes
- Updated help messages referring to the service principal generated by AKS
- Changed default node sizes for `aks create` from "Standard_D1_v2" to "Standard_DS1_v2"
- Improved reliability when locating the dashboard pod for `az aks browse`
- Fixed `aks get-credentials` to handle Unicode errors when loading Kubernetes configuration files
- Added a message to `az aks install-cli` to help get `kubectl` in `$PATH`

Appservice

- Fixed issue where `webapp [backup|restore]` failed because of a null reference
- Added support for default app service plans through `az configure --defaults appserviceplan=my-asp`

CDN

- Added `cdn custom-domain [enable-https|disable-https]` commands

Container

- Added `--follow` option to `az container logs` for streaming logs
- Added `container attach` command that attaches local standard output and error streams to a container in a container group

CosmosDB

- Added support for setting capabilities

Extension

- Added support for `--pip-proxy` parameter to `az extension [add|update]` commands
- Added support for `--pip-extra-index-urls` argument to `az extension [add|update]` commands

Feedback

- Added extension information to telemetry data

Interactive

- Fixed issue where user is prompted to login when using interactive mode in Cloud Shell
- Fixed regression with missing parameter completions

IoT

- Fixed issue where `iot dps access policy [create|update]` would return a 'not found' error on success
- Fixed issue where `iot dps linked-hub [create|update]` would return a 'not found' error on success
- Added `--no-wait` support to `iot dps access policy [create|update]` and `iot dps linked-hub [create|update]`
- Changed `iot hub create` to allow specifying the number of partitions

Monitor

- Fixed `az monitor log-profiles create` command

Network

- Fixed the `--tags` option for the following commands:
  - `network public-ip create`
  - `network lb create`
  - `network local-gateway create`
  - `network nic create`
  - `network vnet-gateway create`
  - `network vpn-connection create`

Profile

- Enabled `az login` in from interactive mode

Resource

- Added back `feature show`

Role

- Added `--available-to-other-tenants` argument to `ad app update`

SQL

- Added `sql server dns-alias` commands
- Added `sql db rename`
- Added support for the `--ids` argument to all sql commands

Storage

- Added `storage blob service-properties delete-policy` and `storage blob undelete` commands to enable soft-delete

VM

- Fixed a crash when VM encryption may not be fully initialized
- Added principal ID output on enabling MSI
- Fixed `vm boot-diagnostics get-boot-log`

January 31, 2018

Version 2.0.26

Core

- Added support raw token retrival in MSI context
- Removed polling indicator string after finishing LRO on Windows cmd.exe
- Added a warning that appears when using a configured default has been changed to an INFO level entry. Use `--verbose` to see
- Add a progress indicator for wait commands

ACS

- Clarified `--disable-browser` argument
- Improved tab completion for `--vm-size` arguments

Appservice

- Fixed `webapp log [tail|download]`
- Removed the `kind` check on webapps and functions

CDN

- Fixed missing client issue with `cdn custom-domain create`

CosmosDB

- Fixed parameter description for failover policies

Interactive

- Fixed issue where command option completions no longer appeared

Network

- Added protection for `--cert-password` to `application-gateway create`
- Fixed issue with `application-gateway update` where `--sku` erroneously applied a default value
- Added protection for `--shared-key` and `--authorization-key` to `vpn-connection create`
- Fixed missing client issue with `asg create`
- Added `--file-name / -f` parameter for exported names to `dns zone export`
- Fixed the following issues with `dns zone export`:
  - Fixed issue where long TXT records were incorrectly exported
  - Fixed issue where quoted TXT records were incorrectly exported without escaped quotes
- Fixed issue where certain records were imported twice with `dns zone import`
- Restored `vnet-gateway root-cert` and `vnet-gateway revoked-cert` commands

Profile

- Fixed `get-access-token` to work inside a VM with identity

Resource

- Fixed bug with `deployment [create|validate]` where warning was incorrectly displayed when a template 'type' field contained uppercase values

Storage

- Fixed issue with migrating Storage V1 accounts to Storage V2
- Added progress reporting for all upload/download commands
- Fixed bug preventing "-n" arg option with `storage account check-name`
- Added 'snapshot' column to table output for `blob [list|show]`

- Fixed bugs with various parameters that needed to be parsed as ints

VM

- Added `vm image accept-terms` command to allow creating VMs from images with additional charges
- Fixed `[vm|vmss create]` to ensure commands can run under proxy with unsigned certificates
- [PREVIEW] Added support for "low" priority to VMSS
- Added protection for `--admin-password` to `[vm|vmss] create`

January 17, 2018

Version 2.0.25

ACR

- Added acr login fallback on Windows credential errors
- Enabled registry logs

ACS

- Fixed `get-credentials` command
- Removed SPN role requirement

Appservice

- Fixed bug with `config ssl upload` where `hosting_environment_profile` was null
- Added support for custom URLs to `browse`
- Fixed slot support for `log tail`

Backup

- Changed `--container-name` option of `backup item list` to be optional
- Added storage account options to `backup restore restore-disks`
- Fixed location check in `backup protection enable-for-vm` to be case insensitive
- Fixed issue where commands failed with an invalid container name
- Changed `backup item list` to include 'Health Status' by default

Batch

- Changed `batch login` to return authentication details

Cloud

- Changed to not require endpoints when setting `--profile` on a cloud

Consumption

- Added new commands for reservations: `consumption reservations summaries` and `consumption reservations details`

Event Grid

- [BREAKING CHANGE] Moved the `az eventgrid topic event-subscription` commands to `eventgrid event-subscription`
- [BREAKING CHANGE] Moved the `az eventgrid resource event-subscription` commands to `eventgrid event-subscription`
- [BREAKING CHANGE] Removed the `eventgrid event-subscription show-endpoint-url` command. Use `eventgrid event-subscription show --include-full-endpoint-url` instead
- Added command `eventgrid topic update`
- Added command `eventgrid event-subscription update`

- Added `--ids` parameter for `eventgrid topic` commands
- Added tab completion support for topic names

Interactive

- Fixed issue where interactive mode did not work with Python 2.x
- Fixed errors on startup
- Fixed issue with some commands not running in interactive mode

IoT

- Added support for device provisioning service
- Added deprecation messages in commands and command help
- Added IoT check to inform users of the IoT Extension

Monitor

- Added multi-diagnostic setting support. The `--name` parameter is now required for `az monitor diagnostic-settings create`
- Added command `monitor diagnostic-settings categories` to get diagnostic settings category

Network

- Fixed issue when trying to change to/from active-standby mode with `vnet-gateway update`
- Added support for HTTP2 to `application-gateway [create|update]`

Profile

- Added support for login with user assigned identities

Role

- Added `--assignee-object-id` argument to `role assignment create` to bypass graph query

Service Fabric

- Added detailed errors to validation response when creating cluster
- Fixed missing client issue with several commands

VM

- [PREVIEW] Cross-zone support for `vmss`
- [BREAKING CHANGE] Changed single-zone `vmss` default to "Standard" load balancer
- [BREAKING CHANGE] Changed `externalIdentities` to `userAssignedIdentities` for EMSI
- [PREVIEW] Added support for OS disk swap
- Added support for using VM images from other subscriptions
- Added `--plan-name`, `--plan-product`, `--plan-promotion-code` and `--plan-publisher` arguments to `[vm|vmss] create`
- Fixed error issues with `[vm|vmss] create`
- Fixed excessive resource usage caused by `vm image list --all`

December 19, 2017

Version 2.0.23

- Added support for login with user assigned identities

Container

- Fixed incorrect order of parameters for container logs

Network

- Added `--disable-bgp-route-propagation` argument to `route-table [create|update]`
- Added `--ip-tags` argument to `public-ip [create|update]`

Storage

- Added support for storage V2

VM

- [PREVIEW] Added support for user-assigned identities for VMs and VMSSes

December 5, 2017

Version 2.0.22

- Removed `az component` commands. Use `az extension` instead

Core

- Modified the `AZURE_US_GOV_CLOUD` AAD authority endpoint from login.microsoftonline.com to login.microsoftonline.us
- Fixed issue where telemetry would continuously resend

ACS

- Added `aks install-connector` and `aks remove-connector` commands
- Improved error reporting for `acs create`
- Fixed usage of `aks get-credentials -f` without fully-qualified path

Advisor

- Initial release

Appservice

- Fixed cert name generation with `webapp config ssl upload`
- Fixed `webapp [list|show]` and `functionapp [list|show]` to display correct apps
- Added default value for `WEBSITE_NODE_DEFAULT_VERSION`

Consumption

- Aded support for API version 2017-11-30

Container

- Fixed default ports regression

Monitor

- Added multi-dimension support to metrics command

Resource

- Added `--include-response-body` argument to `resource show`

Role

- Added display of default assignments for "classic" administraors to `role assignment list`
- Added suport to `ad sp reset-credentials` for adding credentials instead of overwriting
- Improved error reporting for `ad sp create-for-rbac`

SQL

- Added `sql db list-usages` and `sql db show-usage` commands
- Added `sql server conn-policy show` and `sql server conn-policy update` commands

VM

- Added zone information to `az vm list-skus`

November 14, 2017

Version 2.0.21

ACR

- Added support for creating webhooks in replication regions

ACS

- Changed all wording of "agent" to "node" in AKS
- Deprecated `--orchestrator-release` option for `acs create`
- Changed default VM size for AKS to `Standard_D1_v2`
- Fixed `az aks browse` on Windows
- Fixed `az aks get-credentials` on Windows

Appservice

- Added deployment source `config-zip` for webapps and function apps
- Added `--docker-container-logging` option to `az webapp log config`
- Removed the `storage` option from the parameter `--web-server-logging` of `az webapp log config`
- Improved error messages for `deployment user set`
- Added support for creating Linux function apps
- Fixed `list-locations`

Batch

- Fixed bug in pool create command when a resource ID was used with the `--image` flag

Batchai

- Added short option, `-s`, for `--vm-size` when providing VM size in `file-server create` command
- Added storage account name and key arguments to `cluster create` parameters
- Fixed documentation for `job list-files` and `job stream-file`
- Added short option, `-r`, for `--cluster-name` when providing cluster name in `job create` command

Cloud

- Changed `cloud [register|update]` to prevent registering clouds that have missing required endpoints

Container

- Added support to open multiple ports
- Added container group restart policy
- Added support to mount Azure File share as a volume
- Updated helper docs

Data Lake Analytics

- Changed `[job|account] list` to return more concise information

Data Lake Store

- Changed `account list` to return more concise information

Extension

- Added `extension list-available` to allow listing official Microsoft extensions
- Added `--name` to `extension [add|update]` to allow installing extensions by name

IoT

- Added support for certificate authorities (CA) and certificate chains

Monitor

- Added `activity-log alert` commands

Network

- Added support for CAA DNS records
- Fixed issue where endpoints could not be updated with `traffic-manager profile update`
- Fixed issue where `vnet update --dns-servers` didn't work depending on how the VNET was created
- Fixed issue where relative DNS names were incorrectly imported by `dns zone import`

Reservations

- Initial preview release

Resource

- Added support for resource IDs to `--resource` parameter and resource-level locks

SQL

- Added `--ignore-missing-vnet-service-endpoint` parameter to `sql server vnet-rule [create|update]`

Storage

- Changed `storage account create` to use SKU `Standard_RAGRS` as default
- Fixed bugs when dealing with file/blob names that include non-ascii chars
- Fixed bug that prevented using `--source-uri` with `storage [blob|file] copy start-batch`
- Added commands to glob and delete multiple objects with `storage [blob|file] delete-batch`
- Fixed issue when enabling metrics with `storage metrics update`
- Fixed issue with files over 200GB when using `storage blob upload-batch`
- Fixed issue where `--bypass` and `--default-action` were ignored by `storage account [create|update]`

VM

- Fixed a bug with `vmss create` that prevented using the `Basic` size tier
- Added `--plan` arguments to `[vm|vmss] create` for custom images with billing information
- Added `vm secret` [add|remove|list]` commands
- Renamed `vm format-secret` to `vm secret format`
- Added `--encrypt format` argument to `vm encryption enable`

October 24, 2017

Version 2.0.20

Core

- Updated `2017-03-09-profile` to consume `MGMT_STORAGE` API version `2016-01-01`

ACR

- Updated resource management to point to `2017-10-01` API version
- Changed 'bring your own storage' SKU to Classic
- Renamed registry SKUs to Basic, Standard, and Premium

ACS

- [PREVIEW] Added `az aks` commands
- Fixed kubernetes `get-credentials`

Appservice

- Fixed issue where downloaded `webapp` logs may be invalid

Component

- Added clearer deprecation message for all installers and confirmation prompt

Monitor

- Added `action-group` commands

Resource

- Fixed incompatibility with most recent version of msrest dependency in `group export`
- Fixed `policy assignment create` to work with built in policy definitions and policy set definitions

VM

- Added `--accelerated-networking` argument to `vmss create`

October 9, 2017

Version 2.0.19

Core

- Added handling of ADFS authority URLs with a trailing slash to Azure Stack

Appservice

- Added generic update with new command `webapp update`

Batch

- Updated to Batch SDK 4.0.0
- Updated `--image` option of VirtualMachineConfiguration to support ARM image references in addition to publish:offer:sku:version
- Added support for the new CLI extension model for Batch Extensions commands
- Removed Batch support from the component model

Batchai

- Initial release of Batch AI module

Keyvault

- Fixed Key Vault authentication issue when using ADFS on Azure Stack. (#4448)

Network

- Changed `--server` argument of `application-gateway address-pool create` to be optional, allowing for empty address pools
- Updated `traffic-manager` to support latest features

Resource

- Added support for `--resource-group/-g` options for resource group name to `group`
- Added commands for `account lock` to work with subscription-level locks
- Added commands for `group lock` to work with group-level locks
- Added commands for `resource lock` to work with resource-level locks

Sql

- Added support for SQL Transparent Data Encryption (TDE) and TDE with Bring Your Own Key
- Added `db list-deleted` command and `db restore --deleted-time` parameter, allowing the ability to find and

restore deleted databases
- Added `db op list` and `db op cancel`, allowing the ability to list and cancel in-progress operations on database

Storage

- Added support for file share snapshot

Vm

- Fixed a bug in `vm show` where using `-d` caused a crash on missing private ip addresses
- [PREVIEW] Added support for rolling upgrade to `vmss create`
- Added support for updating encryption settings with `vm encryption enable`
- Added `--os-disk-size-gb` parameter to `vm create`
- Added `--license-type` parameter for Windows to `vmss create`

September 22, 2017

Version 2.0.18

Resource

- Added support for showing built-in policy definitions
- Added support mode parameter for creating policy definitions
- Added support for UI definitions and templates to `managedapp definition create`
- [BREAKING CHANGE] Changed `managedapp` resource type from `appliances` to `applications` and `applianceDefinitions` to `applicationDefinitions`

Network

- Added support for availability zone to `network lb` and `network public-ip` subcommands
- Added support for IPv6 Microsoft Peering to `express-route`
- Added `asg` application security group commands
- Added `--application-security-groups` argument to `nic [create|ip-config create|ip-config update]`
- Added `--source-asgs` and `--destination-asgs` arguments to `nsg rule [create|update]`
- Added `--ddos-protection` and `--vm-protection` arguments to `vnet [create|update]`
- Added `network [vnet-gateway|vpn-client|show-url]` commands

Storage

- Fixed issue where `storage account network-rule` commands may fail after updating the SDK

Eventgrid

- Updated Azure Event Grid Python SDK to use newer API version "2017-09-15-preview"

SQL

- Changed `sql server list` argument `--resource-group` to be optional. If not specified, all sql servers in the subscription will be returned
- Added `--no-wait` param to `db [create|copy|restore|update|replica create|create|update]` and `dw [create|update]`

Keyvault

- Added support for Keyvault commands from behind a proxy

VM

- Added for support to availability zone to `[vm|vmss|disk] create`

- Fixed issue where using `--app-gateway ID` with `vmss create` would cause a failure
- Added `--asgs` argument to `vm create`
- Added support for running commands on VMs with `vm run-command`
- [PREVIEW] Added support for VMSS disk encryption with `vmss encryption`
- Added support for performing maintenance on VMs with `vm perform-maintenance`

ACS

- [PREVIEW] Added `--orchestrator-release` argument to `acs create` for ACS preview regions

Appservice

- Added ability to update and show authentication settings with `webapp auth [update|show]`

Backup

- Preview release

September 11, 2017

Version 2.0.17

Core

- Enabled command module to set its own correlation ID in telemetry
- Fixed JSON dump issue when telemetry is set to diagnostics mode

Acs

- Added `acs list-locations` command
- Made `ssh-key-file` come with expected default value

Appservice

- Added ability to create a webapp in a resource group other than the active service plan's

CDN

- Fixed 'CustomDomain is not interable' bug for `cdn custom-domain create`

Extension

- Initial Release

Keyvault

- Fixed issue where permissions were case sensitive for `keyvault set-policy`

Network

- Renamed `vnet list-private-access-services` to `vnet list-endpoint-services`
- Renamed `--private-access-services` argument to `--service-endpoints` for `vnet subnet create/update`
- Added support for multiple IP ranges and port ranges to `nsg rule create/update`
- Added support for SKU to `lb create`
- Added support for SKU to `public-ip create`

Resource

- Allow passing in resource policy parameter definitions in `policy definition create`, and `policy definition update`
- Allow passing in parameter values for `policy assignment create`
- Allow for passing JSON or file for all params
- Incremented API version

SQL

- Added `sql server vnet-rule` commands

VM

- Fixed: Don't assign access unless `--scope` is provided
- Fixed: Use the same extension naming as portal does
- Removed `subscription` from the `[vm|vmss] create` output
- Fixed: `[vm|vmss] create` storage SKU is not applied on data disks with an image
- Fixed: `vm format-secret --secrets` would not accept newline separated IDs

August 31, 2017

Version 2.0.16

Keyvault

- Fixed bug when trying to automatically resolve secret encoding with `secret download`

Sf

- Deprecating all commands in favor of Service Fabric CLI (sfctl)

Storage

- Fixed issue where storage accounts could not be created in regions that don't support the NetworkACLs feature
- Determine content type and content encoding during blob and file upload if neither content type and content encoding are specified

August 28, 2017

Version 2.0.15

CLI

- Added legal note to `--version`

ACS

- Corrected preview regions
- Formatted default `dns_name_prefix` properly
- Optimized acs command output

Appservice

- [BREAKING CHANGE] Fixed inconsistencies in the output of `az webapp config appsettings [delete|set]`
- Added a new alias of `-i` for `az webapp config container set --docker-custom-image-name`
- Exposed `az webapp log show`
- Exposed new arguments from `az webapp delete` to retain app service plan, metrics or dns registration
- Fixed: Detect slot settings correctly

IoT

- Fixed #3934: Policy creation no longer clears existing policies

Network

- [BREAKING CHANGE] Renamed `vnet list-private-access-services` to `vnet list-endpoint-services`
- [BREAKING CHANGE] Renamed option `--private-access-services` to `--service-endpoints` for `vnet subnet [create|update]`

- Added support for multiple IP and port ranges to `nsg rule [create|update]`
- Added support for SKU to `lb create`
- Added support for SKU to `public-ip create`

Profile

- Exposed `--msi` and `--msi-port` to login using a virtual machine's identity

Service Fabric

- Preview release
- Simplified registry user/password rules for command
- Fixed password prompt for user even after passing in the param
- Added support for empty `registry_cred`

Storage

- Enabled setting blob tier
- Added `--bypass` and `--default-action` arguments to `storage account [create|update]` to support service tunneling
- Added commands to add VNET rules and IP based rules to `storage account network-rule`
- Enabled service encryption by customer managed key
- [BREAKING CHANGE] Renamed `--encryption` option to `--encryption-services` for `az storage account create and az storage account update` command
- Fixed #4220: `az storage account update encryption` - syntax mismatch

VM

- Fixed issue where extra, erroneous information was displayed for `vmss get-instance-view` when using `--instance-id *`
- Added support for `--lb-sku` to `vmss create`:
- Removed human names from the admin name blacklist for `[vm|vmss] create`
- Fixed issue where `[vm|vmss] create` would throw an error if unable to extract plan information from an image
- Fixed a crash when creating a vmms scaleset with an internal LB
- Fixed issue where `--no-wait` argument did not work wth `vm availability-set create`

August 15, 2017

Version 2.0.14

ACS

- Corrected sshMaster0 port number for kubernetes

Appservice

- Fixed an exception when creatng a new git based Linux webapp

Event Grid

- Added SDK dependencies

August 11, 2017

Version 2.0.13

ACS

- Added more preview regions

Batch

- Updated to Batch SDK 3.1.0 and Batch Management SDK 4.1.0
- Added a new command show the task counts of a job
- Fixed bug in resource file SAS URL processing
- Batch account endpoint now supports optional 'https://' prefix
- Support for adding lists of more than 100 tasks to a job
- Added debug logging for loading Extensions command module

Component

- Added deprecation warning to 'az component' commands

Container

- `create`: Fixed issue where equals sign was not allowed inside an environment variable

Data Lake Store

- Enabled progress control

Event Grid

- Initial release

Network

- `lb`: Fixed issue where the certain child resource names did not resolve correctly when omitted
- `application-gateway {subresource} delete`: Fixed issue where `--no-wait` was not honored
- `application-gateway http-settings update`: Fixed issue where `--connection-draining-timeout` could not be turned off
- Fixed error unexpected keyword argument `sa_data_size_kilobyes` with `az network vpn-connection ipsec-policy add`

Profile

- `account list`: Added `--refresh` to sync up the latest subscriptions from server

Storage

- Enable update storage account with system assigned identity

VM

- `availability-set`: Exposed fault domain count on convert
- Exposed `list-skus` command
- Support to assign identity w/o creating role assignments
- Apply storage sku on attaching data disks
- Removed default os-disk name and storage SKU when using managed disks

July 28, 2017

Version 2.0.12

- Added container commands
- Added billing and consumption modules

```
azure-cli (2.0.12)

acr (2.0.9)
acs (2.0.11)
appservice (0.1.11)
batch (3.0.3)
billing (0.1.3)
cdn (0.0.6)
cloud (2.0.7)
cognitiveservices (0.1.6)
command-modules-nspkg (2.0.1)
component (2.0.6)
configure (2.0.10)
consumption (0.1.3)
container (0.1.7)
core (2.0.12)
cosmosdb (0.1.11)
dla (0.0.10)
dls (0.0.11)
feedback (2.0.6)
find (0.2.6)
interactive (0.3.7)
iot (0.1.10)
keyvault (2.0.8)
lab (0.0.9)
monitor (0.0.8)
network (2.0.11)
nspkg (3.0.1)
profile (2.0.9)
rdbms (0.0.5)
redis (0.2.7)
resource (2.0.11)
role (2.0.9)
sf (1.0.5)
sql (2.0.8)
storage (2.0.11)
vm (2.0.11)
```

Core

- Output sdk auth info for service principals with certificates
- Fixed deployment progress exceptions
- Use arm endpoint from the current cloud to create subscription client
- Improved concurrent handling of clouds.config file (#3636)
- Refresh client request id for each command execution
- Create subscription clients with right SDK profile (#3635)
- Progress Reporting for template deployments (#3510)
- Added support for picking table output fields through jmespath query (#3581)
- Improved the muting of parse args and append history with gestures (#3434)
- Create subscription clients with right SDK profile
- Move all existing recording files to latest folder
- Fixed idempotency for VM/VMSS create (#3586)
- Command paths are no longer case sensitive
- Certain boolean-type parameters are no longer case sensitive
- Support login to ADFS on prem server like Azure Stack
- Fixed concurrent writes to clouds.config (#3255)

ACR

- Added `show-usage` command for managed registries

- Support SKU update for managed registries
- Added managed registries with managed SKU
- Added webhooks for managed registries with acr webhook command module
- Added AAD authentication with acr login command
- Added delete command for docker repositories, manifests, and tags

ACS

- Support for API version 2017-07-01

Appservice

- Fixed bug where listing Linux webapp would return nothing
- Support to retrieve creds from acr
- Remove all commands under `appservice web`
- Mask docker registry passwords from command output (#3656)
- Ensure default browser is used on macOS without errors (#3623)
- Improve the help of `webapp log tail` and `webapp log download` (#3624)
- Exposed `traffic-routing` command to configure static routing (#3566)
- Added reliability fixes in configuring source control (#3245)
- Removed unsupported `--node-version` argument from `webapp config update` for Windows webapps. Instead use `webapp config appsettings set --settings WEBSITE_NODE_DEFAULT_VERSION=...`

Batch

- Updated to Batch SDK 3.0.0 with support for low-priority VMs in pools
- Renamed `pool create` option `--target-dedicated` to `--target-dedicated-nodes`
- Added `pool create` options `--target-low-priority-nodes` and `--application-licenses`

CDN

- Provided a better error message for `cdn endpoint list` when the profile specified by `--profile-name` does not exist

Cloud

- Changed API version of cloud metadata endpoint to YYYY-MM-DD format
- Gallery endpoint isn't required
- Support for registering cloud just with ARM resource manager endpoint
- Provided an option for `cloud set` to choose the profile while selecting current cloud
- Exposed `endpoint_vm_image_alias_doc`

CosmosDB

- Fixed allowing creation of collection with custom partition key
- Added support for collection default TTL

Data Lake Analytics

- Added commands for compute policy management under the `dla account compute-policy` heading
- Added `dla job pipeline show`
- Added `dla job recurrence list`

Data Lake Store

- Added support for user managed key vault key rotation in `dls account update`
- Updated underlying Data Lake Store filesystem SDK version, addressing a performance issue
- Added command `dls enable-key-vault`. This command attempts to enable a user provided Key Vault for use

encrypting the data ina Data Lake Store account

Interactive

- Improved the start up time by using cached commands
- Increased test coverage
- Enhanced the '?' gesture to also inject into the next command
- Fixed interactive errors with the profile 2017-03-09-profile-preview (#3587)
- Allowed `--version` as a parameter for interactive mode (#3645)
- Stop interactive mode throwing errors from validation completions (#3570)
- Progress reporting for template deployments (#3510)
- Added `--progress` flag
- Removed `--debug` and `--verbose` from completions
- Removed `interactive` from completions (#3324)

IoT

- Fixed policy creation no longer clears existing policies. (#3934)

Key vault

- Added commands for key vault recovery features:
  - `keyvault` subcommands `purge`, `recover`, `keyvault list-deleted`
  - `keyvault secret` subcommands `backup`, `restore`, `purge`, `recover`, `list-deleted`
  - `keyvault certificate` subcommands `purge`, `recover`, `list-deleted`
  - `keyvault key` subcommands `purge`, `recover`, `list-deleted`
- Added service principal key vault integration (#3133)
- Updated key vault dataplane to 0.3.2. (#3307)

Lab

- Added support for claiming any vm in the lab through `az lab vm claim`
- Added table output formatter for `az lab vm list` and `az lab vm show`

Monitor

- Fix for template file with `monitor autoscale-settings get-parameters-template` command (#3349)
- Renamed `monitor alert-rule-incidents list` to `monitor alert list-incidents`
- Renamed `monitor alert-rule-incidents show` to `monitor alert show-incident`
- Renamed `monitor metric-defintions list` to `monitor metrics list-definitions`
- Renamed `monitor alert-rules` to `monitor alert`
- Changed `monitor alert create`:
  - `condition` and `action` subcommands no longer accept JSON
  - Add numerous parameters to simplify the rule creation process
  - `location` no longer required
  - Add name and ID support for target
  - Remove `--alert-rule-resource-name`
  - Rename `is-enabled` to `enabled`, no longer required
  - `description` defaults now based on the supplied condition
  - Add examples to help clarifiy the new format
- Support names or IDs for `monitor metric` commands
- Added convenience arguments and examples to `monitor alert rule update`

Network

- Added `list-private-access-services` command
- Added `--private-access-services` argument to `vnet subnet create` and `vnet subnet update`
- Fixed issue where `application-gateway redirect-config create` would fail
- Fixed issue where `application-gateway redirect-config update` with `--no-wait` would not work
- Fixed bug when using `--servers` argument with `application-gateway address-pool create` and `application-gateway address-pool update`
- Added `application-gateway redirect-config` commands
- Added commands to `application-gateway ssl-policy`: `list-options`, `predefined list`, `predefined show`
- Added arguments to `application-gateway ssl-policy set`: `--name`, `--cipher-suites`, `--min-protocol-version`
- Added arguments to `application-gateway http-settings create` and `application-gateway http-settings update`: `--host-name-from-backend-pool`, `--affinity-cookie-name`, `--enable-probe`, `--path`
- Added arguments to `application-gateway url-path-map create` and `application-gateway url-path-map update`: `--default-redirect-config`, `--redirect-config`
- Added argument `--redirect-config` to `application-gateway url-path-map rule create`
- Added support for `--no-wait` to `application-gateway url-path-map rule delete`
- Added arguments to `application-gateway probe create` and `application-gateway probe update`: `--host-name-from-http-settings`, `--min-servers`, `--match-body`, `--match-status-codes`
- Added argument `--redirect-config` to `application-gateway rule create` and `application-gateway rule update`
- Added support for `--accelerated-networking` to `nic create` and `nic update`
- Removed `--internal-dns-name-suffix` argument from `nic create`
- Added support for `--dns-servers` to `nic update` and `nic create`: Add support for --dns-servers
- Fixed bug where `local-gateway create` ignored `--local-address-prefixes`
- Added support for `--dns-servers` to `vnet update`
- Fixed bug when creating a peering without route filtering with `express-route peering create`
- Fixed bug where `--provider` and `--bandwidth` arguments did not work with `express-route update`
- Fixed bug with `network watcher show-topology` defaulting logic
- Improved output formatting for `network list-usages`
- Use default frontend IP for `application-gateway http-listener create` if only one exists
- Use default address pool, HTTP settings, and HTTP listener for `application-gateway rule create` if only one exists
- Use default frontend IP and backend pool for `lb rule create` if only one exists
- Use default frontend IP for `lb inbound-nat-rule create` if only one exists

Profile

- Support login inside a VM with a managed identity
- Support output for `account show` in SDK auth file format
- Show deprecation warnings when using '--expanded-view'
- Added `get-access-token` command to provide raw AAD token
- Support login with a user account with no associated subscriptions

RDBMS

- Support listing servers across a subscription (#3417)
- Fixed `%s` not processed becasue of missing `% server_type` (#3393)
- Fixed doc source map and added CI task to verify (#3361)
- Fixed MySQL and PostgreSQL help (#3369)

Resource

- Improved prompts for missing parameters for `group deployment create`
- Improved parsing of `--parameters KEY=VALUE` syntax
- Fixed issues where `group deployment create` parameter files were no longer recognized using `@<file>` syntax
- Support `--ids` argument for `resource` and `managedapp` commands
- Fixed up some parsing and error messages (#3584)
- Fixed `--resource-type` parsing for the `lock` command to accept `<resource-namespace>` and `<resource-type>`
- Added parameter checking for template link templates (#3629)
- Added support for specifying deployment parameters using `KEY=VALUE` syntax

Role

- Support output in SDK auth file format for `create-for-rbac`
- Cleaned up role assignments and related AAD application when deleting a service principal (#3610)
- Include time format in `app create` args `--start-date` and `--end-date` descriptions
- Show deprecation warnings when using `--expanded-view`
- Added key vault integration to the `create-for-rbac` and `reset-credentials` commands

Service Fabric

- Fixed an issue with large files in applications being truncated on upload (#3666)
- Added tests for Service Fabric commands (#3424)
- Fixed numerous Service Fabric commands (#3234)

SQL

- Removed broken `sql server create` `--identity` parameter
- Removed password values from `sql server create` and `sql server update` command output
- Added commands `sql db list-editions` and `sql elastic-pool list-editions`

Storage

- Removed `--marker` option from `storage blob list`, `storage container list`, and `storage share list` commands (#3745)
- Enabled creating an https-only storage account
- Updated storage metrics, logging and cors commands (#3495)
- Rephrased exception message from CORS add (#3638) (#3362)
- Converted generator to a list in download batch command dry run mode (#3592)
- Fixed blob download batch dryrun issue (#3640) (#3592)

VM

- Support configuring nsg
- Fixed a bug where the DNS server would not be configured correctly
- Support managed service identities
- Fixed issue where `cmss create` with an existing load balancer required `--backend-pool-name`
- Make datadisks created with `vm image create` lun start with 0

May 10, 2017

Version 2.0.6

- documentdb renamed to cosmosdb
- Add rdbms (mysql, postgres)
- Include Data Lake Analytics and Data Lake Store modules

- Include Cognitive Services module
- Include Service Fabric module
- Include Interactive module (rename of az-shell)
- Add support for CDN commands
- Remove Container module
- Add 'az -v' as shortcut for 'az --version' (#2926)
- Improve performance of package load and command execution (#2819)

```
azure-cli (2.0.6)

acr (2.0.4)
acs (2.0.6)
appservice (0.1.6)
batch (2.0.4)
cdn (0.0.2)
cloud (2.0.2)
cognitiveservices (0.1.2)
command-modules-nspkg (2.0.0)
component (2.0.4)
configure (2.0.6)
core (2.0.6)
cosmosdb (0.1.6)
dla (0.0.6)
dls (0.0.6)
feedback (2.0.2)
find (0.2.2)
interactive (0.3.1)
iot (0.1.5)
keyvault (2.0.4)
lab (0.0.4)
monitor (0.0.4)
network (2.0.6)
nspkg (3.0.0)
profile (2.0.4)
rdbms (0.0.1)
redis (0.2.3)
resource (2.0.6)
role (2.0.4)
sf (1.0.1)
sql (2.0.3)
storage (2.0.6)
vm (2.0.6)
```

Core

- core: capture exceptions caused by unregistered provider and auto-register it
- perf: persist adal token cache in memory till process exits (#2603)
- Fix bytes returned from hex fingerprint -o tsv (#3053)
- Enhanced Key Vault Certificate Download and AAD SP Integration (#3003)
- Add Python location to 'az —version' (#2986)
- login: support login when there are no subscriptions (#2929)
- core: fix a failure when login using a service principal twice (#2800)
- core: Allow file path of accessTokens.json to be configurable through an env var (#2605)
- core: Allow configured defaults to apply on optional args (#2703)
- core: Improved performance
- core: Custom CA Certs - Support setting REQUESTS_CA_BUNDLE environment variable
- core: Cloud configuration - use 'resource manager' endpoint if 'management' endpoint not set

ACS

- fix the master and agent count to be integer instead of string
- expose 'az acs create --no-wait' and 'az acs wait' for async creation
- expose 'az acs create --validate' for dry-run validations
- remove windows profile before PUT call for scale command (#2755)

AppService

- functionapp: add full functionapp supports, including create, show, list, delete, hostname, ssl, etc
- Adding Team Services (vsts) as a continuous delivery option to "appservice web source-control config"
- Create "az webapp" to replace "az appservice web" (for backward compat, "az appservice web" will stay for 2 releases)
- Expose arguments to configure deployment and "runtime stacks" on webapp create
- Expose "webapp list-runtimes"
- support configure connection strings (#2647)
- support slot swap with preview
- Polish errors from appservice commands (#2948)
- Use the app service plan's resource group for cert operations (#2750)

CosmosDB

- Rename documentdb module to cosmosdb
- Added support for documentdb data-plane APIs: database and collection management
- Added support for enabling automatic failover on database accounts
- Added support for new consistency policy ConsistentPrefix

Data Lake Analytics

- Fix a bug where filtering on result and state for job lists would throw an error
- Add support for new catalog item type: package. accessed through: `az dla catalog package`
- Made it possible to list the following catalog items from within a database (no schema specification required):
  - Table
  - Table valued function
  - View
  - Table Statistics. This can also be listed with a schema, but without specifying a table name

Data Lake Store

- Update the version of the underlying filesystem SDK, which gives better support for handling server side throttling scenarios
- Improve performance of package load and command execution (#2819)
- missed help for access show. adding it. (#2743)

Find

- improve search results and allow for versioning of the search index

KeyVault

- BC: `az keyvault certificate download` change -e from string or binary to PEM or DER to better represent the options
- BC: Remove --expires and --not-before from `keyvault certificate create` as these parameters are not supported by the service
- Adds the --validity parameter to `keyvault certificate create` to selectively override the value in --policy

- Fixes issue in `keyvault certificate get-default-policy` where 'expires' and 'not_before' were exposed but 'validity_in_months' was not
- keyvault fix for import of pem and pfx (#2754)

Lab

- Adding create, show, delete & list commands for environment in the lab
- Adding show & list commands to view ARM templates in the lab
- Adding --environment flag in `az lab vm list` to filter VMs by environment in the lab
- Add convenience command `az lab formula export-artifacts` to export artifact scaffold within a Lab's formula
- Add commands to manage secrets within a Lab

Monitor

- Bug Fix: Modeling `--actions` of `az alert-rules create` to consume JSON string (#3009)
- Bug fix - diagnostic settings create does not accept logs/metrics from show commands (#2913)

Network

- Add `network watcher test-connectivity` command
- Add support for `--filters` parameter for `network watcher packet-capture create`
- Add support for Application Gateway connection draining
- Add support for Application Gateway WAF rule set configuration
- Add support for ExpressRoute route filters and rules
- Add support for TrafficManager geographic routing
- Add support for VPN connection policy-based traffic selectors
- Add support for VPN connection IPSec policies
- Fix bug with `vpn-connection create` when using the `--no-wait` or `--validate` parameters
- Add support for active-active VNet gateways
- Remove nulls values from output of `network vpn-connection list/show` commands
- BC: Fix bug in the output of `vpn-connection create`
- Fix bug where '--key-length' argument of 'vpn-connection create' was not parsed correctly
- Fix bug in `dns zone import` where records were not imported correctly
- Fix bug where `traffic-manager endpoint update` did not work
- Add 'network watcher' preview commands

Profile

- Support login when there are no subscriptions found (#2560)
- Support short param name in az account set --subscription (#2980)

Redis

- Adding update command which also adds the ability to scale for redis cache
- Deprecates the 'update-settings' command

Resource

- Add managedapp and managedapp definition commands (#2985)
- Support 'provider operation' commands (#2908)
- Support generic resource create (#2606)
- Fix resource parsing and api version lookup. (#2781)
- Add docs for az lock update. (#2702)
- Error out if you try to list resources for a group that doesn't exist. (#2769)

- [Compute] Fix issues with VMSS and VM availability set update. (#2773)
- Fix lock create and delete if parent-resource-path is None (#2742)

Role

- create-for-rbac: ensure SP's end date will not exceed certificate's expiration date (#2989)
- RBAC: add full support for 'ad group' (#2016)
- role: fix issues on role definition update (#2745)
- create-for-rbac: ensure user provided password is picked up

SQL

- Added az sql server list-usages and az sql db list-usages commands
- SQL - ability to connect directly to resource provider (#2832)

Storage

- Default location to resource group location for `storage account create`
- Add support for incremental blob copy
- Add support for large block blob upload
- Change block size to 100MB when file to upload is larger than 200GB

VM

- avail-set: make UD&FD domain counts optional

  note: VM commands in sovereign clouds Please avoid managed disk related features, including the following:

  1. az disk/snapshot/image
  2. az vm/vmss disk
  3. Inside "az vm/vmss create", use "—use-unmanaged-disk" to avoid managed disk Other commands should work
- vm/vmss: improve the warning text when generates ssh key pairs
- vm/vmss: support create from a market place image which requires plan info (#1209)

April 3, 2017

Version 2.0.2

We released the ACR, Batch, KeyVault, and SQL components in this release

```
azure-cli (2.0.2)

acr (2.0.0)
acs (2.0.2)
appservice (0.1.2)
batch (2.0.0)
cloud (2.0.0)
component (2.0.0)
configure (2.0.2)
container (0.1.2)
core (2.0.2)
documentdb (0.1.2)
feedback (2.0.0)
find (0.0.1b1)
iot (0.1.2)
keyvault (2.0.0)
lab (0.0.1)
monitor (0.0.1)
network (2.0.2)
nspkg (2.0.0)
profile (2.0.2)
redis (0.1.1b3)
resource (2.0.2)
role (2.0.1)
sql (2.0.0)
storage (2.0.2)
vm (2.0.2)
```

Core

- Add acr, lab, monitor, and find modules to default list
- Login: skip erroneous tenant (#2634)
- login: set default subscription to one with the state of "Enabled" (#2575)
- Add wait commands and --no-wait support to more commands (#2524)
- core: support login using service principal with a cert (#2457)
- Add prompting for missing template parameters. (#2364)
- Support setting default values for common arguments like default resource group, default web, default vm
- Support login to specific tenant

ACS

- [ACS] Adding support for configuring a default ACS cluster (#2554)
- Add support for ssh key password prompting. (#2044)
- Add support for windows clusters. (#2211)
- Switch from Owner to Contributor role. (#2321)

AppService

- appservice: support to get external ip address used for DNS A records (#2627)
- appservice: support binding wildcard certificates (#2625)
- appservice: support list publishing profiles (#2504)
- AppService - Trigger source control sync after config (#2326)

DataLake

- Initial release of Data Lake Analytics module
- Initial release of Data Lake Store module

DocuemntDB

- DocumentDB: Adding support for listing connection strings (#2580)

VM

- [Compute] Add AppGateway support to virtual machine scale set create (#2570)
- [VM/VMSS] Improved disk caching support (#2522)
- VM/VMSS: incorporate credentials validation logic used by portal (#2537)
- Add wait commands and --no-wait support (#2524)
- Virtual machine scale set: support * to list instance view across vms (#2467)
- Add --secrets for VM and virtual machine scale set ([#2212}(https://github.com/Azure/azure-cli/pull/2212))
- Allow VM creation with specialized VHD (#2256)

February 27, 2017

Version 2.0.0

This release of Azure CLI 2.0 is the first "Generally Available" release General availability applies to these command modules:

- Container Service (acs)
- Compute (including Resource Manager, VM, virtual machine scale sets, Managed Disks)
- Networking
- Storage

These command modules can be used in production and are supported by standard Microsoft SLA You can open issues directly with Microsoft support or on our github issues list You can ask questions on StackOverflow using the azure-cli tag, or contact the product team at azfeedback@microsoft.com You can provide feedback from the command line with the `az feedback` command

The commands in these modules are stable and the syntax is not expected to change in upcoming releases of this version of Azure CLI

To verify the version of the CLI, use `az --version` The output lists the version of the CLI itself (2.0.0 in this release), the individual command modules, and the versions of Python and GCC that you're using

```
azure-cli (2.0.0)

acs (2.0.0)
appservice (0.1.1b5)
batch (0.1.1b4)
cloud (2.0.0)
component (2.0.0)
configure (2.0.0)
container (0.1.1b4)
core (2.0.0)
documentdb (0.1.1b2)
feedback (2.0.0)
iot (0.1.1b3)
keyvault (0.1.1b5)
network (2.0.0)
nspkg (2.0.0)
profile (2.0.0)
redis (0.1.1b3)
resource (2.0.0)
role (2.0.0)
sql (0.1.1b5)
storage (2.0.0)
vm (2.0.0)

Python (Darwin) 2.7.10 (default, Jul 30 2016, 19:40:32)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)]
```

> **NOTE**
>
> Some of the command modules have a "b*n*" or "rc*n*" postfix These command modules are still in preview and will become generally available in the future

We also have nightly preview builds of the CLI For information, see these instructions on getting the nightly builds, and these instructions on developer setup and contributing code

You can report issues with nightly preview builds in the following ways:

- Report issues in our github issues list
- Contact the product team at azfeedback@microsoft.com
- Provide feedback from the command line with the `az feedback` command

Differences between Azure CLI products

1/17/2019 • 2 minutes to read • Edit Online

As of the end of June 2018, explicit version numbers have been removed from Azure CLI product names. This change helps eliminate confusion that sometimes showed up in documentation where users were told to use "the Azure CLI" but it was unclear what version of the product was being referenced. If you're familiar with the old product names, here is how they have changed:

- Azure CLI versions 2.0 and later are now referred to only as "Azure CLI."
- Earlier Azure CLI versions (1.x and lower) are now referred to as "Azure classic CLI."

The name change to Azure classic CLI makes it clear that this tool is meant to be used only with the classic deployment model. The classic CLI is also no longer updated or maintained. For this reason, and many more, it's recommended that you move any classic deployments to use the Azure Resource Manager model and migrate to the latest available version of the Azure CLI.

If you are still using the classic CLI, you can learn about the process of migrating in the following articles:

- Migrate from Classic to Azure Resource Manager
- Install the Azure CLI
- Migrating from Azure classic CLI to Azure CLI