# Contributing to Azure PowerShell

This repository contains PowerShell cmdlets for developers and administrators to develop, deploy, and manage Microsoft Azure applications.

## Basics

If you would like to become an active contributor to this project (or any other open source Microsoft project), please see the list of resources found on [this page](https://opensource.microsoft.com/resources).

## Before Starting

### Onboarding

All users must sign the [Microsoft Contributor License Agreement (CLA)](https://cla.opensource.microsoft.com/) before any code contribution can be made. For Microsoft employees, make sure that your GitHub account is part of the Azure organization. [Use this page](http://aka.ms/azuregithub) to link your account.

### Code of Conduct

This project has adopted the [Microsoft Open Source Code of Conduct](https://opensource.microsoft.com/codeofconduct/). For more information see the [Code of Conduct FAQ](https://opensource.microsoft.com/codeofconduct/faq/) or contact [opencode@microsoft.com](mailto:opencode@microsoft.com) with any additional questions or comments.

### GitHub Basics

#### GitHub Workflow

If you don't have much experience using GitHub or Git, [here is a guide to understanding the GitHub flow](https://guides.github.com/introduction/flow/) and [here is a guide to understanding the basic Git commands](https://education.github.com/git-cheat-sheet-education.pdf).

#### Forking the Azure/azure-powershell repository

Unless you are working with multiple contributors on the same file, we ask that you fork the repository and submit your pull request from there. [Here is a guide to forks in GitHub](https://guides.github.com/activities/forking/).

## Filing Issues

You can find all of the issues that have been filed in the [Issues](https://github.com/Azure/azure-powershell/issues) section of the repository.

To file an issue, first select one of the [provided templates](https://github.com/Azure/azure-powershell/issues/new/choose) to ensure that the proper information is provided. The following are a few of the templates we have:

- [\_Az module bug report\_](https://github.com/Azure/azure-powershell/issues/new?assignees=&labels=&template=az-module-bug-report.md&title=)

- [\_AzureRM module bug report\_](https://github.com/Azure/azure-powershell/issues/new?assignees=&labels=&template=azurerm-module-bug-report.md&title=)

- [\_Feature request\_](https://github.com/Azure/azure-powershell/issues/new?assignees=&labels=Feature+Request&template=feature\_request.md&title=)

You can find the code complete and release dates of the next three Azure PowerShell releases in the [Milestones](https://github.com/Azure/azure-powershell/milestones) section of the Issue page. Each milestone will display the issues that are being worked on for the corresponding release.

## Submitting Changes

### Pull Requests

You can find all of the pull requests that have been opened in the [Pull Request](https://github.com/Azure/azure-powershell/pulls) section of the repository.

When creating a pull request, keep the following in mind:

- Make sure you are pointing to the fork and branch that your changes were made in

- Choose the correct branch you want your pull request to be merged into

- The \*\*master\*\* branch is for active development; changes in this branch will be in the next Azure PowerShell release

- The \*\*preview\*\* branch is a snapshot of the last `AzureRM` release and \_should not\_ be used for active development

- The \*\*release-X.X.X\*\* branch is for active development during a release

- The pull request template that is provided \*\*should be filled out\*\*; this is not something that should just be deleted or ignored when the pull request is created

- Deleting or ignoring this template will elongate the time it takes for your pull request to be reviewed

- The SLA for reviewing pull requests is \*\*two business days\*\*

### Pull Request Guidelines

A pull request template will automatically be included as a part of your PR. Please fill out the checklist as specified. Pull requests \*\*will not be reviewed\*\* unless they include a properly completed checklist.

The following is a list of guidelines that pull requests opened in the Azure PowerShell repository must adhere to. You can find a more complete discussion of Azure PowerShell design guidelines [here](documentation/development-docs/design-guidelines).

#### General guidelines

The following guidelines must be followed in \*\*EVERY\*\* pull request that is opened.

- Title of the pull request is clear and informative

- The appropriate `ChangeLog.md` file(s) has been updated:

- For any service, the `ChangeLog.md` file can be found at `src/{{SERVICE}}/{{SERVICE}}/ChangeLog.md`

- A snippet outlining the change(s) made in the PR should be written under the `## Upcoming Release` header -- no new version header should be added

- There are a [small number of commits](documentation/development-docs/cleaning-up-commits.md) that each have an informative message

- All files shipped with a module should contain a proper Microsoft license header

- For public API changes to cmdlets:

- a cmdlet design review was approved for the changes in [this repository](https://github.com/Azure/azure-powershell-cmdlet-review-pr) (\_Microsoft internal only\_)

- the markdown help files have been regenerated using the commands listed [here](documentation/development-docs/help-generation.md#updating-all-markdown-files-in-a-module)

#### Testing guidelines

The following guidelines must be followed in \*\*EVERY\*\* pull request that is opened.

- Changes made have corresponding test coverage

- Tests should not include any hardcoded values, such as location, resource id, etc.

- Tests should have proper setup of resources to ensure any user can re-record the test if necessary

- No existing tests should be skipped

#### Cmdlet guidelines

Please see the [\_Cmdlet Best Practices\_](./documentation/development-docs/design-guidelines/cmdlet-best-practices.md) document for information about cmdlet design guidelines.

Examples of standard cmdlets that follow Azure PowerShell patterns and best practices can be found [here](./documentation/development-docs/examples).

#### Parameter guidelines

Please see the [\_Parameter Best Practices\_](./documentation/development-docs/design-guidelines/parameter-best-practices.md) document for information about parameter design guidelines.

#### Piping guidelines

Please see the [\_Piping Best Practices\_](./documentation/development-docs/design-guidelines/piping-best-practices.md) document for information about piping design guidelines.

#### Module guidelines

Please see the [\_Module Best Practices\_](./documentation/development-docs/design-guidelines/module-best-practices.md) document for information about module design guidelines.