Azure CLI Plugin

Plugin Information
View Azure CLI on the plugin site for more information.

A Jenkins plugin to use Azure CLI for managing Azure resources.

The advantage of this plugin that it let's you export the CLI result from each command to environment variables and to the next command.

How to Install

You can install/update this plugin in Jenkins update center (Manage Jenkins -> Manage Plugins, search Azure CLI Plugin).

You can also manually install the plugin if you want to try the latest feature before it's officially released. To manually install the plugin:

1. Clone the repo and build:

   ```
   mvn package
   ```

2. Open your Jenkins dashboard, go to Manage Jenkins -> Manage Plugins.
3. Go to Advanced tab, under Upload Plugin section, click Choose File.
4. Select `azure-cli.hpi` in `target` folder of your repo, click Upload.
5. Restart your Jenkins instance after install is completed.

Deploy & Manage Azure Resources

Prerequisites

To use this plugin, first you need to have an Azure Service Principal in your Jenkins instance.

1. Create an Azure Service Principal through Azure CLI or Azure portal.
2. Open Jenkins dashboard, go to Credentials, add a new Microsoft Azure Service Principal with the credential information you just created.
3. Install Azure CLI in the Jenkins Host

How to use

1. Select Azure CLI Plugin in the Build Steps.
2. Select the Azure Service Principal
3. Type a command such as `az vm create -n MyLinuxVM -g MyResourceGroup --image UbuntuLTS --data-disk-sizes-gb 10 20`
4. You can also use environment variables: `az vm create -n ${VM_NAME} -g MyResourceGroup --image UbuntuLTS --data-disk-sizes-gb 10 20`

Export output as environment variables

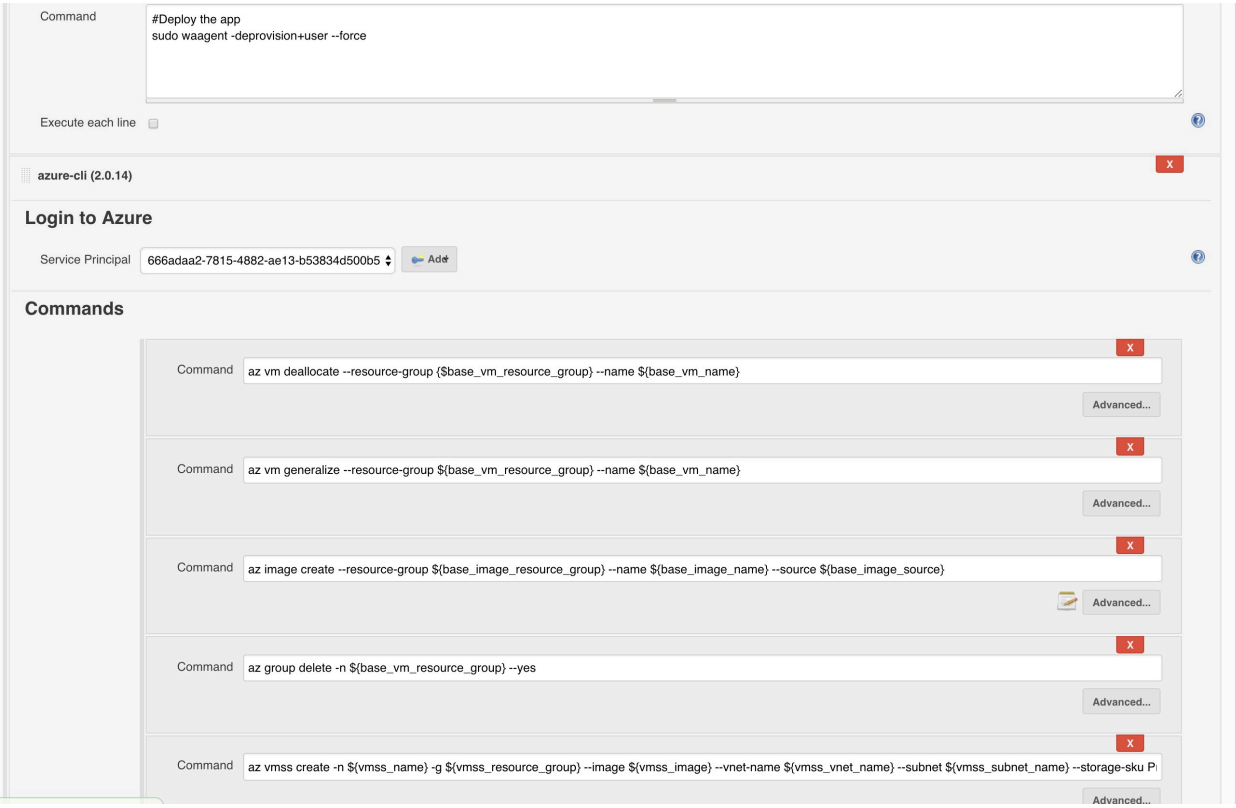
The CLI command output is JSON based:

Output:

```
   {
       "fqdns": "",
       "id": "/subscriptions/some-guid/resourceGroups/MyResourceGroup/providers/Microsoft.Compute
/virtualMachines/MyLinuxVM",
       "location": "northeurope",
       "macAddress": "00-0D-AA-AA-AA-AA",
       "powerState": "VM running",
       "privateIpAddress": "10.0.0.4",
       "publicIpAddress": "52.178.0.0",
       "properties": {
          "provisioningState": "Succeeded"
       },
       "resourceGroup": "MyResourceGroup"
}
```

If you want to export a property to an environment variable that you can use in other build steps, define the parameters in the "advanced" section:

1. `/publicIpAddress|PUBLIC_IP` The '/publicIpAddress' is the path in the JSON and the 'PUBLIC_IP' is the environment variable that will be created.
2. Nested property: `/properties/provisioningState|STATE`
3. Multiple environment variables: `/publicIpAddress|PUBLIC_IP , /properties/provisioningState|STATE`



Deploy using Pipeline

You can also use this plugin in pipeline (Jenkinsfile). Here are some samples to use the plugin in pipeline script:

To create a new resource group and provision a new VM:

```
azureCLI commands: [[exportVariablesString: '', script: 'az group create -n MyResourceGroup --location
northeurope'], [exportVariablesString: '/publicIpAddress|PUBLIC_IP', script: 'az vm create -n MyLinuxVM
-g MyResourceGroup --image UbuntuLTS --data-disk-sizes-gb 10 20']], principalCredentialId: '<credential_
id>'
```

For advanced options, you can use Jenkins Pipeline Syntax tool to generate a sample script.

Deploy using Job DSL

You can also use this plugin with using the Jobs DSL. For example:

To create a linux VM using the CLI:

```
job('AzCommand') {
  steps {
        azCommands('servicePrincipalId',
                        ['az vm create -n MyLinuxVM -g MyResourceGroup --image UbuntuLTS --data-disk-
sizes-gb 10 20 && /publicIpAddress|PUBLIC_IP'])
    }
}
```

Changelog

Version 0.9 (2018-6-28)

- Fixed runtime exec in Linux

Version 0.8 (2018-6-19)

- Fixed run in Windows bug

Version 0.7 (2018-6-4)

- Disabled print login command

Version 0.6 (2017-11-3)

- Fixed pipeline with exported environment variables support

Version 0.5 (2017-9-8)

- Added sort commands

Version 0.4 (2017-8-20)

- Change env var to ${..}

Version 0.2 (2017-8-17)

- Added pipeline support

Version 0.1 (2017-8-16)

- Initial release
- Support Azure CLI 2.0
- Support export environment variable from each command output
- Deploy using Job DSL