Azure VM Agents plugin

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

Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- Unauthorized users could change agent IP configuration
- Information disclosure
- Users with Overall/Read access can enumerate credential IDs

A Jenkins Plugin to create Jenkins agents in Azure Virtual Machines (via Azure ARM template).

Supported features:

1. Windows Agents on Azure Cloud using SSH and JNLP
   - For Windows images to launch via SSH, the image needs to be preconfigured with SSH.
   - For preparing custom windows image, refer to Azure documentation
2. Linux Agents on Azure Cloud using SSH
   - For preparing custom linux image, refer to Azure documentation

How to Install

You can install/update this plugin in Jenkins update center (Manage Jenkins -> Manage Plugins, search Azure VM Agents 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-vm-agents.hpi` in `target` folder of your repo, click Upload.
5. Restart your Jenkins instance after install is completed.

Prerequisites

To use this plugin to create VM agents, 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.

Configure the Plugin

Add a New Azure VM Agents Cloud

1. Within the Jenkins dashboard, click Manage Jenkins -> Configure System -> Scroll to the bottom of the page and find the section with the dropdown "Add a new cloud" -> click on it and select "Microsoft Azure VM Agents"
2. Provide a name for the cloud (plugin will generate one for you if you leave it empty, but it's recommended to give it a meaningful name).
3. Select an existing account from the Azure Credentials dropdown or add new "Microsoft Azure Service Principal" credentials in the Credentials Management page by filling out the Subscription ID, Client ID, Client Secret and the OAuth 2.0 Token Endpoint.
4. Click on “Verify configuration” to make sure that the profile configuration is done correctly.
5. Save and continue with the template configuration (See instructions below).

Add a New Azure VM Agent Template

1. Click the "Add" in "Add Azure Virtual Machine Template" to add a template. A template is used to define an Azure VM Agent configuration, like its VM size, region, or retention time.
2. Provide meaningful name and description for your new template.
3. Provide one or more meaningful labels, e.g. "Windows" or "Linux". Label is used by job to determine which agents will be selected to run the job, so please make sure you give it a meaningful label.
4. Select the desired region, and VM size from dropdown list.
5. Select the Storage Account Type, either Standard_LRS or Premium_LRS. Note that some VM size only supports Standard_LRS.

6. Specify the Azure Storage account name or select an existing Storage account name for storing VM's OS disk. If you choose to create a new one but leave the name blank the plugin will generate a name for you.
7. Select the disk type between Managed Disk (recommended) or Unmanaged Disk.
8. Select the retention strategy

    - Idle Retention Strategy. You can specify the retention time in minutes. This defines the number of minutes Jenkins can wait before automatically deleting an idle agent. Specify 0 if you do not want idle agents to be deleted automatically.
    - Pool Retention Strategy. This retention strategy help you to maintain amount of agents in a specific number. You can specify the retention time in hour and the pool size.
    - Once Retention Strategy. This retention strategy make sure to use one agent only once.

    Retention time define the time of hour before automatically deleting since the agent created. And the pool size define the agent pool size you want to maintain. If you change your cloud name, template name or most of parameters (e.g. Region, Image), we will delete the existing agents at once and provision the new one according to your new template. But if you only change your Retention Time or Pool Size, we will only scale in, scale out or do nothing for you.
9. Select a usage option:

    - If "Utilize this node as much as possible" is selected, then Jenkins may run any job on the agent as long as it is available.
    - If "Only build jobs with label expressions matching this node" is selected, Jenkins will only build a project on this node when that project is restricted to certain nodes using a label expression, and that expression matches this node's name and/or labels. This allows an agent to be reserved for certain kinds of jobs.
10. Select a built-in image, you can choose between Windows Server 2016 and Ubuntu 16.04 LTS. You can also choose to install some tools on the agent, including Git, Maven and Docker (JDK is always installed).
11. Specify Admin Credentials (a username/password credentials), this is the username and password if you want to log into the agent VM.
12. Click Verify Template to make sure all your configurations are correct, then Save.

Run Jenkins Jobs on Azure VM Agents

After you configured an Azure VM agent template, when you run a new Jenkins job, Jenkins will automatically provision a new Azure VM only if there is no executor available.

A more common scenario is you want to restrict some jobs to always be running on a particular VM agent instead of Jenkins master. To achieve that:

1. Open your Jenkins project, under General, check "Restrict where this project can be run".
2. In Label Expression, fill in the label you assigned to your VM template.
3. Save and run the job, you'll see your job is running on the VM agent even if Jenkins master is free.

For how to select agent in pipeline, refer to this doc.

Use a Custom VM Image

The built-in image only has a clean Windows or Ubuntu OS and some tools like Git and Maven installed, in some cases, you may want to have more customization on the image. To use a custom image:

1. In Image Configuration, select "Use Advanced Image Configurations".
2. Choose between two possible alternatives:

    - Use a custom user image (provide image URL and OS type - note, your custom image has to be available into the same storage account in which you are going to create agent nodes);
    - Using any marketplace image by specifying an image reference (provide image reference by publisher, offer, sku and version). You can get the publisher, offer and sku by looking at the ARM template of that image.
3. For the launch method, select SSH or JNLP.

    - Linux agents can be launched only using SSH.
    - Windows agents can be launched using SSH or JNLP. For Windows agents, if the launch method is SSH then check Pre-Install SSH in Windows Slave or image needs to be custom-prepared with an SSH server pre-installed.

    We recommend to use SSH rather than JNLP, for you need less init codes and get much clearer logs.

    > *When using the JNLP launch option, ensure the following:*
    >
    > - *Jenkins URL (Manage Jenkins -> Configure System -> Jenkins Location)*
    > - *The URL needs to be reachable by the Azure agent, so make sure to configure any relevant firewall rules accordingly.*
    > - *TCP port for JNLP agent agents (Manage Jenkins -> Configure Global Security -> Enable security -> TCP port for JNLP agents).*
    > - *The TCP port needs to be reachable from the Azure agent launched using JNLP. It is recommended to use a fixed port so that any necessary firewall exceptions can be made.*
    >
    > *If the Jenkins master is running on Azure, then open an endpoint for "TCP port for JNLP agent agents" and, in case of Windows, add the necessary firewall rules inside virtual machine (Run -> firewall.cpl).*

4. For the Initialization Script, you can provide a script that will be executed after the VM is provisioned. This allows to install any app/tool you need on the agent. Please be noted you need to at least install JRE if the image does not have Java pre-installed. We prepared a sample script for Linux via SSH, Windows via SSH and Windows via JNLP. Please find details in help button.

If you hit the storage scalability limits for your custom images on the storage account where the VHD resides, you should consider using the agent's temporary storage or copy your custom image in multiple storage accounts and use multiple VM templates with the same label within the same agent cloud.

For more details about how to prepare custom images, refer to the below links:

- Capture Windows Image
- Capture Linux Image

Init script should finish in 20 minutes (this time can be configured in Deployment Timeout setting of Azure Profile Configuration). It's not recommended to run complex init script, if the init script is expected to take a long time to complete, it is recommended to use a custom-prepared image.

Advanced Configurations

If you choose Use Advanced Image Configurations, you can click on Advanced button where you can find more VM configurations:

1. Virtual Network Name, Virtual Network Resource Group Name and Subnet name: by default the VM does not belong to any virtual network, you can provide one if you want the VM to be in a virtual network for network security. Please be noted the virtual network must exist.
2. Make VM agent IP private: by default the plugin will create a public IP for the VM so it's public accessible on internet. Check this option if you don't want the public IP to be created.

> *Make VM agent IP private can make the VM more secure, but if you configured to use SSH to launch agent, Jenkins master needs to be able to access the VM. So in this case you need to also specify virtual network and subnet name so the agent and Jenkins master are in the same subnet.*

3. Network Security Group Name: add the VM to a network security group.
4. JVM Options: specify JVM options.
5. Number of Executors: specify the number concurrent builds that a VM agent can run at the same time.
6. Disable template: disable this template temporarily.

Configure VM Template using Groovy Script

It is recommended that you use the `configuration-as-code` plugin for automating the plugin configuration. If you can't do that for some reason then you can use groovy script.

Here is a sample groovy script that creates a new Azure cloud and VM template. You can run it in Manage Jenkins -> Script Console.

Configure cloud with built-in image

```
import com.microsoft.azure.vmagent.builders.*;

def myCloud = new AzureVMCloudBuilder()
    .withCloudName("myAzure")
    .withAzureCredentialsId("<your azure credential ID>")
    .withNewResourceGroupName("<your Resource Group Name>")
    .addNewTemplate()
        .withName("ubuntu")
        .withLabels("ubuntu")
        .withLocation("East US")
        .withVirtualMachineSize("Standard_DS2_v2")
        .withNewStorageAccount("<your Storage Account Name>")
        .addNewBuiltInImage()
            .withBuiltInImageName("Ubuntu 16.14 LTS")
            .withInstallGit(true)
            .withInstallMaven(true)
            .withInstallDocker(true)
        .endBuiltInImage()
        .withAdminCredential("<your admin credential ID>")
    .endTemplate()
    .build();

Jenkins.getInstance().clouds.add(myCloud);
```

Configure cloud with mutli-template of advanced images

```
import com.microsoft.azure.vmagent.builders.*;

def firstTemplate = new AzureVMTemplateBuilder()
    .withName("first-template")
    .withLabels("ubuntu")
    .withLocation("East US")
    .withVirtualMachineSize("Standard_DS2_v2")
    .withNewStorageAccount("<your Storage Account Name>")
    .addNewAdvancedImage()
        .withReferenceImage("Canonical", "UbuntuServer", "16.04-LTS", "latest")
        .withInitScript("sudo add-apt-repository ppa:openjdk-r/ppa -y \n" +
                        "sudo apt-get -y update \n" +
                        "sudo apt-get install openjdk-8-jre openjdk-8-jre-headless openjdk-8-jdk -y")
    .endAdvancedImage()
    .withAdminCredential("<your admin credential ID>")
    .build();

def myCloud = new AzureVMCloudBuilder()
    .withCloudName("myAzure")
    .withAzureCredentialsId("<your azure credential ID>")
    .withNewResourceGroupName("<your Resource Group Name>")
    .addToTemplates(firstTemplate)
    .addNewTemplate()
        .withName("second-template")
        .withLabels("windows")
        .withLocation("Southeast Asia")
        .withVirtualMachineSize("Standard_DS2_v2")
        .withNewStorageAccount("<your Storage Account Name>")
        .addNewAdvancedImage()
            .withReferenceImage("MicrosoftWindowsServer", "WindowsServer", "2016-Datacenter", "latest")
        .endAdvancedImage()
        .withAdminCredential("<your admin credential ID>")
    .endTemplate()
    .build();

Jenkins.getInstance().clouds.add(myCloud);
```

inherit existing template

```
import com.microsoft.azure.vmagent.builders.*;
import com.microsoft.azure.vmagent.*;

AzureVMAgentTemplate baseTemplate = new AzureVMTemplateBuilder()
    .withLocation("Southeast Asia")
    .withVirtualMachineSize("Standard_DS2_v2")
    .withStorageAccountType("Premium_LRS")
    .withNewStorageAccount("<your Storage Account Name>")
    .addNewAdvancedImage()
         .withReferenceImage("Canonical", "UbuntuServer", "16.04-LTS", "latest")
    .endAdvancedImage()
    .withAdminCredential("<your admin credential ID>")
    .build();

def myCloud = new AzureVMCloudBuilder()
    .withCloudName("myAzure")
    .withAzureCredentialsId("<your azure credential ID>")
    .withNewResourceGroupName("<your Resource Group Name>")
    .addNewTemplateLike(baseTemplate)
        .withName("inherit")
        .withLabels("inherit")
        .addNewAdvancedImageLike(baseTemplate.getAdvancedImageInside())
            .withInitScript("sudo add-apt-repository ppa:openjdk-r/ppa -y \n" +
                            "sudo apt-get -y update \n" +
                            "sudo apt-get install openjdk-8-jre openjdk-8-jre-headless openjdk-8-jdk -y")
        .endAdvancedImage()
    .endTemplate()
    .build();

Jenkins.getInstance().clouds.add(myCloud);
```

This sample only contains a few arguments of builder, please find all the arguments in folder builders.

Changelog

Version 1.2.2, 2019-09-18

- Revert images to previous API version

Version 1.2.1, 2019-09-12

- Fix invalid template error for custom image

Version 1.2.0, 2019-09-04

- Enable adding custom tags for created resources
- Support purchase plan for custom image
- Fix possible NullPointer exception for template
- Bump arm template version

Version 1.1.1, 2019-07-02

- Fix UAMI for reference image id template

Version 1.1.0, 2019-06-18

- Add more location options like UK
- Add support for user assigned managed identity

Version 1.0.1, 2019-05-22

- `Fix deserialize losing configuration issue`

Version 1.0.0, 2019-05-22

- Bump Jenkins baseline to 2.60.3
- Add IMDS credential type
- Support configuration as code
- Support incrementalified plugin
- Add management tags for plugin created resource group and storage account
- Fix not support different templates in different regions

Version 0.9.0, 2019-03-11

- Support Image Gallery as an option for agents
- Fix configuration with error pages JENKINS-55935

Version 0.8.1, 2019-03-06

- Fix multiple security issues

Version 0.8.0, 2019-01-28

- Add Availability Set support JENKINS-40635
- Remove periodic jobs' log from Jenkins default log level JENKINS-51282
- Specify OS disk size for agents JENKINS-51528
- Fix customer image version fixed as latest issue

Version 0.7.5, 2018-11-28

- Fix storage account name already exists issue JENKINS-54885
- Use https when communicating with storage account JENKINS-52967

Version 0.7.4, 2018-10-18

- Support MSI for agents JENKINS-53945

Version 0.7.3, 2018-08-06

- Fixed Jackson version conflicts caused by Azure client runtime lib updates JENKINS-52838

Version 0.7.2.1, 2018-08-01

- Fixed an issue that built-in images may fail to provision

Version 0.7.2, 2018-07-25

- Fix missing plan information for some images in reference image configuration JENKINS-52407

Version 0.7.1, 2018-05-23

- Update Provision retry strategy in order to prevent hitting Azure request rate limit
- Update validate logic to avoid bad template by accidentally issue
- Change the way of generating name of VMs to resolve the same name conflict
- Remove sleep while shutting down VMs in OnceRetentionStrategy

Version 0.7.0, 2018-04-09

- Support custom managed disk JENKINS-48076
- Add more checks in Template Name, Admin/Password JENKINS-49150
- Add sync logic to prevent issues when using only shutdown JENKINS-49021
- Clean unused warning logs JENKINS-48901
- Add Cloud Statistics cleaner JENKINS-48345
- Add blank choice for storage account to reduce UI issues related to existing storage account JENKINS-48253
- Check whether reuse existing storage account JENKINS-47923
- Show detail error message when the template cannot be deployed. JENKINS-46337
- Resolve vm delete issue if related cloud have been deleted. JENKINS-49473

Version 0.6.2, 2018-02-24

- Use Tls1.2 to fix the bug in SSH initialization
- Support Windows Server 1709 as SSH slave

Version 0.6.1, 2018-02-09

- Fix location verification on non-global clouds

Version 0.6.0, 2018-01-02

- Use deallocation instead of powerOff when configured shutdown only
- Add built-in windows image with docker
- Redesign verification task to improve performance
- Fix launching issues when using performance limited vms

Version 0.5.0, 2017-11-29

- Add support for the Managed Service Identity (MSI) as credential
- Clean init scripts after the deployment
- Fix some minor bugs

Version 0.4.8, 2017-11-07

- Add Cloud and Template builder with fluent interface
- Add pool retention strategy and once retention strategy
- Fix bugs and improve performance
- Maven version in built-in image update to 3.5.2
- Add Third Party Notice

Version 0.4.7.1, 2017-08-10

- Fixed an issue that built-in init script cannot run correctly under Linux.

Version 0.4.7, 2017-08-01

- Built-in image support. Now besides manually fill in image reference and init script, you can select from two built-in images, Windows Server 2016 and Ubuntu 16.04 LTS.
- Auto tool installation on VM agents. If you're using built-in image, you can choose to install Git, Maven or Docker.
- SSH support for Windows agent. You can now use SSH to launch Windows agent.
- Support managed disk for VM agent.
- Allow user to specify the name for the cloud. This fixes the issue that one cloud will be ignored if you have two with same subscription ID and resource group name.
- Various minor bug fixes.

Version 0.4.6, 2017-06-20

- Add LogRecorder for Azure VM Agent plugin to make it easier for troubleshooting
- Fix an issue that Jenkins crashes in some cases when CloudStatistics is enabled
- Improve VM template creation, you're now able to select from existing resource groups and storage accounts.

Version 0.4.5.1 Beta, 2017-06-09

- Fixed a backward compatibility issue that storage type becomes empty when reading a configuration created from an older version. (JENKINS-44750)

Version 0.4.5 Beta, 2017-06-02

- Added the option to specify different resource group for virtual network. (JENKINS-43909)
- Support multiple cloud profiles with the same subscription ID (JENKINS-43704)
- Support premium storage account for VM template (JENKINS-43097)
- Support Cloud Statistics Plugin (JENKINS-42799)

Version 0.4.4 Beta, 2017-04-12

- Added the option to deploy VM Agents without a public IP. (JENKINS-40620)
- Added the ability to attach a public IP for an already deployed agent. The user need to go to the Nodes management page and configure the desired Azure VM Agent.
- Added the option to attach an existing Azure Network Security Group to the provisioned agents
- Fixed the 'Usage' parameter. (JENKINS-42037)
- Fixed a Null Pointer Exception while trying to fill the VM Size dropdown. (JENKINS-42853)
- The agent password verification now allows more special characters. (JENKINS-43243)

Version 0.4.3 Beta, 2017-03-03

- The plugin now depends on the Azure Credentials plugin

- Existing credentials are now working when upgrading the plugin (JENKINS-42479)
- Fixed an issue where deployments in some existing Azure Resource Groups might have failed
- The plugin now removes the leftover empty containers after a custom-image agent was deleted.

Version 0.4.2 Beta, 2017-01-16

- Extend support to all Azure regions and available VM sizes (JENKINS-40488)
- Fixed an edge case where provisioned VMs were not removed after they were used (JENKINS-41330)
- The fix will ensure that newly created resources are properly disposed, but it won't disposed of any resources deployed with the plugin before and including version 0.4.1
- Updated the 'Max Jenkins Agents Limit' label in the cloud configuration page to reflect the quota on the number of agents the plugin is allowed to deploy in a resource group (JENKINS-41568)
- Other minor fixes and improvements

Version 0.4.1 Beta, 2016-12-13

- Verify that the storage account name is valid during template verification (JENKINS-40289)
- The auto-generated storage account name is unique across Azure (JENKINS-40288)
- The SSH session is re-established after the init script runs (JENKINS-40291)
- Other minor fixes and improvements

Version 0.4.0 Beta, 2016-12-06

- Initial release