README.md

VMware Cloud on AWS Packer examples

This repository contains examples to help you get started with automating the creation of virtual machine (VM) templates in a VMware Cloud on AWS software-defined datacenter (SDDC) (or vSphere cluster) with HashiCorp Packer. Each example leverages the `vsphere-iso` builder and includes the high performance vmxnet3 network adapter and NVMe controller. The VMware Paravirtual SCSI controller was tested with these too though.

Of note, the prerequisites and default variable values in the example definition files are oriented to a VMware Cloud on AWS software-defined datacenter (SDDC), but these examples should also be usable in most VMware vSphere environments with little to no modifications required.

Considerations

- The example definition files provide the minimum necessary configuration for demonstration purposes. These VM templates are not hardened or otherwise intended

for production purposes as-is. Building production-grade VM templates is possible, but out of scope for this project.

- Since these are examples, the host-based firewall is disabled and unconfigured. Additionally, since the intended use case is VMware Cloud on AWS, the expectation is that the NSX-T gateway and distributed firewalls would be used instead.
- A timestamp is appended to the VM template name so that you know exactly when it was built, and to prevent name collisions for subsequent builds.
- The AWS CLI is installed to provide an example of installing a package during the `provisioners` phase, but it's not necessary.

Considerations for the `ubuntu-server` VM template

- As of 2020-09-04, Canonical's new automated Ubuntu server installation system that leverages `cloud-init` configuration, Subiquity, is not interoperable with VMware's guest customization feature. VMware has an existing open source project for providing some interoperability with `cloud-init`. If guest customization is a requirement for your environment, use the `ubuntu-server-legacy` template instead, which leverages the legacy `debian-installer` preseeding system.

Considerations for the `windows-server` VM template

- Sysprep (generalize) is not run at the end of the build because the expectation is that the security identity (SID) will be reset via the guest customization specification created in the prerequisites below.
- Chocolatey is installed for programmatically installing software packages, but its not necessary.
- The OpenSSH Server feature is installed as a remote management option for your VMs, but this isn't necessary either.

Prerequisites

Packer

- Packer v1.6.3 or greater if building with the NVMe storage controllers
  - Each Packer template was tested with Packer v1.6.2 when building with the VMware Paravirtual SCSI controller.

VMware vSphere environment

- A VMware Cloud on AWS software-defined datacenter (SDDC) (or a vSphere cluster)

- A network segment (or port group) with DHCP and internet connectivity
  - Note: If specific destinations and ports are needed for building outbound firewall policy, please refer to the definition files as these may change over time, and the definition files will always be authoritative.
- Packer installed in a location with the following connectivity:
  - HTTPS (443/tcp) connectivity to vCenter
  - SSH (22/tcp) connectivity to the target network segment listed above for communicating with the VM during the `provisioners` phase
  - WinRM-HTTPS (5986/tcp) connectivity to the target network segment listed above for communicating with Windows VMs during the `provisioners` phase
- Sufficient storage capacity for storing the VM guest operating sytem installation ISO image files, as well as the VM templates' virtual hard disks and other files in your vSAN WorkloadDatastore (or a writeable datastore)
  - Note: As of 2020-08-24, the `vsphere-iso` builder supports content libraries as a source location for ISO files. This feature isn't well-documented yet, but was released as part of v1.6.2.
- vCenter credentials with cloudadmin (or administrative) rights
  - Custom fine-grained permissions are possible, but beyond the scope of this project

Prerequisites for Linux VM templates

- Create a Linux guest customization specification.

Prerequisites for Windows VM templates

- Create a Windows guest customization specification that generates a new security identity (SID).
- On a Windows server or client where you have administrative rights...
  - Download the latest Windows Assessment and Deployment Kit (Windows ADK).
    - Note: Only the `Deployment tools` feature that includes the Windows System Image Manager (SIM) is necessary.

Getting started

Build preparation

- Download the ISO image files for the VM template(s) that you want to build:
  - Examples:

    - `ubuntu-server`
        - Ubuntu Server 20.04.1 LTS
    - `ubuntu-server-legacy`
        - Ubuntu Server 20.04.1 LTS (legacy Debian installer)
    - `windows-server`
        - Windows Server 2019
        - VMware Tools 11.1.0 (or whichever version is appropriate for your vSphere cluster)

- Upload the VM guest operating system installation ISO files to a directory/folder named `ISO` in the target datastore

- Create one or more `.pkrvars.hcl` variable definition files for defining values for variables that you want to persist between builds

- Example variable definition file for building an Ubuntu Server 18.04 LTS VM template with the `ubuntu-server-legacy.pkr.hcl` definition file:

```
# ./ubuntu-server-18-legacy.pkrvars.hcl

# http://cdimage.ubuntu.com/releases/18.04/release/ubuntu-18.04.5-server-
iso_filename = "ubuntu-18.04.5-server-amd64.iso"
vm_name = "template-ubuntu-server-18.04-amd64-legacy"
```

Preparing to build the `ubuntu-server` VM template

```
.
├── http/
│   ├── scripts/
│   │   └── linux/
│   │       └── awscli.sh
│   └── ubuntu-server/
│       ├── meta-data
│       └── user-data
└── ubuntu-server.pkr.hcl
```

- Note: The `./http/ubuntu-server/user-data` and `./http/ubuntu-server/meta-data` are the `cloud-init` configuration files that are used to provide all of the input necessary to build the VM template without manual intervention, and `./http/ubuntu-server/meta-data` file is supposed to be empty.

- Create a password hash with mkpasswd
  - Example:

```
$ mkpasswd --method=SHA-512 --rounds=4096
Password:
[password hash]
```

- In the `./http/ubuntu-server/user-data` file, set the password for the `ubuntu` user account:

```
# ./http/ubuntu-server/user-data

autoinstall:
  identity:
    password: [password hash]
```

Preparing to build the `ubuntu-server-legacy` VM template

```
.
├── http/
│   ├── scripts/
│   │   └── linux/
│   │       └── awscli.sh
│   └── ubuntu-server-legacy/
│       └── ubuntu-server-legacy.seed
└── ubuntu-server-legacy.pkr.hcl
```

- Note: The `./http/ubuntu-server-legacy/ubuntu-server-legacy.seed` file is the `debian-installer` preseed configuration file that is used to provide all of the input necessary to build the VM template without manual intervention.
- Create a password hash with mkpasswd
  - Example:

```
$ mkpasswd --method=SHA-512 --rounds=4096
Password:
[password hash]
```

- In the `./http/ubuntu-server-legacy/ubuntu-server-legacy.seed` file, set the password for the `ubuntu` user account:

```
# ./http/ubuntu-server-legacy/ubuntu-server-legacy.seed

d-i passwd/user-password-crypted password [password hash]
```

Preparing to build the `windows-server` VM template

```
.
├── http/
│   ├── scripts/
│   │   └── windows/
│   │       ├── Initialize-WinRM.ps1
│   │       ├── Install-AWSCLI.ps1
│   │       ├── Install-Chocolatey.ps1
│   │       ├── Install-OpenSSHServer.ps1
│   │       └── Reset-AutoLogonCount.ps1
│   └── windows-server/
│       └── 2019/
│           ├── autounattend.xml
│           └── install_Windows Server 2019 SERVERDATACENTER.clg
└── windows-server.pkr.hcl
```

- Note: The example answer file (`./http/scripts/windows-server/autounattend.xml`) and catalog file (`./http/scripts/windows-server/install_Windows Server 2019 SERVERDATACENTER.clg`) are configured for a silent install of Windows Server 2019 Datacenter Evaluation Edition that also installs VMware Tools and temporarily enables WinRM (for the Packer provisioner phase). When you want to programmatically build other types of Windows VMs, please checkout the Unattended Windows Setup Reference.
- In Window System Image Manager, go to `File > Open Select Windows Image...` and open `./http/windows-server/2019/install_Windows Server 2019 SERVERDATACENTER.clg`
- Then go to `File > Open Answer File...` and open the example answer file: `./http/windows-server/2019/autounattend.xml`
- In the Windows Image pane, expand `amd64_Microsoft-Windows-Shell-Setup_10.0.17763.1_neutral > UserAccounts`, then right-click `AdministratorPassword` and select `Add Setting to Pass 7 oobeSystem`, and then set the password

    - `Value` = [desired password]
      - Note: By default, the passwords will be masked when saved via a hash.
- In the Windows Image pane, expand `amd64_Microsoft-Windows-Shell-Setup_10.0.17763.1_neutral > AutoLogon`, then right-click `Password` and select `Add Setting to Pass 7 oobeSystem`, and then set the password
    - `Value` = [desired password]
- Then go to `File > Save Answer File` to save the changes

Build your VM template

- Run `packer build` with the appropriate parameters
    - Example:

```
packer build -var-file='./sddc.pkrvars.hcl' -var-file='./ubuntu.a
```

- A few minutes later, you'll have a fresh, new VM template.
- Voila! You're done.

Troubleshooting

- Please see Debugging Packer Builds.

Next steps

- Deploy a VM from your new VM template(s) and apply your guest customization specification that you created in the prerequisites (except when building with the `ubuntu-server` definition files due to the guest customization issue mentioned above)
- Customize the definition files and build new VM templates.
- Start building all of your VM templates programmatically!

Reference

- `ubuntu-server`: `cloud-init` documentation
- `ubuntu-server-legacy`: `debian-installer` preseed documentation
- `windows-server`: Unattended Windows Setup

Security

See CONTRIBUTING for more information.

License

This library is licensed under the MIT-0 License. See the LICENSE file.

Releases 6

v0.1.5 Latest
on Dec 1, 2021

+ 5 releases

Contributors 2

Languages

HCL 76.7% PowerShell 22.1% Shell 1.2%