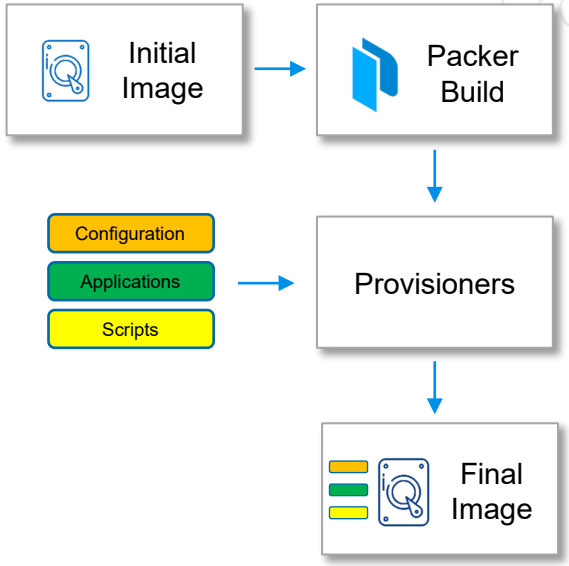
Provisioners

Introduction to Provisioners

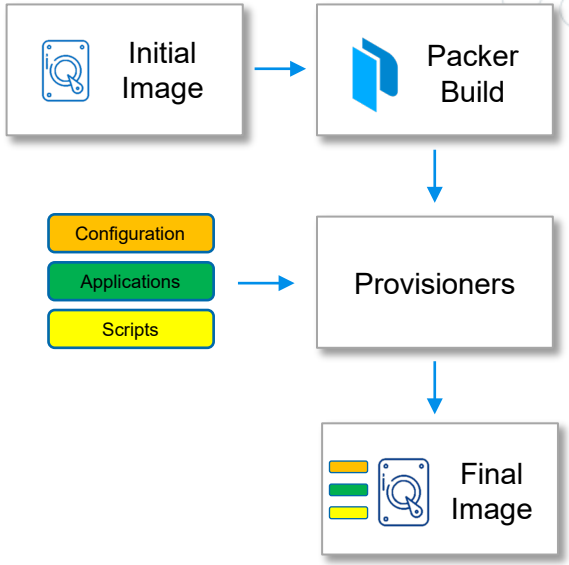
- Provisioners use built-in and third-party integrations to install packages and configure the machine image
- Built-in integrations include file and different shell options
- Third-party integrations include:
  - Ansible – run playbooks
  - Chef – run cookbooks
  - InSpec – run InSpec profiles
  - PowerShell – execute PowerShell scripts
  - Puppet – run Puppet manifest
  - Salt – configure based on Salt state
  - Windows Shell – runs commands using Windows cmd



Introduction to Provisioners

Provisioners prepare the system for use, so common use cases for provisioners include:

- installing packages
- patching the kernel
- creating users
- downloading application code

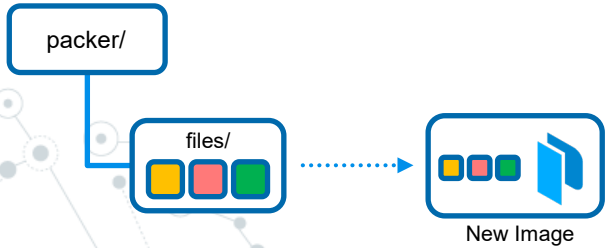


Examples of Provisioners

Upload a Single File

```
provisioner "file" {
  source = "packer.zip"
  destination = "/tmp/packer.zip"
}
```

Upload Multiple Files in a Local Directory




```
provisioner "file" {
  source      = "/files"
  destination = "/tmp"
}
```

Examples of Provisioners

Execute a Script

```
provisioner "shell" {
  script = "install_something.sh"
}
```

Run Commands on the Machine

```
provisioner "shell" {
  inline = [
    "echo Installing Updates",
    "sudo apt-get update",
    "sudo apt-get install -y nginx"
  ]
}
```

Examples of Provisioners

Run an Ansible Playbook

```
provisioner "ansible" {
  ansible_env_vars = ["ANSIBLE_HOST_KEY_CHECKING=False"]
  extra_arguments  = ["--extra-vars", "desktop=false"]
  playbook_file    = "${path.root}/playbooks/playbook.yml"
  user             = var.ssh_username
}
```

Execute a PowerShell script

```
provisioner "powershell" {
  script = [".scripts/win2019.ps1"]
}
```

Useful Features of Provisioners

Run on Specific Builds

- Use `only` and `except` to run provisioner with specific builds
- This was shown in Target Cloud and Build Types demo (builders section)

Build-Specific Overrides

- Can override builds if they behave differently in order to achieve similar results
- Could be used if building images across different platforms so you end up with a like-for-like images
  - Example: Admin access or agent installs

```
provisioner "shell-local" {
  inline = ["./tmp/install_vmware-tools.sh"]
  override = {
    aws = {
      inline = ["./tmp/install_cloudwatch_agent.sh"]
    }
  }
}
```

Useful Features of Provisioners

Using the On Error Provisioner

- Special provisioner that only runs if the related-provisioner fails
- Runs before the instance is shutdown/terminated
- Write data to a file, unsubscribe from a service, clean up custom work

```
build {
  sources = ["aws-amazonlinux-build"]
  provisioner "shell-local" {
    inline = ["sudo yum update -y"]
  }
  error-cleanup-provisioner "shell-local" {
    inline = ["echo 'update provisioner failed'> packer_log.txt"]
  }
}
```

Only runs if other provisioner fails

Useful Features of Provisioners

Introducing a Pause

- Introduce a waiting period before running
- Helpful when it takes a bit for the OS to come up, or other processes are running that could conflict with provisioner
- Uses the `pause_before` value. By defaut, there is no pause.

```
build {
  sources = ["aws-ubuntu-build"]
  provisioner "shell-local" {
    inline = ["sudo yum update -y"]
    pause_before = "10s"
  }
}
```

Wait 10 seconds before executing

Useful Features of Provisioners

Retries

- Instruct a provisioner to try again if it fails
- Helpful if a provisioner depends on external data/processes to complete successfully
- Uses the `max_retries` value. By default, this value is `0` and there is no retry

```
build {
  sources = ["aws-ubuntu-build"]
  provisioner "shell-local" {
    inline = ["sudo yum update -y"]
    max_retries = 5
  }
}
```

Try again (up to 5 times)

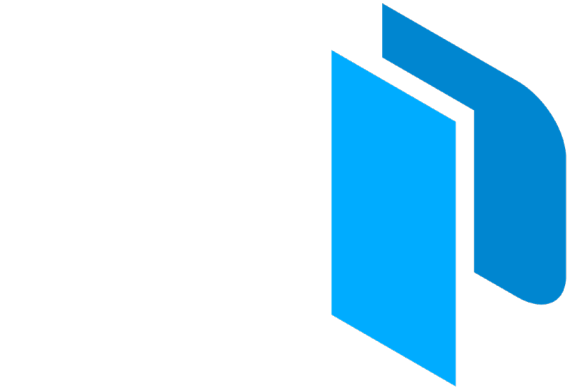
Useful Features of Provisioners

Timeouts

- Retrict the time the provisioner should wait until you consider it failed
- Uses the `timeout` value. By default, there is no timeout

```
build {
  sources = ["aws-ubuntu-build"]
  provisioner "shell-local" {
    inline = ["./install_something.sh"]
    timeout = "5m"
  }
}
```

Wait 5 minutes before marking it as failed



END OF SECTION