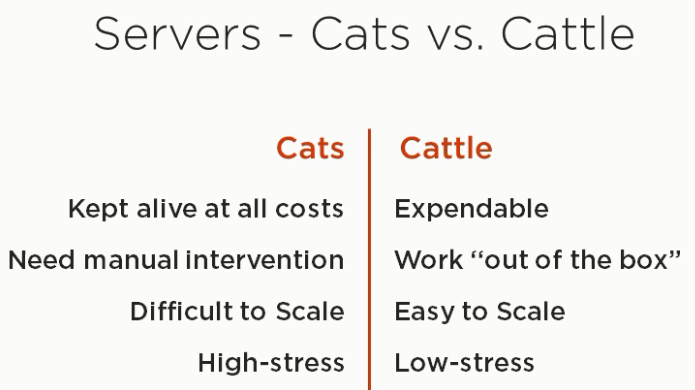
**Immutable infrastructure**

Docker is a great way to implement immutable infrastructure. But it requires a more significant investment in automation and process change than other options, which not all companies can manage.

Packer to Docker makes a good transition

By the end of this course we will know how to use Packer to create an **immutable image** with a basic operating system hosted on Amazon Web Services.

Immutable – unchanging over the time, or unable to be changed



Other benefits

* Testable Infrastructure

We can codify the infrastructure and create the automated test cases for the infrastructure

* Reproduce Production

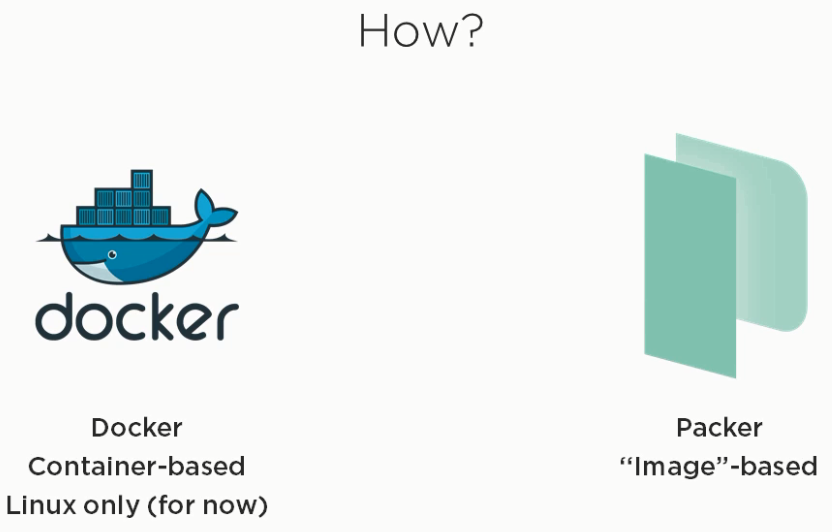
We can create an image that we know is running unchanged in the production and we can recreate the exact same environment anywhere we are from the testing environment to even on developers machine

* Unit of deployment

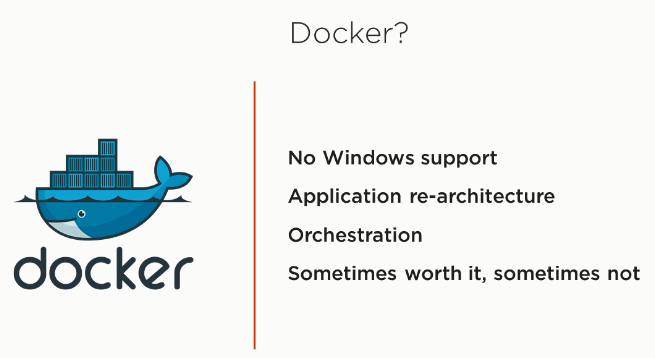
Infrastructure becomes a unit of deployment and now developer’s code is no longer separate from the server hosting it. They are deployed together as a one single package.

* Confident changes

If anything were to wrong you can easily switch to the older version



Why not Docker yet?



Why Packer instead?



A packer template

It is a simple JSON file which can be edited with any editor and should be stored in some version control repository.

There are 3 main parts to any packer template

1. Builders
2. Provisoners
3. Post-processors

**Builders**

* Generate the image
* Provider specific
* Multiple Builders= cross-provider builds!

For instance if you are working with AWS then builder may suggest you with which ami (amazon machine image) to start with, region to create that ami in and how to authenticate AWS

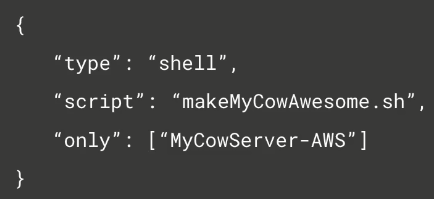


**Provisoners**

Customize the image

Scripts or configuration management

Can be builder specific



**Post processors**

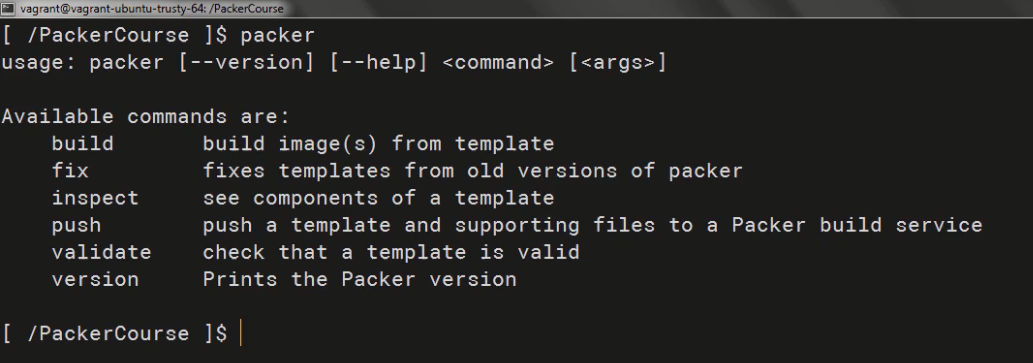
Put on finishing touches

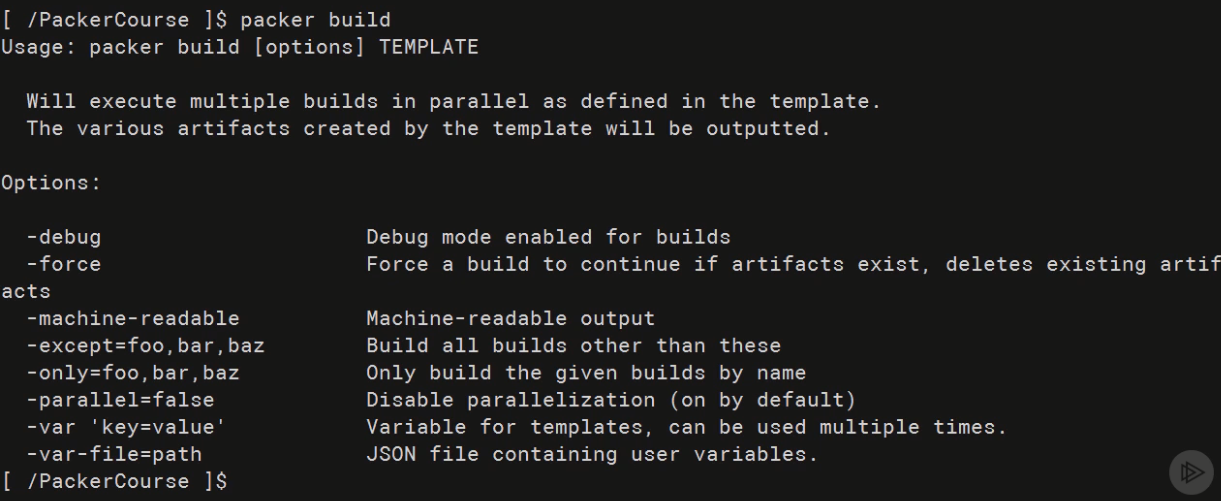
Integration with other services

Key to Docker conversion

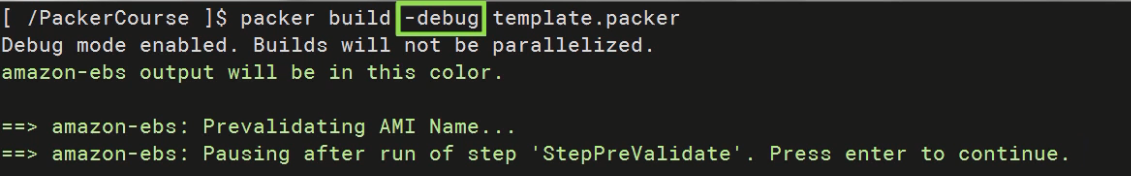
**Packer installation and commands**

Download the packer > store it in a directory in C:\ > include the path in path variable





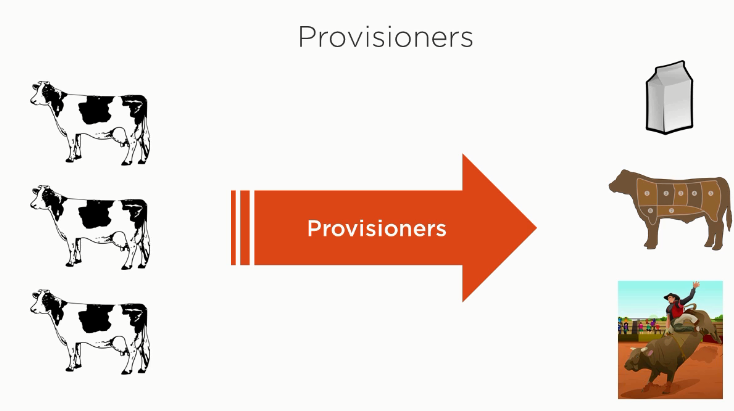
**Packer -debug** option we can use it to our advantage of having confirmation after the each step.



After the (windows+c) key is pressed, the packer generally cleans up the any AMI, extra volumes, deletes the temporary security group, deletes the temporary key pair

**Creating base images with Packer**

**Provisioners**



**Provisioner types**

Configuration management

File

Script



Hands on Ansible – course

The problems in the middle tier

