Packer is used to create images.

Terraform is used to create working machines.

Builders- tells what is the environment where we want to work on

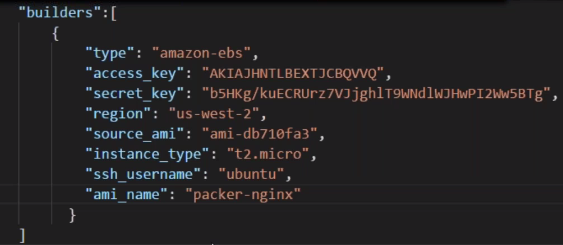
Like aws,azure,virtualbox,etc

The below packer file sample created by referring samples from packer.io and it is better refer to the docs in this site whenever we are getting started.

<https://packer.io/docs/builders/amazon-ebs.html>

builders is a list in json, so opens and closes with [ and ].

**nginx.json (or) nginx.packer:**

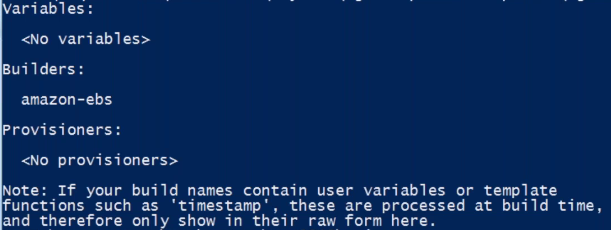


packer --version

packer validate .\nginx.json



packer inspect .\nginx.json

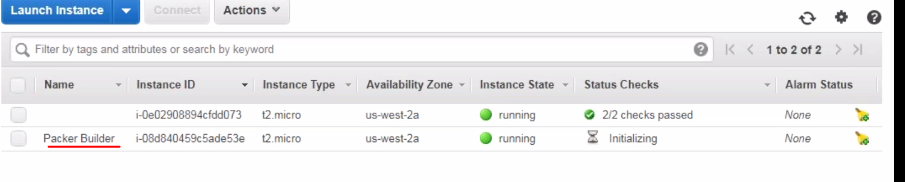


packer build --help

if we use -debug, it will ask for confirmation for every step.

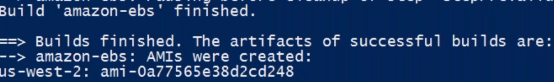
packer build -debug .\nginx.json

Because of above command, it will create a machine first.

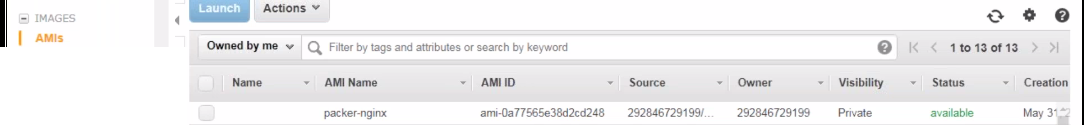


Then, using this machine, it will create an image.

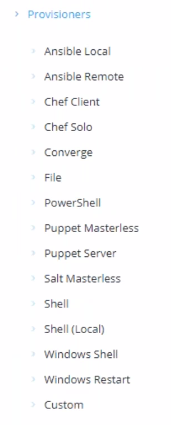
And it will show ami id of image at the end of output on console.



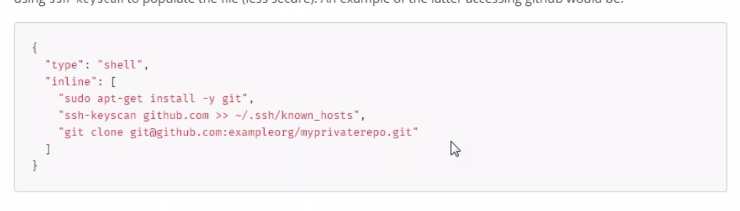
In the builder section, we specify the cloud type or the environment type.But, what to get installed or configured on the machine, we use provisioners.



Different types of provisioners available for packer.

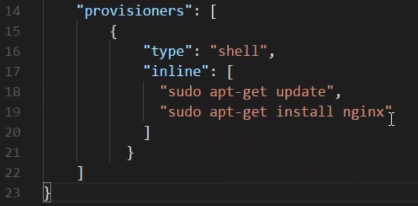


Click on above particular link of provisioner type we want to work with.Let’s say a basic shell script, so click on Shell.



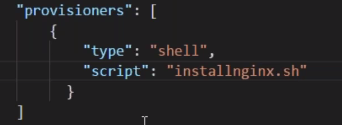
Add a new section next to the Builders section in the above packer file.

provisioners is also a list.And this list is of complex objects, so in { }

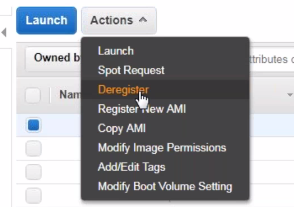


inline, script written inside packer file.

path points to an external shell script file.



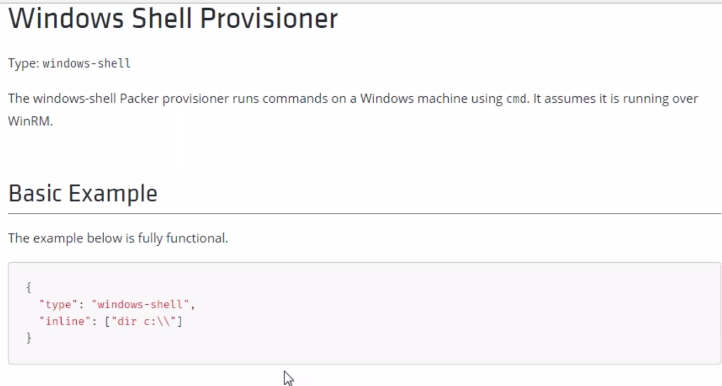
To delete an image from aws console 🡪 images 🡪 Amis. Select image and click on action and select deregister.



packer build .\nginx.json

The script will fail above as -y not given after shell commands, so it waits for yes/no and waits for sometime and fails as we can’t provide user inputs.

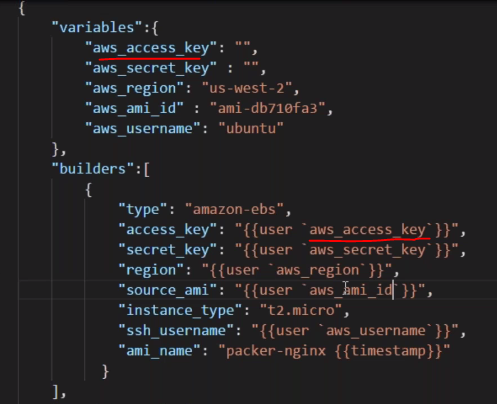
For windows, below sample.

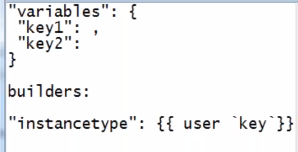


**Variables**

Can be created under variables section.

As json, we can keep variables before or after any section as it can pick from anywhere.





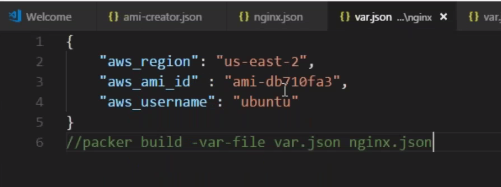
For environement variables,



We can pass the values like password or access key secret empty in the file and pass as command line arguments.



We can keep variables in a separate json file and use with the packer main file by passing the var file path.But should keep the same variables in the main file variables section as empty values so that can be used as declarations or default values.



When we pass a value both with var.json file and also as argument then preference will be given to argument.