**AWS EC2**- Virtual machine in AWS is called EC2.

EC2 uses key based authentication mechanism. They share private key and keep public key with them. One key handshake happens successfully, then we will get active session.

[ec2-user@ip-172-31-47-123 ~]$ ssh-keygen Generating public/private rsa key pair.

Now, go to home directory, and there will be a hidden folder by name .ssh

[ec2-user@ip-172-31-47-123 .ssh]$ cat id\_rsa

-----BEGIN OPENSSH PRIVATE KEY-----



-----END OPENSSH PRIVATE KEY-----

[ec2-user@ip-172-31-47-123 .ssh]$ cat id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDIyp5k4WhapMczUlCvbEpF/5aW+rYU17MYMRBhtXnatuWRrrFA9bN7gK59XMnVbRYFmFnHiyAsfY7pX5M9jg8cjfLeQJdAu4tq9eUSDU3NVkv9mhigsxWnRBSNzfk+VdqTgEIq7ixwXfZ7vzipYK3OJi2nr0uZ1MorvJTM9OUH9ciIPcp06DtiloBw0MteCMpW4Xt6PBE/sUzghJsc86Q7oixQaD92UFUy1L65e7tm1zxbXvwYMJ1vso++bGVMXFTInCkdFFR1L5pLc9Pinr3wxMrV4Mjr9Fj4XYqJr4eoLgsuz9SKANs8B+M9kMH2gphckIjfaGt56LAJly6tsvQ3 ec2-user@ip-172-31-47-123.us-east-2.compute.internal

[ec2-user@ip-172-31-47-123 .ssh]$ cat authorized\_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCa5jWtj0JZGnnzz5zZcRIEOqmOaJaW2Zv63mxCGHniOP71lKJi1GesCnv9+QfNjLDOS4DucgKkE8j1JPkvIxz5WtTq99+eI03nnixNLxOWIYdLpVSgSJnudLfHzKVM+3v15kz0MrcfX9Ki/kybbAUPtHlMB6xTfv4gIMxRuxapeMqnnvb6fc8hXem6SNLekcoReb7qjHMXdhS/XaTSG1WrPYFCiJ/43KjBTLxGbp+VN2XVXkYSd8bJ6sWzV02pSQYScx5VDmxjfwsToIQawJZGyXKRh91hdZC40v1Idw8B0RqMqiNCkqYkuxI5vaqC3qGXVsD+7R+vcIYUc9fndk4B GIT

[ec2-user@ip-172-31-47-123 .ssh]$

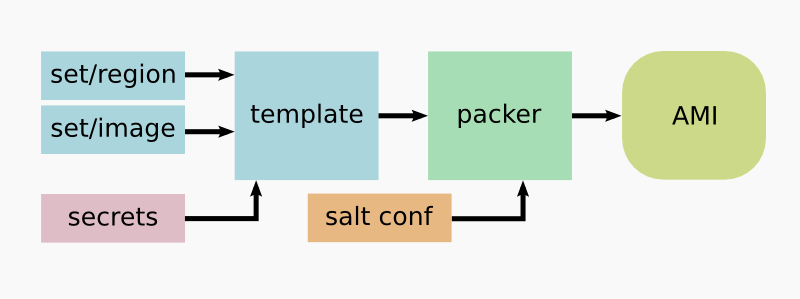
**Why Amazon uses key based authentication? (AWS hold public key, and end user hold private)** Answer: It considered as safer option for communication.

Note- Amazon uses reusable images for creating the virtual box.



**My requirement here**: I want to create reusable image

Case study- There is a tool that has business intelligent in it and depending on user input it create reusable image.

Example- Packer.

Note- For creating image, we provide JSON file as input to Packer.

Packer has 2 important components-

1. Builder
2. Provisioning.
3. Postprocessing

**Builders**

Builders are responsible for creating machines and generating images from them for various platforms. For example, there are separate builders for EC2, VMware, VirtualBox, etc. Packer comes with many builders by default and can also be extended to add new builders.

**Provisioning:**

It’s a custom work that need to perform, like installing java on box.

**Postprocessing**

Once your work is done, you can change the format also.

LAB- Need one system with GIT, telnet and tree installed on it, and make a reusable image out of it [root@ip-172-31-27-149 ~]# yum install -y git telnet tree

Now, go to AWS console, then select your instance, and click on create image. From next time onwards use this AMI so by default you will be getting GIT, telnet and tree installed on your system.

Amazon CLI- It’s a command line tool by which we can perform all the operation(s) that we are performing through AWS console.

IP address- Whenever any machine or mobile connected by internet then they use ip address to identify the identity of system.

IPV4 and IPV6 –

IPV4 – (0-255).(0-255).(0-255).(0-255) -> (8digit0and1. 8digit0and1. 8digit0and1. 8digit0and1) so, by using ipv4 we can connect to 2thepower32 devices.

IPV6-They 128-bit address, so 2tothepower32 devices can we connected. Example- 2001:0000:3238:DFE1:0063:0000:0000:FEFB

Cloud Computing often referred to as “the cloud”, in simple terms means storing or accessing your data and programs over the internet rather than your own hard drive.

Packer installation- download from site, unzip it and check version by below command. **Command-** packer –version

Note: Docker is used to create image for container, and packer is used to create image for virtual machine.