

DVS Technologies Aws & Devops

Compiled and Scrutinized by
Mr. Shaan Shaik
(Senior DevOps Lead)

Words To The Students

Though we have taken utmost efforts to present you this book error free, but still it may contain some errors or mistakes. Students are encouraged to bring, if there are any mistakes or errors in this document to our notice. So that it may be rectified in the next edition of this document.

“Suppressing your doubts is Hindering your growth”.

We urge you to work hard and make use of the facilities we are providing to you, because there is no substitute for hard work. We wish you all the best for your future.

“The grass isn’t greener on the other side; the grass is greener where you water it.”

You and your suggestions are valuable to us; Help us to serve you better. In case of any suggestions, grievance, or complaints, please feel free to write us your suggestions, grievance and feedback on the following

Dvs.training@gmail.com

DVS Technologies Aws & Devops

Packer

Introduction:

Packer plays an important role in customizing the images as per our requirement it supports multiple

platforms like aws,vmware,docker & etc

for detailed list of information you can check it on "<https://packer.io/docs/builders/>"
<https://www.packer.io/docs/builders/amazon/ebs/>

Structure:

It is having three sections variables,builder,providers

variables --> you can declare the variables you want to use for the image

builders --> Aws,Azure,Google,hyper-v etc ..

provisioners --> Ansible,powershell,shell & etc ..

Installation:

`rm -f /usr/sbin/packer`

`sudo curl -O https://releases.hashicorp.com/packer/1.6.0/packer_1.6.0_linux_amd64.zip`

`sudo yum install -y unzip`

`sudo unzip packer_1.6.0_linux_amd64.zip -d /usr/local/bin/`

```
[root@packer ~]# rm -f /usr/sbin/packer
[root@packer ~]# sudo curl -O https://releases.hashicorp.com/packer/1.6.0/packer_1.6.0_linux_amd64.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 28.2M  100 28.2M    0     0  83.2M      0 --:--:-- --:--:-- --:--:--  83.2M
[root@packer ~]# sudo yum install -y unzip
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                | 3.7 kB  00:00:00
Package unzip-6.0-20.amzn2.x86_64 already installed and latest version
Nothing to do
[root@packer ~]# sudo unzip packer_1.6.0_linux_amd64.zip -d /usr/local/bin/
Archive: packer_1.6.0_linux_amd64.zip
  inflating: /usr/local/bin/packer
[root@packer ~]# packer -version
1.6.0
[root@packer ~]#
```

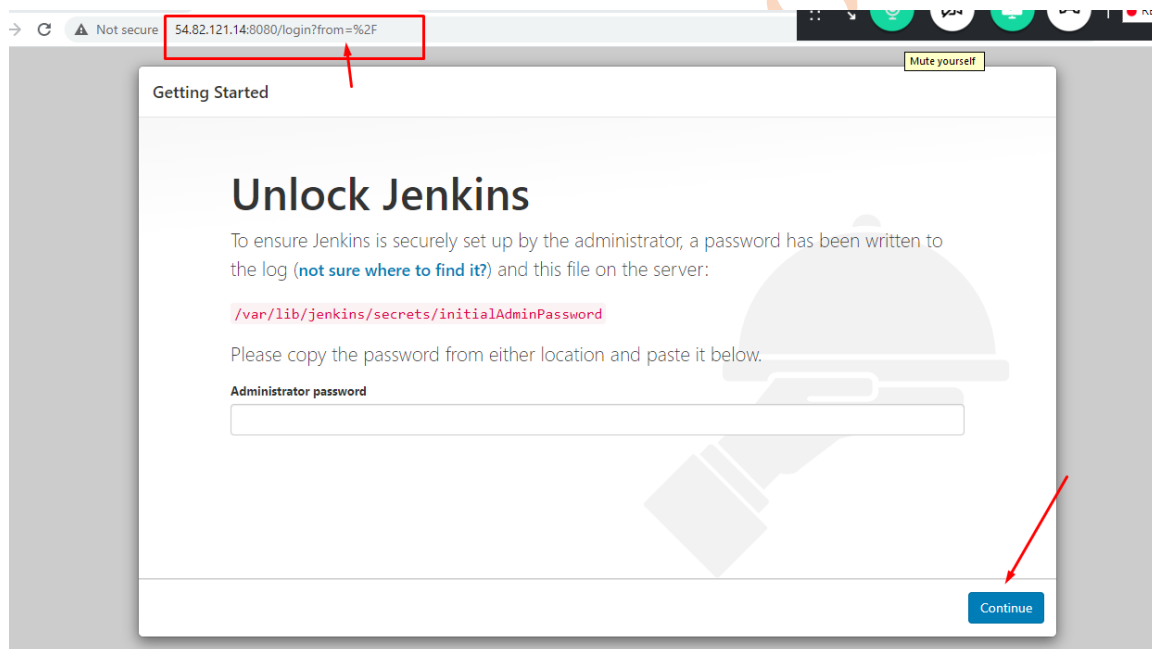
DVS Technologies Aws & Devops

Let's try to install Jenkins manually & test if our code is working or not:

Execute the below to get the latest jenkins version

```
yum install java-1.8.0-openjdk-devel -y
java -version
wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
rpm --import http://pkg.jenkins.io/redhat-stable/jenkins.io.key
yum clean all
yum install jenkins -y
yum-config-manager --disable jenkins
systemctl enable jenkins
systemctl restart jenkins
systemctl status jenkins
```

Once you are done with the above you can check the jenkins in the console



Now we can confirm that the above code is working fine now let's start creating our customized code which can give us the images on fly

DVS Technologies Aws & Devops

Working with Packer:

Configuring Aws Cli:

```
[root@packer myjenkins]# aws configure
AWS Access Key ID [None]: AKIA5GXPURK7[REDACTED]
AWS Secret Access Key [None]: EVmPiSusEQe/kujPvne2Y2ee[REDACTED]
Default region name [None]: us-east-1
Default output format [None]: json
[root@packer myjenkins]#
```

```
[root@packer myjenkins]# export AWS_DEFAULT_REGION="us-east-1"
[root@packer myjenkins]# echo $AWS_DEFAULT_REGION
us-east-1
[root@packer myjenkins]#
```

Packer Code:

```
[root@packer myjenkins]# vi setup.sh
[root@packer myjenkins]# cat setup.sh
yum install java-1.8.0-openjdk-devel -y
java -version
wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
rpm --import http://pkg.jenkins.io/redhat-stable/jenkins.io.key
yum clean all
yum install jenkins -y
yum-config-manager --disable jenkins
systemctl enable jenkins
systemctl restart jenkins
systemctl status jenkins
[root@packer myjenkins]#
```

Code:

DVS Technologies Aws & Devops

```
[root@packer myjenkins]# cat ami.json
{
  "variables": {
    "region": "",
    "ami_final_image_name": "",
    "base_image": ""
  },
  "builders": [
    {
      "type": "amazon-ebs",
      "profile": "default",
      "region": "{{ user `region` }}",
      "source_ami": "{{ user `base_image` }}",
      "instance_type": "t2.micro",
      "ssh_username": "ec2-user",
      "ami_name": "{{ user `ami_final_image_name` }}"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "script": "./setup.sh",
      "execute_command": "sudo -E -S sh '{{ .Path }}'"
    }
  ]
}
```

Variables:

```
[root@packer myjenkins]# vi variables.json
[root@packer myjenkins]# cat variables.json
{
  "region": "us-east-1",
  "ami_final_image_name": "dvsbatch4-jenkins",
  "base_image": "ami-0947d2ba12ee1ff75"
}
[root@packer myjenkins]#
```

DVS Technologies Aws & Devops

Final Execution:

```
[root@packer myjenkins]# ls -l variables.json
-rw-r--r-- 1 root root 110 Sep 30 15:44 variables.json
[root@packer myjenkins]# packer build -var-file=variables.json ami.json
amazon-efs: output will be in this color.

==> amazon-efs: Prevalidating any provided VPC information
==> amazon-efs: Prevalidating AMI Name: dvsbatch4-jenkins
amazon-efs: Found image ID: ami-0947d2ba12ee11175
==> amazon-efs: Creating temporary keypair: packer 5f74a875-104e-bb50-cda6-ddceea93b403
==> amazon-efs: Creating temporary security group for this instance: packer 5f74a876-76c9-22bb-93f3-fe5f07ec53e1
==> amazon-efs: Authorizing access to port 22 from [0.0.0.0/0] in the temporary security groups...
==> amazon-efs: Launching a source AWS instance...
==> amazon-efs: Adding tags to source instance
amazon-efs: Adding tag: "Name": "Packer Builder"
amazon-efs: Instance ID: i-05a2509faa93ed299
==> amazon-efs: Waiting for instance (i-05a2509faa93ed299) to become ready...
==> amazon-efs: Using ssh communicator to connect: 54.210.35.155
```

```
amazon-efs: AMI: ami-06f8a99b74a91c3a6
==> amazon-efs: Waiting for AMI to become ready...
==> amazon-efs: Terminating the source AWS instance...
==> amazon-efs: Cleaning up any extra volumes...
==> amazon-efs: No volumes to clean up, skipping
==> amazon-efs: Deleting temporary security group...
==> amazon-efs: Deleting temporary keypair...
Build 'amazon-efs' finished.

==> Builds finished. The artifacts of successful builds are:
--> amazon-efs: AMIs were created:
us-east-1: ami-06f8a99b74a91c3a6
```

```
[root@packer myjenkins]#
```

The screenshot shows the AWS Management Console interface for the EC2 Image Builder service. The 'Launch' tab is active, displaying a table of AMIs. The AMI 'dvsbatch4-jenkins' is selected, and its details are shown below the table.

Name	AMI Name	AMI ID	Source
<input checked="" type="checkbox"/>	dvsbatch4-jenkins	ami-06f8a99b74a91c3a6	90781440680
<input type="checkbox"/>	DONTDELETEME	ami-0c2020fcd36ffc7f3	90781440680
<input type="checkbox"/>	myapacheimage	ami-0edb0c57c0682558c	90781440680

Image: ami-06f8a99b74a91c3a6

Details Permissions Tags

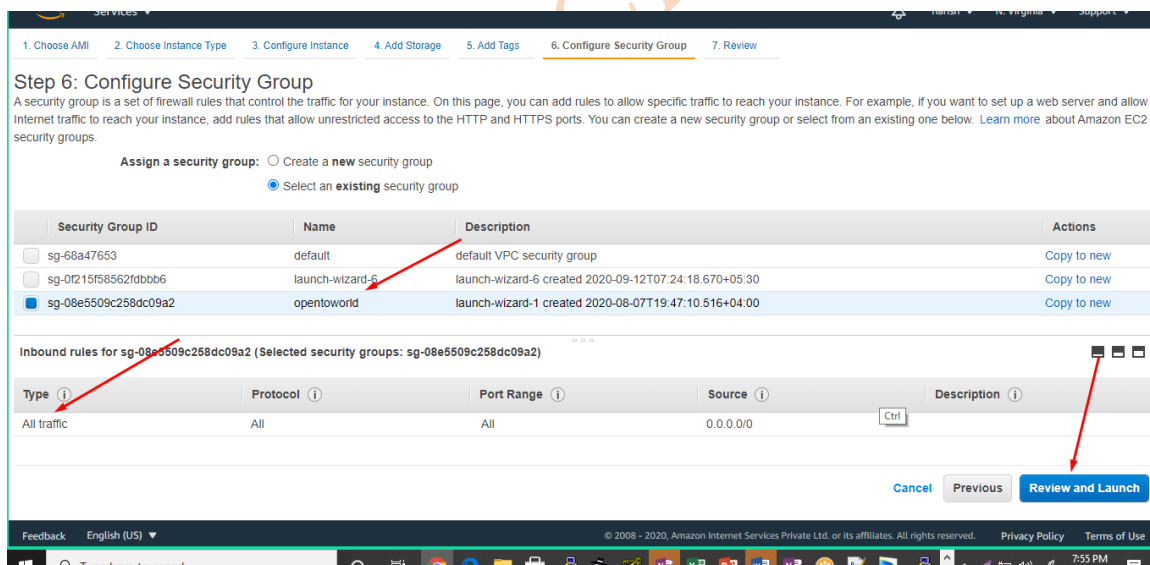
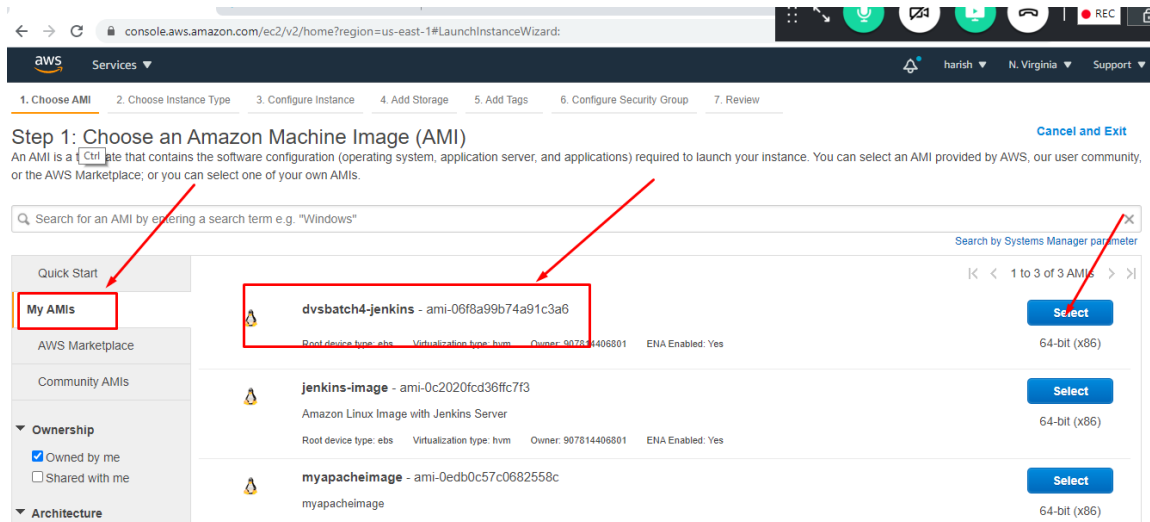
AMI ID	AMI Name
ami-06f8a99b74a91c3a6	dvsbatch4-jenkins
Owner	907814406801
Status	available
Creation date	September 30, 2020 at 7:49:05 PM UTC+4
Architecture	x86_64
Image Type	machine

Source	State Reason	Platform details	Usage operation	Virtualization type
90781440680				

DVS Technologies Aws & Devops

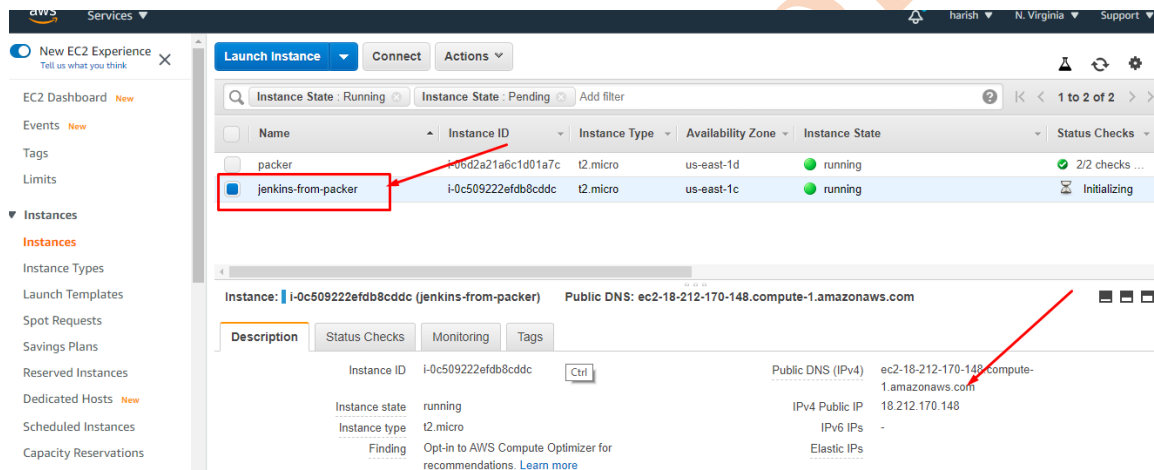
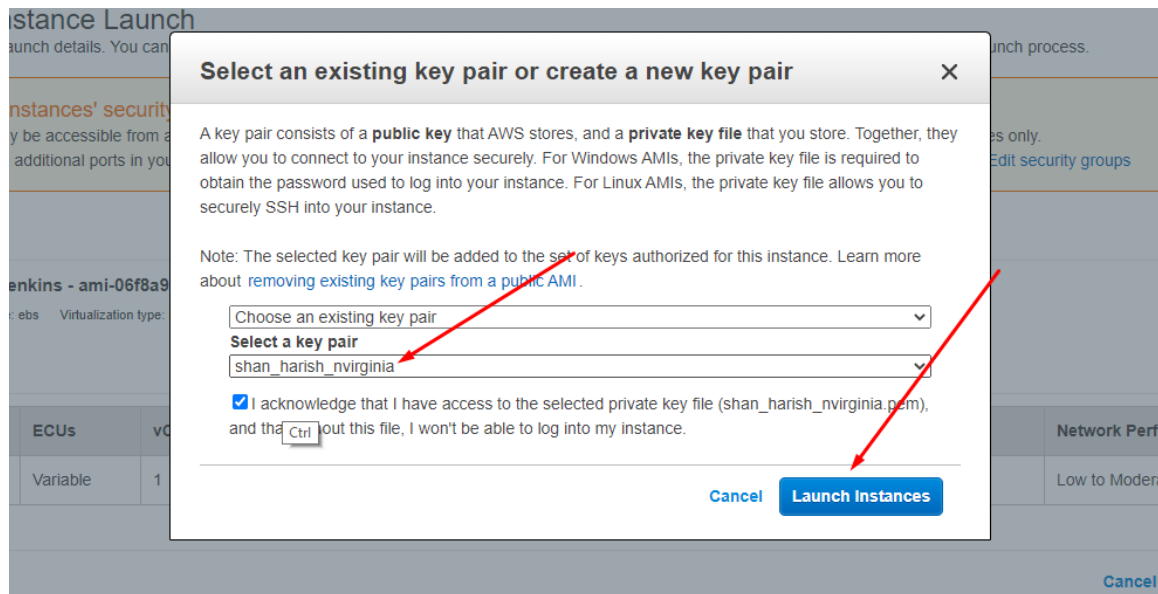
Let's test our image:

Create a server from our customized packer ami like below.

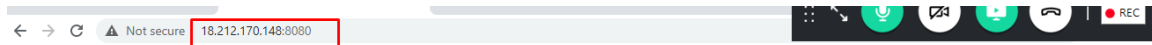


DVS Technologies, Opp Home Town, Beside Biryani Zone, Marathahalli, Bangalore Phone: 9632558585 Mobile: 8892499499 Mail : dvs.training@gmail.com Web: www.dvstechnologies.in

DVS Technologies Aws & Devops

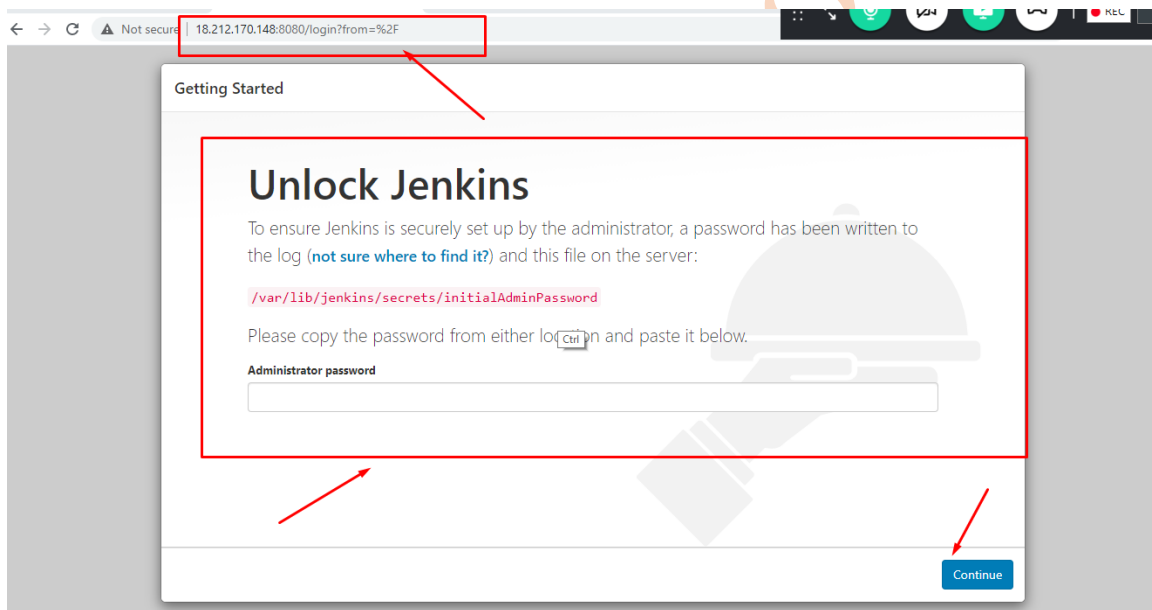


DVS Technologies Aws & Devops



Please wait while Jenkins is getting ready to work ...

Your browser will reload automatically when Jenkins is ready.



DVS Technologies Aws & Devops

Handy Code:

```
[root@master jenkins]# cat setup.sh
#!/bin/bash
yum install java-1.8.0-openjdk-devel -y
wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
rpm --import http://pkg.jenkins.io/redhat-stable/jenkins.io.key
yum install jenkins* -y
yum-config-manager --disable jenkins
systemctl enable jenkins
systemctl restart jenkins
[root@master jenkins]#
```

```
[root@master jenkins]# cat ami.json
{

"variables": {
    "region": "",
    "ami_final_image_name": "",
    "base_image": ""
},

"builders": [
    {
        "type": "amazon-ebs",
        "profile": "default",
        "region": "{{ user `region` }}",
        "source_ami": "{{ user `base_image` }}",
        "instance_type": "t2.micro",
        "ssh_username": "ec2-user",
        "ami_name": "{{ user `ami_final_image_name` }}"
    }
],
```

DVS Technologies Aws & Devops

```
"provisioners": [{  
  "type": "shell",  
  "script": "./setup.sh",  
  "execute_command": "sudo -E -S sh '{{ .Path }}'"  
}]  
}
```

[root@ ~]# cat variables.json

```
{  
  "region": "us-east-1",  
  "ami_final_image_name": "dvbatch2-jenkins",  
  "base_image": "ami-0323c3dd2da7fb37d"  
}
```

[root@ ~]# cat setup.sh

#!/bin/bash

yum install java-1.8.0-openjdk-devel -y

wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>

rpm --import <http://pkg.jenkins.io/redhat-stable/jenkins.io.key>

yum install jenkins* -y

yum-config-manager --disable jenkins

systemctl enable jenkins

systemctl restart jenkins

Final Code Execution::

packer build ami.json

packer build -var myregion="us-east-2" ami.json

packer build -var-file=myvars.json ami.json