

## **AWS - Github-Jenkins Integration Using Docker Container Images**

### **Objective:**

Integrate CI/CD tool Jenkins with Github and AWS to automate creation of Cloudformation stack (Infrastructure support tool).

### **Steps:**

1. Create Ec2 instance with CentOS
2. Install Docker engine and Jenkins container image
3. Upload code on github
4. Configure build jobs in Jenkins

Spin Ec2 instance with CentOS. CentOS version 7 will be used as a host operating system to run Docker engine. Jenkins and Docker images will be pulled. Later setup github by uploading code to create vpc as a resource. Next step is to integrate Jenkins with AWS to create cloudformation stack.

In summary, the user will create and upload code in github. Jenkins will pull the code from github and communicate with cloudformation service in AWS in order to create cloudformation stack. The cloudformation stack will create a resource in the AWS region.

Log in to AWS console:

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Home: mohandbrinda

Services Resource Groups

## Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the [EC2 Videos](#).

### Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

[Launch Instance](#)


### Migrate a Machine

Use CloudEndure Migration to simplify, expedite, and automate large-scale migrations from physical, virtual, and cloud-based infrastructure to AWS.

[Get started with CloudEndure Migration](#)

Note: Your instances will launch in the US East

Launch CentOS ec2 instance



Free tier eligible

**CentOS 7 (x86\_64) - with Updates HVM**

This is the Official CentOS 7 x86\_64 HVM image that has been built with a minimal profile, suitable for use in HVM instance types only. The image contains just enough packages to run within AWS, bring up an SSH Server and allow users to login. Please note that this is the default CentOS-7 image that we recommend everyone uses. It contains ...

[More info](#)

[View Additional Details in AWS Marketplace](#)

**Product Details**

By	Centos.org
Customer Rating	★★★★★ (61)
Latest Version	1901_01
Base Operating System	Linux/Unix, CentOS 7
Delivery Method	64-bit (x86) Amazon Machine Image (AMI)
License Agreement	<a href="#">End User License Agreement</a>
On Marketplace Since	10/5/14

**Highlights**

All official CentOS Linux images are built with SELINUX set to enforcing mode. However, we test the images with both Selinux enabled as well as permissive. Starting with CentOS-7 we now include cloud-init support in all CentOS AMI's, the

**Pricing Details**

**Hourly Fees**

Instance Type	Software	EC2	Total
t2.nano	\$0.00	\$0.006	<b>\$0.006/hr</b>
t2.micro	\$0.00	\$0.012	<b>\$0.012/hr</b>
t2.small	\$0.00	\$0.023	<b>\$0.023/hr</b>
t2.medium	\$0.00	\$0.046	<b>\$0.046/hr</b>
t2.large	\$0.00	\$0.093	<b>\$0.093/hr</b>
t2.xlarge	\$0.00	\$0.186	<b>\$0.186/hr</b>
t2.2xlarge	\$0.00	\$0.371	<b>\$0.371/hr</b>
t3a.nano	\$0.00	\$0.005	<b>\$0.005/hr</b>
t3a.micro	\$0.00	\$0.009	<b>\$0.009/hr</b>
t3a.small	\$0.00	\$0.019	<b>\$0.019/hr</b>
t3a.medium	\$0.00	\$0.038	<b>\$0.038/hr</b>
t3a.large	\$0.00	\$0.075	<b>\$0.075/hr</b>
t3a.xlarge	\$0.00	\$0.15	<b>\$0.15/hr</b>
t3a.2xlarge	\$0.00	\$0.301	<b>\$0.301/hr</b>
t3.nano	\$0.00	\$0.005	<b>\$0.005/hr</b>
t3.micro	\$0.00	\$0.01	<b>\$0.01/hr</b>
t3.small	\$0.00	\$0.021	<b>\$0.021/hr</b>

→ ↻ 🔍 <https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard> 📖 ☆ ⚙️ 📄 📄 📄

aws Services ▾ Resource Groups ▾ ⚙️

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of instances** ⓘ  [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ  [Create new VPC](#)

**Subnet** ⓘ  [Create new subnet](#)

**Auto-assign Public IP** ⓘ

**Placement group** ⓘ ☐ Add instance to placement group

**Capacity Reservation** ⓘ  [Create new Capacity Reservation](#)

**IAM role** ⓘ  [Create new IAM role](#)

**Shutdown behavior** ⓘ

**Enable termination protection** ⓘ ☐ Protect against accidental termination

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

## Configure instance storage

← → ↻ 🔍 <https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard> 📖 ☆ ⚙️ 📄 📄 📄

aws Services ▾ Resource Groups ▾ ⚙️

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/sda1	snap-07e85e7d93d5718ae	<input type="text" value="8"/>	General Purpose <input type="text" value="SSD"/>	100 / 3000	N/A	<input type="checkbox"/>	Not Encrypted <input type="text" value=""/>

[Add New Volume](#)

## Create security group

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups

mohanbrinda Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group  
☐ Select an existing security group

Security group name: CentOS 7 -x86\_64- - with Updates HVM-1901\_01-AutogenByAWSMP-

Description: This security group was generated by AWS Marketplace and is based on recomr

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP I	TCP	8080	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Create key pair

### Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

Choose an existing key pair

Select a key pair

MyJenkinsKeyPair

☐ I acknowledge that I have access to the selected private key file (MyJenkinsKeyPair.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

Launch ec2 instance

The screenshot shows the AWS Management Console interface for an EC2 instance. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar lists various AWS services and categories like 'INSTANCES', 'IMAGES', and 'ELASTIC BLOCK STORE'. The main content area displays the details of a specific EC2 instance, i-0a26a2a23dd600e36, which is a t2.micro instance in the us-east-2b availability zone, currently in a 'running' state. The instance's public DNS is ec2-18-189-7-64.us-east-2.compute.amazonaws.com. The 'Description' tab is selected, showing a table of instance details.

Property	Value
Instance ID	i-0a26a2a23dd600e36
Instance state	running
Instance type	t2.micro
Elastic IPs	-
Availability zone	us-east-2b
Security groups	CentOS 7 -x86_64- - with Updates HVM-1901_01-AutogenByAWSMP- view inbound rules. view
Public DNS (IPv4)	ec2-18-189-7-64.us-east-2.compute.amazonaws.com
IPv4 Public IP	18.189.7.64
IPv6 IPs	-
Private DNS	ip-172-31-29-81.us-east-2.compute.internal
Private IPs	172.31.29.81
Secondary private IPs	-

Convert the MyJenkinskeyPair.pem file to MyJenkinskeyPair.ppk using puttygen.  
Login to the ec2 instance using putty.

Create IAM user for communication between AWS and Jenkins. The user needs programmatic access and not management console access.



## Add user

1 2 3 4 5

### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

➕ Add another user

### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type\* ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☐ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

Cancel

Next: Permissions

Give Admin access to the user.

### Create group



Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

Create policy

↻ Refresh

Filter policies ▾ 🔍 Search Showing 472 results

	Policy name ▾	Type	Used as	Description
<input checked="" type="checkbox"/>	▶ AdministratorAccess	Job function	None	Provides full access to AWS ser
<input type="checkbox"/>	▶ AlexaForBusinessD...	AWS managed	None	Provide device setup access to
<input type="checkbox"/>	▶ AlexaForBusinessF...	AWS managed	None	Grants full access to AlexaForB
<input type="checkbox"/>	▶ AlexaForBusinessG...	AWS managed	None	Provide gateway execution acc
<input type="checkbox"/>	▶ AlexaForBusinessR...	AWS managed	None	Provide read only access to Ale
<input type="checkbox"/>	▶ AmazonAPIGatewa	AWS managed	None	Provides full access to create/e


Cancel


Create group


## Add user

1 2 3 4 5

### Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

### Add user to group

Create group Refresh

Search		Showing 1 result
Group	Attached policies	
<input checked="" type="checkbox"/> Adminaccess	AdministratorAccess	

Cancel Previous Next: Tags

Review create user and download access and secret access key.



→ ↺ 🏠 🔒 https://console.aws.amazon.com/iam/home?region=us-east-2#/users\$new?step=review&accessKey&userName

aws

Services ▾ Resource Groups ▾ ⭐

🔔 mohanbrinda ▾ Global ▾ Support

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name

jenkins-user

AWS access type

Programmatic access - with an access key

Permissions boundary

Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	<a href="#">Adminaccess</a>

Tags

No tags were added.

Cancel

Previous

Create user

→ ↺ 🏠 🔒 https://console.aws.amazon.com/iam/home?region=us-east-2#/users\$new?step=final&accessKey&userNames:

aws

Services ▾ Resource Groups ▾ ⭐

🔔 mohanbrinda ▾ Global ▾ Support ▾

Add user

1 2 3 4 5

✔ Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://436050113826.signin.aws.amazon.com/console>

📄 Download .csv

	User	Access key ID	Secret access key
▶ ✔	jenkins-user	AKIAWLBU4WUREXEGPKBI	***** <a href="#">Show</a>

Close

Loginto the instance using putty and install docker after performing yum updates

```
root@ip-172-31-29-81:/home/centos
login as: centos
Authenticating with public key "imported-openssh-key"
centos@ip-172-31-29-81 ~]$ sudo su
root@ip-172-31-29-81 centos]# yum update -y
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirror.cisp.com
 * extras: mirrors.usinternet.com
 * updates: mirror.cisp.com
base | 3.6 kB 00:00
extras | 3.4 kB 00:00
updates | 3.4 kB 00:00
1/4): extras/7/x86_64/primary_db | 215 kB 00:00
2/4): base/7/x86_64/group_gz | 166 kB 00:00
3/4): base/7/x86_64/primary_db | 6.0 MB 00:00
4/4): updates/7/x86_64/primary_db | 7.4 MB 00:01
Resolving Dependencies
--> Running transaction check
--> Package bind-libs-lite.x86_64 32:9.9.4-72.el7 will be updated
--> Package bind-libs-lite.x86_64 32:9.9.4-74.el7_6.2 will be an update
--> Package bind-license.noarch 32:9.9.4-72.el7 will be updated
--> Package bind-license.noarch 32:9.9.4-74.el7_6.2 will be an update
--> Package cloud-init.x86_64 0:18.2-1.el7.centos.1 will be updated
--> Package cloud-init.x86_64 0:18.2-1.el7.centos.2 will be an update
```

```
root@ip-172-31-29-81:/home/centos
systemd-libs.x86_64 0:219-62.el7_6.9
systemd-sysv.x86_64 0:219-62.el7_6.9
teamd.x86_64 0:1.27-6.el7_6.1
tuned.noarch 0:2.10.0-6.el7_6.4
tzdata.noarch 0:2019b-1.el7
util-linux.x86_64 0:2.23.2-59.el7_6.1
vim-minimal.x86_64 2:7.4.160-6.el7_6
xfsprogs.x86_64 0:4.5.0-19.el7_6

Complete!
[root@ip-172-31-29-81 centos]# yum install docker -y
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.cisp.com
 * extras: mirrors.usinternet.com
 * updates: mirror.cisp.com
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 2:1.13.1-102.git7f2769b.el7.centos will be installed
--> Processing Dependency: docker-common = 2:1.13.1-102.git7f2769b.el7.centos fo
r package: 2:docker-1.13.1-102.git7f2769b.el7.centos.x86_64
--> Processing Dependency: docker-client = 2:1.13.1-102.git7f2769b.el7.centos fo
r package: 2:docker-1.13.1-102.git7f2769b.el7.centos.x86_64
--> Processing Dependency: subscription-manager-rhsm-certificates for package: 2
```

Check docker status and start service

```
root@ip-172-31-29-81:/home/centos
device-mapper-event-libs.x86_64 7:1.02.149-10.el7_6.8
device-mapper-persistent-data.x86_64 0:0.7.3-3.el7
docker-client.x86_64 2:1.13.1-102.git7f2769b.el7.centos
docker-common.x86_64 2:1.13.1-102.git7f2769b.el7.centos
libaio.x86_64 0:0.3.109-13.el7
lvm2.x86_64 7:2.02.180-10.el7_6.8
lvm2-libs.x86_64 7:2.02.180-10.el7_6.8
oci-register-machine.x86_64 1:0-6.git2b44233.el7
oci-systemd-hook.x86_64 1:0.2.0-1.git05e6923.el7_6
oci-umount.x86_64 2:2.5-1.el7_6
python-pytoml.noarch 0:0.1.14-1.git7dea353.el7
subscription-manager-rhsm-certificates.x86_64 0:1.21.10-3.el7.centos
yajl.x86_64 0:2.0.4-4.el7

Complete!
[root@ip-172-31-29-81 centos]#
[root@ip-172-31-29-81 centos]# service docker status
Redirecting to /bin/systemctl status docker.service
• docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
   Docs: http://docs.docker.com
[root@ip-172-31-29-81 centos]#
```

Create a new project in github and upload the json code to create vpc later

```
root@ip-172-31-29-81:/home/centos

docker-client.x86_64 2:1.13.1-102.git7f2769b.el7.centos
docker-common.x86_64 2:1.13.1-102.git7f2769b.el7.centos
libaio.x86_64 0:0.3.109-13.el7
lvm2.x86_64 7:2.02.180-10.el7_6.8
lvm2-libs.x86_64 7:2.02.180-10.el7_6.8
oci-register-machine.x86_64 1:0-6.git2b44233.el7
oci-systemd-hook.x86_64 1:0.2.0-1.git05e6923.el7_6
oci-umount.x86_64 2:2.5-1.el7_6
python-pytoml.noarch 0:0.1.14-1.git7dea353.el7
subscription-manager-rhsm-certificates.x86_64 0:1.21.10-3.el7.centos
yajl.x86_64 0:2.0.4-4.el7

complete!
root@ip-172-31-29-81 centos]#
root@ip-172-31-29-81 centos]# service docker status
redirecting to /bin/systemctl status docker.service
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor p
t: disabled)
   Active: inactive (dead)
     Docs: http://docs.docker.com
root@ip-172-31-29-81 centos]# service docker start
redirecting to /bin/systemctl start docker.service
root@ip-172-31-29-81 centos]#
```

Check the status again



```
root@ip-172-31-29-81:/home/centos
└─521 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/libe...
└─528 /usr/bin/docker-containerd-current -l unix:///var/run/docker...

Sep 14 18:35:11 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal systemd[1]: Starte...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-29-81 centos]#
```

Next step is to pull jenkins docker container image

```
root@ip-172-31-29-81:/home/centos
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]: ^
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal systemd[1]: Starte...
Sep 14 18:35:12 ip-172-31-29-81.us-east-2.compute.internal dockerd-current[521]:
...
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-29-81 centos]# docker pull jenkins/jenkins:latest
Trying to pull repository docker.io/jenkins/jenkins ...
latest: Pulling from docker.io/jenkins/jenkins
9cc2ad81d40d: Pull complete
e6cb98e32a52: Pull complete
ae1b8d879bad: Pull complete
42cfa3699b05: Pull complete
8d27062ef0ea: Pull complete
9b91647396e3: Pull complete
7498c1055ea3: Pull complete
7cacb0f4ba6a: Pull complete
9b43de82d54d: Pull complete
54a8d03a5e09: Pull complete
f02d4bale2d0: Pull complete
```

```
root@ip-172-31-29-81:/home/centos
Trying to pull repository docker.io/jenkins/jenkins ...
latest: Pulling from docker.io/jenkins/jenkins
9cc2ad81d40d: Pull complete
e6cb98e32a52: Pull complete
ae1b8d879bad: Pull complete
42cfa3699b05: Pull complete
8d27062ef0ea: Pull complete
9b91647396e3: Pull complete
7498c1055ea3: Pull complete
7cacb0f4ba6a: Pull complete
9b43de82d54d: Pull complete
54a8d03a5e09: Pull complete
f02d4ba1e2d0: Pull complete
57724f7d3b4d: Pull complete
e63e1634315f: Pull complete
ca38052bd2e1: Pull complete
fe9aa054a31a: Pull complete
7c2baf636006: Pull complete
586cc3f79f76: Pull complete
a89d540835e1: Pull complete
545bdeec3592: Pull complete
Digest: sha256:5c9ab286b3ad430db0c2543f590599cf6ebbde8bf87d21ad97adf32a08183e86
Status: Downloaded newer image for docker.io/jenkins/jenkins:latest
[root@ip-172-31-29-81 centos]#
```

Create a repository in github and upload the code to create the vpc

[→](#) [GitHub, Inc. \[US\]](#) <https://github.com/mohanbrinda/AWS-Apche-modules> [📖](#) [☆](#) [🌟](#) [🔗](#)

mohanbrinda / AWS-Apche-modules

👁 Unwatch 1

★ Star 0

🍴 Fork 0

[🔗 Code](#) [🔔 Issues 0](#) [🔀 Pull requests 0](#) [📁 Projects 0](#) [📖 Wiki](#) [🔒 Security](#) [📊 Insights](#) [⚙ Settings](#)

No description, website, or topics provided.

[Manage topics](#)

📦 2 commits

🌿 1 branch

📦 0 releases

👤 1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

mohanbrinda Create Readme

Latest commit 0590c22 2 hours ago

[AWS APACHE MODULES.pdf](#)

Add files via upload

2 hours ago

[Readme](#)

Create Readme

2 hours ago

📖 Readme

This project covers the following three tasks:

- 1) By default, when user browses for Apache ServerName, a get index.html file is printed in the browser even if no filename is mentioned in the URL. The task is to change that behaviour and make default index file as test.html.
- 2) Configure Apache in such a way that if someone browses for a page in apache (eg: test1.html), their browser redirects them to a different page. (eg: test2.html) (Tip: Use rewrite module in Apache)
- 3) Enable Password Authentication in Apache, so that when someone browse your website, they get a popup asking them to enter a username and password, before apache will display pages to them on their browser (Tip: use module auth\_basic)



Run the docker image. Host jenkins application on port 8080

Docker run -p 8080:8080 -p 50000:50000 jenkins:latest

```
[root@ip-172-31-29-81 bin]# docker run -p 8080:8080 -p 50000:50000 jenkins:latest
Unable to find image 'jenkins:latest' locally
Trying to pull repository docker.io/library/jenkins ...
latest: Pulling from docker.io/library/jenkins
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4eee24d4dac: Pull complete
c58988e753d7: Pull complete
794a04897db9: Pull complete
70fcfa476f73: Pull complete
0539c80a02be: Pull complete
```

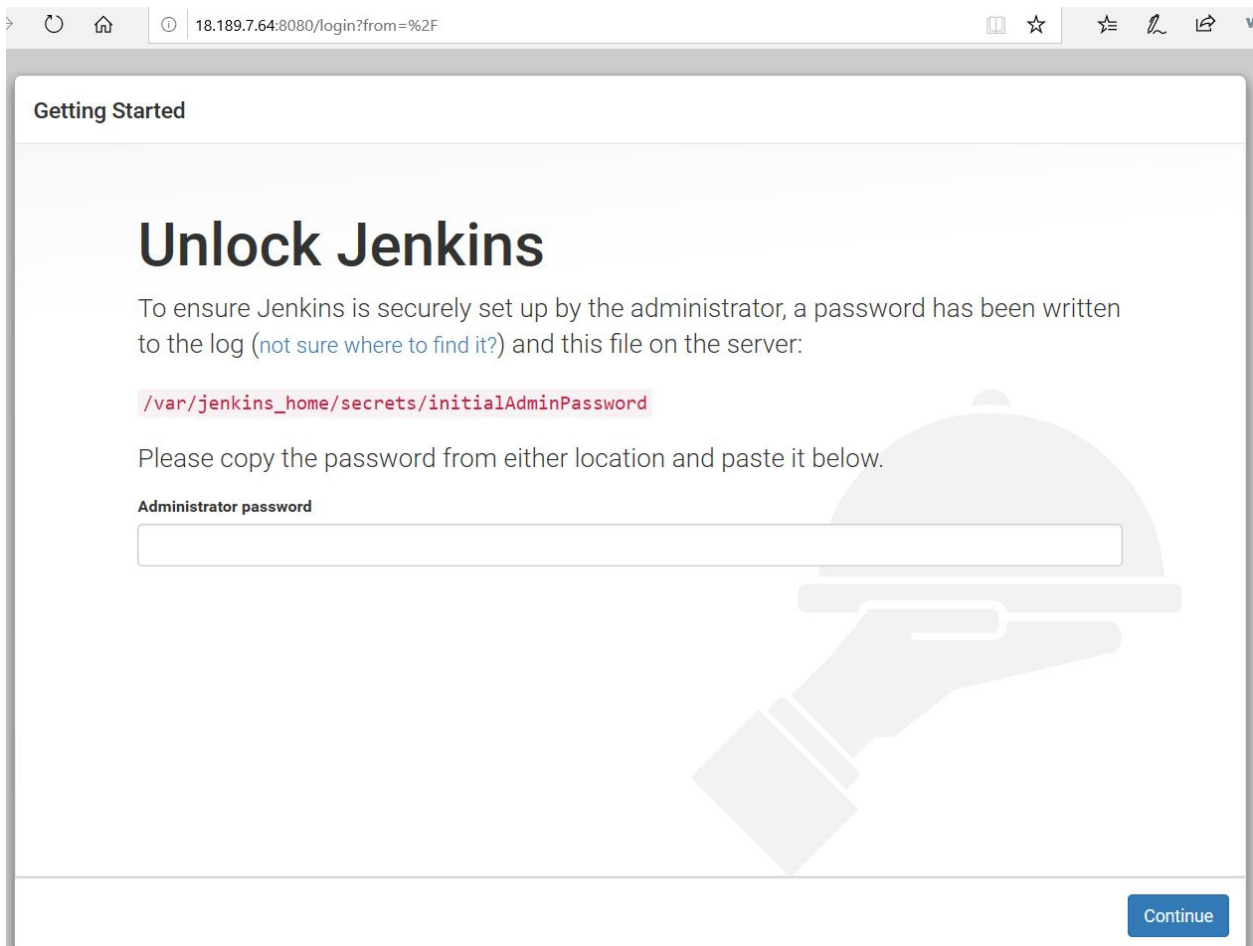
```
*****
*****
Sep 14, 2019 8:41:24 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Sep 14, 2019 8:41:24 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Sep 14, 2019 8:41:25 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tools.JDKInstaller
Sep 14, 2019 8:41:25 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Finished Download metadata. 8,759 ms
-> setting agent port for jnlp
-> setting agent port for jnlp... done
Sep 14, 2019 8:41:29 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Sep 14, 2019 8:41:29 PM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
```

Check the docker status

```
root@ip-172-31-29-81:/home/centos
login as: centos
Authenticating with public key "imported-openssh-key"
Last login: Sat Sep 14 18:06:04 2019 from pool-96-225-59-182.nwrknj.fios.verizon
net
centos@ip-172-31-29-81 ~]$ sudo su
[root@ip-172-31-29-81 centos]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
root@ip-172-31-29-81 centos]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAME
eac24af7eb8        jenkins:latest     "/bin/tini -- /usr..." 7 minutes ago
Up 7 minutes      0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp  sill
wozniak
root@ip-172-31-29-81 centos]#
```

Loginto the console after copying the ip address of the ec2 instance from the console

Enter the jenkins installation password



Getting Started

# Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

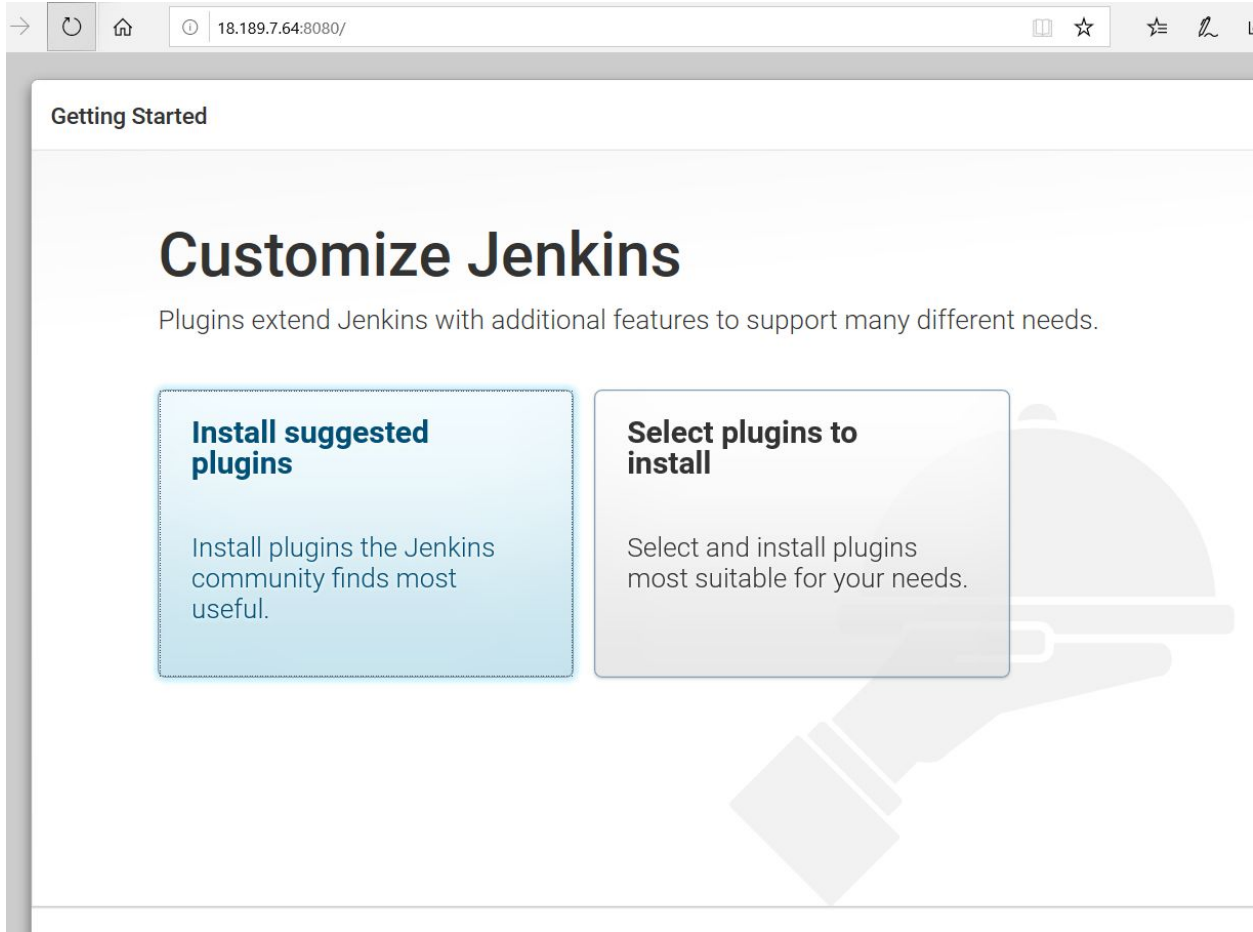
```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

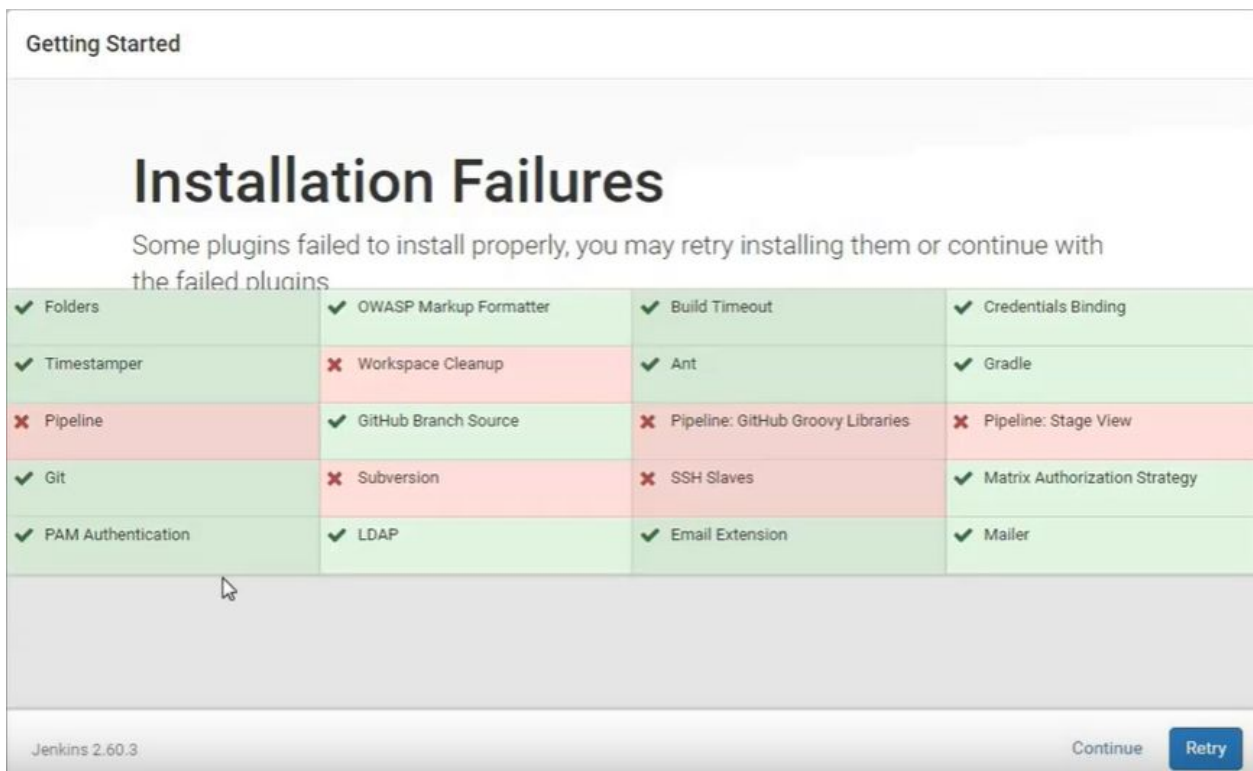
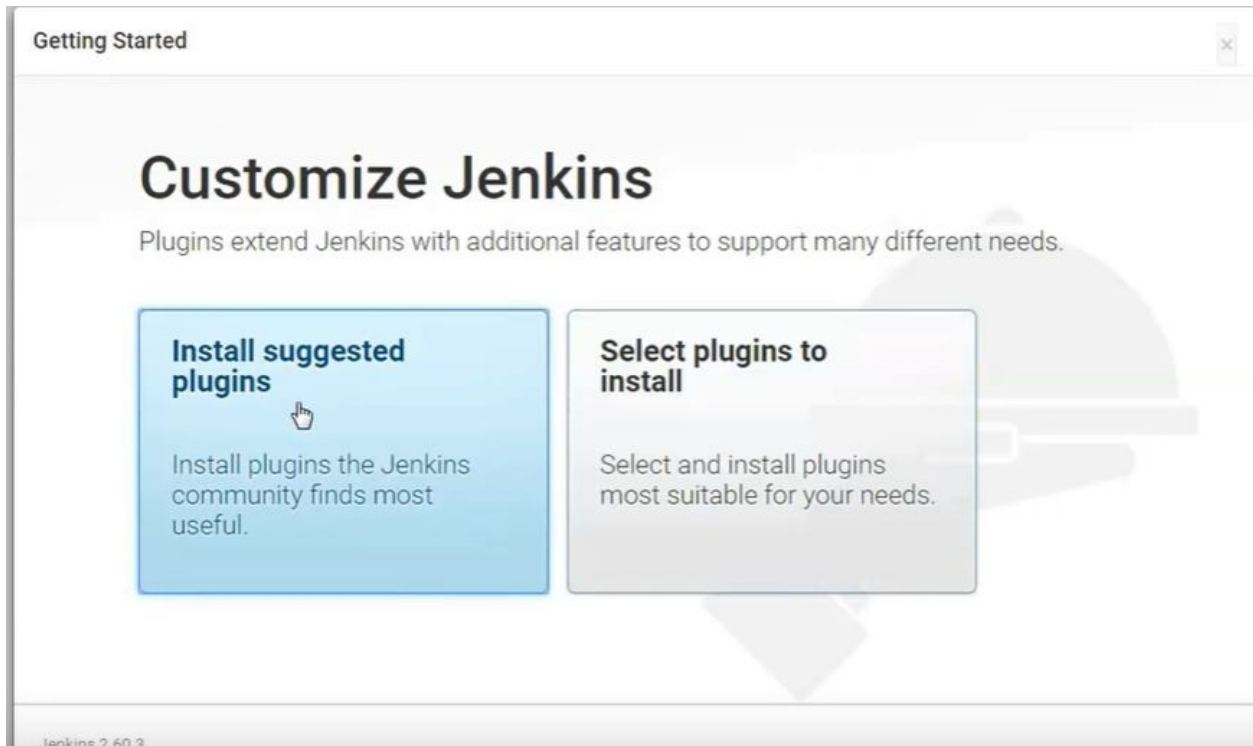
**Administrator password**

Continue

Continue the instructions provided



Select Install suggested plugins



Continue Installing Jenkins with failed plugins, then create the user and complete the installation steps

## Create First Admin User

Username:

Password:

Confirm password:

Full name:

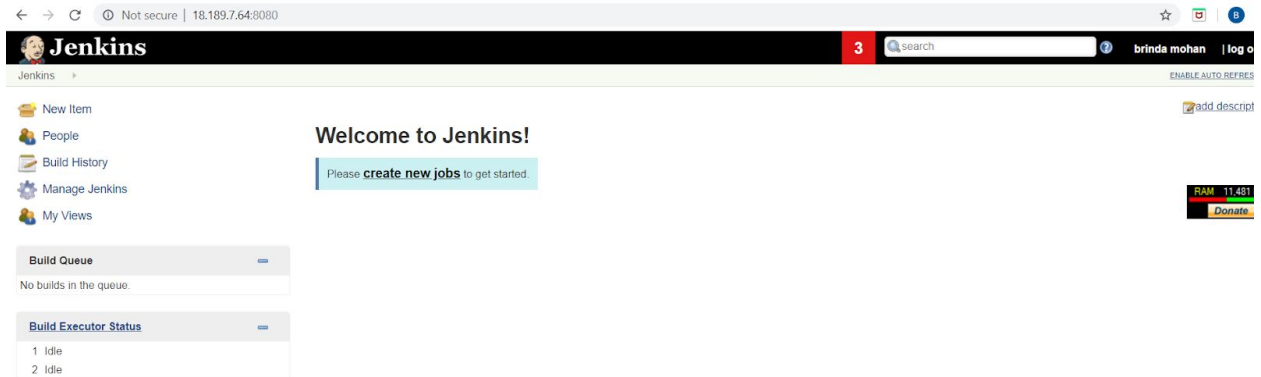
E-mail address:

## Jenkins is ready!

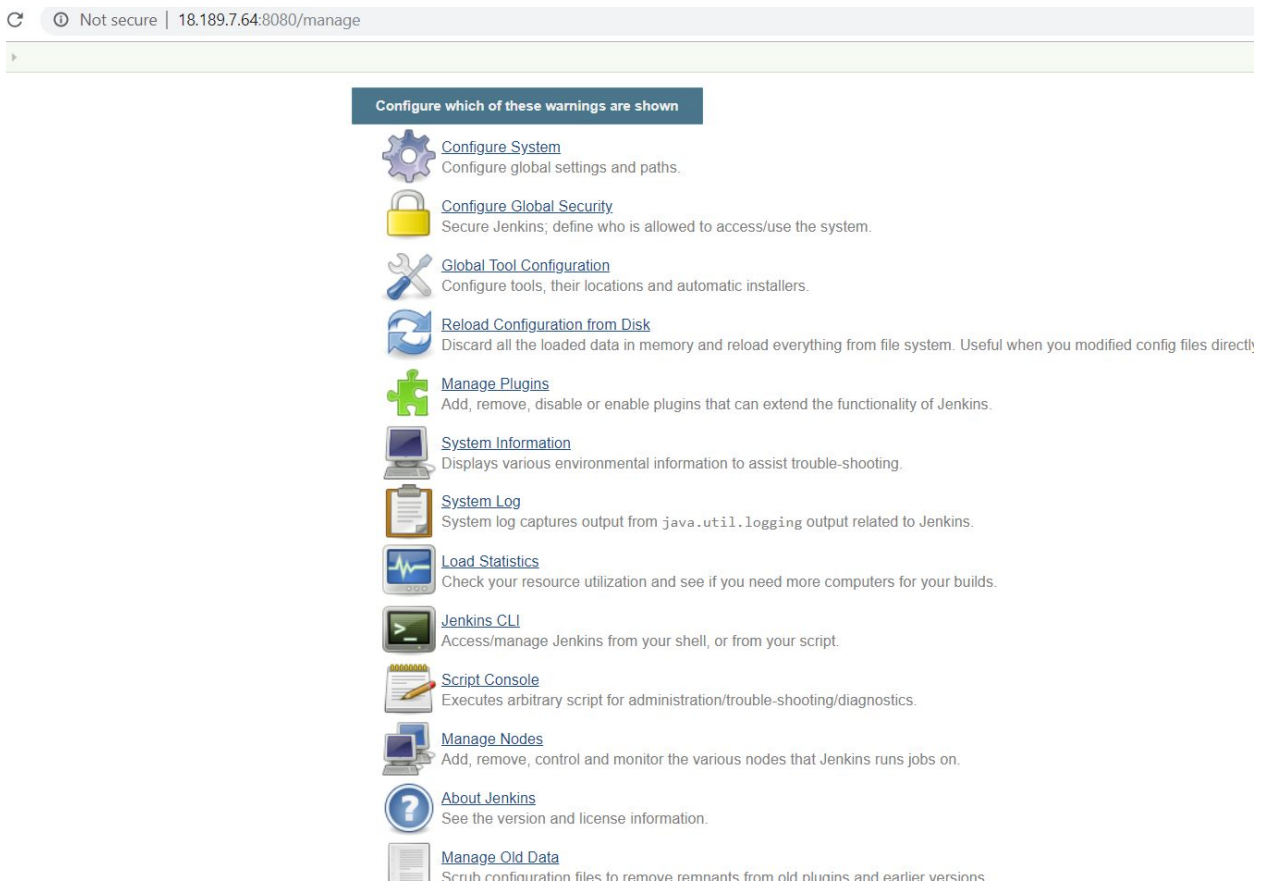
Your Jenkins setup is complete.

[Start using Jenkins](#)





## Select Manage plugins



Will display the Manage plugins page



→ ↻ Not secure | 18.189.7.64:8080/pluginManager/

# Jenkins

3 🔍 search ? brin

ins > Plugin Manager

Back to Dashboard  
Manage Jenkins  
Update Center

Filter: 🔍

Updates Available Installed Advanced

Install	Name ↓	Version	Installed
No updates			

Update information obtained: 17 min ago [Check now](#)

Select [All](#) [None](#)  
This page lists updates to the plugins you currently use.  
Disabled rows are already upgraded, awaiting restart. Shaded but selectable rows are [in progress or failed](#).

## Search for cloudformation and select install without restart

→ ↻ ⚠ Not secure | 18.189.7.64:8080/pluginManager/available

# Jenkins

3 🔍 search ? brinda m

kins > Plugin Manager

Back to Dashboard  
Manage Jenkins  
Update Center

Filter: 🔍 cloudform

Updates Available Installed Advanced

Install ↓	Name
<input checked="" type="checkbox"/>	<p><a href="#">jenkins-cloudformation-plugin</a></p> <p>Adds a build wrapper that can spawn an AWS Cloud Formation recipe at the start of the build and take it down at the end.</p> <p>Warning: This plugin requires dependent plugins that require Jenkins 2.121.1 or newer. Jenkins will refuse to load the dependent plugins requiring a newer version of Jenkins, and in turn loading this plugin will fail.</p> <p>Warning: This plugin version may not be safe to use. Please review the following security notices:</p> <ul style="list-style-type: none"> <li><a href="#">Credentials stored in plain text</a></li> </ul>

[Install without restart](#) [Download now and install after restart](#) Update information obtained: 19 min ago [Check now](#)

## Next upload code(json file) to create vpc in aws to github

[mohanbrinda](#) / [Jenkins](#) Unwatch ▾ 1

[Code](#) Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

*No description, website, or topics provided.*

[Manage topics](#)

1 commit 1 branch 0 releases

Branch: [master ▾](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find](#)

[mohanbrinda](#) Add files via upload






[vpc.json](#) Add files via upload

## Go to the jenkins dashboard and create new job

← → ↻ ⓘ Not secure | 18.189.7.64:8080


# Jenkins

Jenkins ▶


-  New Item
-  People
-  Build History
-  Manage Jenkins
-  My Views

## Welcome to Jenkins!

Please **create new jobs** to get started.

**Build Queue** 

No builds in the queue.

**Build Executor Status** 

- 1 Idle
- 2 Idle

Enter the name, select freestyle project and click ok


← → ↻ ⚠ Not secure | 18.189.7.64:8080/newJob


# Jenkins


Jenkins ▶

**Enter an item name**

» Required field

 **Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this something other than software build.

 **Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a namespace, so you can have multiple things of the same name as long as they are in different folders.



Copy the git repository url and add jenkins credentials

**General** | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

[Plain text] [Preview](#)

- ☐ Discard old builds
- ☐ GitHub project
- ☐ This project is parameterised
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary

[Advanced...](#)

### Source Code Management

☐ None  
☒ Git

Repositories

Repository URL  ❗ Please enter Git repository.

Credentials - none - [Add](#)

Next fill in all the cloudformation stack creation details and click on save

Create AWS CloudFormation stack

**Stack configuration**

AWS Region US East (Northern Virginia) Region

Cloud Formation recipe file/S3 URL. (.json)  ❗ Empty recipe file.

Stack name  ❗ Empty stack name

Stack description

Cloud Formation parameters


Timeout (seconds)

AWS Access Key  ❗ Empty aws access key

AWS Secret Key  ❗ Empty aws secret key

Automatically delete the stack when the job completes ☒

Then , loginto the AWS console to verify the newly created VPC

 **Jenkins Credentials Provider: Jenkins**

**Add Credentials**

Domain:

Kind:

Scope:

Username:

Password:

ID:

Description:

Services ▾ Resource Groups ▾ ⭐

mohanbrinda ▾ Ohio ▾ Support

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▾	VPC ID ▴	State ▾	IPv4 CIDR	IPv6 CIDR	DHCP options
<input checked="" type="checkbox"/>		vpc-9ba84ff0	available	172.31.0.0/16	-	dopt-ab22dbc0

Ensure the vps has been created. Back on the jenkins dashboard click on bid now and ensure that the new stack is present along with the vpc. The console output should have finished successfully

```
myVPC-1179dcf5-e72d-4710-b719-ca3b1b20a552 - AWS::EC2::VPC - DELETE_IN_PROGRESS - null
Successfully created stack: cft-jenkin-vpc
Finished: SUCCESS
```

References:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc.html>

<https://jenkins.io/doc/book/installing/#downloading-and-running-jenkins-in-docker>

<https://docs.docker.com>