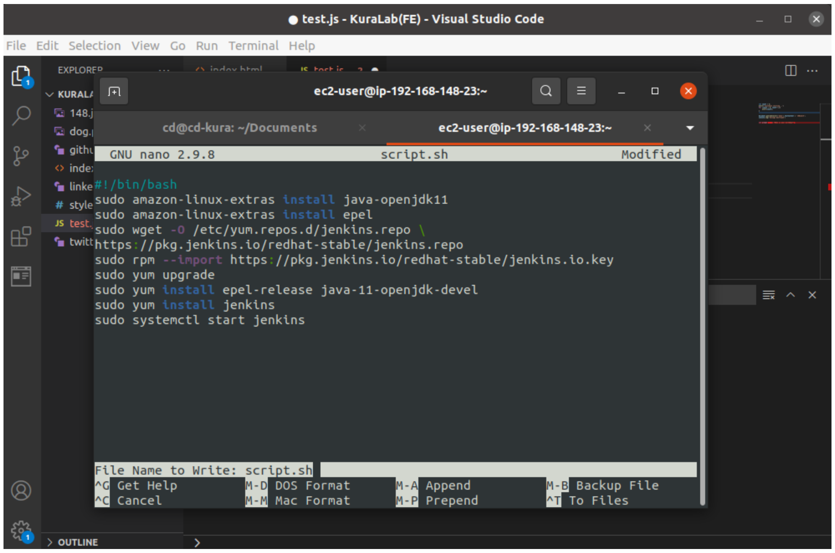
Important Note – Due to the increasing cost on my account I deleted most everything as soon as I was done. As a result, my instances, target group and a few things will appear empty in my step screenshots but proof of completion will be shown at the end.

Step 1: Configure Jenkins using the following

```
sudo amazon-linux-extras install java-openjdk11
sudo amazon-linux-extras install epel
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum upgrade
sudo yum install epel-release java-11-openjdk-devel
sudo yum install jenkins
sudo systemctl start Jenkins
```

```
sudo yum install git
```

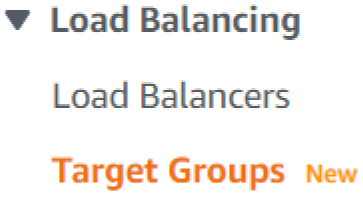
Note: used mine in the terminal

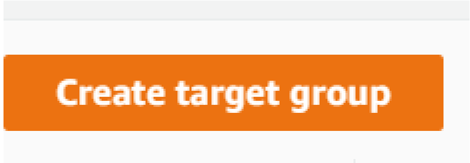


● test.js - KuraLab(FE) - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ec2-user@ip-192-168-148-23:~
cd@cd-kura: ~/Documents
ec2-user@ip-192-168-148-23:~
OUTLINE

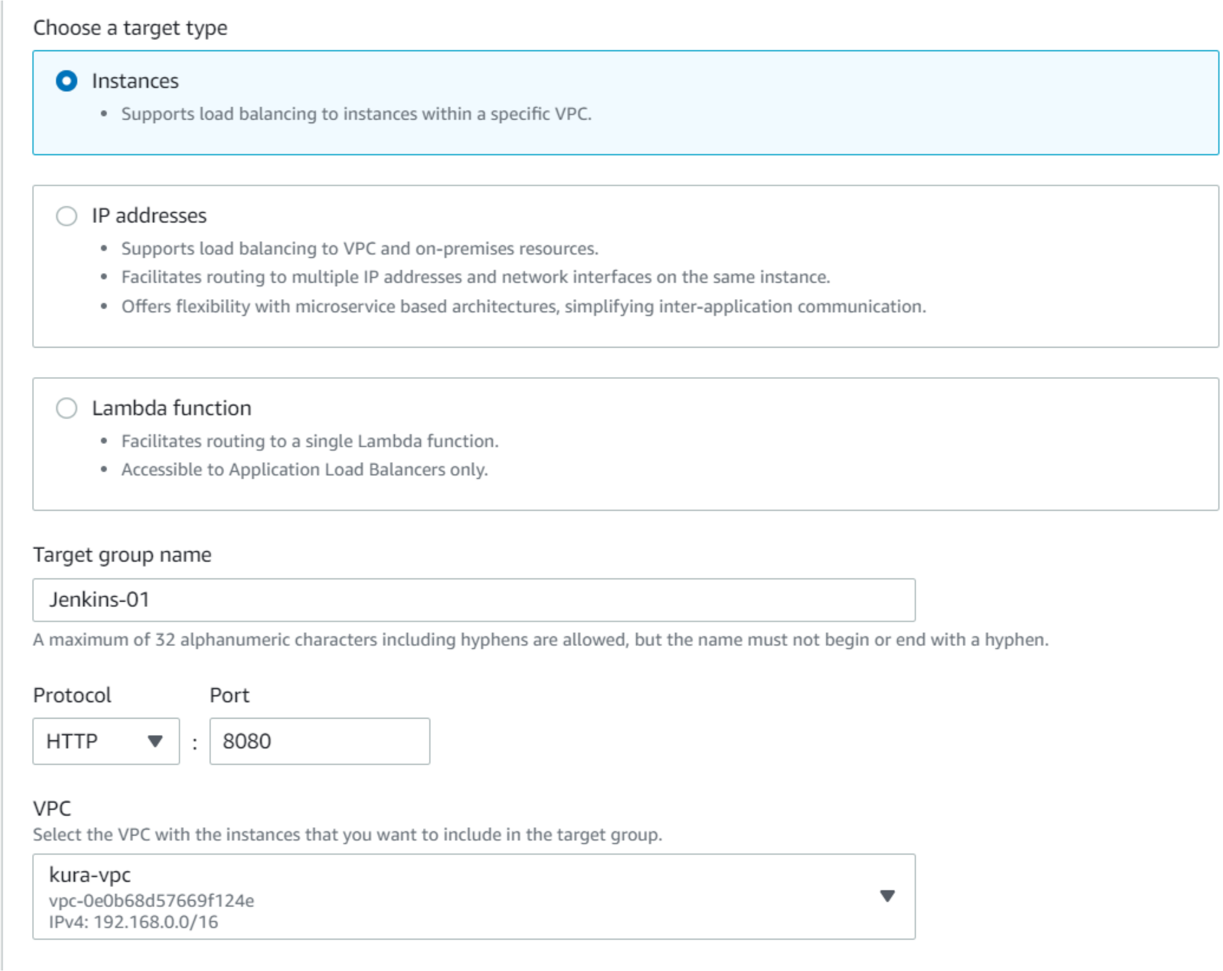
```
  GNU nano 2.9.8                  script.sh                          Modified

#!/bin/bash
sudo amazon-linux-extras install java-openjdk11
sudo amazon-linux-extras install epel
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum upgrade
sudo yum install epel-release java-11-openjdk-devel
sudo yum install jenkins
sudo systemctl start jenkins

File Name to Write: script.sh
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```

Step 2: create a target group





Step 3: Select Instance and give it a name. Then set http port to 8080 and choose your vpc as shown below:


Choose a target type

● Instances
- Supports load balancing to instances within a specific VPC.

○ IP addresses
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

○ Lambda function
- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Target group name

Jenkins-01

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

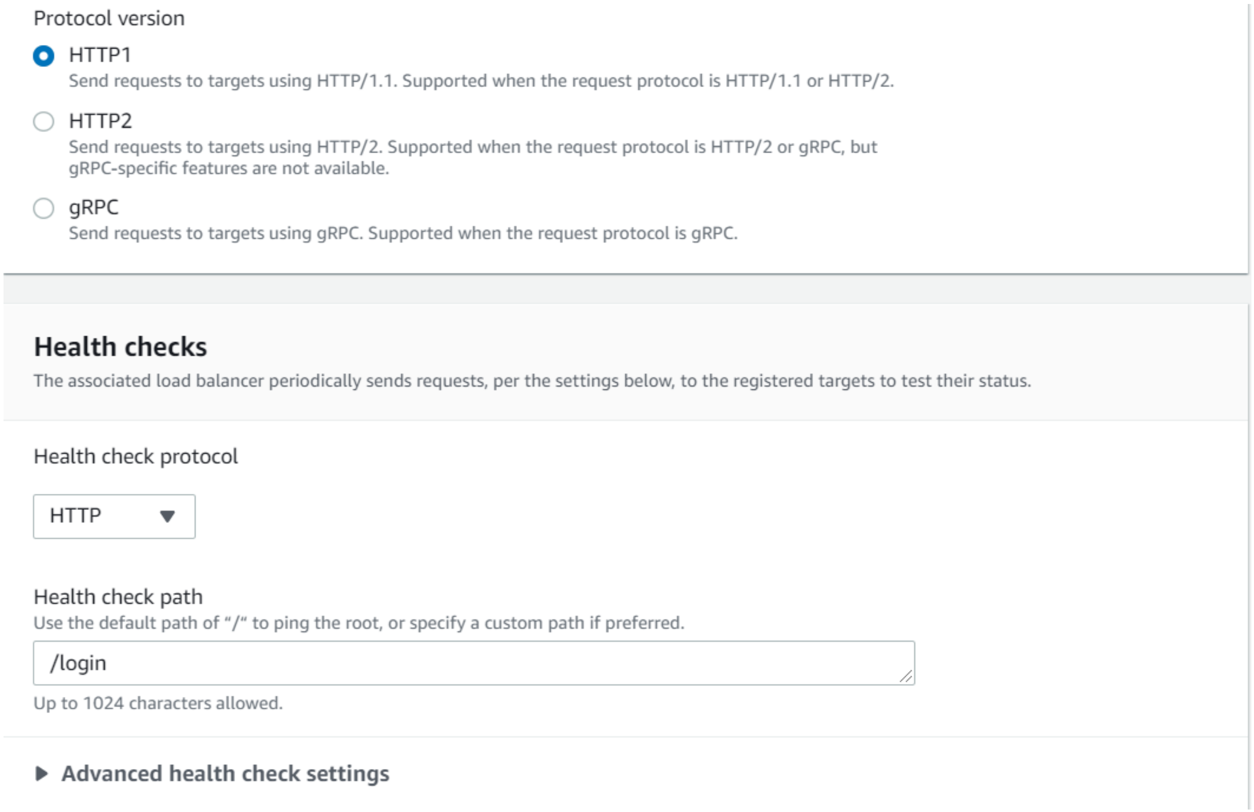
Protocol: HTTP ▼ : Port: 8080

VPC

Select the VPC with the instances that you want to include in the target group.

kura-vpc
vpc-0e0b68d57669f124e
IPv4: 192.168.0.0/16 ▼

Step 4: Set your protocol version to http1 and health check path to “/login” as shown below:



Protocol version

● HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

○ HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

○ gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP ▼

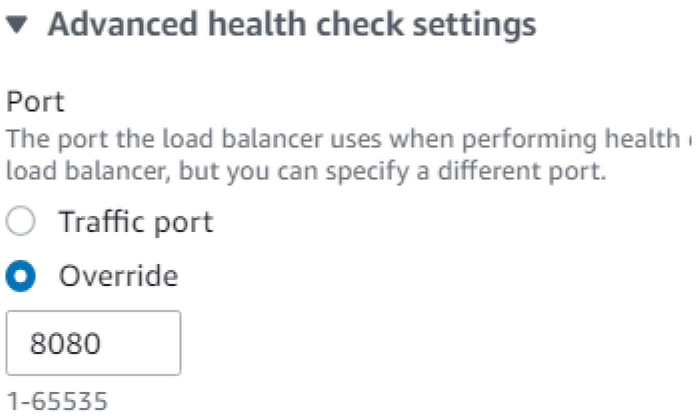
Health check path
Use the default path of “/“ to ping the root, or specify a custom path if preferred.

/login

Up to 1024 characters allowed.

▶ **Advanced health check settings**

Step 5: Select Advanced health check settings and set Override to 8080. Then click next page



▼ **Advanced health check settings**

Port
The port the load balancer uses when performing health
load balancer, but you can specify a different port.

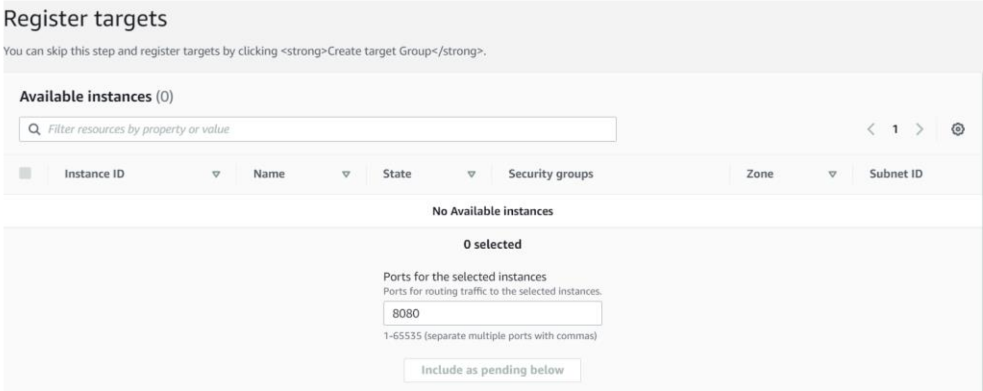
○ Traffic port

● Override

8080

1-65535

Step 6: Register your targets by selecting your private available instance. Make sure your “ports for the selected instances is 8080” and then click include as pending below. Once done create target group.



Register targets

You can skip this step and register targets by clicking <strong>Create target Group</strong>.

Available instances (0)

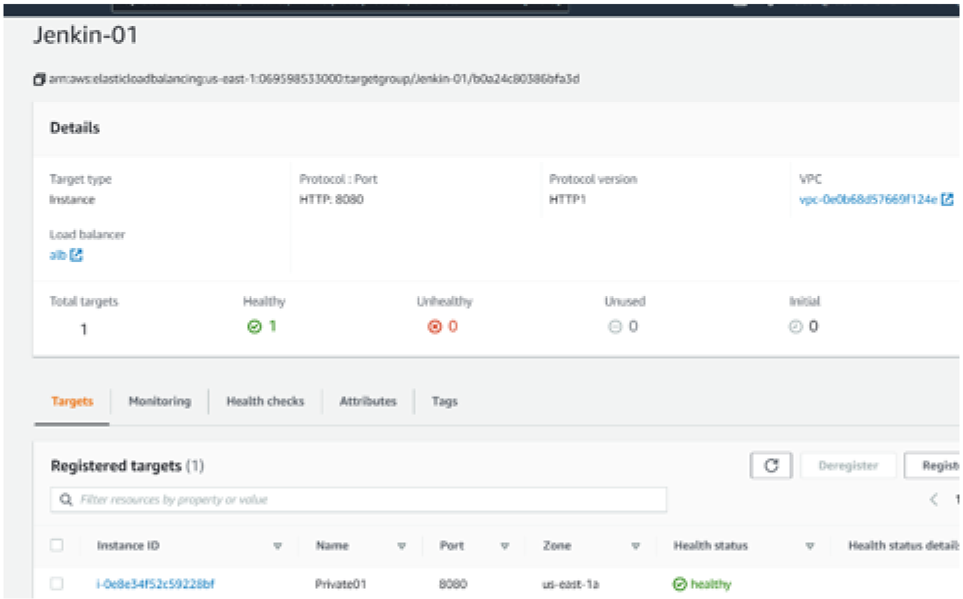
Filter resources by property or value

□	Instance ID	Name	State	Security groups	Zone	Subnet ID

No Available instances

0 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
8080
1-65535 (separate multiple ports with commas)



Jenkin-01

arn:aws:elasticloadbalancing:us-east-1:069598533000:targetgroup/Jenkin-01/b0a24c80386bfa3d

Details

Target type: Instance
Protocol : Port: HTTP: 8080
Protocol version: HTTP1
VPC: vpc-0e0b68d57669f124e
Load balancer: alb

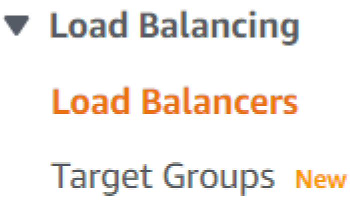
Total targets	Healthy	Unhealthy	Unused	Initial
1	1	0	0	0

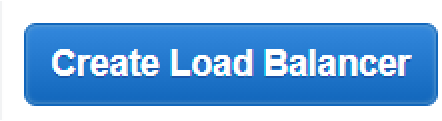
Registered targets (1)

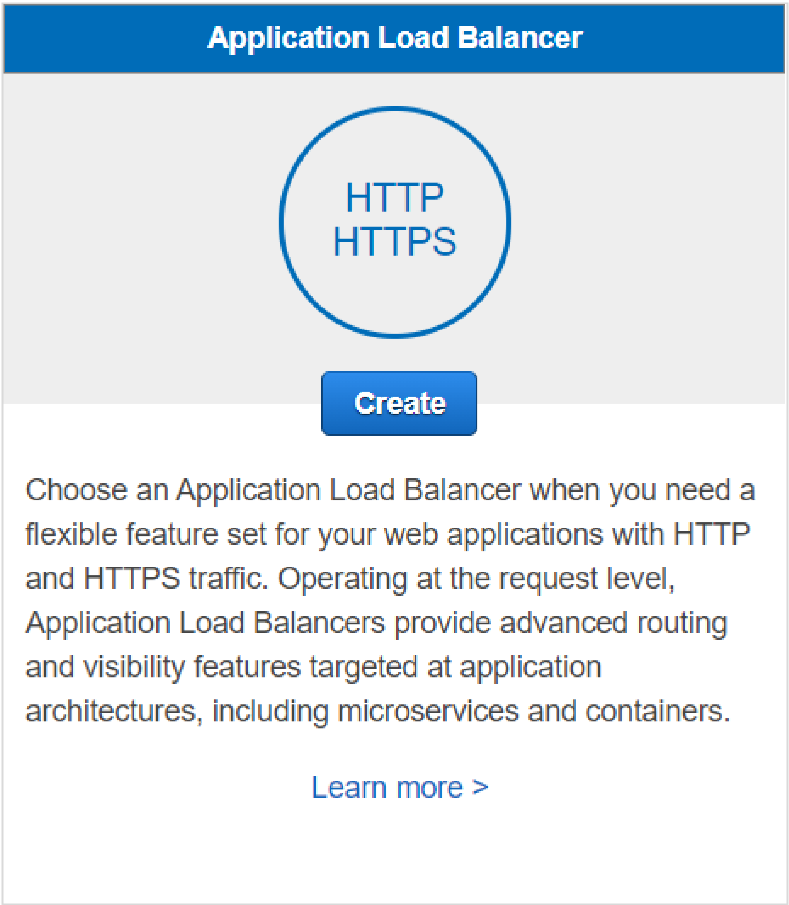
Filter resources by property or value

□	Instance ID	Name	Port	Zone	Health status	Health status detail:
□	i-0e8e34f52c59228bf	Private01	8080	us-east-1a	healthy	

Step 7: Create load balancer by selecting the following:



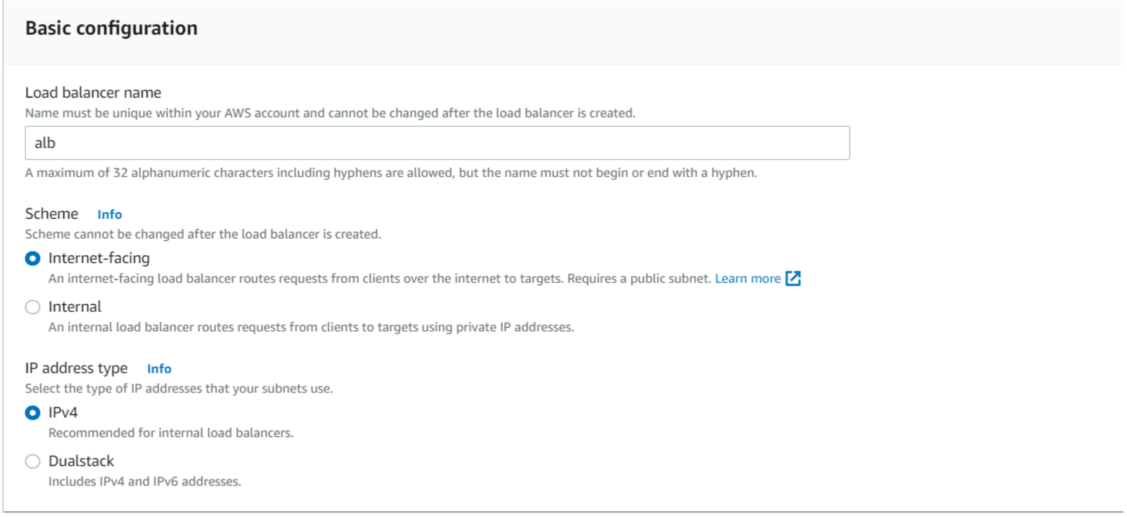




Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Step 8: Make sure to give it a name and keep the other settings as shown



Basic configuration

Load balancer name
Name must be unique within your AWS account and cannot be changed after the load balancer is created.

alb

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme cannot be changed after the load balancer is created.

● Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more

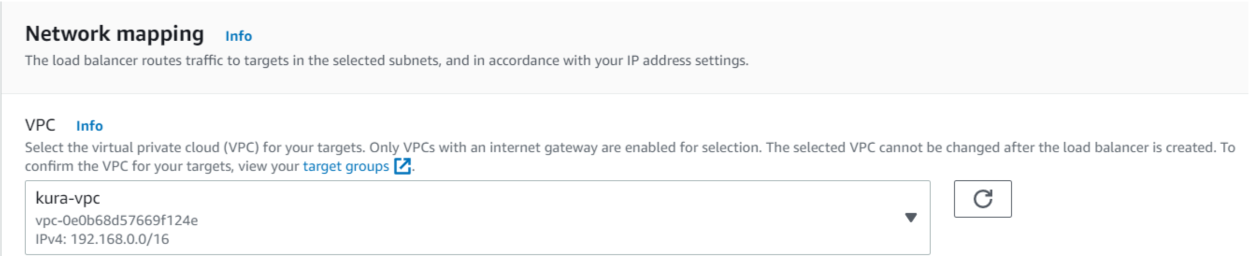
○ Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info
Select the type of IP addresses that your subnets use.

● IPv4
Recommended for internal load balancers.

○ Dualstack
Includes IPv4 and IPv6 addresses.

Step 9: Select your vpc



Network mapping Info

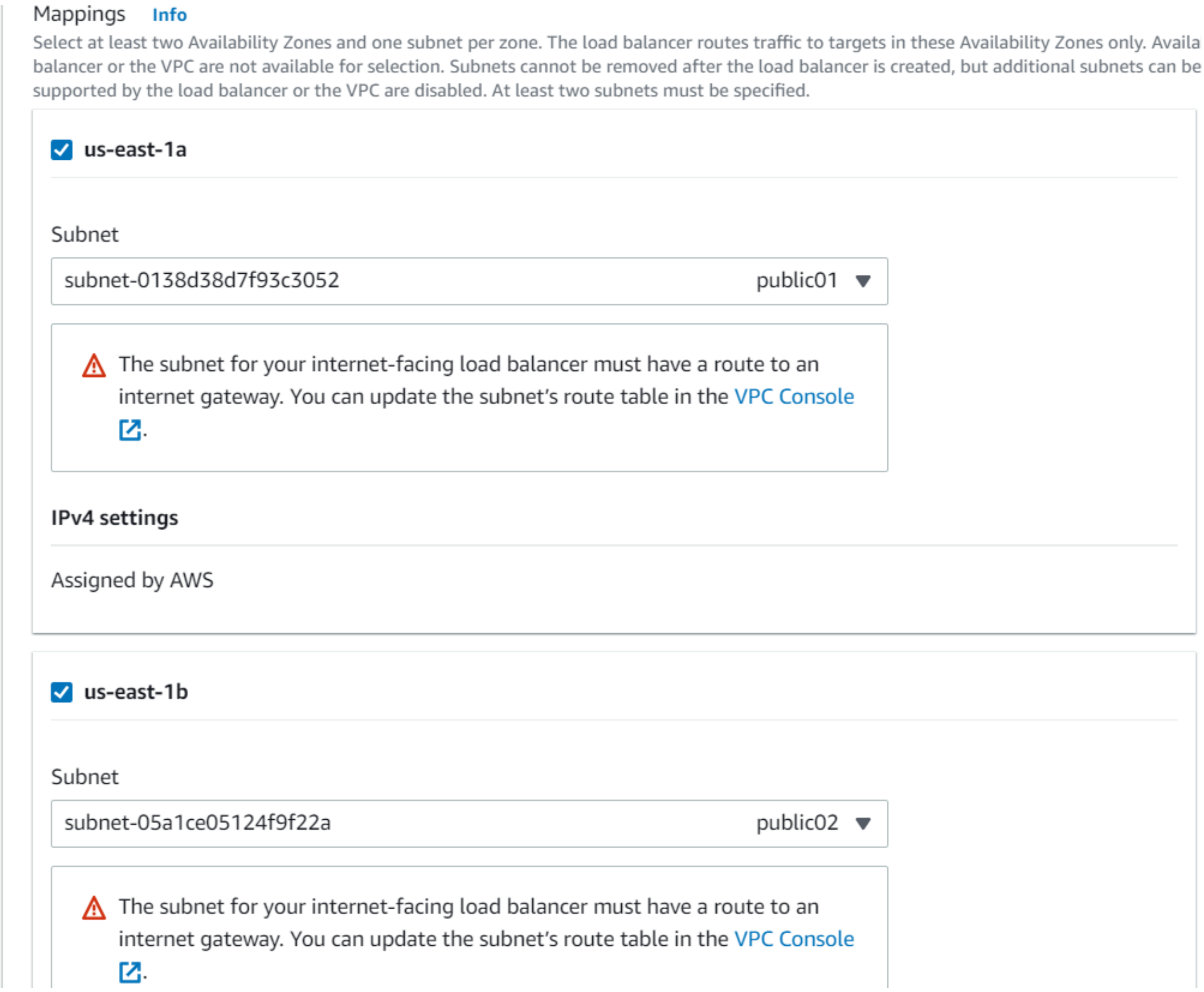
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups .

kura-vpc
vpc-0e0b68d57669f124e
IPv4: 192.168.0.0/16

Step 10: Then select your mapping info ensuring that both subnets are public from the regions shown below



Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availa balancer or the VPC are not available for selection. Subnets cannot be removed after the load balancer is created, but additional subnets can be supported by the load balancer or the VPC are disabled. At least two subnets must be specified.

☑ **us-east-1a**

Subnet

subnet-0138d38d7f93c3052 public01

⚠ The subnet for your internet-facing load balancer must have a route to an internet gateway. You can update the subnet's route table in the VPC Console .

IPv4 settings

Assigned by AWS

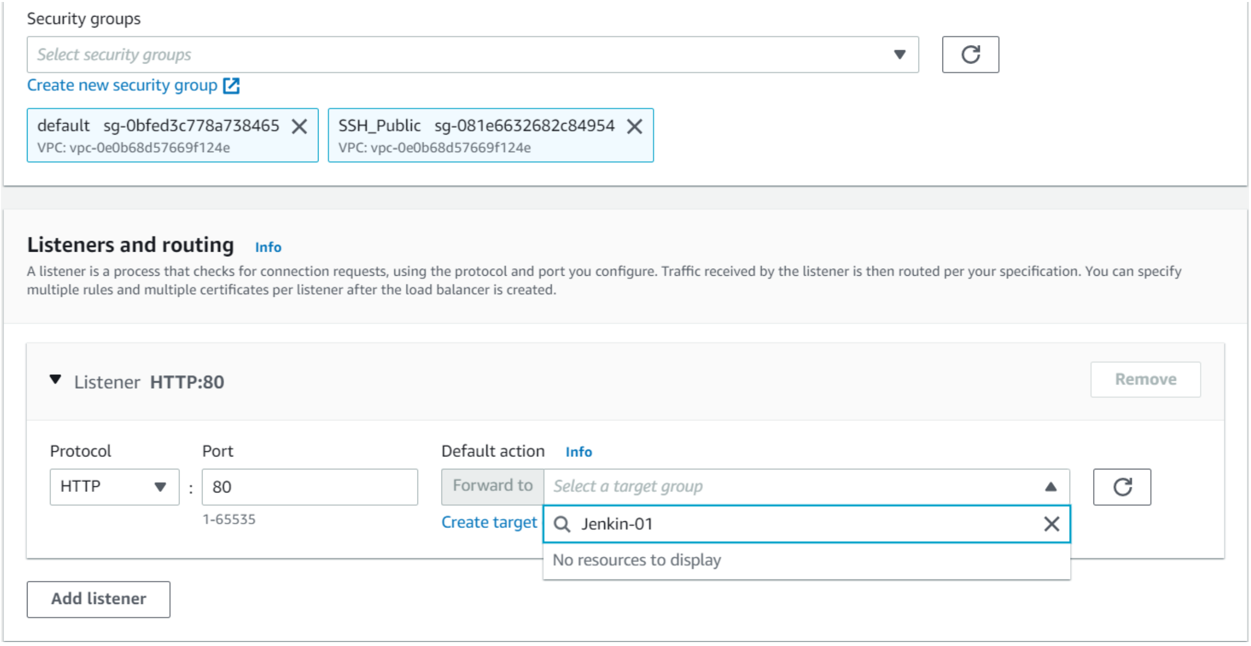
☑ **us-east-1b**

Subnet

subnet-05a1ce05124f9f22a public02

⚠ The subnet for your internet-facing load balancer must have a route to an internet gateway. You can update the subnet's route table in the VPC Console .

Step 11: Select your security group and add your target group where shown. Then create load balancer.



Security groups

Select security groups

Create new security group

default sg-0bfed3c778a738465
VPC: vpc-0e0b68d57669f124e

SSH_Public sg-081e6632682c84954
VPC: vpc-0e0b68d57669f124e

Listeners and routing Info

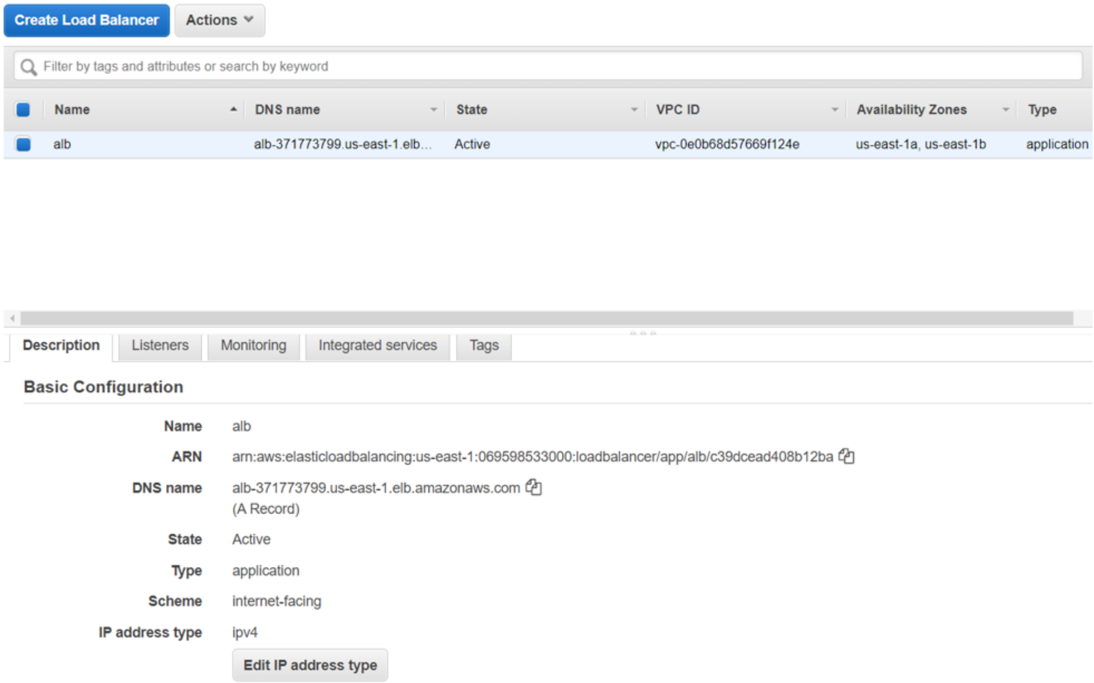
A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

Listener **HTTP:80**

Protocol: HTTP
Port: 80
1-65535

Default action Info
Forward to: Select a target group
Jenkin-01
No resources to display

Note: my target group was already deleted when taking this screenshot. However this is proof it was built successfully.



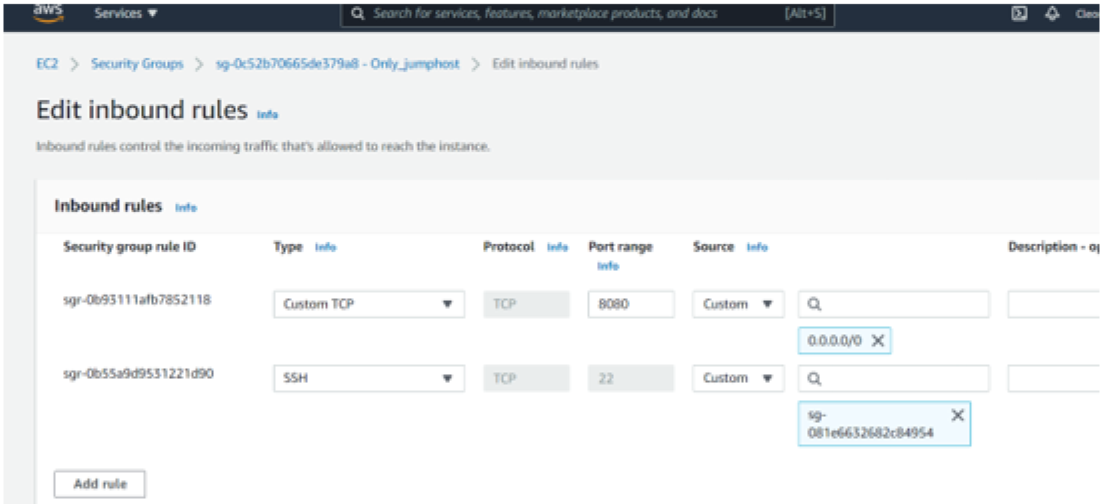
Name	DNS name	State	VPC ID	Availability Zones	Type
alb	alb-371773799.us-east-1.elb...	Active	vpc-0e0b68d57669f124e	us-east-1a, us-east-1b	application

Basic Configuration

Name	alb
ARN	arn:aws:elasticloadbalancing:us-east-1:069598533000:loadbalancer/app/alb/c39dcead408b12ba
DNS name	alb-371773799.us-east-1.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	internet-facing
IP address type	ipv4

After it's built successfully, you'll see the status becomes active

Step 12: Edit only_jumphost security setting. Set Custom TCP port to 8080 and Source to 0.0.0.0.



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]

EC2 > Security Groups > sg-0c52b70665de379a8 - Only_jumphost > Edit inbound rules

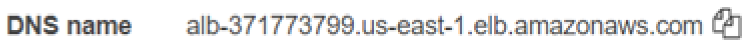
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

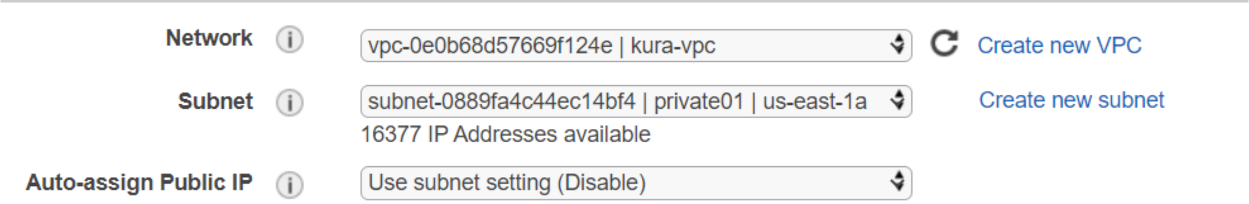
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - o
sgr-0b93111afb7852118	Custom TCP ▼	TCP	8080	Custom ▼	0.0.0.0/0 ✕	
sgr-0b55a9d9531221d90	SSH ▼	TCP	22	Custom ▼	sg-081e6632682c84954 ✕	

Step 13: Go into Load Balancer and copy the DNS name and paste it in url

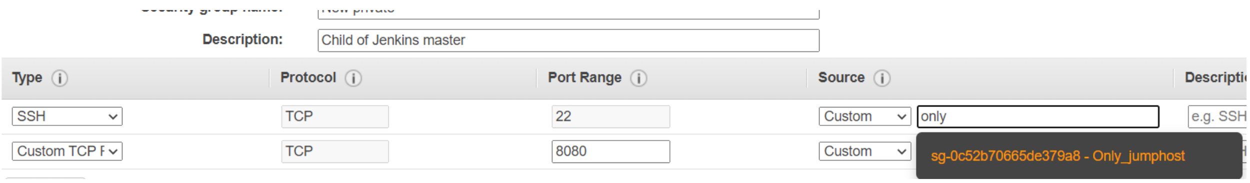


DNS name alb-371773799.us-east-1.elb.amazonaws.com

Step 14: Create another EC2 inside the same private subnet of the Jenkins master (This will be the agent).

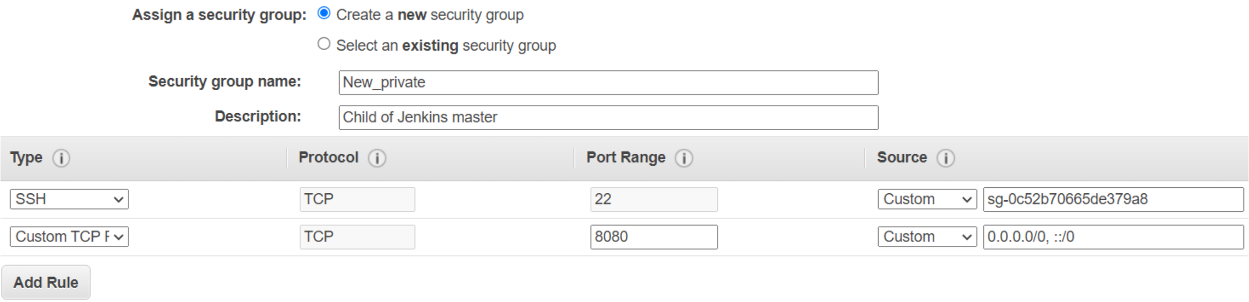


Network	vpc-0e0b68d57669f124e | kura-vpc	Create new VPC
Subnet	subnet-0889fa4c44ec14bf4 | private01 | us-east-1a 16377 IP Addresses available	Create new subnet
Auto-assign Public IP	Use subnet setting (Disable)	



Description: Child of Jenkins master

Type	Protocol	Port Range	Source		Descripti
SSH	TCP	22	Custom	only	e.g. SSH
Custom TCP F	TCP	8080	Custom	sg-0c52b70665de379a8 - Only_jumphost	



Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

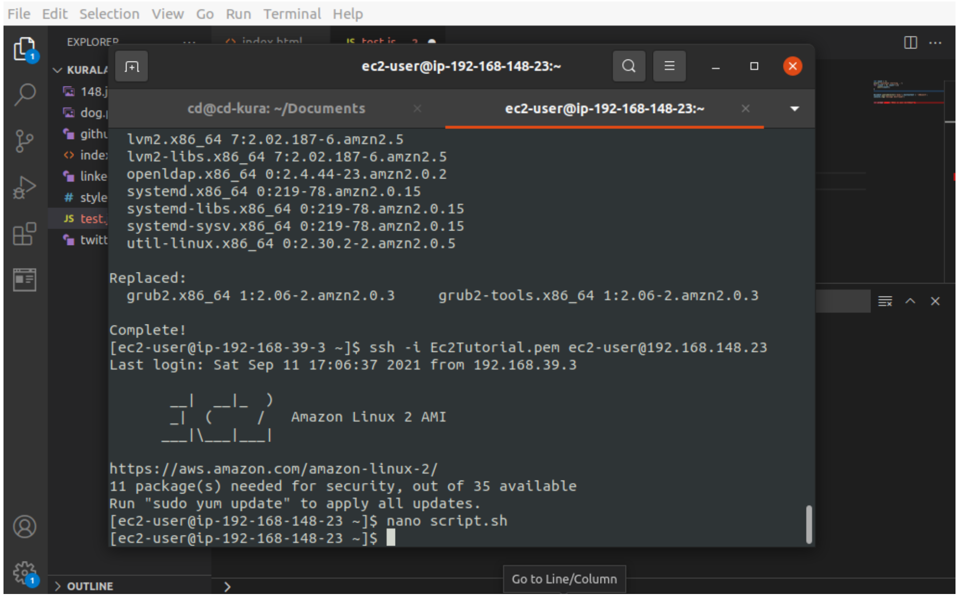
Security group name: New_private

Description: Child of Jenkins master

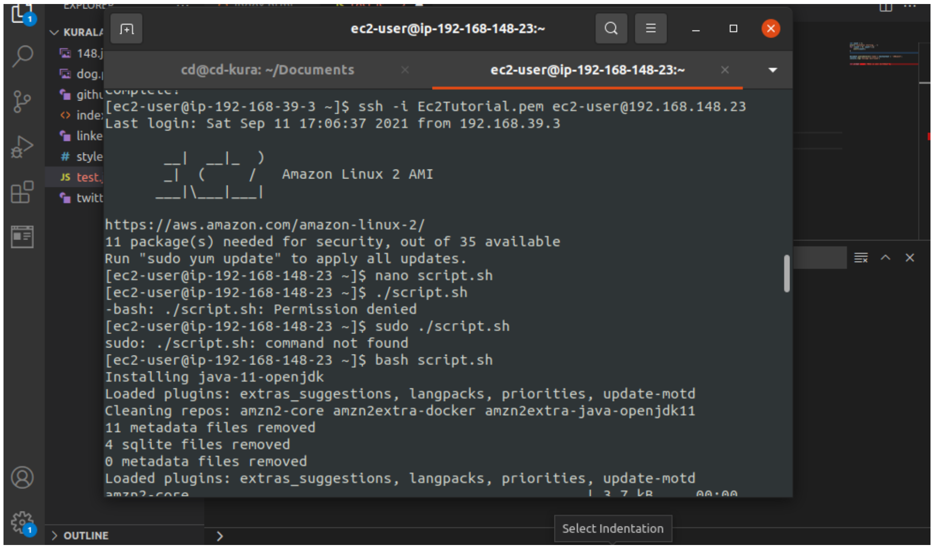
Type	Protocol	Port Range	Source	
SSH	TCP	22	Custom	sg-0c52b70665de379a8
Custom TCP F	TCP	8080	Custom	0.0.0.0/0, ::/0

Put the security group the has the Jenkins on it for the SSH source. In my case it was Only_Jumphost.

Step 15: SSH into JumpHost (Public01) and once inside, then SSH into private01. Then create a new key and put the RSA information into it by using “nano pem name” example nano EC2Tutorial.pem. Then change permissions using chmod 400 EC2Turtial.pem. Once inside there, SSH once again into private01-child using ssh -i EC2Tutorial.pem ec2-user@Private IPv4 addresses.


File Edit Selection View Go Run Terminal Help
EXPLORER
KURALA
148.j
dog.
githu
inde
linke
style
test.
twitt
ec2-user@ip-192-168-148-23:~
cd@cd-kura: ~/Documents
ec2-user@ip-192-168-148-23:~
OUTLINE
Go to Line/Column

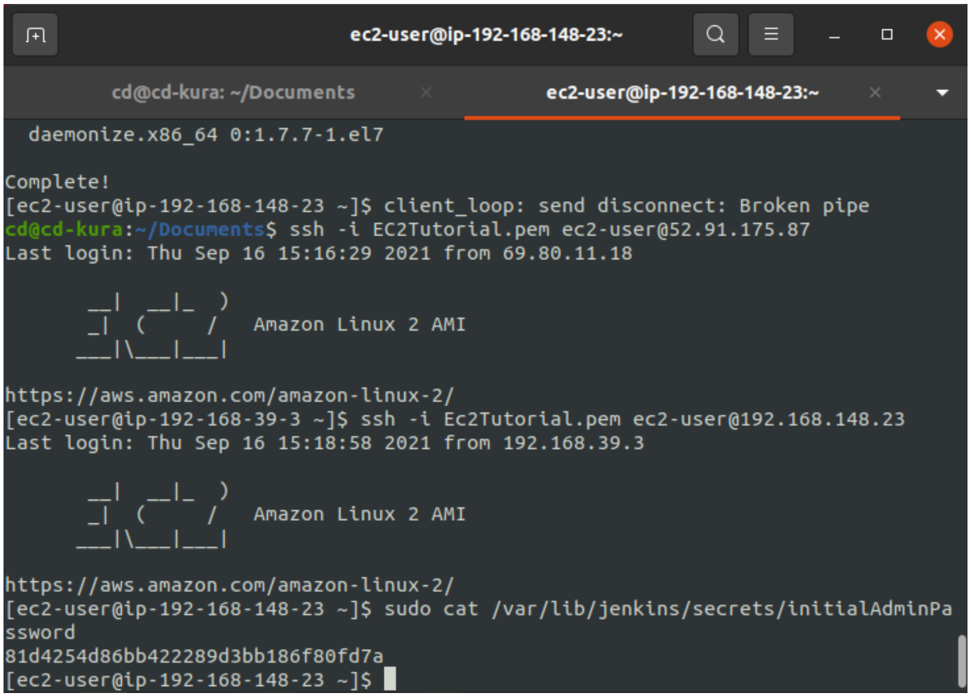
```
  lvm2.x86_64 7:2.02.187-6.amzn2.5
  lvm2-libs.x86_64 7:2.02.187-6.amzn2.5
  openldap.x86_64 0:2.4.44-23.amzn2.0.2
  systemd.x86_64 0:219-78.amzn2.0.15
  systemd-libs.x86_64 0:219-78.amzn2.0.15
  systemd-sysv.x86_64 0:219-78.amzn2.0.15
  util-linux.x86_64 0:2.30.2-2.amzn2.0.5

Replaced:
  grub2.x86_64 1:2.06-2.amzn2.0.3     grub2-tools.x86_64 1:2.06-2.amzn2.0.3

Complete!
[ec2-user@ip-192-168-39-3 ~]$ ssh -i Ec2Tutorial.pem ec2-user@192.168.148.23
Last login: Sat Sep 11 17:06:37 2021 from 192.168.39.3

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
11 package(s) needed for security, out of 35 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-148-23 ~]$ nano script.sh
[ec2-user@ip-192-168-148-23 ~]$ 
```



```
[ec2-user@ip-192-168-39-3 ~]$ ssh -i Ec2Tutorial.pem ec2-user@192.168.148.23
Last login: Sat Sep 11 17:06:37 2021 from 192.168.39.3

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
11 package(s) needed for security, out of 35 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-148-23 ~]$ nano script.sh
[ec2-user@ip-192-168-148-23 ~]$ ./script.sh
-bash: ./script.sh: Permission denied
[ec2-user@ip-192-168-148-23 ~]$ sudo ./script.sh
sudo: ./script.sh: command not found
[ec2-user@ip-192-168-148-23 ~]$ bash script.sh
Installing java-11-openjdk
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-java-openjdk11
11 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

Note: here is where I ran my bootstrap from step one allowing me to install Jenkins.

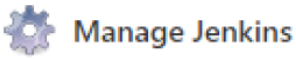
Step 16: Once on Jenkin’s page, sudo cat /var/lib/jenkins/secrets/initialAdminPassword and install suggested plugins.



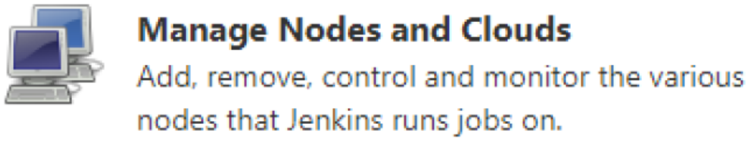
```
  daemonize.x86_64 0:1.7.7-1.el7

Complete!
[ec2-user@ip-192-168-148-23 ~]$ client_loop: send disconnect: Broken pipe
cd@cd-kura:~/Documents$ ssh -i EC2Tutorial.pem ec2-user@52.91.175.87
Last login: Thu Sep 16 15:16:29 2021 from 69.80.11.18

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-39-3 ~]$ ssh -i Ec2Tutorial.pem ec2-user@192.168.148.23
Last login: Thu Sep 16 15:18:58 2021 from 192.168.39.3

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-192-168-148-23 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPa
ssword
81d4254d86bb422289d3bb186f80fd7a
[ec2-user@ip-192-168-148-23 ~]$ 
```

Note: successfully logged into Jenkins.

Step 17: Configure the Jenkins master to SSH into the agent. Once logged into Jenkins, go to Mange Jenkins.



Select manage nodes



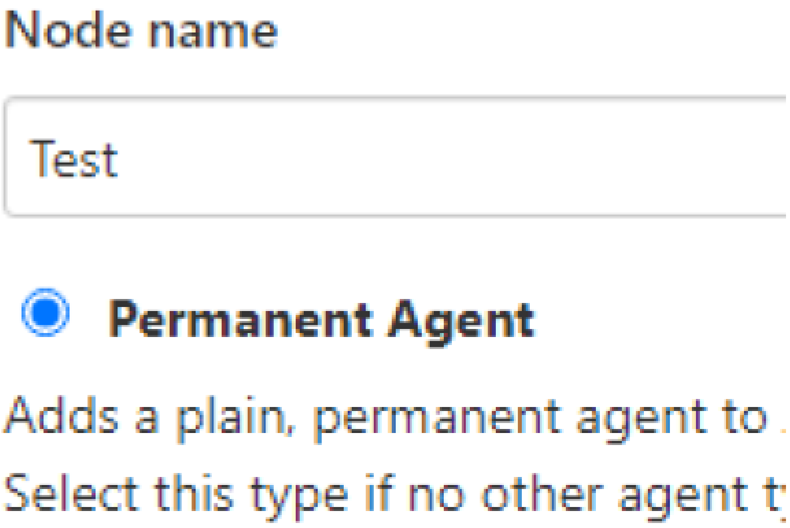
Manage Nodes and Clouds

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Select new node

New Node

Give the node a name and select Permanent Agent. Make sure to also give it a description.



Node name

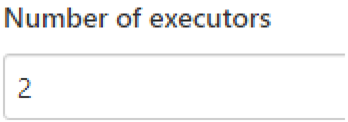
Test

● **Permanent Agent**

Adds a plain, permanent agent to

Select this type if no other agent t

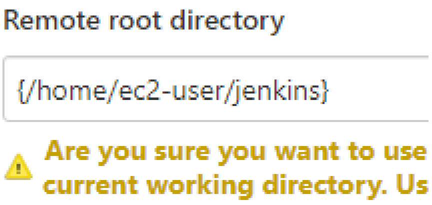
Ensure the number of executors is 2.



Number of executors

2

Enter {/home/ec2-user/jenkins} for remote root directory.

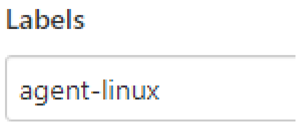


Remote root directory

{/home/ec2-user/jenkins}

⚠ Are you sure you want to use current working directory. Us

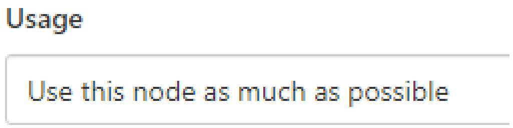
Then create a label and call it agent-linux.



Labels

agent-linux

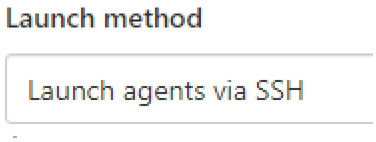
After select use this node as much as possible



Usage

Use this node as much as possible

Select the launch method “launch agent via SSH”



Launch method

Launch agents via SSH

For the Host, enter your private IP address of the agent.

Then, add SSH credentials (username: ec2-user | key: the private key you used to ssh into agent)

For Kind, select SSH Username with private key

For ID, enter worker-ssh and enter description - ssh into agent

For username, enter ec2-user

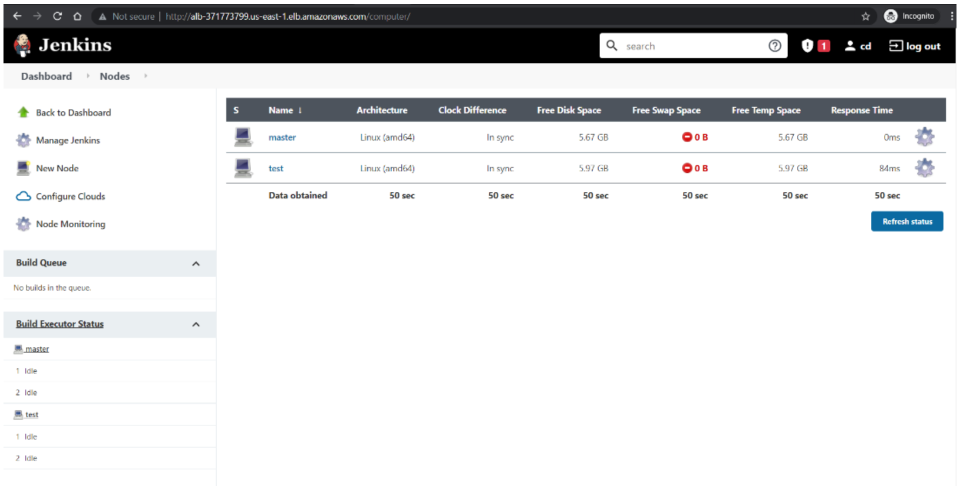
For the private key, enter your RSA key information directly into the box. No passphrase for the key and press Add.

Once the credentials are made, select it.

Then for your Hot Key Verification strategy, select non verifying verification strategy.

Save changes then look at the logs to see if the setup was successful.

Due to the error message in my logs, I use the commands ($ sudo yum install maven) and ($ sudo yum install git) which I forgot to include in my bootstrap when I redid it.

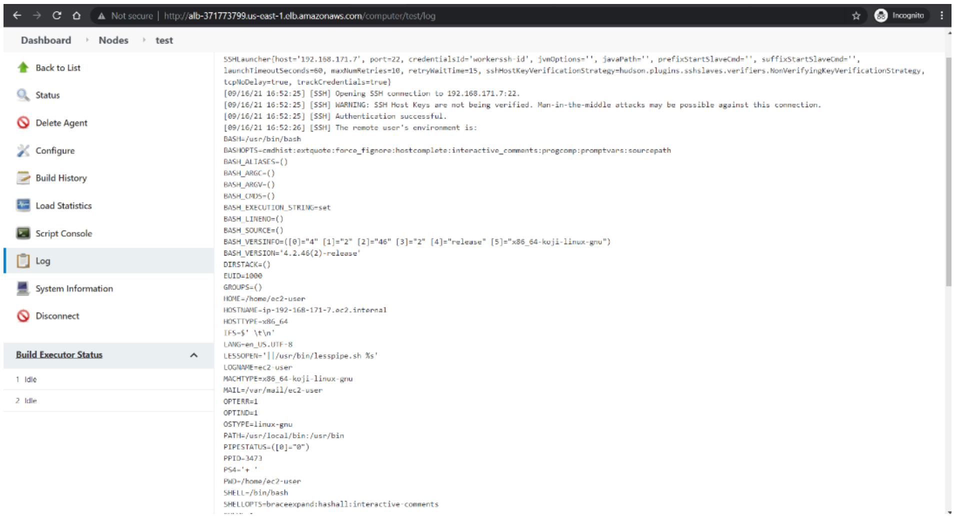


S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	5.67 GB	0 B	5.67 GB	0ms
	test	Linux (amd64)	In sync	5.97 GB	0 B	5.97 GB	84ms
	Data obtained	50 sec	50 sec	50 sec	50 sec	50 sec	50 sec

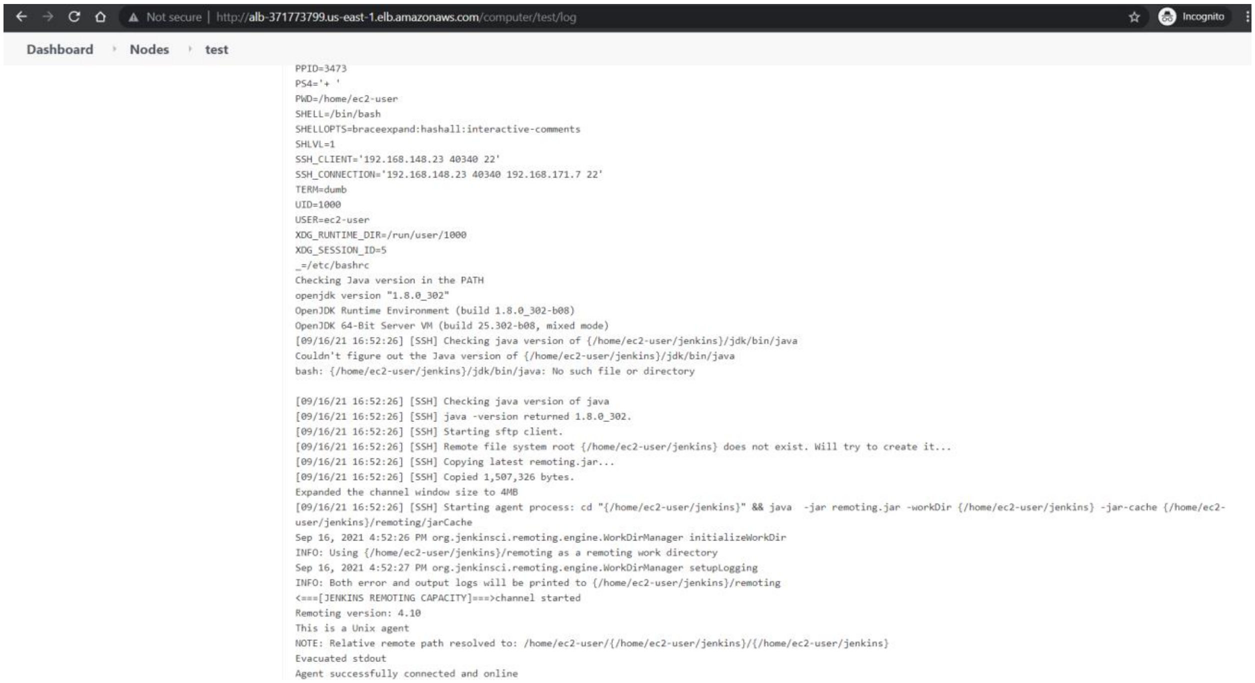
No builds in the queue.

1	Idle
2	Idle

1	Idle
2	Idle



```
SSHLauncher{host='192.168.171.7', port=22, credentialsId='workerssh-id', jvmOptions='', javaPath='', prefixStartSlaveCmd='', suffixStartSlaveCmd='',
launchTimeoutSeconds=60, maxNumRetries=10, retryWaitTime=15, sshHostKeyVerificationStrategy=hudson.plugins.sshslaves.verifiers.NonVerifyingKeyVerificationStrategy,
tcpNoDelay=true, trackCredentials=true}
[09/16/21 16:52:25] [SSH] Opening SSH connection to 192.168.171.7:22.
[09/16/21 16:52:25] [SSH] WARNING: SSH Host Keys are not being verified. Man-in-the-middle attacks may be possible against this connection.
[09/16/21 16:52:25] [SSH] Authentication successful.
[09/16/21 16:52:26] [SSH] The remote user's environment is:
BASH=/usr/bin/bash
BASHOPTS=cmdhist:extquote:force_fignore:hostcomplete:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=()
BASH_ARGV=()
BASH_CMDS=()
BASH_EXECUTION_STRING=set
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="4" [1]="2" [2]="46" [3]="2" [4]="release" [5]="x86_64-koji-linux-gnu")
BASH_VERSION='4.2.46(2)-release'
DIRSTACK=()
EUID=1000
GROUPS=()
HOME=/home/ec2-user
HOSTNAME=ip-192-168-171-7.ec2.internal
HOSTTYPE=x86_64
IFS=$' \t\n'
LANG=en_US.UTF-8
LESSOPEN='||/usr/bin/lesspipe.sh %s'
LOGNAME=ec2-user
MACHTYPE=x86_64-koji-linux-gnu
MAIL=/var/mail/ec2-user
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/local/bin:/usr/bin
PIPESTATUS=([0]="0")
PPID=3473
PS4='+ '
PWD=/home/ec2-user
SHELL=/bin/bash
SHELLOPTS=braceexpand:hashall:interactive-comments
```



Not secure | http://alb-371773799.us-east-1.elb.amazonaws.com/computer/test/log
Incognito
Dashboard
Nodes
test

```
PPID=3473
PS4='+ '
PWD=/home/ec2-user
SHELL=/bin/bash
SHELLOPTS=braceexpand:hashall:interactive-comments
SHLVL=1
SSH_CLIENT='192.168.148.23 40340 22'
SSH_CONNECTION='192.168.148.23 40340 192.168.171.7 22'
TERM=dumb
UID=1000
USER=ec2-user
XDG_RUNTIME_DIR=/run/user/1000
XDG_SESSION_ID=5
_=/etc/bashrc
Checking Java version in the PATH
openjdk version "1.8.0_302"
OpenJDK Runtime Environment (build 1.8.0_302-b08)
OpenJDK 64-Bit Server VM (build 25.302-b08, mixed mode)
[09/16/21 16:52:26] [SSH] Checking java version of {/home/ec2-user/jenkins}/jdk/bin/java
Couldn't figure out the Java version of {/home/ec2-user/jenkins}/jdk/bin/java
bash: {/home/ec2-user/jenkins}/jdk/bin/java: No such file or directory

[09/16/21 16:52:26] [SSH] Checking java version of java
[09/16/21 16:52:26] [SSH] java -version returned 1.8.0_302.
[09/16/21 16:52:26] [SSH] Starting sftp client.
[09/16/21 16:52:26] [SSH] Remote file system root {/home/ec2-user/jenkins} does not exist. Will try to create it...
[09/16/21 16:52:26] [SSH] Copying latest remoting.jar...
[09/16/21 16:52:26] [SSH] Copied 1,507,326 bytes.
Expanded the channel window size to 4MB
[09/16/21 16:52:26] [SSH] Starting agent process: cd "{/home/ec2-user/jenkins}" && java  -jar remoting.jar -workDir {/home/ec2-user/jenkins} -jar-cache {/home/ec2-
user/jenkins}/remoting/jarCache
Sep 16, 2021 4:52:26 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using {/home/ec2-user/jenkins}/remoting as a remoting work directory
Sep 16, 2021 4:52:27 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to {/home/ec2-user/jenkins}/remoting
<===[JENKINS REMOTING CAPACITY]===>channel started
Remoting version: 4.10
This is a Unix agent
NOTE: Relative remote path resolved to: /home/ec2-user/{/home/ec2-user/jenkins}/{/home/ec2-user/jenkins}
Evacuated stdout
Agent successfully connected and online
```