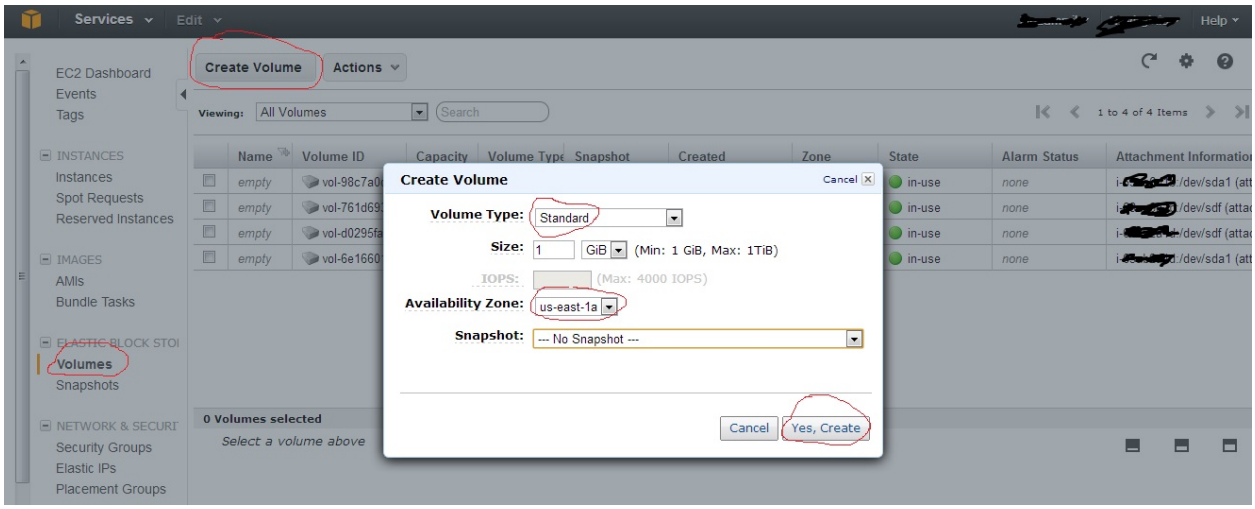
In this solution we use tomcat instead of other web/application servers

1) Remote login to your EC2 instance and execute the following command.
   sudo apt-get install tomcat7 (This installs tomcat)

2) Create and attach EBS volume from web portal dashboard. EBS should be in same AZ as the EC2 instance. Use the snap shot below for reference.



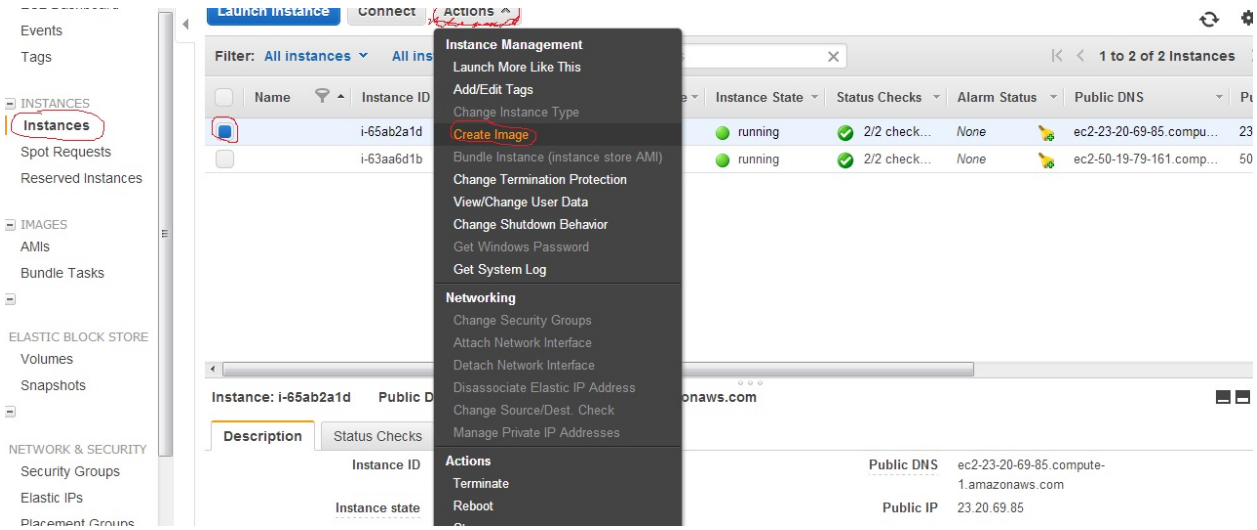
3) After creation goto Actions and attach the EBS to the EC2 instance.
4) Again on the remote login session execute the following commands:
   lsblk (this command to list volumes)
   sudo mkfs -t ext3 /dev/xvdf (where /dev/xvdf is the new volume name as listed with lsblk cmd)
   sudo mkdir /mnt/abc (create new directory /mnt/abc to mount the new volume)
   sudo mount /dev/xvdf /mnt/abc (this command mounts the volume in the newly created dir)
   sudo mkdir /mnt/abc/tomcat-deploy (create new directory to place web app files)
   sudo cp –r /var/lib/tomcat7/webapps/ROOT /mnt/abc/tomcat-deploy
5) Edit tomcat configuration files to change deployment dir from webapps to /mnt/abc/tomcat-deploy. Open the file using 'vi' (vi /etc/tomcat7/server.xml) and make the below change:

<Host name="localhost" appBase="**/mnt/abc/tomcat-deploy**"

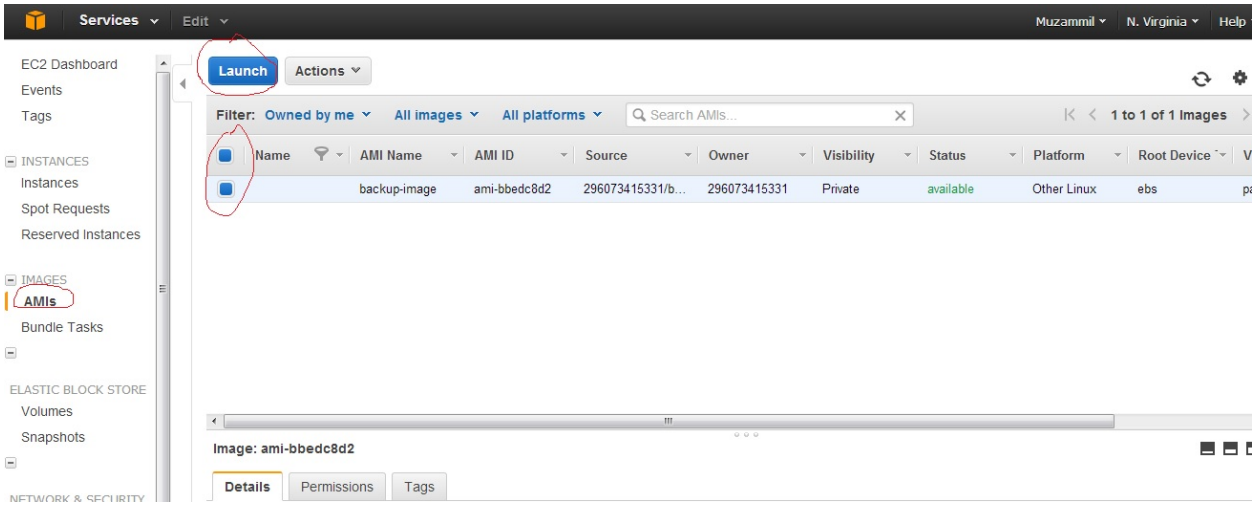
Open the file 'vi /mnt/abc/tomcat-deploy/ROOT/index.html' and put some text that identifies this instance when this web page will be loaded.

6) sudo service tomcat7 restart (Restart tomcat)
7) Create image of EC2 instance (EBS snapshot will be automatically created)

Use the below snapshot for reference.

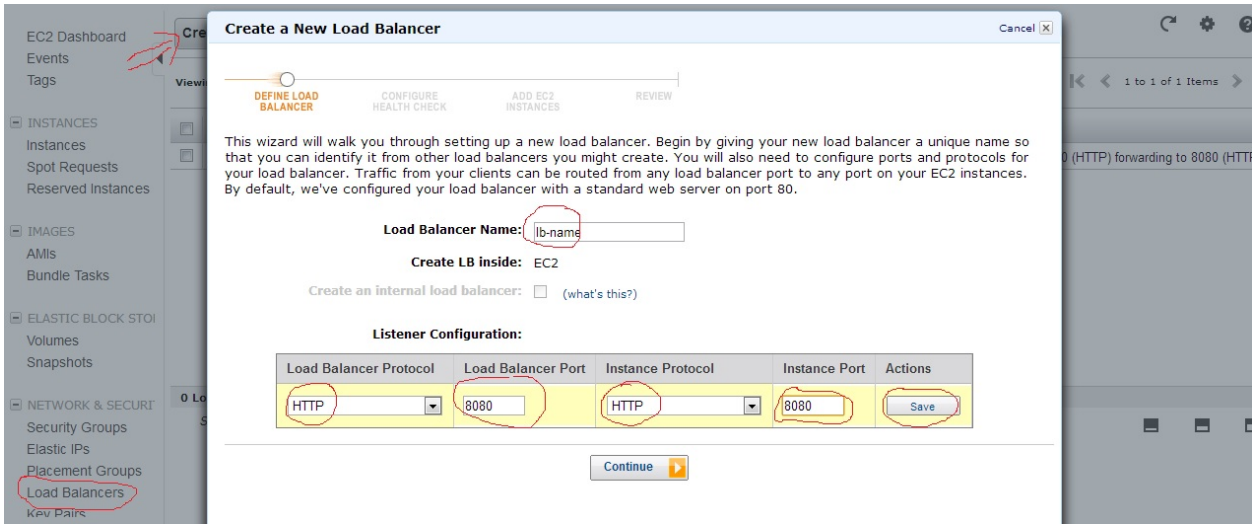


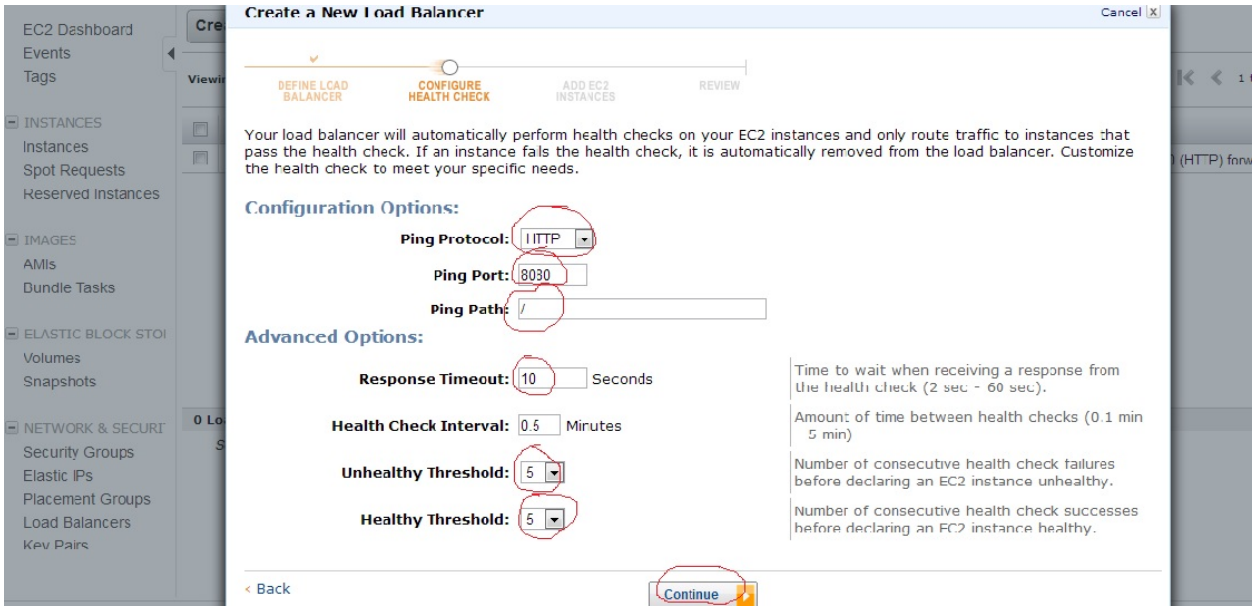
8) Goto IMAGES -> AMIs and launch new instance with AMI in different zone.
Use the below snapshots for reference.

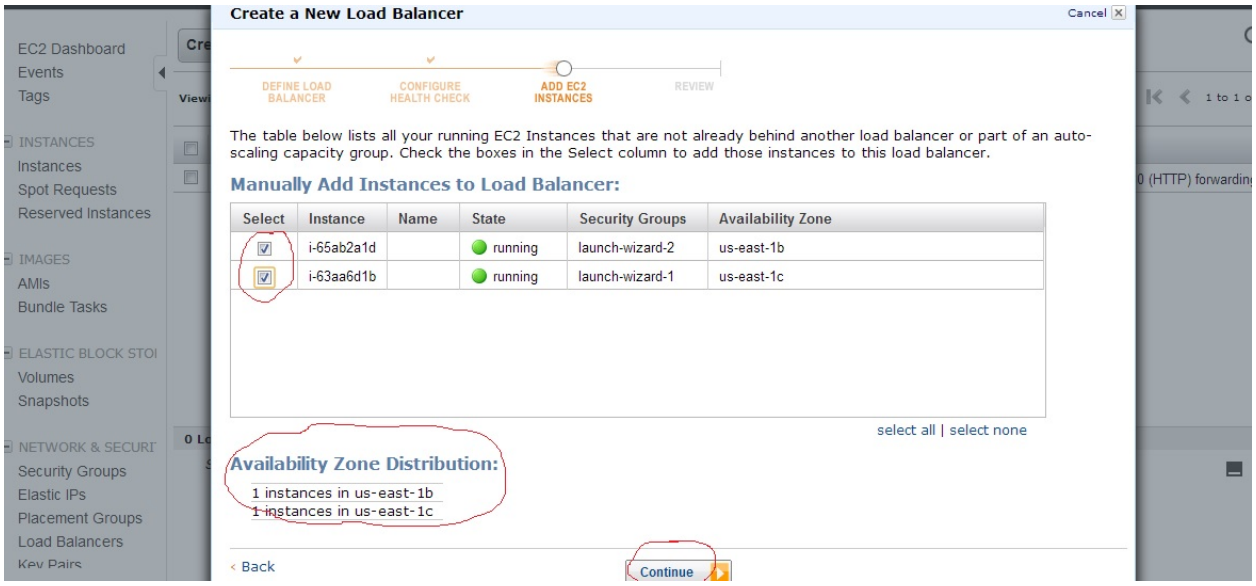


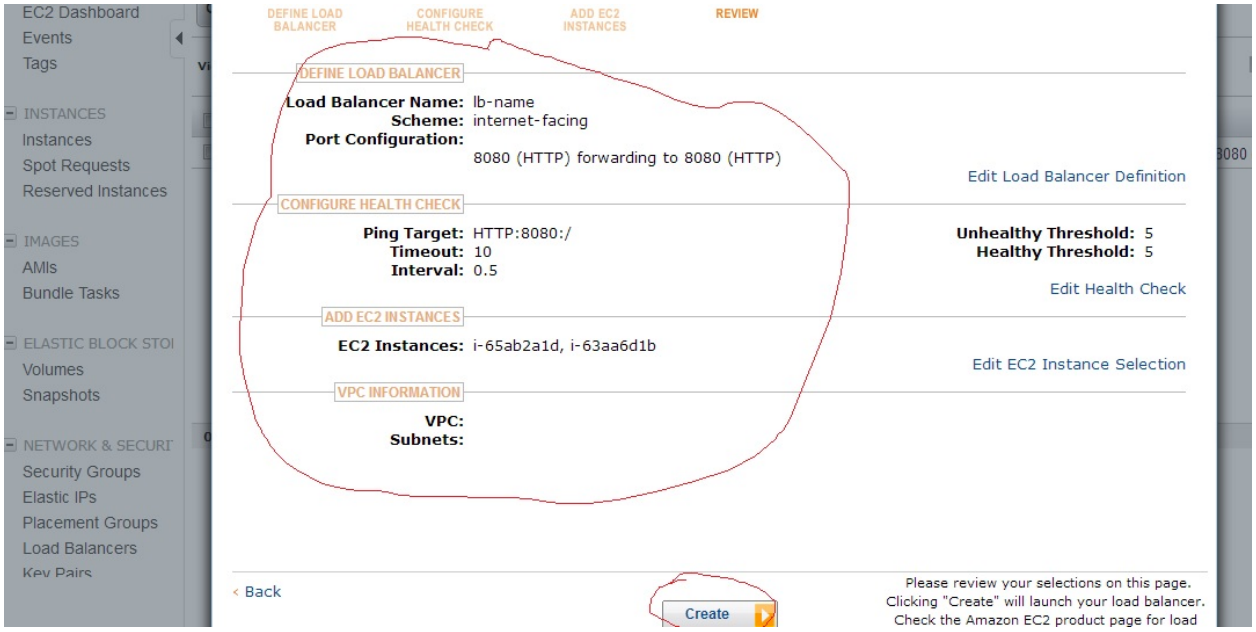
9) Repeat tomcat EBS mounting related and tomcat related steps for this new EC2 instance.

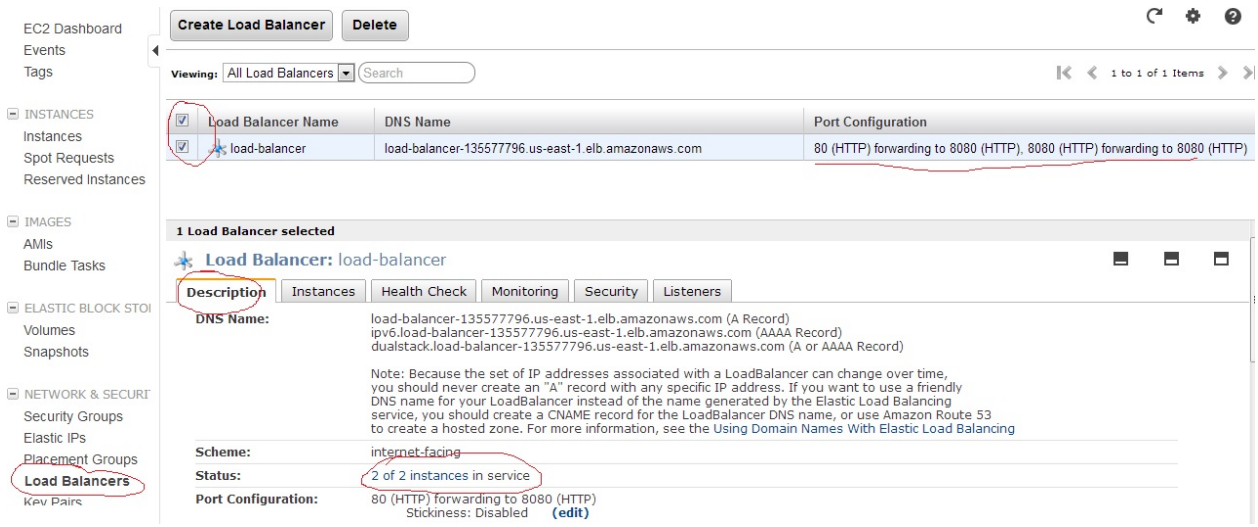
10) Create Load balancer. Use snapshots below for reference.






Create a New Load Balancer
Cancel
DEFINE LOAD BALANCER
CONFIGURE HEALTH CHECK
ADD EC2 INSTANCES
REVIEW
The table below lists all your running EC2 Instances that are not already behind another load balancer or part of an auto-scaling capacity group. Check the boxes in the Select column to add those instances to this load balancer.
Manually Add Instances to Load Balancer:
Select
Instance
Name
State
Security Groups
Availability Zone
i-65ab2a1d
running
launch-wizard-2
us-east-1b
i-63aa6d1b
running
launch-wizard-1
us-east-1c
select all | select none
Availability Zone Distribution:
1 instances in us-east-1b
1 instances in us-east-1c
Back
Continue
EC2 Dashboard
Events
Tags
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STOI
Volumes
Snapshots
NETWORK & SECURI
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs


DEFINE LOAD BALANCER
CONFIGURE HEALTH CHECK
ADD EC2 INSTANCES
REVIEW
DEFINE LOAD BALANCER
Load Balancer Name: lb-name
Scheme: internet-facing
Port Configuration:
8080 (HTTP) forwarding to 8080 (HTTP)
Edit Load Balancer Definition
CONFIGURE HEALTH CHECK
Ping Target: HTTP:8080:/
Timeout: 10
Interval: 0.5
Unhealthy Threshold: 5
Healthy Threshold: 5
Edit Health Check
ADD EC2 INSTANCES
EC2 Instances: i-65ab2a1d, i-63aa6d1b
Edit EC2 Instance Selection
VPC INFORMATION
VPC:
Subnets:
Back
Create
Please review your selections on this page.
Clicking "Create" will launch your load balancer.
Check the Amazon EC2 product page for load
EC2 Dashboard
Events
Tags
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STOI
Volumes
Snapshots
NETWORK & SECURI
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs



After adding the instance in the load balancer it takes some time to check the health of the instances attached. Once the setup is ready you see Status as 2 of 2 instances in service(shown in the last snapshot)

To test the setup use the public DNS Name of load balancer and make a request typing the following in the browser :
http://<public dns name>:8080
Refresh the page multiple times or use different browser for multiple requests.
You should now be able to differentiate the serving instance based on instance specific text you put in the index.html file