Practical Exercise: Introduction to load balancing a simple website

One of the major benefits of cloud computing is the ability to grow and shrink the capacity to service user demand efficiently as this demand varies over time – hence the term "elastic" computing. An additional benefit is that this capacity can be spread across multiple physical locations to provide increased reliability. This is usually achieved by replicating resources and balancing the load between these replicas.

In this lab we will install a web server on a "master" instance and configure a **load balancer** to direct traffic to multiple replicas of this. For this initial exercise, these replicas will be created manually rather than with auto-scaling.

Part A. "Master" instance and AMI (this part is the same as in auto scaling exercise)

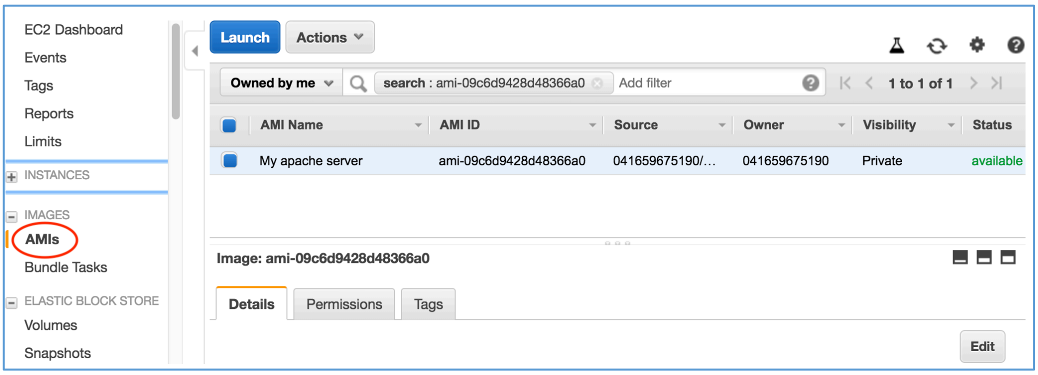
1. Launch a standard Amazon Linux 2 instance.
   - Choose t2.micro in step 1
   - Tag the instance with a suitable name in step 3 – e.g. "Master web server"
   - Ensure the security group chosen in step 6 allows inbound SSH and HTTP.
2. Connect to your EC2 instance with SSH/PuTTY.
3. Install the apache web server (httpd), enable at startup, and start service.

   ```
   sudo yum update -y               # ensures OS patches are applied
   sudo yum install httpd -y        # install server
   sudo systemctl enable httpd      # start server at boot
   sudo service httpd start         # start server now
   ```

   You should now be able to browse to the public IP address or the public DNS name of your instance and see the apache default page.
4. Create a new default page:

   ```
   sudo nano /var/www/html/index.html        # editor to add custom content (CTRL-x to quit)
   ```

   Any content is fine here, even "Hello World". Refresh the page in your browser to check that it worked.
5. Stop the instance. When it has stopped, select **Actions** -> **Image** -> **Create Image**
6. Give the image (AMI) a name and description. It takes a couple of minutes for the image to become available. You can check the status by clicking on AMIs on the left side menu in the EC2 console.



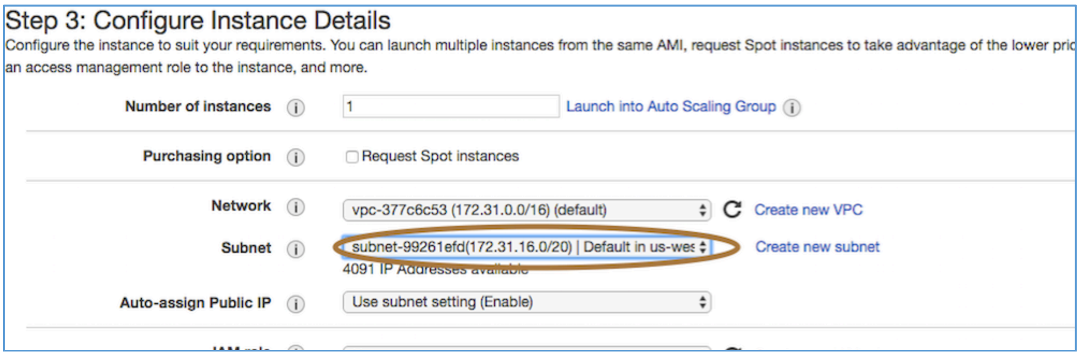
AMI Name	AMI ID	Source	Owner	Visibility	Status
My apache server	ami-09c6d9428d48366a0	041659675190/...	041659675190	Private	available

Image: ami-09c6d9428d48366a0

Part B. Load balancing

The steps below are similar to those detailed at:
http://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/load-balancer-getting-started.html

7. Launch three instances based on your AMI, one for each of the three Availability Zones in the Ireland region. Do this by selecting the AMI and clicking *Launch*.
   - Leave as t2.micro in Step 2
   - In Step 3: Configure Instance Details, change the Subnet setting from "No preference" to your chosen subnet and a different one in each case (options are eu-west-1a, eu-west-1b, eu-west-1c). Select an appropriate VPC (e.g. default).



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 pric an access management role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-377c6c53 (172.31.0.0/16) (default)

Subnet: subnet-99261efd(172.31.16.0/20) | Default in us-wes

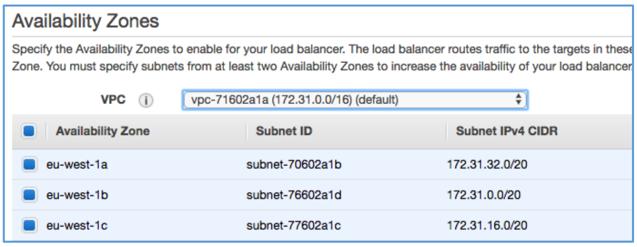
4091 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

- In Step 5, I suggest tagging your instances with a Name of "a", "b", or "c" to reflect the availability zone subnet.
- Ensure the security group chosen allows inbound HTTP and SSH traffic
- Log into each instance separately and make a small edit to the home page to help identify it – i.e. zone a, b, c.

8. Go to *Load Balancers* in the EC2 dashboard and select *Create Load Balancer* and *Create* under Application Load Balancer.

9. In Step 1 give your load balancer a name and under *Availability Zones* select the correct VPC (e.g. default).The default listener (port 80) is ok. Under Availability Zones, select all three subnets.



Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer

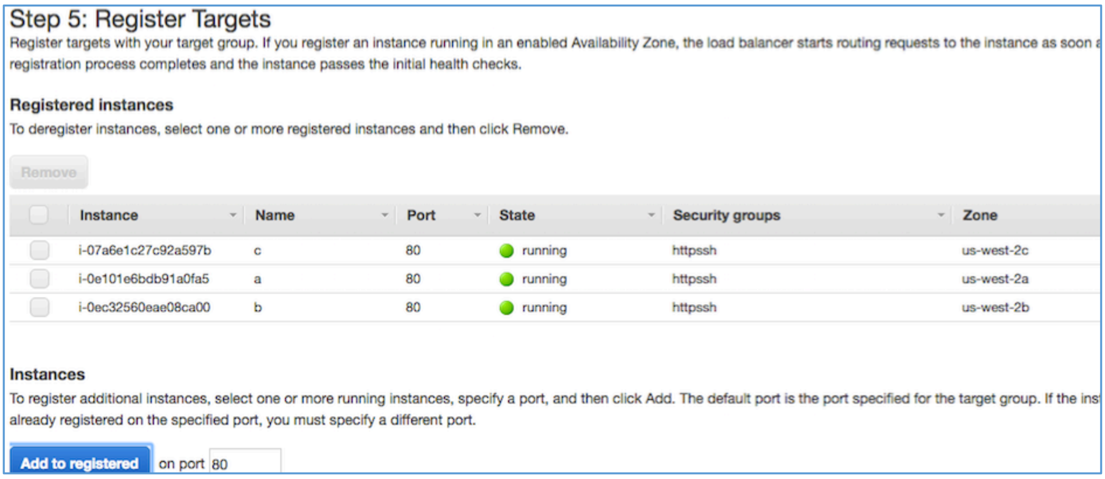
VPC: vpc-71602a1a (172.31.0.0/16) (default)

☑	Availability Zone	Subnet ID	Subnet IPv4 CIDR
☑	eu-west-1a	subnet-70602a1b	172.31.32.0/20
☑	eu-west-1b	subnet-76602a1d	172.31.0.0/20
☑	eu-west-1c	subnet-77602a1c	172.31.16.0/20

10. Skip Step 2. In Step 3 select a security group that allows inbound HTTP (port 80)

11. In Step 4, give your target group a name and leave the defaults for everything else.

12. Step 5 allows you to register each of your instances with the load balancer. Select your instances and “Add to registered”.



Step 5: Register Targets

Register targets with your target group. If you register an instance running in an enabled Availability Zone, the load balancer starts routing requests to the instance as soon a registration process completes and the instance passes the initial health checks.

Registered instances

To deregister instances, select one or more registered instances and then click Remove.

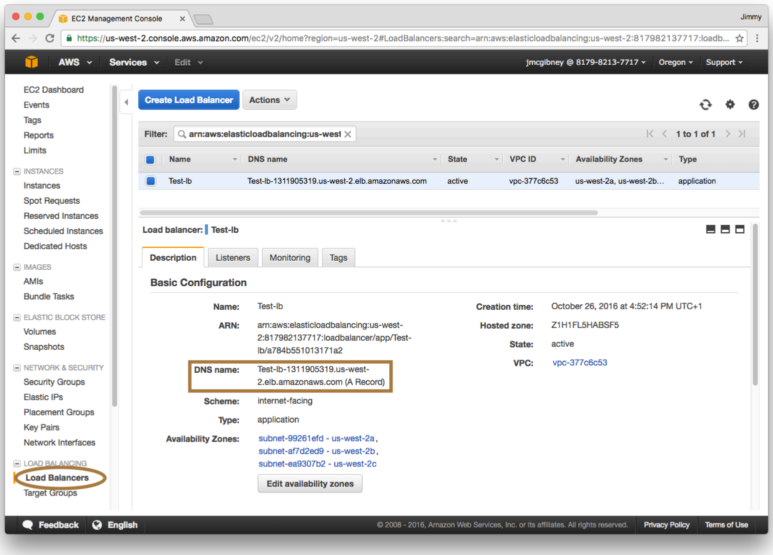
□	Instance	Name	Port	State	Security groups	Zone
□	i-07a6e1c27c92a597b	c	80	● running	httpssh	us-west-2c
□	i-0e101e6bdb91a0fa5	a	80	● running	httpssh	us-west-2a
□	i-0ec32560eae08ca00	b	80	● running	httpssh	us-west-2b

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the ins already registered on the specified port, you must specify a different port.

on port 80

13. After a few minutes you should be able to test your load balancer with a browser by navigating to: [http://*load-balancer-dns-name*](http://load-balancer-dns-name). This DNS name can be found on the main Load Balancers view:



https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#LoadBalancers:search=arn:aws:elasticloadbalancing:us-west-2:817982137717:loadb...

Filter: arn:aws:elasticloadbalancing:us-west

1 to 1 of 1

Name	DNS name	State	VPC ID	Availability Zones	Type
Test-lb	Test-lb-1311905319.us-west-2.elb.amazonaws.com	active	vpc-377c6c53	us-west-2a, us-west-2b...	application

Load balancer: Test-lb

Basic Configuration

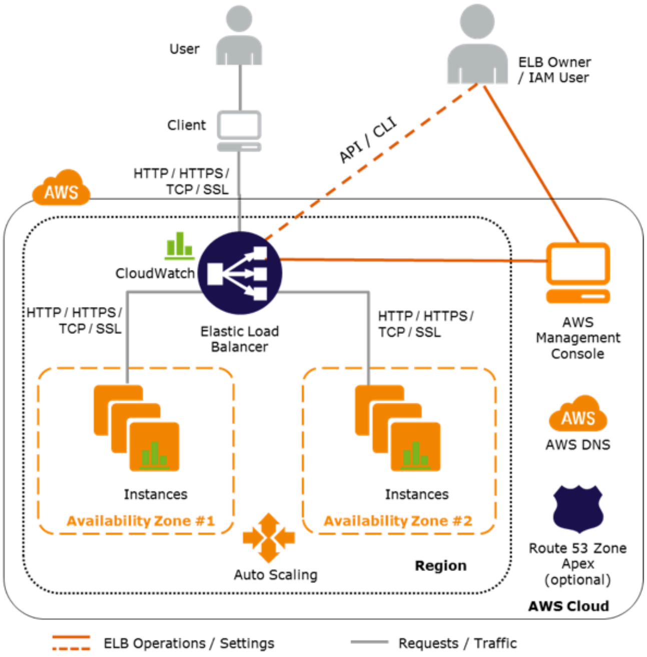
Name:	Test-lb	Creation time:	October 26, 2016 at 4:52:14 PM UTC+1
ARN:	arn:aws:elasticloadbalancing:us-west-2:817982137717:loadbalancer/app/Test-lb/a784b551013171a2	Hosted zone:	Z1H1FL5HABSF5
DNS name:	Test-lb-1311905319.us-west-2.elb.amazonaws.com (A Record)	State:	active
Scheme:	internet-facing	VPC:	vpc-377c6c53
Type:	application		
Availability Zones:	subnet-99261efd - us-west-2a, subnet-af7d2ed9 - us-west-2b, subnet-ea9307b2 - us-west-2c		

14. **Exercise**: Examine the apache log files on each of your instances and try to trace a particular refresh of http://*load-balancer-dns-name* to find out which instance handled that particular request. You can find apache logs in **/var/log/httpd/**. A useful command for this is:

```
sudo tail -f /var/log/httpd/access_log
```

15. Delete your load balancer when you are finished testing it rather than leave it there indefinitely. It is easy to re-create it. Application load balancers are charged at 2.52 cents per hour which works out a little over $4 per week. Note there is no need to keep the master instance that you created in Part A. This can be re-created any time from the AMI.

Appendix: AWS Load Balancing Architecture



AWS Load Balancing Architecture