lab

lab title

Highly Available and Fault Tolerant Architecture for Web Applications inside a VPC V1.02

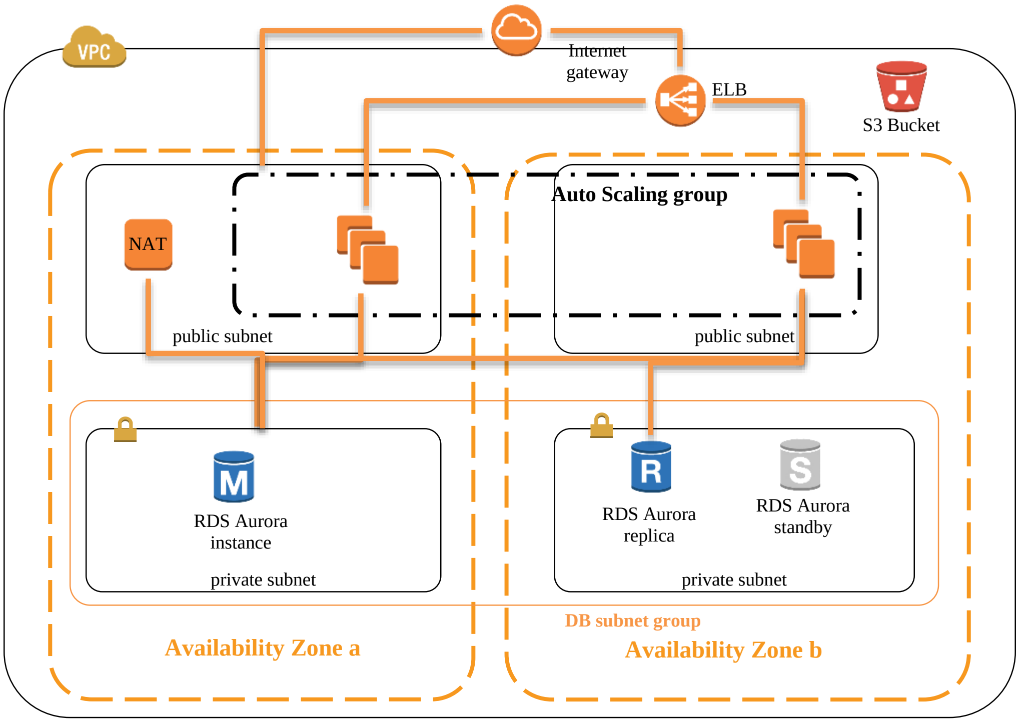
Course title

AWS Certified Solutions Architect Associate

Table of Contents

Contents

About the Lab



These lab notes are to support the instructional videos on AWS VPC architecture in the BackSpace AWS Certified Solutions Architect Associate course.

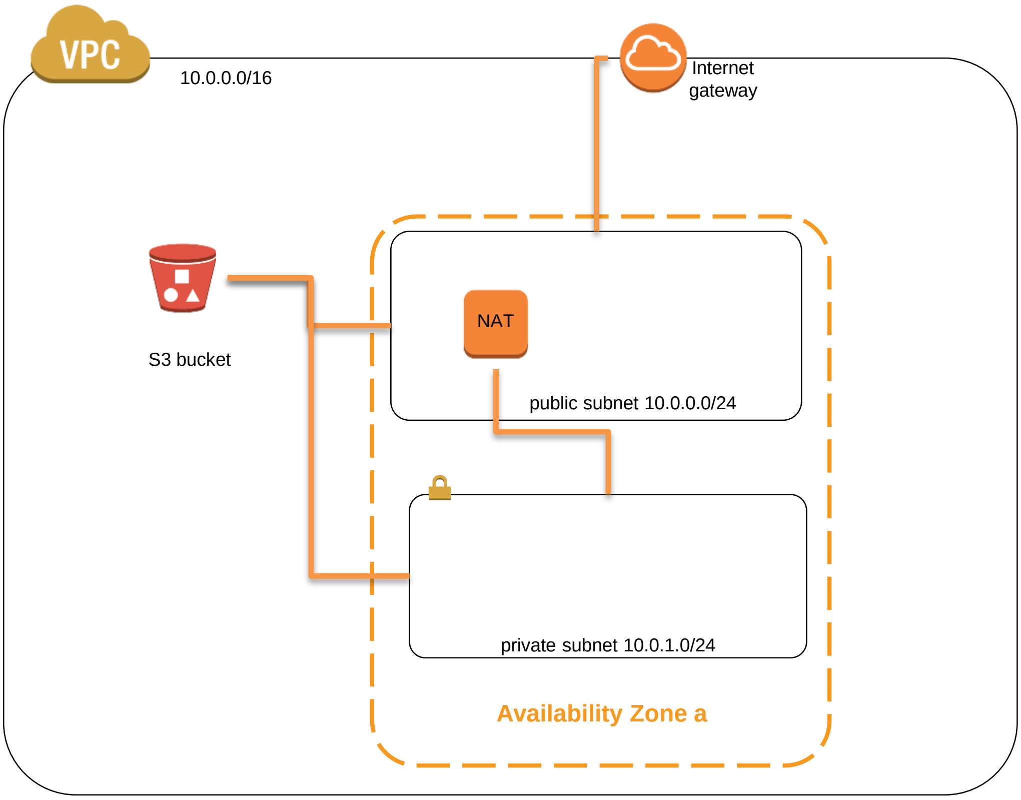
This lab is the culmination of many aspects of AWS Architecure that you have learnt throughout the course. The focus will be on ensuring all the concepts essential for certification are clearly understood.

The architecture we will developing detailed in the diagram below is typical for a web application such as WordPress site.

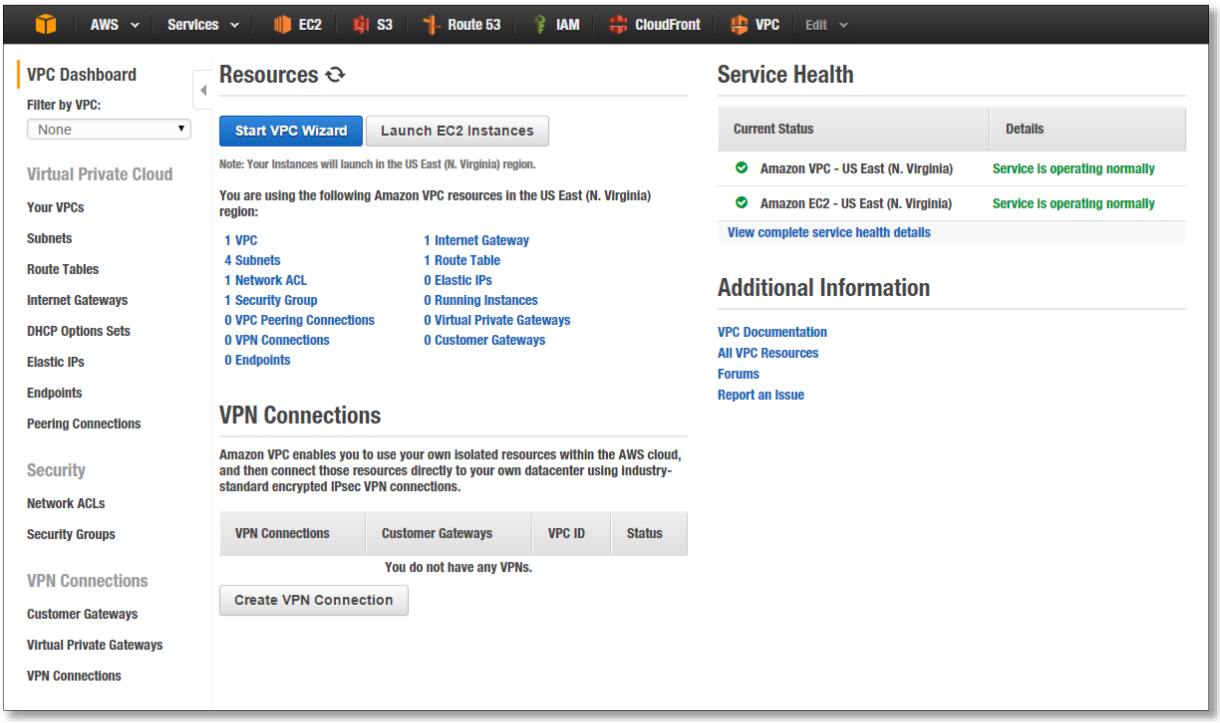
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

Creating a VPC with Public and Private Subnets and a NAT Instance

In this section we will use the VPC Wizard to create a VPC with public and private subnets and, a NAT instance to allow instances in the private subnet to download updates from the Internet. We will then look at improving the security of the VPC.



Make sure you are in US-East region and select the VPC console



VPC Dashboard

Filter by VPC:

None

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Peering Connections

Security

Network ACLs

Security Groups

VPN Connections

Customer Gateways

Virtual Private Gateways

VPN Connections

Resources

Start VPC Wizard | Launch EC2 Instances

Note: Your instances will launch in the US East (N. Virginia) region.

You are using the following Amazon VPC resources in the US East (N. Virginia) region:

1 VPC	1 Internet Gateway
4 Subnets	1 Route Table
1 Network ACL	0 Elastic IPs
1 Security Group	0 Running Instances
0 VPC Peering Connections	0 Virtual Private Gateways
0 VPN Connections	0 Customer Gateways
0 Endpoints	

VPN Connections

Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.

VPN Connections	Customer Gateways	VPC ID	Status

You do not have any VPNs.

Create VPN Connection

Service Health

Current Status	Details
✓ Amazon VPC - US East (N. Virginia)	Service is operating normally
✓ Amazon EC2 - US East (N. Virginia)	Service is operating normally

View complete service health details

Additional Information

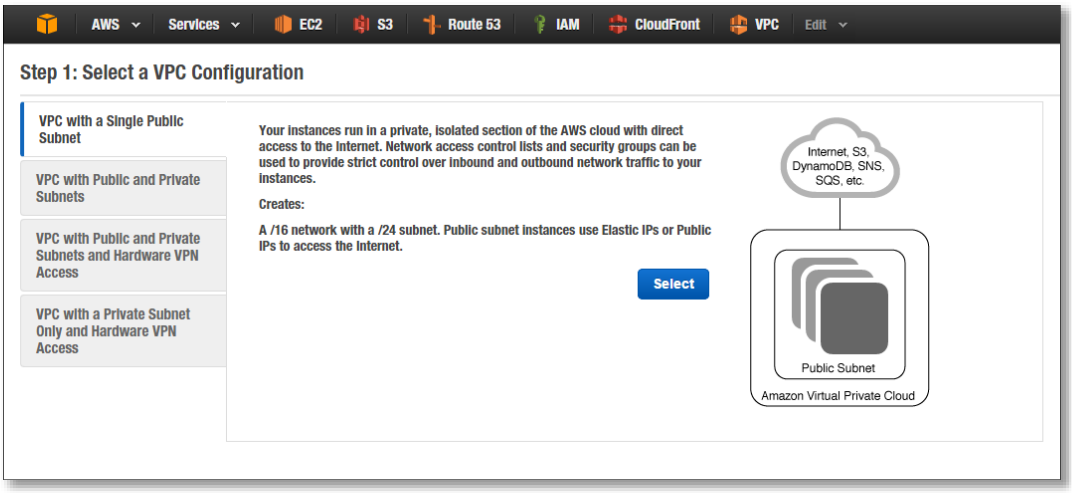
VPC Documentation

All VPC Resources

Forums

Report an Issue

Click “Start VPC Wizard”


AWS
Services
EC2
S3
Route 53
IAM
CloudFront
VPC
Edit
Internet, S3, DynamoDB, SNS, SQS, etc.
Public Subnet
Amazon Virtual Private Cloud

Step 1: Select a VPC Configuration

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

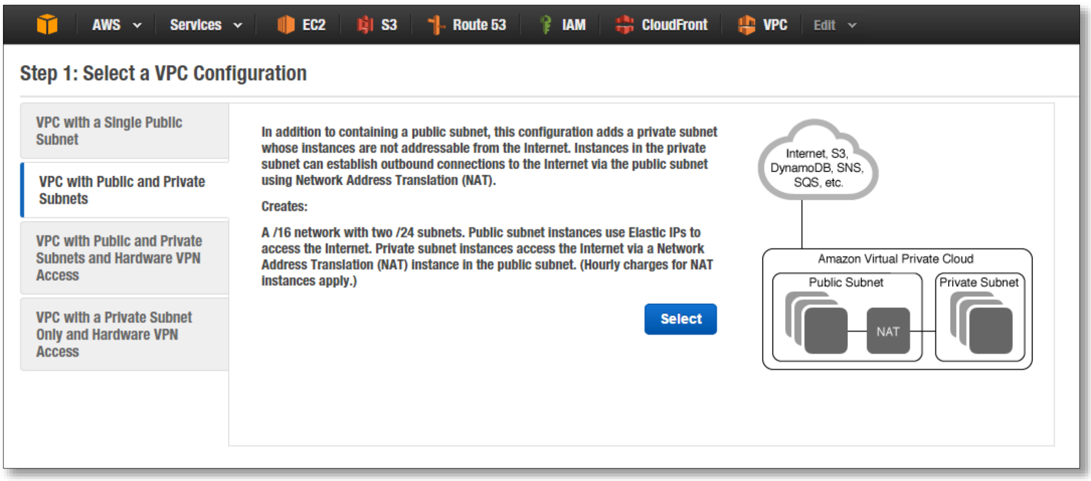
Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

Creates:

A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

Select

Select “VPC with Public and Private Subnets”



Step 1: Select a VPC Configuration

- VPC with a Single Public Subnet
- VPC with Public and Private Subnets
- VPC with Public and Private Subnets and Hardware VPN Access
- VPC with a Private Subnet Only and Hardware VPN Access

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

Creates:

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply.)

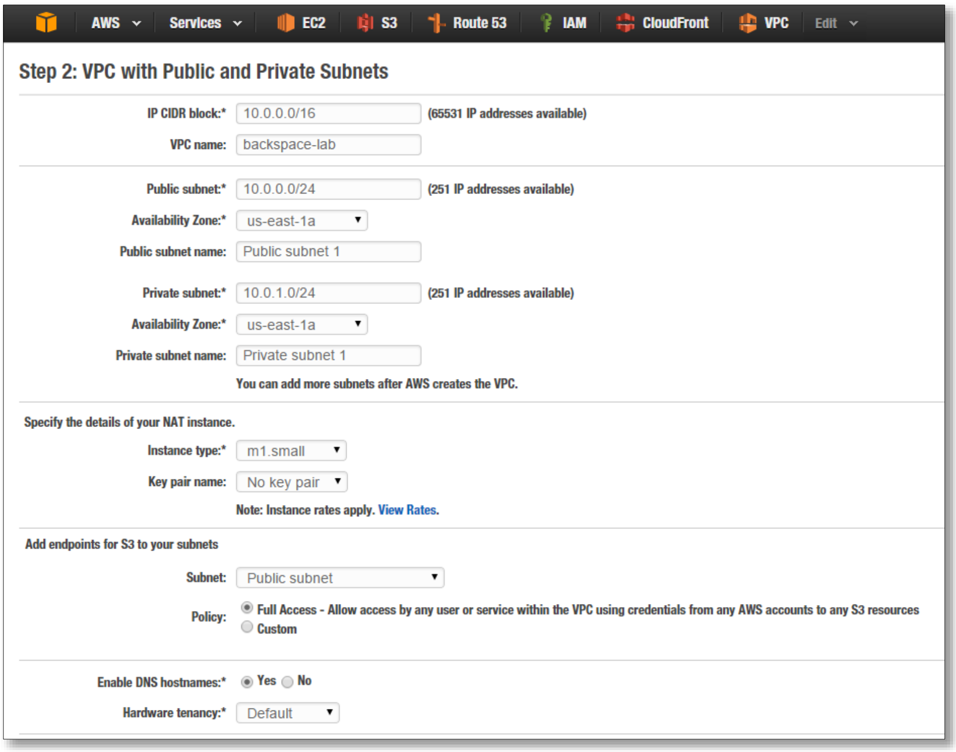
Give the VPC a name.

Select us-east-1a for both subnets.

Call the subnets Public subnet 1 and Private subnet 1

Select “Public and Private” subnet for “Add endpoints for S3 to your subnets”. This allows instances in your subnets to directly access an S3 bucket in the same region. This can be used to bootstrap instances with latest code from a Git repository.

Click “Create VPC”



Step 2: VPC with Public and Private Subnets

IP CIDR block:*	10.0.0.0/16 (65531 IP addresses available)
VPC name:	backspace-lab
Public subnet:*	10.0.0.0/24 (251 IP addresses available)
Availability Zone:*	us-east-1a
Public subnet name:	Public subnet 1
Private subnet:*	10.0.1.0/24 (251 IP addresses available)
Availability Zone:*	us-east-1a
Private subnet name:	Private subnet 1

You can add more subnets after AWS creates the VPC.

Specify the details of your NAT instance.

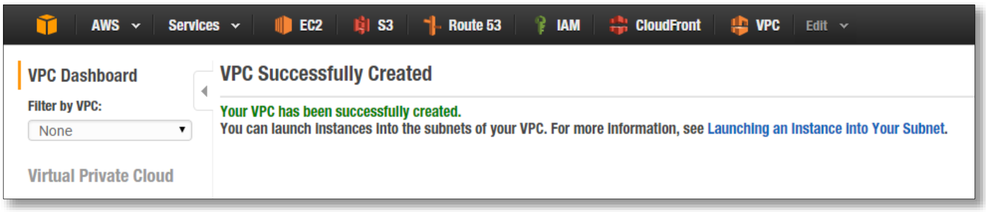
Instance type:*	m1.small
Key pair name:	No key pair

Note: Instance rates apply. View Rates.

Add endpoints for S3 to your subnets

Subnet:	Public subnet
Policy:	● Full Access - Allow access by any user or service within the VPC using credentials from any AWS accounts to any S3 resources ○ Custom
Enable DNS hostnames:*	● Yes ○ No
Hardware tenancy:*	Default

You should eventually get the success screen



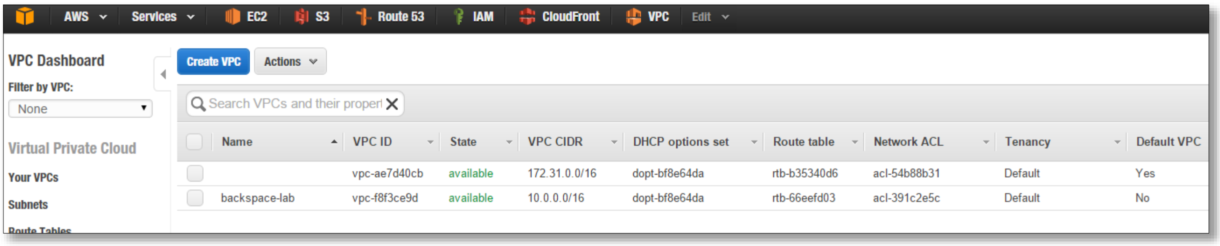
VPC Successfully Created

Your VPC has been successfully created.
You can launch instances into the subnets of your VPC. For more information, see Launching an Instance into Your Subnet.

Click OK

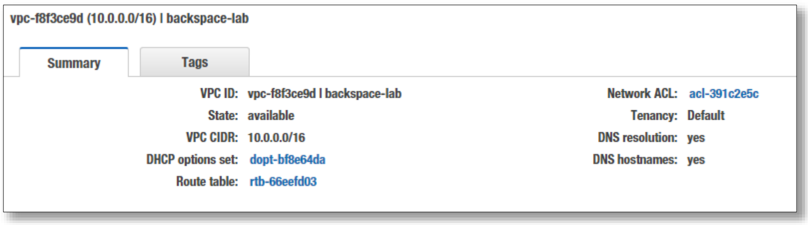
Click on “Your VPCs”.

Here you see the default VPC and the new VPC.



Name	VPC ID	State	VPC CIDR	DHCP options set	Route table	Network ACL	Tenancy	Default VPC
	vpc-ae7d40cb	available	172.31.0.0/16	dopt-bf8e64da	rtb-b35340d6	acl-54b88b31	Default	Yes
backspace-lab	vpc-f8f3ce9d	available	10.0.0.0/16	dopt-bf8e64da	rtb-66eefd03	acl-391c2e5c	Default	No

Click on the newly created VPC to see its details



vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

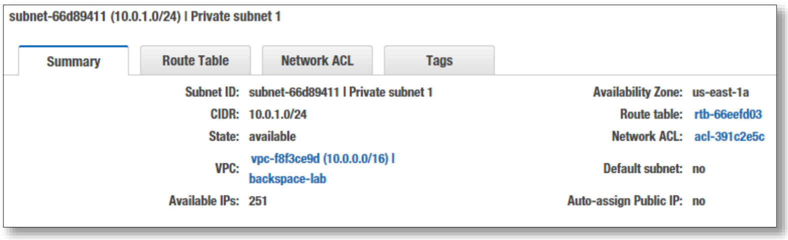
VPC ID:	vpc-f8f3ce9d | backspace-lab	Network ACL:	acl-391c2e5c
State:	available	Tenancy:	Default
VPC CIDR:	10.0.0.0/16	DNS resolution:	yes
DHCP options set:	dopt-bf8e64da	DNS hostnames:	yes
Route table:	rtb-66eefd03		

Here you can see a network ACL and Route Table has been created and associated to the VPC.

The Route Table is the “Main Route Table” and is implicitly associated to a subnet where no explicit association has been created.

Click on subnets to see Private subnet 1 and Public subnet 1.

Click on Private subnet 1 to see its details

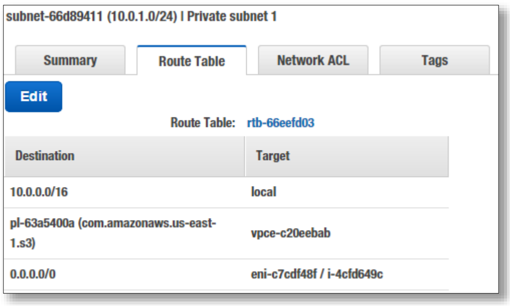


subnet-66d89411 (10.0.1.0/24) | Private subnet 1

Subnet ID: subnet-66d89411 | Private subnet 1
CIDR: 10.0.1.0/24
State: available
VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab
Available IPs: 251
Availability Zone: us-east-1a
Route table: rtb-66eefd03
Network ACL: acl-391c2e5c
Default subnet: no
Auto-assign Public IP: no

Here you can see the Main Route table has been implicitly associated with the subnet.

Click on the Route Table tab.



subnet-66d89411 (10.0.1.0/24) | Private subnet 1

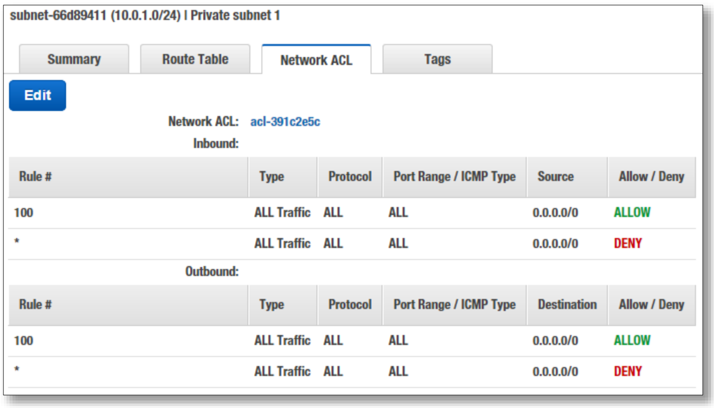
Route Table: rtb-66eefd03

Destination	Target
10.0.0.0/16	local
pl-63a5400a (com.amazonaws.us-east-1.s3)	vpce-c20eebab
0.0.0.0/0	eni-c7cdf48f / i-4cfd649c

The following routes have been created by the VPC Wizard

1. Route for local VPC traffic.
2. Destination the S3 service with target the VPC endpoint for the S3 service.
3. Destination all other traffic with target the ENI of the NAT instance.

Click on the Network ACL tab.



subnet-66d89411 (10.0.1.0/24) | Private subnet 1

Network ACL: acl-391c2e5c

Inbound:

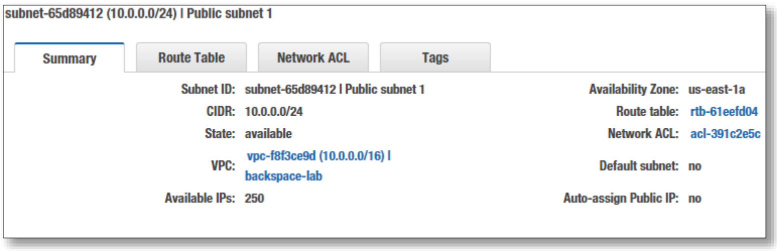
Rule #	Type	Protocol	Port Range / ICMP Type	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Outbound:

Rule #	Type	Protocol	Port Range / ICMP Type	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Here an NACL has been defined with an explicit allow for inbound and outbound traffic. We will be tightening this up later when we add our second availability zone by creating public and private subnet ACLs that restrict port access.

Now click on Public Subnet 1 to see its details.

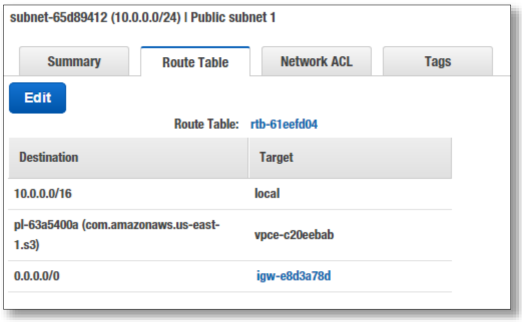


subnet-65d89412 (10.0.0.0/24) | Public subnet 1

Subnet ID:	subnet-65d89412 | Public subnet 1	Availability Zone:	us-east-1a
CIDR:	10.0.0.0/24	Route table:	rtb-61eefd04
State:	available	Network ACL:	acl-391c2e5c
VPC:	vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab	Default subnet:	no
Available IPs:	250	Auto-assign Public IP:	no

Here you can see that the same NACL has been defined but a different Route Table has been created by the VPC Wizard.

Click on the Route Table tab.



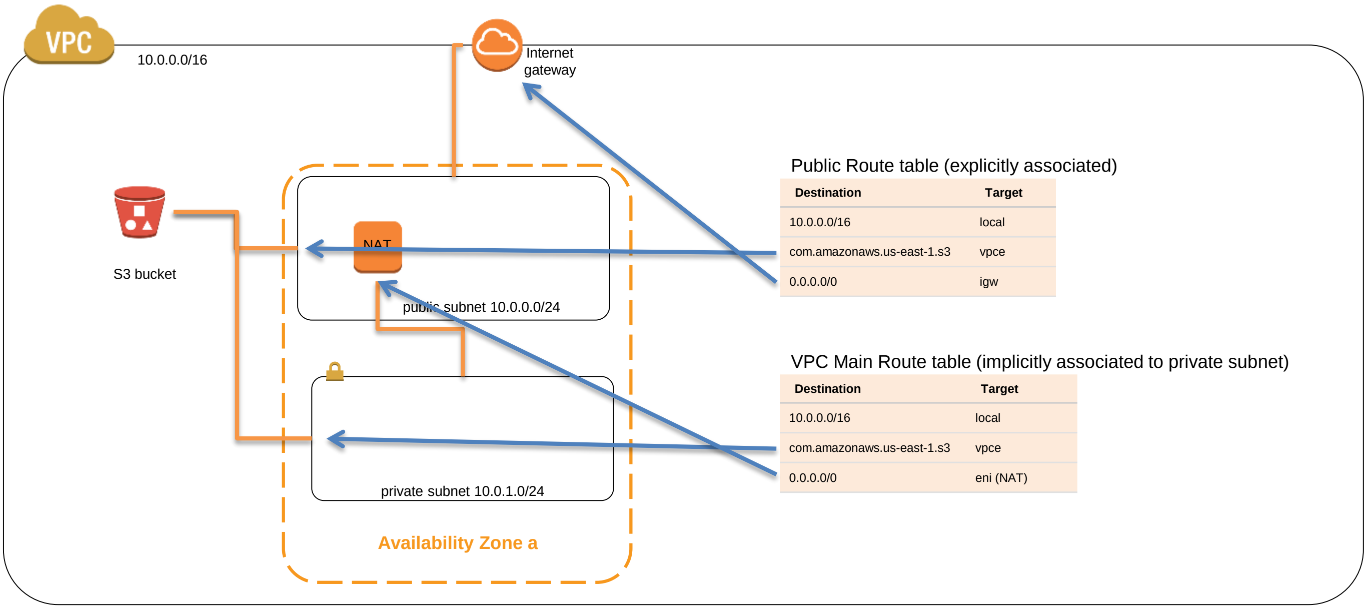
subnet-65d89412 (10.0.0.0/24) | Public subnet 1

Route Table: rtb-61eefd04

Destination	Target
10.0.0.0/16	local
pl-63a5400a (com.amazonaws.us-east-1.s3)	vpce-c20eebab
0.0.0.0/0	igw-e8d3a78d

The following routes have been created by the VPC Wizard

1. Route for local VPC traffic.
2. Destination the S3 service with target the VPC endpoint for the S3 service.
3. Destination all other traffic with target the VPC Internet Gateway.



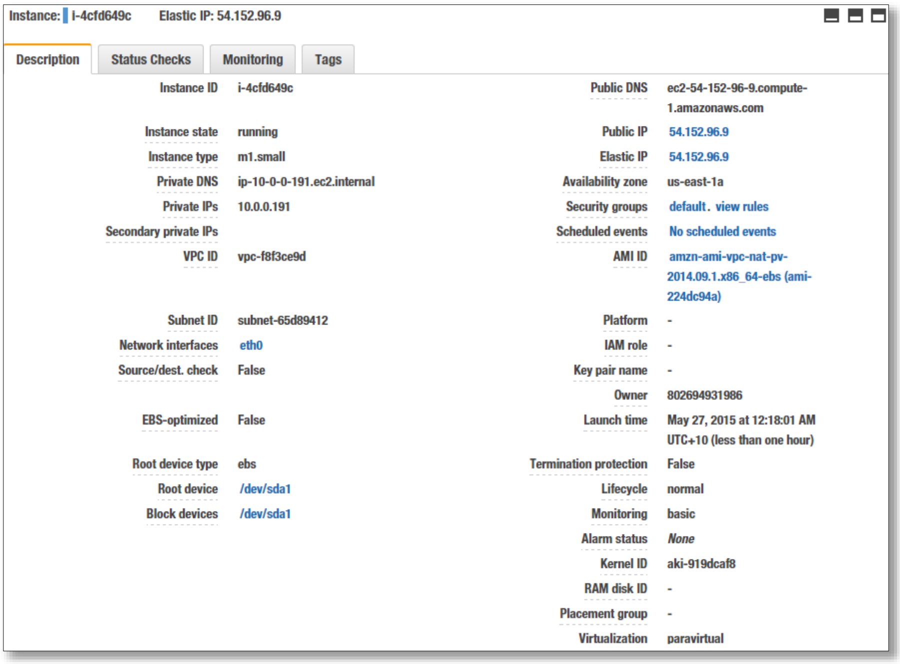
Public Route table (explicitly associated)

Destination	Target
10.0.0.0/16	local
com.amazonaws.us-east-1.s3	vpce
0.0.0.0/0	igw

VPC Main Route table (implicitly associated to private subnet)

Destination	Target
10.0.0.0/16	local
com.amazonaws.us-east-1.s3	vpce
0.0.0.0/0	eni (NAT)

Now go to the EC2 Console and select instances to see the NAT instance.



Instance: i-4cfd649c Elastic IP: 54.152.96.9

Instance ID	i-4cfd649c	Public DNS	ec2-54-152-96-9.compute-1.amazonaws.com
Instance state	running	Public IP	54.152.96.9
Instance type	m1.small	Elastic IP	54.152.96.9
Private DNS	ip-10-0-0-191.ec2.internal	Availability zone	us-east-1a
Private IPs	10.0.0.191	Security groups	default. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-f8f3ce9d	AMI ID	amzn-ami-vpc-nat-pv-2014.09.1.x86_64-ebs (ami-224dc94a)
Subnet ID	subnet-65d89412	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	False	Key pair name	-
		Owner	802694931986
EBS-optimized	False	Launch time	May 27, 2015 at 12:18:01 AM UTC+10 (less than one hour)
Root device type	ebs	Termination protection	False
Root device	/dev/sda1	Lifecycle	normal
Block devices	/dev/sda1	Monitoring	basic
		Alarm status	*None*
		Kernel ID	aki-919dcaf8
		RAM disk ID	-
		Placement group	-
		Virtualization	paravirtual

There are a couple of things to be aware of with this NAT instance created by the VPC wizard:

The virtualization type is paravirtual (PV). It is recommended by AWS for long term support to use hardware virtual machine (HVM) instances. More information on Virtualization Types can be found in the EC2 User Guide for Linux.

The default security group has been used. The following security group rules are recommended for NAT instances.

NATSG: Recommended Rules

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The security group ID (sg-xxxxxxxx)	All	All	Allow inbound traffic from instances assigned to the same security group.
10.0.1.0/24	TCP	80	Allow inbound HTTP traffic from database servers in the private subnet
10.0.1.0/24	TCP	443	Allow inbound HTTPS traffic from database servers in the private subnet
Your network's public IP address range	TCP	22	Allow inbound SSH access to the NAT instance from your network (over the Internet gateway)
Outbound			
Destination	Protocol	Port Range	Comments
0.0.0.0/0	TCP	80	Allow outbound HTTP access to the Internet (over the Internet gateway)
0.0.0.0/0	TCP	443	Allow outbound HTTPS access to the Internet (over the Internet gateway)

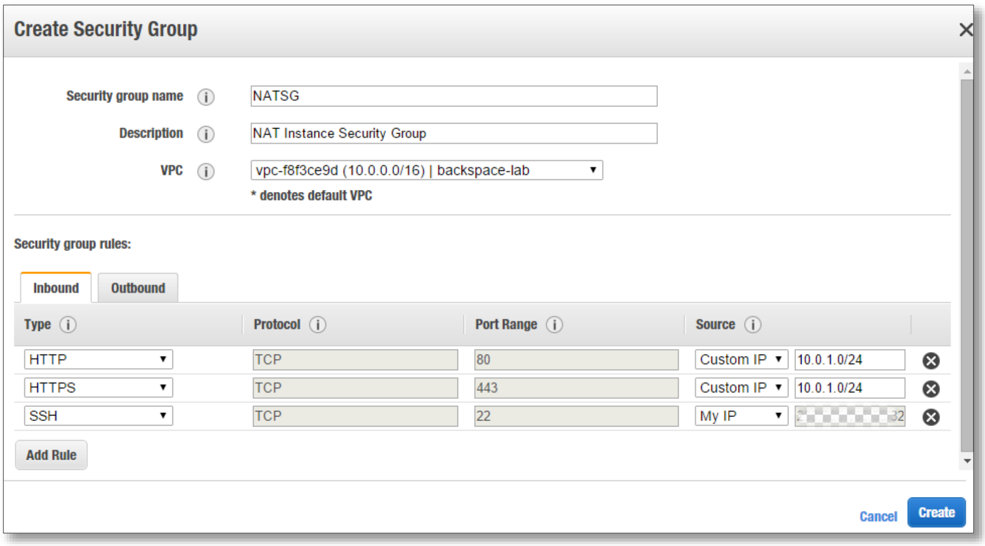
Click on Security Groups

Click “Create Security Group”

Name it NATSG.

Select our newly created VPC.

Add the inbound rules as detailed in the above table. Select “My IP” for port 22.



Create Security Group

Security group name: NATSG

Description: NAT Instance Security Group

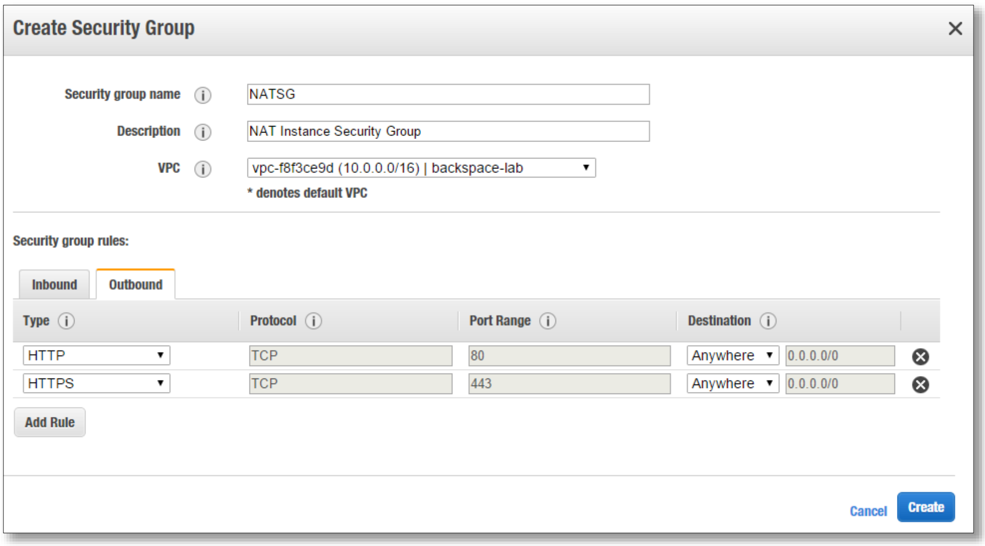
VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

* denotes default VPC

Security group rules:

Type	Protocol	Port Range	Source	
HTTP	TCP	80	Custom IP	10.0.1.0/24
HTTPS	TCP	443	Custom IP	10.0.1.0/24
SSH	TCP	22	My IP	[redacted]

Add the outbound rules as detailed in the above table.



Create Security Group

Security group name: NATSG

Description: NAT Instance Security Group

VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

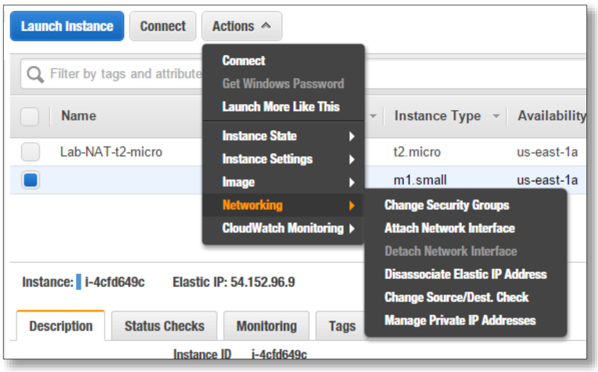
* denotes default VPC

Security group rules:

Type	Protocol	Port Range	Destination	
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0

After the security group is created go back to our instance description page.

Click on Change Security Groups



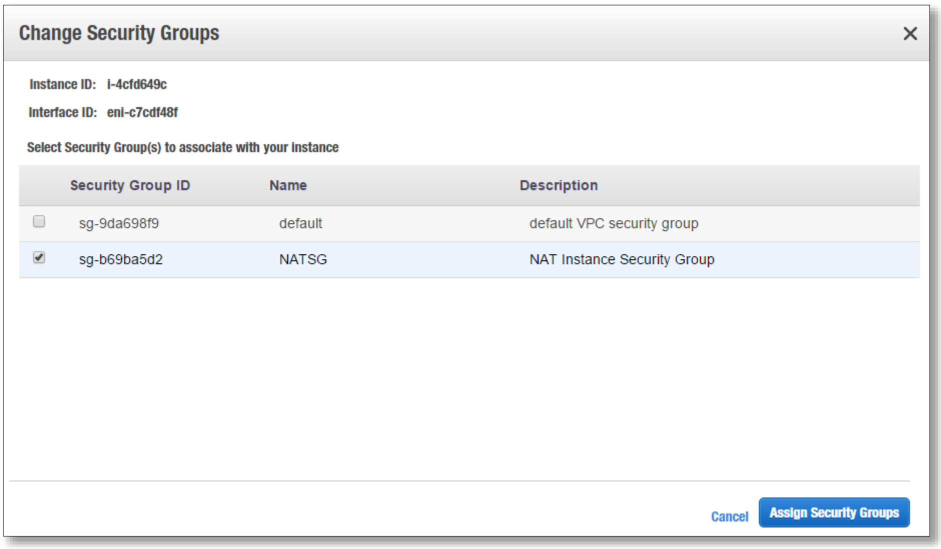
Name	Instance Type	Availability
Lab-NAT-t2-micro	t2.micro	us-east-1a
	m1.small	us-east-1a

Instance: i-4cfd649c Elastic IP: 54.152.96.9

Instance ID i-4cfd649c

Deselect the default security group.

Select the NATSG security group.



Instance ID: i-4cfd649c

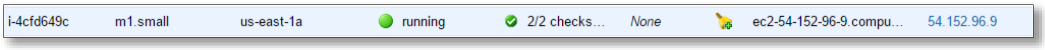
Interface ID: eni-c7cdf48f

Select Security Group(s) to associate with your instance

	Security Group ID	Name	Description
☐	sg-9da698f9	default	default VPC security group
☑	sg-b69ba5d2	NATSG	NAT Instance Security Group

Click Assign Security Groups.

Here you can also see that an Elastic IP has been associated to the m1.small instance.



i-4cfd649c	m1.small	us-east-1a	running	2/2 checks…	None	ec2-54-152-96-9.compu…	54.152.96.9

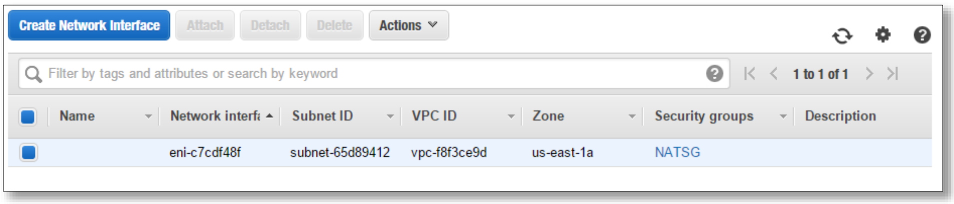
Copyright 2015 all rights reserved - BackSpace.Academy

Moving Elastic Network Interfaces to another Instance

In this section we will look at instance virtualisation types and how to change an Elastic Network Interface connection from one instance to another. This allows us to replace the NAT instance created by the VPC Wizard to a custom NAT instance we create.

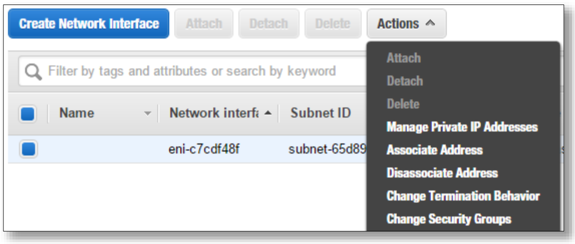
Click on “Network Interfaces”.

Here you can see the VPC Wizard has associated the NAT Elastic IP with an ENI.

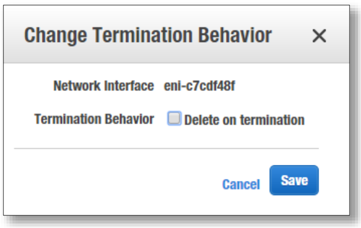


☐	Name	Network interf	Subnet ID	VPC ID	Zone	Security groups	Description
☐		eni-c7cdf48f	subnet-65d89412	vpc-f8f3ce9d	us-east-1a	NATSG	

We want to ensure this ENI is not accidently deleted so we will change the termination behaviour by unchecking “Delete on termination”.



☐	Name	Network interf	Subnet ID
☐		eni-c7cdf48f	subnet-65d89

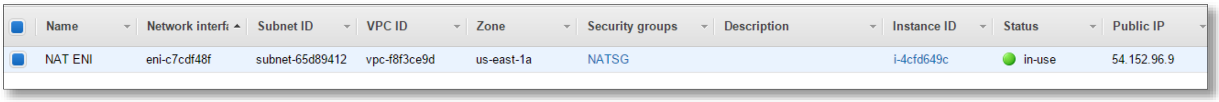


Change Termination Behavior ✕

Network Interface eni-c7cdf48f

Termination Behavior □ Delete on termination

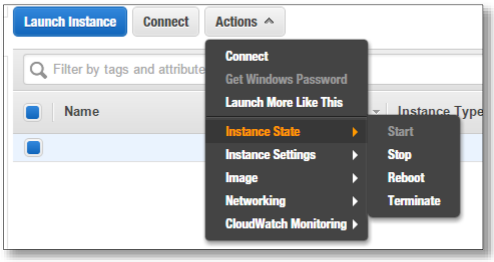
Name the ENI “NAT-ENI” so you can find it easily.



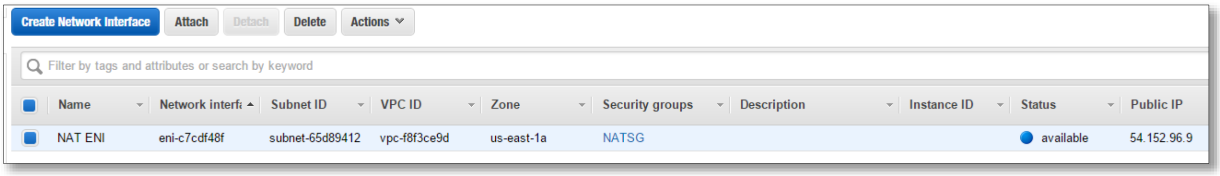
□	Name	Network interf	Subnet ID	VPC ID	Zone	Security groups	Description	Instance ID	Status	Public IP
■	NAT ENI	eni-c7cdf48f	subnet-65d89412	vpc-f8f3ce9d	us-east-1a	NATSG		i-4cfd649c	● in-use	54.152.96.9

Go back to instances

Now terminate the NAT instance.



After the instance has terminated, go back to Network Interfaces



□	Name	Network interf	Subnet ID	VPC ID	Zone	Security groups	Description	Instance ID	Status	Public IP
■	NAT ENI	eni-c7cdf48f	subnet-65d89412	vpc-f8f3ce9d	us-east-1a	NATSG			● available	54.152.96.9

Here you can see the ENI is now available to be attached to another instance.

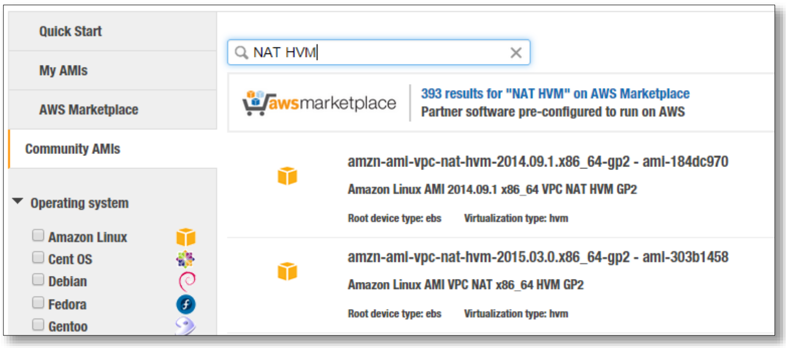
For this example we will remove this NAT instance and replace it with a smaller t2.micro instance. Notice the virtualisation type of the current NAT instance is paravirtual which does not support the new instance types like T2. For long term support it better to use HVM type AMIs. We will create a new NAT from a “HVM” type AMI.

We will then detach the ENI from the m1 instance and attach it to the new T2 instance.

Go back to instances

Select “Launch Instance”

Search Community AMIs for “NAT HVM”



NAT HVM

393 results for "NAT HVM" on AWS Marketplace
Partner software pre-configured to run on AWS

amzn-ami-vpc-nat-hvm-2014.09.1.x86_64-gp2 - ami-184dc970
Amazon Linux AMI 2014.09.1 x86_64 VPC NAT HVM GP2
Root device type: ebs Virtualization type: hvm

amzn-ami-vpc-nat-hvm-2015.03.0.x86_64-gp2 - ami-303b1458
Amazon Linux AMI VPC NAT x86_64 HVM GP2
Root device type: ebs Virtualization type: hvm

Select a suitable AMI and then select the t2.micro instance type.

Click “Next: Configure Instance Details”

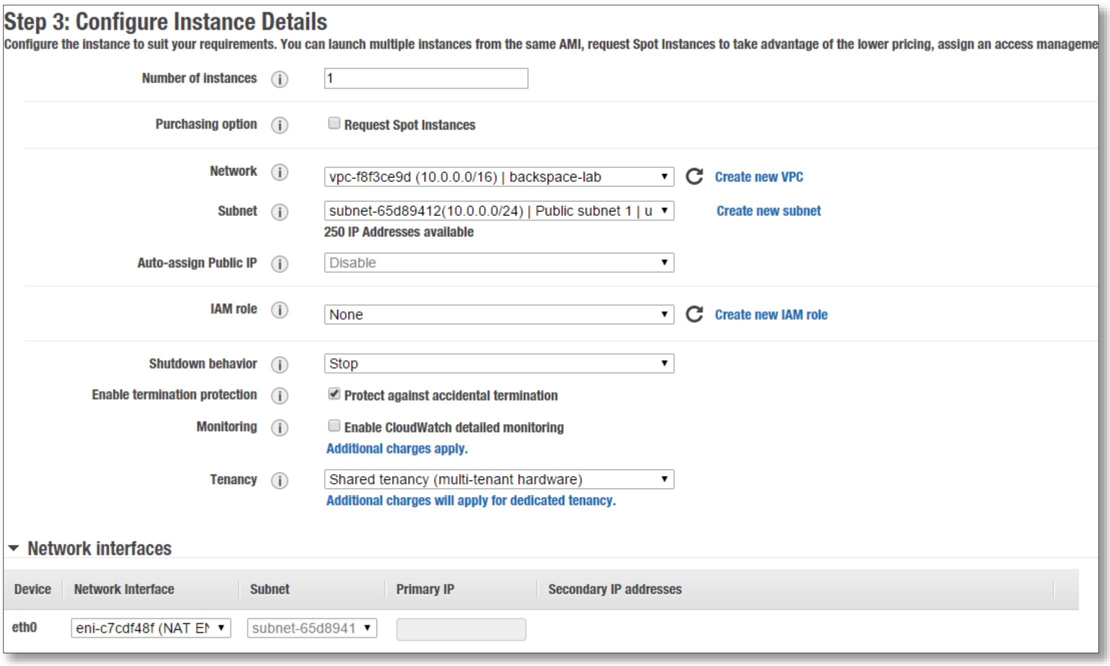
Select the new VPC

Select “Public Subnet 1”

Click “Protect against accidental termination”

Click on Network Interfaces

Select the “NAT ENI”



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 manageme

Number of instances	1
Purchasing option	□ Request Spot Instances
Network	vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab ▾
Subnet	subnet-65d89412(10.0.0.0/24) | Public subnet 1 | u ▾ 250 IP Addresses available
Auto-assign Public IP	Disable ▾
IAM role	None ▾
Shutdown behavior	Stop ▾
Enable termination protection	☑ Protect against accidental termination
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy	Shared tenancy (multi-tenant hardware) ▾ Additional charges will apply for dedicated tenancy.

▾ Network interfaces

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses
eth0	eni-c7cdf48f (NAT EI ▾	subnet-65d8941 ▾		

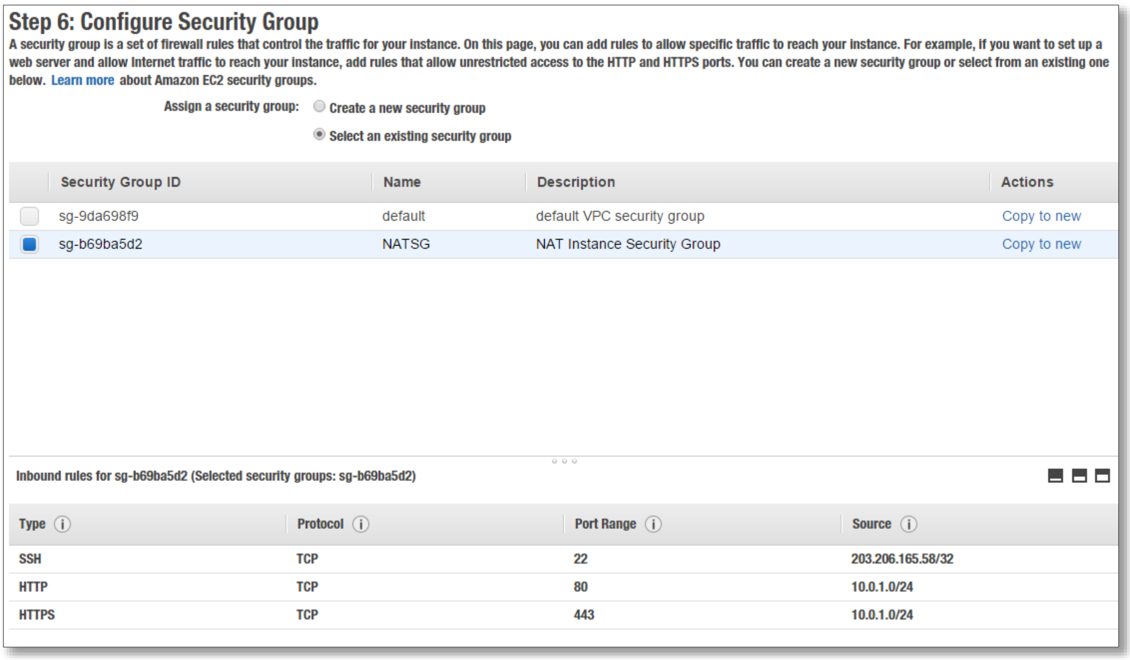
Click “Next: Add Storage”

Click “Next: Tag Instance”

Name it “Lab-NAT-t2-micro”

Click “Next: Configure Security Group”

Select the NATSG security group.



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

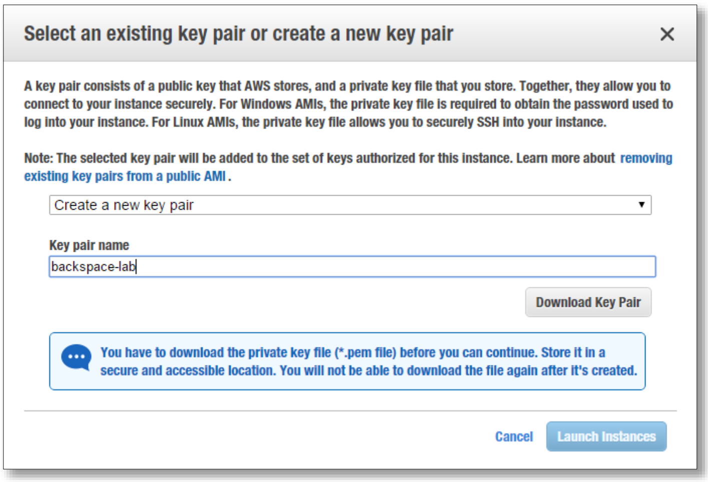
	Security Group ID	Name	Description	Actions
□	sg-9da698f9	default	default VPC security group	Copy to new
☑	sg-b69ba5d2	NATSG	NAT Instance Security Group	Copy to new

Inbound rules for sg-b69ba5d2 (Selected security groups: sg-b69ba5d2)

Type	Protocol	Port Range	Source
SSH	TCP	22	203.206.165.58/32
HTTP	TCP	80	10.0.1.0/24
HTTPS	TCP	443	10.0.1.0/24

Click “Review and Launch”

Click “Launch”



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 .

Create a new key pair

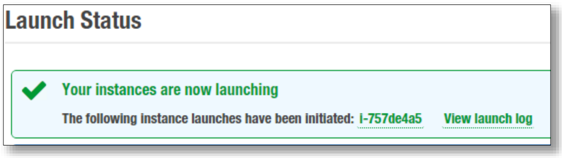
Key pair name: backspace-lab

You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.

Select an existing key pair or create a new one.

Make sure you have downloaded the key pair before proceeding.

Click “Launch Instances”



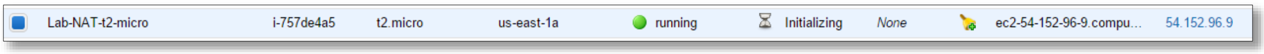
Launch Status

Your instances are now launching

The following instance launches have been initiated: i-757de4a5 View launch log

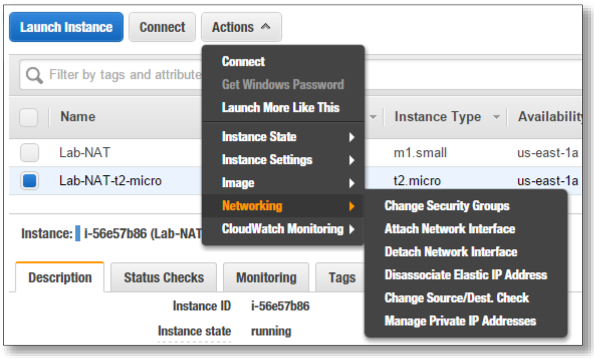
Go back to “Instances”

You will now see that the Elastic IP is now associated with the new t2.micro instance.



Lab-NAT-t2-micro	i-757de4a5	t2.micro	us-east-1a	running	Initializing	None	ec2-54-152-96-9.compu...	54.152.96.9

Now check that “source destination check” on the new instance is disabled. Without this disabled, NAT communication cannot be established.



Name	Instance Type	Availabilit
Lab-NAT	m1.small	us-east-1a
Lab-NAT-t2-micro	t2.micro	us-east-1a

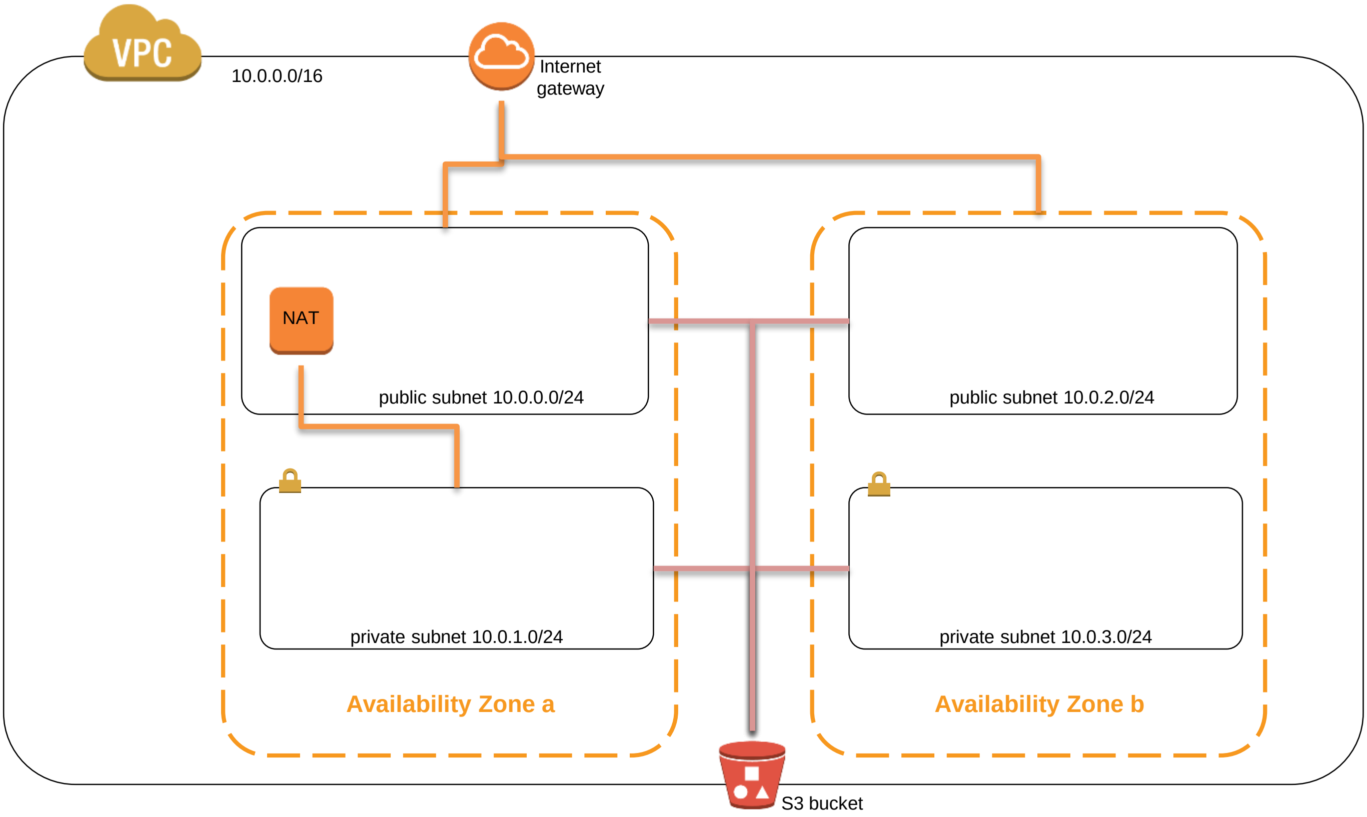
Instance: i-56e57b86 (Lab-NAT

Instance ID i-56e57b86

Instance state running

Creating Public and Private Subnets in a Second AZ

In this section we will look increase the availability of our VPC architecture by creating subnets in a second availability zone.


VPC
10.0.0.0/16
Internet gateway
NAT
public subnet 10.0.0.0/24
private subnet 10.0.1.0/24
Availability Zone a
public subnet 10.0.2.0/24
private subnet 10.0.3.0/24
Availability Zone b
S3 bucket

Go to “Subnets”

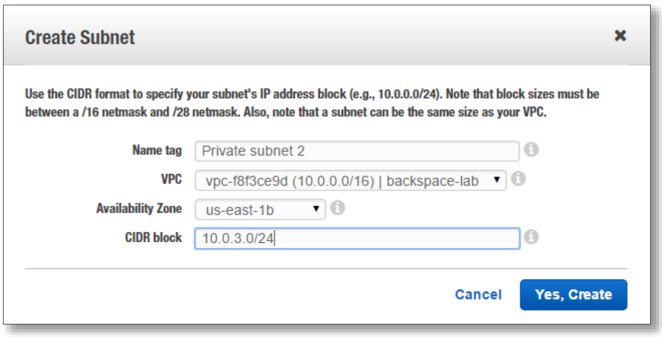
Click “Create Subnet”

Call it “Private subnet 2”

Select the newly created VPC

Select us-east-1b

Use CIDR block 10.0.3.0/24



Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC.

Name tag: Private subnet 2

VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

Availability Zone: us-east-1b

CIDR block: 10.0.3.0/24

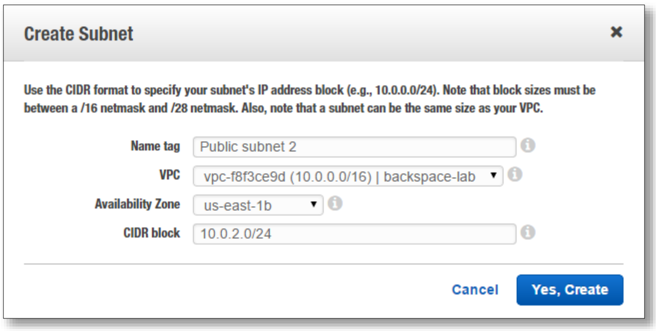
Click “Create Subnet” again

Call it “Public subnet 2”

Select the newly created VPC

Select us-east-1b

Use CIDR block 10.0.2.0/24



Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC.

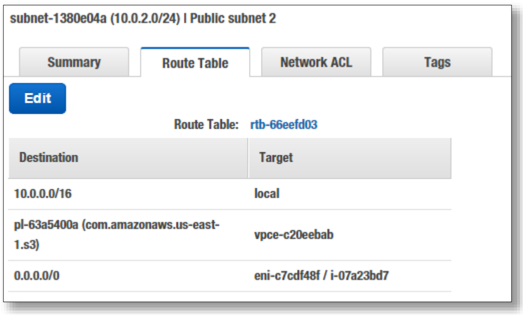
Name tag: Public subnet 2

VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

Availability Zone: us-east-1b

CIDR block: 10.0.2.0/24

Click on Public subnet 2 and click the Route Table tab.



subnet-1380e04a (10.0.2.0/24) | Public subnet 2

Route Table: rtb-66eefd03

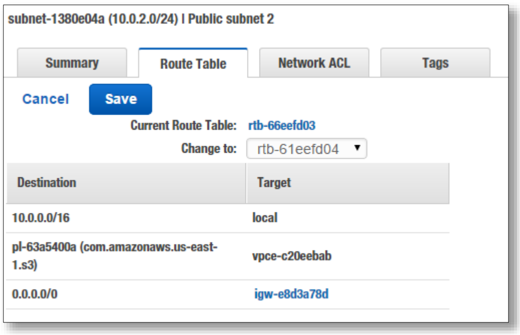
Destination	Target
10.0.0.0/16	local
pl-63a5400a (com.amazonaws.us-east-1.s3)	vpce-c20eebab
0.0.0.0/0	eni-c7cdf48f / i-07a23bd7

A private subnet has been created because we have not explicitly defined a route to the Internet Gateway.

We will now change from the Main Route table to the Public Route table defined previously.

Click on edit and select the Public Route table defined in Public subnet 1. When you select it the target will change to show the IGW.

Click Save



subnet-1380e04a (10.0.2.0/24) | Public subnet 2

Current Route Table: rtb-66eefd03

Change to: rtb-61eefd04

Destination	Target
10.0.0.0/16	local
pl-63a5400a (com.amazonaws.us-east-1.s3)	vpce-c20eebab
0.0.0.0/0	igw-e8d3a78d

We will now tighten up our NACL by creating public and private subnet ACLs and only defining the ports required to be open:

- HTTP (port 80)
- HTTPS (port 443)
- SSH (port 22)
- MySQL (port 3306)
- Linux kernels use ephemeral (short lived) ports 32768-61000.
- Requests originating from ELBs use ephemeral ports 1024-65535.

ACL Rules for the Private Subnet

Inbound					
Rule #	**Source IP**	**Protocol**	**Port**	**Allow/Deny**	**Comments**
100	10.0.0.0/24	TCP	3306	ALLOW	Allows web servers in public subnet 1 to read and write to MySQL servers in the private subnet
110	10.0.2.0/24	TCP	3306	ALLOW	Allows web servers in public subnet 2 to read and write to MySQL servers in the private subnet
120	10.0.0.0/24	TCP	22	ALLOW	Allows inbound SSH traffic from the SSH bastion in public subnet 1
130	10.0.2.0/24	TCP	22	ALLOW	Allows inbound SSH traffic from the SSH bastion in public subnet 2
140	0.0.0.0/0	TCP	32768-61000	ALLOW	Allows inbound return traffic from NAT instance in the public subnet for requests originating in the private subnet
*	0.0.0.0/0	all	all	DENY	Denies all inbound traffic not already handled by a preceding rule (not modifiable)
Outbound					
Rule #	**Dest IP**	**Protocol**	**Port**	**Allow/Deny**	**Comments**
100	0.0.0.0/0	TCP	80	ALLOW	Allows outbound HTTP traffic from the subnet to the Internet
110	0.0.0.0/0	TCP	443	ALLOW	Allows outbound HTTPS traffic from the subnet to the Internet
120	10.0.0.0/24	TCP	32768-61000	ALLOW	Allows outbound responses to public subnet 1 (for example, responses to web servers in the public subnet that are communicating with DB Servers in the private subnet)
130	10.0.2.0/24	TCP	32768-61000	ALLOW	Allows outbound responses to public subnet 2 (for example, responses to web servers in the public subnet that are communicating with DB Servers in the private subnet)
*	0.0.0.0/0	all	all	DENY	Denies all outbound traffic not already handled by a preceding rule (not modifiable)

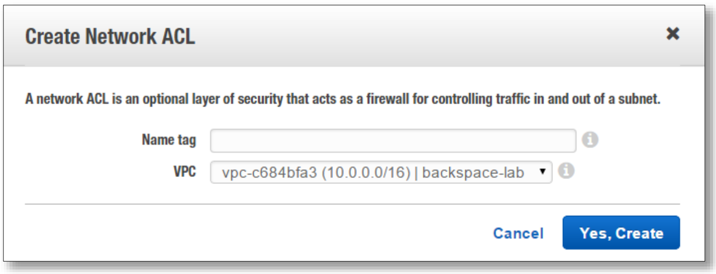
ACL Rules for the Public Subnet

Inbound					
Rule #	**Source IP**	**Protocol**	**Port**	**Allow/Deny**	**Comments**
100	0.0.0.0/0	TCP	80	ALLOW	Allows inbound HTTP traffic from anywhere
110	0.0.0.0/0	TCP	443	ALLOW	Allows inbound HTTPS traffic from anywhere
120	Public IP address range of your home network	TCP	22	ALLOW	Allows inbound SSH traffic from your home network (over the Internet gateway)
130	0.0.0.0/0	TCP	1024-65535	ALLOW	Allows inbound return traffic from requests originating in the subnet and from the ELB
*	0.0.0.0/0	all	all	DENY	Denies all inbound traffic not already handled by a preceding rule (not modifiable)
Outbound					
Rule #	**Dest IP**	**Protocol**	**Port**	**Allow/Deny**	**Comments**
100	0.0.0.0/0	TCP	80	ALLOW	Allows outbound HTTP traffic from the subnet to the Internet
110	0.0.0.0/0	TCP	443	ALLOW	Allows outbound HTTPS traffic from the subnet to the Internet
120	10.0.1.0/24	TCP	3306	ALLOW	Allows outbound MySQL access to database servers in private subnet 1
130	10.0.3.0/24	TCP	3306	ALLOW	Allows outbound MySQL access to database servers in private subnet 2
140	0.0.0.0/0	TCP	1024-65535	ALLOW	Allows outbound responses to the Internet or the ELB
150	10.0.1.0/24	TCP	22	ALLOW	Allows outbound SSH access to instances in your private subnet 1 (from the SSH bastion)
160	10.0.3.0/24	TCP	22	ALLOW	Allows outbound SSH access to instances in your private subnet 2 (from the SSH bastion)
*	0.0.0.0/0	all	all	DENY	Denies all outbound traffic not already handled by a preceding rule (not modifiable)

We will create two new NACLs called Public subnet ACL and Private subnet ACL. Unlike security groups, NACLs are stateless and require response traffic to be allowed on both inbound and outbound rules.

Go to the Network ACLs page and click “Create Network ACL”

Call it “Public NACL”



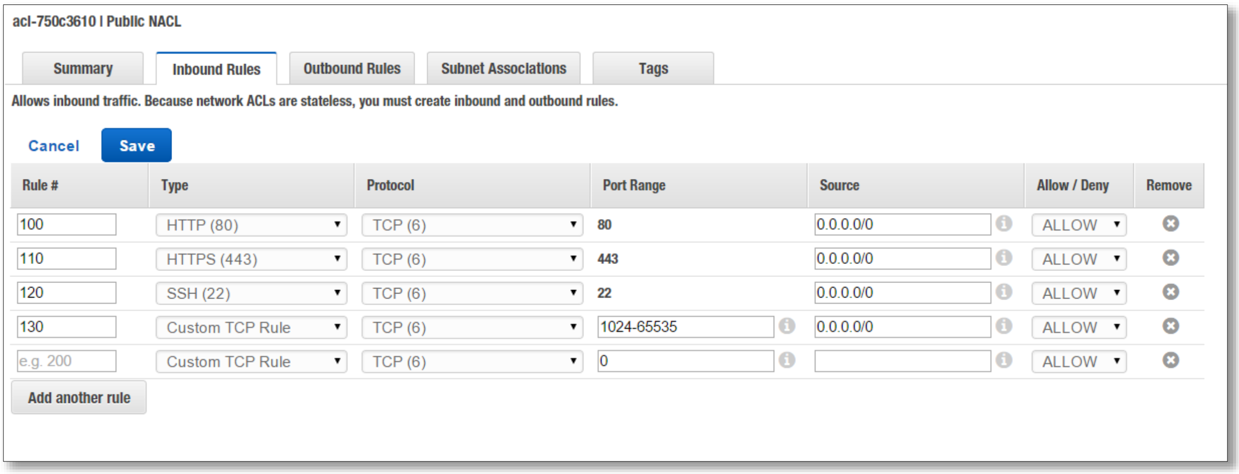
Create Network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Name tag:

VPC: vpc-c684bfa3 (10.0.0.0/16) | backspace-lab

Create the inbound rules for Public NACL

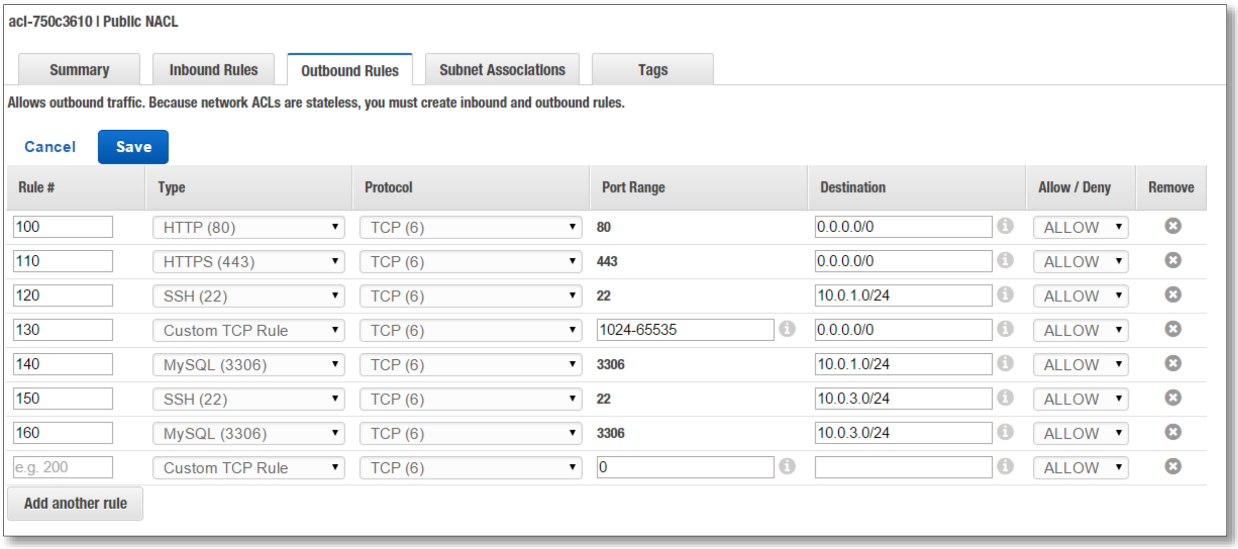


acl-750c3610 | Public NACL

Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny	Remove
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW	⊗
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW	⊗
120	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW	⊗
130	Custom TCP Rule	TCP (6)	1024-65535	0.0.0.0/0	ALLOW	⊗
e.g. 200	Custom TCP Rule	TCP (6)	0		ALLOW	⊗

Create the inbound rules for Public NACL

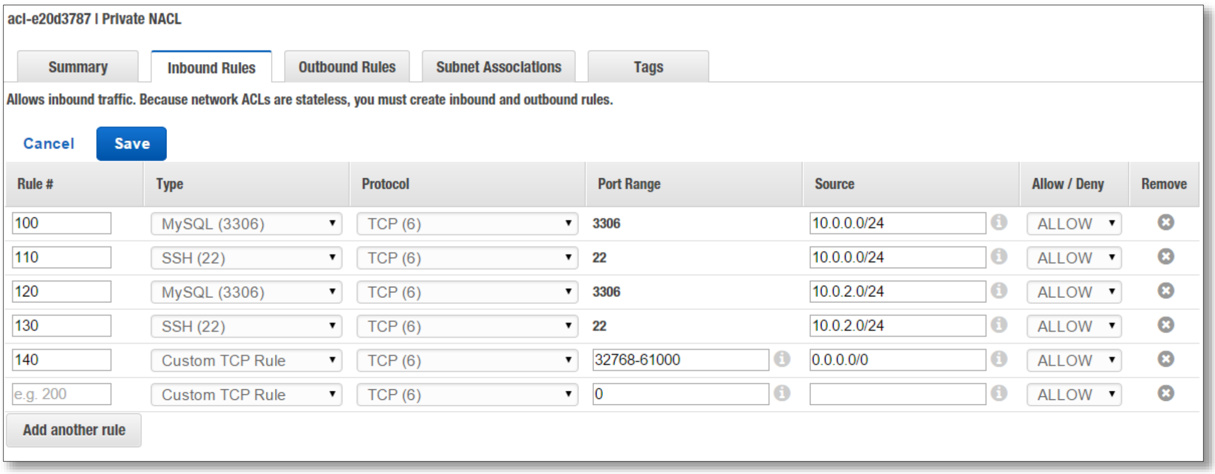


acl-750c3610 | Public NACL

Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

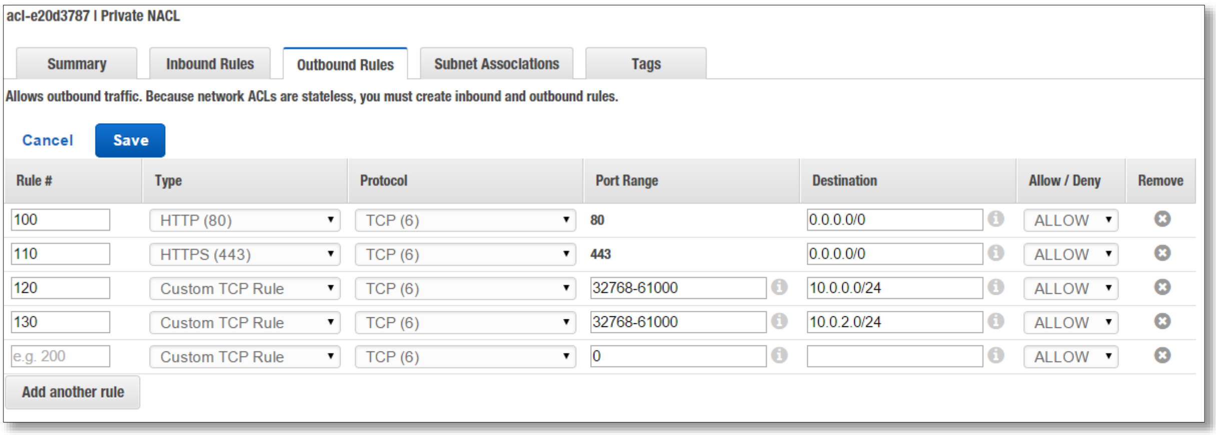
Rule #	Type	Protocol	Port Range	Destination	Allow / Deny	Remove
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW	⊗
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW	⊗
120	SSH (22)	TCP (6)	22	10.0.1.0/24	ALLOW	⊗
130	Custom TCP Rule	TCP (6)	1024-65535	0.0.0.0/0	ALLOW	⊗
140	MySQL (3306)	TCP (6)	3306	10.0.1.0/24	ALLOW	⊗
150	SSH (22)	TCP (6)	22	10.0.3.0/24	ALLOW	⊗
160	MySQL (3306)	TCP (6)	3306	10.0.3.0/24	ALLOW	⊗
e.g. 200	Custom TCP Rule	TCP (6)	0		ALLOW	⊗

Now do the same for the Private NACL



Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny	Remove
100	MySQL (3306)	TCP (6)	3306	10.0.0.0/24	ALLOW	⊗
110	SSH (22)	TCP (6)	22	10.0.0.0/24	ALLOW	⊗
120	MySQL (3306)	TCP (6)	3306	10.0.2.0/24	ALLOW	⊗
130	SSH (22)	TCP (6)	22	10.0.2.0/24	ALLOW	⊗
140	Custom TCP Rule	TCP (6)	32768-61000	0.0.0.0/0	ALLOW	⊗
e.g. 200	Custom TCP Rule	TCP (6)	0		ALLOW	⊗



Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny	Remove
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW	⊗
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW	⊗
120	Custom TCP Rule	TCP (6)	32768-61000	10.0.0.0/24	ALLOW	⊗
130	Custom TCP Rule	TCP (6)	32768-61000	10.0.2.0/24	ALLOW	⊗
e.g. 200	Custom TCP Rule	TCP (6)	0		ALLOW	⊗

After we have created our subnets in another availability zone in the later lesson we will associate all our subnets with the public and private subnet ACLs. For now we will leave it as it is.

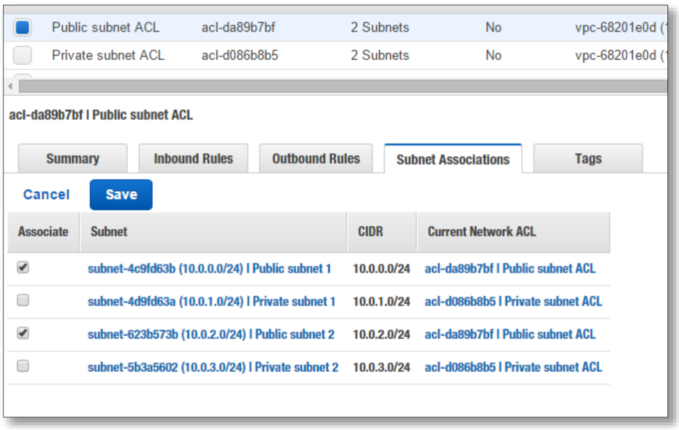
Now that we have our Public and Private NACLs we created in our first lesson we can associate them with our four subnets.

Click on Network ACLs.

Click on the Public subnet ACL.

Click on the Subnet Associations tab

Select the two public subnets and Save

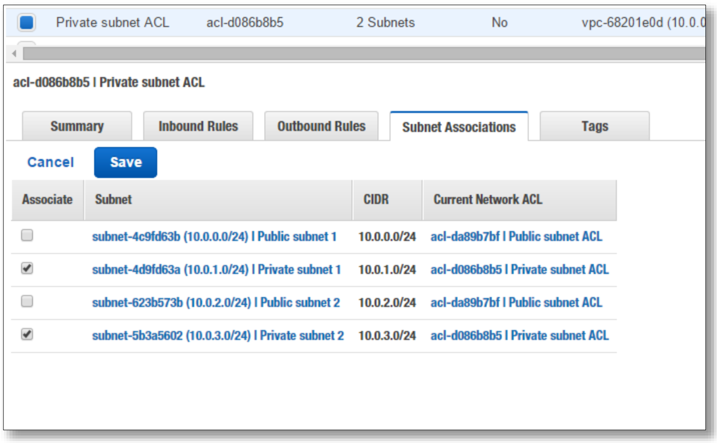


■	Public subnet ACL	acl-da89b7bf	2 Subnets	No	vpc-68201e0d (
□	Private subnet ACL	acl-d086b8b5	2 Subnets	No	vpc-68201e0d (

acl-da89b7bf | Public subnet ACL

Associate	Subnet	CIDR	Current Network ACL
☑	subnet-4c9fd63b (10.0.0.0/24) | Public subnet 1	10.0.0.0/24	acl-da89b7bf | Public subnet ACL
□	subnet-4d9fd63a (10.0.1.0/24) | Private subnet 1	10.0.1.0/24	acl-d086b8b5 | Private subnet ACL
☑	subnet-623b573b (10.0.2.0/24) | Public subnet 2	10.0.2.0/24	acl-da89b7bf | Public subnet ACL
□	subnet-5b3a5602 (10.0.3.0/24) | Private subnet 2	10.0.3.0/24	acl-d086b8b5 | Private subnet ACL

Do the same for the Private NACL and select the two Private subnets.



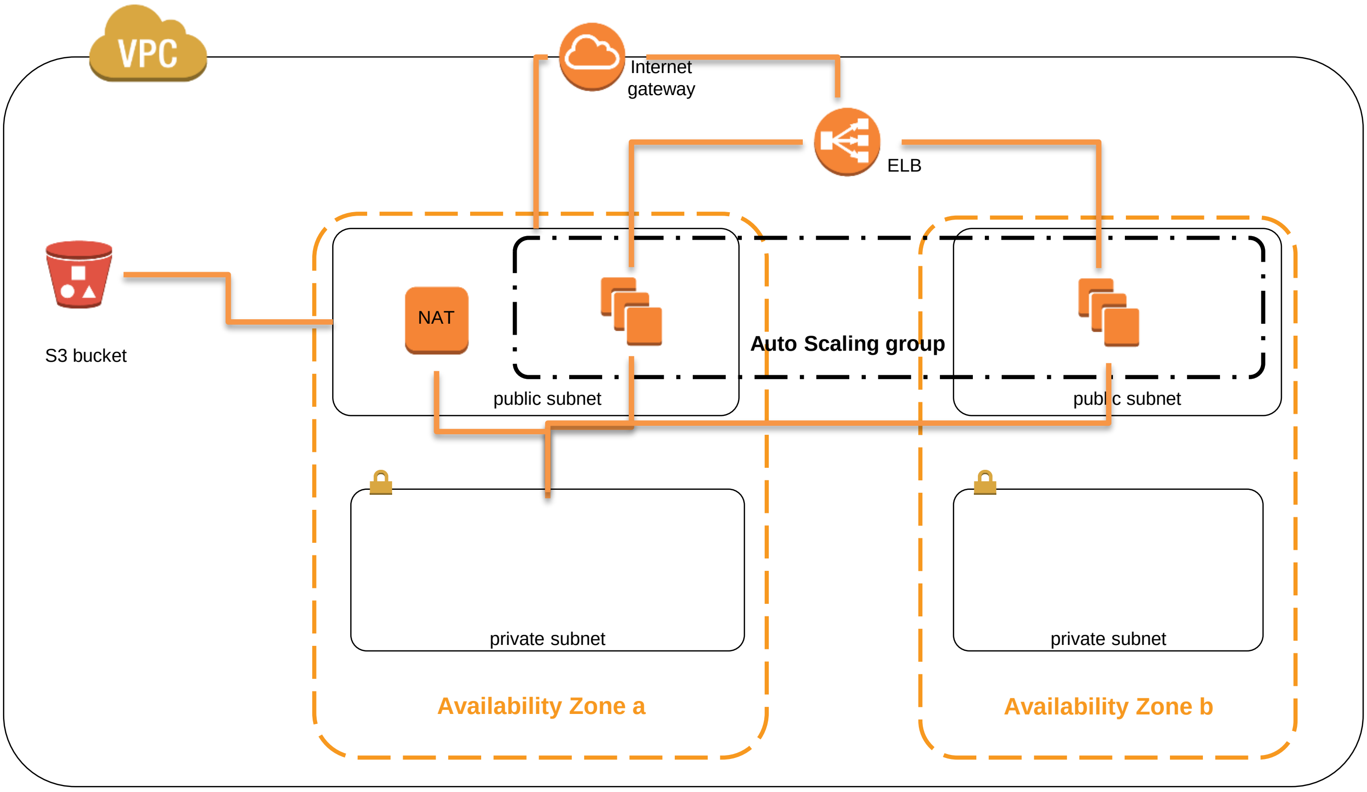
■	Private subnet ACL	acl-d086b8b5	2 Subnets	No	vpc-68201e0d (10.0.0

acl-d086b8b5 | Private subnet ACL

Associate	Subnet	CIDR	Current Network ACL
□	subnet-4c9fd63b (10.0.0.0/24) | Public subnet 1	10.0.0.0/24	acl-da89b7bf | Public subnet ACL
☑	subnet-4d9fd63a (10.0.1.0/24) | Private subnet 1	10.0.1.0/24	acl-d086b8b5 | Private subnet ACL
□	subnet-623b573b (10.0.2.0/24) | Public subnet 2	10.0.2.0/24	acl-da89b7bf | Public subnet ACL
☑	subnet-5b3a5602 (10.0.3.0/24) | Private subnet 2	10.0.3.0/24	acl-d086b8b5 | Private subnet ACL

Creating an ELB and Auto Scaling Group

In this section we will look at increasing the availability and fault tolerance of our VPC architecture by creating an Auto Scaling Group and balancing traffic across instances using an Elastic Load Balancer.



Go back to the EC2 console and select Security Groups

The security group settings recommended for a web server are:

WebServerSG: Recommended Rules

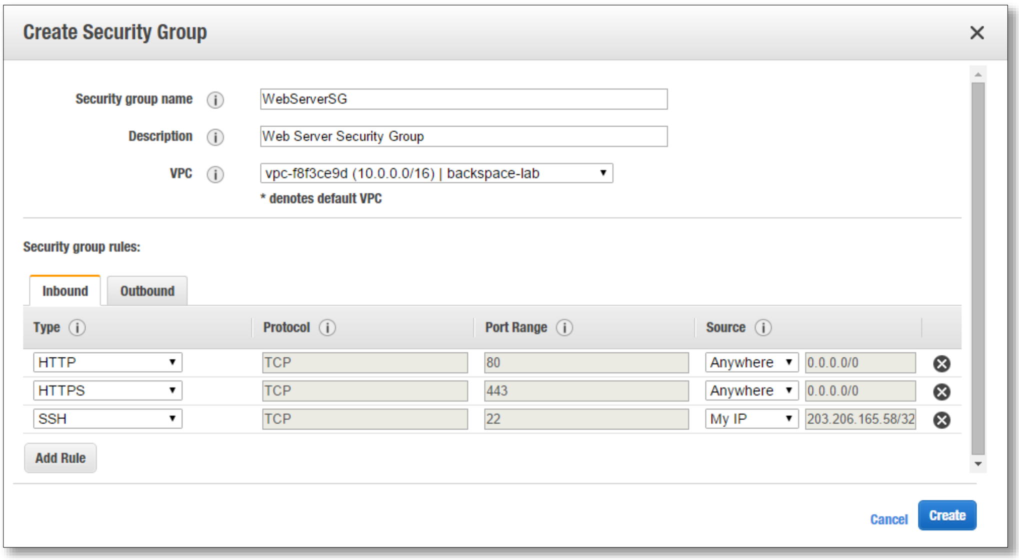
Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The security group ID (sg-xxxxxxxx)	All	All	Allow inbound traffic from instances assigned to the same security group.
0.0.0.0/0	TCP	80	Allow inbound HTTP access to the web servers from anywhere
0.0.0.0/0	TCP	443	Allow inbound HTTPS access to the web servers from anywhere
Your home network's public IP address range	TCP	22	Allow inbound SSH access to Linux instances from your home network (over the Internet gateway)
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
The ID of your DBServerSG security group	TCP	3306	Allow outbound MySQL access to the database servers assigned to DBServerSG

Click Security Groups

Click “Create Security Group”

Name the security group WebServerSG.

Add the inbound rules as detailed in the above table. Select “My IP” for port 22. We will add the outbound rules after the DB Security Group is created.



Create Security Group

Security group name: WebServerSG

Description: Web Server Security Group

VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

* denotes default VPC

Security group rules:

Inbound | Outbound

Type	Protocol	Port Range	Source	
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0
SSH	TCP	22	My IP	203.206.165.58/32

Select Load Balancers

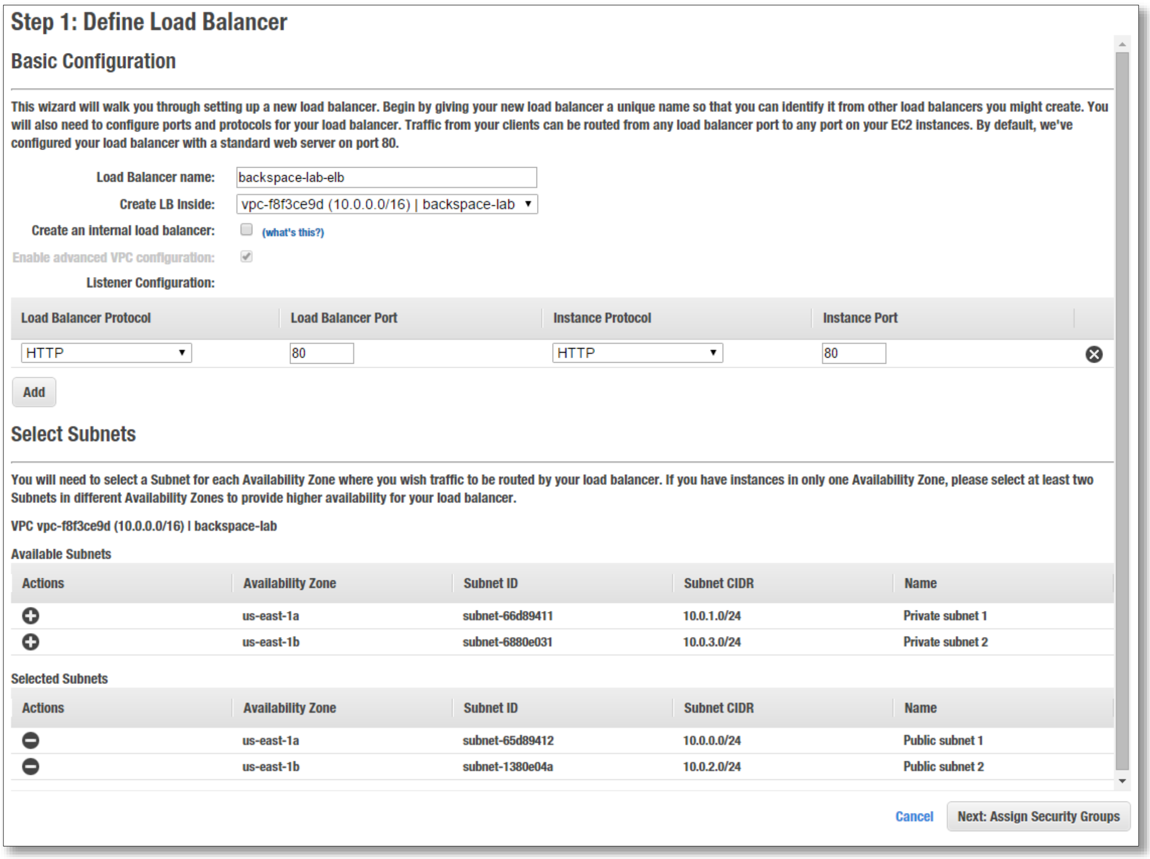
Click “Create Load balancer”

Name it “backspace-lab-elb”

Select the newly created VPC

In a production we would add an https listener. As this would require us to purchase and upload an SSL certificate we will not add it for this lab.

Select the two public subnets.



Step 1: Define Load Balancer

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name: backspace-lab-elb

Create LB Inside: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab ▾

Create an internal load balancer: □ (what's this?)

Enable advanced VPC configuration: ☑

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	
HTTP ▾	80	HTTP ▾	80	⊗

Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-f8f3ce9d (10.0.0.0/16) I backspace-lab

Available Subnets

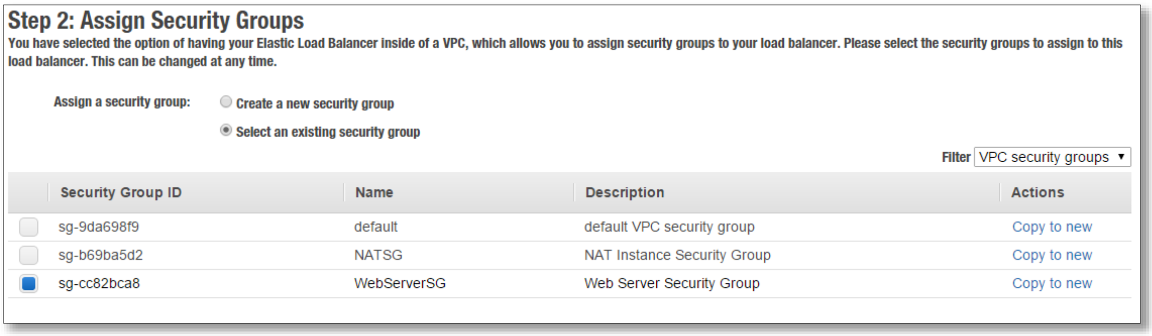
Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊕	us-east-1a	subnet-66d89411	10.0.1.0/24	Private subnet 1
⊕	us-east-1b	subnet-6880e031	10.0.3.0/24	Private subnet 2

Selected Subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊖	us-east-1a	subnet-65d89412	10.0.0.0/24	Public subnet 1
⊖	us-east-1b	subnet-1380e04a	10.0.2.0/24	Public subnet 2

Click “Next: Assign Security Groups”

Select the WebServerSG security group.



Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group: ○ Create a new security group

● Select an existing security group

Filter VPC security groups ▾

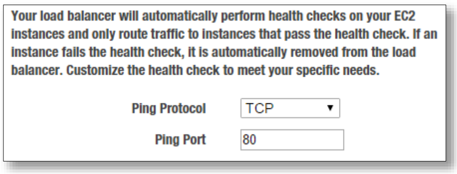
	Security Group ID	Name	Description	Actions
□	sg-9da698f9	default	default VPC security group	Copy to new
□	sg-b69ba5d2	NATSG	NAT Instance Security Group	Copy to new
☑	sg-cc82bca8	WebServerSG	Web Server Security Group	Copy to new

Click “Next: Configure Security Settings”

A warning will come up informing us that we have not added an https listener. Ignore this for the lab.

Click “Next: Configure Health Check”

Change to Ping Protocol TCP and Ping Port 80.



Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

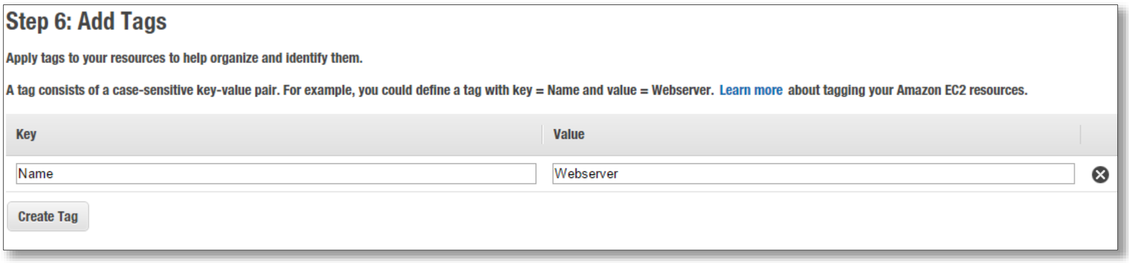
Ping Protocol: TCP

Ping Port: 80

Click “Next: Add EC2 Instances”

Click “Next: Add Tags”

Add the tag Name Webserver for the EC2 instances.



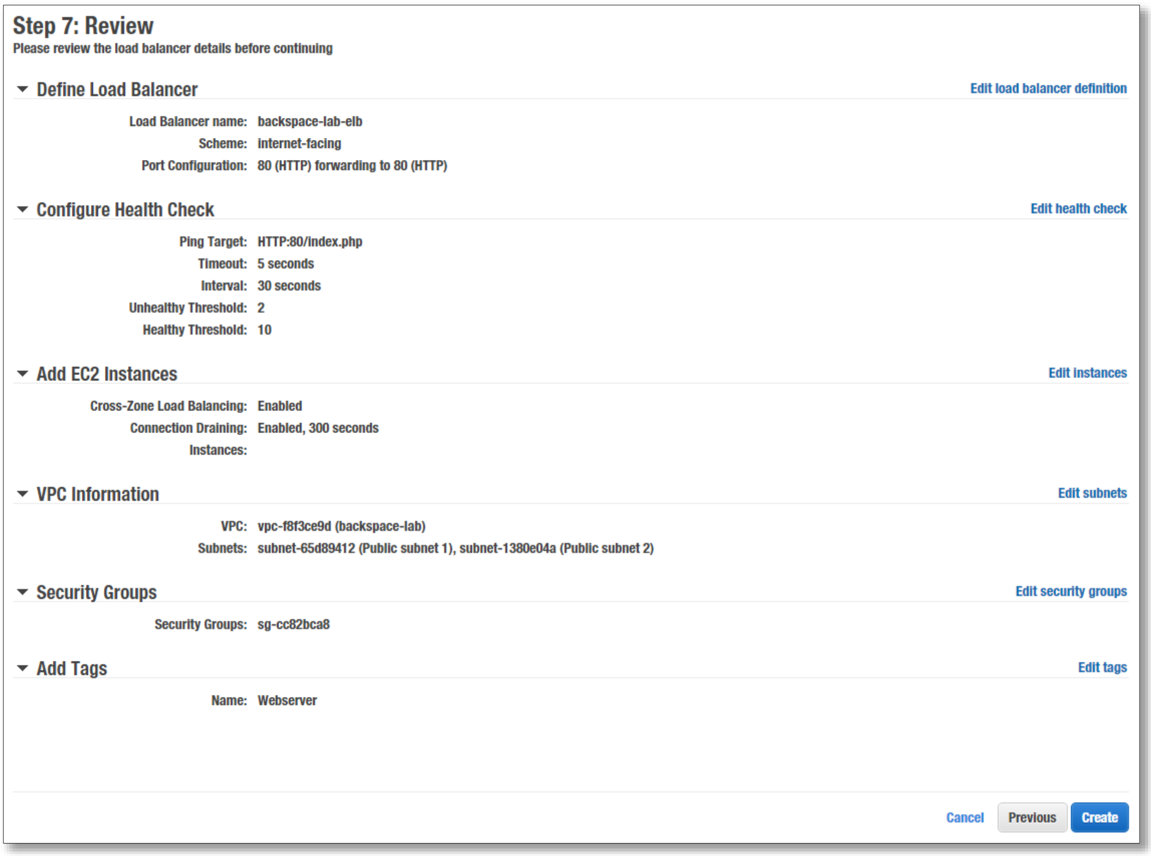
Step 6: Add Tags

Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key	Value
Name	Webserver

Click “Review and Create”



Step 7: Review

Please review the load balancer details before continuing

Define Load Balancer

Load Balancer name: backspace-lab-elb
Scheme: internet-facing
Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)

Configure Health Check

Ping Target: HTTP:80/index.php
Timeout: 5 seconds
Interval: 30 seconds
Unhealthy Threshold: 2
Healthy Threshold: 10

Add EC2 Instances

Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 300 seconds
Instances:

VPC Information

VPC: vpc-f8f3ce9d (backspace-lab)
Subnets: subnet-65d89412 (Public subnet 1), subnet-1380e04a (Public subnet 2)

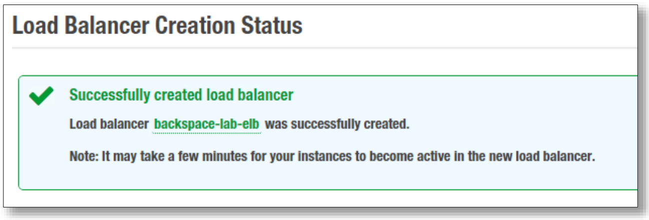
Security Groups

Security Groups: sg-cc82bca8

Add Tags

Name: Webserver

Click Create



Load Balancer Creation Status

Successfully created load balancer

Load balancer backspace-lab-elb was successfully created.

Note: It may take a few minutes for your instances to become active in the new load balancer.

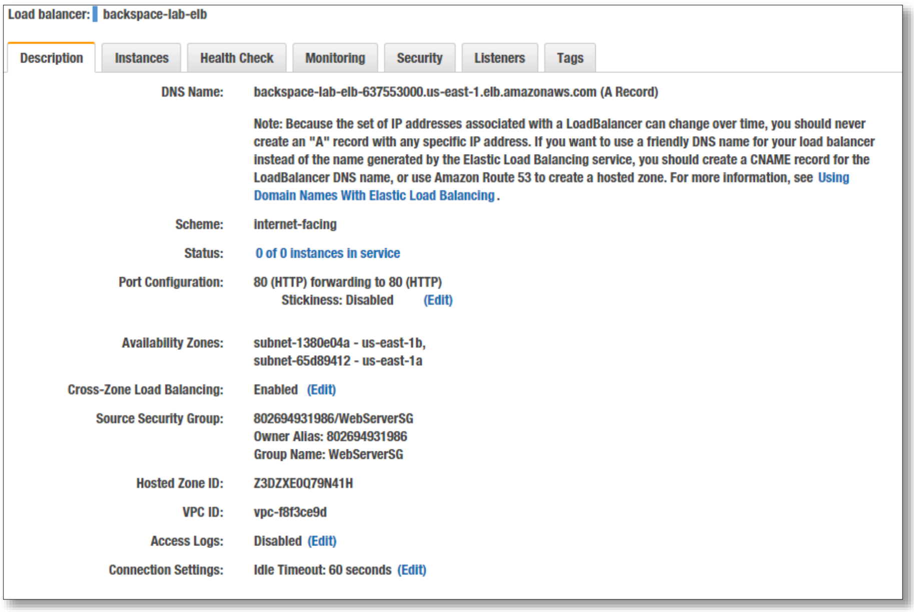
Looking at the newly created load balancer we note that stickiness on Port Configuration is disabled.

It is important to maintain session state when using Auto Scaling otherwise when traffic is diverted to another instance the temporary application data is lost. Session state can be maintained two ways:

1. Storing session state in persistent storage such as AWS ElastiCache or DynamoDB.
2. Using stickiness on the port configuration.

Using stickiness is the simplest solution although it does have disadvantages. If you have one instance fully loaded and another instance kicks in, the second instance will have little traffic. This means the instances aren't well balanced. This will change over time as users drop off and new users come online.

If you effectively scale horizontally, this will reduce the balancing disadvantage of stickiness. You should use many small instances rather than large instances. For example if you have the minimum number of instances set to 4, when another instance kicks in, the effect is not as significant.



Load balancer: backspace-lab-elb

DNS Name: backspace-lab-elb-637553000.us-east-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see Using Domain Names With Elastic Load Balancing.

Scheme: internet-facing

Status: 0 of 0 instances in service

Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)
Stickiness: Disabled (Edit)

Availability Zones: subnet-1380e04a - us-east-1b,
subnet-65d89412 - us-east-1a

Cross-Zone Load Balancing: Enabled (Edit)

Source Security Group: 802694931986/WebServerSG
Owner Alias: 802694931986
Group Name: WebServerSG

Hosted Zone ID: Z3DZXE0Q79N41H

VPC ID: vpc-f8f3ce9d

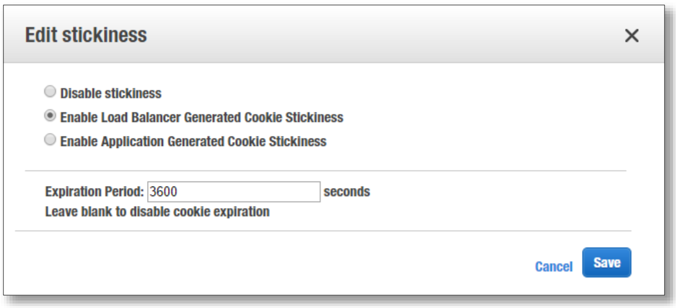
Access Logs: Disabled (Edit)

Connection Settings: Idle Timeout: 60 seconds (Edit)

Click on "Edit" next to Stckiness:Disabled.

Set "Enable Load Balancer Generated Cookie Stickiness"

Set Expiration Period to 60 seconds.



Edit stickiness

○ Disable stickiness
● Enable Load Balancer Generated Cookie Stickiness
○ Enable Application Generated Cookie Stickiness

Expiration Period: 3600 seconds
Leave blank to disable cookie expiration

Select Launch Configurations.

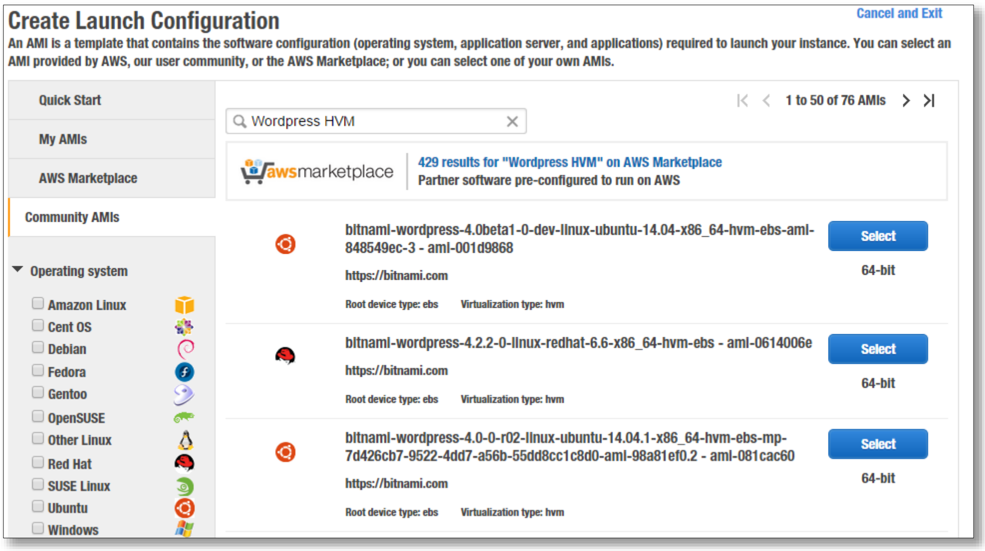
Click “Create Auto Scaling Group”

Click “Create Launch Configuration”

Select Community AMIs.

Select a Wordpress AMI with Virtualization type: hvm.

Note: In a production application you would first create an EC2 instance from the AMI, configure the WordPress application and database settings, create any bootstrapping scripts, and then create another AMI to use with your auto scaling group.



Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Wordpress HVM

429 results for "Wordpress HVM" on AWS Marketplace
Partner software pre-configured to run on AWS

bitnami-wordpress-4.0beta1-0-dev-linux-ubuntu-14.04-x86_64-hvm-ebs-ami-848549ec-3 - ami-001d9868
https://bitnami.com
Root device type: ebs Virtualization type: hvm
64-bit

bitnami-wordpress-4.2.2-0-linux-redhat-6.6-x86_64-hvm-ebs - ami-0614006e
https://bitnami.com
Root device type: ebs Virtualization type: hvm
64-bit

bitnami-wordpress-4.0-0-r02-linux-ubuntu-14.04.1-x86_64-hvm-ebs-mp-7d426cb7-9522-4dd7-a56b-55dd8cc1c8d0-ami-98a81ef0.2 - ami-081cac60
https://bitnami.com
Root device type: ebs Virtualization type: hvm
64-bit

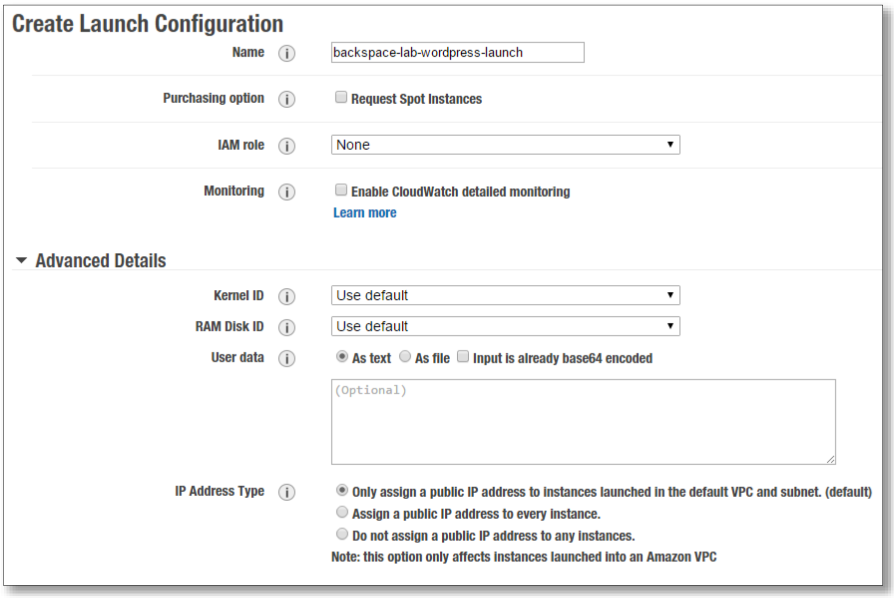
Click Next

Select t2.micro as the instance type

Click Next

Name the Launch Configuration “backspace-lab-wordpress-launch”

The user data section is where you would put bootstrap scripts to update software etc on instance start up.



Create Launch Configuration

Name: backspace-lab-wordpress-launch

Purchasing option: □ Request Spot Instances

IAM role: None

Monitoring: □ Enable CloudWatch detailed monitoring
Learn more

Advanced Details

Kernel ID: Use default

RAM Disk ID: Use default

User data: ● As text ○ As file □ Input is already base64 encoded

(Optional)

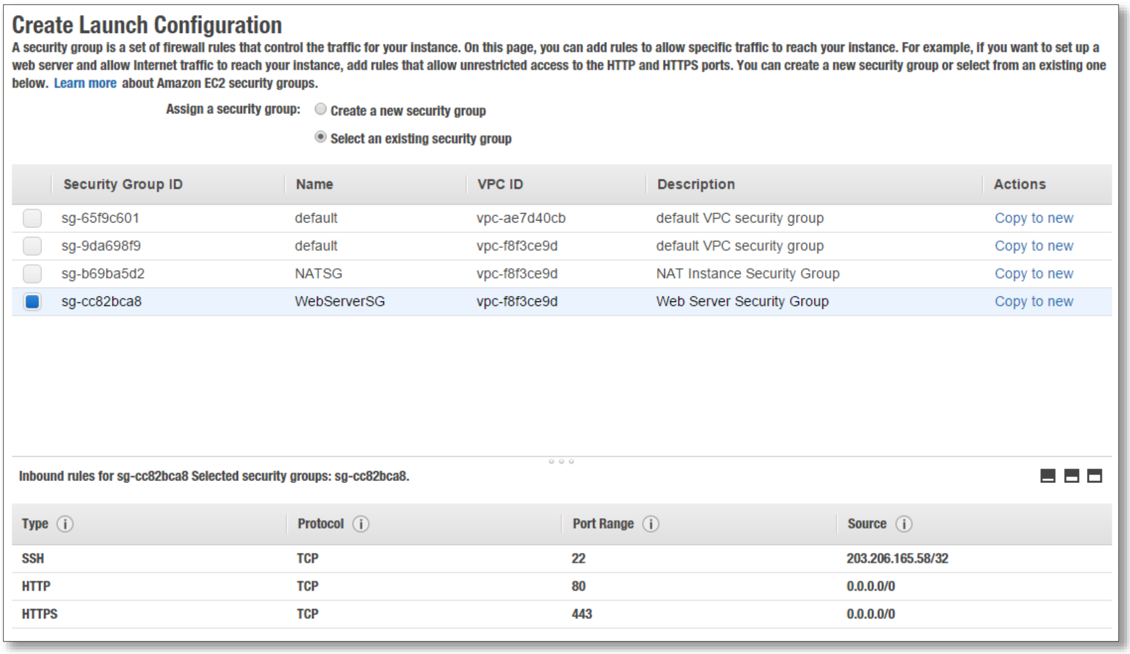
IP Address Type:
● Only assign a public IP address to instances launched in the default VPC and subnet. (default)
○ Assign a public IP address to every instance.
○ Do not assign a public IP address to any instances.
Note: this option only affects instances launched into an Amazon VPC

Click Next

Leave storage the same

Click Next

Select the WebServerSG security group.



Create Launch Configuration

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 ID	Name	VPC ID	Description	Actions
□	sg-65f9c601	default	vpc-ae7d40cb	default VPC security group	Copy to new
□	sg-9da698f9	default	vpc-f8f3ce9d	default VPC security group	Copy to new
□	sg-b69ba5d2	NATSG	vpc-f8f3ce9d	NAT Instance Security Group	Copy to new
☑	sg-cc82bca8	WebServerSG	vpc-f8f3ce9d	Web Server Security Group	Copy to new

Inbound rules for sg-cc82bca8 Selected security groups: sg-cc82bca8.

Type	Protocol	Port Range	Source
SSH	TCP	22	203.206.165.58/32
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

Click Review

Click “Create Launch Configuration”

Select a key pair or create a new one and save it.

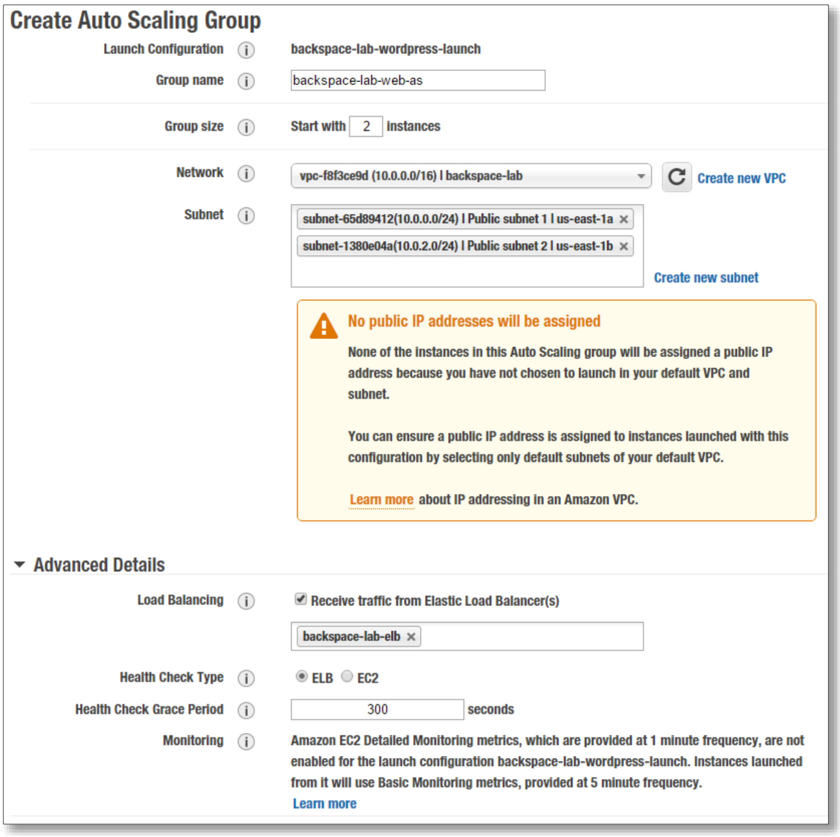
Name your auto scaling group “backspace-lab-web-as”

In the advanced details section select “Receive traffic from Elastic Load Balancer(s)”.

Select the newly created ELB.

Select Health Check Type ELB

Click Next



Create Auto Scaling Group

Launch Configuration: backspace-lab-wordpress-launch

Group name: backspace-lab-web-as

Group size: Start with 2 instances

Network: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

Subnet: subnet-65d89412(10.0.0.0/24) | Public subnet 1 | us-east-1a
subnet-1380e04a(10.0.2.0/24) | Public subnet 2 | us-east-1b

No public IP addresses will be assigned

None of the instances in this Auto Scaling group will be assigned a public IP address because you have not chosen to launch in your default VPC and subnet.

You can ensure a public IP address is assigned to instances launched with this configuration by selecting only default subnets of your default VPC.

Learn more about IP addressing in an Amazon VPC.

Advanced Details

Load Balancing: ☑ Receive traffic from Elastic Load Balancer(s)

backspace-lab-elb

Health Check Type: ● ELB ○ EC2

Health Check Grace Period: 300 seconds

Monitoring: Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration backspace-lab-wordpress-launch. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency.

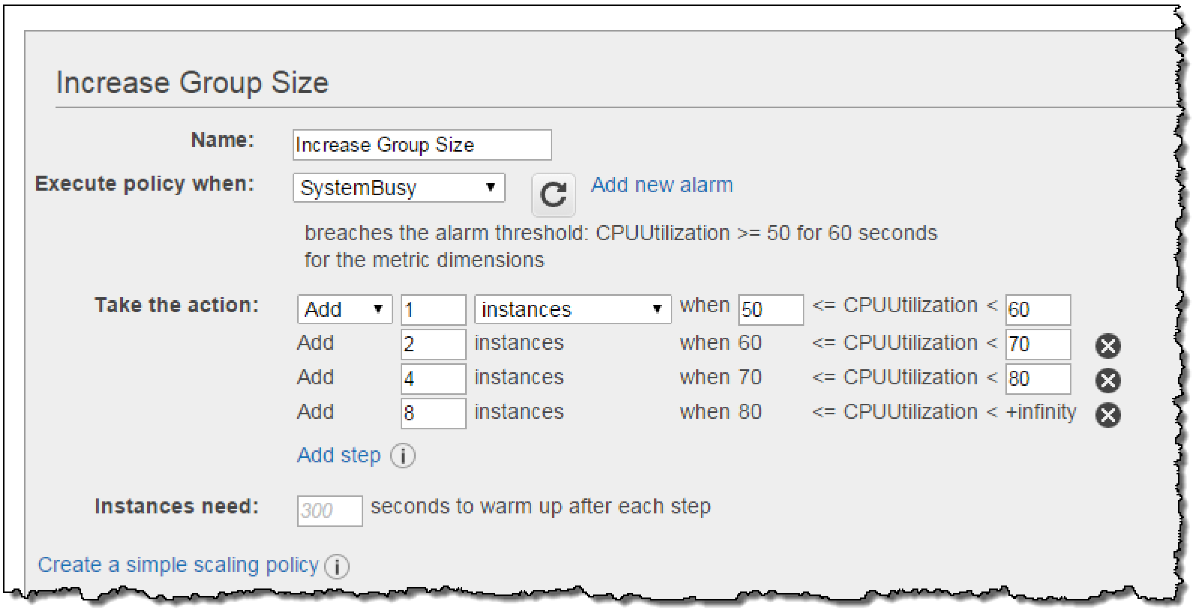
Update from AWS:

On 8 July 2015 AWS anounced the introduction of scaling policies with steps:

Today we are making Auto Scaling even more flexible with the addition of new scaling policies with steps.

Our goal is to allow you to create systems that can do an even better job of responding to rapid and dramatic changes in load. You can now define a scaling policy that will respond to the *magnitude* of the alarm breach in a proportionate and appropriate way. For example, if you try to keep your average CPU utilization below 50% you can have a standard response for a modest breach (50% to 60%), two more for somewhat bigger breaches (60% to 70% and 70% to 80%), and a super-aggressive one for utilization that exceeds 80%.

Here’s how I set this up for my Auto Scaling group:



Increase Group Size

Name: Increase Group Size

Execute policy when: SystemBusy ▾ Add new alarm

breaches the alarm threshold: CPUUtilization >= 50 for 60 seconds
for the metric dimensions

Take the action:

Add ▾	1	instances ▾	when	50	<= CPUUtilization <	60
Add	2	instances	when	60	<= CPUUtilization <	70
Add	4	instances	when	70	<= CPUUtilization <	80
Add	8	instances	when	80	<= CPUUtilization <	+infinity

Add step

Instances need: 300 seconds to warm up after each step

Create a simple scaling policy

In this example I added a fixed number (1, 2, 4, or 8) of instances to the group. I could have chosen to define the policies on a percentage basis, increasing the instance count by (say) 50%, 100%, 150%, and 200% at the respective steps. The empty upper bound in the final step is effectively positive infinity. You can also define a similar set of increasingly aggressive policies for scaling down.

As you can see from the example above, you can also tell Auto Scaling how long it should take for an instance to warm up and be ready to start sharing the load. While this waiting period is in effect, Auto Scaling will include the newly launched instances when it computes the current size of the group. However, during this scaling time, the instances are not factored in to the CloudWatch metrics for the group. This avoids unnecessary scaling while the new instances prepare themselves to take on their share of the load.

Step policies continuously evaluate the alarms during a scaling activity and while unhealthy instances are being replaced with new ones. This allows for faster response to changes in demand. Let’s say the CPU load increases and the first step in the policy is activated. During the specified warm up period (300 seconds in this example), the load might continue to increase and a more aggressive response might be appropriate. Fortunately, Auto Scaling is in violent agreement with this sentiment and will switch in to high gear (and use one of the higher steps) automatically. If you create multiple step scaling policies for the same resource (perhaps based on CPU utilization and inbound network traffic) and both of them fire at approximately the same time, Auto Scaling will look at both policies and choose the one that results in the change of the highest magnitude.

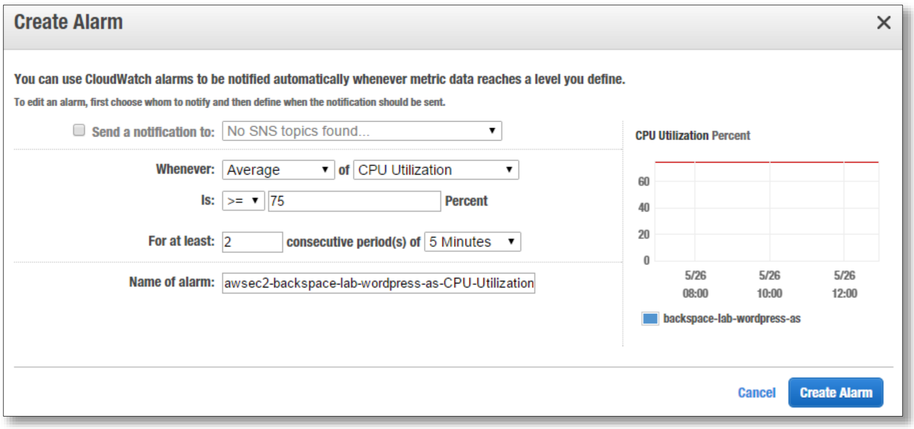
You can also create these new scaling policies using the AWS Command Line Interface (CLI) or the Auto Scaling API.

Select “Use scaling policies to adjust the capacity of this group”

Add alarm to “Increase Group Size

Disable SNS notification.

Alarm at >= 75% CPU utilisation for 2 x 5 minute periods



Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

□ Send a notification to: No SNS topics found...

Whenever: Average of CPU Utilization

Is: >= 75 Percent

For at least: 2 consecutive period(s) of 5 Minutes

Name of alarm: awsec2-backspace-lab-wordpress-as-CPU-Utilization

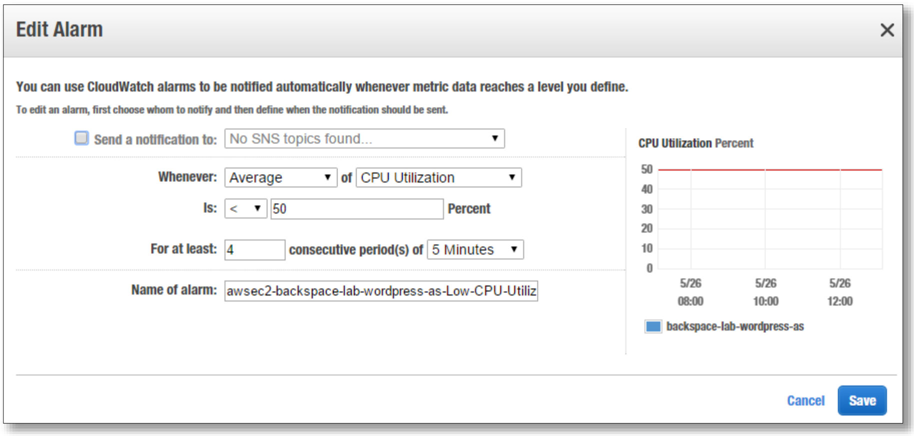
Click Create Alarm

Add alarm to “Decrease Group Size”

Disable SNS notification.

Alarm at < 50% CPU utilisation for 4 x 5 minute periods

Change the name of the alarm from high to low (otherwise you will change the high alarm).



Edit Alarm ×

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

□ Send a notification to: No SNS topics found... ▾

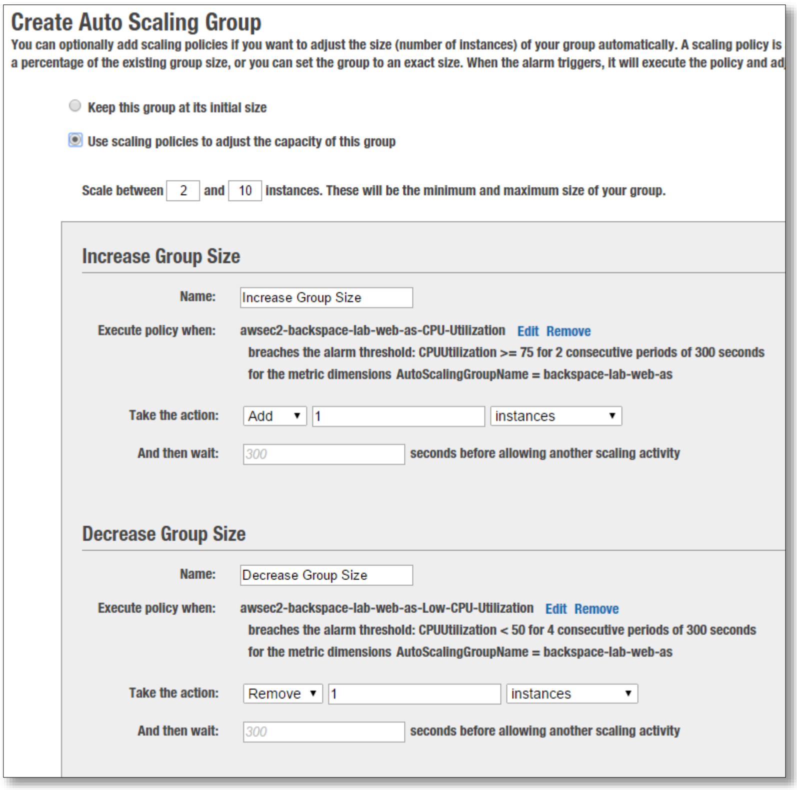
Whenever: Average ▾ of CPU Utilization ▾

Is: < ▾ 50 Percent

For at least: 4 consecutive period(s) of 5 Minutes ▾

Name of alarm: awsec2-backspace-lab-wordpress-as-Low-CPU-Utiliz

Click Save



Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and ad

○ Keep this group at its initial size

● Use scaling policies to adjust the capacity of this group

Scale between 2 and 10 instances. These will be the minimum and maximum size of your group.

Increase Group Size

Name: Increase Group Size

Execute policy when: awsec2-backspace-lab-web-as-CPU-Utilization Edit Remove
breaches the alarm threshold: CPUUtilization >= 75 for 2 consecutive periods of 300 seconds
for the metric dimensions AutoScalingGroupName = backspace-lab-web-as

Take the action: Add ▾ 1 instances ▾

And then wait: 300 seconds before allowing another scaling activity

Decrease Group Size

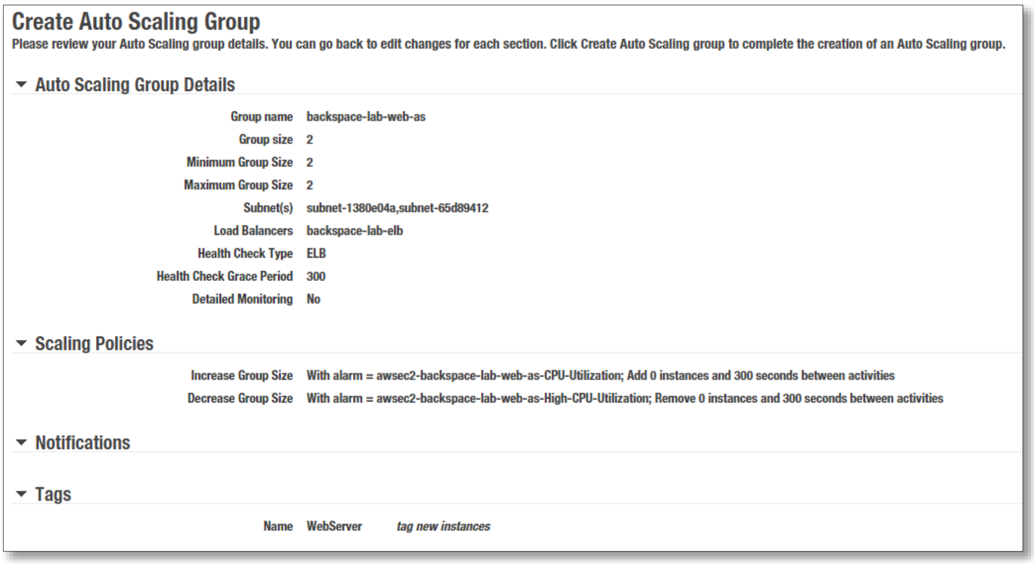
Name: Decrease Group Size

Execute policy when: awsec2-backspace-lab-web-as-Low-CPU-Utilization Edit Remove
breaches the alarm threshold: CPUUtilization < 50 for 4 consecutive periods of 300 seconds
for the metric dimensions AutoScalingGroupName = backspace-lab-web-as

Take the action: Remove ▾ 1 instances ▾

And then wait: 300 seconds before allowing another scaling activity

Click Review



Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click Create Auto Scaling group to complete the creation of an Auto Scaling group.

Auto Scaling Group Details

Group name	backspace-lab-web-as
Group size	2
Minimum Group Size	2
Maximum Group Size	2
Subnet(s)	subnet-1380e04a,subnet-65d89412
Load Balancers	backspace-lab-elb
Health Check Type	ELB
Health Check Grace Period	300
Detailed Monitoring	No

Scaling Policies

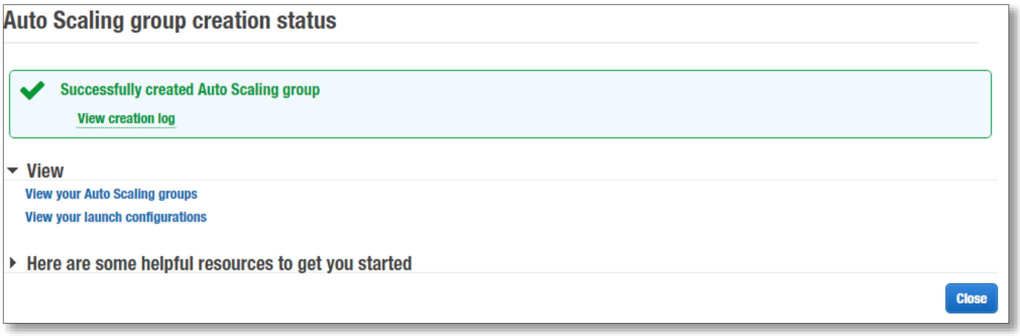
Increase Group Size	With alarm = awsec2-backspace-lab-web-as-CPU-Utilization; Add 0 instances and 300 seconds between activities
Decrease Group Size	With alarm = awsec2-backspace-lab-web-as-High-CPU-Utilization; Remove 0 instances and 300 seconds between activities

Notifications

Tags

Name WebServer *tag new instances*

Click Create Auto Scaling Group



Auto Scaling group creation status

✓ Successfully created Auto Scaling group

View creation log

View

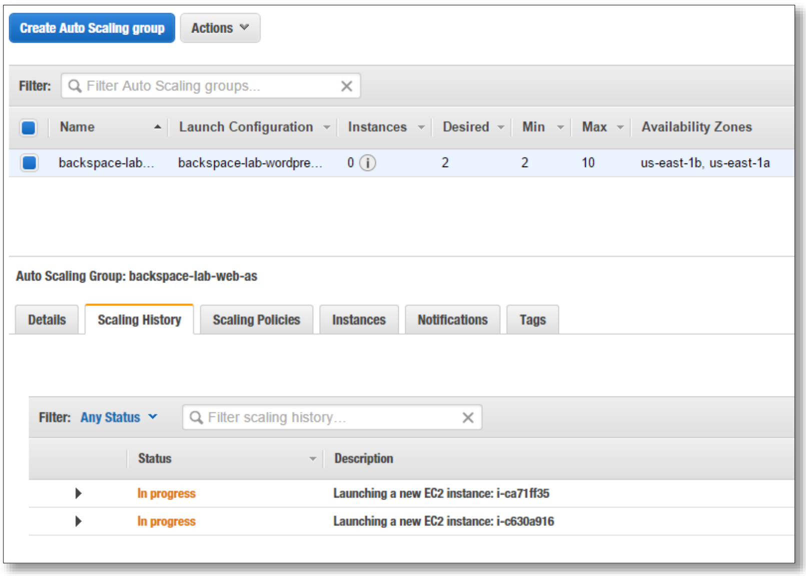
View your Auto Scaling groups

View your launch configurations

Here are some helpful resources to get you started

Click Close

If you click on the “Scaling History” tab you can see the instances being created in the two availability zones.

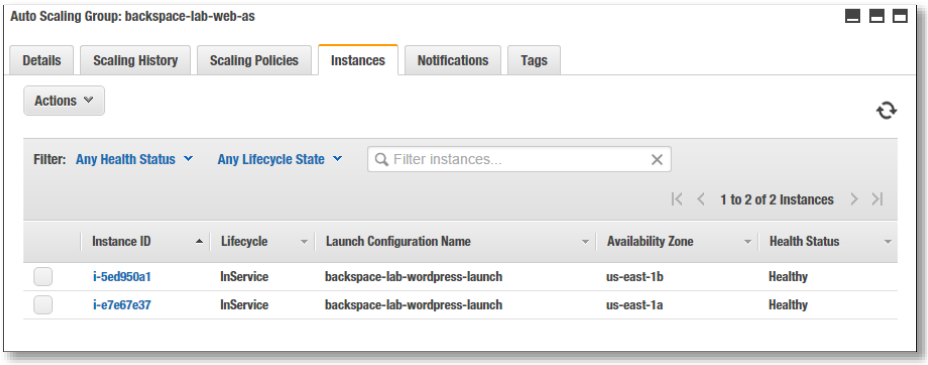


	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones
■	backspace-lab...	backspace-lab-wordpre...	0 ⓘ	2	2	10	us-east-1b, us-east-1a

Auto Scaling Group: backspace-lab-web-as

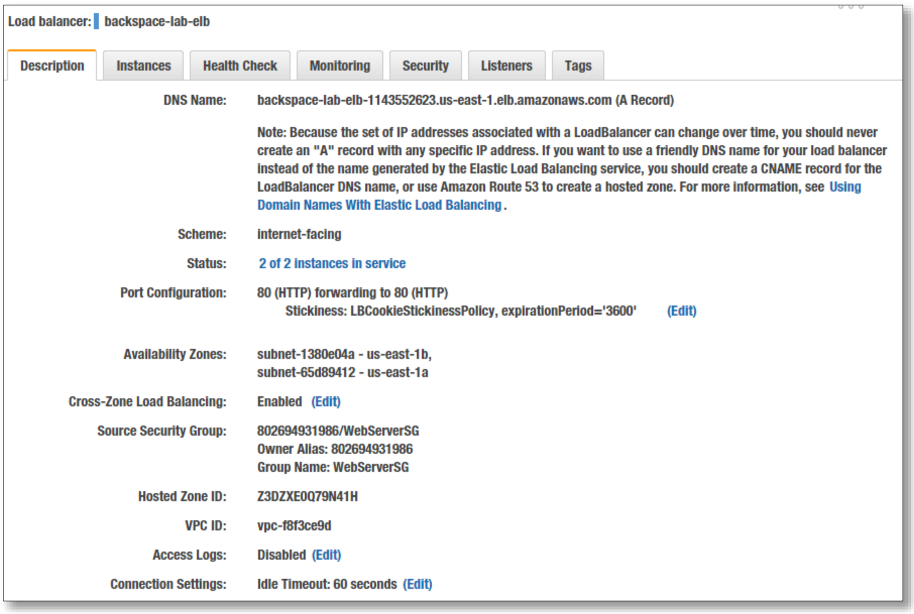
	Status	Description
▸	In progress	Launching a new EC2 instance: i-ca71ff35
▸	In progress	Launching a new EC2 instance: i-c630a916

If you click on the “Instances” tab you can see the instances are healthy.


Auto Scaling Group: backspace-lab-web-as

	Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status
□	i-5ed950a1	InService	backspace-lab-wordpress-launch	us-east-1b	Healthy
□	i-e7e67e37	InService	backspace-lab-wordpress-launch	us-east-1a	Healthy

If you go to the Load Balancer and select the load balancer you will find the DNS name.



Load balancer: backspace-lab-elb

DNS Name: backspace-lab-elb-1143552623.us-east-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see Using Domain Names With Elastic Load Balancing.

Scheme: internet-facing

Status: 2 of 2 instances in service

Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)
Stickiness: LBCookieStickinessPolicy, expirationPeriod='3600' (Edit)

Availability Zones: subnet-1380e04a - us-east-1b,
subnet-65d89412 - us-east-1a

Cross-Zone Load Balancing: Enabled (Edit)

Source Security Group: 802694931986/WebServerSG
Owner Alias: 802694931986
Group Name: WebServerSG

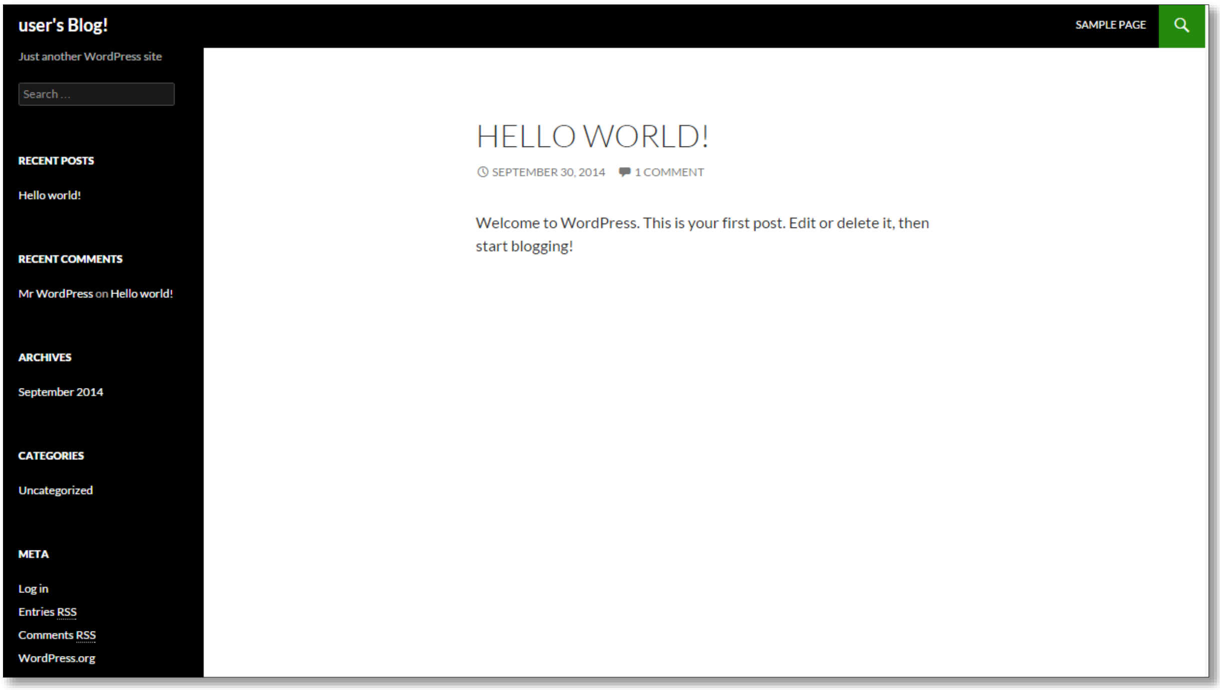
Hosted Zone ID: Z3DZXE0Q79N41H

VPC ID: vpc-f8f3ce9d

Access Logs: Disabled (Edit)

Connection Settings: Idle Timeout: 60 seconds (Edit)

If you point your browser to the ELB DNS name you will see the WordPress application.



user's Blog!

Just another WordPress site

RECENT POSTS

Hello world!

RECENT COMMENTS

Mr WordPress on Hello world!

ARCHIVES

September 2014

CATEGORIES

Uncategorized

META

Log in
Entries RSS
Comments RSS
WordPress.org

HELLO WORLD!

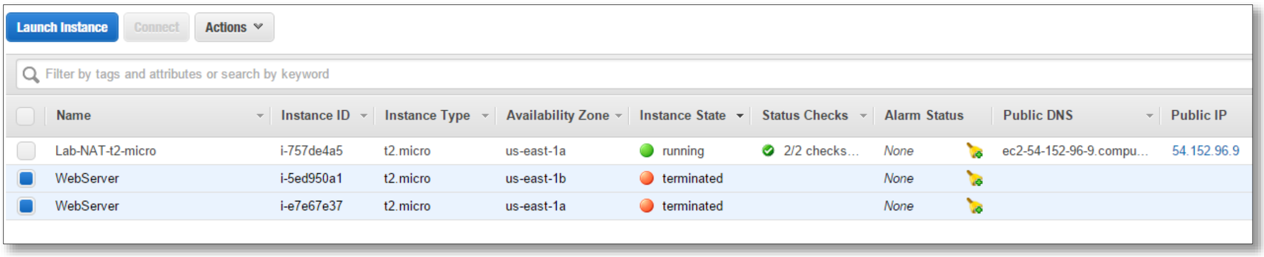
SEPTEMBER 30, 2014 1 COMMENT

Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!

*If you find that instances are failing health checks and being shut down by your ELB regularly then check your public NACL settings are correct for the ELB allowing TCP traffic on ports 1024-65535. If you still have problems extend your ELB health check timeout and time between checks settings.

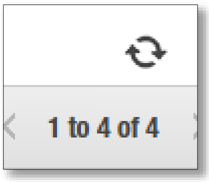
So now let’s check if our auto scaling group is working.

Terminate the 2 Webserver instances

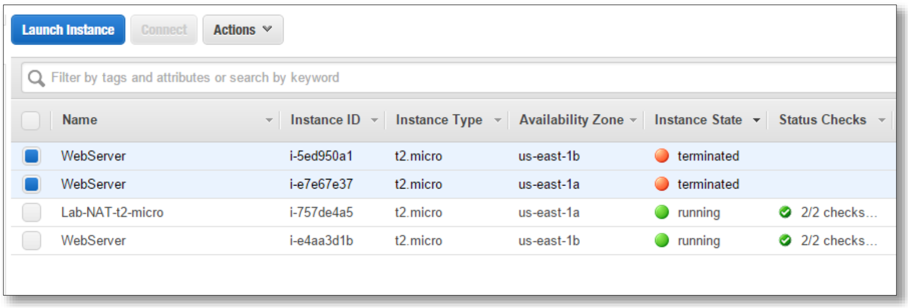


☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
☐	Lab-NAT-t2-micro	i-757de4a5	t2.micro	us-east-1a	● running	✓ 2/2 checks...	None	ec2-54-152-96-9.compu...	54.152.96.9
☑	WebServer	i-5ed950a1	t2.micro	us-east-1b	● terminated		None		
☑	WebServer	i-e7e67e37	t2.micro	us-east-1a	● terminated		None		

Wait for quite a while and click the refresh icon



Soon you will see another two instances automatically created.

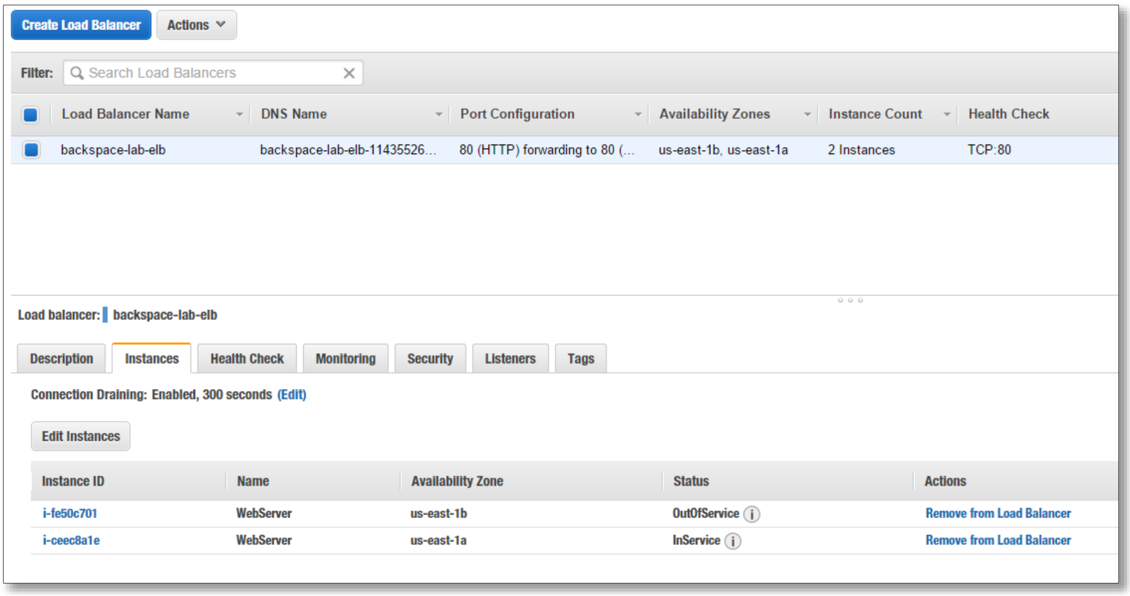


☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
☑	WebServer	i-5ed950a1	t2.micro	us-east-1b	● terminated	
☑	WebServer	i-e7e67e37	t2.micro	us-east-1a	● terminated	
☐	Lab-NAT-t2-micro	i-757de4a5	t2.micro	us-east-1a	● running	✓ 2/2 checks...
☐	WebServer	i-e4aa3d1b	t2.micro	us-east-1b	● running	✓ 2/2 checks...

It will take a bit longer for the instances to complete their start up checks and register with the ELB.

You can check status of this by going back to the load balancer screen and clicking the instances tab.

The status will change from OutOfService to InService.



	Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count	Health Check
■	backspace-lab-elb	backspace-lab-elb-11435526…	80 (HTTP) forwarding to 80 (…	us-east-1b, us-east-1a	2 Instances	TCP:80

Load balancer: backspace-lab-elb

Connection Draining: Enabled, 300 seconds (Edit)

Instance ID	Name	Availability Zone	Status	Actions
i-fe50c701	WebServer	us-east-1b	OutOfService	Remove from Load Balancer
i-ceec8a1e	WebServer	us-east-1a	InService	Remove from Load Balancer

Adding a Multi AZ RDS instance and Read Replica

In this section we will add a Multi-AZ RDS instance deployed across our two private subnets. We will need to create a DB Subnet across two availability zones to utilise Multi-AZ. We also will create a security group and update the WebServerSG security group to allow communication from the Web Server instances. We will then create a read replica in the second AZ.

First let's create a Security Group for our Aurora DB Cluster. We also need to add the security group to the WebServer security group's outbound rules.

Use the following rules for RDS and Web Servers:

DBServerSG: Recommended Rules

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
The security group ID (sg-xxxxxxxx)	All	All	Allow inbound traffic from instances assigned to the same security group.
The ID of your WebServerSG security group	TCP	3306	Allow web servers assigned to WebServerSG MySQL access to database servers assigned to DBServerSG
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allow outbound HTTP access to the Internet (for example, for software updates)
0.0.0.0/0	TCP	443	Allow outbound HTTPS access to the Internet (for example, for software updates)

WebServerSG: Recommended Outbound Rules

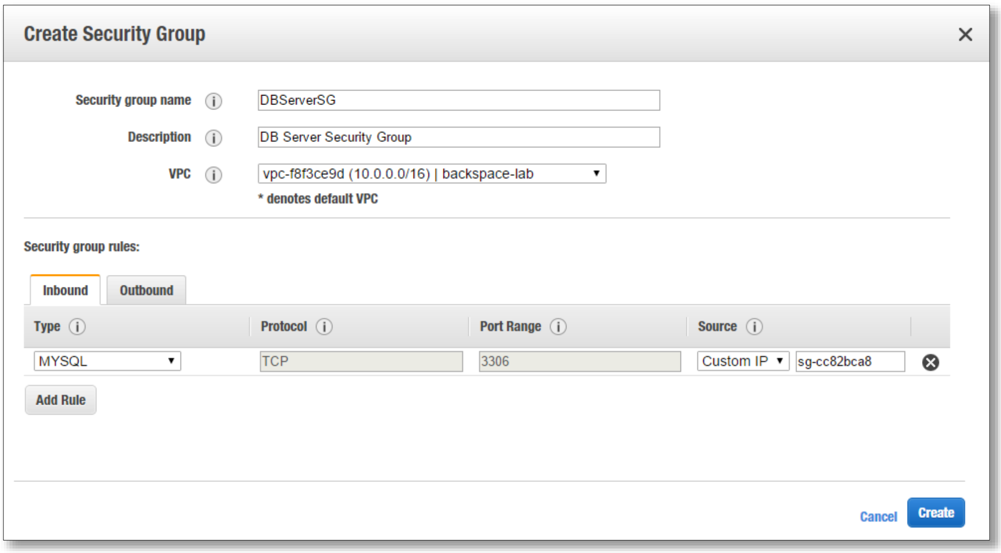
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
The ID of your DBServerSG security group	TCP	3306	Allow outbound MySQL Server access to the database servers assigned to DBServerSG

Go to Security Groups.

Copy the WebServer security group ID.

Click “Create Security Group”

Create an inbound MySQL TCP rule on port 3306 with source the WebServer security group ID.



Create Security Group

Security group name: DBServerSG

Description: DB Server Security Group

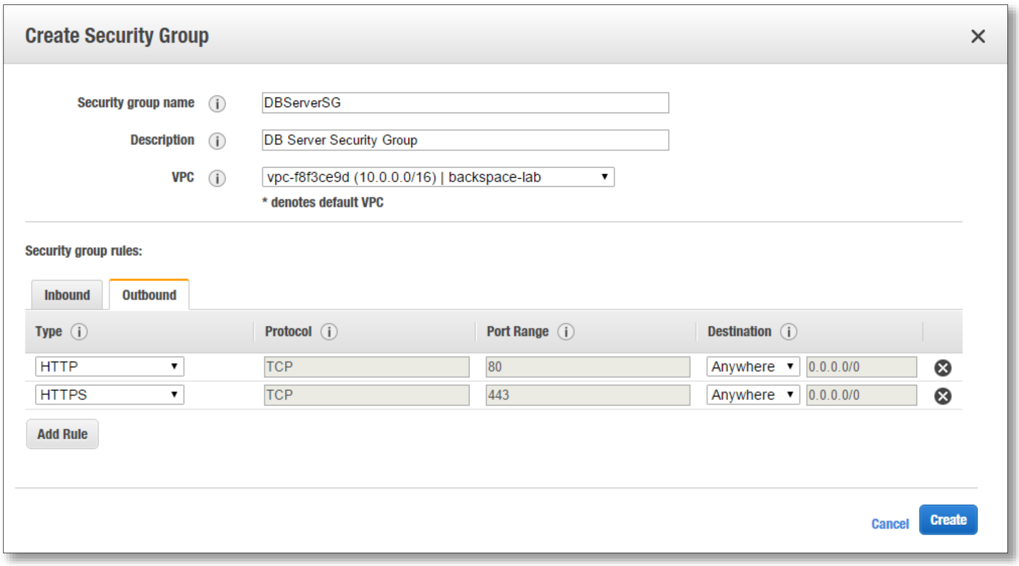
VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

* denotes default VPC

Security group rules:

Type	Protocol	Port Range	Source
MYSQL	TCP	3306	Custom IP sg-cc82bca8

Create the outbound rules for ports 80 and 443.



Security group name: DBServerSG

Description: DB Server Security Group

VPC: vpc-f8f3ce9d (10.0.0.0/16) | backspace-lab

* denotes default VPC

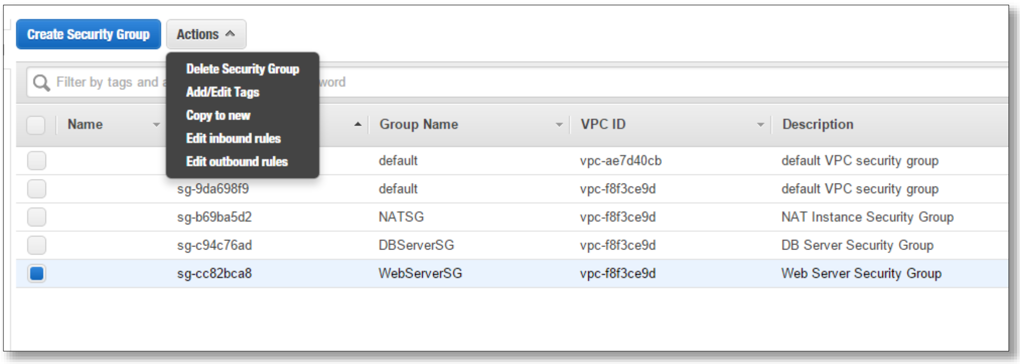
Security group rules:

Type	Protocol	Port Range	Destination	
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0

Click Create

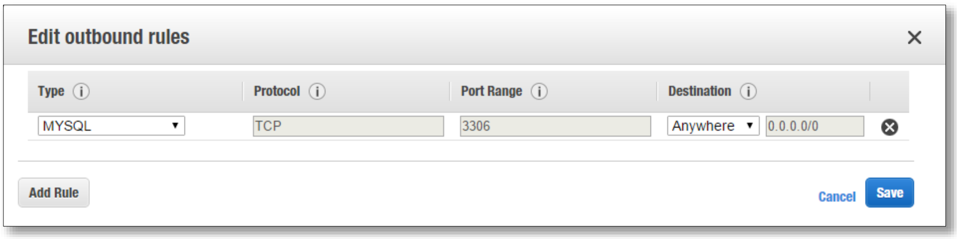
Now copy the newly created DBServerSG security group ID.

Now select and edit the outbound rules of the DBServerSG security group.



☐	Name	Group ID	Group Name	VPC ID	Description
☐			default	vpc-ae7d40cb	default VPC security group
☐		sg-9da698f9	default	vpc-f8f3ce9d	default VPC security group
☐		sg-b69ba5d2	NATSG	vpc-f8f3ce9d	NAT Instance Security Group
☐		sg-c94c76ad	DBServerSG	vpc-f8f3ce9d	DB Server Security Group
☑		sg-cc82bca8	WebServerSG	vpc-f8f3ce9d	Web Server Security Group

Create an outbound MySQL TCP rule on port 3306 with destination the DBServer security group ID.



Type	Protocol	Port Range	Destination	
MYSQL	TCP	3306	Anywhere	0.0.0.0/0

Click save

The last thing that you need before you can create an Aurora DB cluster is a DB subnet group. Your RDS DB subnet group identifies the subnets that your DB cluster will use from the VPC that you created in the previous steps. Your DB subnet group must include at least two subnets in at least two Availability Zones.

Go to the RDS Console.

Select the Subnet Groups page

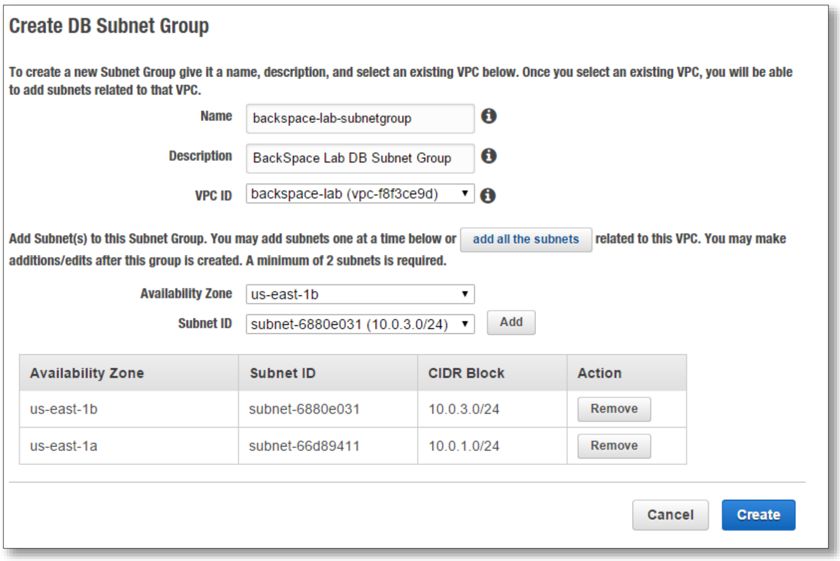
Click create DB Subnet Group

Name the DB subnet group backspace-lab-subnetgroup

Select the backspace-lab VPC

Add the 10.0.1.0/24 private subnet 1

Add the 10.0.3.0/24 private subnet 2



Create DB Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

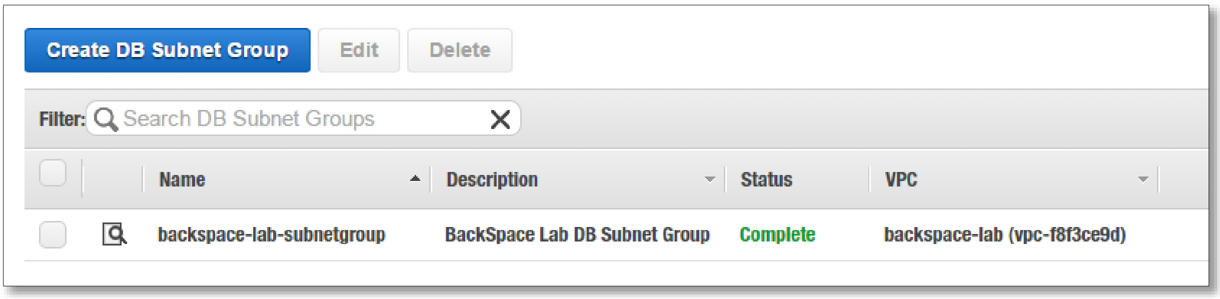
Name	backspace-lab-subnetgroup
Description	BackSpace Lab DB Subnet Group
VPC ID	backspace-lab (vpc-f8f3ce9d)

Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Availability Zone	us-east-1b
Subnet ID	subnet-6880e031 (10.0.3.0/24)

Availability Zone	Subnet ID	CIDR Block	Action
us-east-1b	subnet-6880e031	10.0.3.0/24	Remove
us-east-1a	subnet-66d89411	10.0.1.0/24	Remove

Click Create (you may have to click the refresh icon to see the subnet group)

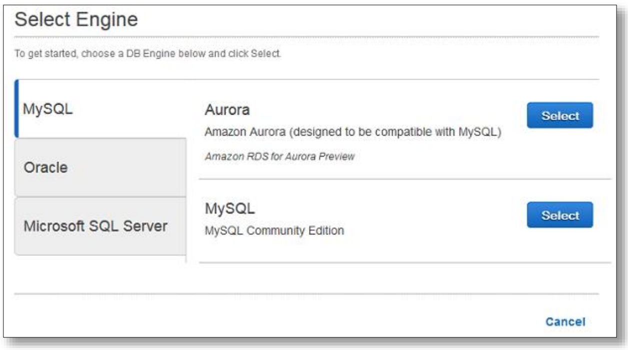


Name	Description	Status	VPC
backspace-lab-subnetgroup	BackSpace Lab DB Subnet Group	Complete	backspace-lab (vpc-f8f3ce9d)

Go to the RDS Instances page

Click “Launch DB Instance”

Select one of the MySQL engines. Aurora at the time of writing is in preview and requires you to sign up beforehand. Aurora provides up to five times better performance than MySQL and will become the preferred option when fully released by AWS.



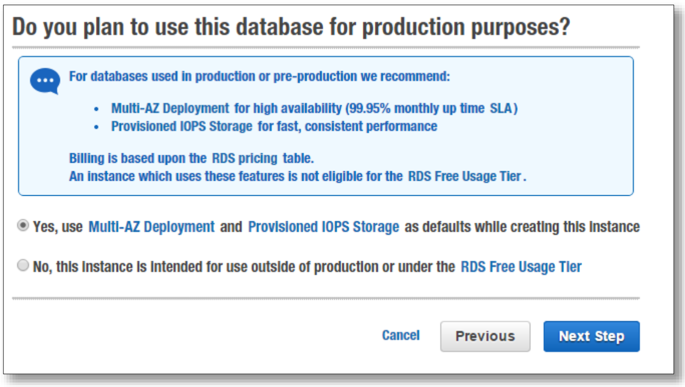
Select Engine

To get started, choose a DB Engine below and click Select.

Aurora
Amazon Aurora (designed to be compatible with MySQL)
Amazon RDS for Aurora Preview

MySQL
MySQL Community Edition

Select Yes for Multi-AZ deployment



Do you plan to use this database for production purposes?

For databases used in production or pre-production we recommend:

- Multi-AZ Deployment for high availability (99.95% monthly up time SLA)
- Provisioned IOPS Storage for fast, consistent performance

Billing is based upon the RDS pricing table.
An instance which uses these features is not eligible for the RDS Free Usage Tier.

● Yes, use Multi-AZ Deployment and Provisioned IOPS Storage as defaults while creating this instance

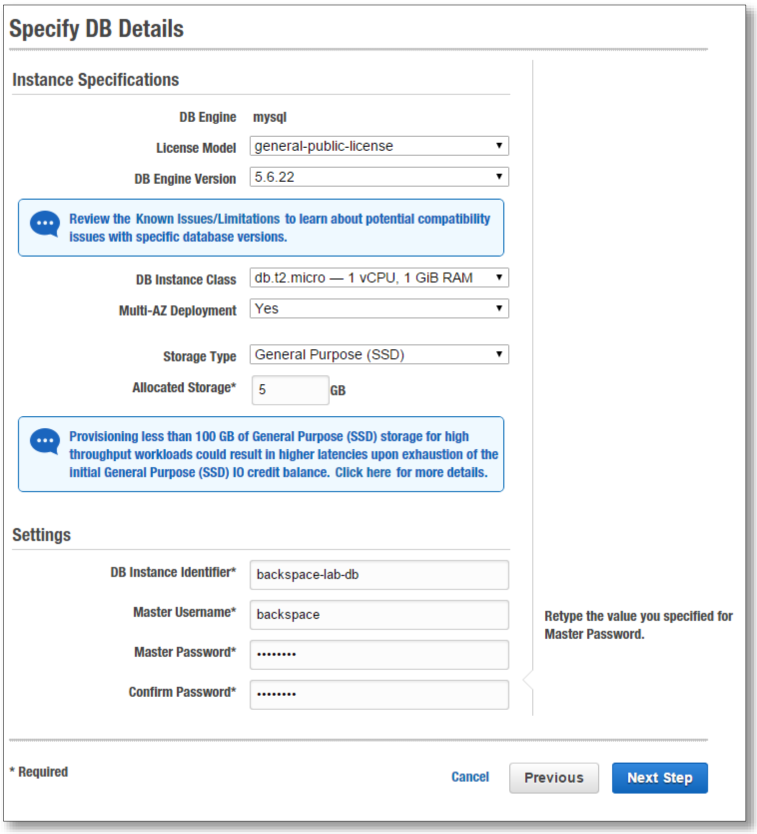
○ No, this instance is intended for use outside of production or under the RDS Free Usage Tier

Select a t2 micro instance size

Give the DB an identifier backspace-lab-db

Provide a username and password to use with the database.

Click Next



Specify DB Details

Instance Specifications

DB Engine	mysql
License Model	general-public-license
DB Engine Version	5.6.22

Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

DB Instance Class	db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment	Yes
Storage Type	General Purpose (SSD)
Allocated Storage*	5 GB

Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Click here for more details.

Settings

DB Instance Identifier*	backspace-lab-db
Master Username*	backspace
Master Password*	••••••••
Confirm Password*	••••••••

Retype the value you specified for Master Password.

* Required

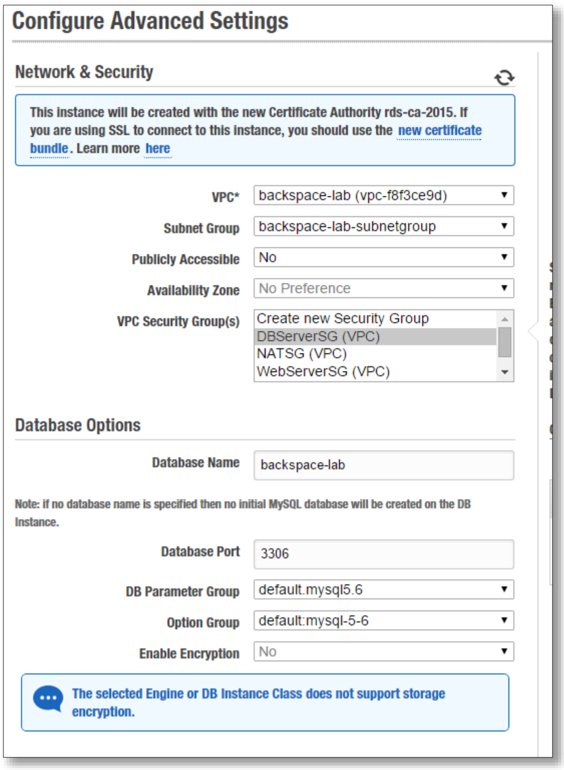
Select the backspace-lab VPC

Select the backspace-lab-subnetgroup

Select the DBServerDG security group.

Use BackSpaceLabDB for the database name

Leave other settings default



Configure Advanced Settings

Network & Security

This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the new certificate bundle. Learn more here

VPC*	backspace-lab (vpc-f8f3ce9d)
Subnet Group	backspace-lab-subnetgroup
Publicly Accessible	No
Availability Zone	No Preference
VPC Security Group(s)	Create new Security Group DBServerSG (VPC) NATSG (VPC) WebServerSG (VPC)

Database Options

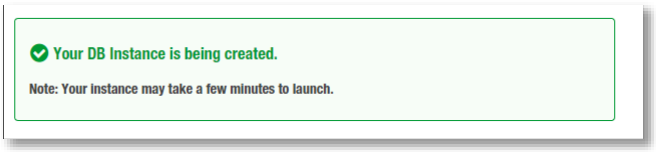
Database Name	backspace-lab

Note: if no database name is specified then no initial MySQL database will be created on the DB instance.

Database Port	3306
DB Parameter Group	default.mysql5.6
Option Group	default:mysql-5-6
Enable Encryption	No

The selected Engine or DB Instance Class does not support storage encryption.

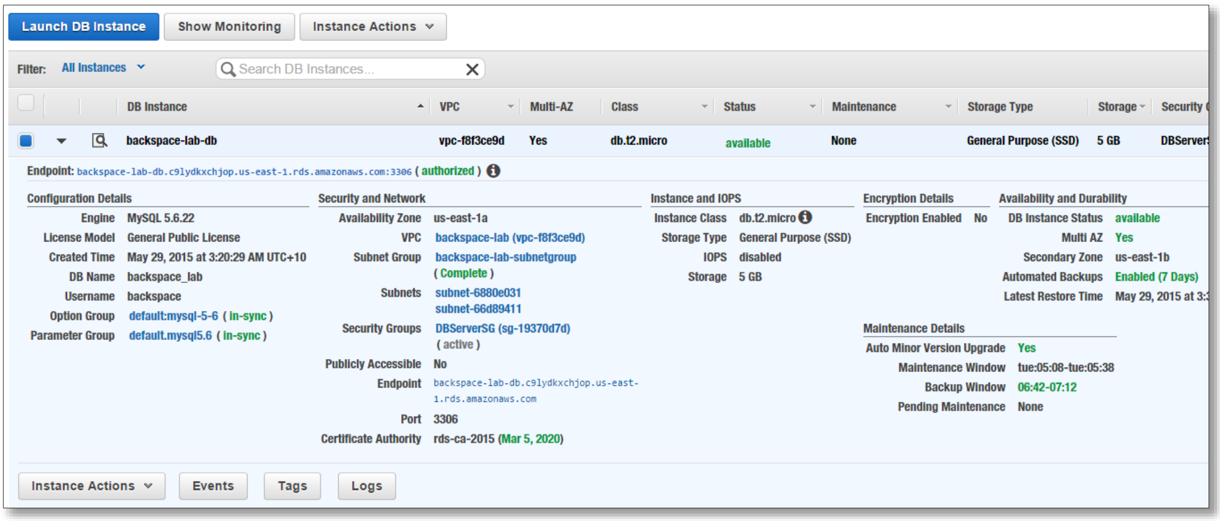
Click Launch DB Instance



Your DB Instance is being created.

Note: Your instance may take a few minutes to launch.

After quite some time the instance will be set up.



DB Instance	VPC	Multi-AZ	Class	Status	Maintenance	Storage Type	Storage	Security (
backspace-lab-db	vpc-f8f3ce9d	Yes	db.t2.micro	available	None	General Purpose (SSD)	5 GB	DBServer

Endpoint: backspace-lab-db.c9lydkxchjop.us-east-1.rds.amazonaws.com:3306 (authorized)

Configuration Details

Engine	MySQL 5.6.22
License Model	General Public License
Created Time	May 29, 2015 at 3:20:29 AM UTC+10
DB Name	backspace_lab
Username	backspace
Option Group	default:mysql-5-6 (in-sync)
Parameter Group	default.mysql5.6 (in-sync)

Security and Network

Availability Zone	us-east-1a
VPC	backspace-lab (vpc-f8f3ce9d)
Subnet Group	backspace-lab-subnetgroup (Complete)
Subnets	subnet-6880e031 subnet-66d89411
Security Groups	DBServerSG (sg-19370d7d) (active)
Publicly Accessible	No
Endpoint	backspace-lab-db.c9lydkxchjop.us-east-1.rds.amazonaws.com
Port	3306
Certificate Authority	rds-ca-2015 (Mar 5, 2020)

Instance and IOPS

Instance Class	db.t2.micro
Storage Type	General Purpose (SSD)
IOPS	disabled
Storage	5 GB

Encryption Details

Encryption Enabled	No

Availability and Durability

DB Instance Status	available
Multi AZ	Yes
Secondary Zone	us-east-1b
Automated Backups	Enabled (7 Days)
Latest Restore Time	May 29, 2015 at 3:3

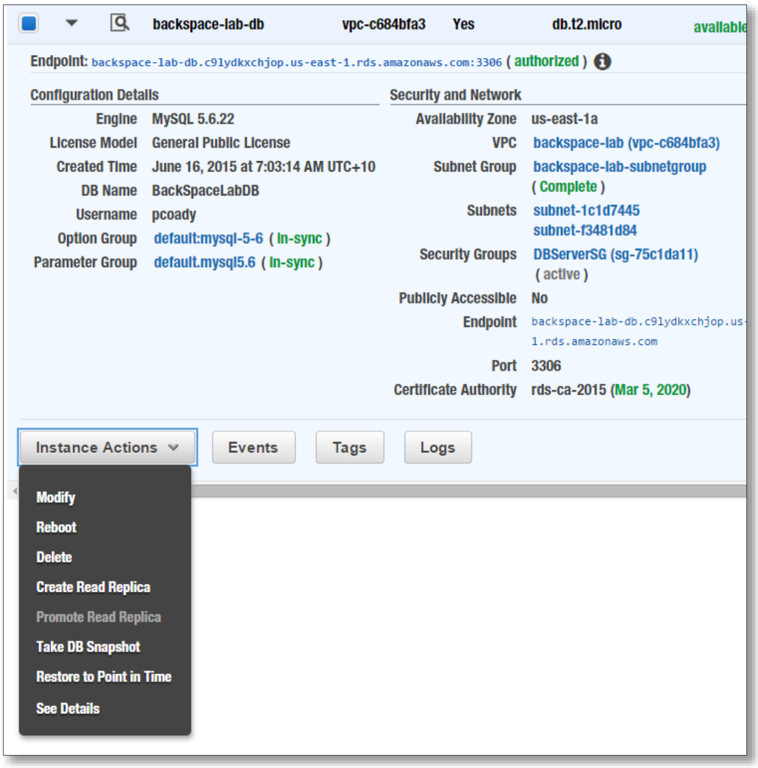
Maintenance Details

Auto Minor Version Upgrade	Yes
Maintenance Window	tue:05:08-tue:05:38
Backup Window	06:42-07:12
Pending Maintenance	None

Here you can see the Instance Endpoint. This is the database endpoint that would be used in your Web Server instances WordPress configuration.

Now let's create a read replica in another availability zone to take some of the load off our master database. We can also promote this read replica to be the master in situations where we are having problems with the master database.

Click on Instance Actions at the bottom of the details and select Create Read Replica.



backspace-lab-db	vpc-c684bfa3	Yes	db.t2.micro	available

Endpoint: backspace-lab-db.c9lydkxchjop.us-east-1.rds.amazonaws.com:3306 (authorized)

Configuration Details

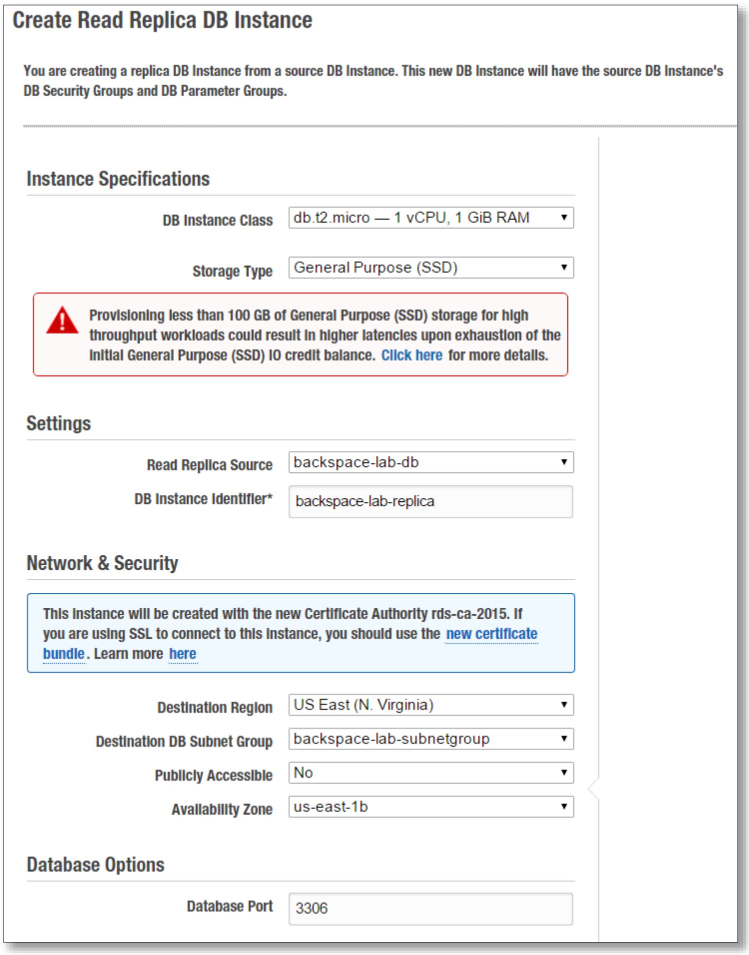
Engine	MySQL 5.6.22
License Model	General Public License
Created Time	June 16, 2015 at 7:03:14 AM UTC+10
DB Name	BackSpaceLabDB
Username	pcoady
Option Group	default:mysql-5-6 (in-sync)
Parameter Group	default.mysql5.6 (in-sync)

Security and Network

Availability Zone	us-east-1a
VPC	backspace-lab (vpc-c684bfa3)
Subnet Group	backspace-lab-subnetgroup (Complete)
Subnets	subnet-1c1d7445 subnet-f3481d84
Security Groups	DBServerSG (sg-75c1da11) (active)
Publicly Accessible	No
Endpoint	backspace-lab-db.c9lydkxchjop.us- 1.rds.amazonaws.com
Port	3306
Certificate Authority	rds-ca-2015 (Mar 5, 2020)

Name the replica backspace-lab-replica

Select availability zone us-east-1b.



Create Read Replica DB Instance

You are creating a replica DB Instance from a source DB Instance. This new DB Instance will have the source DB Instance's DB Security Groups and DB Parameter Groups.

Instance Specifications

DB Instance Class: db.t2.micro — 1 vCPU, 1 GiB RAM

Storage Type: General Purpose (SSD)

Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Click here for more details.

Settings

Read Replica Source: backspace-lab-db

DB Instance Identifier*: backspace-lab-replica

Network & Security

This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the new certificate bundle. Learn more here

Destination Region: US East (N. Virginia)

Destination DB Subnet Group: backspace-lab-subnetgroup

Publicly Accessible: No

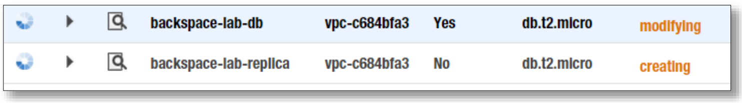
Availability Zone: us-east-1b

Database Options

Database Port: 3306

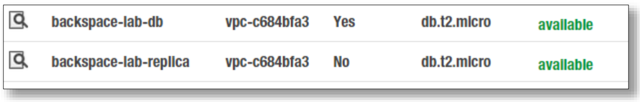
Click Create Read Replica

Your Read Replica will initially show creating and your master database will show modifying



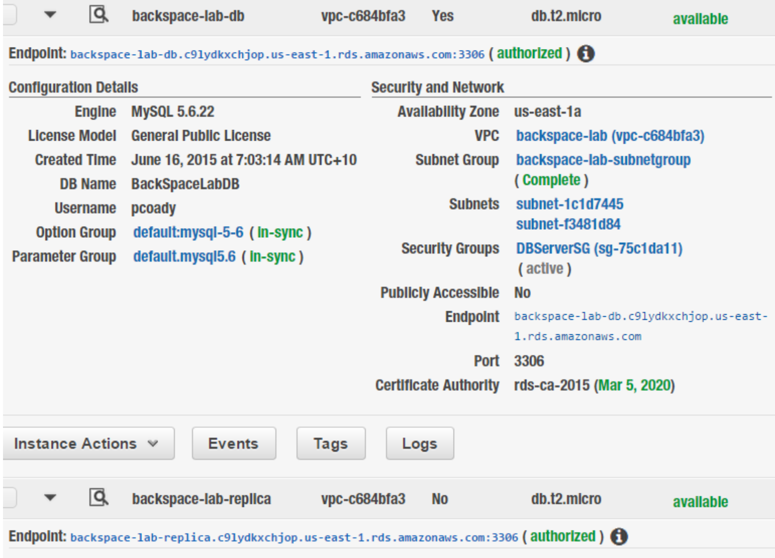
backspace-lab-db	vpc-c684bfa3	Yes	db.t2.micro	modifying
backspace-lab-replica	vpc-c684bfa3	No	db.t2.micro	creating

After a short amount of time they will both be available.



backspace-lab-db	vpc-c684bfa3	Yes	db.t2.micro	available
backspace-lab-replica	vpc-c684bfa3	No	db.t2.micro	available

Now expand the details of both instances



backspace-lab-db	vpc-c684bfa3	Yes	db.t2.micro	available

Endpoint: backspace-lab-db.c91ydkxchjop.us-east-1.rds.amazonaws.com:3306 (authorized)

Configuration Details

Engine	MySQL 5.6.22
License Model	General Public License
Created Time	June 16, 2015 at 7:03:14 AM UTC+10
DB Name	BackSpaceLabDB
Username	pcoady
Option Group	default:mysql-5-6 (in-sync)
Parameter Group	default.mysql5.6 (in-sync)

Security and Network

Availability Zone	us-east-1a
VPC	backspace-lab (vpc-c684bfa3)
Subnet Group	backspace-lab-subnetgroup (Complete)
Subnets	subnet-1c1d7445 subnet-f3481d84
Security Groups	DBServerSG (sg-75c1da11) (active)
Publicly Accessible	No
Endpoint	backspace-lab-db.c91ydkxchjop.us-east-1.rds.amazonaws.com
Port	3306
Certificate Authority	rds-ca-2015 (Mar 5, 2020)

backspace-lab-replica	vpc-c684bfa3	No	db.t2.micro	available

Endpoint: backspace-lab-replica.c91ydkxchjop.us-east-1.rds.amazonaws.com:3306 (authorized)

If you look at the endpoints for both instances they are different. This creates a problem for our Wordpress application that is launched in Multi-Azs. The application will not know which endpoint to use.

You may think that we can just add an internal ELB in front of our RDS instances as we have done with our WordPress EC2 instances and then reference the ELB endpoint. Unfortunately this is not possible with AWS as you can only front EC2 instances with an ELB not RDS instances.

There are two solutions to the problem:

1. Create a HAProxy instance in front of our RDS instances and use this to serve traffic to our RDS instances. You can then reference the endpoint for the HAProxy instance. This is the preferred solution of AWS. Implementing this is beyond the scope of the Architect Associate level but you still need to be aware of it as a solution.
2. Create a script that checks the EC2 instance metadata on startup to identify the availability zone. The Wordpess application will then be able to use the correct endpoint for the availability zone it is launched into. Implementing this is again beyond the scope of the Architect Associate level.

Finishing up the lab

Now that you have completed the lab, make sure that you stop or terminate all the EC2 and RDS instances so that you don’t get billed for them.

You will first need to delete (using the console) or suspend (using the CLI) the auto scaling group otherwise the EC2 instances will be launched again after termination.