

Valaxy DevOps Training Assignment 5

The Goal of this project is to deploy a scalable, highly available and secured Java application on a 3-tier architecture and provide application access to the end users from the public internet.

Pre-Requisites

1. Create an AWS Free Tier account
2. Create a Bitbucket account and a repository to keep your Java source code.
3. Migrate this Java [Source Code](#) to your own Bitbucket repository
Refer to my solution here on how to migrate Instructor's repo
<https://github.com/yemisi/Valaxytraining.git>
4. Create account in Sonarcloud.
5. Create account in Jfrog cloud.

Pre-Deployment

Images can be created using amazon-linux 2 instances launched within a default VPC

1. Create Global AMI

- 1.) AWS CLI

This is Installed by default on amazon linux 2 AMI

- 2.) CloudWatch agent

```
# sudo su && yum -y install amazon-cloudwatch-agent
# systemctl status amazon-cloudwatch-agent.service
```

- 3.) Install AWS SSM agent

This is installed by default on amazon linux 2 AMI. You can verify session manager access by attaching iam role with aws managed policy named *AmazonSSMFullAccess* and connect from the EC2 aws console

2. Create Golden AMI using Global AMI for Nginx application

- 1.) Install Nginx

After installation ensure to enable the service so it starts on reboot

```
#sudo su && amazon-linux-extras install -y nginx1
# systemctl start nginx && systemctl enable nginx && systemctl status nginx
```

- 2.) Push custom memory metrics to Cloudwatch.

- Pushing Custom metrics requires installation and configuration of the cloud watch agent.
- Execute the wizard below to install and cloudwatch agent.
- Accept most of the defaults. Exceptions can include selecting cwagent user or selecting the standard default metics config when prompted

```
# /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

- Start the cloud watch agent specifying the Json config file

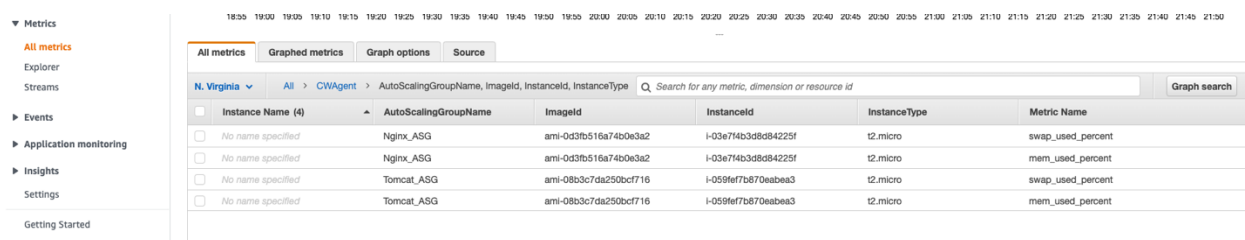
```
# /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a
fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
-s
```

- Verify the cloud-watch agent is running

```
# systemctl status amazon-cloudwatch-agent.service
```

To verify metrics will be pushed, attach an IAM role with this AWS managed policy name **CloudWatchAgentServerPolicy** to the instance.

A custom namespace with name **CWAgent** will be created and ec2 instances will be seen with custom metrics in Cloudwatch. Screenshot sample below shows memory metrics pushed from ASG instances



3. Create Golden AMI using Global AMI for Apache Tomcat application

1.) Install Apache Tomcat

- Install and unzip downloaded binary package


```
# sudo su && cd /opt
# wget https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.73/bin/apache-tomcat-8.5.73.zip && unzip apache-tomcat-8.5.73.zip
```

2.) Configure Tomcat as Systemd service

- create this unit file with the contents below


```
# vi /etc/systemd/system/tomcat.service
```

```
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application
After=syslog.target network.target

[Service]
Type=forking

ExecStart='/opt/apache-tomcat-8.5.73/bin/startup.sh'
ExecStop=/bin/kill -15 $MAINPID

User=root

RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

- Ensure all files are executable in /opt/apache-tomcat-8.5.73/bin

```
# chmod +x /opt/apache-tomcat-8.5.73/bin/*
```

- Reload all unit config files, start and enable the tomcat service for reboots

```
# systemctl daemon-reload && systemctl start tomcat.service && systemctl enable tomcat.service
```

3. Install JDK 11

```
# sudo su && amazon-linux-extras install -y java-openjdk11 && java --version
```

4. Push custom memory metrics to Cloudwatch.

Same steps as shown for nginx in step 2 above

4. Create Golden AMI using Global AMI for Apache Maven Build Tool

1. Install Apache Maven

Download and unzip installation package for installation

```
# sudo su && cd /opt/ && wget https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.zip && unzip apache-maven-3.8.4-bin.zip
```

2. Install Git

```
# yum install -y git
```

3. Install JDK 11

```
# amazon-linux-extras install -y java-openjdk11 && java --version
```

4. Update Maven Home to the system PATH environment variable

```
export PATH='/opt/apache-maven-3.8.4': '/opt/apache-maven-3.8.4/bin': $PATH
```

Also append to bashrc file

```
# vi ~/.bashrc
```

Verify mvn command executes without explicit path to executable command

```
# mvn --version
```

Four custom AMIs should be created as shown below with commands above.

Navigate on EC2 service console via 'Actions' -> 'Image&Templates' -> 'Create Image' to create images from instances

Amazon Machine Images (AMIs) (4) Info									
Owned by me		Search						< 1 > ©	
<input type="checkbox"/>	Name	AMI ID	AMI name	Visibility	Status	Creation date	Platform	Root device type	
<input type="checkbox"/>	Global_AMI	ami-04061ac453e460090	Global_AMI	Private	Available	2021/12/12 02:23 GMT-5	Linux/UNIX	ebs	
<input type="checkbox"/>	Nginx_GoldenAMI	ami-0464e0cf3eb32f745	Nginx_GoldenAMI	Private	Available	2021/12/12 03:31 GMT-5	Linux/UNIX	ebs	
<input type="checkbox"/>	Tomcat_GoldenAMI	ami-072bb2903cd82de6b	Tomcat_GoldenAMI	Private	Available	2021/12/12 04:28 GMT-5	Linux/UNIX	ebs	
<input type="checkbox"/>	Maven_GoldenAMI	ami-08dbaa5d24ced81ec	Maven_GoldenAMI	Private	Available	2021/12/12 05:07 GMT-5	Linux/UNIX	ebs	

VPC (Network Setup)

1. Build VPC network (192.168.0.0/16) for Bastion Host deployment as per the architecture shown above.
2. Build VPC network (172.32.0.0/16) for deploying Highly Available and Auto Scalable application servers as per the architecture shown above.
ProdVPC is shown here
3. Create NAT Gateway in Public Subnet and update Private Subnet associated Route Table accordingly to route the default traffic to NAT for outbound internet connection.
4. Create Transit Gateway and associate both VPCs to the Transit Gateway for private communication.
5. Create Internet Gateway for each VPC and update Public Subnet associated Route Table accordingly to route the default traffic to IGW for inbound/outbound internet connection.

Bastion

1. Deploy Bastion Host in the Public Subnet with EIP associated.
2. Create Security Group allowing port 22 from public internet

AWS INFRASTRUCTURE SETUP SOLUTION

Bastion and Prod VPC configuration is shown below with DNS hostnames enabled.

The screenshot displays the AWS Management Console interface for VPCs. At the top, there's a section titled 'Your VPCs (1/2)' with a search bar and filters. Below this, a table lists two VPCs: Bastion_VPC and Prod_VPC. The table columns include Name, VPC ID, State, IPv4 CIDR, DHCP options set, Main route table, Main network ACL, and Default VPC. Bastion_VPC has VPC ID vpc-0aa2872e25e5b4fb3, State Available, IPv4 CIDR 192.168.0.0/16, DHCP options set dopt-43012a39, Main route table rtb-06005a2a2955881c8, Main network ACL acl-0f04d2c46e36c5ea0, and is not the Default VPC. Prod_VPC has VPC ID vpc-04d43f45127455847, State Available, IPv4 CIDR 172.32.0.0/16, DHCP options set dopt-43012a39, Main route table rtb-008aa3f8fd25a6cca / Default_RT, Main network ACL acl-01b766525618d0916, and is not the Default VPC.

Below the table, the details for vpc-0aa2872e25e5b4fb3 / Bastion_VPC are shown. The 'Details' tab is selected, displaying the following information:

Details			
VPC ID	State	DNS hostnames	DNS resolution
vpc-0aa2872e25e5b4fb3	Available	Enabled	Enabled
Tenancy	DHCP options set	Main route table	Main network ACL
Default	dopt-43012a39	rtb-06005a2a2955881c8	acl-0f04d2c46e36c5ea0
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	192.168.0.0/16	-	-

Prod_VPC configuration along with CIDR range is shown below

Your VPCs (1/2) info

Filter VPCs

VPC ID: vpc-0aa2872e25e5b4fb3 X VPC ID: vpc-04d43f45127455847 X Clear filters

	Name	VPC ID	State	IPv4 CIDR	DHCP options set	Main route table	Main network ACL	Default VPC
<input type="checkbox"/>	Bastion_VPC	vpc-0aa2872e25e5b4fb3	Available	192.168.0.0/16	dopt-43012a39	rtb-06005a2a2955881c8	acl-0f04d2c46e36c5ea0	No
<input checked="" type="checkbox"/>	Prod_VPC	vpc-04d43f45127455847	Available	172.32.0.0/16	dopt-43012a39	rtb-008aa3f8fd25a6cca / Default_RT	acl-01b766525618d0916	No

vpc-04d43f45127455847 / Prod_VPC

Details CIDRs Flow logs Tags

Details

VPC ID vpc-04d43f45127455847	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-43012a39	Main route table rtb-008aa3f8fd25a6cca / Default_RT	Main network ACL acl-01b766525618d0916
Default VPC No	IPv4 CIDR 172.32.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -

Create and attach IGW to both VPCs

Internet gateways (2) info

Filter internet gateways

VPC ID: vpc-0aa2872e25e5b4fb3 X VPC ID: vpc-04d43f45127455847 X Clear filters

	Name	Internet gateway ID	State	VPC ID
<input type="checkbox"/>	Prod_VPC	igw-0472027a54d42add6	Attached	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Bastion_IGW	igw-0b97a9f716e75c9cd	Attached	vpc-0aa2872e25e5b4fb3 Bastion_VPC

Subnets CIDRs for both VPCs are created as described and shown in screenshot below:

- 1 public subnet is created in Bastion VPC.
- 1 public and various Private subnets are created in ProdVPC as shown. 2 azs are used for high availability for app and nginx instances. Corresponding azs (public and private) are required for NLBs to balance traffic

Subnets (10) info

Filter subnets

VPC: vpc-04d43f45127455847 X VPC: vpc-0aa2872e25e5b4fb3 X Clear filters

	Name	Subnet ID	State	VPC	IPv4 CIDR	Avail...	Route table	Default subnet	Auto-assign public IPv4 ad.
<input type="checkbox"/>	Private_subnet1_Nginx	subnet-05427a5a010cfd6	Available	vpc-04d43f45127455847 Prod_VPC	172.32.2.0/24	us-east-1a	rtb-02a7c84ccd211b34 Private_RT_withNATroute	No	No
<input type="checkbox"/>	Private_subnet2_Nginx	subnet-0f3d771179bdc241f	Available	vpc-04d43f45127455847 Prod_VPC	172.32.3.0/24	us-east-1b	rtb-02a7c84ccd211b34 Private_RT_withNATroute	No	No
<input type="checkbox"/>	Private_Subnet3_App	subnet-0f957f92f7537a87	Available	vpc-04d43f45127455847 Prod_VPC	172.32.4.0/24	us-east-1a	rtb-08c04539a314d0d19 Private_RT_InternalOnly	No	No
<input type="checkbox"/>	Private_Subnet4_App	subnet-0b65d28b1f1396ad8	Available	vpc-04d43f45127455847 Prod_VPC	172.32.5.0/24	us-east-1b	rtb-08c04539a314d0d19 Private_RT_InternalOnly	No	No
<input type="checkbox"/>	Private_subnet5_NLB	subnet-052e1a161e32c2449	Available	vpc-04d43f45127455847 Prod_VPC	172.32.6.0/24	us-east-1a	rtb-08c04539a314d0d19 Private_RT_InternalOnly	No	No
<input type="checkbox"/>	Private_subnet5_NLB	subnet-052e1a161e32c2449	Available	vpc-04d43f45127455847 Prod_VPC	172.32.9.0/24	us-east-1b	rtb-08c04539a314d0d19 Private_RT_InternalOnly	No	No
<input type="checkbox"/>	Private_subnet6_Maven	subnet-03d8b4c5b74622fc	Available	vpc-04d43f45127455847 Prod_VPC	172.32.7.0/24	us-east-1b	rtb-02a7c84ccd211b34 Private_RT_withNATroute	No	No
<input type="checkbox"/>	Public_subnet_bastion	subnet-0a48271a264981139	Available	vpc-0aa2872e25e5b4fb3 Bastion_VPC	192.168.1.0/24	us-east-1a	rtb-088012b2628075e4 public_RT_Bastion	No	Yes
<input type="checkbox"/>	Public_Subnet_NLB_NAT	subnet-0a1c3746f58375ddc	Available	vpc-04d43f45127455847 Prod_VPC	172.32.1.0/24	us-east-1a	rtb-0c4f01736324b7e3e Public_RT	No	No
<input type="checkbox"/>	Public_Subnet_NLB_NAT_2	subnet-0c8d4234d575a166e	Available	vpc-04d43f45127455847 Prod_VPC	172.32.8.0/24	us-east-1b	rtb-0c4f01736324b7e3e Public_RT	No	No

Route tables should be created as follows.

- 1 public route table in bastion VPC with IGW.

- 1 public route table in Prod_VPC for Public NLB
- At least 1 private route table in Prod_VPC for internal app, nginx, RDS and NLB servers. (My solution varied with 2 private route tables)

Route tables (4) [Info](#)

Filter route tables

VPC: vpc-04d43f45127455847 VPC: vpc-0aa2872e25e5b4fb3 Main: No Clear filters

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input type="checkbox"/>	Private_RT_InternalOnly	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNATroute	rtb-02a7c84ccd211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a26498...	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

The Private route table in Prod_VPC should include a local route, a route to transit gw and route to natgw in public subnet. For instance natgw is required by Tomcat application server to access artifacat from Jfrog repository.

Route tables (1/4) [Info](#)

Filter route tables

VPC: vpc-04d43f45127455847 VPC: vpc-0aa2872e25e5b4fb3 Main: No Clear filters

<input checked="" type="checkbox"/>	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input checked="" type="checkbox"/>	Private_RT_InternalOnly	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNATroute	rtb-02a7c84ccd211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a26498...	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-08c04539a314d0d19 / Private_RT_InternalOnly

Details **Routes** Subnet associations Edge associations Route propagation Tags

Routes (3)

Filter routes Both

Destination	Target	Status	Propagated
172.32.0.0/16	local	Active	No
192.168.0.0/16	tgw-0b5ea4125a698786c	Active	No
0.0.0.0/0	nat-0f488441dea93990e	Active	No

Public Route table in Prod_Vpc should include routes to IGW, transit gw and local routes. This will be used by the public facing NLB

Route tables (1/4) Info

Filter route tables

VPC: vpc-04d43f45127455847 VPC: vpc-0aa2872e25e5b4fb3 Main: No Clear filters

Actions

Create route table

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input type="checkbox"/>	Private_RT_InternalOnly	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNATroute	rtb-02a7c84ccdc211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input checked="" type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a26498...	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-0c4f01736324b7e3e / Public_RT

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (3)

Filter routes Both

Destination

Target

Status

Propagated

172.32.0.0/16	local	Active	No
192.168.0.0/16	tgw-0b5ea4125a698786c	Active	No
0.0.0.0/0	igw-0472027a54d42add6	Active	No

Bastion VPC needs only one route table which includes one public route to IGW, transit gateway and local route.

Route tables (1/4) Info

Filter route tables

Main: No VPC: vpc-0aa2872e25e5b4fb3 VPC: vpc-04d43f45127455847 Clear filters

Actions

Create route table

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input type="checkbox"/>	Private_RT_Internal...	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNA...	rtb-02a7c84ccdc211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input checked="" type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a26498...	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-088012b262f8075e4 / public_RT_Bastion

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (3)

Filter routes Both

Destination

Target

Status

Propagated

172.32.0.0/16	tgw-0b5ea4125a698786c	Active	No
192.168.0.0/16	local	Active	No
0.0.0.0/0	igw-0b97a9f716e75c9cd	Active	No

Associate routes appropriately to previously created subnets for both VPCs.

Public subnet in bastion_VPC is associated to Its public route table as shown below.

Route tables (1/4) Info

Filter route tables

Main: No VPC: vpc-0aa2872e25e5b4fb3 VPC: vpc-04d43f45127455847 Clear filters

Actions

Create route table

< 1 >

	Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
<input type="checkbox"/>	Private_RT_Internal...	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNA...	rtb-02a7c84ccdc211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input checked="" type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a264981139 / Public_Subnet_bastion	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-088012b262f8075e4 / public_RT_Bastion

Details Routes Subnet associations Edge associations Route propagation Tags

Explicit subnet associations (1)

Find subnet association

< 1 >

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0a48271a264981139 / Public_Subnet_bastion	192.168.1.0/24	-

Associate the public subnets to the public route table in prod_VPC

Route tables (1/4) Info

Filter route tables

VPC: vpc-04d43f45127455847 VPC: vpc-0aa2872e25e5b4fb3 Main: No Clear filters

Actions

Create route table

< 1 >

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input type="checkbox"/>	Private_RT_InternalOnly	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNATroute	rtb-02a7c84ccdc211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input checked="" type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a26498...	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-0c4f01736324b7e3e / Public_RT

Details Routes Subnet associations Edge associations Route propagation Tags

Explicit subnet associations (2)

Find subnet association

< 1 >

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0cbd4234d575a166e / Public_Subnet_NLB_NAT_2	172.32.8.0/24	-
subnet-0a1c3746f58375ddc / Public_Subnet_NLB_NAT	172.32.1.0/24	-

Associate all other private subnets to your private route table(s) in prodVPC.

Route tables (1/4) Info

VPC: vpc-04d43f45127455847

VPC: vpc-0aa2872e25e5b4fb3

Main: No

Clear filters

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input checked="" type="checkbox"/>	Private_RT_InternalOnly	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Private_RT_withNATroute	rtb-02a7c84ccd211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a26498...	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-08c04539a314d0d19 / Private_RT_InternalOnly

Details

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (4)

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0b65d28b1f1396ad8 / Private_Subnet4_App	172.32.5.0/24	-
subnet-0fc957c92f7537a87 / Private_Subnet3_App	172.32.4.0/24	-
subnet-052e1a161e32c2449 / Private_subnet5b_NLB	172.32.9.0/24	-
subnet-0af1785f179fd6378 / Private_subnet5_NLB	172.32.6.0/24	-

Route tables (1/4) Info

Main: No

VPC: vpc-0aa2872e25e5b4fb3

VPC: vpc-04d43f45127455847

Clear filters

	Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
<input type="checkbox"/>	Private_RT_InternalOnly	rtb-08c04539a314d0d19	4 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input checked="" type="checkbox"/>	Private_RT_withNATroute	rtb-02a7c84ccd211b34	3 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	Public_RT	rtb-0c4f01736324b7e3e	2 subnets	-	No	vpc-04d43f45127455847 Prod_VPC
<input type="checkbox"/>	public_RT_Bastion	rtb-088012b262f8075e4	subnet-0a48271a264981139 / Public_Subnet_bastion	-	No	vpc-0aa2872e25e5b4fb3 Bastion_VPC

rtb-02a7c84ccd211b34 / Private_RT_withNATroute

Details

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (3)

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-03d8bf4c5b74622fc / Private_subnet6_Maven	172.32.7.0/24	-
subnet-05427a5a01c0cfd6 / Private_subnet1_Nginx	172.32.2.0/24	-
subnet-0f3d771179bbd241f / Private_subnet2_Nginx	172.32.3.0/24	-

Using a Transit GW (TGW) provides a hub like solution for connecting VPCs or on-premise network to VPC.

Create the TGW as shown below. Only the TGW name is required.

Transit gateways (1/1) [Info](#)

[Refresh](#) [Actions](#) [Create transit gateway](#)

<input checked="" type="checkbox"/>	Name	Transit gateway ID	State
<input checked="" type="checkbox"/>	Transit-GW-1	tgw-Ob5ea4125a698786c	Available

tgw-Ob5ea4125a698786c / Transit-GW-1

[Details](#) [Sharing](#) [Tags](#)

Details

Transit gateway ID tgw-Ob5ea4125a698786c	State Available	Amazon ASN 64512	DNS support Enable
Transit gateway ARN arn:aws:ec2:us-east-1:123456789012:gateway/tgw-Ob5ea4125a698786c	Default association route table Enable	Association route table ID tgw-rtb-Oddc1711f5520321f	Auto accept shared attachments Disable
Owner ID 123456789012	Default propagation route table Enable	Propagation route table ID tgw-rtb-Oddc1711f5520321f	VPN ECMP support Enable
Description transit gw for bastion and prod vpc	Transit gateway CIDR blocks -	Multicast support Disable	

Two TGW attachments are required to connect both VPCs. Configuration of only Bastion TGW attachment is shown below. Similar configuration should be created for Prod_VPC TGW attachment

Transit gateway attachments (1/2) [Info](#)

[Refresh](#) [Actions](#) [Create transit gateway attachment](#)

<input type="checkbox"/>	Name	Transit gateway attachment ID	Transit gateway ID	Resource type	Resource ID	State	Association route table ID
<input type="checkbox"/>	Prod-tgw-attachment	tgw-attach-01448895475f539d2	tgw-Ob5ea4125a698786c	VPC	vpc-04d43f45127455847	Available	tgw-rtb-Oddc1711f5520321f
<input checked="" type="checkbox"/>	Bastion-tgw-attachment	tgw-attach-0d77b6f9b953130d3	tgw-Ob5ea4125a698786c	VPC	vpc-0aa2872e25e5b4fb3	Available	tgw-rtb-Oddc1711f5520321f

tgw-attach-0d77b6f9b953130d3 / Bastion-tgw-attachment

[Details](#) [Tags](#)

Details

Transit gateway attachment ID tgw-attach-0d77b6f9b953130d3	State Available	Resource type VPC	Association state Associated
Transit gateway ID tgw-Ob5ea4125a698786c	Resource owner ID 814109103016	Resource ID vpc-0aa2872e25e5b4fb3	Association route table ID tgw-rtb-Oddc1711f5520321f
Transit gateway owner ID 814109103016	DNS support Enable	IPv6 support Disable	Subnet IDs subnet-0a48271a264981139

After which a Transit GW default route table is auto populated with associations to TGW attachments to route traffic to both VPCs. Note route tables of ProdVPC subnet will also need to be updated for 2way communication.

Transit gateway route tables (1/1) Info

Filter transit gateway route tables

Actions

Create transit gateway route table

<input checked="" type="checkbox"/>	Name	Transit gateway route table ID	Transit gateway ID	State	Default association route table	Default propagation route table
<input checked="" type="checkbox"/>	-	tgw-rtb-0ddc1711f5520321f	tgw-0b5ea4125a698786c	Available	Yes	Yes

tgw-rtb-0ddc1711f5520321f

Details Associations Propagations Prefix list references Routes Tags

Associations (2) Info

Filter associations

Delete association

Create association

<input type="checkbox"/>	Attachment ID	Resource type	Resource ID	State
<input type="checkbox"/>	tgw-attach-01448895475f539d2	VPC	vpc-04d43f45127455847	Associated
<input type="checkbox"/>	tgw-attach-0d77b6f9b953130d3	VPC	vpc-0aa2872e25e5b4fb3	Associated

Security groups (SG) for Bastion, Tomcat, nginx and mysql RDS instance are shown below. Note that a security group can't be associated with a NLB unlike an application load balancer.

RDS mysql SG only accepts traffic from Tomcat Application SG on port 3306

Security Groups (1/4) Info

Filter security groups

Security group name: MySQL-SG X

Security group name: App-SG X

Security group name: Bastion_SG X

Security group name: Ngunx-SG X

Clear filters

Actions

Export security groups to CSV

Create security group

<input checked="" type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Inbound rules count
<input type="checkbox"/>	Nginx-SG	sg-0df0f3b09bc5545c5	Ngunx-SG	vpc-04d43f45127455847	Allow 80	2 Permission entries
<input checked="" type="checkbox"/>	MySQL-SG	sg-02ada923aac4884e1	MySQL-SG	vpc-04d43f45127455847	allow app servers on 3306	1 Permission entry
<input type="checkbox"/>	Bastion-SG	sg-0d9c3a2e92e8b166c	Bastion_SG	vpc-0aa2872e25e5b4fb3	allow 22	1 Permission entry
<input type="checkbox"/>	App-SG	sg-07ebf1f1697acd938	App-SG	vpc-04d43f45127455847	Allow 8080 from internal NLB	2 Permission entries

sg-02ada923aac4884e1 - MySQL-SG

Details Inbound rules Outbound rules Tags

Inbound rules (1/1)

Filter security group rules

Manage tags

Edit inbound rules

<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	-	sg-0e013385cda100af4	-	MySQL/Aurora	TCP	3306	sg-07ebf1f1697acd938 / App-SG	-

SG rule for Bastion SG is restricted to a particular IP on port 22 as shown.

Security Groups (1/4) Info

Filter security groups

Security group name: MySQL-SG X

Security group name: App-SG X

Security group name: Bastion_SG X

Security group name: Ngunx-SG X

Clear filters

	Name	Security group ID	Security group name	VPC ID	Description	Inbound rules count
<input type="checkbox"/>	Nginx-SG	sg-0df0f3b09bc5545c5	Ngunx-SG	vpc-04d43f45127455847	Allow 80	2 Permission entries
<input type="checkbox"/>	MySQL-SG	sg-02ada923aac4884e1	MySQL-SG	vpc-04d43f45127455847	allow app servers on 3306	1 Permission entry
<input checked="" type="checkbox"/>	Bastion-SG	sg-0d9c3a2e92e8b166c	Bastion_SG	vpc-0aa2872e25e5b4fb3	allow 22	1 Permission entry
<input type="checkbox"/>	App-SG	sg-07ebf1f1697acd938	App-SG	vpc-04d43f45127455847	Allow 8080 from Internal NLB	2 Permission entries

sg-0d9c3a2e92e8b166c - Bastion_SG

Details

Inbound rules

Outbound rules

Tags

Inbound rules (1/1)

Filter security group rules

Manage tags

Edit inbound rules

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	-	sg-r0f3b12df0067d9912	IPv4	SSH	TCP	22	97.108.231.162/32	-

Nginx SG port 80 inbound rule can be further restricted to VPC CIDR instead of accepting from anywhere as shown below.

Security Groups (1/1) Info

Filter security groups

Security group name: Ngunx-SG

Clear filters

Name

Security group ID

Security group name

VPC ID

Description

Inbound rules count

☒

Nginx-SG

sg-0df0f3b09bc5545c5

Ngunx-SG

vpc-04d43f45127455847

Allow 80

2 Permission entries

sg-0df0f3b09bc5545c5 - Ngunx-SG

Details

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (2)

Filter security group rules

Manage tags

Edit inbound rules

Name

Security group rule...

IP version

Type

Protocol

Port range

Source

Description

☐

-

sgr-0db691e371a5a3f02

IPv4

SSH

TCP

22

0.0.0.0/0

-

☐

-

sgr-00c4d909c06caf4ec

IPv4

HTTP

TCP

80

0.0.0.0/0

Allow http from anywhere

Tomcat server SG port 8080 rule can also be restricted to local VPC CIDR

Security Groups (1/1) [info](#) [Refresh](#) [Actions](#) [Export security groups to CSV](#) [Create security group](#)

[Security group name: App-SG](#) [Clear filters](#)

<input checked="" type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Inbound rules count
<input checked="" type="checkbox"/>	App-SG	sg-07ebf1f1697acd938	App-SG	vpc-04d43f45127455847	Allow 8080 from Internal NLB	2 Permission entries

sg-07ebf1f1697acd938 - App-SG

[Details](#) [Inbound rules](#) [Outbound rules](#) [Tags](#)

[You can now check network connectivity with Reachability Analyzer](#) [Run Reachability Analyzer](#)

Inbound rules (2) [Refresh](#) [Manage tags](#) [Edit inbound rules](#)

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0ff36a05316dcce8e	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	8080 from private NLB
<input type="checkbox"/>	-	sgr-07d5c278358ccd6b1	IPv4	SSH	TCP	22	0.0.0.0/0	-

One S3 bucket was created for pulling nginx config and tomcat log rotation script (referenced in launch template userdata).

Amazon S3 > valaxysuccess

valaxysuccess [info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	[REDACTED]	war	December 11, 2021, 23:41:03 (UTC-05:00)	23.3 MB	Standard
<input type="checkbox"/>	nginx.conf	conf	December 18, 2021, 00:27:13 (UTC-05:00)	2.4 KB	Standard
<input type="checkbox"/>	tomcat_logrotation.sh	sh	December 18, 2021, 02:43:00 (UTC-05:00)	266.0 B	Standard
<input type="checkbox"/>	tomcatlogs/	Folder	-	-	-

Tomcat logs also will be pushed to S3 via a cornjob script. These logs below were pushed to S3

Amazon S3 > valaxysuccess > tomcatlogs/ [Copy S3 URI](#)

tomcatlogs/ [Properties](#)

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	catalina.out.ip-172-32-5-108.ec2.internal.2021.12.18-07.34.43	43	December 18, 2021, 02:34:44 (UTC-05:00)	40.0 B	Standard
<input type="checkbox"/>	catalina.out.ip-172-32-5-247.ec2.internal.2021.12.18-08.42.28	28	December 18, 2021, 03:42:30 (UTC-05:00)	24.5 KB	Standard

Maven (Build)

1. Create EC2 instance using Maven Golden AMI

Maven instance is launched in prodVPC, private subnet and with custom Maven AMI

The screenshot displays the AWS Management Console interface for an EC2 instance named 'Maven'. The instance is in a 'Running' state, as indicated by the green checkmark and 'Running' text. The instance type is 't2.micro', and the status check shows '2/2 checks passed'. The instance is located in the 'us-east-1b' availability zone. The details panel is open, showing various configuration options such as 'Instance type', 'Platform', 'Launch time', and 'AMI ID'. The AMI ID is 'ami-0f77a0d9a27897c13', and the AMI name is 'Maven_GoldenAMI'. The instance is associated with the 'prodVPC' and 'Private_subnet6_Maven' subnet.

2. Clone Bitbucket repository to VSCode and update the pom.xml with Sonar and JFROG deployment details.

I forked instructor's repo and cloned from my bitbucket repo

```
# git clone remote_url && cd java-login-app
# git branch feature
# git checkout feature
```

For sonarcloud integration

- create an organization and a project in sonar cloud account.
- After which, instructions are provided for integration. Execute them on maven ec2 instance.
- Amongst other instructions this includes updating the pom.xml with organization name and sonar host url as shown below

```
<properties>
  <java.version>1.8</java.version>
  <sonar.organization>valaxy</sonar.organization>
  <sonar.host.url>https://sonarcloud.io</sonar.host.url>
</properties>
```

For jfrog integration:

- First create a repository on jfrog.
- Afterwards use the 'Quick Setup' option to generate deployment configuration.
- Click 'set me up' for your 'local' type repo. In this case, local repo is named 'assignment-libs-release-local'.
- click "deploy" tab on jfrog Web UI. This generates configuration to use at maven to upload generated artifact to jfrog local repository.
- Afterwards update the pom.xml file with generated distributionManagement config.

```
<distributionManagement>
  <repository>
    <id>central</id>
    <name>a0ajgojf2dwue-artifactory-primary-0-releases</name>
    <url>https://yemisprojects.jfrog.io/artifactory/assignment5-libs-release-local</url>
  </repository>
</distributionManagement>
```

3. Add settings.xml file to the root folder of the repository with the JFROG credentials and JFROG repo to resolve the dependencies.

- To generate settings.xml, use the 'Quick Setup' option in jfrog
 - Select 'default-maven-virtual' repo for downloading dependencies
 - Click 'configure' using 'default-maven-virtual' repo
 - A settings configuration for maven to connect to jfrog and download dependencies is auto-generated
 - Place configuration in /root/.m2/settings.xml file on maven instance
- Settings.xml file should include credentials and reference to default-maven-virtual jfrog repo.

Sample setting.xml file is shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.2.0 http://maven.apache.org/xsd/s
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <servers>
    <server>
      <username>loginusername@yahoo.com</username>
      <password>loginpasswd</password>
      <id>central</id>
    </server>
    <server>
      <username>loginusername@yahoo.com</username>
      <password>loginpasswd</password>
      <id>snapshots</id>
    </server>
  </servers>
  <profiles>
    <profile>
      <repositories>
        <repository>
          <snapshots>
            <enabled>>false</enabled>
          </snapshots>
          <id>central</id>
          <name>default-maven-virtual</name>
          <url>https://yemisprojects.jfrog.io/artifactory/default-maven-virtual</url>
        </repository>
        <repository>
          <snapshots />
          <id>snapshots</id>
          <name>default-maven-virtual</name>
          <url>https://yemisprojects.jfrog.io/artifactory/default-maven-virtual</url>
        </repository>
      </repositories>
    </profile>
  </profiles>
</settings>
```

4. Update application.properties file with JDBC connection string to authenticate with MySQL.

This 'src/main/resources/application.properties' git file should be updated with RDS endpoint name and connection credentials

```
.....
spring.datasource.url = jdbc:mysql://valaxy-db-1.cqsqkkkpnzxm.us-east-
1.rds.amazonaws.com:3306/UserDB

with corresponding credential also updated
spring.datasource.username = admin
.....
```

5. Push the code changes to feature branch of Bitbucket repository

```
# git add . && git commit -m "All changes with pom and properties file"
# git push origin feature
```


6. Raise Pull Request to approve the PR and Merge the changes to Master branch.

Link below has information on how to raise and approve pull requests from previous assignment

<https://github.com/yemisi/Valaxytraining.git>

7. Login to EC2 instance and clone the [Bitbucket repository](#)

```
# git clone remote_repo_url && cd java-login-app
```

8. Build the source code using maven arguments “-s settings.xml”

```
# mvn -s ~/.m2/settings.xml deploy
```

9. Integrate Maven build with Sonar Cloud and generate analysis dashboard with default Quality Gate profile.

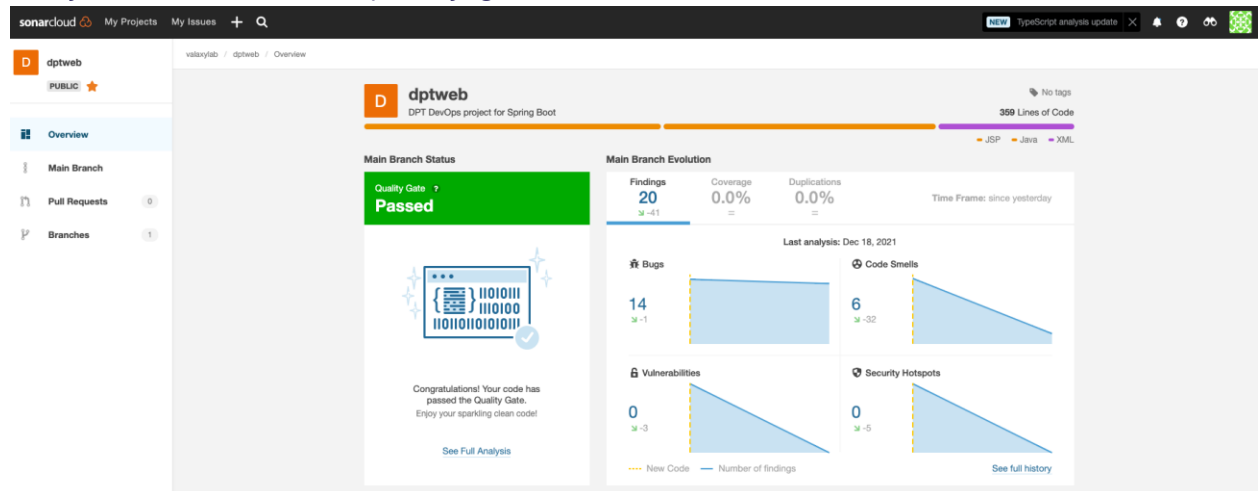
As stated in step 2, execute remaining instructions on maven instance.

Export environment variable and run mvn verify command

```
# export SONAR_TOKEN=xxxxxxxxxx
```

```
# mvn verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -  
Dsonar.projectKey=assignment5
```

Analysis dashboard with quality gate.



3-Tier Architecture

Database (RDS)

1. Deploy Multi-AZ MySQL RDS instance into private subnets

This was implemented as a standalone with Free tier for cost savings. Note VPC, security group and subnets selected. Endpoint name was auto generated

RDS > Databases > valaxy-db-1

valaxy-db-1 Modify Actions

Summary

DB identifier valaxy-db-1	CPU 4.83%	Status Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-1b

[Connectivity & security](#)
[Monitoring](#)
[Logs & events](#)
[Configuration](#)
[Maintenance & backups](#)
[Tags](#)

Connectivity & security

Endpoint & port

Networking

Security

Endpoint & port

Endpoint
valaxy-db-1.cqgkqkprxm.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1b

VPC
Prod_VPC (vpc-04d43f45127455847)

Subnet group
default-vpc-04d43f45127455847

Subnets
subnet-0fc957c92f7537a87
subnet-0b65d28b1f1396a88

Security

VPC security groups
MySQL-SG (sg-02ada923aac4884e1)
Active

Publicly accessible
No

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 01:08 (UTC+1:08)

Security group rules (2)

Security group	Type	Rule
MySQL-SG (sg-02ada923aac4884e1)	EC2 Security Group - Inbound	sg-07ebf1f1697acd938
MySQL-SG (sg-02ada923aac4884e1)	CIDR/IP - Outbound	0.0.0.0/0

2. Create Security Group allowing port 3306 from App instances and from Bastion Host.

Security Groups (1/4)
info
Filter security groups
Actions
Export security groups to CSV
Create security group

Security group name: MySQL-SG
Security group name: App-SG
Security group name: Bastion_SG
Security group name: Ngunx-SG
Clear filters

	Name	Security group ID	Security group name	VPC ID	Description	Inbound rules count
<input type="checkbox"/>	Nginx-SG	sg-0df0f3b09bc5545c5	Ngunx-SG	vpc-04d43f45127455847	Allow 80	2 Permission entries
<input checked="" type="checkbox"/>	MySQL-SG	sg-02ada923aac4884e1	MySQL-SG	vpc-04d43f45127455847	allow app servers on 3306	1 Permission entry
<input type="checkbox"/>	Bastion-SG	sg-0d9c3a2e92e8b166c	Bastion_SG	vpc-0aa2872e25e5b4fb3	allow 22	1 Permission entry
<input type="checkbox"/>	App-SG	sg-07ebf1f1697acd938	App-SG	vpc-04d43f45127455847	Allow 8080 from internal NLB	2 Permission entries

sg-02ada923aac4884e1 - MySQL-SG

Details
Inbound rules
Outbound rules
Tags

Inbound rules (1/1)
Filter security group rules
Manage tags
Edit inbound rules

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	-	sg-r-0e013385cda100af4	-	MYSQL/Aurora	TCP	3306	sg-07ebf1f1697acd938 / App-SG	-

Tomcat (Backend)

1. Create private facing Network Load Balancer and Target Group.

The internal NLB listens on port 8080 and forwards to App Target group

Create Load Balancer Actions

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
Public-NLB	Public-NLB-b25eab718015d518.elb.us-east-1.amazonaws.com	Active	vpc-04d43f45127455847	us-east-1a, us-east-1b	network	December 16, 2021 at 3:12:...	
Internal-NLB	Internal-NLB-d503414b4f99c6a5.elb.us-east-1.amazonaws.com	Active	vpc-04d43f45127455847	us-east-1a, us-east-1b	network	December 16, 2021 at 3:24:...	

Load balancer: Internal-NLB

Description Listeners Monitoring Integrated services Tags

Listeners listen for connection requests using their protocol and port. You can add, remove, or update listeners and listener rules.

To view and edit listener attributes, select the listener and choose Edit.

Add listener Edit Delete

Listener ID	Security policy	SSL Certificate	ALPN policy	Default action
TCP : 8080 arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/TCP-8080/0c4fb8b9bd93f501/0c4fb8b9bd93f501	N/A	N/A	N/A	forwarding to APP-TG

App Target group is set to listen on port 8080, since tomcat listens on same port by default

EC2 > Target groups

Target groups (1/2) Info

Search or filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer
APP-TG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/APP-TG/0c4fb8b9bd93f501	8080	TCP	Instance	Internal-NLB
NginxTG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/NginxTG/3f0237a7e67f310d	80	TCP	Instance	Public-NLB

APP-TG

arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/APP-TG/0c4fb8b9bd93f501

Details Targets Monitoring Health checks Attributes Tags

Details

Target type Instance	Protocol : Port TCP: 8080	VPC vpc-04d43f45127455847	IP address type IPv4
Load balancer Internal-NLB			

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
1	1	0	0	0	0

2. Create Launch Configuration with below configuration.

1. Tomcat Golden AMI
2. User Data to deploy .war artifact from JFROG into webapps folder.
3. Security Group allowing Port 22 from Bastion Host and Port 8080 from private NLB.

When creating launch config/template, specify the Tomcat golden AMI, keypair, App-SG and user data which downloads jfrog artifact and specifies a cronjob script to rotate log files to S3. Launch Template must have IAM role for S3, SSM and for cloudwatch agent to push custom metrics.

EC2 > Launch templates

Launch templates (1/2) info Actions Create launch template

Filter by tags or properties or search by keyword

Launch template ID	Launch template name	Default version	Latest version	Create time
lt-06a4efb21d810cbee	Nginx_LT	1	1	2021-12-18T08:01:21.000Z
lt-03324309885dcb2de	Tomcat_LT	1	1	2021-12-18T08:11:33.000Z

Metadata version
-

User data

```
#!/bin/sh
cd /opt/apache-tomcat-8.5.73/webapps
wget -O valaxy.war --user=yemisiomonijo20@yahoo.com --password=Iloveadunade1! https://yemisprojects.jfrog.io/artifactory/assignment5-libs-release-local/com/devopsrealtime/dptweb/1.0/dptweb-1.0.war
chmod +x valaxy.war
aws s3 cp s3://valaxysucess/tomcat_logrotation.sh /root/tomcat_logrotation.sh
chmod 755 /root/tomcat_logrotation.sh
(crontab -l 2>/dev/null ; echo "0 0 * * * /root/tomcat_logrotation.sh") | crontab -
```

Base64-encoded user data has been decoded for readability.

To push metrics, access S3 and use session manager, the following AWS managed policies attached to a IAM role was used in both Launch templates for Nginx and Tomcat

Roles > EC2-S3fullaccess Delete role

Summary

Role ARN `arn:aws:iam::[redacted]:role/EC2-S3fullaccess`

Role description Allows EC2 instances to call S3 service on your behalf. [Edit](#)

Instance Profile ARNs `arn:aws:iam::[redacted]:instance-profile/EC2-S3fullaccess`

Path /

Creation time 2021-12-11 23:37 EST

Last activity 2021-12-18 16:34 EST (Today)

Maximum session duration 1 hour [Edit](#)

Permissions Trust relationships Tags (1) Access Advisor Revoke sessions

▼ Permissions policies (3 policies applied)

[Attach policies](#) [Add inline policy](#)

Policy name	Policy type
AmazonS3FullAccess	AWS managed policy ✕
CloudWatchAgentServerPolicy	AWS managed policy ✕
AmazonSSMFullAccess	AWS managed policy ✕

3. Create Auto Scaling Group

The ASG is configured as shown below.

The screenshot displays the AWS Management Console interface for configuring an Auto Scaling Group (ASG). The top section shows a list of ASGs with columns for Name, Launch template/configuration, Instances, Status, Desired capacity, M..., M..., and Availability Zones. The 'Tomcat_ASG' is selected, showing it is using the 'Tomcat_LT | Version Default' launch template with 1 instance, a desired capacity of 1, and is located in the 'us-east-1a, us-east-1b' availability zones.

The 'Launch template' section provides details for the 'Tomcat_LT' template (ID: lt-03324309885dcb2de). It includes the AMI ID (ami-08b3c7da250bc7716), Instance type (t2.micro), Security groups (sg-07ebf1f1697acd938), Key pair name (valaxy5.pem), and Create time (Sat Dec 18 2021 03:11:33 GMT-0500 (Eastern Standard Time)).

The 'Network' section shows the ASG is configured for the 'us-east-1a, us-east-1b' availability zones and is associated with the subnet 'subnet-0b65d28b1f1396ad8, subnet-0f937c92f7537a87'.

Nginx (Frontend)

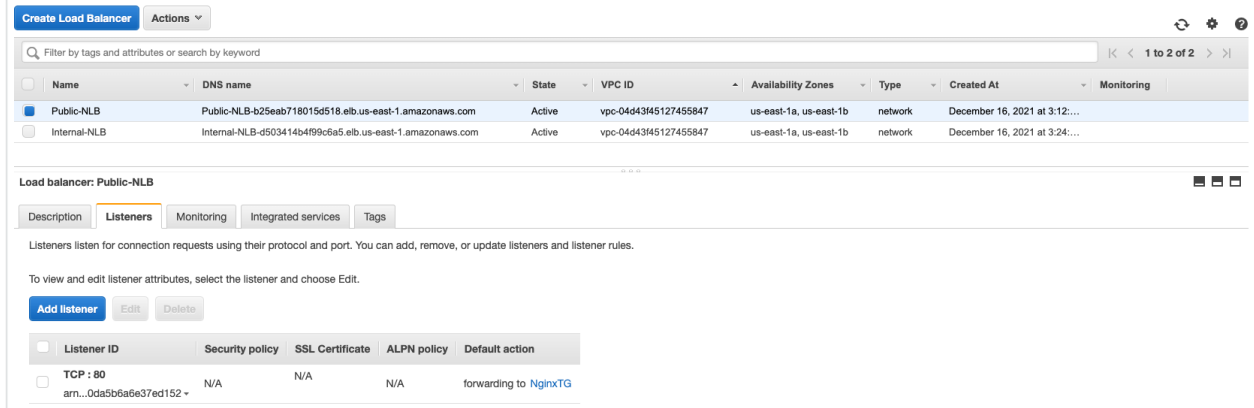
1. Create public facing Network Load Balancer and Target Group.

Nginx target group is set to port 80 since nginx listens on port 80

The screenshot displays the AWS Management Console interface for configuring a Target Group. The top section shows a list of Target Groups with columns for Name, ARN, Port, Protocol, Target type, and Load balancer. The 'NginxTG' is selected, showing it is using the 'am:aws:elasticloadbalancing:us-east-1:targetgroup/NginxTG/3f0237a7e67f310d' ARN, port 80, TCP protocol, and is associated with the 'Public-NLB' load balancer.

The 'Details' section provides information about the 'NginxTG' target group, including its Target type (Instance), Protocol (TCP: 80), VPC (vpc-04d43f45127455847), and IP address type (IPv4).

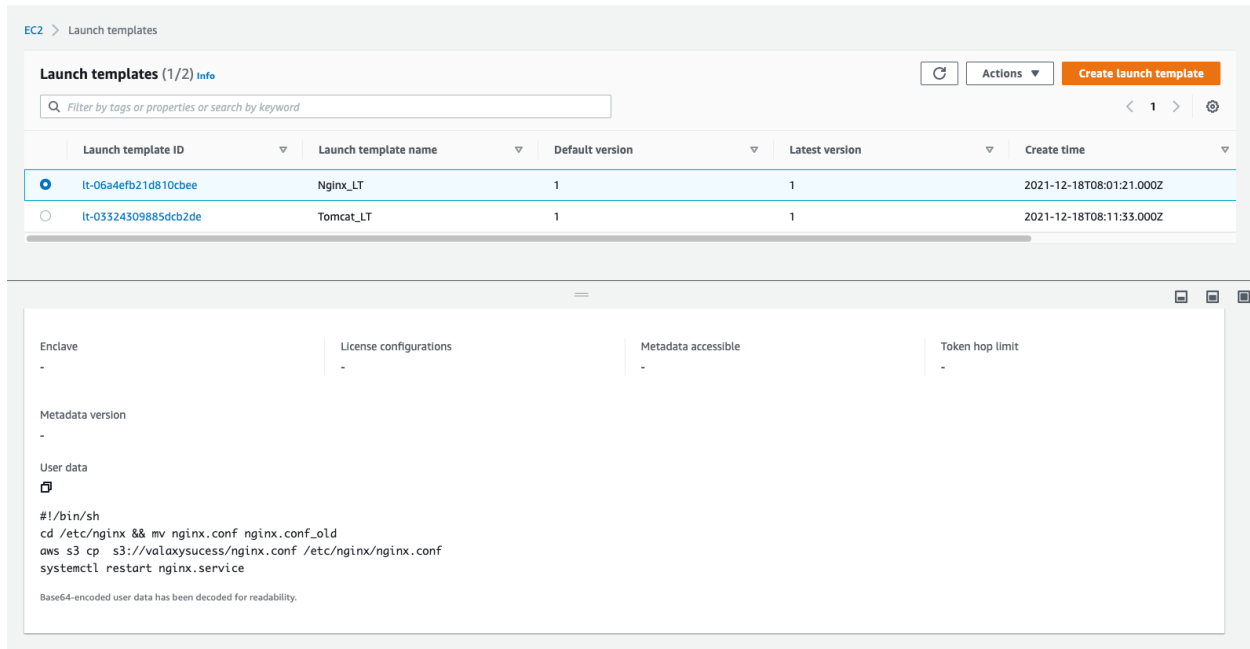
The 'Targets' section shows the status of the target group, including Total targets (1), Healthy (1), Unhealthy (0), Unused (0), Initial (0), and Draining (0).



2. Create Launch Configuration with below configuration

1. Nginx Golden AMI
2. User Data to update proxy_pass rules in nginx.conf file and reload nginx service.
3. Security Group allowing Port 22 from Bastion Host and Port 80 from Public NLB.

The userdata in the Nginx launch template below downloads a modified default nginx config file from S3. An alternative solution can edit the nginx config file in place using the sed command. I think the former is simpler.



Nginx.conf file is updated in the server block with the following directive

```
.....
location / {
    proxy_pass http://Internal-NLB-d503414b4f99c6a5.elb.us-east-1.amazonaws.com:8080/valaxy;
}
.....
```

3. Create Auto Scaling Group

EC2 > Auto Scaling groups

Auto Scaling groups (1/2)

Search your Auto Scaling groups

	Name	Launch template/configuration	Instances	Status	Desired capacity	M...	M...	Availability Zones
<input type="checkbox"/>	Tomcat_ASG	Tomcat_LT Version Default	1	-	1	1	1	us-east-1a, us-east-1b
<input checked="" type="checkbox"/>	Nginx_ASG	Nginx_LT Version Default	1	-	1	1	1	us-east-1a, us-east-1b

Launch template

Launch template: Nginx_LT | It-06a4efb21d810cbee

Version: Default

Description: Launch template for nginx

Request Spot Instances: No

View details in the launch template console

AMI ID: ami-0d5fb516a74b0e3a2

Security groups: -

Key pair name: valaxy5.pem

Create time: Sat Dec 18 2021 03:01:21 GMT-0500 (Eastern Standard Time)

Instance type: t2.micro

Security group IDs: sg-0d4f0f3b09bc5545c5

Storage (volumes):

Created by: [REDACTED]

Network

Availability Zones: us-east-1a, us-east-1b

Subnet ID: subnet-05427a5a01c0cfd6f, subnet-0f3d771179bbd241f

Application Deployment

1. Artifact deployment taken care by User Data script during Application tier EC2 instance launch process.
2. Login to MySQL database from Application Server using MySQL CLI client and create database and table schema to store the user login data (Instructions are update in README.md file in the Bitbucket repo)

Login to tomcat server, install mysql client and configure DB schema

```
# yum install mysql -y
```

```
# mysql -u admin -p -h valaxy-db-1.cqsqkkkpnzxm.us-east-1.rds.amazonaws.com
```

Enter password:

```
MySQL [(none)]> CREATE DATABASE UserDB;
```

```
MySQL [(none)]> use UserDB;
```

Database changed

```
MySQL [UserDB]> CREATE TABLE Employee ( id int unsigned auto_increment not null, first_name varchar(250), last_name varchar(250), email varchar(250), username varchar(250), password varchar(250), regdate timestamp, primary key (id) );
```

```
Query OK, 0 rows affected (0.09 sec)
```

Post-Deployment

1. Configure Cronjob to push the Tomcat Application log data to S3 bucket and also rotate the log data to remove the log data on the server after the data pushed to S3 Bucket.

Tomcat Launch template was set with daily cronjob to execute following script to rotate log data. An alternative could be to define a tomcat config In `‘/etc/logrotate.d/tomcat’` and use *logrotate* to rotate log files.

```
#!/bin/sh
cd /opt/apache-tomcat-8.5.73/logs/
file_name="catalina.out"
current_time=$(date "+%Y.%m.%d-%H.%M.%S")
servername=$(hostname)
new_filename=$file_name.$servername.$current_time

aws s3 cp catalina.out s3://valaxysucess/tomcatlogs/$new_filename
>catalina.out
```

Log files are renamed and pushed to S3 as shown below

Amazon S3 > valaxysucess > tomcatlogs/

tomcatlogs/ Copy S3 URI

Objects Properties

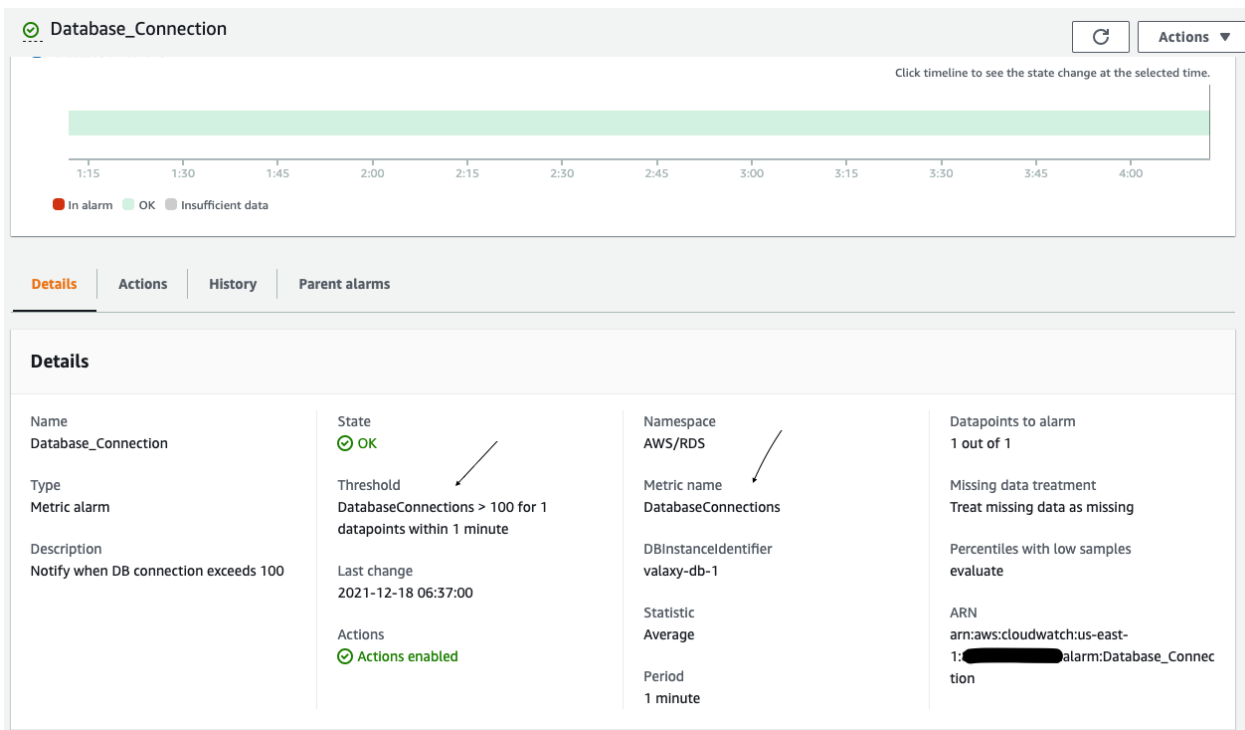
Objects (2)
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	catalina.out-172-32-5-108.ec2.internal.2021.12.18-07.34.43	43	December 18, 2021, 02:34:44 (UTC-05:00)	40.0 B	Standard
<input type="checkbox"/>	catalina.out-172-32-5-247.ec2.internal.2021.12.18-08.42.28	28	December 18, 2021, 03:42:30 (UTC-05:00)	24.5 KB	Standard

2. Configure Cloudwatch alarms to send E-Mail notification when database connections are more than 100 threshold.



Validation

1. Verify you as an administrator able to login to EC2 instances from session manager & from Bastion Host.
ssh forwarding was used to access Nginx and App instances from bastion host
start ssh agent and add the key you want to forward to the agent
\$ eval "\$(ssh-agent)"
Agent pid 83448
\$ ssh-add valaxy5.pem && ssh -A ec2-user@100.24.42.154

Confirmed access to Tomcat instance from bastion

```
[ec2-user@ip-192-168-1-90 ~]$ ssh ec2-user@172.32.5.247
Last login: Sat Dec 18 08:41:45 2021 from ip-192-168-1-90.ec2.internal

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

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-32-5-247 ~]$
```

Confirmed access to Nginx access from bastion

```
[ec2-user@ip-192-168-1-90 ~]$ ssh ec2-user@172.32.3.136
Last login: Sat Dec 18 08:43:44 2021 from ip-192-168-1-90.ec2.internal

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

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-32-3-136 ~]$
```

Session Manager access for Nginx aided with IAM role defined in launch template

```
Session ID: admin-Qa750853d22200f00 Instance ID: i-03e7f4b5d8b84225f Terminate
sh-4.2$ sudo su ec2-user
[ec2-user@ip-172-32-3-136 bin]$ systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-12-18 08:17:24 UTC; 20h ago
     Process: 3276 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Process: 3273 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 3271 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Main PID: 3279 (nginx)
   CGroup: /system.slice/nginx.service
           └─3279 nginx: master process /usr/sbin/nginx
             └─3280 nginx: worker process
```

Session Manager access for Tomcat aided with IAM role defined in launch template

```
Session ID: admin-0c6b518b40e0b67d8 Instance ID: i-059fef7b870eabea3 Terminate
sh-4.2$ sudo su ec2-user
[ec2-user@ip-172-32-5-247 bin]$ systemctl status tomcat.service
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-12-18 08:19:37 UTC; 20h ago
     Main PID: 2892 (java)
   CGroup: /system.slice/tomcat.service
           └─2892 /usr/bin/java -Djava.util.logging.config.file=/opt/apache-tomcat-8.5.73/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeyS...

Dec 18 08:19:36 ip-172-31-4-133.ca-central-1.compute.internal systemd[1]: Starting Apache Tomcat Web Application Container...
Dec 18 08:19:37 ip-172-31-4-133.ca-central-1.compute.internal startup.sh[2869]: Tomcat started.
Dec 18 08:19:37 ip-172-31-4-133.ca-central-1.compute.internal systemd[1]: Started Apache Tomcat Web Application Container.
[ec2-user@ip-172-32-5-247 bin]$
```

2. Verify if you as an end user able to access application from public internet browser.

Custom domain was used to complete the setup. Here is the route53 alias record to the public NLB

Route 53 > Hosted zones > cloud2hit.com

cloud2hit.com

Delete zone Test record Configure query logging

Hosted zone details

Edit hosted zone

Records (3)

DNSSEC signing

Hosted zone tags (0)

Records (1/3)

Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Filter records by property or value

Type Routing policy Alias

< 1 >

	Record name	Type	Routing policy	Value/Route traffic to
<input type="checkbox"/>	cloud2hit.com	NS	Simple	ns-1052.awsdns-03.org. ns-1560.awsdns-03.co.uk. ns-589.awsdns-09.net. ns-474.awsdns-59.com.
<input type="checkbox"/>	cloud2hit.com	SOA	Simple	ns-1052.awsdns-03.org. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400
<input checked="" type="checkbox"/>	www.cloud2hit.com	A	Simple	public-nlb-b25eab718015d518.elb.us-east-1.amazonaws.com.

Record details

Edit record

Record name

www.cloud2hit.com

Record type

A

Value

public-nlb-b25eab718015d518.elb.us-east-1.amazonaws.com.

TTL (seconds)

-

Routing policy

Simple

Following user “Yemisi” successfully registered via the website

```
MySQL [UserDB]> SELECT * FROM Employee;
```

id	first_name	last_name	email	username	password	regdate
1	Oluwayemisi	Odunade	test@example.com	yemisi	yemisi	2021-12-18 00:00:00

1 row in set (0.01 sec)

User ‘yemisi’ with email ‘test@example.com’ was able to login to
‘www.cloud2hit.com’

← → 🔍 ⚙️ Not Secure | cloud2hit.com/login

Welcome test@example.com

[iwayQ.com](#) | Instant Information Site

[Home](#) | [Gallery](#) | [Videos](#) | [Articles](#) | [Tutorials](#) | [Assessment](#) | [Poll & Survey](#) | [Chat](#)