**VALAXY TRAINING ASSIGNMENT 3**

**Pre-Deployment**

Customize the application dependencies mentioned below on AWS EC2 instance and create the Golden AMI.

1. AWS CLI
2. Install Apache Web Server
3. Install Git
4. Cloudwatch Agent
5. Push custom memory metrics to Cloudwatch.
6. AWS SSM Agent

Number 1 & 6 are installed by default on AmazonLinux 2, so install the rest.

* Launch an instance in a default VPC and run the following commands

sudo su

yum install -y httpd git

yum install amazon-cloudwatch-agent -y

* Run this cloudwatch config wizard and select the defaults, but ensure to select the memory option when prompted and the cwagent user

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

* Start the cloudwatch agent

/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 cloudwatch agent is running

systemctl status amazon-cloudwatch-agent.service

* To Push custom memory metrics to Cloudwatch, attach an IAM role to the instance with this AWS managed policy named CloudWatchFullAccess
* If you need to test the session manager works also attach AmazonSSMFullAccess AWS managed policy to the existing IAM role

Once all dependencies are installed, create the AMI as shown below

Graphical user interface, text, application

Description automatically generated

**VPC Deployment**

1. Build VPC network ( 192.168.0.0/16 ) for Bastion Host deployment as per the architecture shown above.

Implemented my whole assignment in Canada (central) region.

Create Bastion VPC with this basic configuration

A screenshot of a computer

Description automatically generated

1. Build VPC network ( 172.32.0.0/16 ) for deploying Highly Available and Auto Scalable application servers as per the architecture shown above.

Create App VPC with this basic configuration

A screenshot of a computer

Description automatically generated

1. 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.

Create all 4 subnets shown for APP VPC. Note route table IDs, AZs and CIDRs

Graphical user interface, application, Word

Description automatically generated

Create Nat gateway. Note connectivity type, EIP and public subnet

A screenshot of a computer

Description automatically generated

See private and public route table rules.

Note 3 route with targets for local, natgw and transit gw in private route table

A screenshot of a computer

Description automatically generated

Private route table Subnet associations

A screenshot of a computer

Description automatically generated

Public route table in APP VPC with 0.0.0.0/0 route

Graphical user interface, application

Description automatically generated

Public route table Subnet associations

A screenshot of a computer

Description automatically generated

1. Create Transit Gateway and associate both VPCs to the Transit Gateway for private communication.

Click create transit gateway and add only a name

Graphical user interface, text, application, email

Description automatically generated

You need to create 2 transit gw attachment to each VPC. Add name, select existing transit gw ID, select Bastion & App VPC

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create Internet Gateway for each VPC and Public Subnet associated Route Table accordingly to route the default traffic to IGW for inbound/outbound internet connection.

Create 2 Internet GWz and attach to each VPC. Routes are shown with 0.0.0.0/0 in previous step snapshot

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

This transit gateway route table will be created by default with these associations to each VPC

A screenshot of a computer

Description automatically generated

Afterwards update the private route table in ApP vpc and public route table in Bastion VPC, with the routes shown in previous snapshot

1. Create Cloudwatch Log Group with two Log Streams to store the VPC Flow Logs of both VPCs.

To create 2 logroups, Click logroup in cloudwatch and click create log group. These 2 for Bastion and App VPCs were created with a retention 3 days. No logs streams will be seen until next step

Graphical user interface, application

Description automatically generated

1. Enable Flow Logs for both VPCs and push the Flow Logs to Cloudwatch Log Groups and store the logs in the respective Log Stream for each VPC.

To enable flow logs, click Actions and Create flow logs. Repeat for each VPC

A screenshot of a computer

Description automatically generated

This is bastion VPC flow log settings after creation. Note the destination Name from previous step

A screenshot of a computer

Description automatically generated

Similar config for APP vpc flow log

A screenshot of a computer

Description automatically generated

These log streams will auto stream once steps above are completed. APP vpc sample flow logs are Shown below

A screenshot of a computer

Description automatically generated

1. Create Security Group for bastion host allowing port 22 from public.

Open port 22 and icmp (optional for ping tests)

A screenshot of a computer

Description automatically generated

1. Deploy Bastion Host EC2 instance in the Public Subnet with EIP associated.

Create Bastion subnet and public route table. Note the CIDR and linked route table

A screenshot of a computer

Description automatically generated

Public route table of the bastion subnet is shown here. The red route is for a route via the transit gateway to the App VPC and the blue route is for a route for Internet access in and out of the bastion subnet. The tgw ID will only appear after setting up the transit gateway

A screenshot of a computer

Description automatically generated

1. Create S3 Bucket to store application specific configuration.

This bucket was created in Canada region with default settings

Graphical user interface, text, application

Description automatically generated

1. Create Launch Configuration with below configuration.
   1. Golden AMI
   2. Instance Type – t2.micro
   3. Userdata to pull the code from Bitbucket Repository  to document root folder of webserver and start the httpd service.
   4. IAM Role granting access to Session Manager and to S3 bucket created in the previous step to pull the configuration. (Do  not grant S3 Full Access)
   5. Security Group allowing port 22 from Bastion Host and Port 80 from Public.
   6. Key Pair

Specification shown here. Create a launch template

Referenced previously taken golden amiGraphical user interface, text, application, email

Description automatically generated

Launch template security group, Graphical user interface, text, application, email

Description automatically generated

Userdata clones repo and starts apache

Graphical user interface, text, application, email

Description automatically generated

This is the security group rules for launcg template for reference

A screenshot of a computer

Description automatically generated

1. Create Auto Scaling Group with Min: 2 Max: 4 with two Private Subnets associated to 1a and 1b zones.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generated

1. Create Target Group and associate it with ASG.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Create Network Load balancer in Public Subnet and add Target Group as target.

Graphical user interface, text, application, email

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Update route53 hosted zone with CNAME record routing the traffic to NLB.

Graphical user interface, text, application, email

Description automatically generated

#### Validation

1. As DevOps Engineer login to Private Instances via Bastion Host.

Text, letter

Description automatically generated

1. Login to AWS Session Manager and access the EC2 shell from console.

Text

Description automatically generated

1. Browse web application from public internet browser using domain name and verify that page loaded.

Graphical user interface, website

Description automatically generated