

AWS CLOUD SOLUTIONS PROJECT 3

STEP-BY-STEP PROCESS OF SETTING UP A 3-TIER ARCHITECTURE OF A WEB APP

A 3-tier architecture is a software architecture pattern where the application is broken down into three logical tiers:

1. the presentation layer,
2. the application (business logic) layer and
3. the data storage layer.

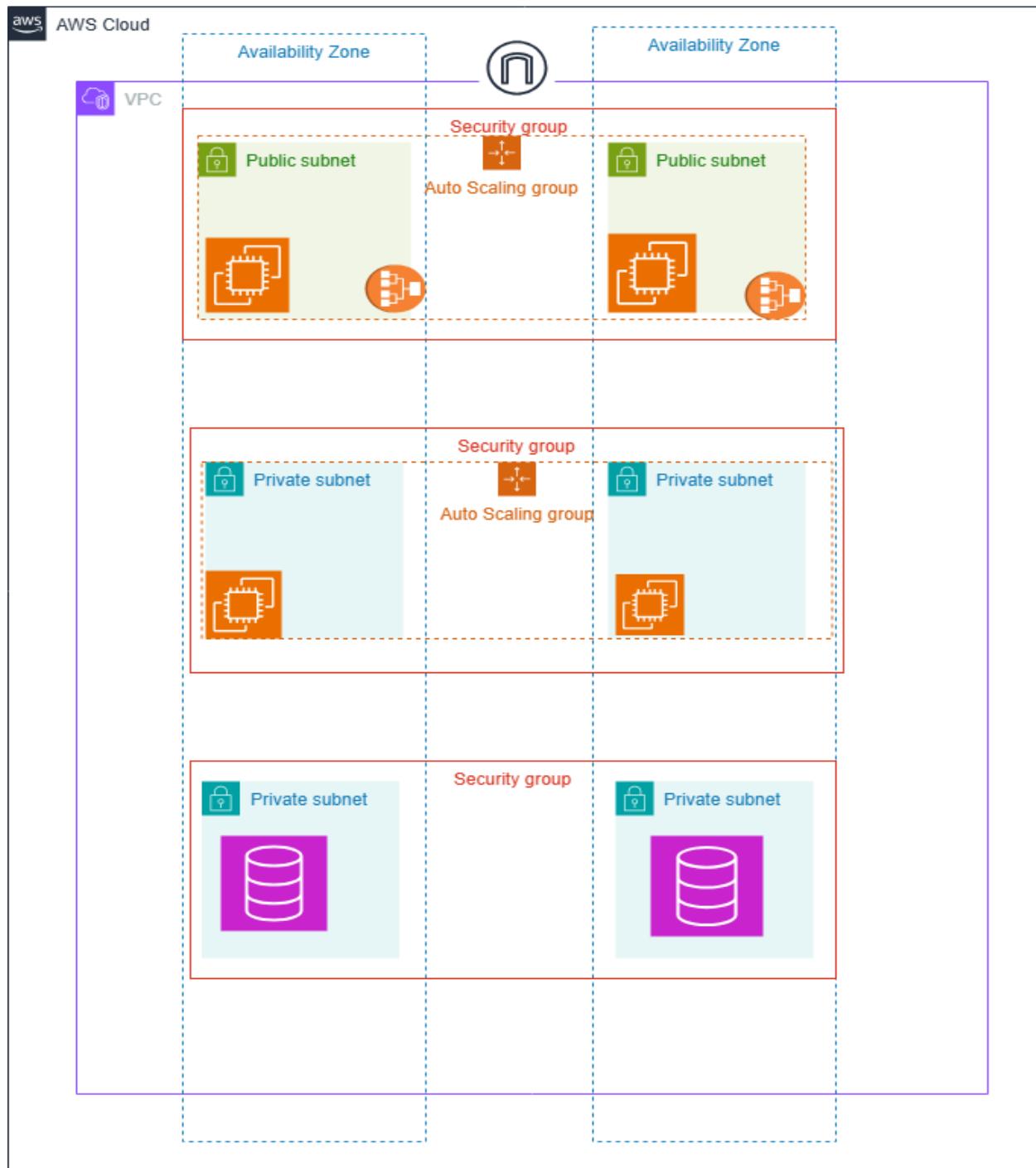
This architecture is used in a client-server application such as a web application that has the frontend, the backend and the database. Each of these layers or tiers does a specific task and can be managed independently of each other.

In this project, I will be using the following AWS services to design and build a three-tier cloud infrastructure:

Elastic Compute Cloud (EC2), Auto Scaling Group, Virtual Private Cloud (VPC), Elastic Load Balancer (ELB), Security Groups and the Internet Gateway, NAT Gateway, Health check.

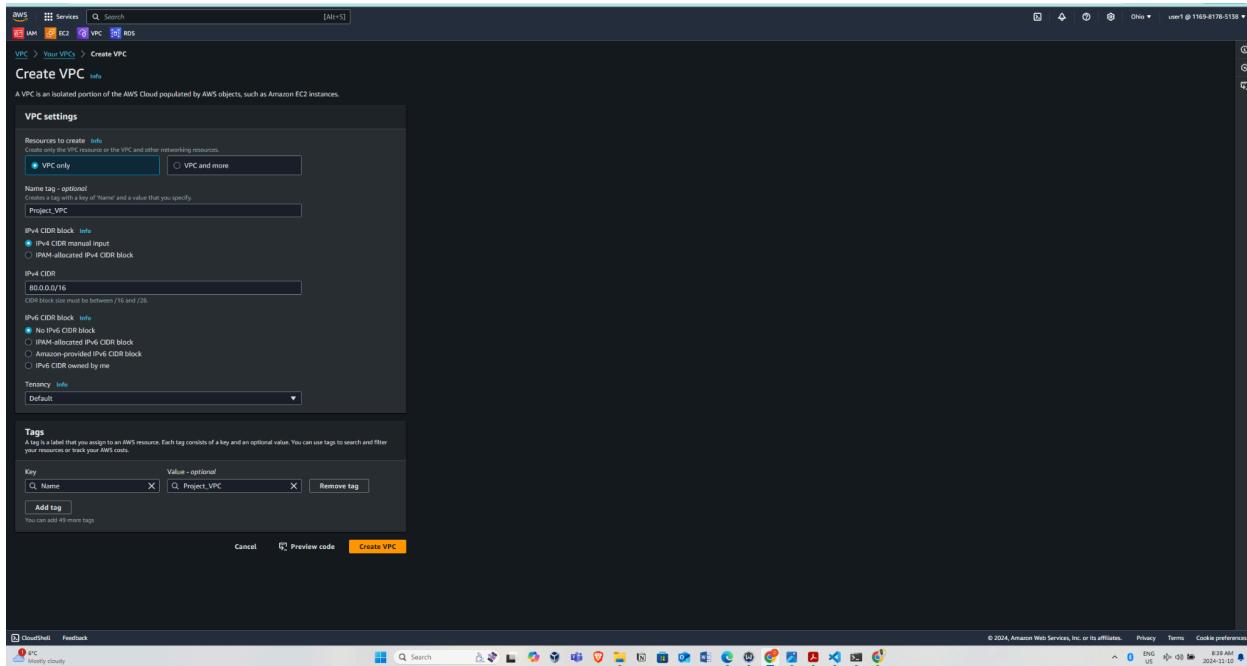
This infrastructure will be designed to be secured, highly available and fault tolerant.

Below is the architectural diagram of this project:



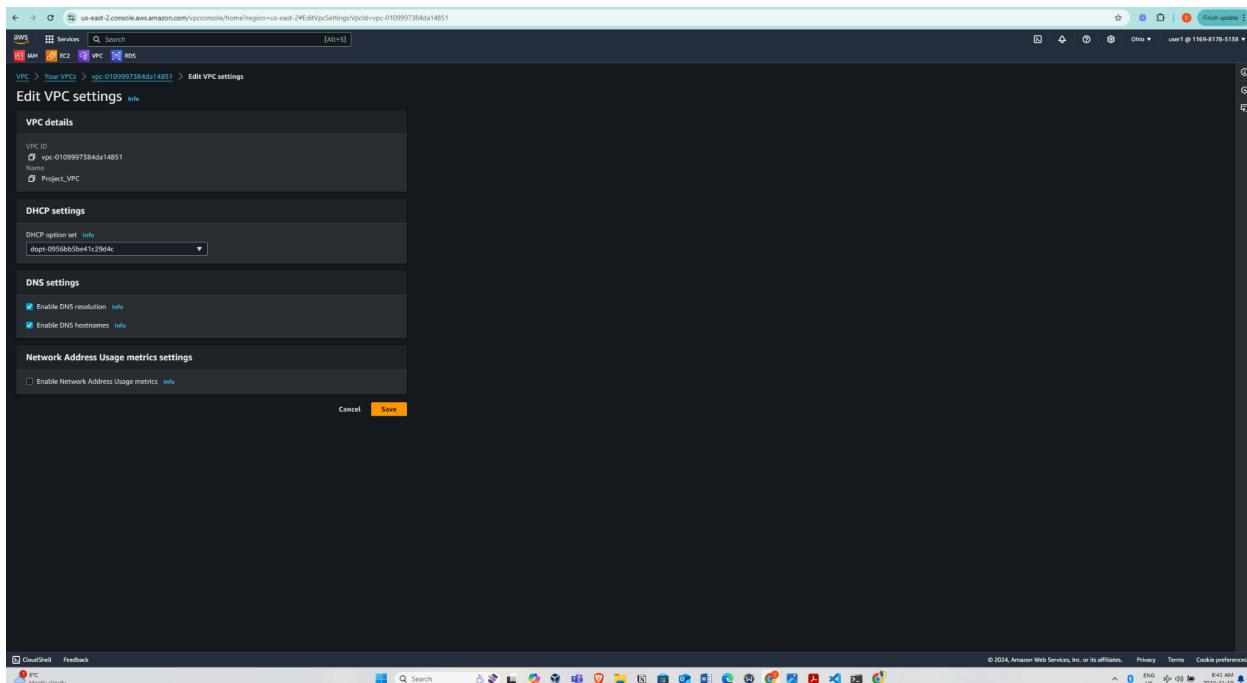
Step 1

Creating a VPC



Step 1b

Edit VPC settings and enable DNS host names. This would allow resources provisioned within this VPC that require a DNS host name to be allocated a DNS host name.



Step 3

Create 6 subnets in 2 availability zones. I will be creating 2 public subnets in each AZs, 2 private subnets in each AZs and 2 database subnets which would still be private in each AZs. This makes a total of 6 subnets.

The screenshot shows the AWS VPC console interface for creating subnets. It displays two main sections: 'Associated VPC CIDRs' and 'Subnet settings'.

Associated VPC CIDRs: Shows one entry: IPv4 CIDR: 80.0.0.0/16.

Subnet settings: Contains two sub-sections for Subnet 1 and Subnet 2, followed by a summary section for Subnet 6.

- Subnet 1 of 6:** Subnet name: project_pub_sza2, Availability Zone: US East (Ohio) / us-east-2a, IPv4 VPC CIDR block: 80.0.0.0/16, IPv4 subnet CIDR block: 80.0.0.0/24, Tags: project_pub_sza2.
- Subnet 2 of 6:** Subnet name: project_prv_sza2, Availability Zone: US East (Ohio) / us-east-2a, IPv4 VPC CIDR block: 80.0.0.0/16, IPv4 subnet CIDR block: 80.0.4.0/24, Tags: project_prv_sza2.
- Subnet 6 of 6:** Subnet name: project_dbns2b, Availability Zone: US East (Ohio) / us-east-2b, IPv4 VPC CIDR block: 80.0.0.0/16, IPv4 subnet CIDR block: 80.0.5.0/24, Tags: project_dbns2b.

At the bottom, there is a 'Create subnet' button.

The screenshot shows the AWS VPC dashboard with a list of subnets. The subnets are categorized into two main groups: private (project_priv_sn2b, project_pub_sn2b, project_pub_sn2a, project_dsn2b) and public (project_priv_sn2a, project_dsn2a). The table includes columns for Name, Subnet ID, State, VPC, IPv4 CIDR, IPv6 CIDR, IPv6 CIDR association ID, Available IPv4 addresses, and Availability Zone. The Availability Zone column is highlighted with a blue box.

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID	Available IPv4 addresses	Availability Zone
project_priv_sn2b	subnet-055f162c2edde	Available	vpc-0109997584d414851	80.0.3.0/24	-	-	251	us-east-2b
project_pub_sn2b	subnet-05d1e4d9f99a387	Available	vpc-0109997584d414851	80.0.2.0/24	-	-	251	us-east-2b
project_pub_sn2a	subnet-0672bc7f2c4266	Available	vpc-0109997584d414851	80.0.0.0/24	-	-	251	us-east-2b
project_dsn2b	subnet-06456b21626c2edde	Available	vpc-002459fbcc077e9a1	80.0.5.0/24	-	-	251	us-east-2b
project_priv_sn2a	subnet-0bb02d2443750170	Available	vpc-0109997584d414851	80.0.1.0/24	-	-	251	us-east-2a
project_dsn2a	subnet-002459fbcc077e9a1	Available	vpc-0109997584d414851	80.0.4.0/24	-	-	251	us-east-2a

Step 3b

Edit subnet settings of the public subnets and enable auto-assign IP addresses. This would dynamically allocate public IP addresses to resources provisioned within the public subnets.

The screenshot shows the 'Edit subnet settings' dialog box for the subnet project_pub_sn2b. The dialog is divided into several sections:

- Subnet:** Shows the subnet ID (subnet-05d1e4d9f99a387) and name (project_pub_sn2b).
- Auto-assign IP settings:** Includes a note about enabling auto-assign for public IPv4 or IPv6. The 'Enable auto-assign public IPv4 address' checkbox is checked.
- Resource-based name (RBN) settings:** Includes a note about specifying a hostname type for EC2 instances. It shows options for 'Enable resource name DNS A record on launch' and 'Enable resource name DNS AAAA record on launch'. The 'IP name' radio button is selected.
- DNS64 settings:** Includes a note about enabling DNS64 for IPv6-only services. The 'Enable DNS64' checkbox is checked.

At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

Step 4

Create an internet gateway and attach it to the VPC

The screenshot shows the AWS VPC Internet Gateways Details page. The URL is us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#InternetGateway/internetGatewayId=igw-0817b63fa2517e8c5. The page displays the following information:

- Internet gateway ID:** igw-0817b63fa2517e8c5
- Name:** Project_IGW
- State:** detached
- VPC ID:** -
- Owner:** user1 @ 1169-817b-5138

The left sidebar shows the VPC dashboard with various options like EC2 Global View, Virtual private cloud, Subnets, Route tables, Internet gateways, Security, DNS firewall, Network Firewall, and Virtual private network (VPN).

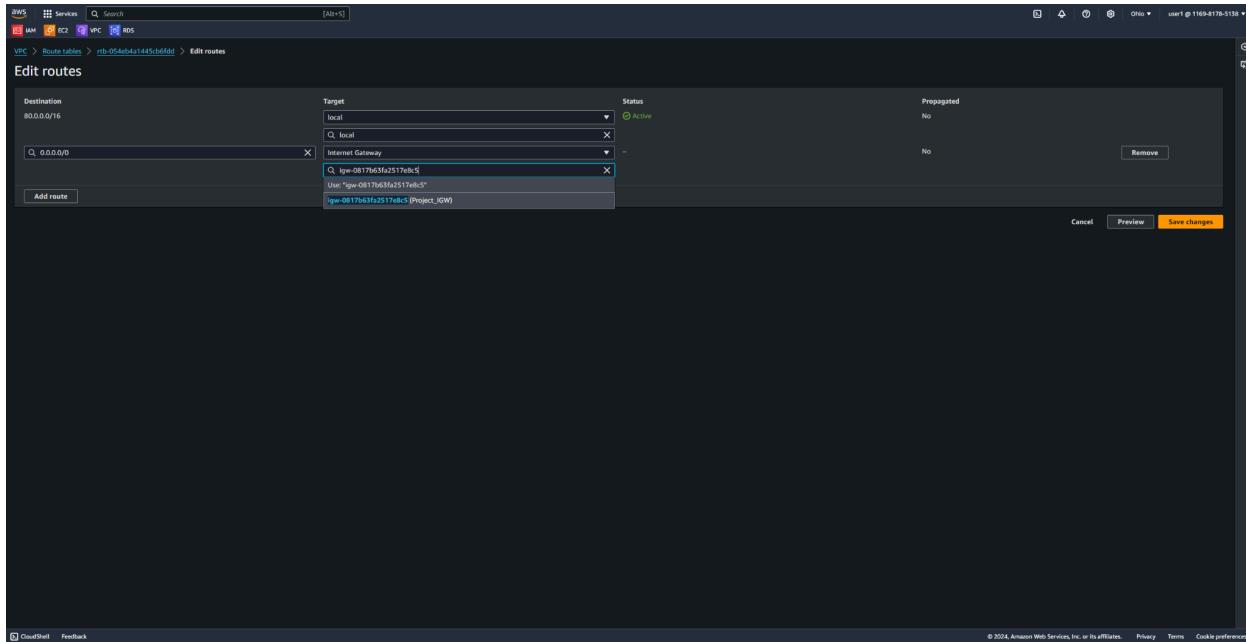
The screenshot shows the "Attach to VPC" dialog box. The URL is us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#AttachInternetGateway/internetGatewayId=igw-0817b63fa2517e8c5. The dialog box lists available VPCs:

- vpc-0109997384da14851
- vpc-0109997384da14851 - Project_VPC

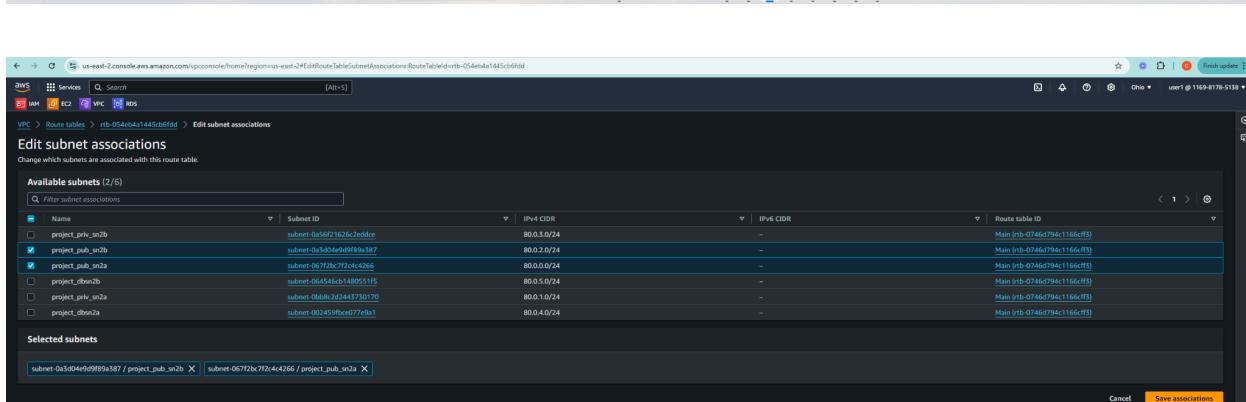
The "Use" dropdown is set to "vpc-0109997384da14851". At the bottom are "Cancel" and "Attach internet gateway" buttons.

Step 5

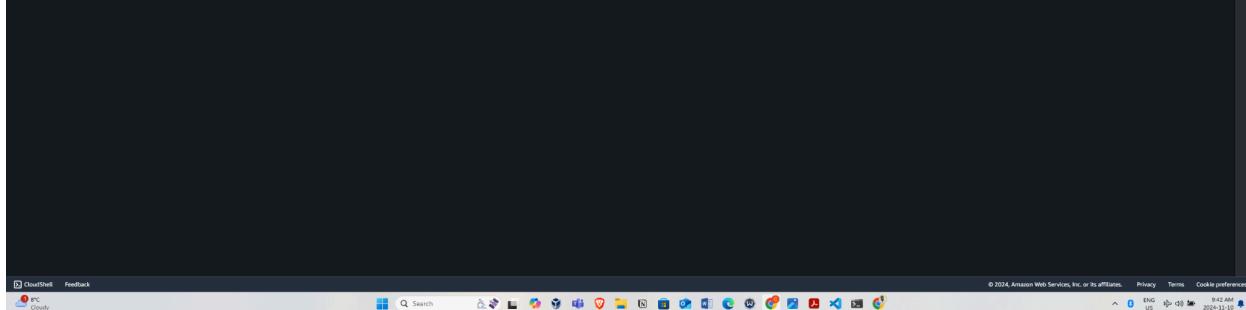
Create 2 route tables and associate them with their respective subnets. A public route table which will have a route pointing to the IGW and a private route table with a route pointing to Nat_GW.



The screenshot shows the 'Edit routes' page for a specific route table. A route is being added for destination 0.0.0.0/0, targeting the local endpoint ('local') with status 'Active'. The 'Propagated' field is set to 'No'. Below the table, there are buttons for 'Cancel', 'Preview', and 'Save changes'.



The screenshot shows the 'Edit subnet associations' page. Under 'Available subnets', several subnets are listed, including project_prv_sn2b, project_pub_sn2b, project_pub_sn2a, project_dbsn2b, project_prv_sn2a, and project_dbsn2a. Under 'Selected subnets', two subnets are selected: subnet-0x3d4e9d9f89a58f7 / project_pub_sn2b and subnet-067f2bc7f2c4c4766 / project_pub_sn2a. The 'Save associations' button is highlighted.



This screenshot is identical to the one above, showing the 'Edit subnet associations' page with the same selected subnets and the 'Save associations' button highlighted.

Private subnet association

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#editRouteTable/SubnetAssociations?RouteTableId=rtb-045d1f903a2150a2ae>. The page title is "Edit subnet associations".

Available subnets (2/6)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
project_priv_sn2b	subnet-0x56f21626c2edddc	80.0.3.0/24	-	Main (rtb-0746f794c1166cf5)
project_pub_sn2b	subnet-0x5d4e9ff9fb9a3b7	80.0.2.0/24	-	rtb-0546b41145cb6fd / Project_pubRT
project_pub_sn2a	subnet-06712bc7f7dc42766	80.0.0.0/24	-	rtb-0546b41145cb6fd / Project_pubRT
project_dbsn2b	subnet-064546c1480551f5	80.0.5.0/24	-	Main (rtb-0746f794c1166cf5)
project_priv_sn2a	subnet-0bb6b26244370170	80.0.1.0/24	-	Main (rtb-0746f794c1166cf5)
project_dbsn2a	subnet-002459bf077fe9a1	80.0.4.0/24	-	Main (rtb-0746f794c1166cf5)

Selected subnets

- subnet-0x56f21626c2edddc / project_priv_sn2b
- subnet-0bb6b26244370170 / project_priv_sn2a

Buttons: Cancel, Save associations.

Step 6

Create a Nat_GW in one of the public Subnets. Then add a new route to the private route table pointing to the Nat_GW. This will enable resources in the private subnet which needs to go to the internet for legitimate reasons to have a route to access the internet.

The screenshot shows the AWS VPC console with the URL <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#CreateNatGateway>. The page title is "Create NAT gateway".

Create NAT gateway

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name: project_Nat_GW

Subnet: Subnet-06712bc7f7dc42766 (project_pub_sn2a)

Connectivity type: Public

Elastic IP allocation ID: eipalloc-04497ecc18ae45f5

Tags: Key: Name, Value: project_Nat_GW

Buttons: Cancel, Create NAT gateway.

The screenshot shows the AWS VPC Edit routes interface. A route is being added for destination 80.0.0.0/16 with target 'NAT Gateway'. The selected NAT Gateway is 'nat-064b333bf7601892a' (Project_Nat_GW). The status is 'Active' and propagation is 'No'. There is a 'Remove' button and a 'Cancel' button.

Tier 1 Set up: This is the presentation Tier

Step 1

Create a launch template for Tier 1

The screenshot shows the AWS EC2 Create launch template interface. It's creating a launch template named 'Tier1_lunch_template' for resource lunch in tier 1. The software image is Amazon Linux 2 Kernel 5.10 AMI. The virtual server type is t2.micro. The purchasing option is a free tier for one year. The maximum price is set to the spot price. The request type is 'Valid to'. The interruption behaviour is 'The behavior when a Spot instance is interrupted is specified'. The summary table includes fields like Software Image (AMI), Virtual server type (instance type), Purchasing option, Maximum price, Request type, and Interruption behaviour.

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateTemplate

Network settings

Subnet: [Info](#) Don't include in launch template [Create new subnet](#)

Firewall (security groups)

A security group is a set of firewalls that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group [Create security group](#)

Security group name (required): **Tier1_SG**

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./@([!\$%^&`])

Description - required: [Info](#) Allows SSH access to developers

VPC: [Info](#) vpc-0109973844d14851 (Project_VPC) 80.0.0.16

Inbound Security Group Rules

Security group rule 1 (TCP: 22, 0.0.0.0/0)

Type: [Info](#) TCP Port range: [Info](#) 22

Source type: [Info](#) Anywhere [Add CIDR, prefix list or security group](#) e.g. SSH for admin desktop 0.0.0.0/0

Security group rule 2 (TCP: 80, 0.0.0.0/0)

Type: [Info](#) HTTP Port range: [Info](#) 80

Source type: [Info](#) Anywhere [Add CIDR, prefix list or security group](#) e.g. SSH for admin desktop 0.0.0.0/0

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security groups instead to allow access from known IP addresses only.

Summary

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... [read more](#) ami-04a6d4e047f74

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier. 100 IPv4 address blocks per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the Internet.

[Cancel](#) [Create launch template](#)

Purchasing option: Request a Spot Instance. If you don't specify a spot instance, EC2 launches an On-Demand Instance by default. Spot Instances are unused EC2 instances made available for less than the On-Demand price. Spot Instances can be interrupted, so use them for applications with flexible run times and for applications that can be interrupted. [Learn more](#)

Maximum price: Request Spot Instances at the Spot price, capped at the On-Demand price

Request type: Specify a persistent request so that interrupted Spot Instances are requested again. If you do not specify a request type, EC2 makes a one-time request by default. Persistent requests are only supported when interruption behavior is set to either hibernate or stop.

Valid to: Requests a Spot Instance without an expiry date

Interruption behaviour: The behavior when a Spot Instance is interrupted. For persistent requests, valid values are stop and hibernate. For one-time requests, only terminate is valid. If you do not specify a value, it defaults to stop.

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

CloudShell Feedback ENG ENL 12:02 PM 2024-11-12

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateTemplate

Specify CPU options

The t2.micro instance type does not support configuring CPUs.

AMD SEV-SNP: [Info](#) AMD SEV-SNP is not supported with the selected instance type and the selected AMI.

Metadata accessible: [Info](#) Enabled

Metadata IPv6 endpoint: [Info](#) Don't include in launch template

Metadata version: [Info](#) V2 only (taken required)

For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit: [Info](#) 2

Allow tags in metadata: [Info](#) Don't include in launch template

User data (optional): [Info](#) Upload a file with your user data or enter it in the field. [Choose file](#)

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd
sudo systemctl restart httpd
sudo systemctl enable httpd
```

User data has already been base64 encoded

Summary

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... [read more](#) ami-04a6d4e047f74

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier. 100 IPv4 address blocks per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the Internet.

[Cancel](#) [Create launch template](#)

Purchasing option: Request a Spot Instance. If you don't specify a spot instance, EC2 launches an On-Demand Instance by default. Spot Instances are unused EC2 instances made available for less than the On-Demand price. Spot Instances can be interrupted, so use them for applications with flexible run times and for applications that can be interrupted. [Learn more](#)

Maximum price: Request Spot Instances at the Spot price, capped at the On-Demand price

Request type: Specify a persistent request so that interrupted Spot Instances are requested again. If you do not specify a request type, EC2 makes a one-time request by default. Persistent requests are only supported when interruption behavior is set to either hibernate or stop.

Valid to: Requests a Spot Instance without an expiry date

Interruption behaviour: The behavior when a Spot Instance is interrupted. For persistent requests, valid values are stop and hibernate. For one-time requests, only terminate is valid. If you do not specify a value, it defaults to stop.

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

CloudShell Feedback ENG ENL 12:02 PM 2024-11-12

Step2

Create an internet facing ALB and create a SG for the ALB allowing only HTTP traffic.

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS CloudFormation console. The 'Basic configuration' step is selected. Key settings include:

- Scheme:** Internet-facing (selected)
- Load balancer IP address type:** IPv4 (selected)
- Network mapping:** VPC (selected)

Other visible fields include 'Load balancer name' (Project_ALB), 'Subnets' (two subnets selected from a dropdown), and 'Availability Zones' (two zones selected from a dropdown).

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS CloudFormation console, continuing from the previous step. Advanced configuration sections include:

- Mappings:** Subnet mappings for two subnets (us-east-2a and us-east-2b) with their respective IPv4 addresses assigned by AWS.
- Security groups:** A security group named 'ALB_SG' is selected.
- Listeners and routing:** A listener named 'HTTP:80' is configured to forward traffic to a target group named 'Project-TG' (Type: Instances, IPv4).

At the bottom, there are tabs for 'CloudShell' and 'Feedback'.

Step 3

Create an auto scaling group for Tier 1

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateAutoScalingGroup

aws Services Search [Alt+S]

EC2 Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional Configure advanced options

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group
Project-tier1-ASG

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

For accounts created after May 31, 2021, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended available via the UI and API until December 31, 2022.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Tier1_launch_template

Create a launch template Info

Version
Default (1)

Create a launch template version Info

Description
This launch template is for resource lunch in tier 1

AMI ID
ami-09caad68fb0de947fc

Key pair name
Ohio-KP

Launch template
Tier1_launch_template Info

AMI ID
ami-07be6779c774a5e6

Security groups
-

Security group IDs
sg-083c2f0de4d0eaac Info

Instance type
t2.micro

Request Spot Instances
No

Additional details

Storage (volumes)

Date created
Sun Nov 10 2024 12:04:41 GMT-0500 (Eastern Standard Time)

Cancel **Next**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG live 12:48 PM 2024-11-10

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateAutoScalingGroup

aws Services Search [Alt+S]

EC2 Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional Configure advanced options

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info

Override launch template
You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template
Tier1_launch_template Info

Version
Default

Description
This launch template is for resource lunch in tier 1

Instance type
t2.micro

Network Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-0109973834da14851 (Project_VPC)
03103046

Create a VPC Info

Availability Zones and subnets
Define which availability zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets Info

us-east-2a) subnet-0672bc7f2c4c4266
(project_pub_suba
03103046

us-east-2b) subnet-0a3d04e0d0f99a357
(project_pub_subb
03103046

Create a subnet Info

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort:
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Round robin:
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in the unhealthy Availability Zone to prevent load imbalance.

Cancel **Skip to review** **Previous** **Next**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG live 12:48 PM 2024-11-10

Screenshot of the AWS CloudShell interface showing the configuration of an Auto Scaling group. The user is on Step 4: Configure advanced options - optional.

Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- No load balancer (selected)
- Attach to an existing load balancer
- Attach to a new load balancer

VPC Lattice integration options

To improve networking isolation and availability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach:

- No VPC Lattice service (selected)
- Attach to VPC Lattice service

Health checks

Health checks increase availability by reducing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement starts.

EC2 health checks (selected)

Additional health check types - optional

Turn on EC2 Load Balancing health checks

Scaling can reduce it to its next probe's check.

Turn on Amazon EBS health checks

Amazon EBS health checks report volume status or attached volume stats. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance or turn off periodic health checks.

Health check grace period

The time period before the first health check and your instance fresh initializes. It doesn't prevent an instance from terminating when it fails a health check.

300 seconds

Additional settings

Monitoring

- Enable group metrics collection within CloudWatch Metrics (selected)
- Default instance warmup
- Enable default instance warmup

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies preferences

Screenshot of the AWS CloudShell interface showing the configuration of an Auto Scaling group. The user is on Step 5: Configure group size and scaling - optional.

Group size

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

Scaling

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Enter the amount how much your desired capacity can be increased or decreased.

Min desired capacity	2	Max desired capacity	5
Equal or less than desired capacity		Equal or greater than desired capacity	

Automatic scaling - optional

Choose whether to use a target tracking policy.

- No scaling policies (selected)
- Target tracking scaling policy

Instance maintenance policy

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance frontier, instance events, and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements.

- None (selected)
- For replacing unhealthy instances, will launch before terminating others.
- Launch before terminating
- Launch and terminate
- Terminate and launch
- Custom behavior

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies preferences

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
i-02be095294fe3611	i-02be095294fe3611	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2b	ec2-18-116-70-80.us-east-2.amazonaws.com	18.116.70.80	-	-	disabled	Tier1_SG	Ohio-KP
i-04ce42aa7ae95be1	i-04ce42aa7ae95be1	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	ec2-3-135-121-176.us-east-2.amazonaws.com	3.135.121.176	-	-	disabled	Tier1_SG	Ohio-KP

Step 4

Browsing for both Instances

If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you would like to let the administrator of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:
You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.
You are free to use the image below on web sites powered by the Apache HTTP Server.

Powered by APACHE 2.4





This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.

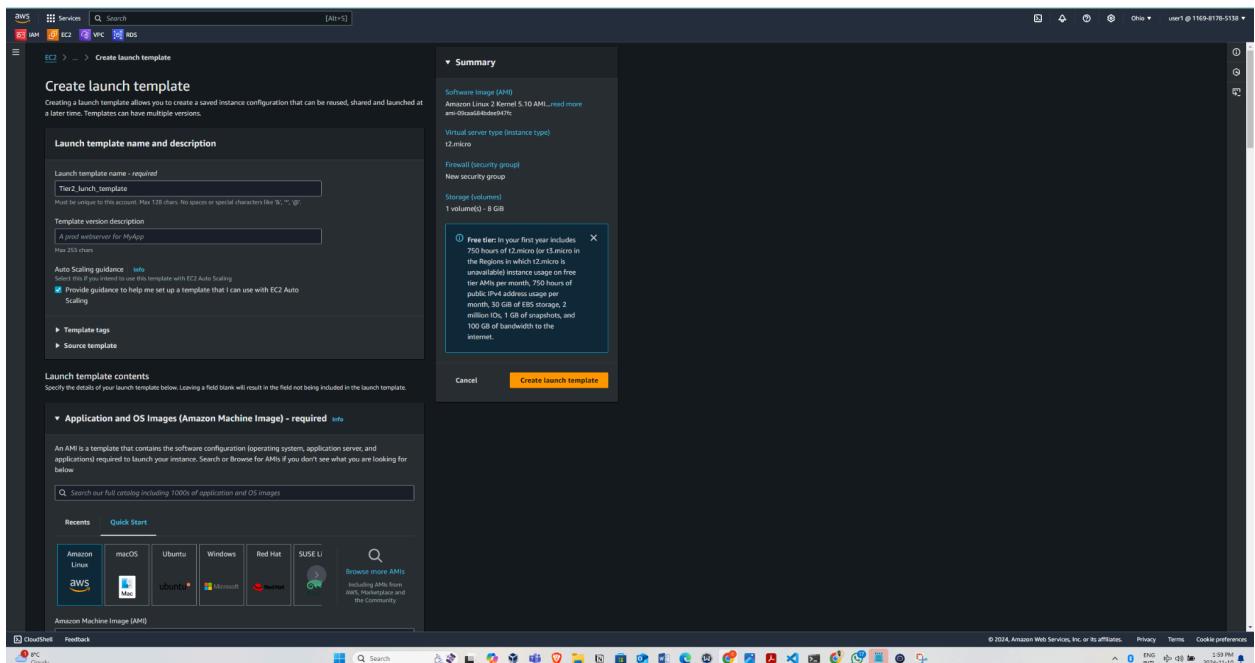
You are free to use the image below on web sites powered by the Apache HTTP Server:



Tier 2 set up: This is the application layer

Step 1

Create a launch template for tier2. The SG that would be created alongside this launch template will only allow ssh traffic from Tier1_SG and http traffic from ALB_SG.



Screenshot of the AWS CloudShell interface showing the creation of an EC2 launch template. The 'Network settings' section is open, displaying security group rules for SSH and HTTP. A modal window provides information about the free tier.

Network settings

Subnet: [Info](#) Don't include in launch template [Create new subnet](#)

Default security group: Select existing security group or Create security group

Security group name (required): tier2_sg

Description (required): This SG is for tier 2 and used to launch resources in a private subnet

VPC: [Info](#) vpc-01099e77384d14851 (Project_VPC) [Remove](#)

Inbound Security Group Rules

- Security group rule 1 (TCP: 22, sg-085c2fed640eacc, This rule will allow ssh traffic...):
 - Type: [Info](#) ssh
 - Protocol: [Info](#) TCP
 - Port range: [Info](#) 22
 - Source type: [Info](#) Custom
 - Source: [Info](#) sg-085c2fed640eacc
 - Description - optional: [Info](#) This rule will allow ssh traffic from sg-085c2fed640eacc
- Security group rule 2 (TCP: 80, sg-0104fe791a245bf3, This rule will allow http traffic...):
 - Type: [Info](#) HTTP
 - Protocol: [Info](#) TCP
 - Port range: [Info](#) 80
 - Source type: [Info](#) Custom
 - Source: [Info](#) sg-0104fe791a245bf3
 - Description - optional: [Info](#) This rule will allow http traffic from sg-0104fe791a245bf3

[Add security group rule](#)

Summary

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI [Read more](#) ami-09aa648ed4e4747c

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GB

Create launch template

Screenshot of the AWS CloudShell interface showing the continuation of EC2 launch template creation. The 'Metadata accessible' section is open, showing user data configuration. A modal window provides information about the free tier.

Metadata accessible: Enabled

Metadata IPv6 endpoint: [Info](#) Don't include in launch template

Metadata version: [Info](#) V2 only (token required)

Metadata response hop limit: [Info](#) 2

Allow tags in metadata: [Info](#) Don't include in launch template

User data - optional: [Info](#) Upload a file with your user data or enter it in the field. [Choose file](#)

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd
sudo systemctl restart httpd
sudo systemctl enable httpd
```

User data has already been base64 encoded

Summary

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI [Read more](#) ami-09aa648ed4e4747c

Virtual server type (instance type): t2.micro

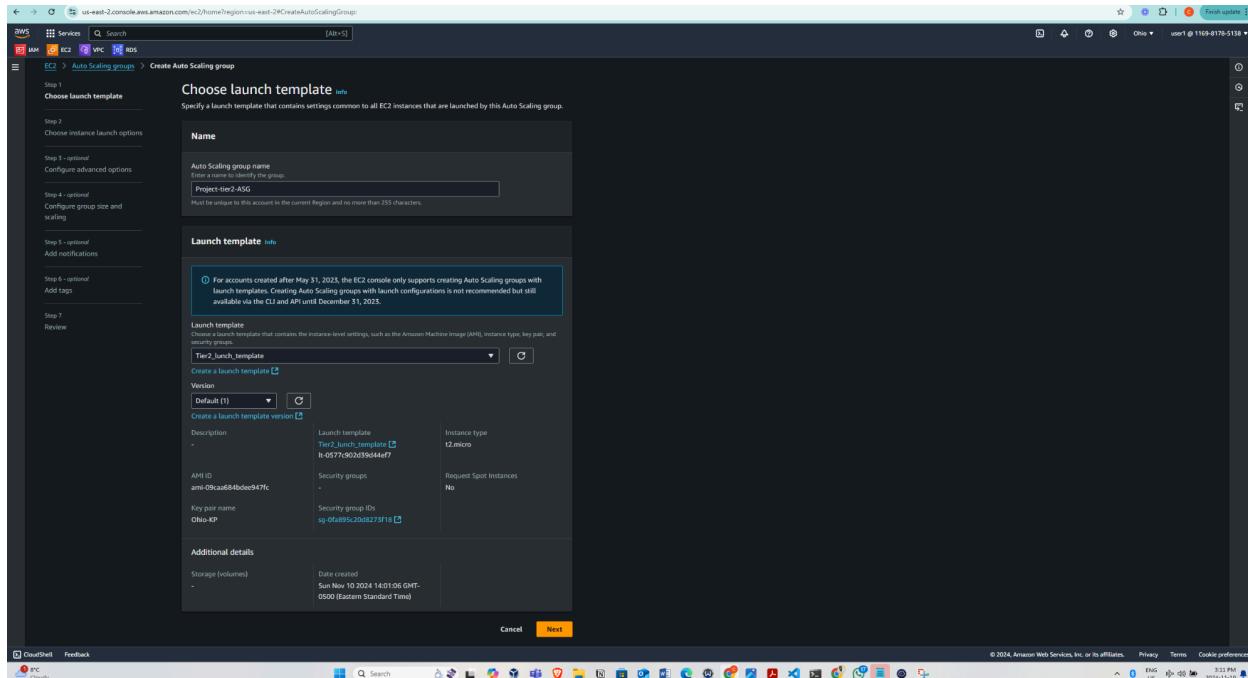
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GB

Create launch template

Step2

Create an ASG for tier2.. This ASG is specifically for launch Instances in the private subnets. The ALB I created earlier will be selected so traffic coming from the internet can access the web servers that would be launched within this ASG



Screenshot of the AWS CloudShell interface showing the 'Choose instance launch options' step of creating an Auto Scaling group.

Step 1: Choose launch template

Step 2: Choose instance launch options

Step 3 - optional: Configure advanced options

Step 4 - optional: Configure group size and scaling

Step 5 - optional: Add notifications

Step 6 - optional: Add tags

Step 7: Review

Choose instance launch options

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template

Tier 1 Launch Template: b-0f770902039604467

Version

Default

Description

t2.micro

Network

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

Choose the VPC that defines the virtual network for your Auto Scaling group.

us-east-2 (Project_VPC) (project_vpc_165) (65.0.0.0/16)

Create a VPC

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-2a (subnet-0bbb2c2d2445750170) (project_prv_sn1a) (65.0.1.0/24)

us-east-2b (subnet-0a56f21626c2ed1e) (project_prv_sn2b) (65.0.3.0/24)

Create a subnet

Availability Zone distribution

Auto Scaling automatically balances instances across Availability Zones. If launch fails in a zone, select a strategy.

Balanced best effort

If launches fail in one Availability Zone, Auto Scaling will immediately attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Balanced only

If launches fail in one Availability Zone, Auto Scaling will immediately attempt to launch in the healthy Availability Zone.

Cancel **Skip to review** **Previous** **Next**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 3:12 PM 2024-11-10

Screenshot of the AWS CloudShell interface showing the 'Configure advanced options - optional' step of creating an Auto Scaling group.

Step 1: Choose launch template

Step 2: Choose instance launch options

Step 3 - optional: Configure advanced options

Step 4 - optional: Configure group size and scaling

Step 5 - optional: Add notifications

Step 6 - optional: Add tags

Step 7: Review

Configure advanced options - optional

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones if zonal shift replacements and monitoring.

Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer

Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer

Choose from your existing load balancers.

Attach to a new load balancer

Quickly create a basic load balancer and attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups

This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

Project-TG | HTTP (Application Load Balancer: Project-ALB)

VPC Lattice integration options

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between Auto Scaling and helps you connect and manage your application across compute services in AWS.

Select VPC Lattice service to attach

No VPC Lattice service

VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service

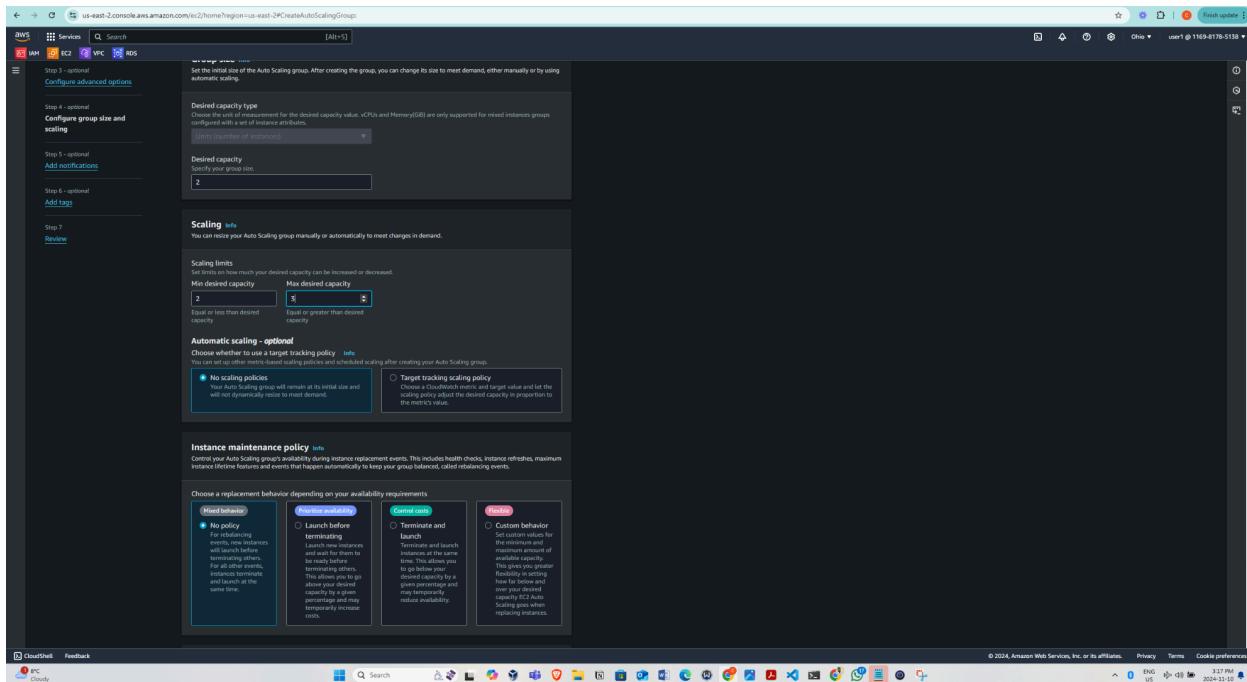
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 3:19 PM 2024-11-10



Step 3

Testing Connectivity to Tier 2 Servers:

To verify connectivity to servers in the Tier 2 (private subnet), I will use SSH to access them via a bastion host. This involves the following steps:

- SSH into Tier 1 Host (Public Subnet): First, I connect to the Tier 1 host in the public subnet.
- SSH from Tier 1 to Tier 2 Host (Private Subnet): Once connected to the Tier 1 host, I use SSH again to access the Tier 2 host in the private subnet.

Since both instances are using the same key pair, I need to transfer the private key to the Tier 1 host before I can successfully SSH into the Tier 2 host.

Step 3b

Transfer of the private key into the tier1(public) host

```

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAQEAoAp0t1MmewvewEw2h1bDmC5T1cfpm+Z25xt5dyC
HmC0dgJ11mpoSejgfwidn82VclL0-99cPdJxzf0fB4c5A7gxTCfH19P
... (long private key string)
-----END RSA PRIVATE KEY-----

```

-- INSERT --

Step 3c

First change the permission of the key to chmod 400. This makes the key private and nobody can access the key. Then ssh into the private server in tier2.

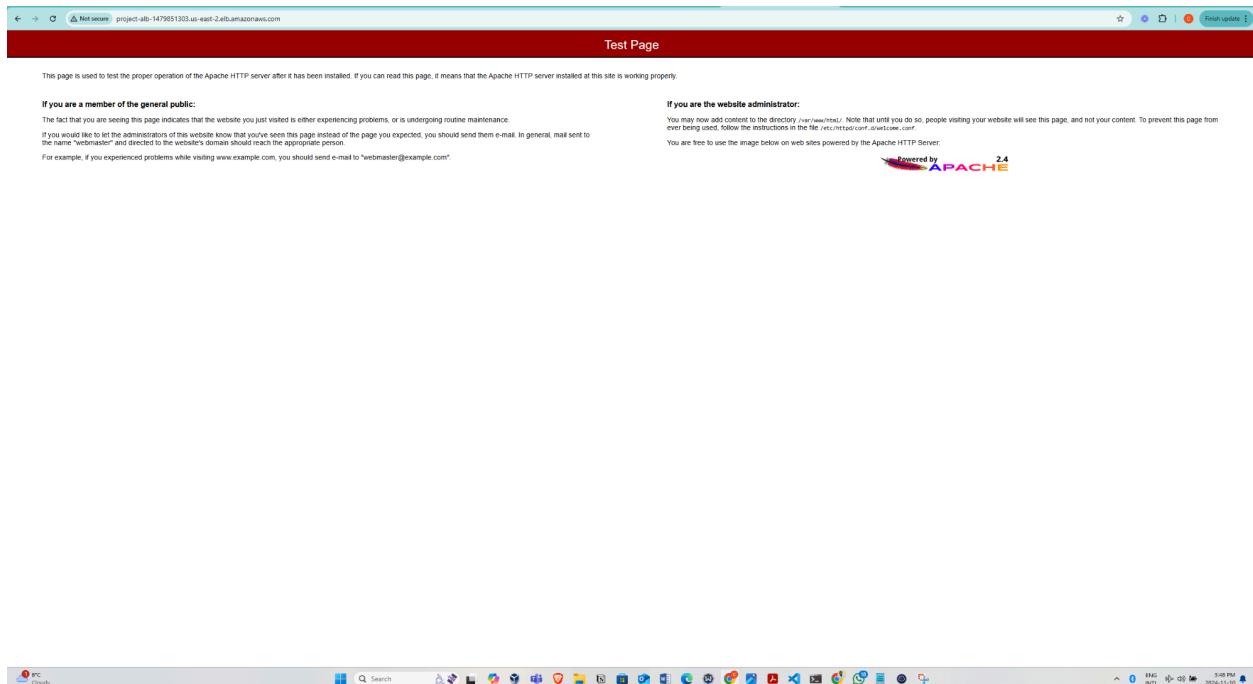
```

[ec2-user@ip-80-0-2-287 ~]$ ll
total 4
-rw-r--r-- 1 ec2-user ec2-user 1853 Nov 16 2023 privatekey.pem
[ec2-user@ip-80-0-2-287 ~]$ chmod 400 privatekey.pem
[ec2-user@ip-80-0-2-287 ~]$ ssh -i "privatekey.pem" ec2-user@80.0.1.224
[ec2-user@ip-80-0-2-287 ~]$ ls
Amazon Linux 2
-- _\_\_\_\_ Amazon Linux 2
-- _\_\_\_\_ AL2 End of Life is 2025-06-30.
-- _\_\_\_\_ A newer version of Amazon Linux is available!
-- _\_\_\_\_ Amazon Linux 2023, GA and supported until 2024-03-15.
-- _\_\_\_\_ https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-80-0-2-224 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/etc/systemd/system/httpd.service; enabled; vendor preset: disabled)
    Active: active (running) since Sun 2024-11-18 20:30:36 UTC; 10min ago
      Main PID: 1987 (httpd)
        Status: "Total requests: 37; Idle/Busy workers: 100/0;Requests/sec: 0.0588; Bytes served/sec: 234.0/sec"
   CGroup: /system.slice/httpd.service
           └─1987 /usr/sbin/httpd -DFOREGROUND
               ├─3153 /usr/sbin/httpd -DFOREGROUND
               ├─3154 /usr/sbin/httpd -DFOREGROUND
               ├─3155 /usr/sbin/httpd -DFOREGROUND
               ├─3156 /usr/sbin/httpd -DFOREGROUND
               ├─3157 /usr/sbin/httpd -DFOREGROUND
               └─3158 /usr/sbin/httpd -DFOREGROUND
Nov 18 20:30:36 ip-80-0-2-224.ec2-east-2.compute.internal system[1]: Starting The Apache HTTP Server...
Nov 18 20:30:36 ip-80-0-2-224.ec2-east-2.compute.internal system[1]: Started The Apache HTTP Server...
[ec2-user@ip-80-0-1-224 ~]$ 

```

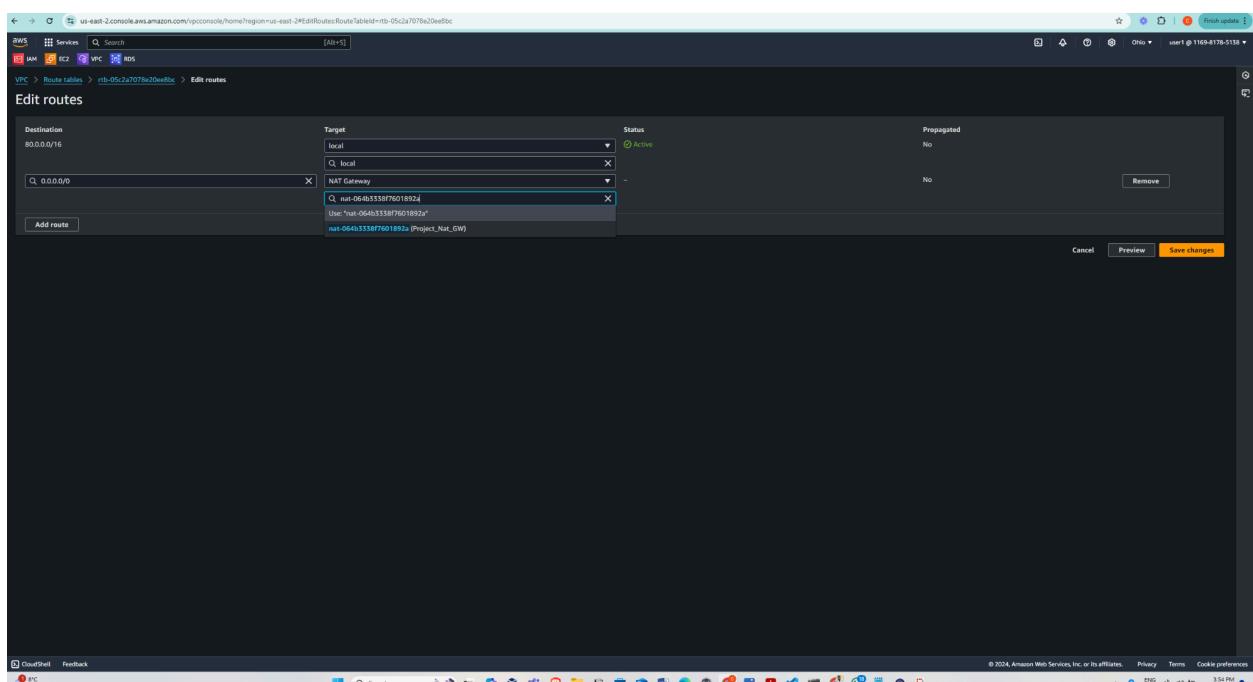
Step4

Browsing for the DNS A record of the loadbalancer to test if the application in the tier2(private) Instances are accessible.



Tier 3 Set up: This is the database layer

Step 1: Create a new route table for the databases add a route pointing to the Nat-GW



Step 1b

Associate the route table with the database subnets

The screenshot shows the AWS VPC Route Tables Details page for route table `rtb-05c2a7078e20ee8bc`. The main pane displays the explicit subnet associations:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
project_dbn2b	subnet-064546b1480551f5	80.0.5.0/24	-
project_dbn2a	subnet-082459fce077d121	80.0.4.0/24	-

Below this, there is a section for subnets without explicit associations, which is currently empty.

The screenshot shows the AWS VPC Route Tables list page. The route table `project_DB_RT` is selected, indicated by a blue selection bar around its row. The table lists several other route tables, including `rtb-0746c791c116dcf3`, `rtb-d402189aef04d4db`, `Cletus-pub-rt`, `Project_Pri-RT`, and `Project_pubRT`.

Step 2

Create database subnet groups and then i will add the database subnets i have created in availability zone 2a and 2b.

The screenshot shows the 'Create DB subnet group' wizard in the AWS RDS console. The 'Subnet group details' step is active. A 'Name' field contains 'Project_Database_SN_Groups'. A note states: 'You won't be able to modify the name after your subnet group has been created.' A 'Description' field is empty. A 'VPC' dropdown shows 'Project_VPC (vpc-010999738f46e4651)'. Under 'Add subnets', 'Availability Zones' are selected from a dropdown: 'us-east-2a' and 'us-east-2b'. 'Subnets' are chosen from a dropdown: 'subnet-002459fbef077efb1 (0.0.4.0/24)' and 'subnet-064546c1480551f1 (0.0.5.0/24)'. A note at the bottom says: 'For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.' A 'Subnets selected [2]' section shows the two selected subnets. A 'Create' button is at the bottom right.

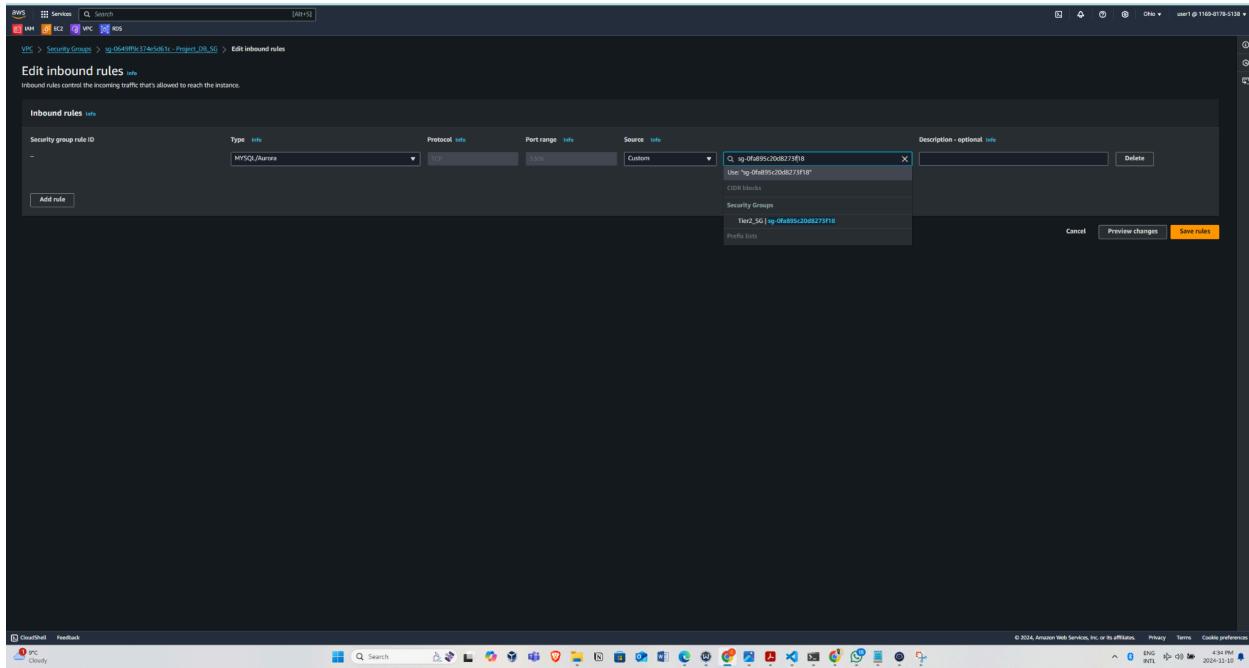
Step 3

Create database instance

The screenshot shows the 'Creating database database-1' page in the AWS RDS console. It displays a summary of the database creation process. The database identifier is 'database-1', status is 'Creating', role is 'instance', engine is 'MySQL Community', and region is 'us-east-2a'. The 'Connectivity & security' tab is selected, showing the endpoint and port (5432), networking (availability zone us-east-2a, VPC Project_VPC, subnet group project_database_sn_groups), and security (no public accessibility, certificate authority info). A 'Connected compute resources' section shows no results. A 'Actions' button is visible at the top right.

Step 4

Add a new rule allowing traffic from tier2-SG on port 3306 and delete the default rule in the SG.



Now a 3-tier web page architecture has been successfully set up.