

Step 1: An application load balancer was created with two instances.

The screenshot shows the 'Details' page for an Application Load Balancer named 'ass6'. The ARN is 'arn:aws:elasticloadbalancing:ap-northeast-1:780163399491:loadbalancer/app/ass6/03a5c09a2e8b5757'. The load balancer type is 'Application'. The DNS name is 'ass6-488438099.ap-northeast-1.elb.amazonaws.com (A Record)'. The status is 'Active'. The IP address type is 'IPv4'. The scheme is 'Internet-facing'. The availability zones are 'subnet-086f4674d3fd09da6 ap-northeast-1a (apne1-az4)' and 'subnet-07eadbc368b31ea67 ap-northeast-1d (apne1-az2)'. The VPC is 'vpc-08e1e73b775e11b82'. The hosted zone is 'Z14GRHDCWA56QT'.

Load balancer type	DNS name	Status	VPC
Application	ass6-488438099.ap-northeast-1.elb.amazonaws.com (A Record)	Active	vpc-08e1e73b775e11b82
IP address type	Scheme	Availability Zones	Hosted zone
IPv4	Internet-facing	subnet-086f4674d3fd09da6 ap-northeast-1a (apne1-az4) subnet-07eadbc368b31ea67 ap-northeast-1d (apne1-az2)	Z14GRHDCWA56QT

Step2: Load balancer was connected with custom VPC and not default VPC. The custom VPC has 2 Availability Zones.

Step 3: A relational database was created and attached to the EC2 instance

The screenshot shows the 'Databases' page in the AWS Management Console. A notification banner at the top suggests creating a Blue/Green Deployment to minimize downtime during upgrades. Below the banner, there are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and 'Create database'. A search bar is present with the text 'Filter by databases'. A table lists the databases, with one entry 'database-1' shown. The table columns are: DB identifier, Role, Engine, Region & AZ, Size, Status, and Actions.

DB identifier	Role	Engine	Region & AZ	Size	Status	Actions
database-1	Instance	MySQL Community	ap-northeast-1d	db.t3.micro	Available	2 Actions

My hashnode article: <https://hashnode.com/draft/63f6e1369f80510008b5d46b>