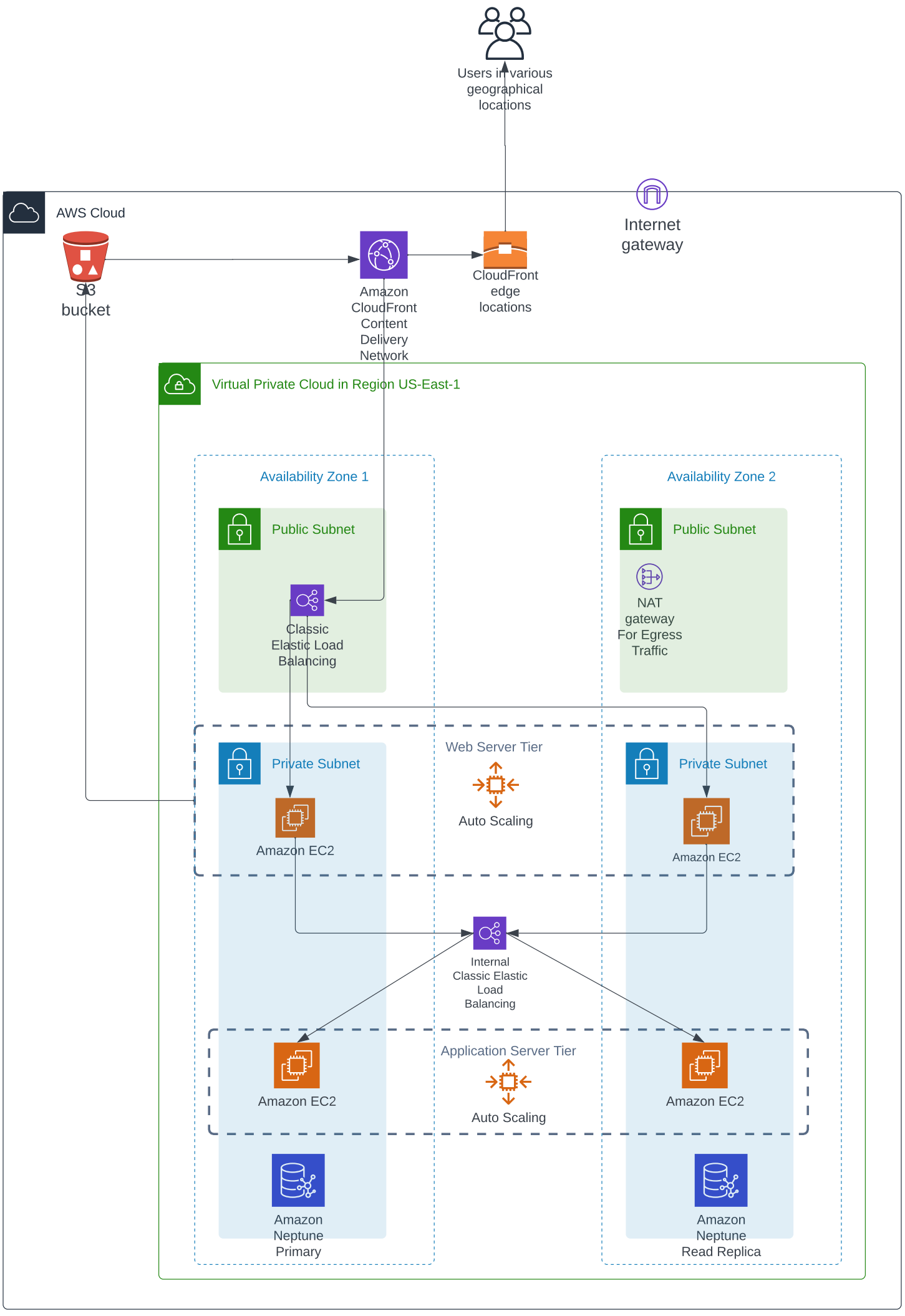


**Sizing/Load Estimates:**

The problem specifies 50K users on a new social media site. Additional metrics are needed in order to size the infrastructure and estimate the cost.  Let's make some ball park guesses and clarify the use cae.

**Use Case:** Our new social media platform prompts all users to upload one and only one image each day. Other interactions between users are text.

.

- On average, all 50K users check the site each day with peak times being before and after work as well as before bed. Assume the users are all in the US and evenly distributed across the time zones. This gives a fairly constant/flat usage profile from 7AM Eastern time until 11PM Pacific time which is 20 hours with a slowdown for a 4 hour nighttime window.
- Each users uploads 1 image from their smart phone each day for ~3MB/day, which is 150 GB/day of uploads.
- For simplicity, don't allow high resolution backdrop images or video.
- Storage for user-uploaded images for 1 month is 150 GB/day * 30 days = 4.5 TB/month.
- Let's say our users spend 30 min/day scrolling to see their friends' posts and pull 30min*5MB/min=150MB of download per day for scrolling thru posts.
- Number of HTTPS requests from CloudFront: To estimate this let's say the user flips thru their posts at 1 page per minute. This would mean 30 HTTPS requests per user per day * 50K users * 30 days = 45 million requests/month
- Assume the AWS estimate of 200KB per HTTPS request is valid.
- User text is stored in the Neptune graph databases which also contain the network of friend associations. Assume that a user posts 1K of text per day * 50K users * 30 days = 1.5 GB. Round up to 2 GB/month stored in Neptune
- For Neptune number of IO requests, assume each users generates 50 IO requests per day against Neptune. For all users, 50 * 50K * 30 days = 75 million IO requests per month
- ELB (classic) estimate is for 2 load balancers: one for the web server tier and one for the application tier.
- For ELB number of bytes per month, 45 million requests/month * 200KB/request ~ 10TB/month for each ELB.
- NAT traffic outbound from web tier handles outbound user text only.
- At any time, there would be 50K users * 30 min/user * (1/ (20 hours * 60 min/hr)) ~ 1250 users

Web tier - Could 4 (2 per AZ) M6g.medium instances support 1250 concurrent users?
App tier - Could 4 (2 per AZ) M6g.xlarge instances support 1250 concurrent users?

Daily upload to S3 thru CloudFront - 150 GB/day = 4.5 TB/month
Daily download thru CloudFront - 45 million requests/month * 3GB per photo * 1 photo/request = 135 PB/month
(This is a lot! There should be an Edge Lamda to reduce the resolution of the photos.)
Number of HTTPS requests thru CloudFront = 45 million/month

S3 storage grows at 4.5 TB/month and after 1 year will reach 540 TB ==> Use S3 Glacier IA for files older than X months

**Infrastrucuture Configuration Notes:**

- The load on a social media site will vary by time of day and by day (such as Christmas).
- The lab parameters specifies the number of users enrolled in the site is 50K. However, this does not indicate how heavily the site is used at any given time. Therefore, elasticity is needed to scale up and scale out as the user load varies. Therefore, EC2 auto-scaling is used.
- The content for a social media site includes both text posts and images. Users must be able to upload images and also view images. CloudFront presigned URLs can be used to allow users to upload their images to S3.
- User IAM roles can secure access to a user's objects in the S3 bucket.
- CloudFront edge locations allow caching of frequently viewed images as well as static and dynamic web pages.
- Classic Elastic Load balancing is used in both the web server tier and the application server tier to distribute the load across servers.
- Web Tier auto-scaling EC2 instances include 3 M6i-large instances per AZ. These are general purpose and not burstable to allow auto-scaling. Additional instances can be added to the group if needed to handle peaks in the variable load. Web traffic is network intenstive, but not CPU or memory intensive so M6i-large should be okay.
- Application Tier is more compute and memory intensive due to imag processing and running machine learning models. Again, use auto-scaling EC2 instances but beef them up to 3 M6i-4xlarge instances per AZ. These are general purpose and not burstable to allow auto-scaling. Additional instances can be added to the group if needed to handle peaks in the variable load.
- The database tier consists of a primary and a read-replica Neptune fully managed RDS instances. Neptune is used becasue graph databases are suited to querying based on relationships between entitiies (e.g. returning sets of friends).
- The graph database would contain attributes which are pointers to objects (e.g. user uploaded images) stored in the S3 bucket.
- S3 Glacier for immediate retrieval would be a good idea for archiving old images. The S3 bucket could be configured to archive images after a period of time, say 1 month. Users rarely scroll back to old posts.