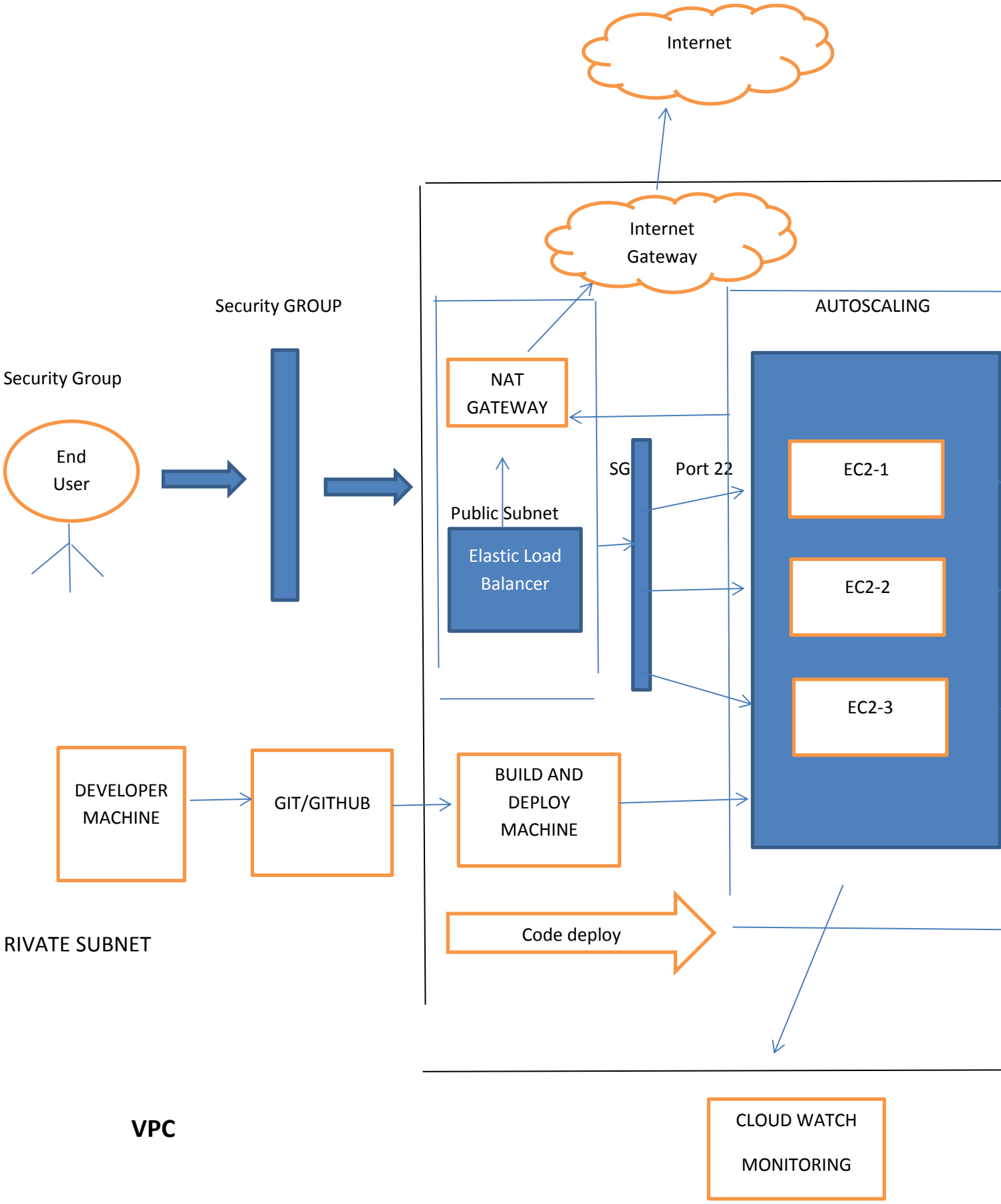
ARCHITECURE OF SNAPCHAT



SECURITY:-

Before we begin to Architect, We should Organize and classify our data into segments such as Publicly available, Available to only members of our organization, available to only certain members of our organization etc.

We should also implement least privilege access system so that people are only able to access what they need.

Also we should encrypt everything where ever possible.

Enforcing network and host-level boundary protection

- ➔ In VPC we should use security groups and Network access control lists
- ➔ how many public subnet and private subnets we use
- ➔ In the host level, how many users have access to the Hosts. Not enabling multiple user accounts
- ➔ EC2 Instance should be deployed Private subnet
- ➔ We should use Jump off node to connect to the EC2 instance instead of directly accessing it.

Enforcing AWS service level protection

- ➔ Not having multiple users able to login to the console
- ➔ Having a group setup instead of creating individual users and providing access to them.
- ➔ Having different users have different privileges within AWS.
- ➔ Having multi factor authentication enabled for the users.
- ➔ Having strong password protection and rotation policy enabled

Protecting the integrity of the Operating System on our AWS Ec2 Instance.

- ➔ Anti-Virus Installed on the OS

We can use detective controls to detect and identify a security breach.

AWS services to achieve this are.

- ➔ AWS Cloud Trail - logs every information into it (it is a regional service)
- ➔ Amazon Cloud watch – It can detect using CPU utilization.

DATA PROTECTION

- We can encrypt the data in Elastic Load Balancer, S3 or RDS.

PRIVILEDGE MANAGEMENT

- Using IAM users, and Multi Factor Authentication

Infrastructure Protection

- How we configured VPC.
- Including correct security group.
- Blocking the Particular Port for Particular users.
- Having NACL (network access control list)
- Having seperate public subnet, private subnets, NAT instances etc.

HIGH AVAILABILITY & DISASTER MANAGEMENT:-

We should become aware of failures, how they occurred, how to respond to them, and then plan on how to prevent these from happening again.

1. Failover using Elastic IPs: Elastic IP is a static IP that is dynamically re-mappable. We can quickly remap and failover to another set of servers so that your traffic is routed to the new servers. It works great when you want to upgrade from old to new versions or in case of hardware failures

2. Utilize multiple Availability Zones: Availability Zones are like logical datacenters. By deploying our architecture to multiple availability zones, we can ensure highly availability. Utilizing Amazon RDS Multi-AZ deployment functionality to automatically replicate database updates across multiple Availability Zones.

3. Maintain an Amazon Machine Image so that we can restore and clone environments very easily in a different Availability Zone.

4. Utilizing Amazon Cloud-Watch (or various real-time open source monitoring tools) to get more visibility and take appropriate actions in case of hardware failure or performance degradation. Setup an Auto scaling group to maintain a fixed size so that it replaces unhealthy Amazon EC2 instances by new ones.

5. Amazon EBS and set up Cron jobs so that incremental snapshots are automatically uploaded to Amazon S3 and data is persisted independent of our instances.

6. Utilizing Amazon RDS and set the retention period for backups, so that it can perform automated backups.

PERFORMANCE:-

It depends on how to use computing resources, efficiently to meet our requirements and how to maintain as the demand changes.

When architecting our system it is important to choose the correct kind of servers. As some applications require high CPU, some require High memory etc. With AWS we can change the type of servers, in which our environment is running on very quickly. We can also switch to running with No server at all using AWS Lambda.

An application like SNAPCHAT, would be mostly used for viewing rather than updating data, having many read replicas, would improve the performance

- Select the appropriate instance type for our system.
- Ensure that we continue to have the most appropriate instance type as new instance types and features are introduced.
- Monitor the instances after launching to ensure they are performing as expected
- We use services such as RDS to add read replicas. Reducing the load on our database and creating multiple copies of our database. Also this helps lower latency.

- We can use global infrastructure to have multiple copies of our environment, in the region that is closest to the customer location.
- We can use caching services such as ElatiCache to reduce latency.

MONITORING :-

There are 2 types of monitoring involved. 1) Manual Monitoring 2) Automated Monitoring

In the manual Monitoring we have to have a glance of Ec2 and cloud watch console dashboard to check on the health of the services.

In the Automated Monitoring , there are few tools such as Instance Status check, System Status check, Cloudwatch Alarm, Cloudwatch Logs etc.

COST OPTIMIZATION:-

- Optimally align supply with Demand. We should not over provision or under provision. Based on the demand grows, so should the supply grow.

  Examples for these are Autoscaling, which would scale the Ec2 Instance only when there is a Demand, Also another example is Lambda, which would scale up when there is a demand. We can also configure cloud watch to keep track of what our demand is.

- Decommission the resources that are no longer needed or stop resources that are temporarily not needed. Also use Cost effective resources such as Ec2 Reserved.
- Monitoring the usage and spending of the resources and Setting up billing alarm using Cloud watch and SNS.
- Optimizing over time by Managing and considering the adoption of new services.

RUNBOOK:-

- **System Overview ,** which would include , Infrastructure and Network Design , Environmental Differences, Fault Tolerance and High-Availability, Contributing Applications, Daemons, and Windows Services, Required Resources , Execution Design,
- **Security and Access Control**
- **System Configuration**
- **Configuration Management** which would have System Backup and Restore, Backup Procedures, Restore Procedures.
- **Monitoring and Alerting –** Events , Error Messages , Health Checks
- **Failure and Recovery Procedures –** Failover, Recovery, Troubleshooting Failover and Recovery
- **Maintenance Tasks -** Patching