

Software Architecture Document

Application: Simpldo

Description: Simpldo for the power of task management and simplicity.

1. Requirements

Summary of the functional requirements for the system:

1. Login with google, traditional username, and password login options.
2. Create to-dos with due date time.
3. Mark todo's as completed with a completion time
4. Archive to-dos
5. Users can add up to 1000 todo's per day

Summary of the non-functional requirements for the system:

1. Rate limit excessive requests to avoid dos attacks.
2. Low latency - Caching at client and server side.
3. High availability - 99.999% (6.05 seconds downtime per week)
4. Autoscaling to support an increase in traffic.

2. Decision & Assumptions

Initial Phase (6 months):

Assumption:

1. 10k Daily Active Users (DAU)
2. 50% of users use simpldo daily
3. Users add 10 to-dos on average per day
4. Data is stored for 1 year

Estimation:

Query per second (QPS) Estimate:

50% of 10k = 5000 DAU's

$5000 * 10 = 50,000$ to-dos per day

$50,000 / 24 / 60 / 60 = 0.57 = \sim 1$ QPS

Data Storage Estimate:

1 Todo = 100 byte

50,000 to-dos per day * 30 = 1.5 million to-dos

1.5 million * 100 = 1.5 billion = ~1.5 Gb

1.5 Gb * 6 = 9 Gb for the initial phase

Cache Storage Estimate:

Decisions:

- EC2 with Kubernetes for auto-scaling
 - Deploying pods in multiple az's in aws cloud for high availability
 - MongoDB to store todo's (Simplicity and fewer relationships are the reason)
 - Using MongoDB data archival to move old data to cold storage.
-

Pricing:

Next Phase (After 6 months):

Assumption:

1. 1M Daily active users (DAU)
2. 50% of daily users use simpldo daily
3. Users add 10 to-dos on average per day
4. Data is stored for 1 year

Estimation:

Query Per Second (QPS) Estimate:

50% of daily active users = 0.5 Million DAU's

10 to-dos per day * 0.5 Million = 5 Million todos

5M / 24 / 60 / 60 = 6 QPS

Data Storage Estimation:

1 to-do = 100 byte

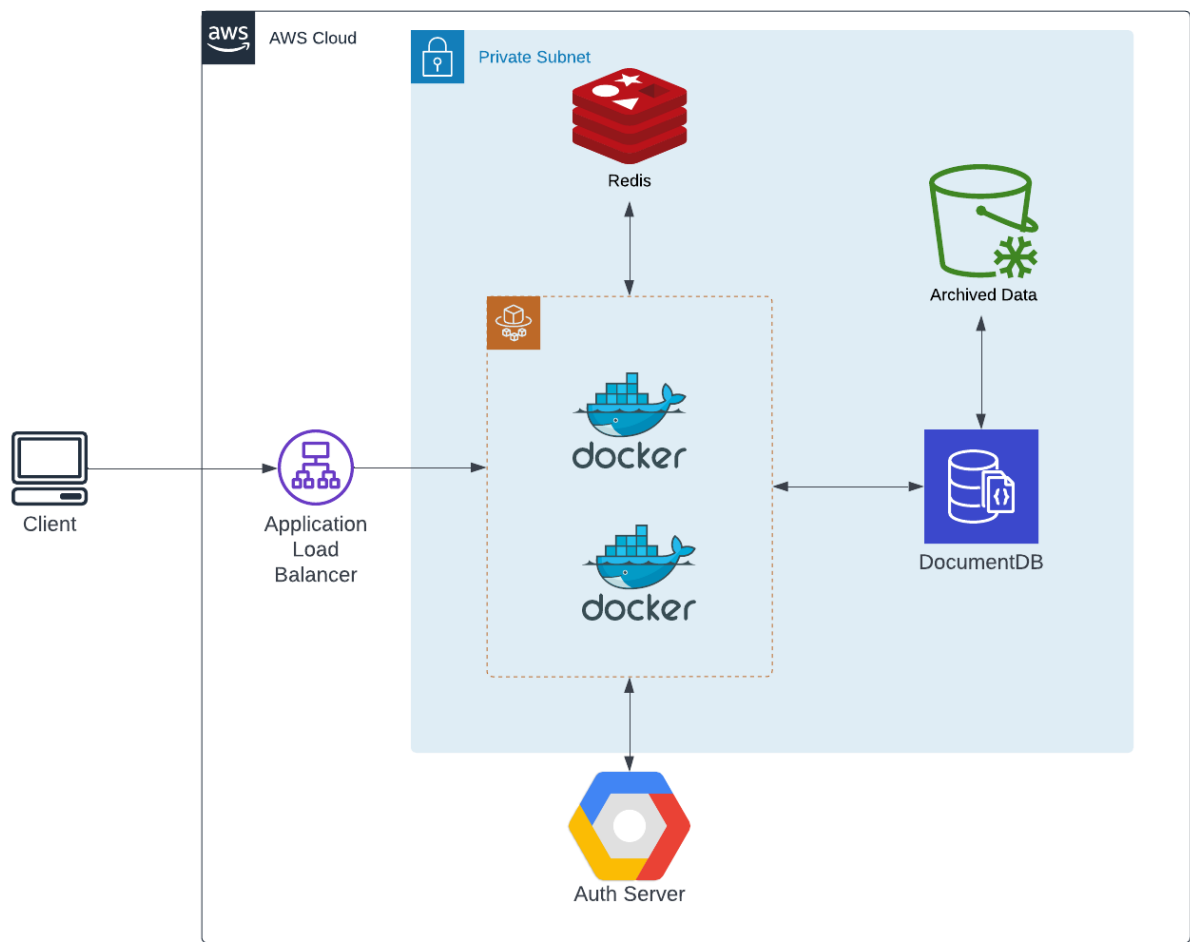
5 million to-dos * 100 bytes = 5 billion bytes = 5 GB data per day

5 GB * 30 days = 150 GB per month

Cache Storage Estimate:

Pricing:

3. Application Architecture



4. Design & Quality Attributes

- P95 less than 100ms
- Initial render time of frontend less than 50ms

5. User Experience

- Gestures to archive, complete, and move tasks.
- Single-click login experience with a persisted session for 7 days.
- Mobile application with PWA to provide offline support.

Harish Sambasivam