

How would your implementation scale if this were a high throughput service, and how could you improve that?

**If this were a high throughput service, I'd be using DynamoDB due to its ability to maintain good I/O speed and low latency with complex keys with the sha256 hash being my partition key for its table. DynamoDB also has auto scaling, and great monitoring tools with cloudwatch support. Also, considering its accessed at a high rate, Dynamo's costs are worth it in comparison to RDS.**

How would you deploy this to the cloud, using the provider of your choosing?  
What would the architecture look like? What tools would you use?

**I would use AWS as my cloud provider due to my familiarity with the products provided by Amazon. I would implement the service into Fargate to not have to manage any server infrastructure. As discussed above, I would also have backups in S3 due to Dynamos backup policy to retain user data.**

How would you monitor this service? What metrics would you collect? How would you act on these metrics?

**I would import the go lang client library for Prometheus and implement counters for post and get successes, as well as 404s. I would also implement a gauge to help determine if the service is up by the last time the api was called. Another gauge that could help is reporting the latency and tracking it via a histogram.**

**Prometheus also has an alerting component, and via set up thresholds, I would alert based on latency, if the service is up or not, and amount of 404s.**