

# CSC 547 - HW 4

Ashwin Sapre  
Shruti Kohakade  
Chinmay Srivatsa

## Regular Problems

### Problem 1

(4.1)

*Problem 4.1. The customer is(n't) always right. Consider the generic business requirements of a cloud consumer we outlined in Section 4.1. Clearly, we need to make a tradeoff between cost and any other requirement. Do we need to make tradeoffs between*

1. *Availability 24/7 and performance?*

Yes, there would have to be a tradeoff between 24/7 availability and performance. To achieve and maintain high performance, quantitative or qualitative, we would require frequent updates to infrastructure (hardware/software) which could result in down time. If the updates are done partially and asynchronously to ensure availability, the throughput and hence the performance could be affected. Performance improvements could be attempted with existing infrastructure by pushing them to their absolute limits (compute, storage etc...) but this would render them susceptible to go down and not maintain 24/7 availability.

2. *Accommodating time-varying workloads and performance?*

No, there would not be a direct tradeoff between accommodating time-varying workloads and performance. The accommodation of time varying workloads is actually implemented to maintain performance, and these two characteristics go hand in hand with each other. If accommodations are not made for time varying workloads, the performance would be suboptimal at peak loads, and if the performance is optimized for the highest peak load, there would be resource wastage during times with low load.

3. *Availability 24/7 and avoiding "vendor lock-in"?*

Maybe, there are scenarios where there could be a tradeoff between 24/7 availability and "vendor lock-in" depending on the particular vendor. If a service is offered by multiple cloud providers and their SLAs state 24/7 availability, then there need not be a tradeoff since there is no "vendor lock-in". However, If a particular service is offered by

multiple vendors, and not all of their SLAs offer 24/7 availability, then there would be a tradeoff and the architects may be forced to use particular vendors that offer 24/7 availability.

4. *Availability 24/7, accommodating time-varying workloads and performance?*

- a. If we choose to optimize availability, we will have to sacrifice performance as stated in part 1, since we would have to update partially and asynchronously to ensure availability which could affect performance. There is no direct trade-off linked to accommodating time-varying workloads in this case.
  - b. If we choose to optimize our ability to accommodate time-varying workloads, availability would not be affected since there is no direct trade-off between the two. As stated in part 2, performance and accommodating time-varying workloads actually goes hand-in-hand so we won't be sacrificing performance either.
  - c. If we choose to optimize performance, availability might go down as discussed in part 1, and accommodation of time-varying workloads is not affected as discussed in part 2.
- 

## Problem 2

(4.3)

*In Example 4.15, we have mentioned that the RIP routing protocol is an example of design that applied the KISS principle.*

1. *In what exact feature of RIP was the principle applied?*

Storing information about its neighbors only: RIP only interacts with its direct neighbors. Whenever it receives new information, it recalculates the distance vectors and sends the results to all of its neighbors. Hence it only needs to store information about its neighbors and not the entire network.

2. *With respect to this feature alone, would you say that OSPF applied/did not apply the KISS principle?*

OSPF does not apply the KISS principle with respect to this feature. A node broadcasts its information to all nodes in the network and maintains information about the entire network.

(4.8)

*Treat servers as disposable resources Principle. Consider the second principle in Example 4.16.*

1. *Describe the principle in your own words.*

As rapid provisioning of resources such as servers is a fundamental characteristic of cloud computing, companies need to forget about the concept of having fixed servers and updating them when new requirements come up. Instead, to ensure reliability, updates should always be performed on new servers with the desired configuration. The client can utilize a technique called bootstrapping, or use AWS Golden Images or Containers to automate server instantiation.

2. *Identify the ?aaS model(s) you could apply this principle to.*

We can apply this principle to IaaS, because it is the only ?aaS where the client is dealing with servers directly, and therefore, has the opportunity to apply this principle. For all the otheraaS models, provisioning of servers and their configuration happens behind the scenes.

3. *Identify a business requirement you could apply this principle to*


We can apply this principle to the 24/7 availability business requirement, since following this principle reduces the chances of faults.

---

## Problem 3

*(4.2) Is the process necessary? Consider the process we outlined in Section 4.4.1. It suggests that six steps are sufficient to create a “good” architecture. Let’s not question sufficiency.*

*However, are the six steps really necessary?*

1. Set  $k = 1$
2. While  $k < 7$
3. Remove step  $k$  from the process in Section 4.4.1.
4. Consider a new process without step  $k$ . Is it “good”? Justify your answer.
5. Set  $k = k+1$
6. Endwhile
7. Listen to  *Pink Floyd - We Don't Need No Education Lyrics in Description!*

Ans:

Given process in section 4.4.1 -

1. Create a list of all business requirements for which an architecture is to be produced.
2. Convert this list into another list of technical requirements.

3. Search for options to address each technical (and hence business) requirement.
4. Evaluate tradeoffs for each option.
5. Select an option for implementation.
6. Document the selection in a design document and/or architectural diagrams.

1. Iteration 1- Remove step 1(Create a list of all business requirements for which an architecture is to be produced) from the process in Section 4.4.1

The new process without step 1 is **not a good process** for following reasons -

- a. The objective of creating a list of business requirements is -
  - i. To gain agreement with stakeholders
  - ii. To provide a foundation to understand what the solution needs to do to satisfy the customer's and business' needs.
  - iii. To provide an input for the next phase of the project
- b. Business requirements are important because they are the foundation for all subsequent steps in the process, describing what inputs and outputs are expected from every step of the process.
- c. Business requirements describes what a system would look like from a business perspective.
- d. Hence, without defining business requirements, the new process won't be a good process.

2. Iteration 2- Remove step 2(Convert this list into another list of technical requirements) from the process in Section 4.4.1

The new process without step 2 is **not a good process** for following reasons -

- a. The objective of creating a list of technical requirements is -
  - i. To describe how the software should function and what should be its behavior.
  - ii. To provide the understanding of technical aspects of the product and its development process.
- b. Technical requirements are important because they help in making critical decisions like determining the budget for the project, dividing the work into tasks and setting timelines for their completion, and determining the risk-management aspects of the project.
- c. Hence, without defining technical requirements, the new process won't be a good process.

3. Iteration 3- Remove step 3(Search for options to address each technical (and hence business) requirement) from the process in Section 4.4.1

The new process without step 3 is **not a good process** for following reasons -

- a. The objective of step 3 is to find possible options to address the technical and business requirements.
- b. This step is important because if we don't find all the options then there might be a case that we select the expensive and non optimized approach to address the technical requirements that will affect the performance, budget and timelines of the completion of the project.

- c. Hence, without searching options to address technical and business requirements, the new process won't be a good process.
  - 4. Iteration 4- Remove step 4(Evaluate tradeoffs for each option) from the process in Section 4.4.1
    - The new process without step 4 is **not a good process** for following reasons -
    - a. The objective of step 4 is to evaluate the tradeoff for each option found out in step 3. It exhausts all the options and finds their pros and cons of implementation.
    - b. This step is important because it provides the basis for selecting the best option to implement the project.
    - c. Hence, without finding tradeoffs for each options, the new process won't be a good process.
  - 5. Iteration 5- Remove step 5(Select an option for implementation) from the process in Section 4.4.1
    - The new process without step 4 is **not a good process** for following reasons -
    - a. The objective of step 5 is to select an optimal option for the implementation of the project.
    - b. This step is important because it compares all the tradeoffs evaluated in step 4 and based on the technical and business requirements it allows the team to select the optimal one.
    - c. It also helps the team to set the timelines, prepare execution plans which will effectively lead to the completion of the project.
    - d. Hence, without selecting the best option to implement the project, the new process won't be a good process.
  - 6. Iteration 6- Remove step 6(Document the selection in a design document and/or architectural diagrams) from the process in Section 4.4.1
    - The new process without step 6 is **not a good process** for following reasons -
    - a. This step is important because it helps to organize and deliver thoughts to stakeholders, which helps them understand the high level of architecture and also how your design decisions will satisfy the user needs and business requirements.
    - b. Documentation of the selection can help the new member of the team to understand the decisions made and high level architecture of the project.
    - c. Hence, without step 6, the new process won't be a good process.
- 

## Problem 4

(4.10)

#### *Problem 4.10. Best Practices for SaaS*

1. *What objective is the best practice addressing?*

Main objective addressed by almost all the best practices for SaaS is meeting business requirements and ensuring consumer satisfaction in an efficient and future proof manner helping all the parties involved.

2. *Does it give advice to an architect or implementor?*

Best practices for SaaS contain advice and recommendations for both the architect/designer and implementer. The advice and recommendations directed towards the architect and designer mainly aims to assist the provider in providing quality services in the most efficient, cost effective and future proof way possible. The advice and recommendations directed towards the implementer on the other hand mostly deal with the delivery and setup of the consumer facing components of SaaS.

3. *Does it discuss tradeoffs?*

Since SaaS platforms may store sensitive information about its users, the developers have to optimize for security. This may conflict with the application's performance levels.

---

## Bonus Problems

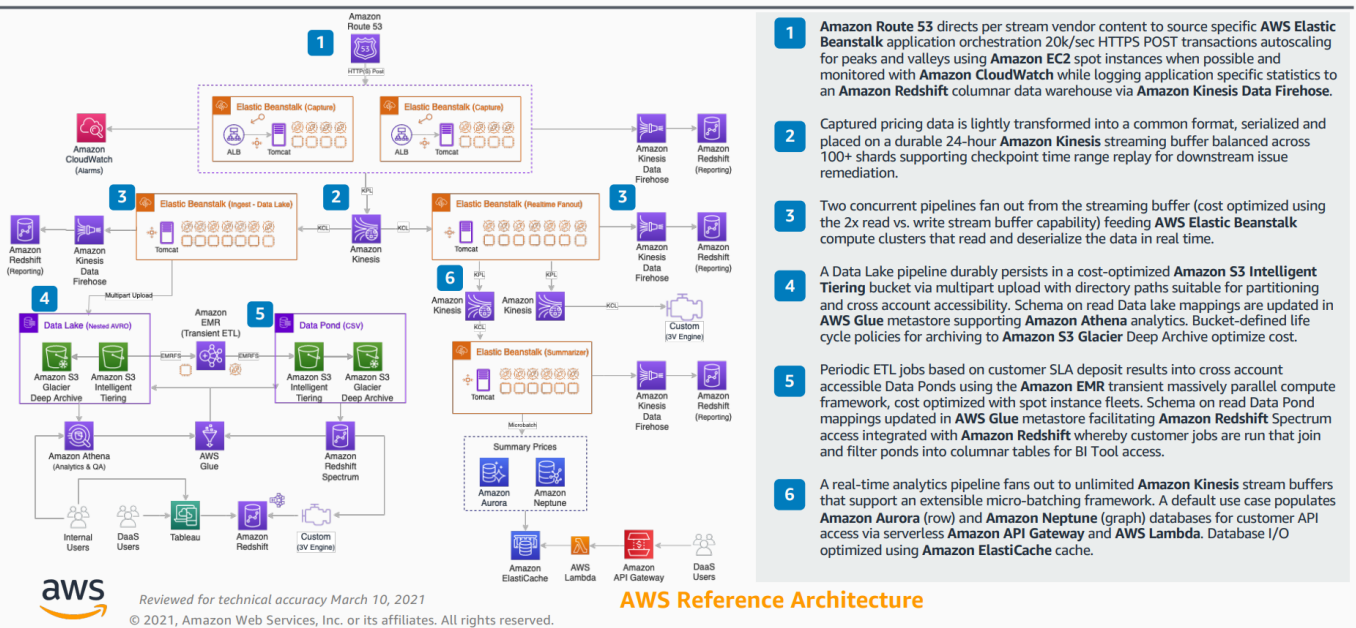
### Problem 5

Check <https://aws.amazon.com/architecture/reference-architecture-diagrams/> it is a repository of reference architecture diagrams. You can filter by technology category or industry.

# Streaming Airline Ticket Shopping Insights

## Predictive Analytics on Timeseries Airline Shopping Data

3Victors implemented an AWS cloud-based architecture to capture and durably store over 10Tb of daily streamed air shopping data messages into a Data Lake. Dozens of ETL jobs run at regular intervals to populate use case specific Data Ponds. Simultaneously, the implementation provides an extensible, real-time predictive analytics pipeline for demand forecasting and deal classification.



### 1. Select an architecture diagram of interest to you.

## Streaming Airline Ticket Shopping Insights (Predictive Analytics on Timeseries Airline Shopping Data)

Client - 3Victors can ingest more than a billion worldwide travel searches and 230 billion priced itineraries every day from multiple data sources.

### 2. Describe, in your own words, the problem solved (what were the objectives?)

The objective of this cloud based solution is-

- To capture and store over 10Tb of daily streamed air shopping data.
- To provide real time predictive analysis for demand forecasting and deal classification.

### 3. The diagram uses “standardized” AWS icons. Explain them.

#### a) Amazon Route 53 -

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It connects user requests to internet applications which are running on AWS or on-premises. It is used to manage network traffic globally, build high availability applications and to set up private DNS.

#### b) Amazon Kinesis -

Amazon Kinesis allows to collect, process and analyze real-time, streaming data.

It provides the cost effective way of scaling, processing and analyzing real time data as it arrives and responds quickly.

c) Amazon Kinesis Data Firehose -

Amazon Kinesis Data Firehose is a service that captures, transforms, and delivers streaming data to the different data stores, analytics services and data lakes.

It provides streaming of data into data lakes and warehouses, boosts security and build ML streaming applications.

d) Amazon Redshift -

Amazon Redshift uses SQL to analyze structured and semi-structured data across operational databases, data lakes and data warehouses.

e) Amazon Cloudwatch -

It monitors the AWS resources and the applications that you run on AWS.

f) Elastic Beanstalk -

It helps in deploying web applications.

g) AWS Glue -

It is a data integration service that makes data preparation simpler, cheaper and faster.

h) AWS Athena -

It is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.

i) AWS Aurora -

It is a relational database service with full MySQL and PostgreSQL compatibility.

j) AWS ElastiCache -

It is a fully managed in-memory data store and cache service

k) AWS lambda -

It is an event-driven, serverless computing platform provided by AWS.

l) Amazon API Gateway -

It is a fully managed service that allows to create, maintain, publish, monitor, and secure APIs at any scale.

m) Amazon S3 Glacier -

Amazon S3 Glacier is an online file storage web service that provides storage for data archiving and backup.

4. *List all the services involved.*

a) Amazon Redshift

b) Amazon Kinesis

c) Amazon EC2 Spot Instances

d) Amazon EMR



5. *Describe the services.*

- a) Amazon Redshift
  - i) It is a fast, fully managed data warehouse service which is simple and cost-effective to analyze data using existing business intelligence tools.
  - ii) It is optimized for datasets ranging from hundreds gigabytes to petabyte.
- b) Amazon Kinesis
  - i) It allows to collect, process and analyze real-time, streaming data.
  - ii) It provides the cost effective way of scaling, processing and analyzing real time data as it arrives and responds quickly.
- c) Amazon EC2 Spot Instances
  - i) It provides the advantage of unused EC2 capacity in the AWS cloud.
  - ii) It is tightly integrated with AWS services such as Auto Scaling, EMR, ECS, Data Pipeline and AWS Batch.
  - iii) It provides the ability to choose to launch and maintain your applications running on Spot Instances.
  - iv) We can combine spot instances on demand.
- d) Amazon EMR
  - i) It is a cloud big data solution for interactive analytics, large scale data processing, and machine learning using open-source frameworks such as Apache Spark and Presto.
  - ii) It allows processing of real time data streams.

6. *Comment on features of the design - e.g., is it scalable? highly available? secure? etc.*

- a) It is extremely fast -
  - By using services from AWS this application processes data extremely fast in real time which gives the customer better insight.
- b) It is operated at reduced cost -
  - By moving the whole application to cloud and running a big data environment on AWS cloud, 3Victors is reducing operational costs as it does not have to maintain the servers. AWS is taking care of it at reduced cost
- c) It is highly scalable and efficient in storage -
  - 3Victors can ingest more than a billion worldwide travel searches and 230 billion priced itineraries every day from multiple data sources in AWS which would have been impossible for an on-premise storage.
- d) It is highly available -

As AWS cloud provider takes care of high availability of the AWS resources, the application has the ability to provide service which is highly scalable.

e) It is secure -

The application is secure as AWS provides the most secure global infrastructure.