Robert Bara

**HW1**

1. Describe the workflow of https protocol as we did in the class.

* Https stands for HyperText Transfer Protocol Secure, and is primarily used to send data between a client and server. This is done by encryption using a Transport Layer Security (TLS) protocol in which two encryption keys are generated. The private key is used to decrypt information that is encrypted by the public key. The public key is available to anyone that interacts with the server, while the private key is particular to the client. The protocol basically works by allowing the client to send a request using a TCP connection, in which the server then accepts the TCP connection and sends a response. Finally, the TCP connection is closed.

2. Using the hybrid system DNS example we discussed during the class, architect a simpler  
hybrid system without a central DNS vpc.

3. During the class, we discussed how to use cache. Caching contents to avoid fetching informa-  
tion through slow I/O process is a popular method to improve the system efficiency. However,  
we have to consider how to refresh our cache data. For example, if we cache some data in the  
client side, when we send our http request to the web server, we would like to know if we need  
retrieve the data from the backend database. This requires us to understand if the cached  
content in the client side is the same as in the backend database. Usually, we use hashcode  
to do the comparison. Do some research of what is the hashcode and how the hashcode is  
generated from the content and answer the following questions:

* Can I use hash key as an unique id? Specify the reason.
  + You could use a hash key as a unique id, however depending on how much information is stored, there could be a possible collision due to data containing the same hash key, therefore being stored within the same section in memory.
* How to find the difference between two list of strings
  + For a list of strings, you could use the ascii values of the string as an input to create a hash id, then store your data into the table.
* How to handle the duplication of the hash code?
  + If two different data types have the same hash id, then we could avoid a collision by writing our hash code so the hash table is a table of linked lists, in which data can then be stored to a linked list of ‘N’ nodes, and we can traverse the linked list at the unique id until we find the target data.
* How to deal with a shared hash table that could be updated/read/written/deleted by  
  multiple users simultaneously?
  + You can deal with multiple users by determining a method to deal with collisions, this can be done in a variety of ways such as implementing an algorithm like a linked list which contains nodes, and each user can work within their own node within the list. The only conflict there would be if a user is using data at the same node, in which case some sort of merge algorithm should be created.
* Assume you are an engineer for a company. The company needs to build its hybrid  
  system which means some applications are going to be migrated to AWS cloud. You  
  need set a direct connect from the on-prem data center to the AWS cloud VPC. You  
  need your VPC in the AWS could refer any application on-prem and any application  
  on-prem could use resources in your AWS cloud.