**Q.1. Describe a high level design for a Distributed Code Deployment System.**

## Ans:-> Building :- Let’s say a developer wants to deploy code corresponding to a commit SHA. What seems to be a right choice is a Queuing mechanism to queue all the requests. And worker pool building the code commits in FIFO manner. Workers will then store the binary in blob store.

-> Since we want to persist the history of jobs. We can think of a SQL table for storing the job entries (this can represent our queue).  
We define our status enum as { QUEUED, RUNNING, FAILED, COMPLETED, CANCELLED}.

-> The fact that we have SQL database, we have ACID transactions. This enables our X number of workers to query and update the jobs as every run is a transaction and hence concurrency safe.

## -> Deploying :- We assumed that the build takes 15 mins and let’s say replication takes another 5 mins. So for us to meet the 30 min deadline for entire deployment, we are left with 10 mins. For 100K machines to download a 10 GB file from the blob store over network seems unreasonable. Hence we can create a Peer-to-Peer network. All the machines in a region are part of a Peer to Peer network, this will enable them to download multiple such binaries really fast.

# Q.2 write a function for an Array and a target sum value to find if there are 3 numbers whose sum is equal to the given target sum value.

**Ans**:->

public class FindNumber

{

public void numbers(int arr[], int sum)

{

int l = arr.length;

int flag=0;

for (int i = 0; i < l - 2; i++)

{

for (int j = i + 1; j < l - 1; j++)

{

for (int k = j + 1; k < l; k++)

{

if (arr[i] + arr[j] + arr[k] == sum)

{

System.out.println(arr[i] + ", " + arr[j] + ", " + arr[k]);

flag=1;

}

}

}

}

if(flag==0)

{

System.out.println("Numbers Not Found");

}

}

public static void main(String args[])

{

FindNumber fn= new FindNumber();

int arr[] = { 1, 2, 6, 9, 3};

int sum = 13;

fn.numbers(arr,sum);

}

}