HDFS architecture:

1.Hvad er grunden til, at man ikke kan bruge SQL til lagering af big data, når filer kan konverteres til byte/byte array og gemmes i en rational database?   
Årsagen til dette er, at der kan være helt op til flere petabytes af data, som overskrider de grænser for data som en relational database kan have for at virke optimalt, derved bliver det uoverskueligt at skulle administere denne database på grund af, at svar tiden er for langsom.

2.Hvilket architecture er HDFS baseret på?  
HDFS’s arkitektur er baseret på et ”master/slave” eller ”primary/secondary” cluster til lagring af dataen.

3.Hvad er NameNode contra DataNode?

NameNode er det primære filsystem som eksempelvis håndtere filadgang, hvor at datanode er de secondary nodes hvor dataen bliver gemt.

4.HDFS architecture i forrige slide image viser en serie af DataNode, kan man godt bare have en enkelt?  
Et optimalt HDFS system har flere datanodes, men kan godt fungere med et enkelt.

5.Kan HDFS installers på en Windows, Linux eller begge?  
HDFS kan godt installeres på begge operativsystemer, men er mest egnet til linux.

6.Kan man tilgå HDFS (skrive kode mod HDFS ligesom man skriver SQL kode mod en SQL server)?  
Man kan godt scripte funktionalitet til et HDFS filsystem. File system shell har prefixet hdfs og diverse kommandoer som kan gøres brug af for at automatisere div. Opgaver.

YARN architecture:

1.Hvad står bogstaverne: YARN for?  
YARN står for ”Yet Another Resource Negotiator”

2.Hvad er YARN ansvarlig for ifølge det viste architecture model på forrige slide?  
YARN er ansvarlig for ressource manageren og styre de node managers der er tilstede ved hvert datanode.

3.Hvad er YARN’s resource manager vs. node manager fra det viste model for architecture?

Ressource manageren er ansvarlig for de node managers der er, men har ikke andet data, hvor en node manager har et datanode/container med data.

4.Hvordan skal begrebet “container” fra det viste model for architecture forstås?

Begrebet ”container” skal kunne ses som et virtuelt miljø hvor der er givet ’x’ antal cpu- og hukommelse ressourcer til, som derved skaber en ”container”.