

Suggested New Exam Questions for CLC_Oral_Exam_Prep_Ans.pdf

Lesson 1: Cloud Computing

- Which are the business drivers for cloud computing? (e.g., Capacity Planning, Cost Reduction, Business Agility)
- Suppose that you would like to deploy a web application composed by: a frontend server to receive http requests and produce http reply/answers, a business logic server to build the content of the web pages; a database server, used by the business logic to extract information. Suppose also that the business logic component should scale (horizontally). Which system architectural style/model could be used for the deployment? (Consider options like Multi-tier, Microservice, SOA)
- In what differ microservices and the SOA? (Consider aspects like communication mechanisms, data handling, size of services, coupling)
- The NIST definition of Cloud computing mentions that "Computing resources are pooled to serve multiple consumers using a multi-tenant model". What does multi-tenancy mean?
- Which of the following definition of capacity planning is correct? (e.g., a methodology to determine the right amount of capacity when needed to avoid Over-provisioning and Under-provisioning)
- What is a microservice architecture? (e.g., an architectural style structuring an application as a collection of loosely coupled, independently deployable services organized around business capabilities)
- Which of the following are challenges in the microservice architecture? (e.g., Finding the right set of services, developing/testing/deploying complex applications, efficient resource management, orchestration, choreography)
- Which of the following are benefits of the microservice architecture? (e.g., Maintainability, independent deployability/scalability, fault isolation, easier adoption of new technologies)
- Consider the difference between communication mechanisms adopted by SOA and by microservices. What of the following assertion is correct? (e.g., Microservices using lightweight/open-source vs. SOA potentially using heavyweight/proprietary mechanisms)
- Suppose you want to add a new record in a database using a REST web service. Are HTTP PUT and GET requests (e.g., `PUT /users` with XML body vs. `GET /adduser?name=Robert`) equivalent for this purpose?
- What represents a single centralized executable business process that coordinates the interaction among different services? (Service orchestration)
- Why should services in a microservice architecture be loosely coupled? (e.g., To minimize dependencies)
- How data are typically handled in a microservice architecture? (e.g., Each service has its own database and data model vs. shared global data model)
- Who is responsible for Security in the cloud, the provider or the customer, or is it shared?
- Consider the Service Oriented Architecture implementation of inter-process communication. Give a description of the main components in the SOA model and of the interaction flow between two nodes communicating using the SOA model.
- What is a capacity planning match strategy? (e.g., adding capacity in small increments as demand increases)

- Assume you are using a service offered via a web application and you do not pay for the use of the service. Which feature should the service have to be classified as a cloud service? (e.g., The service should provide the illusion of having infinite resources)
- What is the difference between Horizontal duplication and Functional decomposition in microservices?
- Let us consider an independent components event-based system. What should a subscriber component provide? (e.g., A callback function executed when an event is activated)

Lesson 2: Containerization

- What is the difference between hypervisor of type I and hypervisor of type II? (e.g., Type I runs on bare metal, Type II runs on host OS)
- Consider the Kubernetes architecture. If you configure container autoscaling mechanisms, what component might be considered a core part of this process that monitors resource usage? (A resource usage monitor module)
- Consider the Kubernetes architecture. What is the component that takes scaling decisions? (kube-controller-manager)
- Consider the Kubernetes architecture. What is the component that decides on which Kubernetes node to start a new container? (kube-scheduler)
- Suppose you must store a container's data on a remote host or a cloud provider rather than locally. Which Docker storage option is the most appropriate (e.g., Volumes, Bind Mount, Tmpfs mount)?
- Consider the docker CLI command `docker run -dp 3000:3000 getting-started`. Which actions are performed by this command? (e.g., run in detached mode, port mapping)
- Consider the docker CLI command `docker build -t getting-started ..`. What actions does this command perform? (e.g., build an image from a Dockerfile in the current directory and tag it)

Lesson 3: Autonomic Computing

- The Ansible automation tool is defined to be an "agentless tool that runs in a push model". What does a "push model" mean in this context? (e.g., management tool passes instructions to existing remote management frameworks)

Lesson 4: Cloud Mechanisms

- Which type of scalability is facilitated by microservices? (e.g., Scaling of single functions)
- Assume you set up a multi-threshold scale-out policy (e.g., based on CPU utilization). Why is it important to wait a certain time interval (cooldown/warm-up) before starting another scaling action after one has occurred? (e.g., to avoid ping-pong effects, allow instances to warm up)
- An autoscaling algorithm is considered proactive when the scaling decision is taken on the basis of a predicted value of the scaling metrics (e.g., predicted CPU utilization in the next 5 minutes). True or False?

Lesson 5: Cloud Storage

- Consider the BigTable datastore architecture. What are some duties of the Tablet server? (e.g., manage tablets, handle read/write requests to loaded tablets, split large tablets)
- Consider the architecture of HDFS. What are some actions performed by the NameNode? (e.g., manages File System Namespace, controls file access, stores metadata, checks DataNode liveness)

- Consider the Google File System architecture. What components (or component) are responsible to handle file access? (e.g., Application, master, chunk server)
- Researchers and engineers at Google designed the Google File System taking into account the file access model. What is a most frequent type of file access? (Sequential read / Append write - note different answers in source files)
- What is an RDD transformation in Spark? (Provide an example like `map` or `filter`, distinct from actions)
- In a MapReduce computation, if the master node fails, what action should be taken to preserve the result of the computation? (e.g., restart entire computation)
- Why does the Google File System use large chunks? (e.g., optimize for large files, reduce metadata)
- Which is the key operation performed by the MapReduce framework (like Hadoop) that allows the Reduce task(s) to be properly executed after the completion of Map tasks? (Shuffle and sort)
- What is the key idea behind MapReduce? (e.g., Divide and conquer algorithm design paradigm)
- What is an RDD action in Spark? (Provide an example like `collect` or `reduce`, distinct from transformations)
- In a MapReduce computation, if a map worker node fails, what action should be taken to preserve the result of the computation? (e.g., re-schedule and re-execute tasks assigned to the failed worker)
- Consider the Google File System architecture. What component(s) are responsible for creating a file? (The master)
- A Bigtable is a sparse, distributed, persistent multidimensional sorted map. Which combination of keys (e.g., Row key and column key; Row key, column key and timestamp; Row key, family key and timestamp) does *not* uniquely locate a specific cell version or does not allow to locate a cell?
- Bigtable uses timestamps to maintain versioning of the data stored. What is a mechanism to garbage collect cell versions automatically? (e.g., Keep only new-enough versions, or last N versions)
- Considering the write protocol of HDFS, who is in charge of sending the "write successful" acknowledgement back to the client (or to the NameNode, depending on specific interpretations of "successful")? (The client often receives the final success ack after the pipeline confirms)
- Hadoop MapReduce optimizes data locality. If data-local execution isn't possible, what alternative deployment policies are used? (e.g., Rack local deployment)
- Provide and comment on an example of a MapReduce program.
- Explain the difference between a transformation of an RDD and an action on an RDD in Spark (bring an example).
- Researchers and engineers at Google designed the Google File System taking into account file characteristics. What is the typical range for file size? (e.g., Kilobytes to Gigabytes, Megabytes to Terabytes)
- What is the chunk size in the Google File System? (e.g., 64MB)
- Consider the BigTable datastore architecture. What are some responsibilities of the Master node? (e.g., assign tablets, detect server addition/expiration, balance load, garbage collection of files in GFS)