DA 410/510 Final

Spring 2026

Name: _____ **Date:** _____

Instructions

- Answer all questions. Show your reasoning clearly and state any assumptions you make.
- Use one novel application idea throughout the exam. The application should use cloud computing in a significant way, not merely be a static website hosted on a virtual machine.
- For compare/contrast questions, address tradeoffs directly rather than listing isolated facts.
- For design questions, explain both what you would do and why.
- You may refer to AWS service names when helpful, but focus on architectural reasoning rather than memorizing product names.

Questions

1. Propose a novel application that uses cloud computing in a significant way. Identify the target users, the main user-facing workflow, and one cloud-dependent capability that would be difficult or impractical to provide from a single local server.

2. Describe a high-level architecture for your application using the three-tier model. Identify the presentation tier, application tier, and data tier. Then explain how requests flow from a user action to a stored result. Your answer should include at least one backend service and at least one persistent data store.

3. Choose how you would deploy the application tier. Compare running your backend directly on virtual machines with running it in containers. Then justify the approach you would use for this application. Your answer should address isolation, startup or replacement time, resource overhead, and operational complexity.

4. Read the Dockerfile below and answer the questions that follow.

```
FROM python:3.12-slim

WORKDIR /app
COPY requirements.txt /app/requirements.txt
RUN pip install --no-cache-dir -r /app/requirements.txt

COPY src/ /app/src/
RUN useradd --create-home --shell /usr/sbin/nologin appuser \
    && chown -R appuser:appuser /app

USER appuser
EXPOSE 8000
CMD ["uvicorn", "src.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

(a) What files or directories are copied into `/app`?

(b) Which user runs the application process?

(c) What command runs when the container starts?

(d) What port is the application intended to listen on inside the container?

5. Design the public entry point and internal network layout for your application. Include a reverse proxy or load balancer, at least one private backend service, and at least one private data service. State which ports or protocols should be reachable from the public internet, and explain how the design improves security and scalability.

6. Choose the best storage option for each need: root volume, separate EBS volume, S3/object storage, relational database, or cache.

(a) Operating system files for an EC2 instance:

(b) User account records and permissions:

(c) Uploaded images, videos, or PDFs:

(d) Database files for a single-node stateful service that must survive instance replacement:

(e) Frequently read but recomputable API results:

(f) Compressed backup archives:

7. Suppose your application becomes popular and traffic grows by a factor of 20 during peak periods. Explain one vertical scaling response and one horizontal scaling response. Then choose the better long-term strategy for your application and discuss at least two complications it introduces, such as load balancing, shared state, database replication, caching, queues, or cost control.

8. Design how your application handles state while scaling horizontally. Identify one piece of user or application state, where it should be stored, and why storing it only inside an application server process would make scaling or recovery harder. Include whether your application servers should be stateless or stateful.

9. Design an operations plan for your application. Describe how code would move from a Git repository into production, how you would test or validate deployments, and how you would roll back a bad release. Your answer should name at least four steps in the deployment workflow.

10. Describe an observability plan for your application. Identify at least two logs, two metrics, and one alert you would collect or configure. For each one, explain what failure mode or operational question it helps you detect.

11. Add one advanced cloud capability to your application: a public API, a data pipeline, distributed processing, machine learning training, or machine learning inference. Explain the new components required, how data flows through them, and how the design handles failures or traffic spikes.

12. Security and access control. For each statement, write secure or insecure, then briefly correct any insecure design.

(a) Store AWS access keys directly in a Git repository so deployment scripts can use them.

(b) Allow SSH only from an administrator's IP address using public-key authentication.

(c) Make the database port publicly reachable so developers can connect from anywhere.

(d) Give a backup job permission to write backup objects to one S3 bucket, but not permission to administer EC2 instances.

(e) Run a public web application container as root even though it does not need root privileges.