Jorge Cintron

SNHU

12/17/2024

CS 470 End of Course Final Exam

https://www.youtube.com/watch?v=UnvQKefZvPc

Experiences and Strengths

CS 470 taught me so much about and gave me real-life experience building full-stack web apps on the cloud. It’s directly a product that has contributed to my career ambition to be a capable, professional, and highly competitive developer delivering modern and scalable cloud products.

Skills Learned, Developed, or Mastered

- Full-Stack: I gained knowledge in MEAN (MongoDB, Express.js, Angular, Node.js) stack for building large-scale web applications.
- Integration of AWS Services: Learned how to integrate with AWS services such as API Gateway and Lambda for use in the cloud.
- Containerization: Learning how to use Docker to create and run applications in different environments.
- Tests and Debugging: I improved quickly at testing APIs and troubleshooting during development.
- Technical Documentation and Presentation: Cleaned up on professional technical documentation to represent the projects.

Strengths as a Software Developer

- Logical Problem-Solving: Breaking down big problems into small pieces and using a system.
- Scalability: Develop applications with scalability in mind.
- Flexible: Easy learning and adopting new technologies such as AWS and Docker.
- Working in teams: Working well with teams, especially for cloud-based deployments and integrations.

Positions I believe I am ready:

- Cloud Application Developer
- Full-Stack Developer
- DevOps Engineer (CI/CD pipelines and containerized projects)
- API Developer (Serverless, microservices)

Planning for Growth

I would use microservices and serverless technologies to scale the web app well as usage increases:

- Scale and Error Handling
- Create serverless functions that auto-scale to new requests using AWS Lambda.
- Implement AWS CloudWatch monitoring and logging in real-time to catch and fix issues immediately.

Cost Prediction

The cost of Serverless platforms such as Lambda is affordable on unpredictability because of the execution-time cost. If you have predictable workloads, containerized solutions with ECS or EKS would be the right option, as they guarantee resource allocation.

- Cost Comparison:
- Serverless:
- Advantages: Dependent on usage (per execution) cost; no infrastructure administration.
- Pros: More expensive scale; less time to run.
- Containers:
- Advantages: Predictable prices; suitable for long-term business processes.
- Cons: Installation and administration of infrastructure required.

Elasticity and Pay-for-Service Models

- Elasticity of AWS resources ensures automatic scaling up and down.
- Pay-for-service models, such as Lambda's pay-per-invocation or S3 storage costs, can also achieve cost optimization and better cost forecasting.
- Examples: Auto Scaling Groups for containerized applications and serverless for irregular, event-driven applications.

Expansion Planning's Advantages and Disadvantages:

- Cons: Greater availability, faster response, cost optimization.
- Cons: You must be alert for surprise overcharges in the pay-for-service setup.

Conclusion

Using this course, I learned how to create and deploy scalable cloud applications and now have the technical and strategic knowledge to do so. Growing is part of my plan for growth, and it's about how well I can combine serverless/microservices architectures, control the cost, and achieve more performance (key competencies for today's software).