1. How might resource constraints (like RAM, CPU/GPU availability) impact the performance and usability of locally deployed LLMs?

LLMs demand substantial computational and memory resources to function effectively. They require significant RAM to load model weights and process input-output tensors during inference. When system memory is insufficient, the model may experience drastically slower inference times and fail to load altogether. Similarly, disk speed influences performance, slow hard drives increase model loading times compared to solid-state drives, making real-time applications less feasible. Additionally, LLMs occupy large storage spaces, often exceeding 10–20 GB for mid-sized models like a 7B parameter system in half-precision. These factors make deployment on consumer-grade devices technically challenging and limit usability for applications requiring continuous responsiveness.

GPU availability is another critical determinant of performance. GPUs are optimized for parallel matrix operations essential to transformer-based models, offering massive speedups over CPUs. Without a capable GPU, inference and fine-tuning rely solely on CPUs, which are often 10–100 times slower depending on the task. Limited CPU cores also constrain parallel token generation, reducing throughput, for example, a standard consumer CPU may generate only 1–2 tokens per second, whereas a high-end GPU can exceed 50 tokens per second. Consequently, users facing hardware constraints may need to opt for smaller models (e.g., 3B instead of 70B parameters), sacrificing fluency, reasoning ability, and contextual understanding. These limitations make locally deployed LLMs impractical for real-time applications such as chatbots.

2. What ethical responsibilities do developers have when deploying and customizing local LLMs?

Developers have a duty to ensure fairness, transparency, and accountability when deploying or customizing local LLMs. They must audit training and fine-tuning data to identify and reduce biases, especially those involving gender, race and socioeconomic representation. Techniques such as dataset balancing can help mitigate these issues and prevent models from reinforcing stereotypes. Continuous evaluation is also essential to detect and correct problematic behaviors as the model interacts with real users.

Equally important is maintaining transparency and user trust. Developers should clearly document model architectures, data sources, and modification processes to allow others to assess limitations and potential risks. Protecting user privacy is another ethical priority, ensuring that local inference or logging does not expose sensitive information. Neglecting these responsibilities can lead to misuse, bias propagation, and erosion of public confidence in AI systems. Ethical stewardship means designing local LLMs that prioritize human well-being as much as technical performance.

3. Discuss the potential benefits and limitations of using AI-assisted tools like Roo Code for data analysis tasks, and what aspects of this process feels different from conventional coding projects.

AI-assisted tools such as Roo Code enhance data analysis by automating routine tasks, suggesting efficient code, and identifying errors in real time. They help users translate natural language instructions into executable code, accelerating workflow and improving accessibility for beginners. For experienced analysts, these tools can boost productivity, support rapid prototyping, and maintain consistency in coding practices. The interactive, conversational interface also fosters a more intuitive and exploratory approach to programming compared to traditional manual coding.

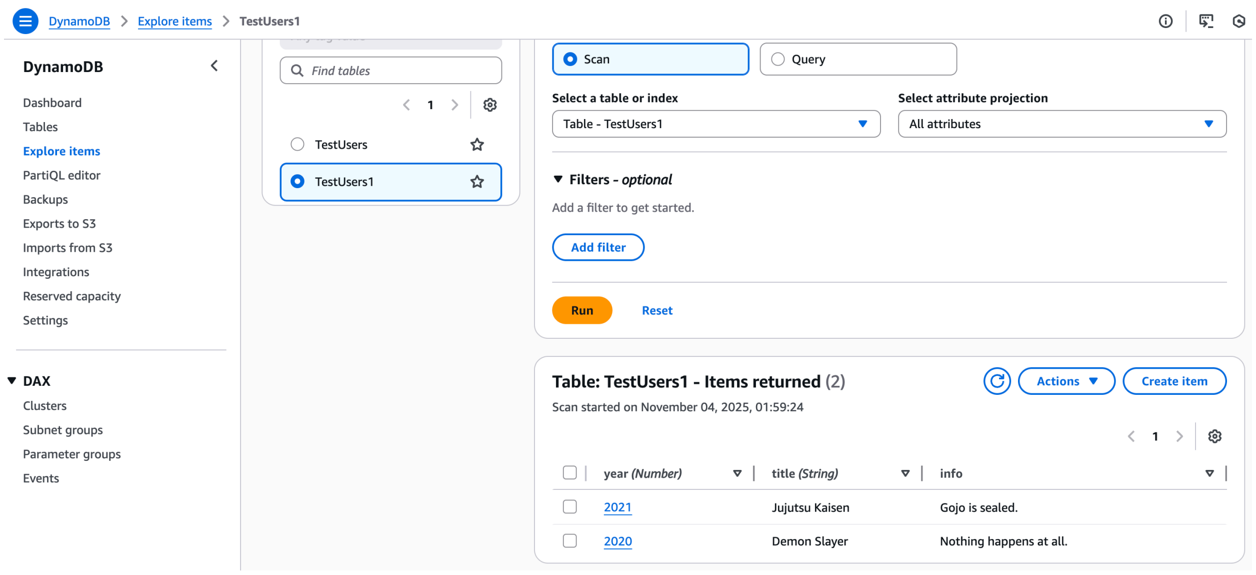
Unlike conventional coding, which emphasizes deliberate planning and debugging, AI-assisted workflows rely more on collaboration with the model, demanding human oversight to verify accuracy and maintain methodological rigor. Thus, while Roo Code enhances efficiency, it must be used critically to ensure analytical validity and reliability.

4. How does virtualisation contribute to the efficiency of cloud computing?

Virtualization allows multiple virtual machines to run on a single physical server by abstracting hardware resources. This enables efficient utilization of CPU, memory, and storage by allocating them dynamically to different users and applications as needed. Cloud providers can scale resources up or down seamlessly, reducing idle capacity and lowering operational costs. It also simplifies deployment and maintenance, allowing rapid provisioning of new environments without physical hardware changes.

It also enhances reliability and security. By isolating workloads in separate VMs, it prevents system failures or security breaches in one environment from affecting others. And supports fast backup and recovery through snapshotting and migration features.

AWS DynamoDB is a fully managed NoSQL database service designed for applications requiring fast and consistent performance at scale. It automatically handles data replication, scaling, and backup, making it ideal for real-time analytics and e-commerce systems. Developers benefit from its low-latency reads and writes, flexible schema design, and seamless integration with other AWS services. However, DynamoDB’s main limitations include higher costs at large scales, limited query flexibility compared to relational databases, and potential complexity when optimizing partition keys for performance.



Select a table or index: Table - TestUsers1

Select attribute projection: All attributes

Filters - *optional*

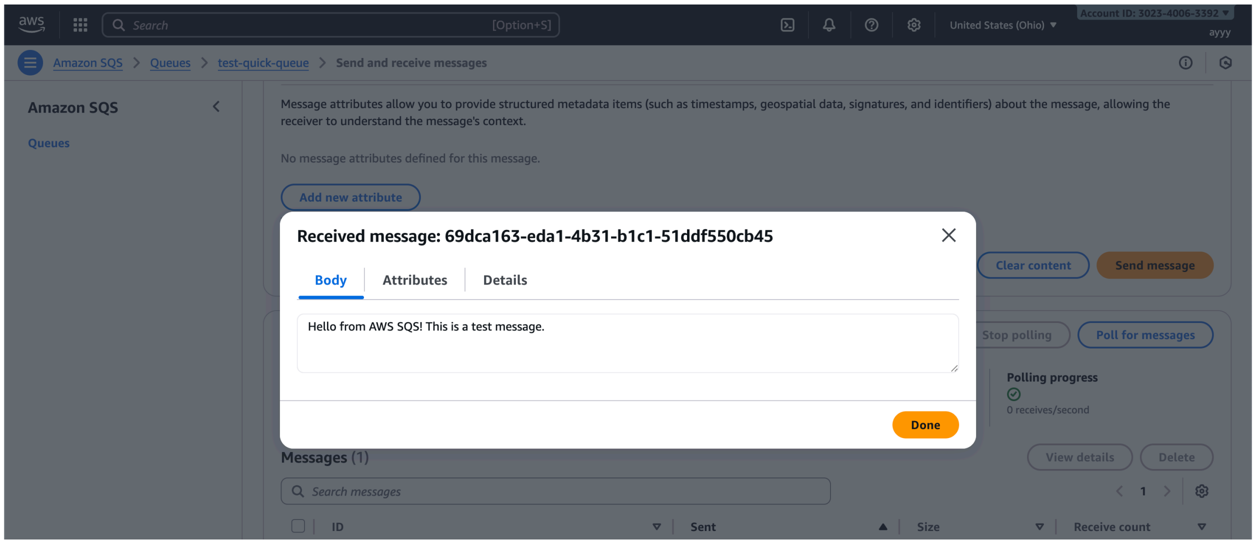
Add a filter to get started.

Table: TestUsers1 - Items returned (2)

Scan started on November 04, 2025, 01:59:24

year *(Number)*	title *(String)*	info
2021	Jujutsu Kaisen	Gojo is sealed.
2020	Demon Slayer	Nothing happens at all.

Amazon Simple Queue Service is a managed message queuing service that enables decoupled communication between distributed systems. It allows asynchronous message exchange, improving reliability and fault tolerance in event-driven architectures. SQS supports both standard and FIFO (First-In-First-Out) queues, ensuring flexibility for various use cases such as background task processing or microservice coordination. Its limitations include eventual consistency in standard queues, potential message duplication, and added latency compared to direct API calls. Despite these trade-offs, SQS remains a powerful tool for building scalable and resilient cloud applications.


aws
Search [Option+S]
United States (Ohio)
Account ID: 3023-4006-3392
ayyy
Amazon SQS > Queues > test-quick-queue > Send and receive messages
Amazon SQS
Queues

Message attributes allow you to provide structured metadata items (such as timestamps, geospatial data, signatures, and identifiers) about the message, allowing the receiver to understand the message's context.

No message attributes defined for this message.

Polling progress

0 receives/second

Received message: 69dca163-eda1-4b31-b1c1-51ddf550cb45

Hello from AWS SQS! This is a test message.