Explain the differences between long polling and short polling in SQS.

Explanation

- **Polling** = The way consumers **check SQS** for new messages.
- There are **two modes**:

1. **Short Polling**
    - The consumer asks SQS → "Do you have a message right now?"
    - If none is available → it immediately returns **empty**.
    - Can lead to **extra API calls** (and cost).
2. **Long Polling**
    - The consumer asks SQS and waits up to **20 seconds** for a message.
    - If a message arrives during that wait, it's delivered.
    - Reduces **empty responses**, saves cost, and improves efficiency.

🔑 Key Differences

Feature	Short Polling	Long Polling
Wait Time	Immediate response	Waits up to 20 sec
API Calls	More (frequent empty responses)	Fewer (waits for messages)
Cost	Higher (more requests billed)	Lower (fewer requests)
Use Case	Less used in prod, only for testing or real-time needs	Best practice for production workloads

---

👨‍💻 DevOps Use Case Example

- **Short Polling:**
  Monitoring script runs every 1 second → Keeps asking SQS "Any message?" → Spams API calls, wastes money.

- **Long Polling (Best Practice):**
  A worker node (EC2, Kubernetes pod, or Lambda) waits up to 20 seconds → If a message arrives anytime during the wait, it's delivered immediately.
  🔷 Example: CI/CD job runners waiting for new build requests.

🔄Diagram

Short Polling (Consumer keeps knocking, many empty responses):

```
Time:   |---1s---|---2s---|---3s---|---4s---|
Calls:  "?"       "?"       "?"       "?"
Reply:  No        No        Yes(MsgA) No
```

Long Polling (Consumer waits and gets message efficiently):

```
Time:   |----------20s----------|
Call:   "?" (waits)
Reply:  Yes(MsgA) arrives after 7s
```

🎯 DevOps Engineer Perspective

- **Always enable long polling** in production (set `WaitTimeSeconds=10-20`).
- Saves cost, reduces unnecessary traffic, and improves message delivery latency.
- Use **short polling** only in rare cases where you need *instant response*, even if it's empty (e.g., testing/debugging).

Short Polling (Default)

- Queries only a **subset of servers** using weighted random distribution
- Returns **immediate response** even if no messages found
- May not return all messages in a single request, but subsequent requests will retrieve remaining messages
- Used when WaitTimeSeconds parameter is set to 0

Long Polling

- Queries **all servers** for messages
- Waits up to **20 seconds maximum** for messages to become available
- Only returns empty response if polling wait time expires
- Activated when WaitTimeSeconds parameter is greater than 0

Long Polling Benefits

- **Reduces empty responses** by waiting for messages to become available
- **Eliminates false empty responses** by checking all servers instead of a subset

- **Returns messages immediately** once they become available
- **Potentially lowers costs** by reducing unnecessary API calls

Choosing Between Options

Consider your application's needs for:

- **Responsiveness** - Short polling provides immediate responses
- **Cost efficiency** - Long polling typically reduces costs by minimizing empty responses

👉 In short:
Short Polling = Fast but wasteful ❌
Long Polling = Efficient & Recommended ✅

What is the maximum wait time for **long polling** in Amazon SQS?
a) 5 seconds
b) 10 seconds
c) 20 seconds
d) 30 seconds
✅ **Answer: c) 20 seconds**

Which statement correctly describes **short polling** in SQS?
a) Waits up to 20 seconds for a message before returning
b) Immediately returns, even if no message is available
c) Always guarantees at least one message is returned
d) Only works with FIFO queues
✅ **Answer: b) Immediately returns, even if no message is available**

From a **cost perspective**, why is **long polling** preferred over short polling in production?
a) It delivers messages faster
b) It reduces the number of empty responses, leading to fewer API calls
c) It guarantees message ordering
d) It is free of charge
✅ **Answer: b) It reduces the number of empty responses, leading to fewer API calls**

Which is the **best practice** for production SQS consumers?
a) Use short polling for real-time efficiency
b) Always set WaitTimeSeconds = 10–20 for long polling
c) Combine short polling with DLQ
d) Use visibility timeout instead of long polling
✅ **Answer: b) Always set WaitTimeSeconds = 10–20 for long polling**