Can you walk me through a time when your CloudFormation stack failed during deployment? How did you approach debugging and resolving the issue?

STAR Answer:

Yes, this is something I've encountered multiple times in real-world deployments. One particular time, I was deploying a new infrastructure stack for a development environment using AWS CloudFormation, and the stack creation failed midway.

My goal was to identify the **root cause of the failure** quickly, resolve it, and ensure that the stack could be **successfully re-deployed**. I also had to make **sure no orphaned resources** were left behind to avoid **unnecessary billing.**

I follow a structured approach whenever stack creation fails:

1. **Start with CloudFormation Events tab:**
   I check the **chronological log of events** to find the exact resource that failed and the error message. In this case, I found:

makefile

Resource: MyEC2Instance
Status: CREATE_FAILED
Reason: Instance type t2.micro not supported in us-east-1a

2. **Categorize the error:**
   I classify errors into categories:
   - **Permission issues** (e.g., Access Denied)
   - **Resource limits** (e.g., Quota exceeded)
   - **Configuration errors** (e.g., Invalid value)
   - **Dependency errors** (e.g., missing Security Group)

This helps me focus on the exact root cause quickly.

3. **Fix and redeploy:**
   In this case, I updated the template to use a supported AZ (us-east-1b instead of us-east-1a), deleted the failed stack which was in ROLLBACK_COMPLETE state, and re-deployed successfully.
4. **Advanced option:**
   For more complex stacks, I sometimes use --disable-rollback during stack creation to preserve the resources and inspect them if needed. This helps especially when debugging intricate dependency chains.
5. **In rare cases:**
   If I encounter ROLLBACK_FAILED, I manually delete the resource that caused the rollback to fail, then use:

aws cloudformation continue-update-rollback --stack-name my-stack
to resume and clean up properly.

By following this structured approach, I was able to resolve the issue quickly, re-deploy the stack, and also document the failure mode for future reference. Over time, this method has helped me minimize downtime and improve team confidence during deployments.

Where should you look first when a CloudFormation stack creation fails?
A. EC2 Dashboard
B. CloudTrail Logs
C. CloudFormation Events tab
D. S3 Bucket
Answer: C

What is the default behavior when a CloudFormation stack creation fails?
A. Partial resources are retained for debugging
B. All AWS resources are preserved
C. A rollback is triggered to delete created resources
D. The stack is moved to ARCHIVED state
Answer: C