

Scenario Based Questions

Scenario 1: Suddenly, API latency increases and customers see slow responses. What do you do?

Answer:

I start by checking:

1. CloudWatch / Instana latency graphs → confirm timing of spike
2. Load balancer 4xx/5xx metrics
3. Pod health in EKS (kubectl describe pod)
4. Database metrics (connections, slow queries, CPU)
5. Autoscaling (HPA triggering or not)
6. Network issues (ALB target failures, DNS delays)
7. Fix by:
 - Scaling pods or nodes
 - Adding caching via Redis/ElastiCache
 - Reducing DB bottlenecks
 - Restarting unhealthy pods
 - Increasing rate limits or capacity

Scenario 2: Your EKS cluster nodes are in NotReady state. What's your approach?

Answer:

- Check node logs using kubectl describe node
- Verify kubelet status
- Check EC2 instance health (CPU/memory/disk)
- Verify IAM role permissions
- Check VPC networking (ENI issues)
- Confirm cluster autoscaler events
- Restart node / detach & reattach ENI
- If necessary, replace the node

Scenario 3: Certificate expired in production and service is down. What do you do?

Answer:

1. Immediately issue new certificate
2. Apply new cert to ALB/CloudFront/EC2
3. Invalidate CloudFront cache if required
4. Validate HTTPS endpoint
5. Post-fix:
 - Add automation for auto-renew (ACM, Bash script)
 - Update certificate trackers
 - Add monitoring alert before expiry

Scenario 5: DynamoDB latency spike across multiple regions.

Answer:

- Check read/write capacity
- Check throttling metrics
- Verify cross-region replication lag
- Tune performance by increasing RCUs/WCUs
- Enable on-demand mode if needed
- Add caching using ElastiCache

Scenario 4: Deployment failed in production. What steps do you take?

Answer:

- Check Harness pipeline logs
- Verify if the image is correct
- Check K8s events for pod failures
- Check DB migrations
- Roll back deployment
- Validate rollback
- Create RCA and add steps to prevent recurrence

Scenario 6: Suddenly CPU spikes to 100% on a production EC2 instance. What do you do?

Answer:

1. Check CPU-intensive processes using top, htop
2. Identify app-specific issues (memory leak / loops)
3. Check application logs
4. Verify network connections (netstat -tulpn)
5. Check disk I/O (iostat)
6. If immediate fix needed:
 - Restart specific service
 - Scale instance vertically or add instances behind ALB
7. Long-term fix:
 - Tune app
 - Update AMI
 - Add autoscaling

Follow-up:

- How do you identify a memory leak?
- How do you reduce EC2 cost after scaling?

Scenario 7: EC2 instance is not reachable via SSH. What's your approach?

Answer:

- Check EC2 reachability checks
- Verify SG → port 22 allowed?
- Verify NACL → outbound traffic allowed?
- Check route table → correct routes?
- Check CPU/disk saturation
- Use EC2 serial console & system logs
- If corrupted file system →
Attach root volume to another instance → fix → reattach

Follow-up:

- What if SSH key is lost?

Scenario 8: NAT Gateway charges are extremely high. What do you check?

Answer:

- Identify heavy traffic using VPC flow logs
- Check private subnet systems pulling updates
- Move S3/API calls to **VPC endpoints**
- Use **two NAT GWs**, region-based
- Use caching
- Tune periodic cron jobs

Scenario 9: Your pod keeps restarting every 5 seconds – CrashLoopBackOff. Steps?

Answer:

- Get logs → kubectl logs
- Check entrypoint errors, env vars
- Check readiness/liveness probes
- Validate ConfigMap/Secret
- Check resource limits
- Validate image tag
- Check persistent volume mount failure

Follow-up:

- How do you detect OOMKilled?
- How to fix imagePullBackOff?

Scenario 10: EKS worker nodes are NotReady. What do you check?

Answer:

- Kubelet logs
- ENI attachment issues
- Disk full
- CoreDNS failing
- CNI plugin failure (AWS VPC CNI)
- IAM role missing required permissions
- Reboot node or replace node

Scenario 11: How do you perform a zero-downtime upgrade in EKS?**Answer:**

1. Upgrade control plane
2. Upgrade node groups
3. Drain node-by-node
4. Ensure PDB (Pod Disruption Budget)
5. Validate deployments
6. Run smoke tests

Scenario 11: Pods are stuck in Pending. What's the reason?**Answer:**

Most common:

- No available nodes / insufficient CPU/Memory
- Other reasons:
- Node affinity issues
 - Taints on nodes
 - Pod requesting GPU nodes
 - PVC unbound
 - CNI issues

Follow-up:

- How do you diagnose CNI failures?
-

Scenario 12: Your S3 bucket suddenly shows huge cost increase. How do you analyze?**Answer:**

- Check top objects using S3 analytics
- Check lifecycle policies
- Identify PUT/GET anomaly with CloudWatch
- Check cross-region replication loops
- Check data transfer costs
- Identify if logs/backup scripts misconfigured

Scenario 13: API Gateway shows 5xx errors. What's your debugging process?

Answer:

- Check Lambda logs
 - Check integration (Lambda / ALB / Backend)
 - Check throttling
 - Check cold starts
 - Check API key usage
 - Check WAF blocking
-

Scenario 14: DynamoDB table is throttling. What do you do?

Answer:

- Check RCUs/WCUs usage
- Enable auto-scaling
- Add DAX caching
- Tune partition keys
- Move to on-demand mode temporarily
- Check hot partitions with CloudWatch

Scenario 15: Production deployment failed during pipeline execution. What are immediate steps?

Answer:

- Check pipeline logs
- Identify which stage failed
- Validate config file changes
- Check image build logs
- Check deployment failure logs on K8s
- Trigger rollback
- Validate rollback success
- Generate RCA

Scenario 16: Rollback is taking too long – what do you do?

Answer:

- Switch to blue environment (Blue-Green)
 - Manually redeploy last stable image
 - Restore config from version control
 - Verify application health
-

Scenario 17: Artifacts are not syncing in JFrog/Artifactory. What do you check?

Answer:

- Repo permissions
- Network connectivity
- Quota size
- Corrupted metadata
- Token/authentication issues

Scenario 18: Latency spikes seen in production API. Steps?**Answer:**

1. Check ALB/NLB target health
 2. Check pod/node load
 3. Check DB latency
 4. Check upstream service
 5. Check throttling
 6. Check logs & traces
 7. Scale pods
 8. Add caching layer
 9. Update auto-scaling policies
-

Scenario 19: Your team is repeatedly hitting SLA breaches. What's your SRE approach?**Answer:**

- Define SLO + Error Budget
 - Add monitoring + alerting improvements
 - Analyze top incidents (Pareto analysis)
 - Increase automation
 - Improve runbooks & playbooks
 - Perform RCA and fix RCA patterns
-

Scenario 20: How do you debug high 5xx errors?**Answer:**

- Check ALB target health
 - Check application logs
 - Check resource exhaustion
 - Check DB connection exhaustion
 - Validate code version
 - Rollback deployment
-

Scenario 21: Expired certificate caused downtime. What preventive strategy do you implement?**Answer:**

- ACM automation
 - CRON-based renewal scripts for non-ACM certs
 - Dashboard + alerts 30/15/7 days before expiry
 - Central certificate manager
 - Golden process document for renewal
-

Scenario 22: IAM user got unnecessary admin access. What do you do?

Answer:

- Immediately revoke the policy
- Check CloudTrail logs
- Enable MFA
- Apply least privilege model
- Use roles instead of users

Scenario 23: RDS storage is hitting 90% usage. Steps to fix?

Answer:

- Increase storage size
 - Enable autoscaling storage
 - Optimize queries
 - Delete old logs
 - Move cold data to S3
 - Check connection leaks
-

Scenario 24: S3 replication is failing. What do you check?

Answer:

- IAM permissions
 - Versioning on both buckets
 - Cross-Region replication config
 - KMS key access
 - Replication status logs
-

Scenario 2: High DNS resolution time. What do you check?

Answer:

- Check Route53 health checks
 - Check DNS TTL values
 - Check EKS CoreDNS pod health
 - Check network latency
 - Check CNI plugin
 - Check firewall rules
-

Scenario 26: Application unreachable after ALB creation. What do you check?

Answer:

- Target group health check port/path
- Security group (inbound/outbound)
- NACL
- Listener rules
- Subnet mappings
- ASG health

Scenario 27: Azure VM keeps freezing. What do you troubleshoot?

Answer:

- CPU credits
- Disk latency
- Boot diagnostics logs
- Azure Monitor metrics
- Scaling issues
- Network bandwidth

Scenario 28: You are leading a P1 outage – what's your first step?

Answer:

- Acknowledge alert
- Notify stakeholders
- Quick triage: Is it app / DB / network / infra?
- Assign roles to responders
- Start live log analysis
- Mitigate impact (scale, restart, failover)
- RCA after restoration

Scenario 29: How do you prevent repeating incidents?

Answer:

- Add automation
- Monitor root causes
- Add alert thresholds
- Improve code or infra stability
- Strengthen deployment pipelines

1. Scenario: Your EC2 instance becomes unreachable but system checks pass. What now?

Answer:

- Check route table associations
 - Check NACL blocking ephemeral ports
 - Security Group misconfig
 - ENI detached
 - Application firewall inside instance blocking SSH
- Fix: Reattach ENI / modify NACL / restart network daemon.

2. Scenario: S3 bucket suddenly shows 10x increase in PUT requests. What do you check?

Answer:

- Check CloudTrail → identify caller
- Check application for retry storms
- Look for misconfigured cron jobs
- Buggy SDK retry logic

- Malware script accidentally uploading files

3. Scenario: NAT Gateway costs are suddenly ₹2 lakh/month. What is root cause?

Answer:

- Traffic going through NAT instead of VPC endpoints
 - ECS/EKS pulling container images excessively
 - Endless cron jobs downloading packages
- Fix: Add S3, ECR, SNS, SQS VPC endpoints and reduce NAT traffic.

4. Scenario: Application performance degrades after enabling encryption on EBS.

Why?

Answer:

- CPU overhead from encryption
 - Old instance family with slow AES support
- Fix: Move to C6i/M6i/R6i with new Intel/AMD chips.

5. Scenario: RDS latency increases during peak traffic. What steps?

Answer:

- Increase instance class
- Add read replicas
- Optimize queries
- Check max_connections
- Look for table locks
- Scale vertically + horizontally

6. Scenario: RDS storage increases rapidly. What do you inspect?

Answer:

- Unused binary logs
 - Slow query logs
 - Large transactions
 - Uncleaned temp tables
- Fix: enable auto-purge logs & use monitoring.

7. Scenario: DynamoDB Global Table shows replication lag. Why?

Answer:

- Write spikes
 - Hot partitions
 - Cross-region throttling
- Fix: Add WCU, enable adaptive capacity, shard partitions.

8. Scenario: Lambda hitting timeout. What do you check?

Answer:

- VPC ENI cold start
- Slow DNS resolution

- Slow DB calls
Fix: Increase timeout, optimize code, use provisioned concurrency.
-

9. Scenario: CloudFront returning 504 errors. Root cause?

Answer:

- Origin timeout
 - Origin overloaded
 - Long-running backend operations
- Fix: Increase origin timeout / scale backend.
-

10. Scenario: ELB deregistering healthy targets. Why?

Answer:

- Health check path wrong
 - App takes too long to respond
 - SG does not allow health check port
- Fix: Update HC path, increase timeout, fix SG.
-

★ Kubernetes / EKS Advanced Scenarios

11. Scenario: Pod is running but application inside is dead. How do you detect?

Answer:

Readiness probe fails → but liveness probe succeeds.
Fix: Proper readiness probe config.

12. Scenario: Kube-proxy high CPU usage across nodes. Why?

Answer:

- Too many services/endpoints
 - IPTables rules explosion
- Fix: Move to IPVS mode.
-

13. Scenario: CoreDNS pods failing with “SERVFAIL”.

Answer:

- Upstream DNS misconfigured
 - Worker node cannot reach DNS servers
- Fix: Update Corefile, check VPC DNS settings.
-

14. Scenario: Kubernetes API server becoming slow. Why?

Answer:

- Too many CRDs
 - Large number of events
 - Excess controller loops
- Fix: Cleanup CRDs, reduce events retention.
-

15. Scenario: PVC bind delays in EKS. Why?

Answer:

- Wrong StorageClass
 - AZ mismatch (PV and pod)
- Fix: Use multi-AZ storage class or correct zone.
-

16. Scenario: CrashLoopBackOff keeps happening though code is fine.

Answer:

- Wrong file permissions
 - Missing environment variables
 - Wrong image entrypoint
- Fix: Add correct permissions & env variables.
-

17. Scenario: Pod keeps restarting due to OOMKilled even with increased memory.

Answer:

- Memory leak in app
 - Large JVM heap
- Fix: Tune app/Garbage Collection, add memory limits.
-

18. Scenario: Node NotReady state due to disk pressure. Why?

Answer:

- Logs inside /var/log filled
 - Container logs not rotated
- Fix: Use log rotation, increase node volume.
-

19. Scenario: EKS cluster autoscaler adds nodes but pods still Pending.

Answer:

- Incorrect instance type
 - Taints on new nodes
 - Pod affinity not matching
- Fix: Remove taints, update nodeSelector.
-

20. Scenario: Application downtime during rolling update. Why?

Answer:

- Readiness probe too fast
 - maxUnavailable too high
- Fix: Update rolling strategy & readiness delay.
-

★ **CI/CD, Terraform, Automation Scenarios**

21. Scenario: Terraform state is lost. Recovery plan?

Answer:

- Recover S3 version history

- Recreate local state using terraform import
 - Use DynamoDB lock table
-

22. Scenario: Terraform plan shows large unwanted changes. Why?

Answer:

- Drift
 - Missing remote state
 - Updated provider versions
- Fix: Re-run init, refresh state, pin provider versions.
-

23. Scenario: Pipeline runs but deployment doesn't reach cluster.

Answer:

- Bad kubeconfig
 - Wrong namespace
 - Ingress not updated
- Fix: Verify cluster context & permissions.
-

24. Scenario: Helm upgrade failed midway—cluster unstable.

Answer:

- Incomplete manifests
- Fix: helm rollback, clean orphaned resources.
-

25. Scenario: You accidentally applied Terraform to production environment.

Answer:

- Immediately stop pipeline
 - Review drift changes
 - Rollback via last known state
 - Add mandatory approvals for prod
-

26. Scenario: Docker image takes 10 minutes to build. How to optimize?

Answer:

- Use multistage builds
 - Reduce layers
 - Cache dependencies
 - Use distroless/alpine
 - Pre-build base images
-

27. Scenario: Docker containers frequently restarting in Jenkins agent. Why?

Answer:

- Low disk
 - Jenkins workspace cleanup
 - OOM kills
- Fix: Increase workspace, allocate more memory.
-

28. Scenario: Secrets found in Git repo. What is your action plan?

Answer:

- Immediately rotate credentials
 - Use Git filter-branch to remove from history
 - Move secrets to Secrets Manager
 - Enforce pre-commit hooks
-

29. Scenario: Deployment works on staging but fails in production.

Answer:

- Prod has different config
 - Different secrets
 - Different IAM roles
- Fix: Ensure env parity; add config versioning.
-

30. Scenario: Unexpected auto-reverts in CI/CD pipelines.

Answer:

- Auto rollback policies triggered
 - Health check failures
- Fix: Tune health checks & reduce strictness.
-

★ SRE / Performance / Monitoring Scenario Answers

31. Scenario: 5xx errors increase but CPU/memory normal.

Answer:

- DB connection pool exhaustion
 - Downstream timeout
 - Thread blocking
 - ALB routing misconfigured
- Fix: Increase DB pool, increase ALB timeout.
-

32. Scenario: Frequent latency spikes at same time daily.

Answer:

- Cron jobs
 - Backup processes
 - DB vacuum / snapshot
- Fix: Reschedule heavy operations.
-

33. Scenario: Memory leak in production identified. Next steps?

Answer:

- Heap dump analysis
- Restart pods with liveness probe
- Fix code
- Add HPA based on memory metrics

34. Scenario: DB connection spikes across microservices.

Answer:

- Missing connection pooling
 - Long-running queries
 - Too many replicas
- Fix: Increase pool, fix slow queries.
-

35. Scenario: Disk IOPS exhaustion on EBS.

Answer:

- Move to io2
 - Add EBS-optimized instances
 - Add caching layer
-

36. Scenario: Error budget exceeded. What now?

Answer:

- Freeze production releases
 - Focus on stability
 - Reduce changes until SLO improves
-

37. Scenario: Client sees slowness but server says 100% health.

Answer:

- CDN edge issue
 - Network latency
 - DNS latency
- Fix: Run traceroute, check Edge locations.
-

38. Scenario: Kubernetes network jitter increasing.

Answer:

- CNI plugin problem
 - Overloaded nodes
 - Too many iptables rules
- Fix: Move to IPVS, increase node size.
-

39. Scenario: Logs suddenly increase 20x. Why?

Answer:

- Debug mode enabled
 - Infinite loops
 - Logging library update
- Fix: Disable debug, rotate logs.
-

40. Scenario: Same incident repeats despite fixes. How to enforce permanent fix?

Answer:

- Document RCA

- Add proactive monitoring
- Add automation
- Add regression tests