1. Structured Troubleshooting Methodology

- **Identify the Problem:**
  • Gather evidence (user reports, alerts, logs)
  • Example: App cannot connect to its database even though the VM is running
- **Isolate the Cause:**
  • Check network connectivity (security groups, ACLs, routing)
  • Use monitoring dashboards to narrow down issues
- **Develop & Test a Hypothesis:**
  • Formulate potential causes (e.g., misconfigured firewall, routing issues)
  • Run tests such as network tracing or load simulations
- **Plan & Implement the Solution:**
  • Outline action steps, consider risks, and implement changes
- **Verify and Document:**
  • Confirm the fix (e.g., connection established, performance improved)
  • Document steps and lessons learned

2. Monitoring Cloud Environments

- **Performance & Connectivity:**

Database Connectivity Issues:
• Check if the database's IP is reachable; investigate security groups and ACL settings

Website Performance:
• Contact the hosting provider; review monitoring alerts for resource bottlenecks or traffic spikes

- **Security Incident Detection:**

Suspicious Email/Spam Breach:
• Collect evidence (email headers, sender info)
• Examine firewall logs and network traffic for unusual patterns
Brute Force Attacks:
• Look for high numbers of failed logins from unfamiliar IP addresses in logs

3. AWS CloudTrail for Monitoring and Security

- **Purpose & Usage:**
  • Logs detailed API calls across your AWS account (who, what, when, and from where)
  • Critical fields: **sourceIPAddress**, **eventName**, **eventTime**
- **Key Use Cases:**

Unauthorized API Calls:
• Use the **sourceIPAddress** field to trace the origin of an unauthorized deletion or modification

Audit Trail Details:
• CloudTrail provides a granular record of actions (ideal for forensic investigations)
Log Integrity:
• Enable log file validation to ensure logs stored in S3 haven't been tampered with

- **Data vs. Management Events:**

By default, CloudTrail logs management events
To capture data events (e.g., S3 GetObject/PutObject), you must explicitly enable them

- **Real-Time Alerts:**

CloudWatch Integration:
• Send CloudTrail events to CloudWatch Logs, create metric filters, and set up alarms for anomalous activity

- **Long-Term Analysis with CloudTrail Lake:**

Aggregates and stores events for extended periods (up to 10 years)
Useful for compliance audits and deep security investigations

4. AWS IAM & Policy Concepts

- **Policy Structure and Key Elements:**

Effect:
• Specifies Allow or Deny (explicit deny always overrides allow)
Principal:
• Specifies which user, role, or service the policy applies to
Action & Resource:
• Defines what operations can be performed on which resources
• Include proper ARNs (use wildcards like `/*` for object-level permissions in S3
Condition:
• Adds context (e.g., IP address, time, or dynamic values like *"Condition" : { "StringEquals" : { "aws:username" : "ana" }}* for user-specific folder access

- **Policy Evaluation:**

Explicit denies always override allows
When conflicting policies exist, the most restrictive rule applies

Identity-based vs. Resource-based Policies

- Identity-based policies:

These are **attached to IAM identities** (users, groups, or roles).
They don't have a Principal element.
The act of attaching them to an identity is what makes them identity-based.

- Resource-based policies:

These are **attached directly to resources** (e.g., S3 buckets, SQS queues).
They always include a Principal element.
The act of attaching them to a resource is what makes them resource-based.

5. Troubleshooting IAM Policies and Roles

- **Common Issues:**

Access Denied Errors:
• Check for typos in ARNs and misconfigured policies
Conflicting Policies:
• Understand that resource-based policies (which may grant read access even after an identity-based policy is removed) and explicit denies beats allows

- **IAM Role Assumptions and Trust Policies:**

Cross-Account Access:
• Create roles in the destination account with a trust policy that includes the source account
Service-Linked Roles:
• Ensure the trusted service is correctly listed in the role's trust policy
• Service-linked roles are IAM roles automatically created and managed by AWS services to perform actions on your behalf.

- **Using Tools for Validation and Simulation:**

IAM Policy Simulator:
• Test and troubleshoot policy effects without impacting production
IAM Access Analyzer:
• Identify unintended resource sharing and unused permissions
CloudTrail Logs:
• Audit and troubleshoot actions taken by IAM users/roles

6. Best Practices and Additional Tips

- **Least Privilege:**

Grant only the permissions needed for tasks.

- **Regular Reviews and Versioning:**

Periodically audit and update policies; maintain version history for rollbacks.

- **Separation of Duties:**

Divide responsibilities between roles to minimize risk.

- **Dynamic Policies:**

Use IAM policy variables to tailor permissions per user (e.g., ${aws:username}).

- **Secure Log Access:**

Restrict access to CloudTrail logs using IAM policies to ensure confidentiality.