

◆ Module Overview

The module focuses on automating deployments using AWS services, particularly **AWS CloudFormation**. It explores tools, best practices, and strategies to make deployments scalable, repeatable, and easier to manage.

◆ Learning Objectives

By the end of this module, you should be able to:

- Identify AWS services for configuration management
 - Understand and resolve deployment challenges
 - Use Infrastructure as Code (IaC)
 - Use and troubleshoot AWS CloudFormation
 - Apply CI/CD techniques for automated deployments
-

◆ Configuration Management in the Cloud

Benefits:

- Increased efficiency and automation
- Validated and documented changes
- Cost control by removing unnecessary resources
- Consistent security enforcement

Tools:

- AWS CloudWatch
- AWS CloudFormation
- AWS OpsWorks

Technologies:

- EC2 User Data
 - AMIs
 - Configuration frameworks (e.g., Chef, Puppet)
-

◆ AMI Strategy

Creating Custom AMIs:

- Use to pre-install software and configurations
- Consider different levels: fully baked, JeOS, or dynamic boot-time configuration
- Copy AMIs across regions if needed

Commands:

```
bash
CopyEdit
aws ec2 create-image
aws ec2 copy-image
aws ec2 register-image
```

◆ EC2 Launch Templates

Used to define standard instance configurations (AMI ID, instance type, key pair, etc.), making launches consistent and less error-prone.

◆ Infrastructure as Code (IaC)

IaC automates the resource lifecycle:

1. Provisioning (CloudFormation)
 2. Configuration (SSM, OpsWorks)
 3. Monitoring (CloudWatch)
 4. Governance (AWS Config)
 5. Optimization (Trusted Advisor)
-

◆ JSON vs YAML

- JSON is lightweight and common in APIs
- YAML is more readable and often used in config files

Both can be used for CloudFormation templates.

◆ AWS CloudFormation Deep Dive

Concepts:

- **Template:** Blueprint in JSON/YAML
- **Stack:** Set of AWS resources created from a template

- **Change Set:** Preview of changes before applying

Template Components:

- Parameters
- Mappings
- Resources
- Outputs
- Metadata (`CloudFormation::Init`)
- WaitConditions (used to delay stack completion until a signal is received)

Tools:

- Visual Studio, Eclipse, AWS CloudFormation Designer

CLI Usage:

```
bash
CopyEdit
aws cloudformation create-stack
aws cloudformation continue-update-rollback
```

◆ Best Practices

- Use modular, reusable templates
 - Apply version control
 - Validate templates before deployment
 - Use rollback protections and test updates in dev/staging
-

◆ Troubleshooting CloudFormation

Common Issues:

- S3 template URL errors
- Insufficient permissions
- Failed WaitConditions
- `cloud-init.log`, `cfn-init.log`, and `cfn-wire.log` for diagnostics

Use `--on-failure DO_NOTHING` to prevent automatic rollback for easier debugging.

◆ CI/CD on AWS

Integrates continuous integration (code/test/build) and delivery/deployment pipelines.

AWS Services:

- **CodePipeline:** Automates end-to-end release
- **CodeBuild, CodeDeploy**, etc.

Challenges Without CI/CD:

- Manual errors
- Inconsistent deployments
- Slow releases

◆ Scenario & Hands-On Activities

Scenario: Global travel company restructures IT for efficiency. Engineers must use `WaitCondition` and `CreationPolicy` during resource bootstrapping and stack coordination.

Labs & Projects:

- Automate deployments using `CloudFormation`
- Troubleshoot deployments (e.g., failed stacks, `WaitConditions`)
- Translate business needs into IaC solutions

🏠 Summary

- Automate deployments using AWS services
- Use IaC for predictable infrastructure management
- Apply CI/CD for faster, safer releases
- Leverage best practices and troubleshooting techniques for `CloudFormation`

1. When using AWS `CloudFormation`, what is the primary function of the `WaitCondition` resource?

- A. It adds a delay to allow manual approvals
- B. It signals the completion of a user-defined task before stack completion
- C. It prevents stack deletion
- D. It sends notifications on template deployment

Correct Answer: B

Explanation: `WaitCondition` pauses stack creation until it receives a success signal or a timeout occurs.

2. Which two resources should be monitored using log files like `cfn-init.log` and `cfn-wire.log` when CloudFormation deployments fail?

- A. EC2 Spot Instances
- B. AWS Lambda functions
- C. User data scripts and `CloudFormation::Init`
- D. RDS Instances

Correct Answer: C

Explanation: These logs help diagnose issues related to bootstrapping and initialization on EC2.

3. Which of the following CloudFormation sections is responsible for creating actual AWS resources?

- A. `Parameters`
- B. `Resources`
- C. `Mappings`
- D. `Outputs`

Correct Answer: B

Explanation: The `Resources` section declares all the AWS resources that should be created or updated.

4. In which two scenarios should `waitCondition` and `CreationPolicy` be used together?

- A. Creating Lambda functions and sending emails
- B. Automatically scaling resources and configuring EC2 instances
- C. Coordinating resource creation and tracking configuration status
- D. Provisioning DynamoDB tables and CloudTrail logs

Correct Answer: C

Explanation: `WaitCondition` and `CreationPolicy` help track EC2 configuration and coordinate stack creation.

5. What happens if a CloudFormation stack creation fails and rollback is disabled using `--on-failure DO_NOTHING`?

- A. All resources are deleted immediately
- B. Only successful resources remain
- C. Resources are retained, allowing for troubleshooting
- D. Stack is frozen and can't be edited

Correct Answer: C

Explanation: `DO_NOTHING` retains the state for inspection and manual correction.

6. What does the `Ref` intrinsic function do in a CloudFormation template?

- A. Defines resource dependencies
- B. Selects elements from a list
- C. Provides a reference to a parameter or resource
- D. Calculates mathematical values

Correct Answer: C

Explanation: `Ref` is used to retrieve the value of a parameter or resource.

7. Which AWS service enables metric-based monitoring for infrastructure and triggers alerts or automation?

- A. AWS Trusted Advisor
- B. AWS Config
- C. Amazon CloudWatch
- D. AWS CodeDeploy

Correct Answer: C

Explanation: Amazon CloudWatch collects and analyzes operational metrics for AWS resources.

8. Which section in a CloudFormation template provides static lookup values like region-to-AMI mappings?

- A. Conditions
- B. Resources
- C. Mappings
- D. Outputs

Correct Answer: C

Explanation: The `Mappings` section holds static name-value pairs often used to retrieve region-specific values.

9. What is a key advantage of using `CloudFormation::Init` over simple user data scripts?

- A. Allows CloudFormation to skip rebooting
- B. Enables rollback on failure and metadata-driven configuration
- C. Supports nested stacks
- D. Executes independently from the template lifecycle

Correct Answer: B

Explanation: `CloudFormation::Init` works with `cfn-init` and provides rollback capability and metadata handling.

10. Which AWS CLI command continues a rollback after a failed update?

- A. `aws cloudformation rollback-stack`
- B. `aws cloudformation fix-stack`
- C. `aws cloudformation continue-update-rollback`
- D. `aws cloudformation recover-stack`

Correct Answer: C

Explanation: This command resumes a rollback after a stack update failure enters `UPDATE_ROLLBACK_FAILED` state.

Module Objectives

At the end of this module, learners will be able to:

- Explain the **purpose and function of tagging** in AWS
- Understand **cost management strategies** related to tagging

- Enforce tagging via **IAM policies**
 - Identify **cloud cost benefits**
 - Understand the role of **AWS Trusted Advisor**
 - Apply **tagging strategies** to manage resources
-

◆ Core Topics Covered

1. Tagging in AWS

- Tags are **key-value pairs** attached to resources for identification, organization, and automation.
- Example:
 - Name = WebServer
 - Environment = Production
- Benefits:
 - Easier resource tracking
 - Cost allocation
 - Automation of tasks (e.g., shutting down dev resources on weekends)
- **Common tags** include:
 - Environment, Application, Owner, Department, Cost Center, Purpose, Stack

2. Enforcing Tagging

- Use **AWS Config** to require tags via the `required-tags` managed rule.
- **AWS CLI** and scripts can create, query, and filter resources using tags.
- IAM policies can enforce tagging with conditions like:

```
json
CopyEdit
"Condition": {
  "StringEquals": {
    "aws:RequestTag/costcenter": "115"
  }
}
```

3. Tagging Use Cases

- Start/stop resources based on tags
- Enforce compliance (e.g., terminate non-compliant instances via automation scripts like Conformity Monkey)
- Example pseudocode to terminate untagged instances:

```
bash
CopyEdit
if !instance.tags.member_of("Required_Tag")
```



```
then aws ec2 terminate-instance (instance)
```

4. Cost Management Tools

- **Cost Benefits of AWS Cloud:**
 - Pay-as-you-go model
 - Schedule-based shut down of dev/test environments
 - Rightsizing and decommissioning reduce waste

AWS Services:

- **AWS Budgets:** Track usage and forecast costs
 - **AWS Cost Explorer:** Visualize spending trends; export to CSV
 - **AWS Cost and Usage Reports:** Detailed billing data
 - **Amazon CloudWatch Billing Alarms:**
 - Trigger alerts when spending thresholds are reached
 - Use SNS for notifications
-

5. Designing for Cost Optimization

- **Turn off unused resources**
 - Use **serverless scripts** (Stopinator) to automate start/stop of instances
 - Consider:
 - Using smaller instance types
 - Removing unnecessary services (e.g., old databases)
-

6. AWS Trusted Advisor

- A tool offering real-time guidance on:
 - **Cost optimization**
 - **Security**
 - **Performance**
 - **Fault tolerance**
 - **Service limits**

Cost Optimization Checks Include:

- Idle EC2/RDS instances
- Unused Elastic IPs
- Underutilized load balancers

- Core checks are available to all AWS accounts; more advanced features require Business/Enterprise support
-

7. Real-World Scenario & Lab

Scenario: A non-profit with several AWS accounts seeks to reduce costs and consolidate billing for transparency to donors. They face issues with large **Service Control Policies (SCPs)**.

 **SCP Document Size Limit: 5,120 bytes**

Lab 7: Managing Resources with Tagging

- Tag existing resources
- Query and filter by tags
- Use scripts to stop/terminate resources based on tags

Project Example: Mom & Pop Café & Bakery

- Optimization task:
 - Remove legacy database
 - Downsize EC2 instance from `t2.small` to `t2.micro`
 - Estimate cost savings using AWS pricing tools
-

Key Takeaways

- **Tags** are critical for managing, automating, and optimizing AWS environments.
 - **Enforcing tags** through IAM and AWS Config ensures consistency and compliance.
 - **Cost tools** (Budgets, Explorer, Alarms) help monitor and control spend.
 - **AWS Trusted Advisor** delivers actionable insights to improve cloud resource usage.
 - Regular reviews and automation (e.g., Stopinator) are essential for cost efficiency.
-

Let me know if you'd like:


- Flashcards or quiz based on this module
 - A visual infographic summary
 - A printable PDF of the summary
- I can prep any format that helps your study or presentation.

Here are **10 advanced and challenging multiple-choice questions (MCQs)** based on the **ACO-10: Managing Resource Consumption** module:

1. Which AWS service allows you to enforce tagging compliance by identifying non-compliant resources using managed rules?

- A. AWS Trusted Advisor
- B. AWS Config
- C. AWS CloudTrail
- D. AWS Identity and Access Management (IAM)


 **Correct Answer: B**

 **Explanation:** AWS Config supports the `required-tags` managed rule to check for compliance based on tagging policies.

2. Which IAM policy condition would you use to enforce the presence of multiple required tag keys during resource creation?

- A. `"ForAllValues:StringLike"`
- B. `"ForAnyValue:Exists"`
- C. `"ForAllValues:StringEquals"`
- D. `"StringNotEqualsIfExists"`


 **Correct Answer: C**

 **Explanation:** `"ForAllValues:StringEquals"` ensures that all specified tag keys are present in the request.

3. In the AWS CLI, what is the correct command to query EC2 instances based on a tag key named `SecurityCheck`?

- A. `aws ec2 list-instances --tags SecurityCheck`
- B. `aws ec2 describe-tags --filter "Name=tag:SecurityCheck"`
- C. `aws ec2 describe-instances --filters "Name=tag-key,Values=SecurityCheck"`
- D. `aws ec2 tag-instance --query "SecurityCheck"`


 **Correct Answer: C**

 **Explanation:** `--filters` is used with `describe-instances` to return resources that match the tag.

4. Which AWS service allows the creation of alerts based on estimated billing charges?

- A. AWS Config
- B. AWS Billing Console
- C. AWS Cost Explorer
- D. Amazon CloudWatch


 **Correct Answer: D**

 **Explanation:** Amazon CloudWatch billing alarms can notify users when charges exceed defined thresholds.

5. What is a typical use case for tags in automated cost-saving operations?

- A. Encrypting EBS volumes
- B. Automatically starting Lambda functions
- C. Scheduling the shutdown of non-production environments
- D. Creating IAM users with restricted permissions


 **Correct Answer: C**

 **Explanation:** Tags like `Environment=Dev` are used to target resources for shutdown after hours to save cost.

6. Which of the following tags would most likely help track usage by department for cost allocation?

- A. Owner
- B. Stack
- C. Application
- D. CostCenter


 **Correct Answer: D**

 **Explanation:** `CostCenter` is commonly used for billing and allocating costs by department or business unit.

7. Which Trusted Advisor check is only available to customers with Business or Enterprise support plans?

- A. Idle EC2 instances
- B. S3 Bucket Permissions
- C. MFA on root account
- D. All checks in the Cost Optimization category


 **Correct Answer: D**

 **Explanation:** Full access to Trusted Advisor checks—including advanced cost checks—is limited to paid support plans.

8. What is the maximum size for a Service Control Policy (SCP) document in AWS Organizations?

- A. 1,510 bytes
- B. 2,048 characters
- C. 5,120 bytes
- D. 10,240 characters


 **Correct Answer: C**

 **Explanation:** SCP documents have a maximum size of **5,120 bytes**.

9. Which CLI command can be used to apply a custom tag with a dynamic timestamp value?

- A. `aws ec2 create-tags --timestamp now`
- B. `aws ec2 apply-tag --date $(date)`
- C. `aws ec2 create-tags --tags "Key=Time,Value=$(date) "`
- D. `aws ec2 create-tags --tags "Key=SecurityCheck,Value=$TIMESTAMP"`

 **Correct Answer: D**

 **Explanation:** This uses an environment variable to dynamically apply the current timestamp.

10. Why is a serverless approach (like using Lambda) recommended for implementing automation scripts such as a Stopinator?

- A. It avoids IAM permissions issues
- B. It is costlier but more reliable
- C. It allows scripts to be run manually by developers
- D. It eliminates the need to keep EC2 instances running for automation tasks



Correct Answer: D



Explanation: Serverless scripts (Step Functions) can run on demand without the need for persistent compute infrastructure.



Module Objectives

By the end of this module, you'll be able to:

- Understand the benefits and functionality of **Amazon CloudWatch**
 - Explain **CloudWatch metrics**, alarms, logs, and events
 - Use **AWS CloudTrail** to track API activity and user actions
 - Understand how **AWS Config** helps with configuration compliance
 - Leverage **Amazon Athena** to query logs
 - Monitor your infrastructure for **security, compliance, and performance**
-



Main Topics Covered

1. Amazon CloudWatch

A **monitoring and observability service** that provides data and actionable insights for AWS resources and applications.

Key Features:

- **Standard Metrics:** Default metrics collected by AWS services
- **Custom Metrics:** Defined and published by users (via CLI/API/Agent)
- **CloudWatch Agent:** Used to collect system-level metrics from EC2 or on-prem servers

Alarms:

- Triggered when a metric meets a threshold (e.g., CPU > 80%)
- Can send alerts, start EC2 actions, or notify SNS

Dashboards:

- Visual summaries of metrics
- Supports widgets and custom layouts

Monitoring Levels:

- **Basic Monitoring** (default): 5-minute intervals
 - **Detailed Monitoring**: 1-minute intervals (additional cost)
-

2. Amazon CloudWatch Logs

Collects and monitors **log files** from EC2, Lambda, and other services.

Features:

- Logs are grouped by **Log Groups**
- **Metric Filters**: Search logs for specific terms or patterns (e.g., “error” or “400” responses)
- Filter patterns are **case-sensitive**
- Enables **custom metrics** based on logs

Insights:

- **CloudWatch Logs Insights** lets you run queries and analyze logs in near real-time
-

3. Amazon CloudWatch Events

Used to **respond to changes in your AWS environment** in near real-time.

Examples:

- Automatically react to EC2 instance state changes
 - Schedule Lambda functions or automated remediation tasks
-

4. AWS CloudTrail

A **governance and auditing service** that records AWS API calls and user activity.

Key Features:

- Tracks **who did what and when** in your AWS account
- Logs API calls from AWS CLI, Console, SDKs
- Automatically stores logs in **Amazon S3**
- Integrates with **CloudWatch Logs** and **Amazon Athena** for deeper analysis

Typical Uses:

- Identify unauthorized access
 - Audit S3 bucket access
 - Investigate changes to security groups
-

5. AWS Config

A configuration management and compliance tool.

Key Features:

- Tracks **resource configuration history**
 - Uses **managed rules** or custom rules to evaluate compliance
 - Example: Check if S3 buckets are encrypted or if resources are tagged properly
-

6. Amazon Athena Integration

Athena allows users to:

- Query **S3-stored logs** (e.g., **CloudTrail**, **VPC Flow Logs**)
 - Use **SQL** to analyze large datasets without ETL
 - Athena is **serverless**—you only pay for the queries you run
-

Labs and Projects

◆ Lab 6: Monitoring Your Applications and Infrastructure

- Tasks: Configure CloudWatch alarms, install agents, monitor metrics, and logs
- Setup monitoring for EC2, view alarms, and understand thresholds

◆ Activity 9: Monitoring with AWS CloudTrail

Scenario: A hacked web server (Mom & Pop Café) needs to be investigated for changes made to the content.

- Use CloudTrail logs to investigate suspicious login attempts and identify **who made the changes**
 - Visualize logs with Athena and run queries to detect access patterns
-

Security & Compliance Use Cases

- Monitor for failed login attempts or access from suspicious IPs
 - Detect when services (e.g., S3 buckets) are made public
 - Integrate alarms with CloudTrail for real-time alerting
-

Key Takeaways

- **Amazon CloudWatch** helps you **monitor performance and resource health**
 - **CloudWatch Logs and Events** support **event-driven actions**
 - **AWS CloudTrail** enables **auditing and visibility** into user activity
 - **AWS Config** ensures **resource compliance and configuration tracking**
 - **Amazon Athena** provides powerful **SQL-based log analysis**
-

Real-World Scenario Example

Hospital Compliance Audit:

- Use CloudWatch Logs and CloudTrail to track user access and system changes
 - Prove HIPAA compliance with detailed logs
 - Confirm the CloudWatch Agent is running via CLI (outputs: Action, Start Time, Status)
-

1. Which of the following statements is **TRUE** about CloudWatch custom metrics?

- A. Custom metrics are automatically collected by AWS services.
- B. Custom metrics are stored in the same namespace as standard metrics.
- C. Custom metrics require a user-defined namespace and must be published via API, CLI, or agent.
- D. Custom metrics are only available in CloudWatch Logs.

 **Correct Answer: C**

 **Explanation:** Custom metrics are not collected automatically; users must define and publish them using tools like the CloudWatch Agent or AWS CLI.

2. What is the default frequency for basic monitoring of EC2 instances in Amazon CloudWatch?

- A. Every 1 second
- B. Every 1 minute
- C. Every 3 minutes
- D. Every 5 minutes

✓ **Correct Answer: D**

📝 **Explanation:** Basic monitoring provides metrics at 5-minute intervals. Detailed monitoring offers 1-minute intervals at additional cost.

3. In CloudWatch Logs, which of the following is TRUE about metric filters?

- A. They only work with predefined AWS log formats.
- B. All terms in a filter pattern must be present in the log event for it to match.
- C. Filter patterns support only lowercase case-sensitive matching.
- D. Metric filters are used exclusively for EC2 metrics.

✓ **Correct Answer: B**

📝 **Explanation:** Metric filters are case-sensitive and require all terms in the pattern to be present in a log event to register a match.

4. Which services can be configured to send log data to Amazon CloudWatch Logs? (Select TWO)

- A. Amazon S3
- B. Amazon RDS
- C. Amazon EC2 (with agent)
- D. AWS Lambda
- E. AWS Config


✓ **Correct Answers: C and D**

📝 **Explanation:** EC2 with the CloudWatch agent and Lambda functions can both send logs directly to CloudWatch Logs.

5. What is the main function of AWS CloudTrail in terms of security and compliance?

- A. Filtering instance logs based on events
- B. Managing EC2 storage lifecycle
- C. Recording and storing AWS account API activity for auditing
- D. Creating IAM policies automatically


 **Correct Answer: C**

 **Explanation:** CloudTrail provides visibility into user activity by recording API calls across AWS services.

6. Which of the following steps is **OPTIONAL** when configuring an AWS CloudTrail trail?

- A. Define an S3 bucket to store logs
- B. Configure an Amazon SNS topic for notifications
- C. Enable encryption with KMS
- D. Create a CloudWatch dashboard


 **Correct Answer: D**

 **Explanation:** CloudWatch dashboards are not required for CloudTrail trails, whereas the others are part of best practices.

7. What AWS CLI output confirms that the CloudWatch Agent is running correctly? (Choose **THREE**)

- A. Status
- B. Effect
- C. Start Time
- D. Action
- E. LogLevel

 **Correct Answers: A, C, D**

 **Explanation:** Valid CLI output fields include `Action`, `Start Time`, and `Status`.

8. What advantage does using Amazon Athena provide when analyzing CloudTrail logs?

- A. You don't need to define a schema for analysis
- B. It's a serverless SQL engine that allows fast queries on S3 data
- C. It only works with CloudWatch metrics
- D. It encrypts the logs automatically for compliance

✓ **Correct Answer: B**

📝 **Explanation:** Athena lets you query logs stored in S3 using SQL, and you only pay per query.

9. Which AWS service allows you to define rules that check if resources comply with required configurations (e.g., encryption)?

- A. AWS Inspector
- B. AWS CloudTrail
- C. AWS Config
- D. AWS Shield

✓ **Correct Answer: C**

📝 **Explanation:** AWS Config uses rules to check if AWS resources are compliant with specified configurations.

10. A CloudWatch alarm transitions into the ALARM state. Which of the following statements is TRUE?

- A. It always indicates a critical system failure
- B. It only triggers if all metrics are above threshold
- C. It triggers based on a metric breaching a threshold; not necessarily an emergency
- D. It disables the affected AWS service automatically

✓ **Correct Answer: C**

📝 **Explanation:** An ALARM state simply means the threshold has been breached; it may not be an emergency depending on the alarm logic.