AWS Resource tagging automation implementation runbook

Version History

Sr No	Version ID	Date	Author	Remark
1	1.0	30-09-2025	Manoj P	First Version

Table of Contents

1. Overview

1.1 Purpose

This runbook provides step-by-step instructions for implementing an automated AWS resource tagging compliance system. The solution ensures all newly created AWS resources are properly tagged according to organizational policies.

1.2 Business Requirements

- Enforce mandatory sponsor=Pratik tag on all AWS resources
- Automatically add user identification and creation timestamp tags
- Send notifications for non-compliant resources
- Support 12+ AWS services with automatic remediation for current version i.e v-1

1.3 Solution Benefits

- **Compliance**: Ensures 100% tag compliance across AWS resources
- **Automation**: Reduces manual tagging overhead
- **Visibility**: Provides audit trail and notifications
- **Cost Management**: Enables accurate cost allocation and tracking

2. Architecture

2.1 High-Level Architecture

AWS Resources Creation -> CloudTrail -> EventBridge -> Lambda Function -> SNS Notifications -> Auto-Tagging

2.2 Components

Component	Purpose
CloudTrail	Captures AWS API calls for resource creation
EventBridge	Filters and routes events to Lambda function
Lambda Function	Core logic for tag compliance and remediation
SNS Topic	Sends notifications for compliance violations
IAM Roles	Provides necessary permissions for the solution

2.3 Supported AWS Services

Service	Events	Resources
Amazon S3	CreateBucket	S3 Buckets
Amazon EC2	RunInstances, CreateImage, CreateVolume, CreateVpc, CreateSubnet	Instances, AMIs, Volumes, VPCs, Subnets
Amazon RDS	CreateDBInstance, CreateDBSnapshot	RDS Instances, Snapshots
AWS Lambda	CreateFunction20150331	Lambda Functions
Amazon SageMaker	CreateDomain, CreateUserProfile, CreateModel	Domains, User Profiles, Models
AWS Glue	CreateJob, CreateDatabase, CreateTable, CreateCrawler	Jobs, Databases, Tables, Crawlers
Amazon MSK	CreateCluster	Kafka Clusters
AWS Service Catalog	CreatePortfolio, CreateProduct	Portfolios, Products
AWS DMS	CreateReplicationInstance, CreateEndpoint, CreateReplicationTask	Replication Instances, Endpoints, Tasks
AWS CodePipeline	CreatePipeline	Pipelines
AWS IAM	CreateRole, CreatePolicy	Roles, Policies
Amazon EventBridge	CreateEventBus, PutRule	Event Buses, Rules

3. Prerequisites

3.1 AWS Account Requirements

- AWS account with appropriate permissions

- CloudTrail enabled and configured
- EventBridge service available
- SNS topic for notifications

3.2 IAM Permissions Required

- Lambda execution role with tagging permissions for all services
- EventBridge permissions to invoke Lambda
- SNS publish permissions
- CloudWatch Logs permissions

3.3 Technical Requirements

- Python 3.9+ runtime for Lambda
- boto3 library (included in Lambda runtime)

4. Implementation Steps

Step 1: Create SNS Topic

aws sns create-topic --name aws-resource-tagging-notifications

Note: Record the Topic ARN for later configuration.

Step 2: Create Lambda Execution Role

Create IAM role with the following policies:

- AWSLambdaBasicExecutionRole
- Custom policy with tagging permissions (see Configuration section)

Step 3: Deploy Lambda Function

1. Create Lambda function with Python 3.9 or 3.9+ runtime
2. Upload the tagging_final.py code
3. Set environment variable SNS_TOPIC_ARN
4. Configure timeout to 5 minutes
5. Assign the execution role created in Step 2

Step 4: Create EventBridge Rule

1. Create new EventBridge rule
2. Use the event pattern from data_team_tagging_resources_evntbridge_rule.json
3. Set Lambda function as target
4. Enable the rule

Step 5: Configure CloudTrail

Ensure CloudTrail captures API calls for all supported services.

5. Configuration

5.1 Environment Variables

Set the following environment variable in Lambda function:

SNS_TOPIC_ARN=arn:aws:sns:region:account-id:aws-resource-tagging-notifications

5.2 EventBridge Rule Pattern

```
{
  "source": ["aws.events", "aws.iam", "aws.s3", "aws.ec2", "aws.sagemaker", "aws.glue",
"aws.rds", "aws.lambda", "aws.kafka", "aws.servicecatalog", "aws.dms", "aws.codepipeline"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": {
    "eventName": ["CreateEventBus", "CreateRule", "CreateRole", "CreatePolicy",
"CreateBucket", "CreateVpc", "CreateSubnet", "RunInstances", "CreateImage",
"CreateVolume", "CreateDomain", "CreateUserProfile", "CreateModel", "CreateJob",
"CreateDatabase", "CreateTable", "CreateCrawler", "CreateDBInstance", "CreateDBSnapshot",
"CreateFunction20150331", "CreateCluster", "CreatePortfolio", "CreateProduct",
"CreateReplicationInstance", "CreateEndpoint", "CreateReplicationTask", "CreatePipeline"]
  }
}
```

5.3 Tagging Rules

The system automatically applies these tags to non-compliant resources:

Tag Key	Tag Value	Description
sponsor	Pratik	Mandatory compliance tag
user	{user_email}	Extracted from event

ResourceCreationDate	{timestamp_IST}	Creation timestamp in IST

6. Testing

6.1 Test Scenarios

1. **S3 Bucket Creation**: Create bucket without sponsor tag
2. **EC2 Instance Launch**: Launch instance without sponsor tag
3. **RDS Instance Creation**: Create RDS instance without sponsor tag

6.2 Expected Behavior

1. Resource created without sponsor tag
2. EventBridge triggers Lambda function
3. Lambda checks for sponsor tag (not found)
4. SNS notification sent
5. Required tags automatically added to resource

6.3 Validation Steps

1. Check CloudWatch Logs for Lambda execution
2. Verify SNS notification received
3. Confirm tags applied to created resource
4. Validate tag values are correct

7. Monitoring

7.1 CloudWatch Metrics

Monitor the following metrics:

- Lambda function invocations
- Lambda function errors
- Lambda function duration
- SNS message delivery

7.2 CloudWatch Alarms

Set up alarms for:

- Lambda function failures
- High Lambda execution duration

- SNS delivery failures

7.3 Logging

Lambda function logs include:

- Event details
- User email extraction
- Resource ARN identification
- Tag compliance status
- Tagging operation results

8. Troubleshooting

8.1 Common Issues

Lambda Function Not Triggered

Symptoms: Resources created but no Lambda execution

Causes:

- EventBridge rule not enabled
- CloudTrail not capturing events
- Event pattern mismatch

Resolution:

1. Verify EventBridge rule is enabled
2. Check CloudTrail configuration
3. Validate event pattern matches actual events

Tagging Failures

Symptoms: Notifications sent but tags not applied

Causes:

- Insufficient IAM permissions
- Resource not found
- Service-specific API limitations

Resolution:

1. Check Lambda execution role permissions
2. Verify resource exists and is accessible
3. Review service-specific tagging requirements

SNS Notifications Not Received

Symptoms: Lambda executes but no notifications

Causes:

- Incorrect SNS topic ARN
- SNS topic permissions
- Email subscription not confirmed

Resolution:

1. Verify SNS_TOPIC_ARN environment variable
2. Check SNS topic permissions
3. Confirm email subscription

8.2 Debug Steps

1. Check CloudWatch Logs for Lambda function
2. Verify EventBridge rule metrics
3. Test SNS topic manually
4. Validate IAM permissions using IAM Policy Simulator