AWS Tag Enforcement Project

1. Project Overview

The AWS Tag Enforcement Project ensures that AWS resources comply with specific tagging policies. A Lambda function validates EC2 instance tags against a predefined set of allowed values stored in DynamoDB. If an instance lacks the required tags or contains incorrect values, it is automatically terminated.

2. Architecture & Components

- **AWS Lambda**: Processes EC2 instance creation events and enforces tagging rules.
- **AWS DynamoDB**: Stores predefined tag keys and allowed values.
- **AWS EventBridge**: Captures instance creation events and triggers Lambda.
- **AWS EC2**: The target resource that must comply with tag requirements.
- **AWS CloudWatch**: Logs Lambda execution details and errors for debugging.

3. Setting Up AWS Lambda Function

1. Open the AWS Lambda Console.
2. Click 'Create Function'.
3. Select 'Author from Scratch'.
4. Enter function name: `TagEnforcementLambda`.
5. Select runtime: Python 3.x.
6. Under 'Permissions', attach a policy with DynamoDB Read and EC2 Terminate permissions.
7. Deploy the Lambda function with the provided code.
8. Configure environment variables if necessary.

4. Creating an EventBridge Rule

1. Open the AWS EventBridge Console.
2. Click 'Rules' and then 'Create Rule'.
3. Enter a name for the rule (e.g., `EC2TagValidation`).
4. Under 'Define pattern', choose 'Event Pattern'.
5. Select 'AWS events' and 'EC2 Instance State-change Notification'.
6. Choose 'Specific detail type': `AWS API Call via CloudTrail`.
7. Under 'Target', select the Lambda function `TagEnforcementLambda`.
8. Click 'Create'.

5. Lambda Function Code

```
import boto3
import json

dynamodb = boto3.resource('dynamodb')
ec2 = boto3.client('ec2')
DYNAMODB_TABLE_NAME = "Pre-definedTags"

def lambda_handler(event, context):
                                                                instance_id               =
event['detail']['responseElements']['instancesSet']['items'][0]['instanceId']
                    tag_specifications     =     event['detail'].get('requestParameters',
{}).get('tagSpecificationSet', [])
    resource_tags = {tag['key']: tag['value'] for tag_spec in tag_specifications for tag
in tag_spec.get('tags', [])}

    table = dynamodb.Table(DYNAMODB_TABLE_NAME)
    mandatory_tag_valid = False

    for tag_key, tag_value in resource_tags.items():
        response = table.scan(FilterExpression="#key = :key_value",
                              ExpressionAttributeNames={"#key": "Key"},
                              ExpressionAttributeValues={":key_value": tag_key})
        if response['Items'] and tag_value in [item.get('Value', '').strip() for item in
response['Items']]:
            mandatory_tag_valid = True
            break

    if not mandatory_tag_valid:
        ec2.terminate_instances(InstanceIds=[instance_id])
```

6. Common Errors & Troubleshooting

- **KeyError: 0**: Happens when `tagSpecificationSet` is missing in the event. Solution: Check event structure.
- **No tags found**: Happens when an instance is created without tags. Solution: Ensure tags are added at launch.
- **Termination failures**: Occur when attempting to terminate an already deleted instance. Solution: Handle missing instance gracefully.

7. Step-by-Step Setup

1. **Create DynamoDB Table**:
   - Table Name: `Pre-definedTags`
   - Attributes: `Key` (Primary Key), `Value`
2. **Deploy Lambda Function**:
   - Grant permissions to read DynamoDB and terminate instances.
3. **Configure EventBridge Rule**:
   - Trigger Lambda when an EC2 instance is created.

4. **Test the Setup**:
   - Create an EC2 instance with valid tags.
   - Create an instance without tags and verify termination.
5. **Monitor CloudWatch Logs**:
   - Review extracted tags and errors for debugging.

8. Conclusion

The AWS Tag Enforcement Project ensures compliance with tagging policies by automating enforcement at instance creation. By integrating AWS Lambda, EventBridge, and DynamoDB, non-compliant instances are automatically terminated. Future improvements include proactive compliance monitoring via AWS Config rules and expanded enforcement across additional AWS resources.