AWS Config

Project 5: AWS Config Security Best Practices Documentation

Introduction

AWS Config is a powerful service that continuously monitors, records, and evaluates AWS resource configurations against best practices and security policies. This project helps automate compliance auditing by setting up AWS Config rules to check security best practices, such as ensuring encryption is enabled, IAM policies follow least privilege principles, and resources are properly tagged.

Prerequisites

- Active AWS account with appropriate permissions
- Basic understanding of AWS services (S3, EC2, IAM)
- Access to AWS Management Console
- Familiarity with security best practices and compliance requirements

Step 1: Set Up AWS Config

1.1 Sign in to AWS Management Console

Navigate to the AWS Management Console and sign in with your credentials.

1.2 Navigate to AWS Config

Open AWS Config from the AWS Console services menu.

1.3 Choose a Recorder

- Click **"Set up AWS Config"**
- Select the AWS **resources** you want to track (e.g., S3, EC2, IAM)

- Choose whether to record all resources or specific resource types

1.4 Set Up an S3 Bucket for Logs

- Create a new S3 bucket or specify an existing one for storing configuration history
- Ensure the bucket has appropriate permissions for AWS Config to write logs
- Enable versioning on the bucket for better audit trails

1.5 Enable AWS Config Recorder

- Start recording resource configurations across your AWS account
- Verify that the recorder status shows as "Recording"

Step 2: Define Compliance Rules

2.1 Navigate to the AWS Config Rules Section

From the AWS Config dashboard, select **Rules** from the left-hand menu.

2.2 Create AWS Config Rules

Predefined Rules (Recommended)

- `s3-bucket-public-read-prohibited` - Ensures S3 buckets are not publicly readable
- `s3-bucket-server-side-encryption-enabled` - Verifies S3 bucket encryption
- `iam-user-no-inline-policies` - Checks that IAM users don't have inline policies
- `ec2-encrypted-volumes` - Ensures EBS volumes are encrypted
- `required-tags` - Validates that resources have mandatory tags

Custom Rules

- Create custom rules using AWS Lambda for organization-specific requirements
- Define custom evaluation logic based on your security policies

2.3 Set Rule Parameters

- Define required settings for each rule
- Configure trigger types (configuration changes or periodic)
- Set evaluation frequency for periodic rules
- Specify resource scopes and exceptions if needed

2.4 Enable Automatic Remediation (Optional)

- Use AWS Systems Manager Automation to fix non-compliant resources automatically
- Select appropriate remediation actions from the available SSM documents
- Configure retry attempts and parameters for remediation
- Test remediation actions in a non-production environment first

Step 3: Evaluate Compliance & Generate Reports

3.1 Monitor Compliance Status

- Go to **AWS Config Dashboard** and check rule evaluations
- Review the compliance summary showing compliant vs. non-compliant resources
- Use filters to view specific resource types or compliance states

3.2 View Non-Compliant Resources

- Identify which AWS resources violate best practices
- Click on individual rules to see detailed compliance information
- Review resource configuration timelines to understand when non-compliance occurred

3.3 Generate Compliance Reports

- Use AWS Config's reporting feature to generate compliance summaries
- Export reports in CSV or JSON format for documentation
- Schedule regular report generation for ongoing compliance tracking

- Share reports with stakeholders and auditors as needed

3.4 Enable Amazon SNS Notifications (Optional)

- Create an SNS topic for compliance notifications
- Subscribe email addresses or other endpoints to the topic
- Configure AWS Config to send notifications when resources become non-compliant
- Set up filtering rules to avoid alert fatigue

Step 4: Automate Compliance Auditing with AWS Lambda (Optional)

4.1 Create an AWS Lambda Function

Write a Python function to check compliance and log results. Example structure:

```
import boto3
import json

def lambda_handler(event, context):
    config = boto3.client('config')

    # Get compliance details
    compliance_summary = config.describe_compliance_by_config
_rule()

    # Process and log results
    for rule in compliance_summary['ComplianceByConfigRule
s']:
        rule_name = rule['ConfigRuleName']
        compliance_type = rule['Compliance']['ComplianceTyp
e']

        print(f"Rule: {rule_name}, Status: {compliance_typ
e}")
```

```
    return {
        'statusCode': 200,
        'body': json.dumps('Compliance check completed')
    }
```

4.2 Integrate with AWS Config

- Configure AWS Lambda to trigger when a rule is violated
- Set up EventBridge rules to capture Config compliance changes
- Pass relevant event data to the Lambda function
- Ensure the Lambda execution role has necessary permissions

4.3 Remediate Issues Automatically

- Use AWS Systems Manager to enforce compliance automatically
- Create custom SSM automation documents for specific remediation tasks
- Implement approval workflows for critical remediation actions
- Log all remediation actions for audit purposes

Best Practices

- **Start with critical resources:** Begin by monitoring high-priority resources before expanding coverage
- **Regular review:** Schedule periodic reviews of compliance rules and update them as needed
- **Tag strategy:** Implement a consistent tagging strategy across all resources
- **Cost management:** Monitor AWS Config costs, especially for high-frequency evaluations
- **Testing:** Always test rules and remediation actions in non-production environments

- **Documentation:** Maintain documentation of all custom rules and remediation procedures

Common Compliance Rules to Implement

Security Rules

- S3 bucket encryption enabled
- S3 bucket public access blocked
- EBS volumes encrypted
- RDS instances encrypted
- Security groups with restricted ingress
- IAM password policy compliance

Operational Rules

- Required tags present on resources
- EC2 instances using approved AMIs
- CloudTrail enabled in all regions
- VPC flow logs enabled

Troubleshooting

Config Recorder Not Recording

- Verify IAM role permissions for AWS Config
- Check S3 bucket policies allow Config to write
- Ensure the recorder is in "Recording" state

Rules Not Evaluating

- Check rule configuration and trigger settings
- Verify resource types are being recorded

- Review CloudWatch Logs for Lambda-based custom rules

False Positives

- Review rule parameters and adjust as needed
- Use resource exemptions for valid exceptions
- Consider creating custom rules for specific requirements

Cost Considerations

- **Configuration items recorded:** Charged per configuration item recorded
- **Rule evaluations:** Charged per rule evaluation
- **S3 storage:** Storage costs for configuration snapshots
- **Optimization tips:** Limit recording to essential resources and use appropriate evaluation frequencies

Conclusion

By leveraging AWS Config, automated rules, and optional remediation actions, you can continuously audit your AWS environment for security best practices. This ensures compliance with industry standards, improves security posture, and simplifies governance. Regular monitoring and updating of Config rules will help maintain a secure and compliant AWS infrastructure.

Additional Resources

- AWS Config Developer Guide
- AWS Config Managed Rules Reference
- AWS Systems Manager Automation
- AWS Compliance Programs