Prowler

Opensource Security Tool

AWS ProServe – Choi, Jaewook

2023.03.23

Disclaimer

Disclaimers. The following applies to this document and all other documents, information, data, and responses (written or verbal) provided by Amazon Web Services, Inc. or any of its affiliates (collectively, "AWS") in connection with responding to this request and other related requests (collectively, this "Response"): This Response is expressly (a) informational only and provided solely for discussion purposes, (b) non-binding and not an offer to contract that can be accepted by any party, (c) provided "as is" with no representations or warranties whatsoever, and (d) based on AWS's current knowledge and may change at any time due to a variety of factors such as changes to your requirements or changes to AWS's service offerings. All obligations must be set forth in a separate, definitive written agreement between the parties. Neither party will have any liability for any failure or refusal to enter into a definitive agreement. All use of AWS's service offerings will be governed by the AWS Customer Agreement available at http://aws.amazon.com/agreement/ (or other definitive written agreement between the parties governing the use of AWS's service offerings) (as applicable, the "Agreement"). If the parties have an applicable Nondisclosure Agreement ("NDA"), then the NDA will apply to all Confidential Information (as defined in the NDA) disclosed in connection with this Response. AWS's pricing is publicly available and subject to change in accordance with the Agreement. Pricing information (if any) provided in this Response is only an estimate and is expressly not a binding quote. Fees and charges will be based on actual usage of AWS services, which may vary from the estimates provided. Nothing in this Response will modify or supplement the terms of the Agreement or the NDA. No part of this Response may be disclosed without AWS's prior written consent.

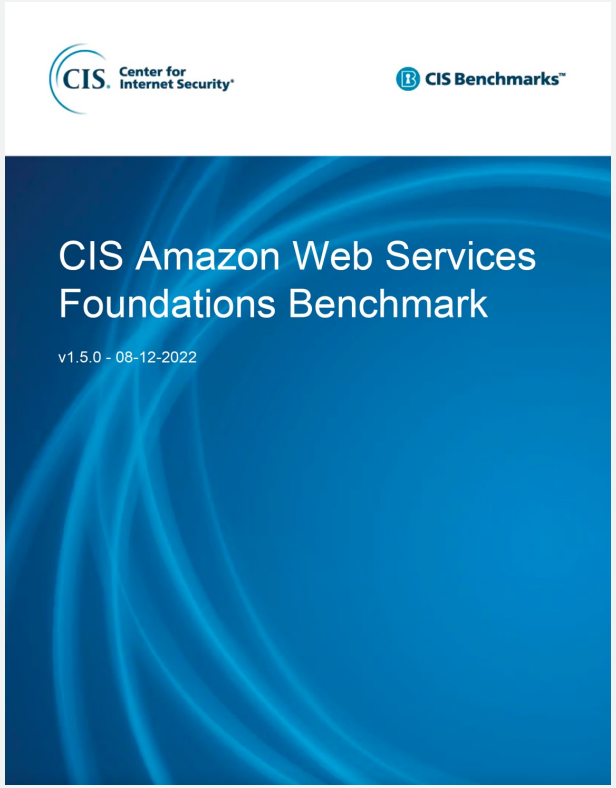
Contents

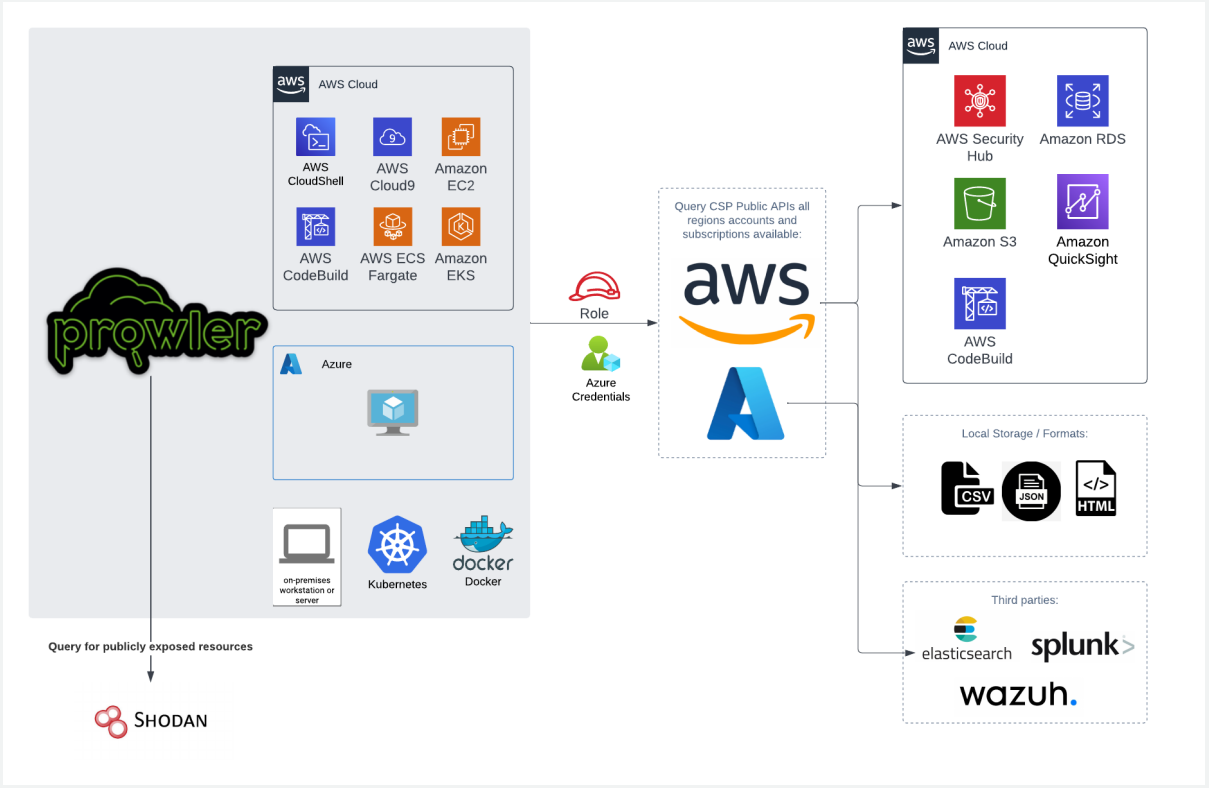
Prowler 소개

Prowler 소개

Prowler는 AWS 및 Azure 보안 모범 사례 평가, 감사, 사고 대응, 지속적인 모니터링, 강화 및 포렌식 준비를 수행하는 오픈 소스 보안 도구입니다.

- Prowler는 CIS, PCI-DSS, ISO27001, GDPR, HIPAA, FFIEC, SOC2, AWS FTR, ENS 및 사용자 지정 보안 프레임워크를 다루는 수백 가지 컨트롤이 포함되어 있습니다.





Prowler 소개

버전 비교

	Version 2	Version 3
Compliance	CIS Level1, CIS Level2, FFIEC, HIPPA, PCI, SOC2, CIS Implementation Group2, ISO 27001	cis_1.4, cis_1.5, ens_rd2022, aws_audit_manager_control_tower_guardrails, aws_foundational_security_best_practices, cisa, fedramp_low_revision_4, fedramp_moderate_revision_4, ffiec, gdpr, gxp_eu_annex_11, gxp_21_cfr_part_11, hipaa, nist_800_53_revision_4, nist_800_53_revision_5, nist_800_171_revision_2, nist_csf_1.1, pci_3.2.1, rbi_cyber_security_framework, soc2
결과 지원 형식	text, mono, csv, json, json-asff, junit-xml, html	csv, json, json-asff, html
결과 저장소	Local, S3 Bucket, Security Hub, DB	Local, S3 Bucket, Security Hub, DB
Multi Account 지원	가능	가능
Inventory 출력	가능	가능
예외처리	가능	가능
Container	가능	가능

Prowler IAM Policies

Prowler IAM Policies

필수 & 추가 IAM Policies

Prowler를 실행하기 위해 아래의 2가지 AWS Managed IAM Policy가 필수적으로 필요합니다.

- **arn:aws:iam::aws:policy/SecurityAudit**
- **arn:aws:iam::aws:policy/job-function/ViewOnlyAccess**

정확한 점검을 위해 다음과 같은 추가 IAM Policy가 필요합니다.

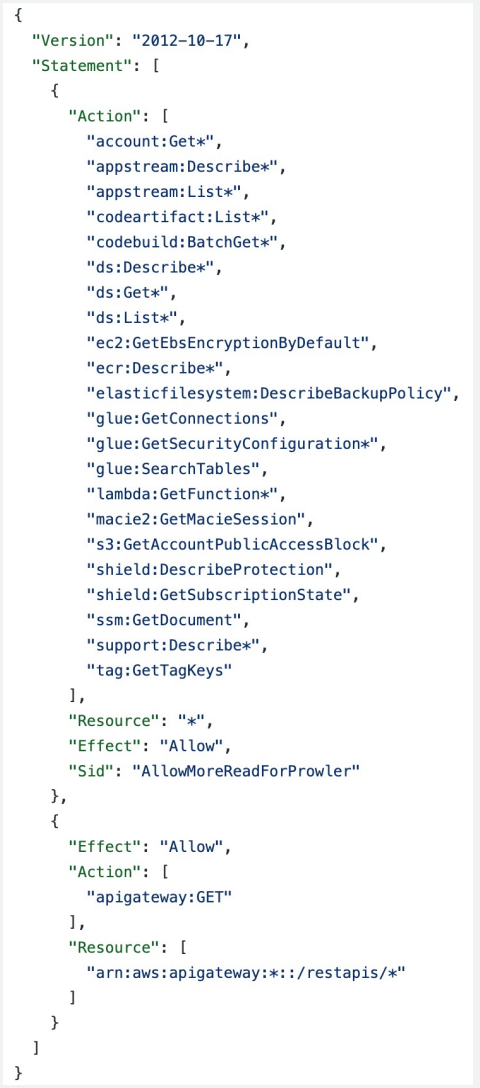
- **prowler-additions-policy.json**

만약 점검 결과를 AWS Security Hub에 보내고자 한다면,
다음과 같은 IAM Policy가 필요합니다.

- **prowler-security-hub.json**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "securityhub:BatchImportFindings",
        "securityhub:GetFindings"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

< prowler-security-hub.json >



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "account:Get*",
        "appstream:Describe*",
        "appstream:List*",
        "codeartifact:List*",
        "codebuild:BatchGet*",
        "ds:Describe*",
        "ds:Get*",
        "ds:List*",
        "ec2:GetEbsEncryptionByDefault",
        "ecr:Describe*",
        "elasticfilesystem:DescribeBackupPolicy",
        "glue:GetConnections",
        "glue:GetSecurityConfiguration*",
        "glue:SearchTables",
        "lambda:GetFunction*",
        "macie2:GetMacieSession",
        "s3:GetAccountPublicAccessBlock",
        "shield:DescribeProtection",
        "shield:GetSubscriptionState",
        "ssm:GetDocument",
        "support:Describe*",
        "tag:GetTagKeys"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowMoreReadForProwler"
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
      "Resource": [
        "arn:aws:apigateway:*::/restapis/*"
      ]
    }
  ]
}
```

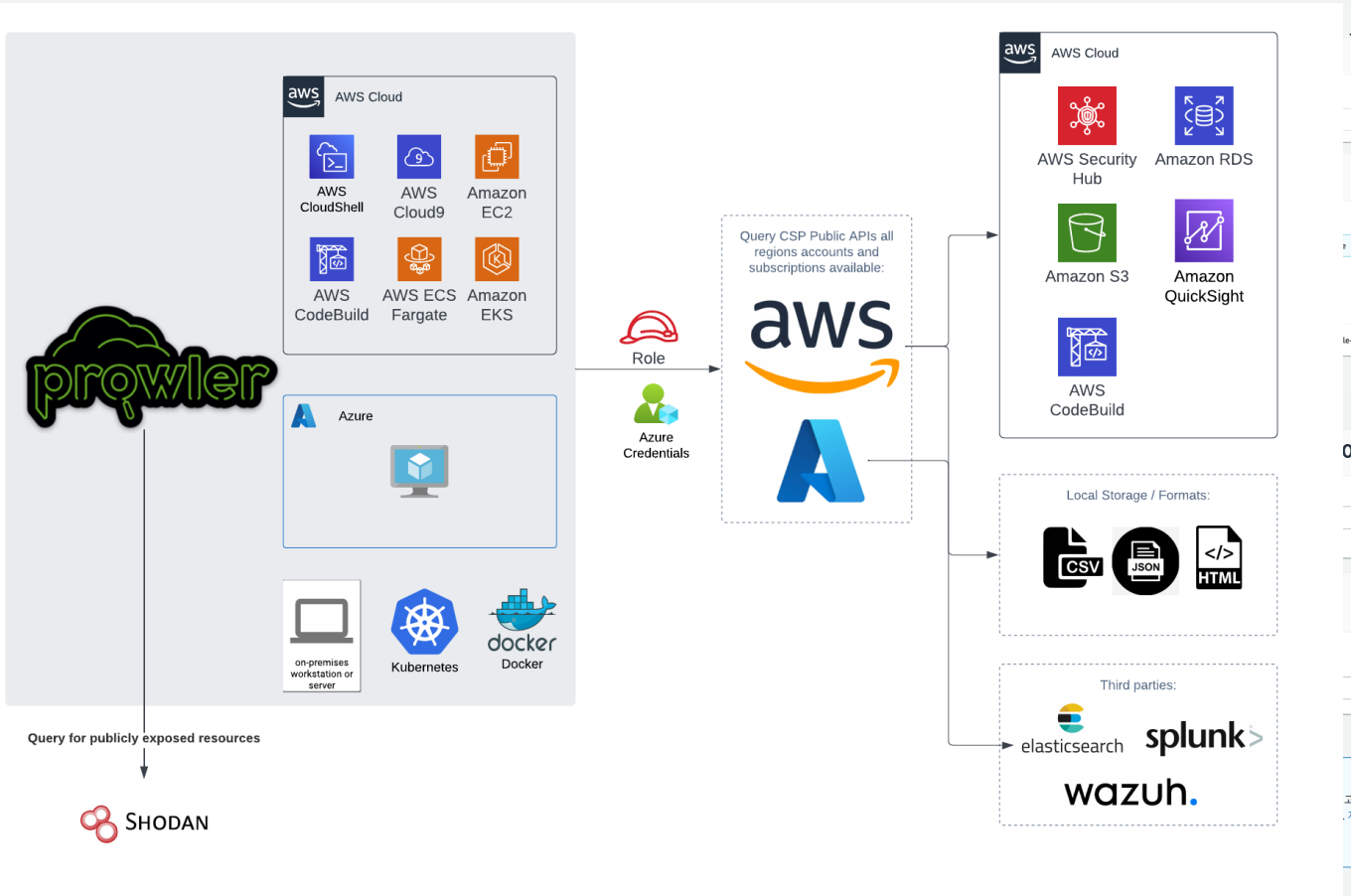
< prowler-additionas-policy.json >

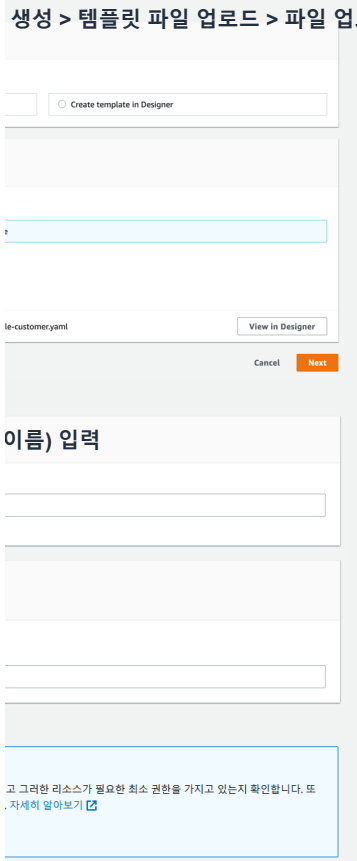
Prowler IAM Policies

점검 대상에 IAM Role 생성

점검 수행 Account와 점검 대상 Account가 동일한 경우, 아래 Code로 CloudFormation 스택 생성

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: |
  This template creates an AWS IAM Role with an inline policy and two AWS
  attached for security assessment
Parameters:
  ProwlerRoleName:
    Description: |
      Name of the IAM role that will have these policies attached. Default
    Type: String
    Default: 'ECSAProwlerRole'
Resources:
  ECSAProwlerRole:
    Type: AWS::IAM::Role
    Properties:
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/SecurityAudit'
        - 'arn:aws:iam::aws:policy/job-function/ViewOnlyAccess'
      RoleName: !Sub ${ProwlerRoleName}
      Policies:
        - PolicyName: ProwlerExecRoleAdditionalViewPrivileges
          PolicyDocument:
            Version : '2012-10-17'
            Statement:
            - Effect: Allow
              Action:
                - 'account:Get*'
                - 'appstream:DescribeFleets'
                - 'codeartifact:ListRepositories'
                - 'codebuild:BatchGetBuilds'
                - 'ds:Get*'
                - 'ds:Describe*'
                - 'ds:List*'
                - 'ec2:GetEbsEncryptionByDefault'
                - 'ecr:Describe*'
                - 'elasticfilesystem:DescribeBackupPolicy'
                - 'glue:GetConnections'
                - 'glue:GetSecurityConfiguration'
                - 'glue:SearchTables'
                - 'lambda:GetFunction'
                - 'macie2:GetMacieSession'
                - 's3:GetAccountPublicAccessBlock'
                - 's3:GetEncryptionConfiguration'
                - 's3:GetBucketPublicAccessBlock'
                - 'shield:DescribeProtection'
                - 'shield:GetSubscriptionState'
                - 'ssm:GetDocument'
                - 'support:Describe*'
                - 'tag:GetTagKeys'
              Resource: '*'
            - Effect: Allow
              Action:
                - 'apigateway:GET'
              Resource: 'arn:aws:apigateway:*::/restapis/*'
            - Effect: Allow
              Action:
                - 'securityhub:BatchImportFindings'
                - 'securityhub:GetFindings'
              Resource: '*'
```



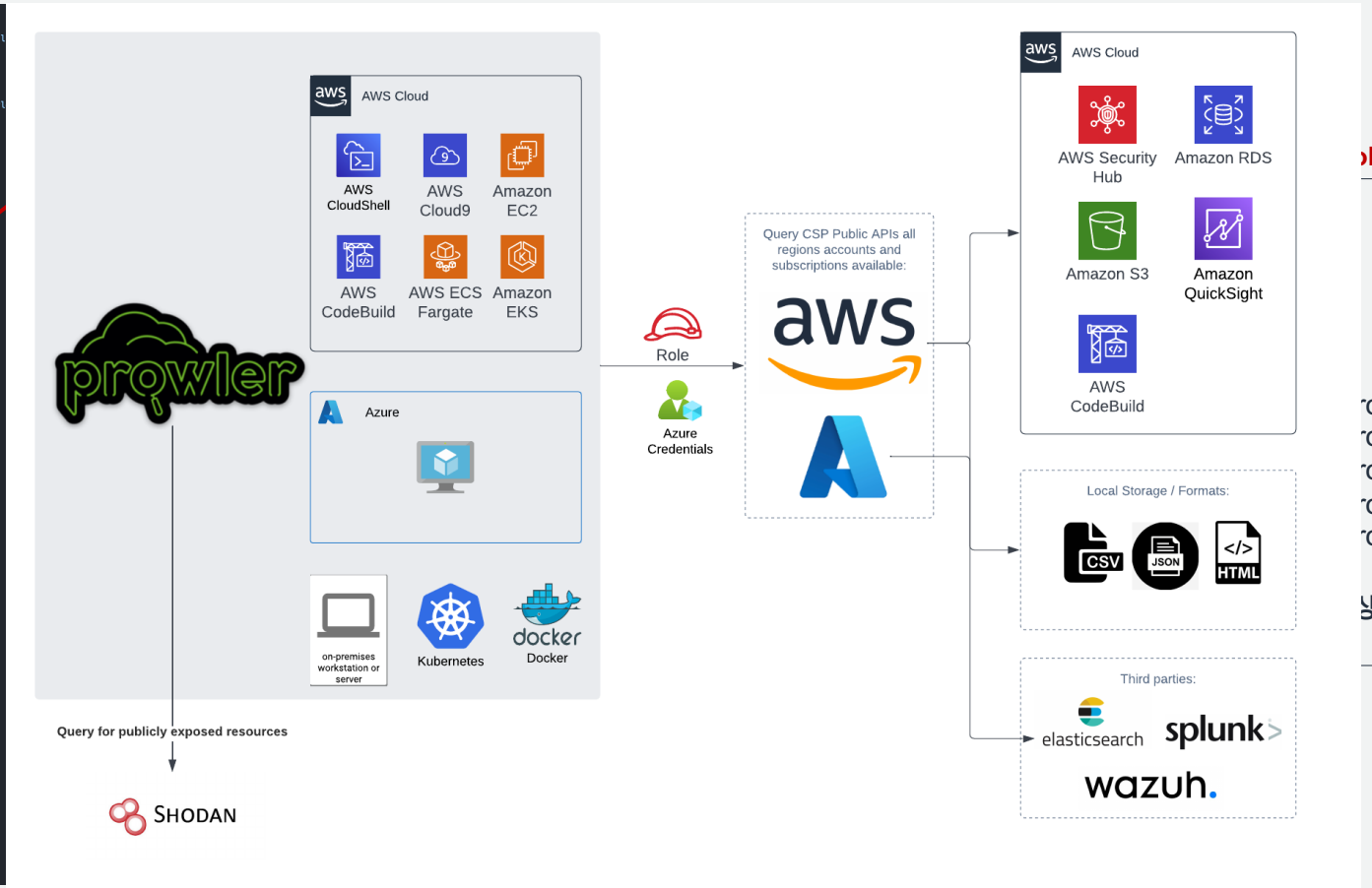


Prowler IAM Policies

점검 대상에 IAM Role 생성

점검 수행 Account와 점검 대상 Account가 다른 경우, 점검 대상 Account에 아래 Code로 CloudFormation 스택 생성

```
AWSTemplateFormatVersion: '2010-09-09'
Description: |
  This template creates an AWS IAM Role with an inline policy and two AWS managed pol
  attached for security assessment
Parameters:
  ProwlerRoleName:
    Description: |
      Name of the IAM role that will have these policies attached. Default: ECSAProwl
    Type: String
    Default: 'ECSAProwlerRole'
Resources:
  ECSAProwlerRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              AWS:  arn:aws:iam::{account_id}:root
            Action: 'sts:AssumeRole'
      MaxSessionDuration: 43200
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/SecurityAudit'
        - 'arn:aws:iam::aws:policy/job-function/ViewOnlyAccess'
      RoleName: !Sub ${ProwlerRoleName}
      Policies:
        - PolicyName: ProwlerExecRoleAdditionalViewPrivileges
          PolicyDocument:
            Version : '2012-10-17'
            Statement:
            - Effect: Allow
              Action:
                - 'account:Get*'
                - 'appstream:DescribeFleets'
                - 'codeartifact:ListRepositories'
                - 'codebuild:BatchGetBuilds'
                - 'ds:Get*'
                - 'ds:Describe*'
                - 'ds:List*'
                - 'ec2:GetEbsEncryptionByDefault'
                - 'ecr:Describe*'
                - 'elasticfilesystem:DescribeBackupPolicy'
                - 'glue:GetConnections'
                - 'glue:GetSecurityConfiguration'
                - 'glue:SearchTables'
                - 'lambda:GetFunction'
                - 'macie2:GetMacieSession'
                - 's3:GetAccountPublicAccessBlock'
                - 's3:GetEncryptionConfiguration'
                - 's3:GetBucketPublicAccessBlock'
                - 'shield:DescribeProtection'
                - 'shield:GetSubscriptionState'
                - 'ssm:GetDocument'
                - 'support:Describe*'
                - 'tag:GetTagKeys'
              Resource: '*'
            - Effect: Allow
              Action:
                - 'apigateway:GET'
              Resource: 'arn:aws:apigateway:*::/restapis/*'
            - Effect: Allow
              Action:
                - 'securityhub:BatchImportFindings'
                - 'securityhub:GetFindings'
              Resource: '*'
```



ole을 생성

owlerRole",
owlerRole",
owlerRole",
owlerRole",
owlerRole"

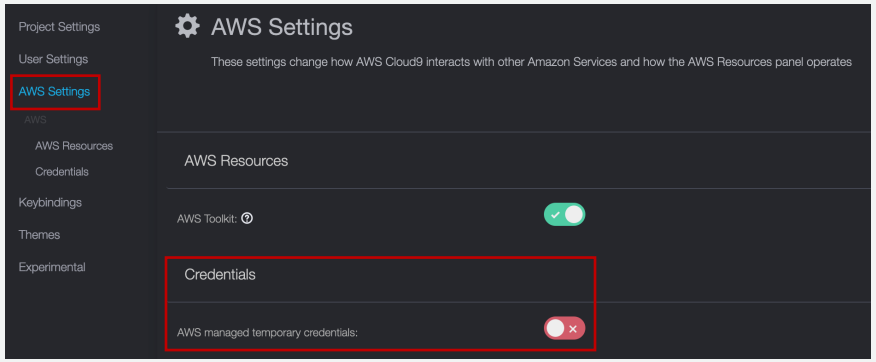
상 Account ID

Scanning to EC2

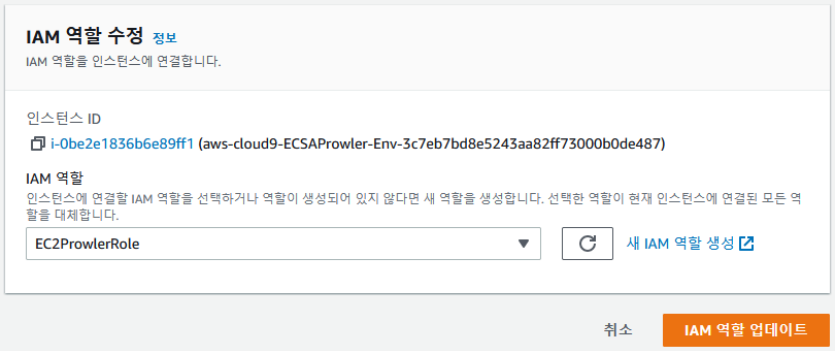
Scanning to EC2

Version 2 - 점검 환경 생성 및 점검 실행

1. Cloud9 생성 및 임시 자격증명 비활성화



2. EC2 > Cloud9 인스턴스 우클릭 > 보안 > IAM 역할 수정 > 생성한 IAM Role로 수정

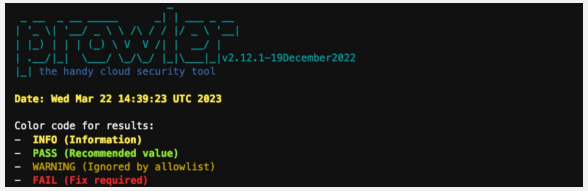


3. Cloud9 접속 후, 아래 Command로 Prowler v2.12.1 설치

```
sudo yum install -y python3 jq git
sudo pip3 install detect-secrets==1.0.3
sudo curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo unzip awscliv2.zip
sudo ./aws/install -b /bin --update
sudo git clone https://github.com/prowler-cloud/prowler
cd prowler
sudo git checkout 2.12.1
```

4. 점검 수행

```
sudo ./prowler -A {점검 대상 Account ID} –R {점검 대상 IAM Role 이름} -M {결과 형식 지정} &
```



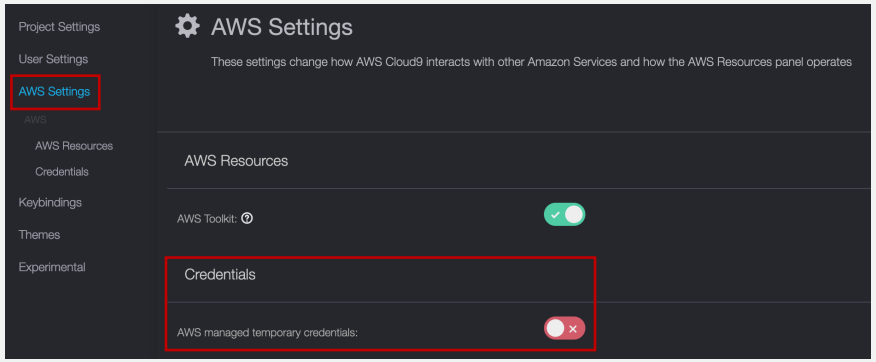
주요 옵션

-A : 점검 대상 Account ID	-B : S3 Bucket 저장
-R : 사용할 Role	-I : 점검 항목 출력
-M : 출력 결과 형식	-f : 특정 리전 선택
-i : Inventory 스캔	-S : Security Hub에 결과 전송

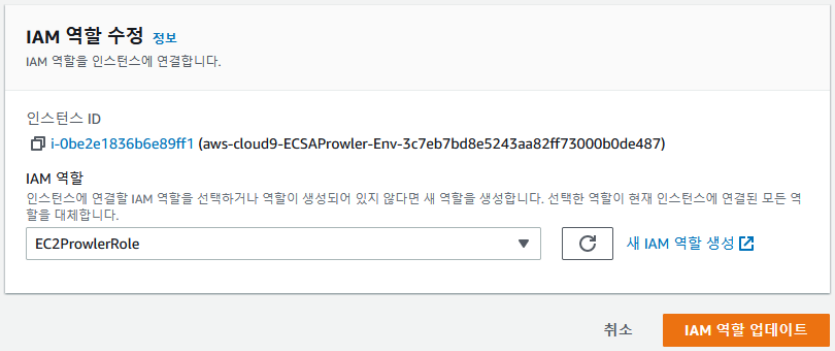
Scanning to EC2

Version 3 - 점검 환경 생성 및 점검 실행

1. Cloud9 생성 및 임시 자격증명 비활성화



2. EC2 > Cloud9 인스턴스 우클릭 > 보안 > IAM 역할 수정 > 생성한 IAM Role로 수정

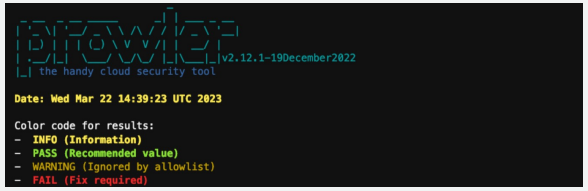


3. Cloud9 접속 후, 아래 Command로 Prowler v3 설치

```
sudo yum install -y gcc openssl-devel bzip2-devel libffi-devel
wget https://www.python.org/ftp/python/3.9.16/Python-3.9.16.tgz
tar xf Python-3.9.16.tgz
cd Python-3.9.16
./configure --enable-optimizations
sudo make altinstall
pip3.9 install prowler
```

4. 점검 수행

```
prowler –R arn:aws:iam::<점검 대상 Account ID>:role/< 점검 대상 IAM Role 이름 > -M {결과 형식 지정}
```

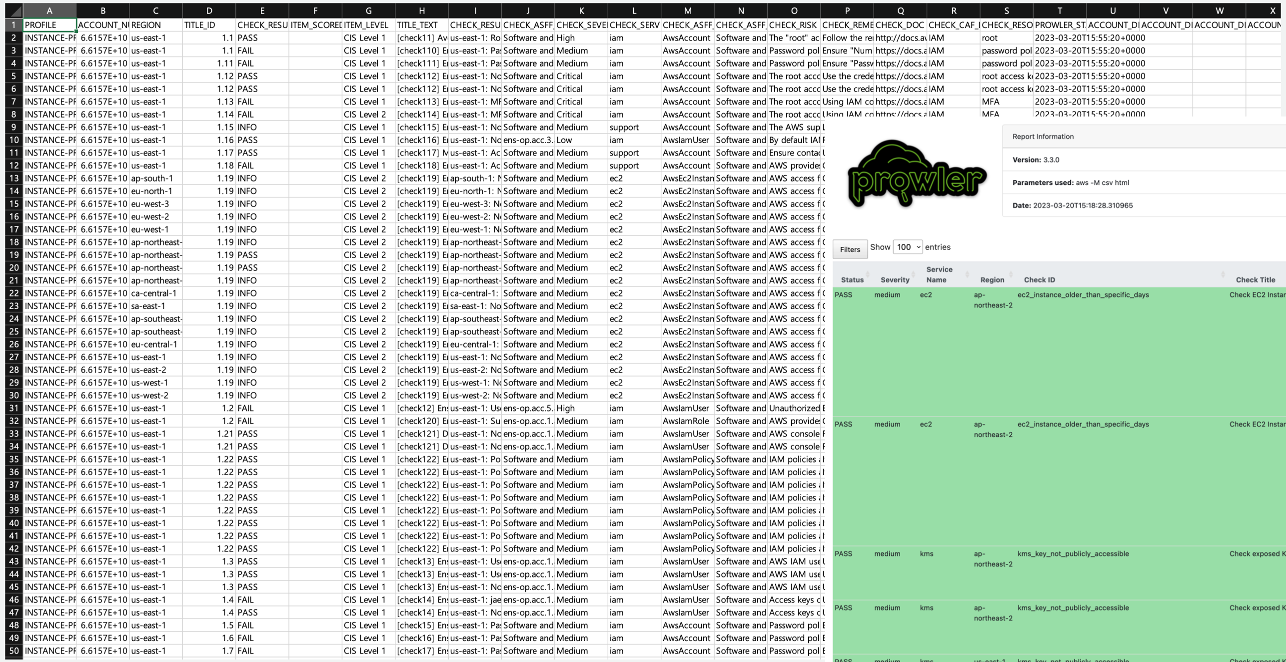


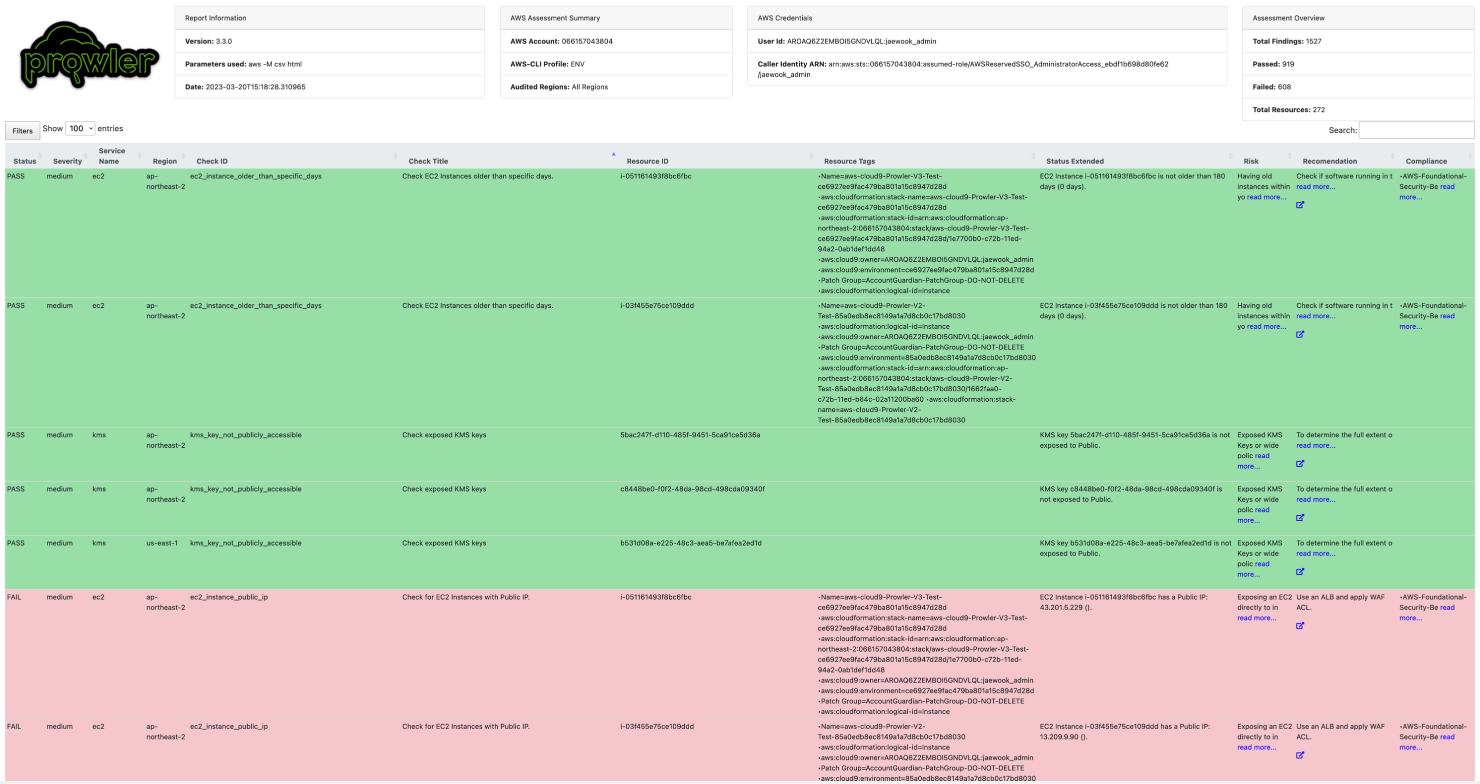
주요 옵션

-R : 점검 대상 Role	-I : 점검 항목 출력
-M : 출력 결과 형식	-f : 특정 리전 선택
-i : Inventory 스캔	-S : Security Hub에 결과 전송
-B : S3 Bucket 저장	

Scanning to EC2

점검 결과물 샘플 (csv, html)





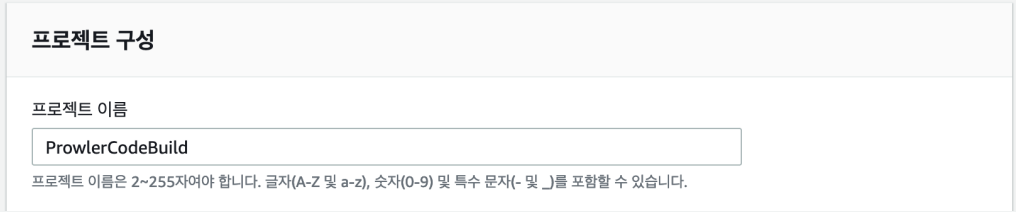
Scanning to CodeBuild

Scanning to CodeBuild

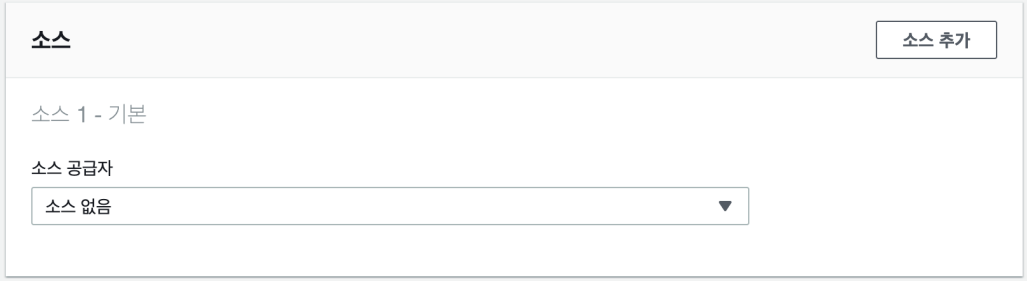
점검 환경 생성 및 점검 실행

1. CodeBuild > 빌드 > 프로젝트 빌드 > 빌드 프로젝트 생성 > 프로젝트 이름 입력

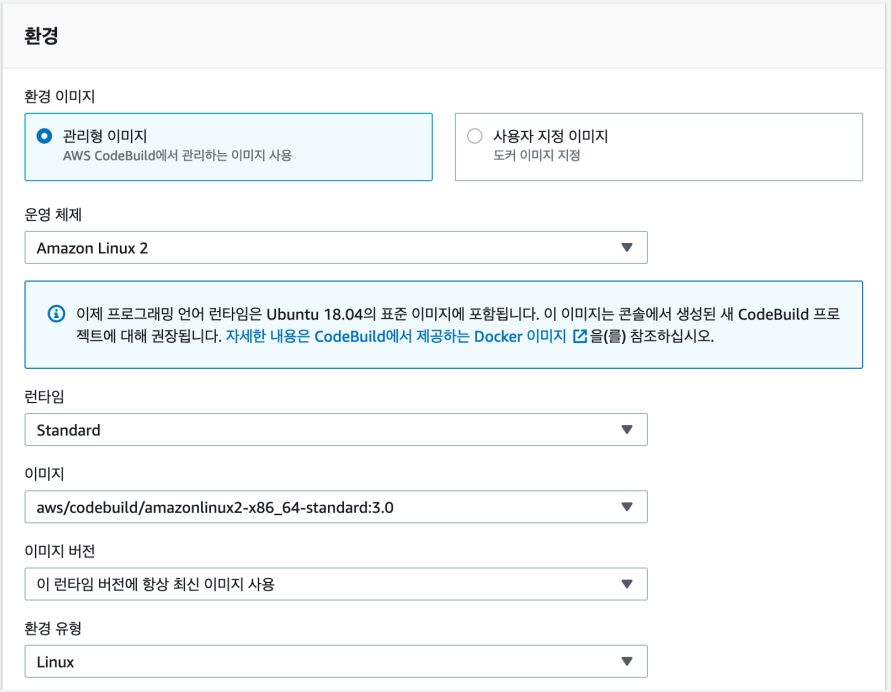
빌드 프로젝트 생성



2. 소스 공급자 > 소스 없음



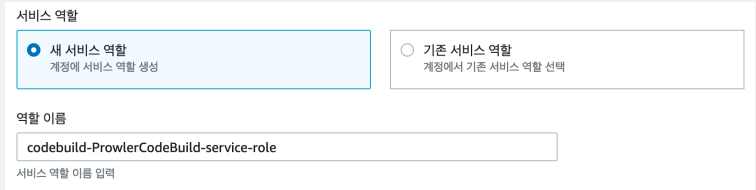
3. 환경 설정
운영 체제: Amazon Linux 2
런타임: Standard
이미지: aws/codebuild/amazonlinux2-x86_64-standard:3.0
이미지 버전: 이 런타임 버전에 항상 최신 이미지 사용
환경 유형: Linux



Scanning to CodeBuild

점검 환경 생성 및 점검 실행

4. 서비스 역할 > 새 서비스 역할 > 역할 수정 필수

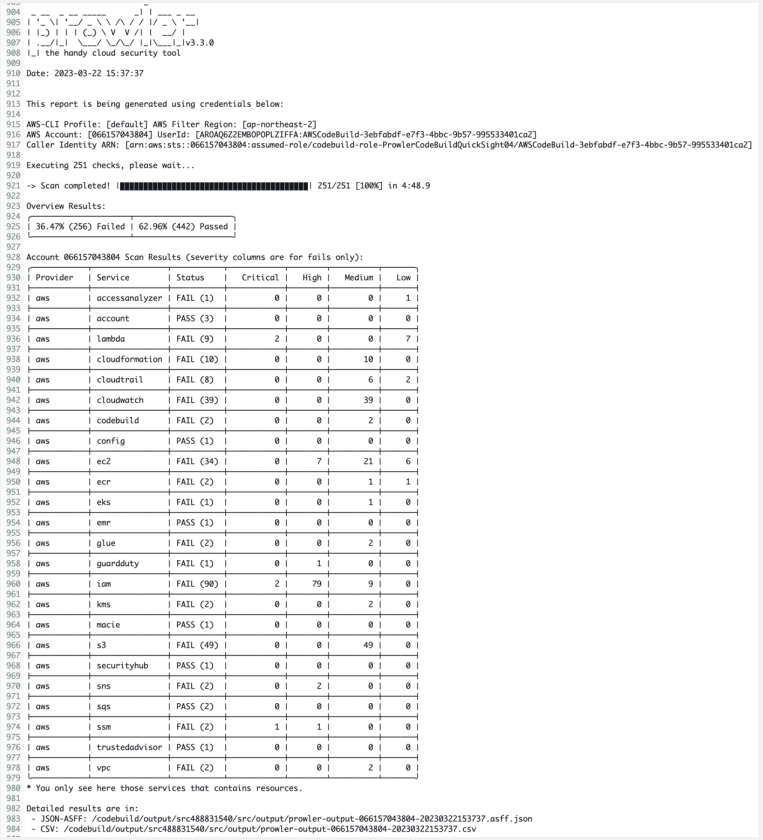


5. 추가 구성 > 환경 변수 추가
이름: $PROWLER_OPTIONS
값: -M csv, json-asff

6. 아래의 buildspec 작성

```
version: 0.2
phases:
  install:
    runtime-versions:
      python: 3.9
    commands:
      - echo "Installing Prowler and dependencies..."
      - curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
      - unzip awscliv2.zip
      - ./aws/install -b /bin --update
      - pip install prowler
  build:
    commands:
      - echo "Running Prowler as prowler aws $PROWLER_OPTIONS"
      - prowler aws $PROWLER_OPTIONS
```

7. 빌드 시작 > 빌드 로그 > 테일 로그에서 점검 상태 확인

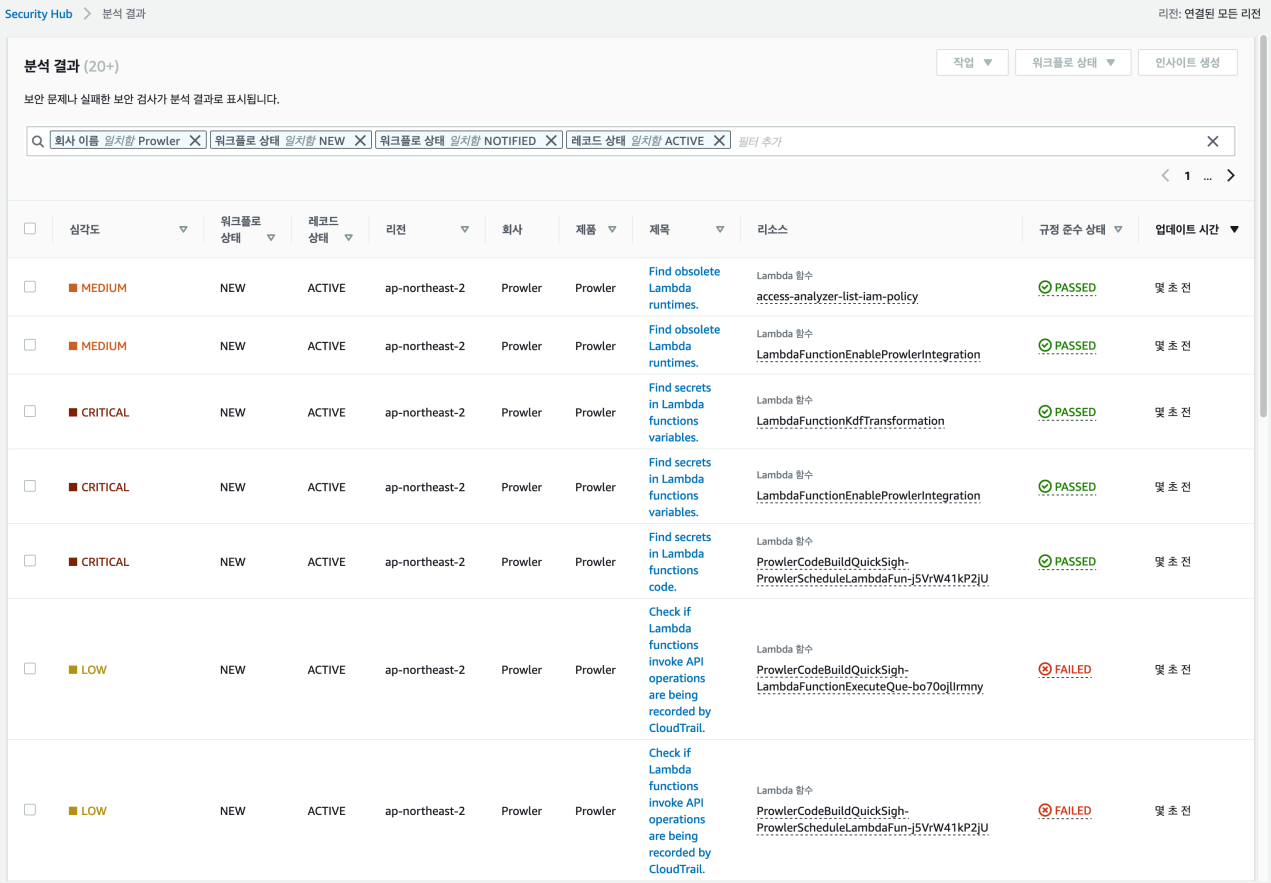


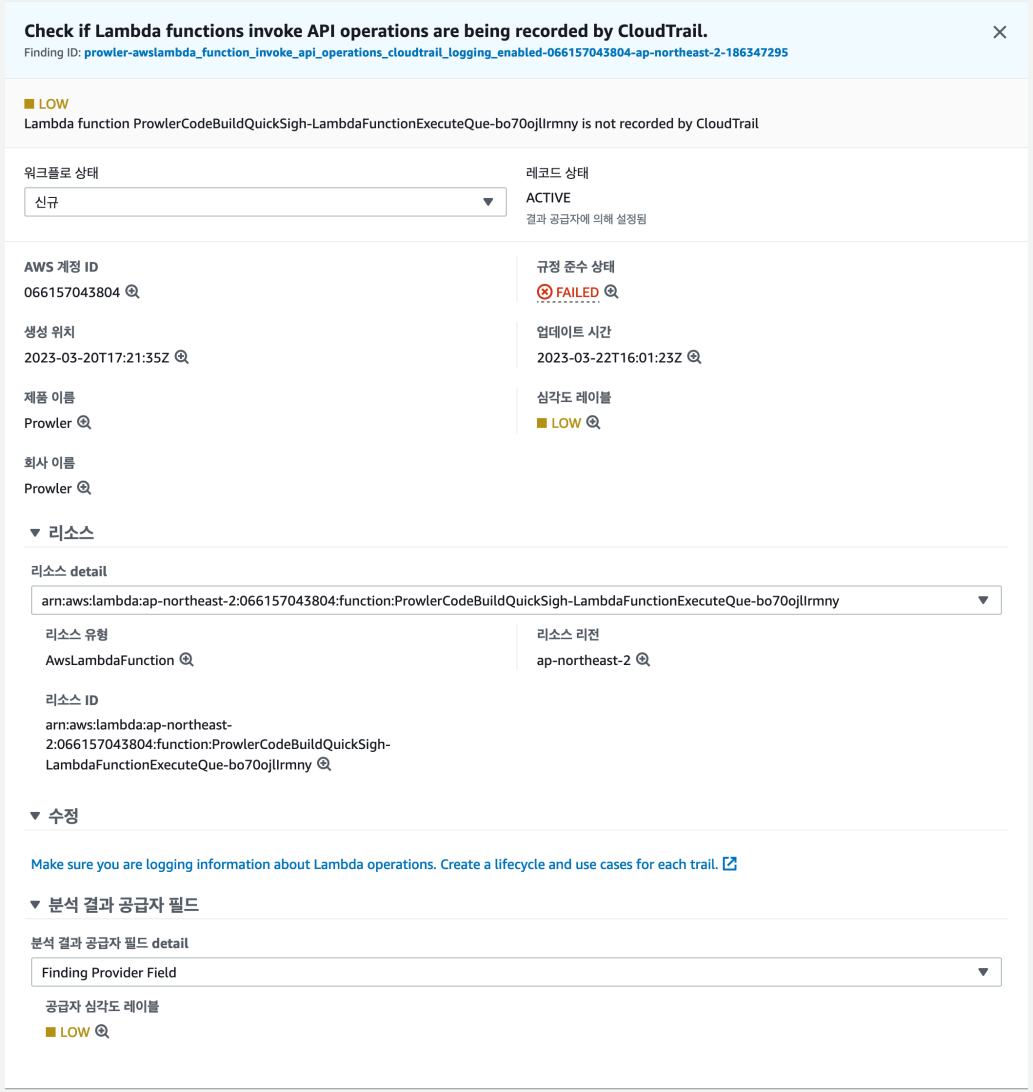
```
 _ __  _ __ _____      _| | ___ _ __
| '_ \| '__/ _ \ \ /\ / / |/ _ \ '__|
| |_) | | | (_) \ V  V /| |  __/ |
| .__/|_|  \___/ \_/\_/ |_|\___|_|v3.3.0
|_| the handy cloud security tool

Date: 2023-03-22 15:37:37


This report is being generated using credentials below:

AWS-CLI Profile: [default] AWS Filter Region: [ap-northeast-2]
AWS Account: [066157043804] UserId: [AROAQ6Z2EMBOPOPLZIFFA:AWSCodeBuild-3ebfabdf-e7f3-4bbc-9b57-995533401ca2]
Caller Identity ARN: [arn:aws:sts::066157043804:assumed-role/codebuild-role-ProwlerCodeBuildQuickSight04/AWSCodeBuild-3ebfabdf-e7f3-4bbc-9b57-995533401ca2]

Executing 251 checks, please wait...

-> Scan completed! |████████████████████████████████████| 251/251 [100%] in 4:48.9

Overview Results:
┌──────────────────────┬──────────────────────┐
| 36.47% (256) Failed | 62.96% (442) Passed |
└──────────────────────┴──────────────────────┘

Account 066157043804 Scan Results (severity columns are for fails only):
┌──────────┬──────────────┬──────────┬──────────┬──────┬────────┬─────┐
| Provider | Service      | Status   | Critical | High | Medium | Low |
├──────────┼──────────────┼──────────┼──────────┼──────┼────────┼─────┤
| aws      | accessanalyzer | FAIL (1) |        0 |    0 |      0 |   1 |
| aws      | account      | PASS (3) |        0 |    0 |      0 |   0 |
| aws      | lambda       | FAIL (9) |        2 |    0 |      0 |   7 |
| aws      | cloudformation | FAIL (10) |        0 |    0 |     10 |   0 |
| aws      | cloudtrail   | FAIL (8) |        0 |    0 |      6 |   2 |
| aws      | cloudwatch   | FAIL (39) |        0 |    0 |     39 |   0 |
| aws      | codebuild    | FAIL (2) |        0 |    0 |      2 |   0 |
| aws      | config       | PASS (1) |        0 |    0 |      0 |   0 |
| aws      | ec2          | FAIL (34) |        0 |    7 |     21 |   6 |
| aws      | ecr          | FAIL (2) |        0 |    0 |      1 |   1 |
| aws      | eks          | FAIL (1) |        0 |    0 |      1 |   0 |
| aws      | emr          | PASS (1) |        0 |    0 |      0 |   0 |
| aws      | glue         | FAIL (2) |        0 |    0 |      2 |   0 |
| aws      | guardduty    | FAIL (1) |        0 |    1 |      0 |   0 |
| aws      | iam          | FAIL (90) |        2 |   79 |      9 |   0 |
| aws      | kms          | FAIL (2) |        0 |    0 |      2 |   0 |
| aws      | macie        | PASS (1) |        0 |    0 |      0 |   0 |
| aws      | s3           | FAIL (49) |        0 |    0 |     49 |   0 |
| aws      | securityhub  | PASS (1) |        0 |    0 |      0 |   0 |
| aws      | sns          | FAIL (2) |        0 |    2 |      0 |   0 |
| aws      | sqs          | PASS (2) |        0 |    0 |      0 |   0 |
| aws      | ssm          | FAIL (2) |        1 |    1 |      0 |   0 |
| aws      | trustedadvisor | PASS (1) |        0 |    0 |      0 |   0 |
| aws      | vpc          | FAIL (2) |        0 |    0 |      2 |   0 |
└──────────┴──────────────┴──────────┴──────────┴──────┴────────┴─────┘
* You only see here those services that contains resources.

Detailed results are in:
 - JSON-ASFF: /codebuild/output/src488831540/src/output/prowler-output-066157043804-20230322153737.asff.json
 - CSV: /codebuild/output/src488831540/src/output/prowler-output-066157043804-20230322153737.csv
```

Visualization

Visualization

Security Hub

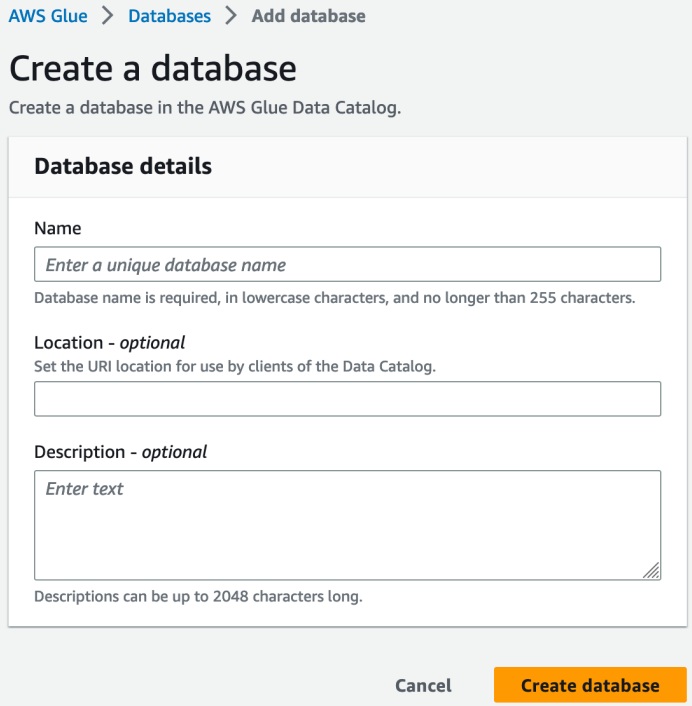




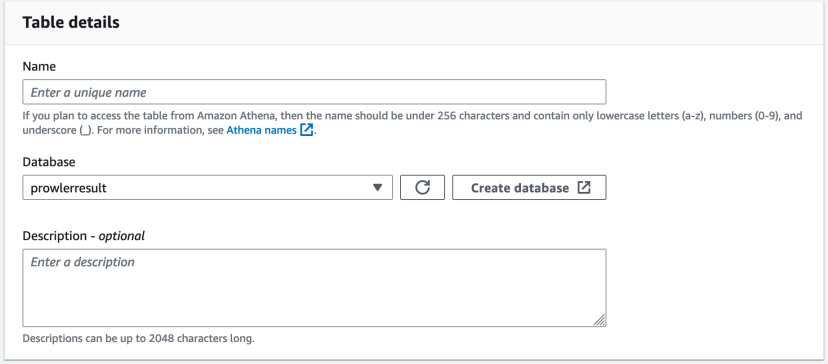
Visualization

Athena Query – Data Catalog 생성하기 01

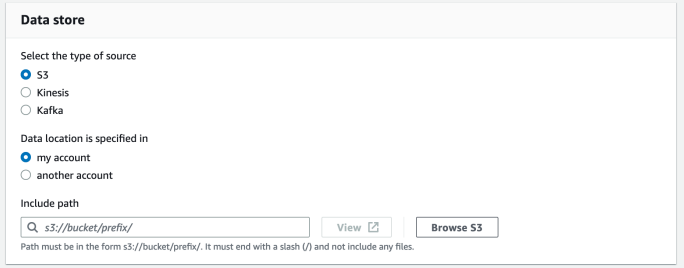
1. Glue > Data Catalog > Databases < Create database > database 이름 입력



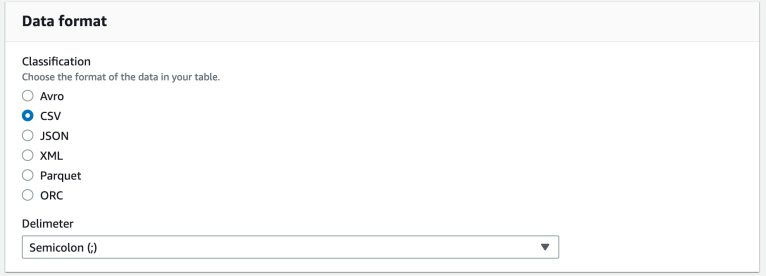
2. Glue > Data Catalog > Tables > Table 이름 입력



3. Data store: S3 선택 > 결과가 저장된 S3 선택



4. Data format: CSV > Delimeter: Semicolon(;) 선택)

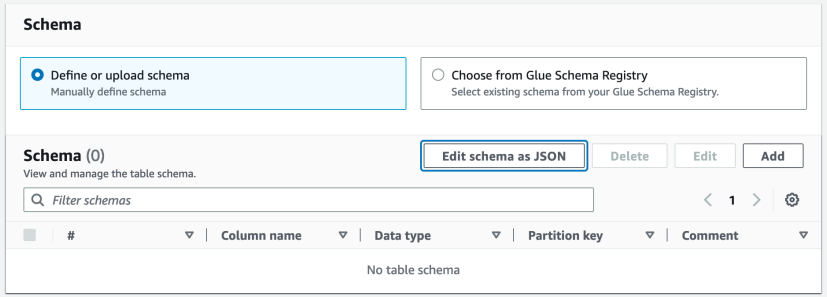


Visualization

Athena Query – Data Catalog 생성하기 02

5. Edit schema as JSON 선택

Choose or define schema



6. 아래 내용을 복사하고 Table 생성

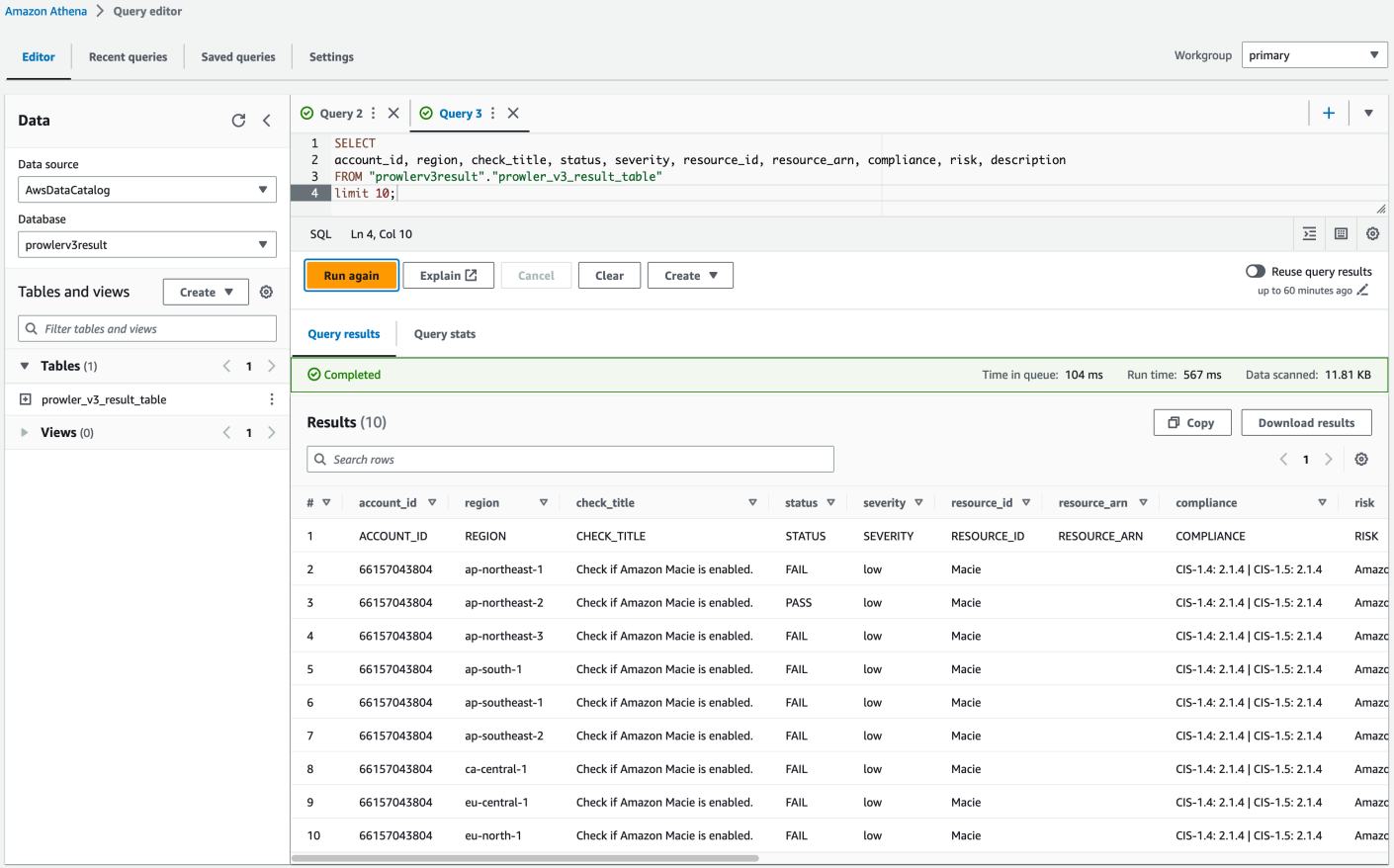
```
[
  {
    "Name": "assessment_start_time",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "finding_unique_id",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "provider",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "check_id",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "check_title",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "check_type",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "status",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "status_extended",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "service_name",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "subservice_name",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "severity",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "resource_type",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "resource_details",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "resource_tags",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "description",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "risk",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "related_url",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "remediation_recommendation_text",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "remediation_recommendation_url",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "remediation_recommendation_code_nativeiac",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "remediation_recommendation_code_terraform",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "remediation_recommendation_code_cli",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "remediation_recommendation_code_other",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "compliance",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "categories",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "depends_on",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "related_to",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "notes",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "profile",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "account_id",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "account_name",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "account_email",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "account_arn",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "account_org",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "account_tags",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "region",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "resource_id",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "resource_arn",
    "Type": "string",
    "Comment": ""
  }
]
```

Visualization

Athena Query – 조회

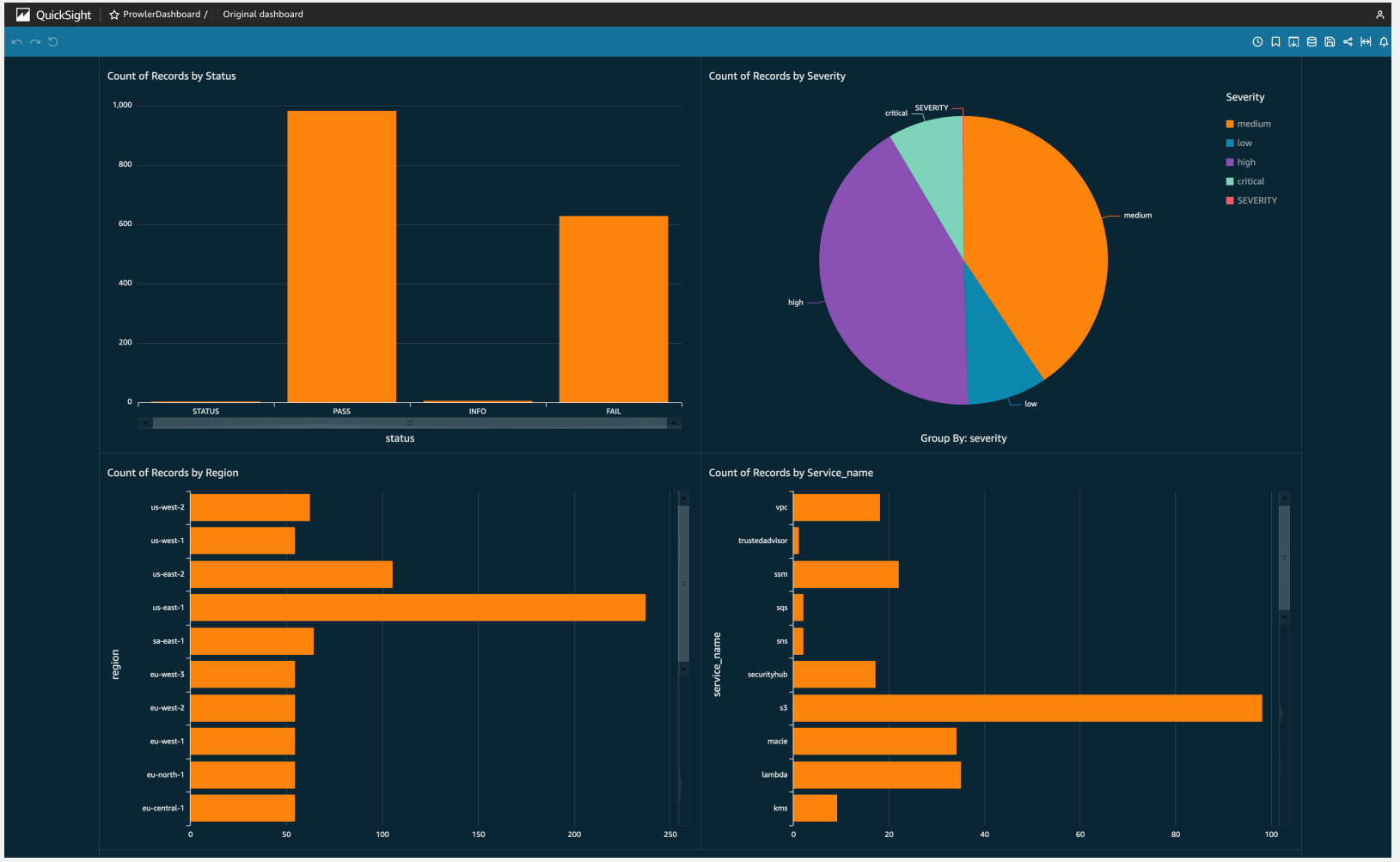
7. Athena에서 다음과 같이 필요에 따라 Query문을 수정하여 결과를 조회

```
SELECT
account_id, region, check_title, status, severity, resource_id,
resource_arn, compliance, risk, description
FROM ”{Database 이름}"."{Table이름}”
limit 10;
```



Visualization

QuickSight Dashboard



Thank you!