Cloud-native Cross-team SDLC automation PoC

A very simple demonstration

Somnath Mukherjee, 2025

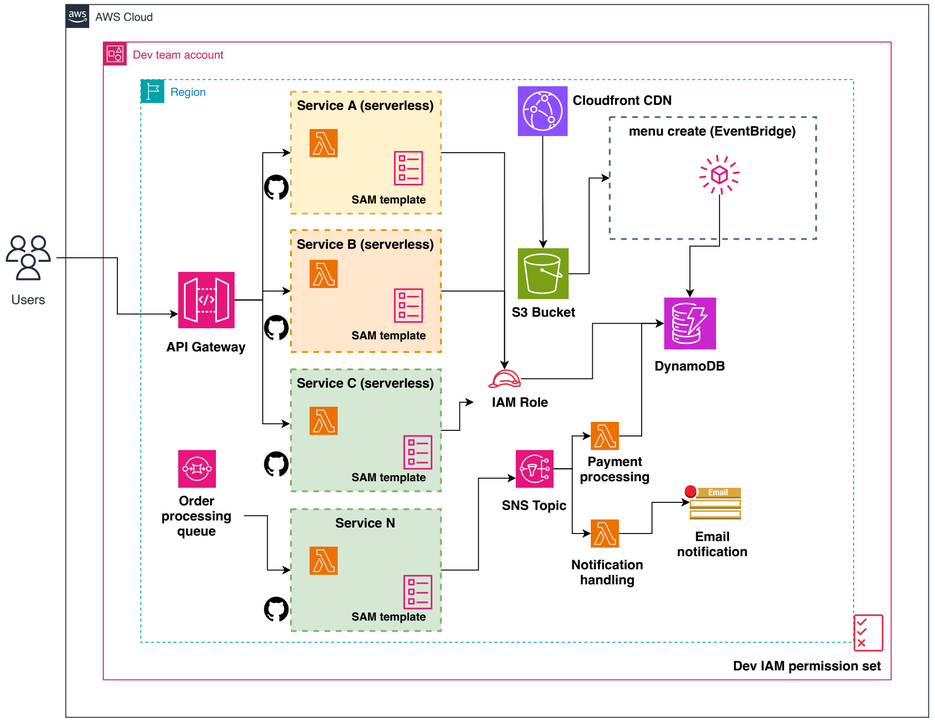
That implements:

- Cross-team ops using modern SDLC automation
- Full team autonomy
- Single responsibility across teams
- Centralised code governance & compliance
- Centralised proactive threat mitigation measures
- GitOps style

Example scenario:

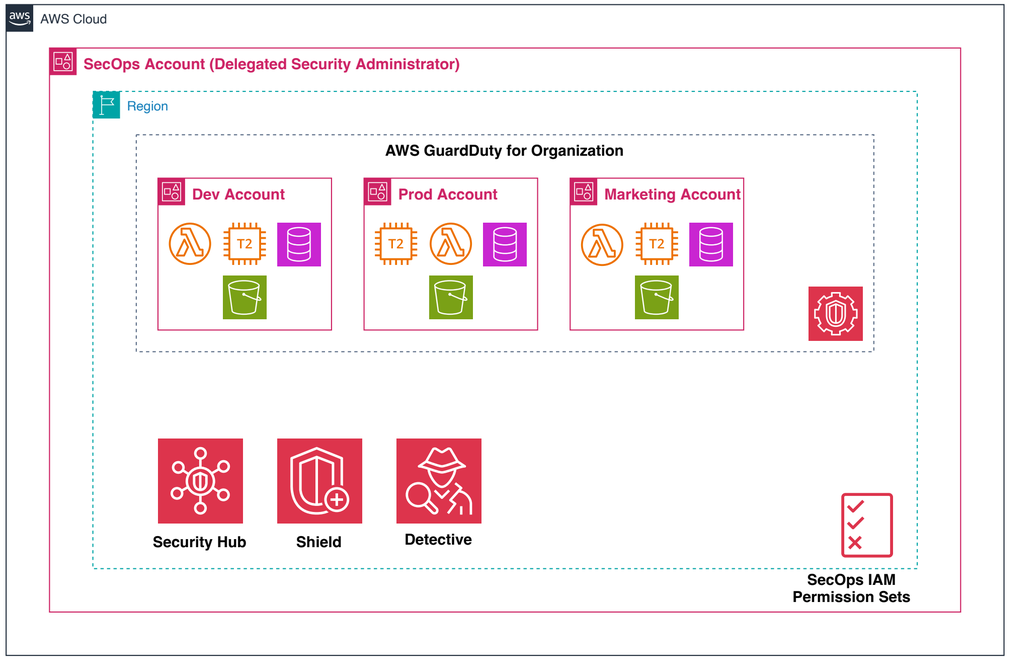
A Typical Online Food Ordering system

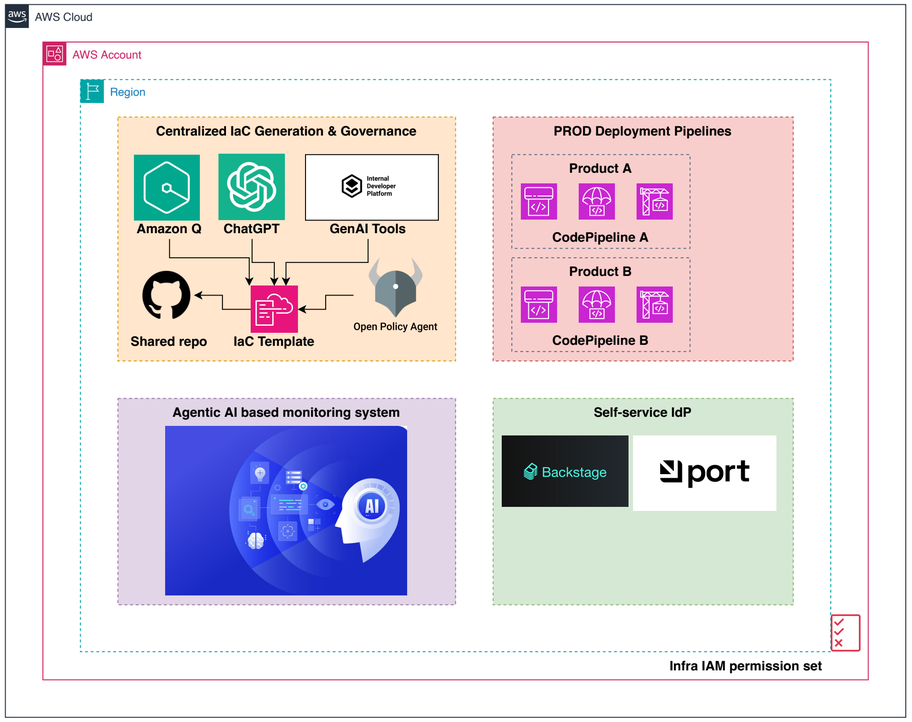
Gus Fring wants to build a **Breaking Bad** style order fulfilment system using cloud-native SDLC automation with several teams



Dev team

SecOps team

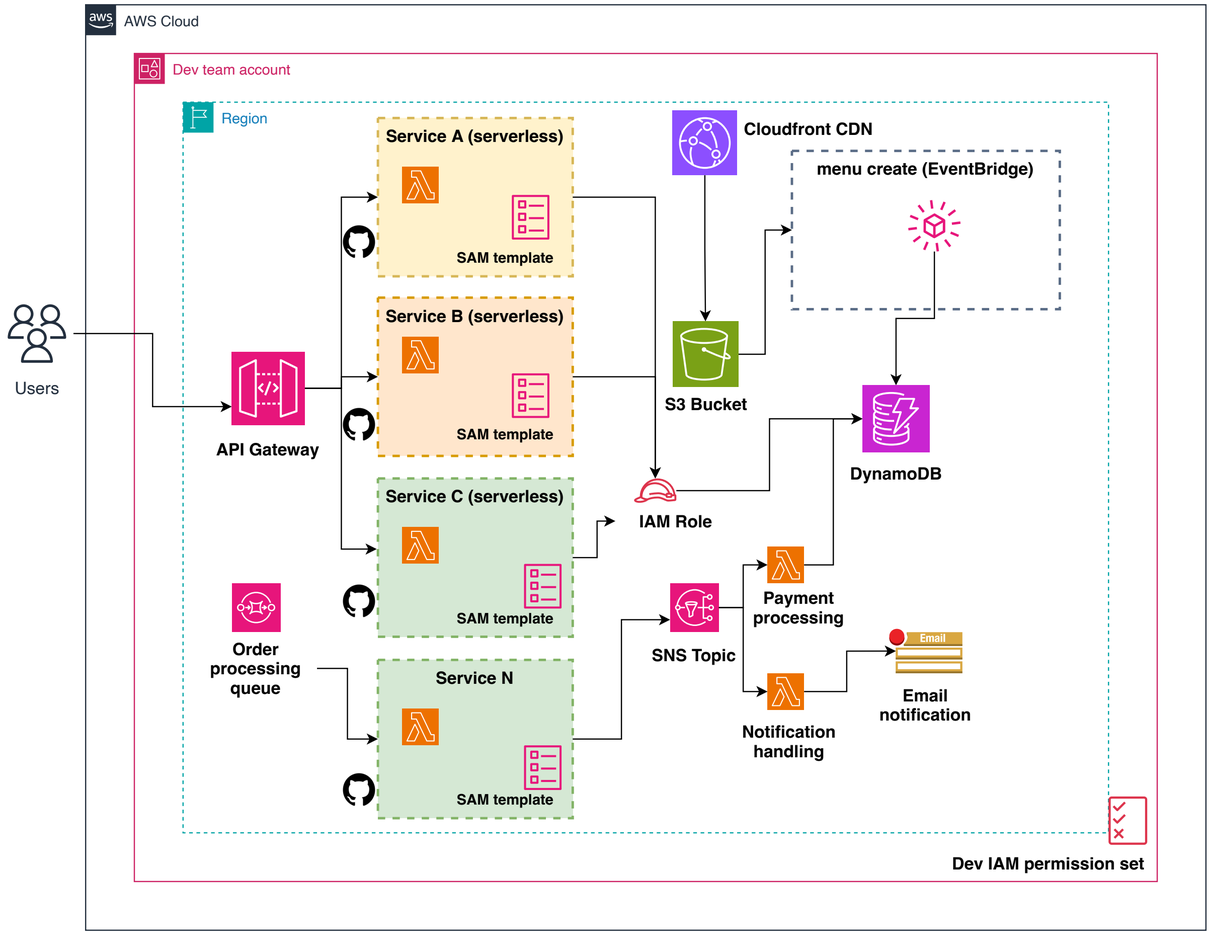




Infra / Platform team

Dev team

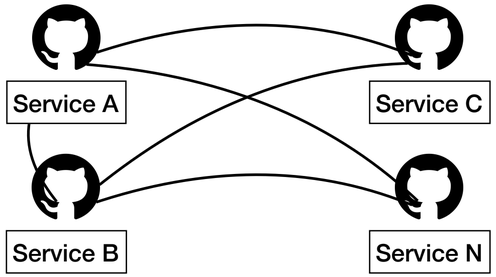
- Scales to thousands of microservices, per repo
- Each service supports environment promotion with manual approval (Dev -> Test -> Staging -> Prod)
- Each service with independent pipelines using pipelines-as-code templates
- Idempotent service deployment using IaC templates collaborated by Infra team
- Full autonomy: Developers have freedom to build & deploy upto a certain point
- Least privilege principle: only granted permissions that are required, using IAM Permission Sets

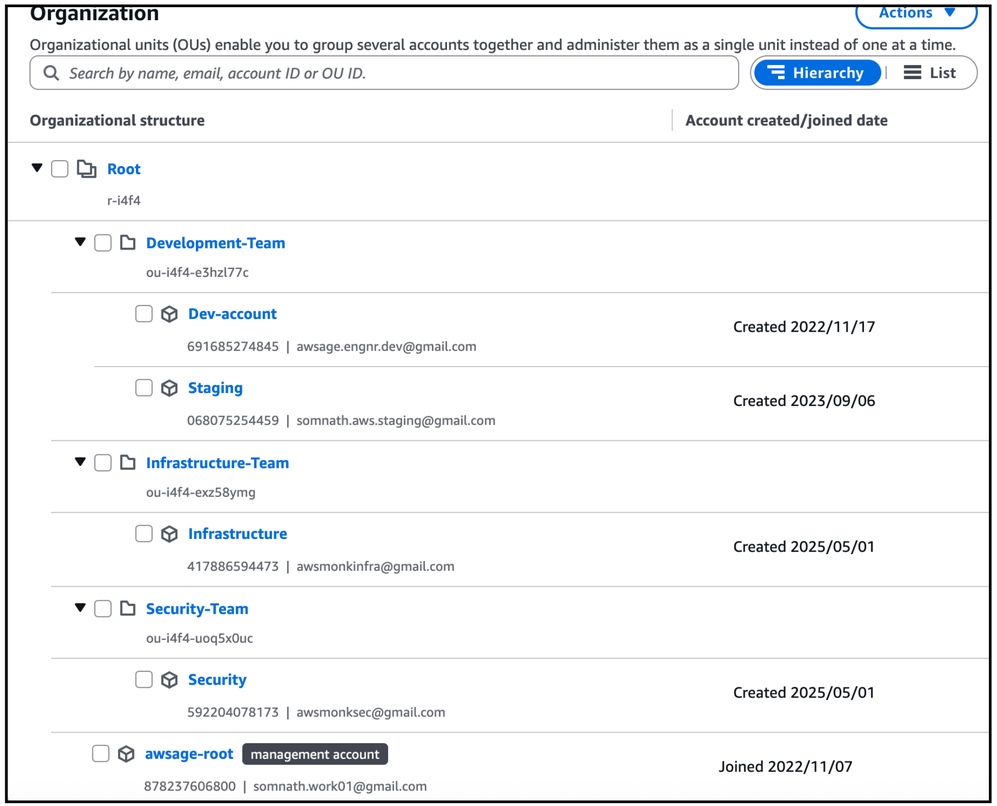


Org & repo structure

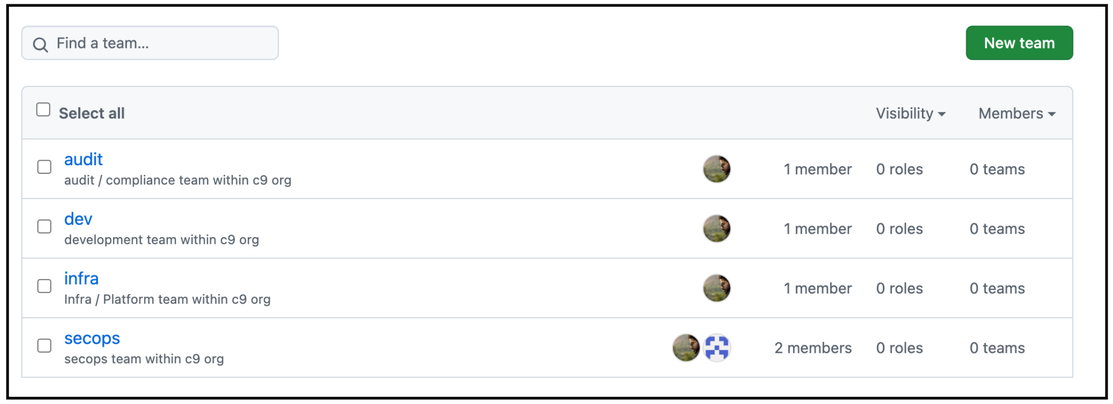
.github/workflows	updated sam deployment role	yesterday
events	added moe test	last week
src	[menu_get_svc] code 400 message modified	2 days ago
tests	[menu_get_svc] lambda test event path typo 2	2 days ago
.gitignore	Initial commit	last week
README.md	[menu_get_svc] added CI build passing badge	2 days ago
template.yaml	updated sam template v2	yesterday

A Sample microservice (python)





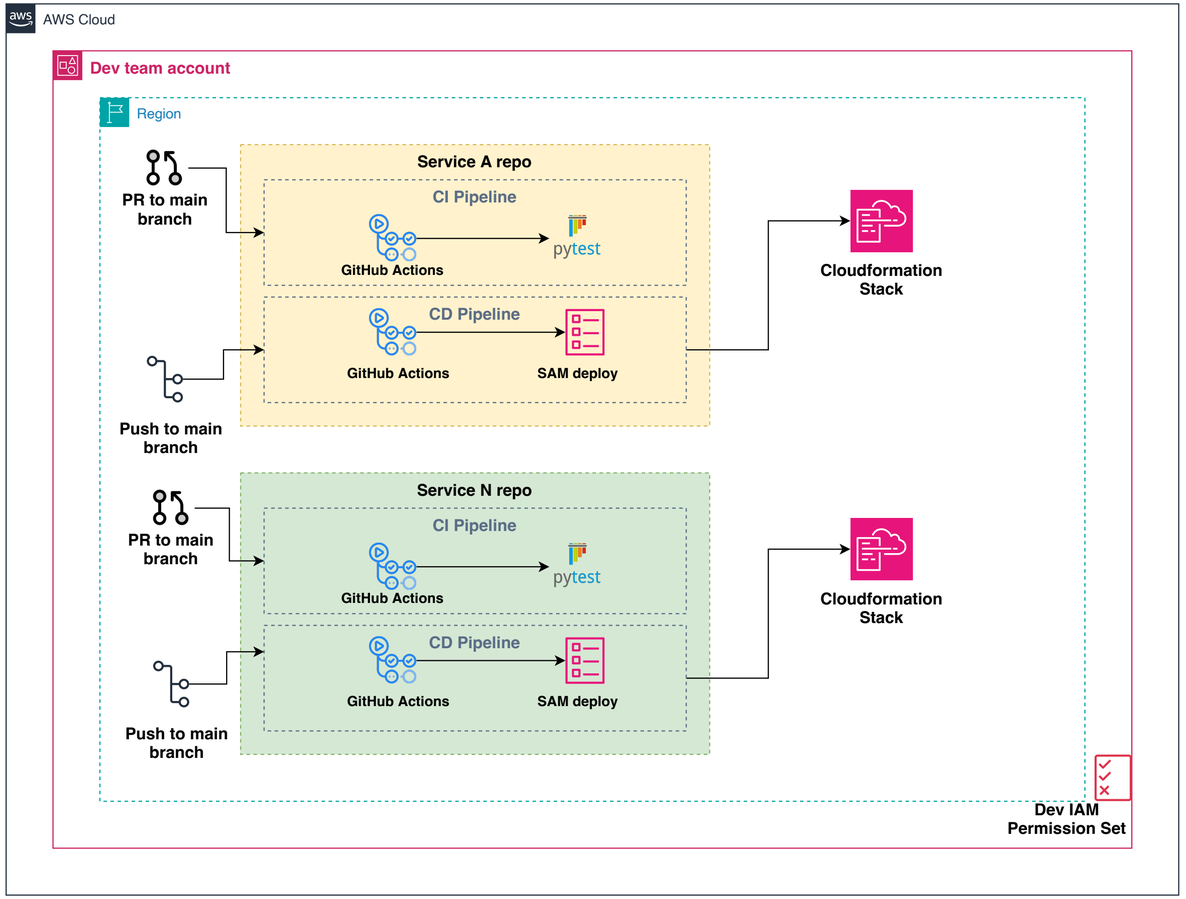
Team setup in AWS



Team setup in GitHub

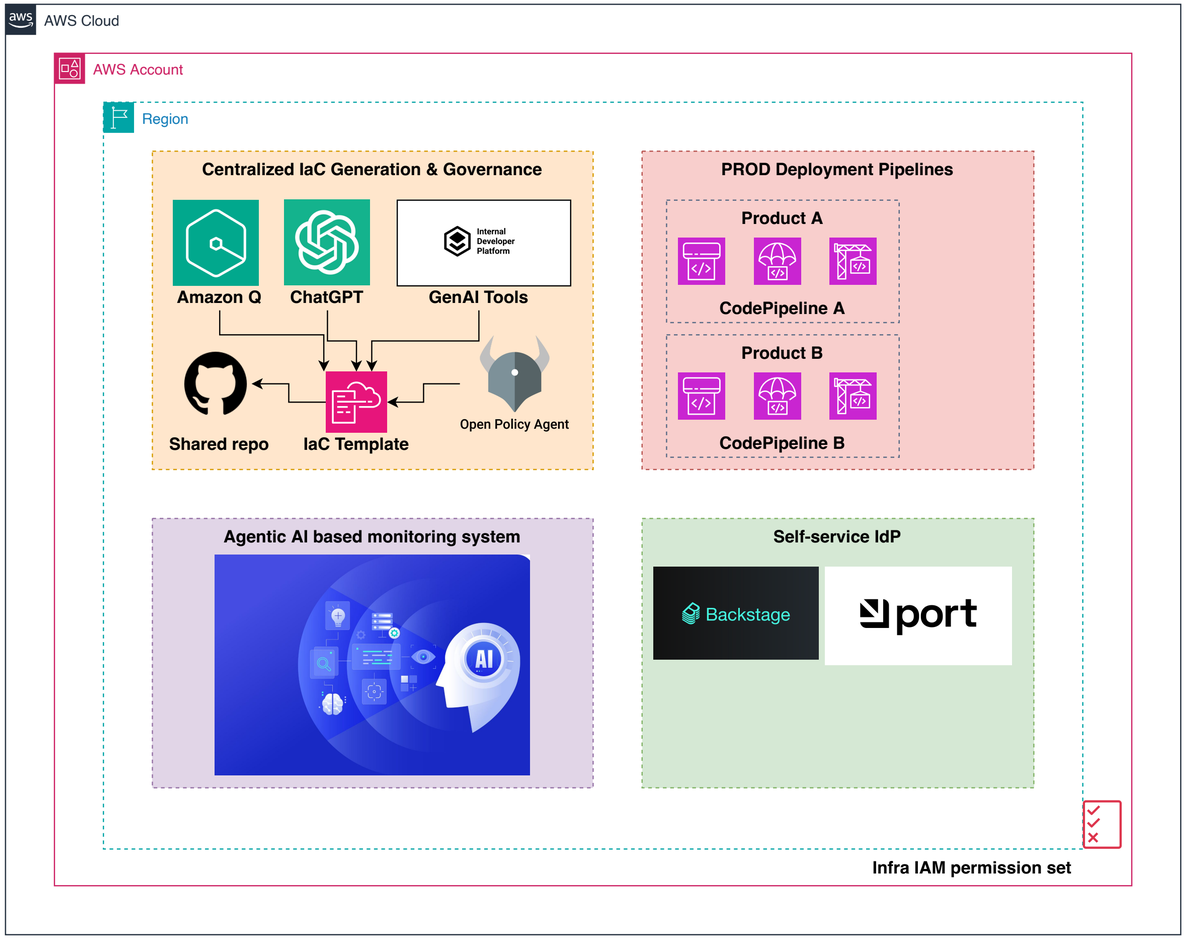
Dev team's pipeline

- Dev team fully owns a service's creation, development & deployment lifecycle
- Dev team can self-service it's operation through a portal
- Each service is a repo to minimise deployment friction
- GitOps is the source of truth
- Deployment is made with IaC templates written by Infra team



Infra team

- Easier service deployment using standard IaC templates, generated by GenAI (e.g. ChatGPT)
- Generated IaC templates are validated using Policy-as-Code tools such as Checkov or OPA before deployment. This enforces compliance & governance
- Privileged access: to Prod environment pipelines
- Provides a self-service IdP portal for all teams to quickly do things without repetition
- Implements an Agentic AI based monitoring system that learns workloads and it’s deployment pattern to optimise cost



Infra team's resource

A CLI tool to generate IaC template using ChatGPT like prompts

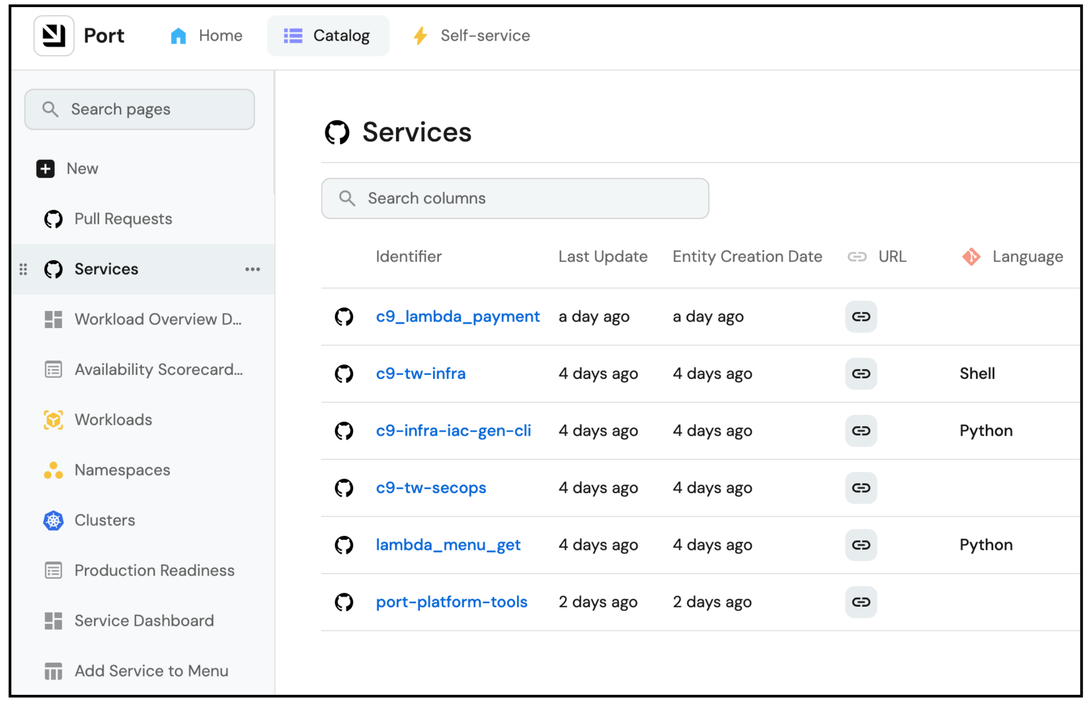
```
→  src git:(main) python3 app.py "generate a simple lambda function SAM template"
```

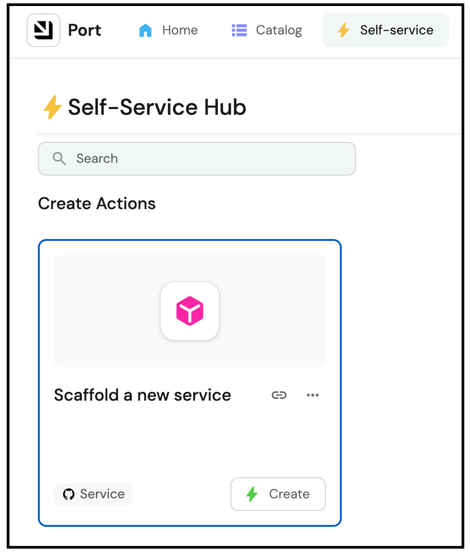
```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  SimpleFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: nodejs18.x
      CodeUri: s3://your-bucket/your-function-code.zip
      MemorySize: 128
      Timeout: 3
```

Generated SAM template

```
By Prisma Cloud | version: 3.2.436

cloudformation scan results:

Passed checks: 3, Failed checks: 3, Skipped checks: 0

Check: CKV_AWS_173: "Check encryption settings for Lambda environment variable"
        PASSED for resource: AWS::Serverless::Function.SimpleFunction
        File: /cloudformation.yaml:4-11
Check: CKV_AWS_363: "Ensure Lambda Runtime is not deprecated"
        PASSED for resource: AWS::Serverless::Function.SimpleFunction
        File: /cloudformation.yaml:4-11
Check: CKV_AWS_45: "Ensure no hard-coded secrets exist in Lambda environment"
        PASSED for resource: AWS::Serverless::Function.SimpleFunction
        File: /cloudformation.yaml:4-11
Check: CKV_AWS_117: "Ensure that AWS Lambda function is configured inside a VPC"
        FAILED for resource: AWS::Serverless::Function.SimpleFunction
        File: /cloudformation.yaml:4-11

                4  |  SimpleFunction:
                5  |    Type: AWS::Serverless::Function
                6  |    Properties:
                7  |      Handler: index.handler
                8  |      Runtime: nodejs18.x
                9  |      CodeUri: s3://your-bucket/your-function-code.zip
                10 |      MemorySize: 128
                11 |      Timeout: 3
Check: CKV_AWS_116: "Ensure that AWS Lambda function is configured for a Dead Letter Queue(DLQ)"
```

Policy as code checking

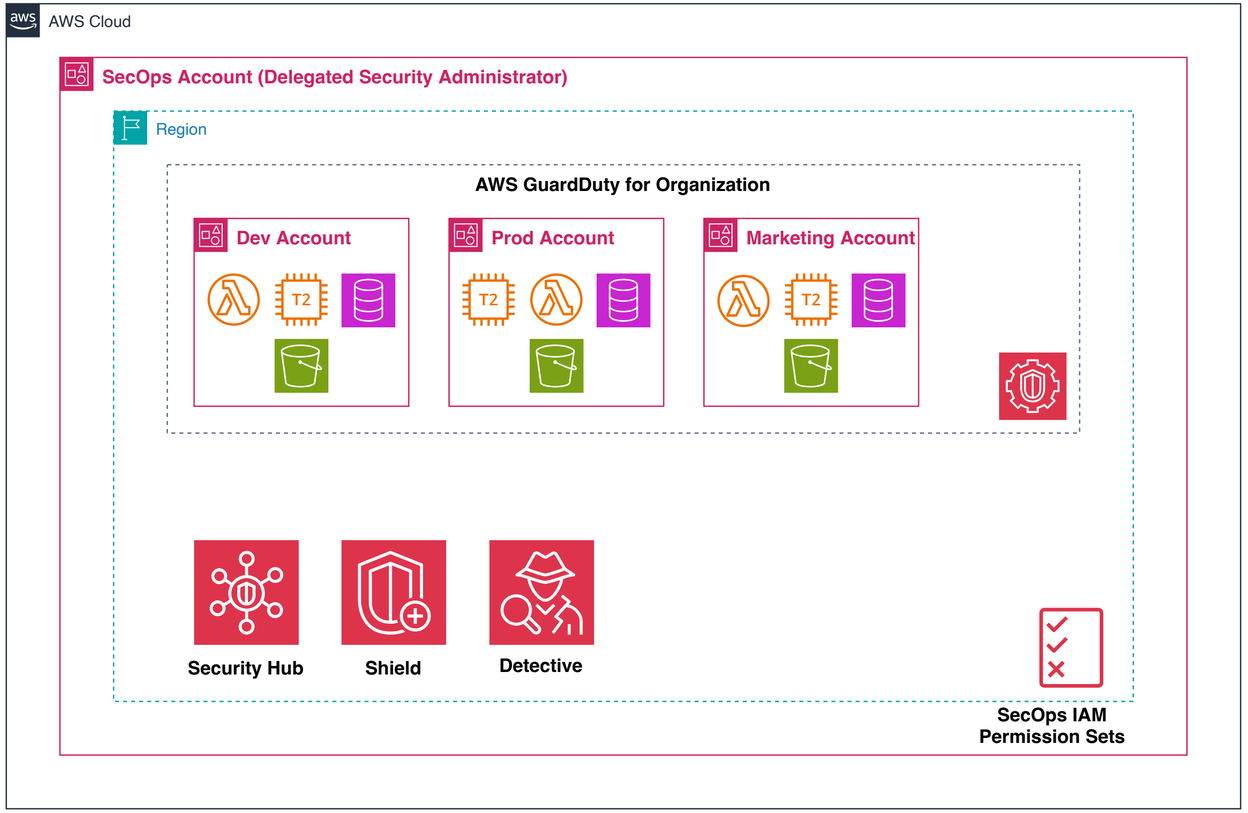




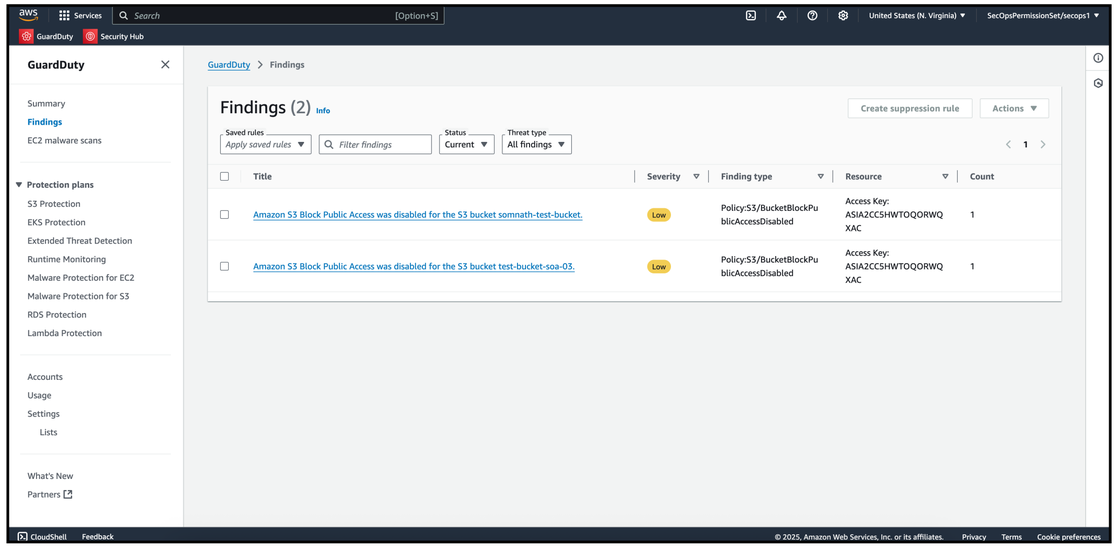
A Internal Developer Portal (IdP) using Port to self-service all team's servicing needs

SecOps team

- Acts as a delegated security administrator across Org
- Centrally scans all team's resources for potential vulnerabilities & threats
- Generate & share GuardDuty findings across teams
- Implement additional proactive ITSIEM tools



SecOps team's findings & remedies



Amazon GuardDuty findings

```yaml
S3PublicReadProhibitedRule:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: s3-bucket-no-public-read
    Descrption: "Prohibit public-read on S3 buckets"
    Source:
      Owner: AWS
      SourceIdentifier: S3_BUCKET_PUBLIC_READ_PROHIBITED
    Scope:
      ComplianceResourceTypes:
      - AWS::S3::Bucket

S3PublicReadRemediation:
  Type: AWS::Config::RemediationConfiguration
  Properties:
    ConfigRuleName: !Ref S3PublicReadProhibitedRule
    TargetType: SSM_DOCUMENT
    TargetId: AWS-DisableS3BucketPublicReadWrite
    Automatic: true
    Parameters:
      AutomationAssumeRole:
        StaticValue:
          Values:
          - !GetAtt ConfigRemediationRole.Arn
      BucketName:
        ResourceValue:
          Value: RESOURCE_ID
    ResourceType: AWS::S3::Bucket
```

Amazon SSM auto-remediation

Thank you