

# **Cloud-native Cross-team SDLC automation PoC**

**A very simple demonstration**

**Somnath Mukherjee, 2025**

# That implements:

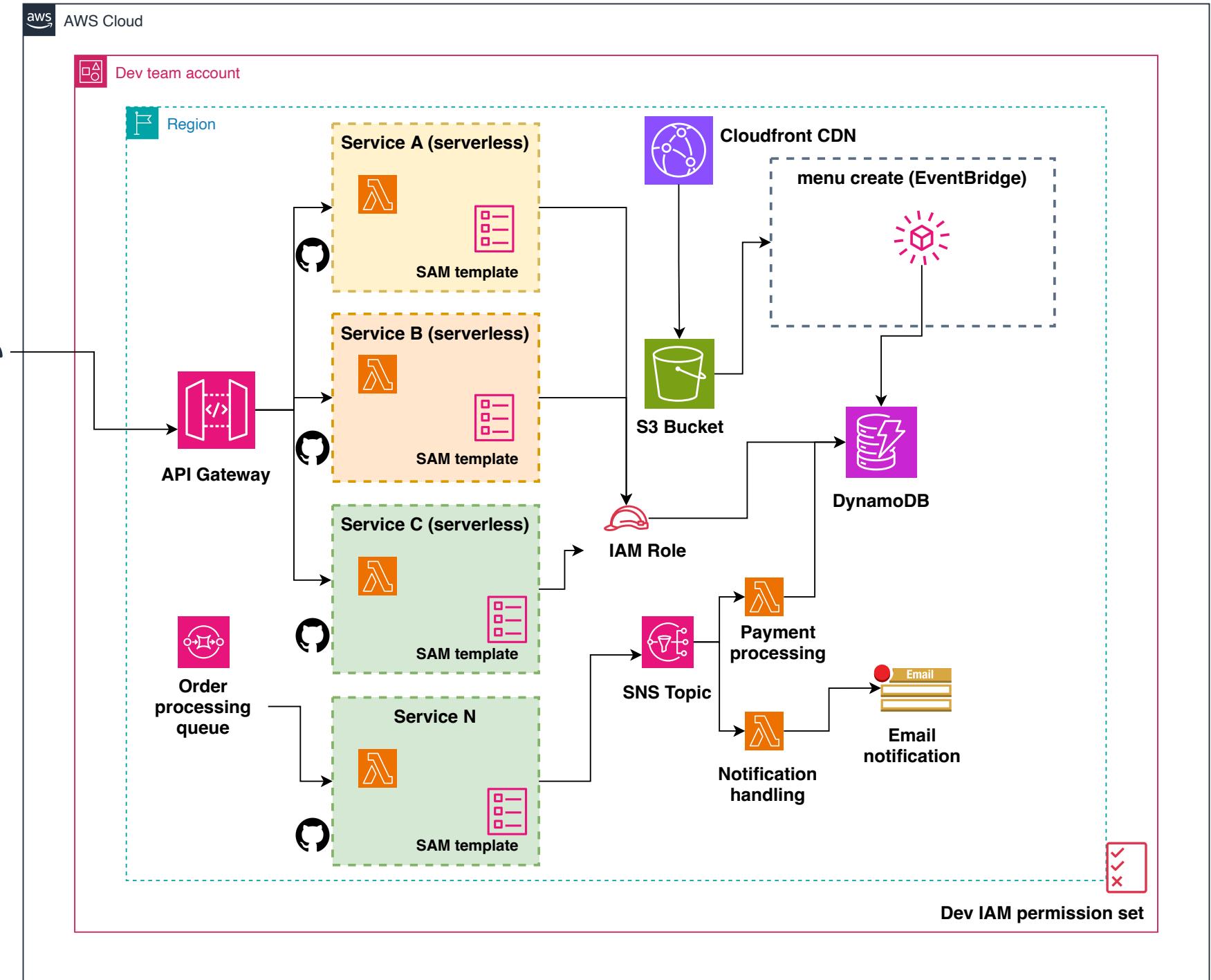
- **Cross-team ops using modern SDLC automation**
- **Full team autonomy**
- **Single responsibility across teams**
- **Centralised code governance & compliance**
- **Centralised proactive threat mitigation measures**
- **GitOps style**

# Example scenario:

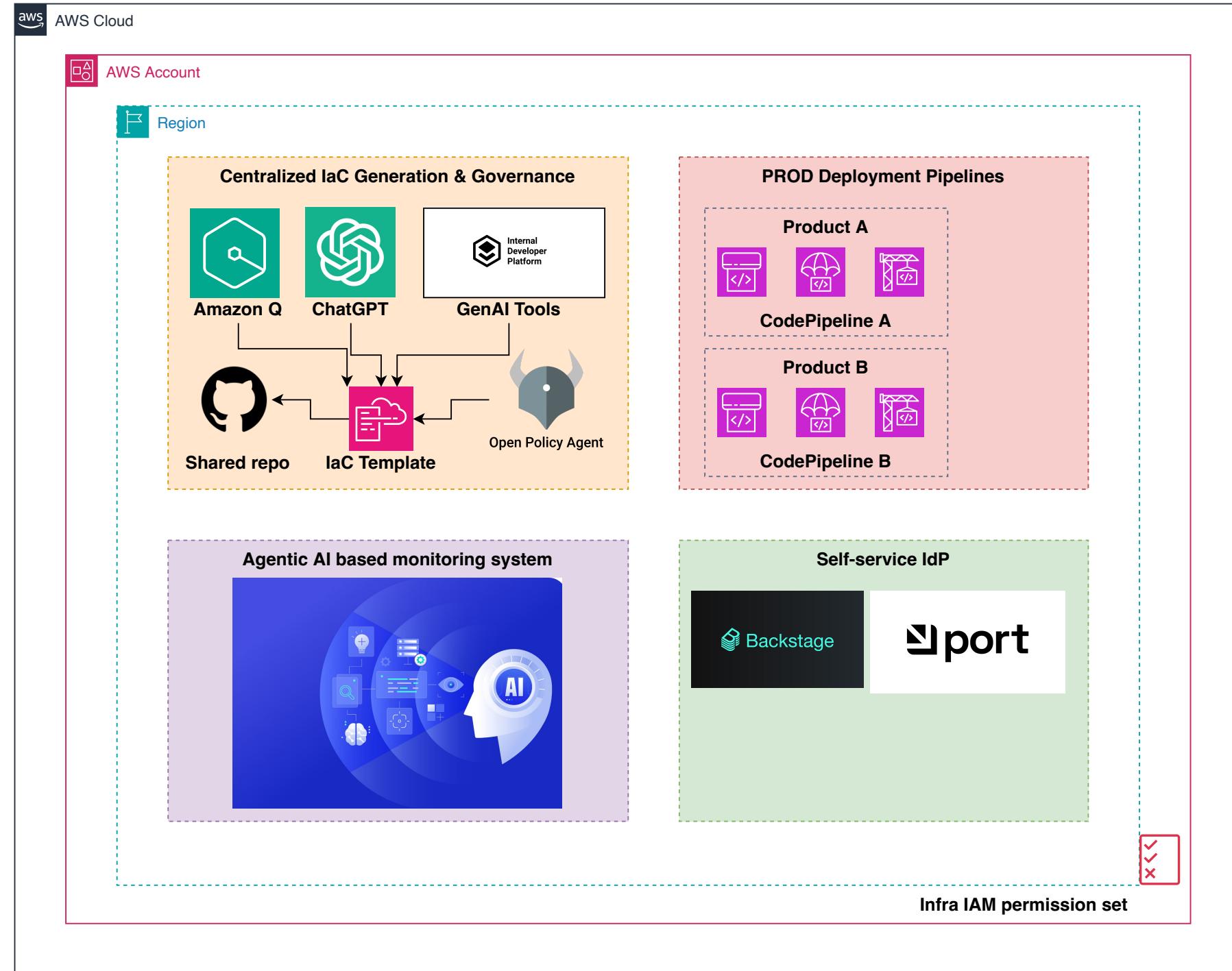
A Typical Online Food Ordering system



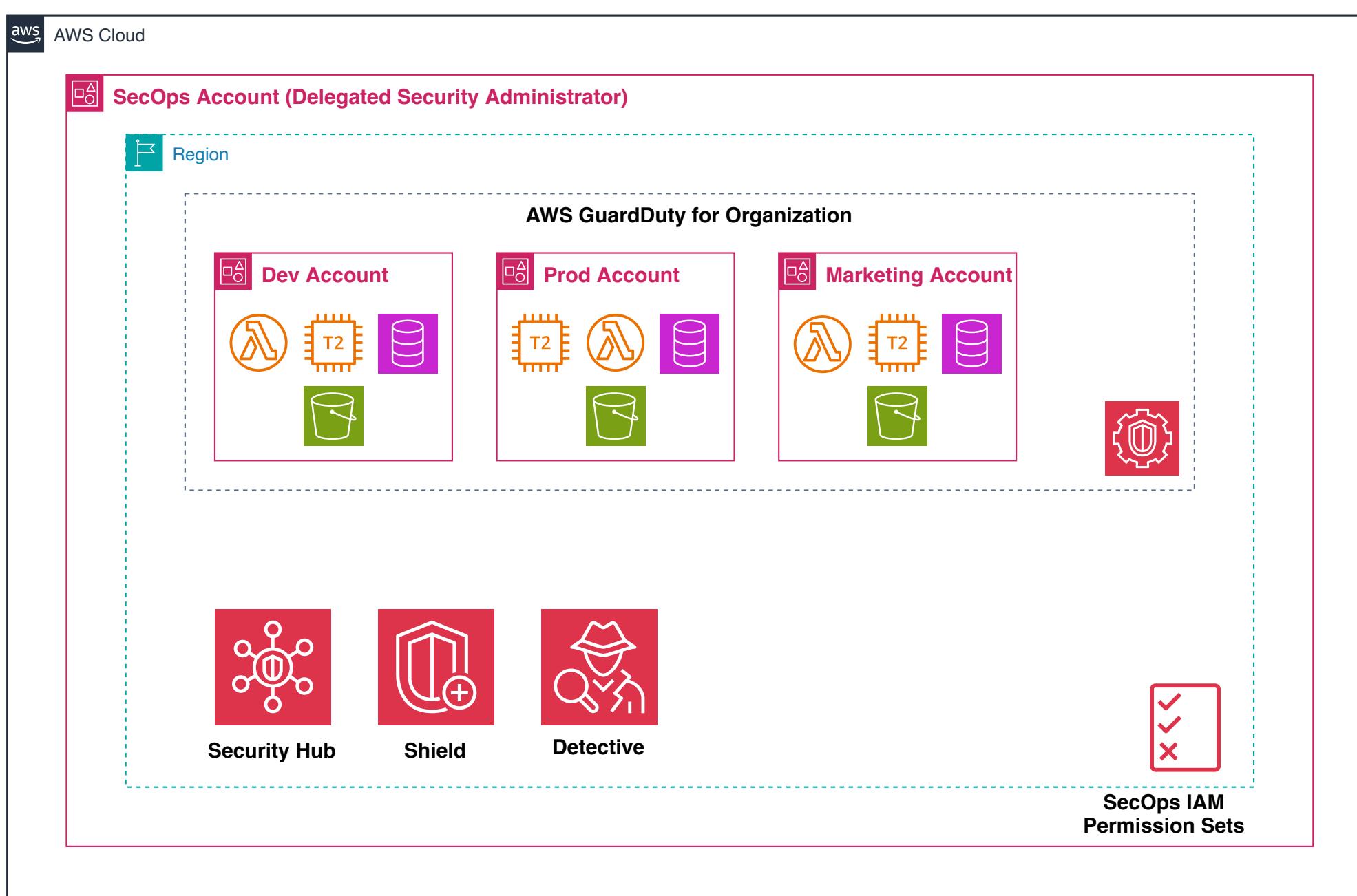
Gus Fring wants to build a **Breaking Bad** style order fulfilment system using cloud-native SDLC automation with several teams



**Dev team**



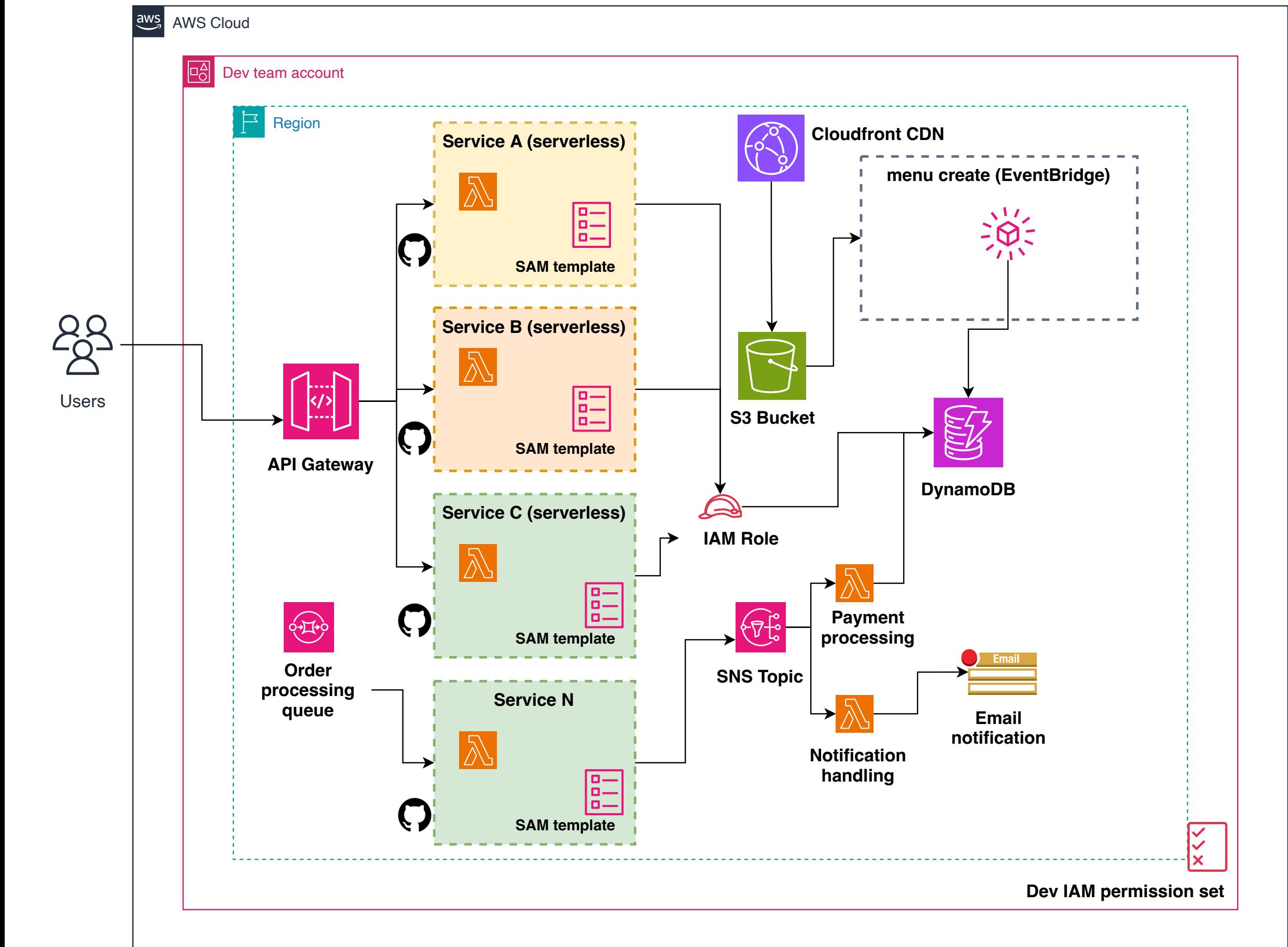
**SecOps team**



**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

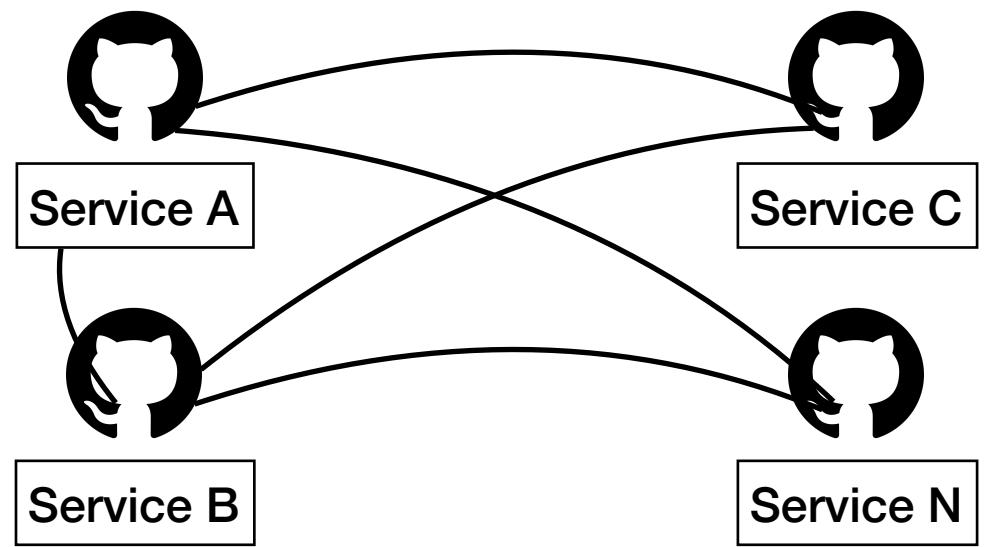


# Org & repo structure

.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)



**Organization**

Organizational units (OUs) enable you to group several accounts together and administer them as a single unit instead of one at a time.

Search by name, email, account ID or OU ID.

Hierarchy | List

Organizational structure | Account created/joined date

- Root (r-i4f4)
  - Development-Team (ou-i4f4-e3hzl77c)
    - Dev-account (Created 2022/11/17)  
691685274845 | awsage.engnr.dev@gmail.com
    - Staging (Created 2023/09/06)  
068075254459 | somnath.aws.staging@gmail.com
  - Infrastructure-Team (ou-i4f4-exz58ymg)
    - Infrastructure (Created 2025/05/01)  
417886594473 | awsmonkinfra@gmail.com
  - Security-Team (ou-i4f4-u0q5x0uc)
    - Security (Created 2025/05/01)  
592204078173 | awsmonksec@gmail.com
  - awsage-root (management account) (Joined 2022/11/07)  
878237606800 | somnath.work01@gmail.com



Team setup in AWS

Find a team...

New team

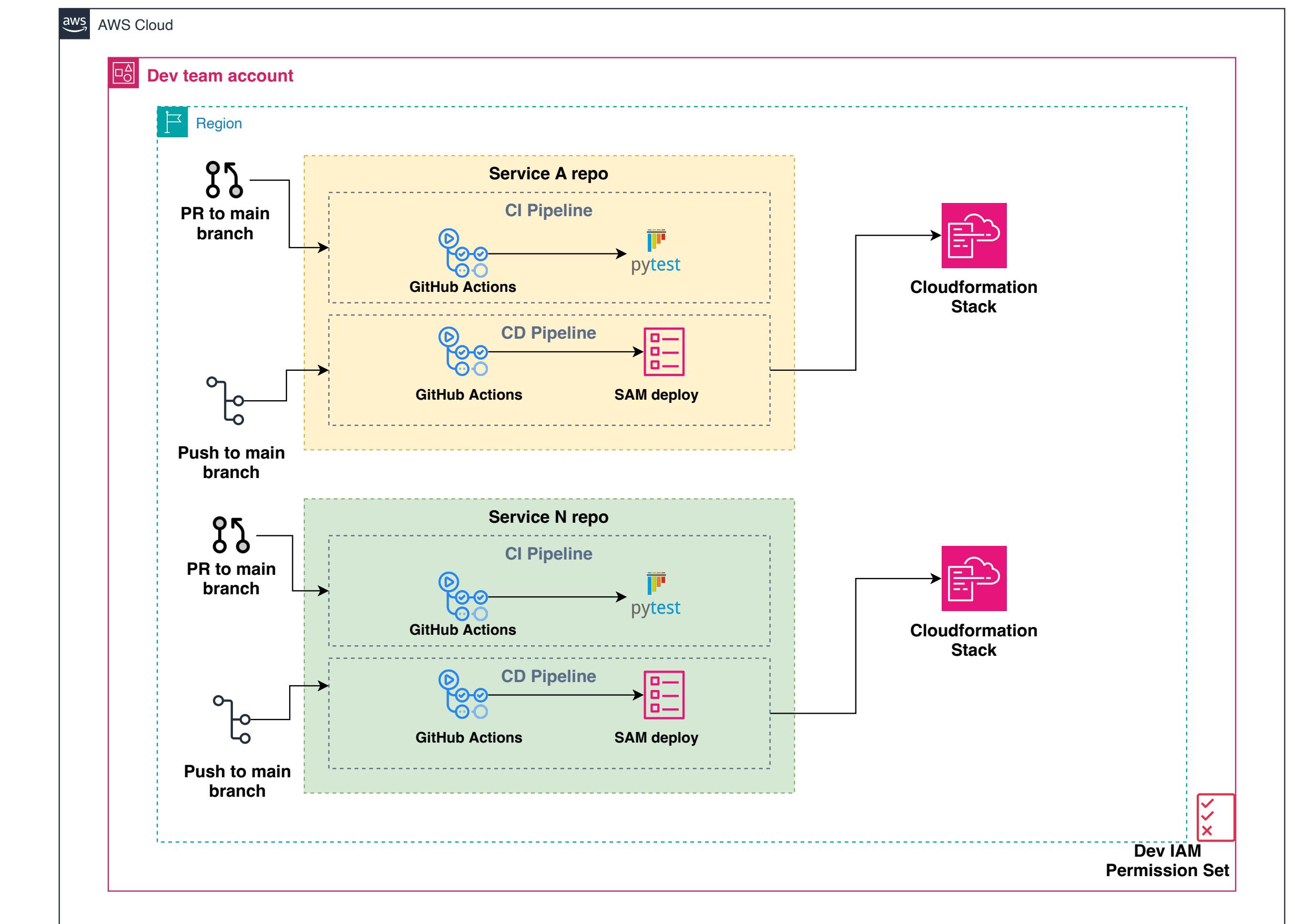
<input type="checkbox"/> Select all	Visibility ▾	Members ▾
<input type="checkbox"/> audit audit / compliance team within c9 org		1 member 0 roles 0 teams
<input type="checkbox"/> dev development team within c9 org		1 member 0 roles 0 teams
<input type="checkbox"/> infra Infra / Platform team within c9 org		1 member 0 roles 0 teams
<input type="checkbox"/> secops secops team within c9 org		2 members 0 roles 0 teams



Team setup in GitHub

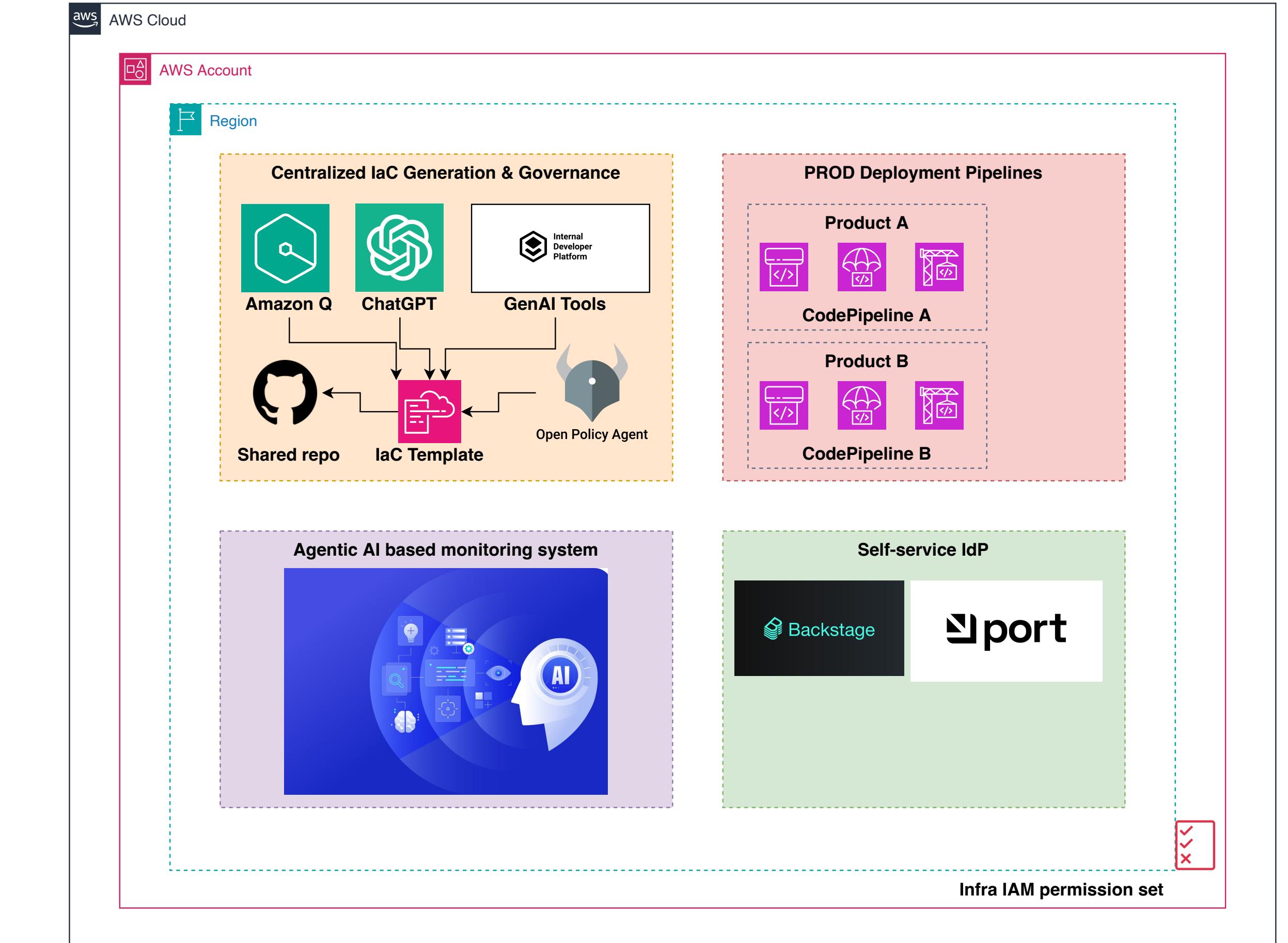
# Dev team's pipeline

- 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 its 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"
```



```
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

Check: CKV\_AWS\_116: "Ensure that AWS Lambda function is configured for a Dead Letter Queue(DLQ)"  
File: /cloudformation.yaml:4-11

Policy as code checking

The screenshot shows the Port self-service catalog interface. On the left, a sidebar lists navigation options: New, Pull Requests, Services (which is selected and highlighted in blue), Workload Overview D..., Availability Scorecard..., Workloads, Namespaces, Clusters, Production Readiness, Service Dashboard, and Add Service to Menu. The main area is titled "Services" and contains a table with the following data:

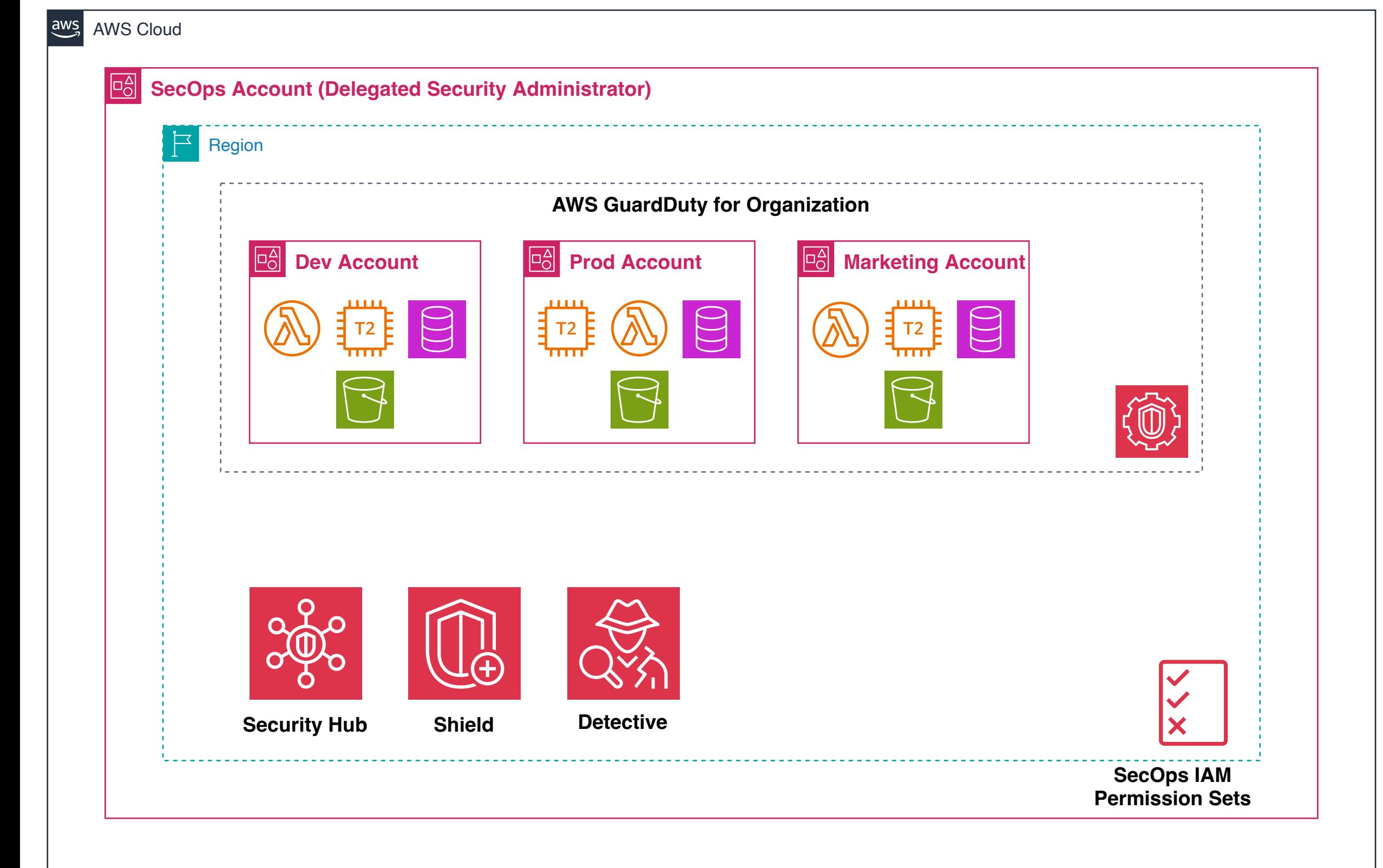
Identifier	Last Update	Entity Creation Date	URL	Language
c9_lambda_payment	a day ago	a day ago	<a href="#">🔗</a>	
c9-tw-infra	4 days ago	4 days ago	<a href="#">🔗</a>	Shell
c9-infra-iac-gen-cli	4 days ago	4 days ago	<a href="#">🔗</a>	Python
c9-tw-secops	4 days ago	4 days ago	<a href="#">🔗</a>	
lambda_menu_get	4 days ago	4 days ago	<a href="#">🔗</a>	Python
port-platform-tools	2 days ago	2 days ago	<a href="#">🔗</a>	

The screenshot shows the Port self-service hub interface. At the top, it says "Self-Service Hub". Below that is a search bar and a "Create Actions" section. A large button labeled "Scaffold a new service" has a blue outline and is highlighted. At the bottom right of this button is a green "Create" button with a lightning bolt icon.

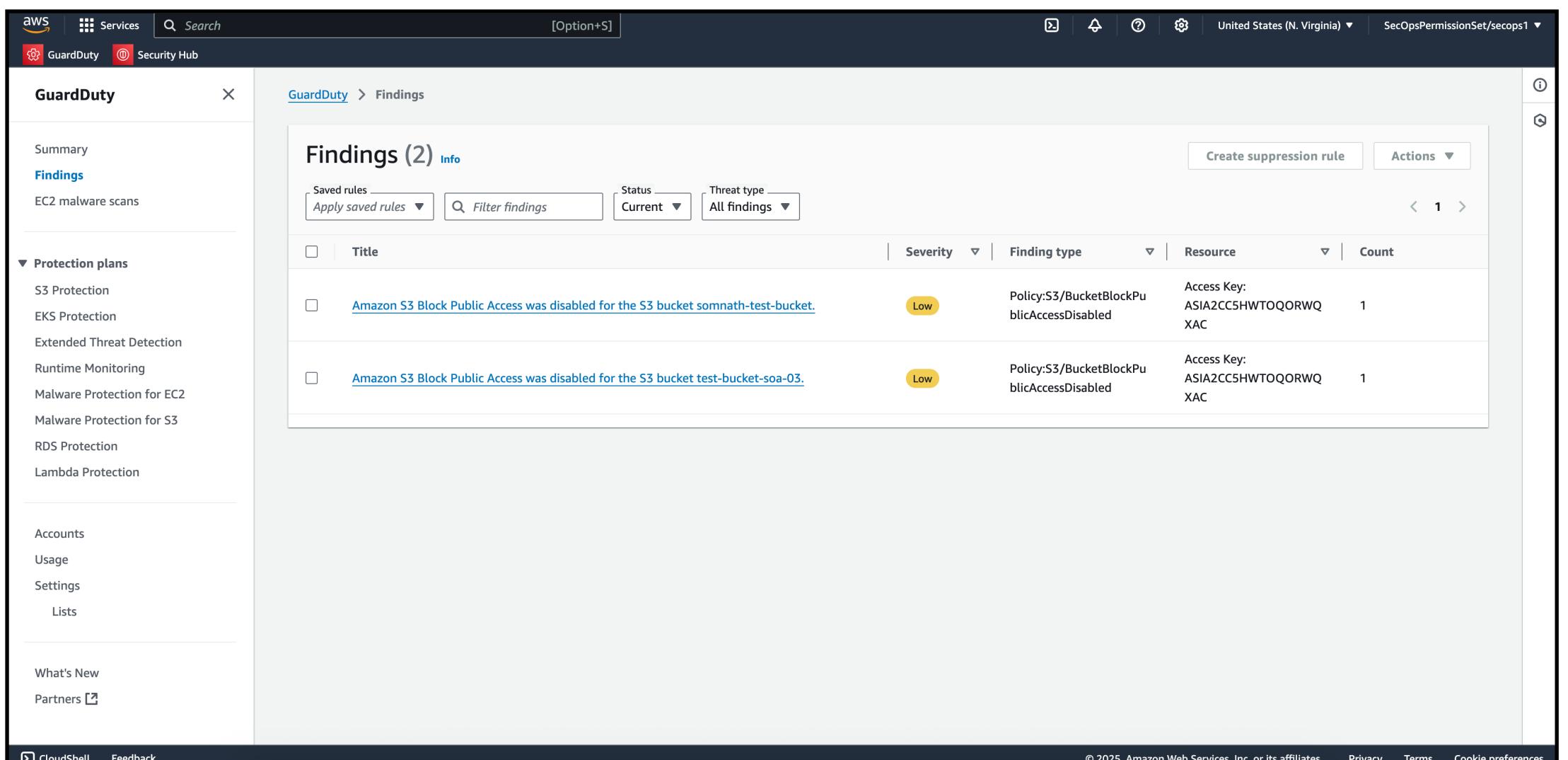
A Internal Developer Portal (IdP) using Port to self-service all team's servicing needs

# 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



The screenshot shows the AWS GuardDuty interface under the 'Findings' section. There are two findings listed:

Title	Severity	Finding type	Resource	Count
Amazon S3 Block Public Access was disabled for the S3 bucket somnath-test-bucket.	Low	Policy:S3/BucketBlockPublicAccessDisabled	Access Key: ASIA2CC5HWTOQQRWQ XAC	1
Amazon S3 Block Public Access was disabled for the S3 bucket test-bucket-soa-03.	Low	Policy:S3/BucketBlockPublicAccessDisabled	Access Key: ASIA2CC5HWTOQQRWQ XAC	1



Amazon GuardDuty findings



```
S3PublicReadProhibitedRule:  
Type: AWS::Config::ConfigRule  
Properties:  
  ConfigRuleName: s3-bucket-no-public-read  
  Description: "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*