Security and Compliance Guide

PyDocStrGen: Context-Aware Python Docstring Generator using GenAI

1. Introduction

This guide outlines the security and compliance best practices for deploying and operating PyDocStrGen within a customer's AWS account. PyDocStrGen is distributed via AWS Marketplace as a hardened AMI and deployed through a CloudFormation template supporting both new and existing VPCs. **It aims to help you minimize security risks, maintain data confidentiality, integrity, availability, and comply with your organization's security policies.**"

1.1 AWS Shared Responsibility Model

AWS secures the infrastructure (hardware, software, networking, and facilities). Customers are responsible for security in the cloud, including IAM, OS configuration, application security, and data governance.

2. Identity and Access Management (IAM)

- IAM roles are scoped using least privilege.
- No wildcard actions or open access are used.
- Access is limited to SSM Session Manager.
- IAM permissions allow:
    - Bedrock model invocation
    - CloudWatch logging
    - S3 access to specific buckets only

IAM Policy that will be generated by CloudFormation template -

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:ListFoundationModels",
        "bedrock:InvokeInlineAgent",
        "bedrock:InvokeAgent",
        "bedrock:GetAgent",
        "bedrock:CreateAgent",
        "bedrock:UpdateAgent",
        "bedrock:GetFoundationModel",
        "bedrock:InvokeModelWithResponseStream",
        "bedrock:GetInferenceProfile",
        "ssm:StartSession",
        "ssm:SendCommand",
        "ssm:GetParameters",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::{SourceS3Bucket} "
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::{SourceS3Bucket} /*"
      ],
      "Effect": "Allow"
    },
    {
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "true"
        }
      },
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::{SourceS3Bucket} /*"
      ],
      "Effect": "Allow"
    }
  ]
}    }
```

```
]
```

2.1 IAM Role and Policy Summary

The CloudFormation template provisions the following IAM resources:

EC2 Instance Role – PyDocStrGenInstanceRole

This role allows secure operation of the EC2 instance.

Permissions granted (inline policy):

- **Bedrock:**
  - bedrock:InvokeModel
  - bedrock:ListFoundationModels
  - bedrock:InvokeInlineAgent
  - bedrock:InvokeAgent
  - bedrock:GetAgent
  - bedrock:CreateAgent
  - bedrock:UpdateAgent
  - bedrock:GetFoundationModel
  - bedrock:InvokeModelWithResponseStream
  - bedrock:GetInferenceProfile

- **Systems Manager (SSM):**
  - ssm:StartSession
  - ssm:SendCommand

    - ssm:GetParameters

- **CloudWatch:**
    - logs:CreateLogGroup
    - logs:CreateLogStream
    - logs:PutLogEvents

- **S3 :** scoped to specified buckets
    - s3:GetObject
    - s3:PutObject
    - s3:ListBucket

CloudFormation Execution Role

This role is used internally by CloudFormation to provision and manage the solution's resources during stack launch and updates. All IAM resources follow least privilege principles and minimize wildcard usage.

Managed policies attached:

- AmazonSSMManagedInstanceCore
- CloudWatchAgentServerPolicy

All IAM resources follow **least privilege principles** and minimize wildcard usage.

2.2 Securing Access

- Access to the EC2 instance is provided through **AWS Systems Manager Session Manager**.
- SSH is disabled by default.
- IAM roles must include ssm:StartSession to enable secure CLI or browser-based connections.
- Customers may optionally provide a KeyPair to enable SSH access, but this is not required. If SSH use is needed, customer will need to open SSH incoming port in security group manually.

3. Credential Management

PyDocStrGen does not require customers to store any external credentials directly within the application environment. There is no need for AWS Secrets Manager integration in the base product for core operations, as it is designed for an 'S3 in, S3 out' workflow. Consequently, secrets rotation procedures are not applicable as no secrets or external credentials are stored or managed by the solution itself.

4. Network Security

- Instances are launched in private subnets with no public IP.
- NAT Gateway is used for outbound access if deploying a new VPC.
- Public subnets are not used unless explicitly enabled.

4.1 Security Group Configuration

The CloudFormation template provisions a dedicated security group with:

Inbound Rules:

- None. All inbound traffic is denied by default.

Outbound Rules:

- HTTPs traffic to 0.0.0.0/0 is allowed, which enables secure HTTPS access to:
  - AWS Systems Manager
  - Amazon Bedrock
  - CloudWatch Logs
  - Amazon S3

Customers can modify this security group post-deployment to enable inbound/outbound access if necessary (e.g., for HTTP or SSH).

5. Data Protection

- The root EBS volume is not encrypted by default to allow flexibility. **This is by design, enabling customers to choose** their own KMS key (customer-managed or AWS-managed) **or whether to encrypt based on their specific compliance or cost requirements.** Customers can enable encryption using AWS default or custom KMS keys.
- PyDocStrGen processes your valuable source code. It is critical to ensure that this code is only processed within the secure AWS environment where PyDocStrGen is deployed. Customers are responsible for ensuring there is no unintentional egress of raw source code to external services or unencrypted storage outside their control. The EC2 instances where code is processed have local storage for temporary operations; customers should ensure this storage is properly secured (e.g., through

EBS volume encryption enabled post-deployment) and consider methods for clearing temporary data after processing if required by their security policies.

6. Logging and Monitoring

- **CloudWatch Agent** is installed and configured to forward logs:
  - System logs: /var/log/syslog
  - Application logs: /var/log/pydocstrgen/app.log
- Customers can create alarms and configure retention policies for CloudWatch Logs. By default, the log retention period is set to two weeks.

7. Vulnerability Management and Patching

- The AMI is based on Ubuntu 24.04 LTS with all patches applied at build time.
- **No vulnerability scanners** are installed by default.
- Customers are encouraged to run their own tools (e.g., AWS Inspector, Trivy) after deployment.
- Customers are recommended to configure **AWS Patch Manager** and maintenance windows.

8. Backup and Restore

- EC2 instances can be backed up via EBS snapshots.
- All outputs and persistent data, including **docstrings and source code**, should be stored in Amazon S3 (created or existing).

- Customers should maintain copy of their source code in a separate code versioning system.

9. Compliance and Governance

- Supports **resource tagging** (Environment, Application, etc.)
- Helps customers with:
    - Budget management
    - IAM condition keys
    - Resource automation and tracking

10. Incident Response

- Logs are forwarded to CloudWatch to support incident response, **leveraging monitoring and alerting capabilities to detect unusual or malicious activity.** Customers should define internal alerting and escalation.
- Refer to: AWS Security Incident Response Guide

11. Forensics and Remediation (Customer Responsibility)

If a security incident occurs, your security team will be responsible for forensics, containment, eradication, and recovery.

- **Containment:** For PyDocStrGen, this might involve isolating compromised EC2 instances, blocking suspicious IPs via Security Groups/NACLs, or temporarily disabling access to affected S3 buckets.
- **Eradication:** Removing the root cause of the compromise (e.g., patching vulnerabilities, rotating compromised credentials).

- **Recovery:** Restoring PyDocStrGen to a known good state from backups, or relaunching the stack from CloudFormation.

12. Security Contact

Please report security concerns or vulnerabilities to:

📧 support-pydocstrgen@forgeahead.io