ADDF(Autonomous Driving Data Framework) 보안 및 운영 가이드

AWS 권장 가이드

AWS 권장 가이드: ADDF(Autonomous Driving Data Framework) 보안 및 운영 가이드

Table of Contents

ADDF(Autonomous Driving Data Framework) 보안 및 운영 가이드

Andreas Falkenberg, Junjie Tang, Torsten Reitemeyer, Srinivas Reddy Cheruku, Amazon Web Services

2022년 11월(문서 기록)

ADDF(Autonomous Driving Data Framework)는 중앙 집중식 데이터 스토리지, 데이터 처리 파이프라인, 시각화 메커니즘, 검색 인터페이스, 시뮬레이션 워크로드, 분석 인터페이스, 사전 구축된 대시보드 구성 등 ADAS(Advanced Driver-Assistance System)를 위한 일반 작업을 구현하려는 자동차 팀을 위해 재사용 가능한 모듈식 코드 아티팩트를 제공하도록 설계된 오픈 소스 프로젝트입니다. ADDF를 사용하면 완전히 사용자 지정 가능한 모듈을 공유, 수정 또는 생성하여 이러한 솔루션을 생성하고 배포하는 데 필요한 노력을 줄일 수 있습니다.

이 가이드는 AWS 클라우드에서 ADDF를 안전하게 배포하고 운영하기 위한 모범 사례를 이해하는 데 도움이 됩니다. 다음 주제에 대해 설명합니다.

- 아키텍처 및 용어 - 일반 아키텍처, 워크플로 및 중요 용어를 검토합니다.
- 공동 책임 모델 - ADDF 배포 및 클라우드 리소스를 보호하는 데 AWS 있어 역할과의 역할을 이해합니다.
- 보안 검토 프로세스 – ADDF는 오픈 소스 프로젝트이므로 AWS 및 기여자가 보안 검토를 완료하는 방법을 검토합니다.
- 내장 보안 기능 - ADDF 오픈 소스 프로젝트와 배포 프레임워크에 보안 모범 사례와 기능이 어떻게 구축되어 있는지 검토합니다.
- 보안 설정 및 운영 - AWS 클라우드에서 ADDF를 배포하고 운영하는 방법을 알아봅니다.

수강 대상

이 가이드는 ADDF 평가, 배포, 사용자 지정 및 운영을 담당하는 개발 운영(DevOps) 팀, 인프라 엔지니어, 관리자, IT 보안 직원 및 인시던트 대응 팀을 대상으로 합니다. 개념 증명 또는 프로덕션 환경에 이 가이드의 권장 사항을 적용할 수 있습니다.

이 가이드에서는 ADDF에 대한 사전 지식이 없는 것으로 가정합니다. 그러나 진행하기 전에 ADDF readme(GitHub)를 읽어보는 것이 좋습니다.

목표 비즈니스 성과

이 가이드는 개발 및 프로덕션 환경에서 ADDF를 보다 확실하고 안전하게 설정하고 운영하는 데 도움이 되도록 설계되었습니다.

ADDF 아키텍처 및 용어

이 가이드의 보안 및 운영 주제를 이해하려면 먼저 ADDF(Autonomous Driving Data Framework)의 용어, 구성 요소 및 아키텍처에 대한 개략적 이해가 필요합니다. 이 섹션은 다음 주제로 구성됩니다.

- ADD 용어
- ADDF 아키텍처

ADD 용어

ADDF의 주요 용어는 다음과 같습니다.

- ADDF 모듈 - 모듈은 ADAS(Advanced Driver-Assistance System)에서 일반 작업을 구현하는 코드형 인프라(IaC)입니다. 일반 작업으로는 중앙 집중식 데이터 스토리지, 데이터 처리 파이프라인, 시각화 메커니즘, 검색 인터페이스, 시뮬레이션 워크로드, 분석 인터페이스 및 사전 구축된 대시보드 구성 등이 있습니다. 요구 사항에 따라 모듈을 생성하거나 기존 모듈을 재사용 또는 사용자 지정할 수 있습니다.

  AWS Cloud Development Kit (AWS CDK) 를 사용하여 ADDF 모듈을 정의하거나 Hashicorp Terraform 또는와 같은 공통 IaC 프레임워크 AWS CloudFormation를 사용하여 ADDF 모듈을 구현할 수 있습니다. 모듈에는 입력 파라미터 세트가 있습니다. 입력 파라미터는 다른 모듈의 출력 값에 따라 달라질 수 있습니다. ADDF 모듈은 ADDF 대상 AWS 계정에 대한 가장 작은 배포 단위입니다.
- ADDF 배포 매니페스트 파일 - 이 파일은 독립 실행형 ADDF 모듈의 오케스트레이션을 정의합니다. 오케스트레이션은 모듈의 배포 순서를 나타냅니다. ADDF 배포 매니페스트 파일에서 ADDF 그룹을 사용하여 관련 모듈을 그룹화할 수 있습니다. 이 파일에서는 ADDF 도구 체인, AWS 계정 ADDF 대상 AWS 계정및 대상도 정의합니다 AWS 리전.
- ADDF 배포 프레임워크 -이 프레임워크는 ADDF 배포 매니페스트 파일에 정의된 오케스트레이션을 AWS 계정 기반으로 ADDF 모듈을 ADDF 대상에 배포합니다. ADDF 배포 프레임워크는 다음 AWS 오픈 소스 프로젝트를 사용하여 구현됩니다.
  - SeedFarmer(GitHub) – SeedFarmer는 ADDF 배포에 사용되는 CLI 도구로서, 각 모듈 상태를 관리하고, 모듈 코드를 준비 및 패키징하고, ADDF 배포 역할에 대한 최소 권한 정책을 생성하고, CodeSeeder가 배포에 사용하는 의미 체계 지침을 제공합니다. SeedFarmer와 직접 상호 작용하여 ADDF 배포를 실행하거나 지속적 통합 및 지속적 배포(CI/CD) 파이프라인에 통합할 수 있습니다.

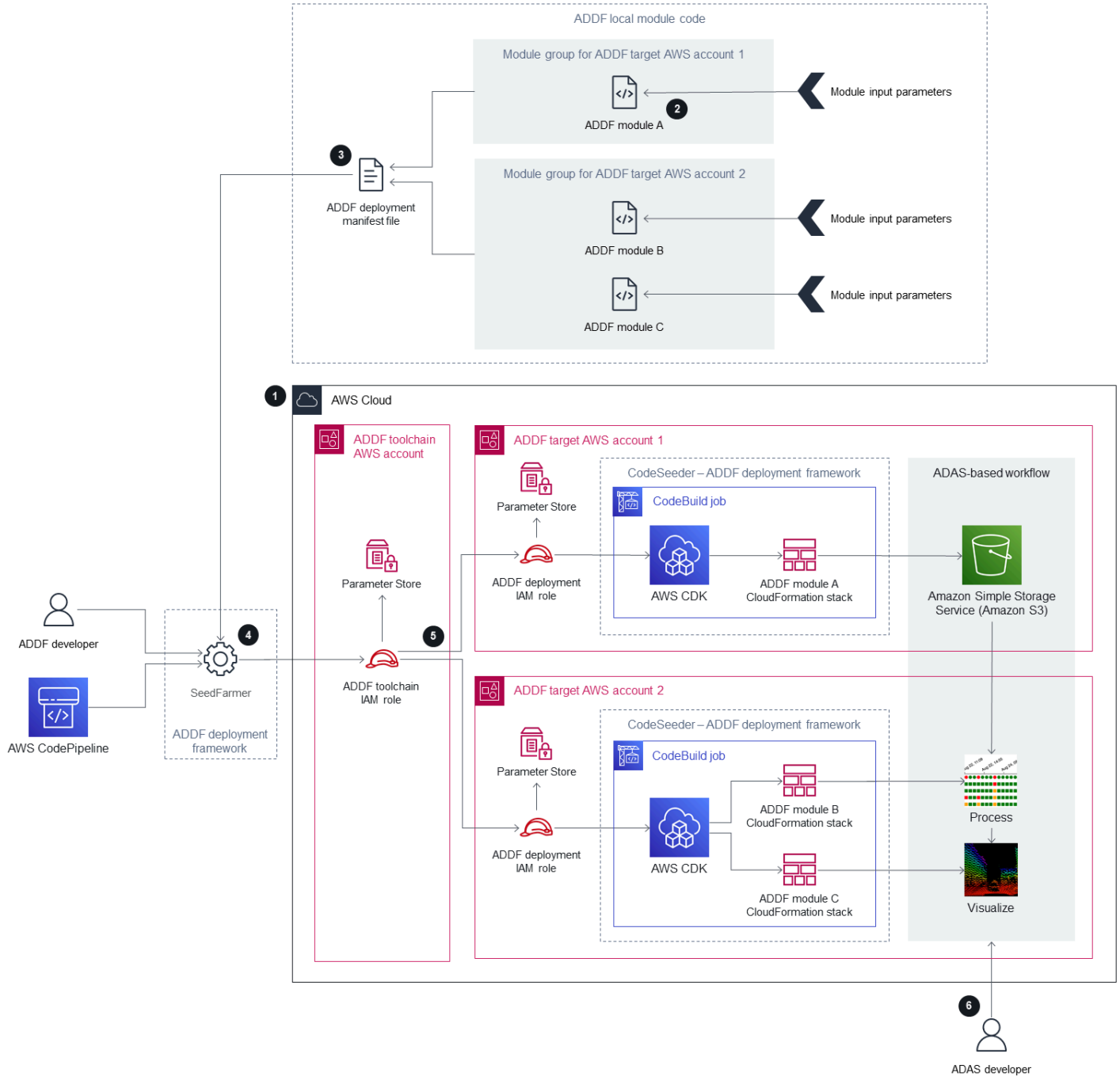
  - CodeSeeder(GitHub) - CodeSeeder는 AWS CodeBuild 작업을 통해 임의의 인프라를 코드 패키지로 배포합니다. SeedFarmer는 CodeSeeder를 자동으로 오케스트레이션하고 실행합니다. SeedFarmer만 CodeSeeder와 직접 상호 작용합니다.

  ADDF 배포 프레임워크는 단일 계정 및 다중 계정 아키텍처에서 배포를 지원하도록 설계되었습니다. 조직의 요구 사항에 따라 단일 계정 아키텍처가 필요한지 다중 계정 아키텍처가 필요한지 결정합니다.

- ADDF 도구 체인 AWS 계정 -이 계정은 ADDF 배포 매니페스트 파일의 정의를 AWS 계정기반으로 ADDF 대상에 모듈 배포를 오케스트레이션하고 관리합니다. ADDF 배포에는 ADDF 도구 체인 AWS 계정이 하나만 있을 수 있습니다. 단일 계정 아키텍처에서는 ADDF 도구 체인 AWS 계정 이 ADDF 대상 AWS 계정이기도 합니다. 이 계정에는 ADDF 배포 프로세스 중에 SeedFarmer가 수임하는 ADDF 도구 체인 IAM 역할이라는 AWS Identity and Access Management (IAM) 역할이 포함되어 있습니다. 이 가이드에서는 ADDF 도구 체인을 도구 체인 계정 AWS 계정 이라고 합니다.
- ADDF 대상 AWS 계정 - ADDF 모듈을 배포하는 대상 계정입니다. 하나 이상의 대상 계정이 있을 수 있습니다. 이러한 계정에는 ADDF 배포 매니페스트 파일과 매핑된 모듈에 설명된 리소스 및 애플리케이션 로직이 포함됩니다. 단일 계정 아키텍처에서 ADDF 대상 AWS 계정 은 ADDF 도구 체인이기도 합니다 AWS 계정. 각 ADDF 대상 계정에는 배포 프로세스 중에 CodeSeeder가 수임하는 ADDF 배포 IAM 역할이라는 IAM 역할이 포함되어 있습니다. 이 가이드에서는 ADDF 대상을 대상 계정 AWS 계정 이라고 합니다.
- ADDF 인스턴스 - ADDF 배포 매니페스트 파일에 정의된 대로 ADDF와 모듈을 클라우드에 배포하면 이는 ADDF 인스턴스가 됩니다. ADDF 인스턴스는 단일 계정 또는 다중 계정 아키텍처를 가질 수 있으며 여러 ADDF 인스턴스를 배포할 수 있습니다. 인스턴스 수 선택 및 사용 사례에 맞는 계정 아키텍처 설계에 대한 자세한 내용은 ADDF 아키텍처 정의 섹션을 참조하세요.

ADDF 아키텍처

다음 다이어그램에서는 AWS 클라우드의 ADDF 인스턴스에 대한 개략적 아키텍처를 보여줍니다. 전용 도구 체인 계정과 2개의 대상 계정을 포함한 다중 계정 아키텍처가 있습니다. 이 가이드에서는 ADDF를 사용하여 대상 계정에 리소스를 배포하는 엔드 투 엔드 프로세스에 대해 설명합니다.



1. ADDF AWS 계정을 생성하고 부트스트랩합니다.

   제대로 작동하려면 각 계정을 ADDF와 AWS CDK에 부트스트랩해야 합니다. 새 ADDF 배포이거나 새 대상 계정을 추가하는 경우 다음을 수행합니다.

   a. 도구 체인 계정 및 각 대상 계정 AWS CDK 의 부트스트랩입니다. 지침은 Bootstrapping(AWS CDK 설명서)을 참조하세요. ADDF는 AWS CDK 를 사용하여 인프라를 배포합니다.

b. 도구 체인 계정과 각 대상 계정에서 ADDF를 부트스트랩합니다. 지침은 ADDF 배포 안내서의 부트스트랩 AWS 계정(들)을 참조하세요. 그러면 SeedFarmer와 CodeSeeder에 필요한 모든 ADDF 관련 IAM 역할이 설정됩니다.

> **Note**
>
> 처음에 ADDF를 배포하거나 새 대상 계정을 추가하는 경우에만 이 단계를 수행해야 합니다. 이 단계는 이미 설정된 ADDF 인스턴스에 대한 반복 ADDF 배포의 일부가 아닙니다.

2. ADDF 모듈을 생성하거나 사용자 지정합니다.

   해결하려는 특정 문제에 따라 ADDF 모듈을 생성하거나 사용자 지정합니다. 모듈은 격리된 작업 또는 작업 그룹을 나타내야 합니다. 필요에 따라 모듈의 입력 파라미터를 정의하고 모듈 출력 값을 다른 모듈의 입력 파라미터로 사용합니다.

3. ADDF 배포 매니페스트 파일에 모듈 오케스트레이션을 정의합니다.

   ADDF 매니페스트 파일에서 모듈을 그룹으로 구성하고 배포 순서와 모듈 간의 종속성을 정의합니다. 이 파일에서는 각 ADDF 그룹 및 해당 모듈의 단일 도구 체인 계정과 대상 계정(포함 AWS 리전)도 지정합니다.

4. ADDF 배포 매니페스트 파일을 평가하고 배포 범위를 설정합니다.

   ADDF 개발자 또는와 같은 CI/CD 파이프라인은 CLI 도구인 SeedFarmer 배포 매니페스트 파일의 평가를 AWS CodePipeline시작합니다. 평가를 시작하려면 다음을 수행하세요.

   - SeedFarmer는 ADDF 배포 매니페스트 파일을 평가를 위한 입력 파라미터로 사용합니다.
   - ADDF 도구 체인 IAM 역할을 수임하기 위해 SeedFarmer에는 1단계의 ADDF 부트스트랩 프로세스 중에 정의된 것과 동일하고 유효한 IAM 역할 또는 사용자의 보안 인증 정보가 필요합니다.

   SeedFarmer에 ADDF 도구 체인 IAM 역할을 수임할 수 있는 올바른 보안 인증 정보가 없거나 ADDF 배포 매니페스트 파일에 액세스할 수 없는 경우 평가가 시작되지 않습니다.

   SeedFarmer가 평가를 시작할 수 있으면 도구 체인 계정에서 ADDF 도구 체인 IAM 역할을 수임합니다. 여기에서 SeedFarmer는 해당 계정에서 ADDF 배포 IAM 역할을 수임하여 모든 대상 계정에 액세스할 수 있습니다. 그런 다음 SeedFarmer는 도구 체인 계정과 대상 계정에서 ADDF 메타데이터를 읽으려고 시도합니다. 다음 중 하나가 발생합니다.

   - 읽을 ADDF 메타데이터가 없으면 이는 새 ADDF 인스턴스임을 나타냅니다. SeedFarmer는 배포 범위가 ADDF 배포 매니페스트 파일과 해당 콘텐츠 전체인지 확인합니다.

- ADDF 메타데이터가 존재하는 경우 SeedFarmer는 ADDF 배포 매니페스트 파일과 해당 콘텐츠를 대상 계정에 있는 배포된 기존 아티팩트의 MD5 해시와 비교합니다. 배포 가능한 변경 사항이 감지되면 이 프로세스가 계속됩니다. 배포 가능한 변경 사항이 감지되지 않으면 프로세스가 완료된 것입니다.

5. 대상 계정에 범위 내 ADDF 모듈을 배포합니다.

   이제 CodeSeeder에는 ADDF 배포 매니페스트 파일과 이전 단계의 평가 결과에 따라 실행할 순서가 지정된 배포 목록이 있습니다. 순서가 지정된 목록에 따라 CodeSeeder는 연결된 각 대상 계정에서 ADDF 배포 IAM 역할을 수임합니다. 그런 다음 AWS CodeBuild 작업에서 CodeSeeder를 실행하여 ADDF 모듈에 대한 AWS CDK 애플리케이션과 같은 개별 IaC 배포를 생성하거나 업데이트합니다. 기본적으로 ADDF는를 IaC 프레임워크 AWS CDK 로 사용하지만 다른 일반적인 IaC 프레임워크도 지원됩니다. 각 대상 계정에 대한 프로세스가 완료되면 ADDF 배포 매니페스트 파일에 정의한 대로 완전히 배포된 크로스 계정 엔드 투 엔드 ADAS 기반 워크플로가 완성됩니다.

   단일 계정 아키텍처를 사용하는 경우 도구 체인 계정과 대상 계정은 동일한 계정이며, 하나의 계정에 설명된 모든 기능이 있습니다.

6. ADDF에서 배포한 인프라를 사용합니다.

   ADAS 개발자는 사용 사례에 정의된 대로 배포된 ADAS 기반 워크플로를 사용할 수 있습니다.

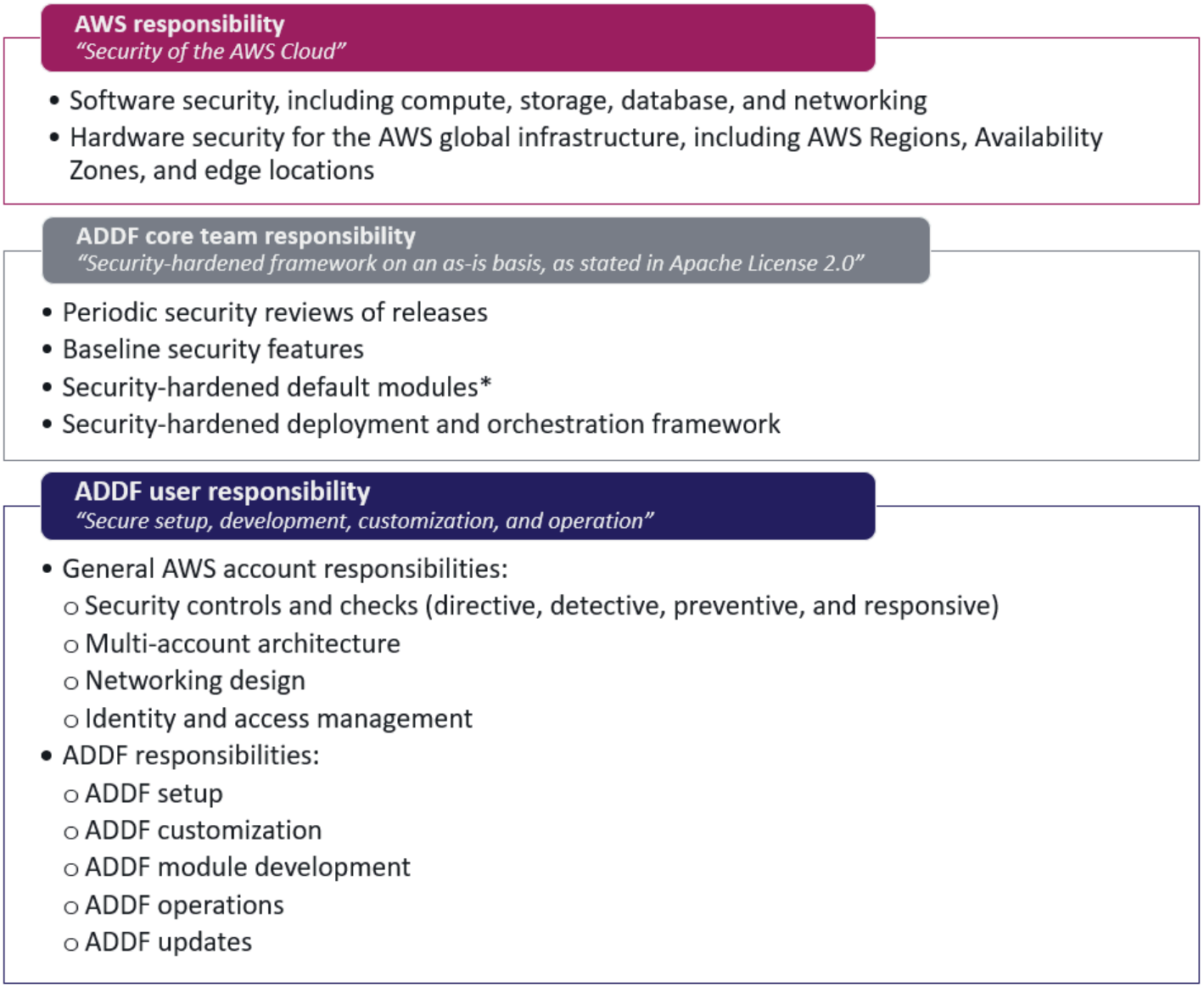
   이 워크플로는 ADDF 다중 계정 환경의 단일 인스턴스 아키텍처를 설명합니다. 개발, 배포 및 운영 모델에 따라 다단계 환경에서 여러 ADDF 인스턴스를 실행하는 것이 좋습니다. 일반적인 설정에는 개발, 테스트 및 프로덕션 AWS 계정 용 브랜치와 같이 각 배포 단계 전용가 있는 전용 ADDF 인스턴스가 포함될 수 있습니다. 각 ADDF 인스턴스에 대해 고유한 리소스 네임스페이스를 생성했다고 AWS 리전가정하면 동일한 단일 계정 또는 다중 계정 환경의 동일한에서 여러 ADDF 인스턴스를 실행할 수도 있습니다. 자세한 내용은 ADDF 아키텍처 정의 단원을 참조하십시오.

ADDF 공동 책임 모델

에 적용되는 공동 책임 모델은 자율 주행 데이터 프레임워크(ADDF) AWS 서비스 에도 적용됩니다. 다음 다이어그램에 명시된 대로 ADDF 보안 책임을 공유하는 엔터티는 다음과 같습니다.

- AWS - 클라우드 인프라 공급자 상품입니다 AWS 서비스.
- ADDF 핵심 팀 – ADDF 핵심 팀은 ADDF 리포지토리(GitHub)에 ADDF 릴리스를 게시하는 엔터티입니다.
- ADDF 사용자 – ADDF 사용자에는 다음이 포함되지만 이에 국한되지는 않습니다.
  - ADDF 개발자– 새로운 ADDF 모듈 코드를 변경, 사용자 지정 또는 생성하는 사람입니다.
  - ADDF 운영자 - ADDF 인스턴스를 설정하고 운영하는 사람입니다.
  - ADAS 개발자 - ADDF에서 배포한 리소스의 최종 사용자 또는 소비자입니다. 예를 들어, ADAS 개발자는 ADDF 배포의 일부로 생성된 시각화 프런트엔드를 쿼리할 수 있습니다.

다음 다이어그램은 AWS ADDF 코어 팀과 ADDF 사용자 간의 공동 책임을 요약합니다.



AWS 책임

AWS 는 공동 책임 모델에 정의된 AWS 클라우드대로에서 제공되는 모든 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 이 인프라는 AWS 클라우드 서비스를 실행하는 하드웨어, 소프트웨어, 네트워킹 및 시설로 구성됩니다.

ADDF 핵심 팀 책임

ADDF 핵심 팀은 Apache License 2.0(GitHub)에 따라 최선의 노력을 바탕으로 자체적으로 안전한 프레임워크를 제공합니다. ADDF 핵심 팀은 다음을 담당합니다.

- 릴리스에 대한 주기적인 보안 검토
- 기본 보안 기능
- 보안이 강화된 기본 모듈(`/modules/demo-only/` 폴더에 있는 모든 모듈은 제외됩니다. 이러한 모듈은 개념 증명 목적으로만 사용되며 보안 강화를 받지 않습니다.)
- 보안이 강화된 배포 및 오케스트레이션 프레임워크

이러한 보안 책임은 수정이나 사용자 정의 없이 GitHub 리포지토리에 제공된 대로 프레임워크에만 적용됩니다. 여기에는 `modules/demo-only/` 폴더의 ADDF 모듈을 제외한 모든 ADDF 모듈이 포함됩니다. 이 폴더의 ADDF 모듈은 보안이 강화되지 않았으므로 프로덕션 환경 또는 민감하거나 보호되는 데이터가 있는 환경에 배포해서는 안 됩니다. 이러한 모듈은 시스템 기능을 보여주기 위해 포함되어 있으며, 보안이 강화된 사용자 지정 자체 모듈을 생성하기 위한 기반으로 사용할 수 있습니다.

> **Note**
>
> 프레임워크형 ADDF는 있는 그대로 제공됩니다. Apache License 2.0(GitHub)에 명시된 대로 어떠한 책임과 보증도 제공되지 않습니다. ADDF에 대한 자체 보안 평가를 수행하고 조직의 특정 보안 요구 사항을 준수하는지 확인해야 합니다.

ADDF 사용자 책임

ADDF와 해당 모듈은 ADDF가 안전한 방식으로 설정, 사용자 지정 및 운영되는 경우에만 안전합니다. ADDF 사용자는 다음의 보안을 전적으로 책임집니다.

- 일반적인 AWS 계정 책임:
  - 보안 제어 및 검사(지침, 탐지, 예방 및 대응)
  - 다중 계정 아키텍처
  - 네트워킹 설계
  - ID 및 액세스 관리
- ADDF별 책임:
  - ADF 설정
  - ADDF 사용자 지정
  - ADDF 모듈 개발
  - ADDF 작업

- ADDF 업데이트

일반적인 AWS 계정 책임

ADDF 관련 리소스를에 배포하기 전에 AWS Well-Architected Framework의 모범 사례에 따라를 구성해야 AWS 계정 AWS 계정 합니다. 여기에는 지침, 탐지, 예방 및 대응 보안 제어가 포함됩니다. 보안 위반이나 인시던트가 발생할 경우를 대비하여 상세한 완화 프로세스를 마련해야 합니다. 조직의 정책에는 ID, 액세스 및 네트워킹을 중앙에서 관리하기 위한 요구 사항이 포함되어야 합니다. 일반적으로 전용 랜딩 존 팀에서 이러한 요구 사항과 서비스를 처리합니다.

ADDF별 책임

보안 ADDF 설정

ADDF 사용자의 책임은 ADDF 설명서에 따라 ADDF를 안전하게 설정하는 것부터 시작됩니다. ADDF Deployment Guide(GitHub)의 지침을 따르는 것이 좋습니다. ADDF를 안전하게 설정하는 방법에 대한 자세한 내용은 ADDF 아키텍처 정의 및 초기 설정 섹션을 참조하세요.

보안 ADDF 사용자 지정

CodeSeeder, SeedFarmer 및 ADDF 핵심 모듈과 같은 ADDF 핵심 기능을 사용자 지정하는 경우 ADDF 사용자는 해당 변경에 대한 전적인 책임을 집니다. 자세한 내용은 ADDF 배포 프레임워크 코드 사용자 지정 단원을 참조하십시오.

보안 ADDF 모듈 개발

ADDF를 사용하여 배포되는 모든 사용자 지정 모듈에 대한 책임은 전적으로 ADDF 사용자에게 있습니다. 또한 ADDF 제공 모듈의 모든 코드 변경에 대한 책임은 ADDF 사용자에게 있습니다. 자세한 내용은 ADDF에서 사용자 지정 모듈 작성 단원을 참조하십시오.

보안 ADDF 업데이트 및 운영

프레임워크가 발전함에 따라 ADDF는 기능 및 보안 업데이트를 받습니다. GitHub 리포지토리에 게시된 업데이트를 정기적으로 확인하고 ADDF를 장기간 안전하게 운영하는 것은 ADDF 사용자의 책임입니다. 자세한 내용은 반복되는 ADDF 배포, 반복되는 보안 감사, ADDF 업데이트 및 서비스 해제 부분을 참조하세요.

ADDF 보안 검토 프로세스

ADDF(Autonomous Driving Data Framework)는 보안을 염두에 두고 구축되었습니다. 일반에 공개 전에 AWS 는 ADDF에 대한 초기 내부 보안 검토를 수행하고 확인된 보안 문제를 해결했습니다. AWS 및 오픈 소스 커뮤니티 모두 프레임워크의 지속적인 보안 검토에 기여합니다.

의 정기 보안 검토 AWS

ADDF는 소유한 awslabs GitHub 조직에 게시됩니다 AWS.는이 조직의 코드에 대한 정기적인 자동 및 수동 보안 검토를 AWS 수행하여 최선을 다해 보안을 확인합니다. AWS 정책에 따라 AWS 는 사용된 보안 검토 빈도, 접근 방식 또는 도구에 대한 정보를 공개하지 않습니다. 또한 ADDF에 대한 내부 감사 보고서를 게시 AWS 하지 않습니다. 그러나 확인된 보안 조사 결과는 풀 요청을 통해 긴급하게 수정되고 게시됩니다.

> **Note**
>
> 프레임워크로서의 ADDF는 Apache License 2.0(GitHub)에 명시된 바와 같이 어떠한 종류의 명시적 또는 묵시적 보증이나 조건(소유권, 비침해성, 상품성 또는 특정 목적에의 적합성에 대한 보증이나 조건을 포함하되 이에 국한되지 않음) 없이 '있는 그대로' 제공됩니다. 사용자는 ADDF에 대한 자체 보안 평가를 수행하고 ADDF가 조직의 특정 보안 요구 사항을 준수하는지 확인해야 하며, Apache License 2.0에 명시된 대로 ADDF 사용 또는 재배포의 적절성 확인을 전적으로 책임지고 해당 라이선스에 따른 행사 또는 권한과 관련된 모든 위험을 감수합니다.

오픈 소스 보안 검토 및 기여

ADDF는 기여를 환영하는 오픈 소스 프로젝트입니다. 프레임워크에 대한 자체 보안 검토를 수행하고 보안 관련 결과를 보고하여 기여할 수 있도록 모든 사용자를 초대합니다. 코드에서 문제를 발견하면 Security issue notifications(ADDF 설명서)의 지침을 따르세요.

ADDF 내장 보안 기능

ADDF(Autonomous Driving Data Framework)에는 다양한 보안 기능이 내장되어 있습니다. 기본적으로 이러한 기능은 보안 프레임워크를 설정하고 조직이 일반적인 엔터프라이즈 보안 요구 사항을 충족하는 데 도움이 되도록 설계되었습니다.

다음은 내장 보안 기능입니다.

- ADDF 모듈 코드에 대한 최소 권한
- 코드형 인프라
- IaC에 대한 자동 보안 검사
- AWS CDK 배포 역할에 대한 사용자 지정 최소 권한 정책
- 모듈 deployspec 파일에 대한 최소 권한 정책
- 데이터 암호화
- Secrets Manager를 사용하는 보안 인증 스토리지
- SeedFarmer 및 CodeSeeder의 보안 검토
- CodeSeeder AWS CodeBuild 역할에 대한 권한 경계 지원
- AWS 다중 계정 아키텍처
- 다중 계정 배포에 대한 최소 권한

ADDF 모듈 코드에 대한 최소 권한

최소 권한은 작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 최소 권한 적용을 참조하세요. ADDF에서 제공하는 모듈은 다음과 같이 코드와 배포된 리소스에 대한 최소 권한 원칙을 엄격하게 따릅니다.

- ADDF 모듈에 대해 생성된 모든 AWS Identity and Access Management (IAM) 정책에는 사용 사례에 필요한 최소 권한이 있습니다.
- AWS 서비스 는 최소 권한 원칙에 따라 구성되고 배포됩니다. ADDF 제공 모듈은 특정 사용 사례에 필요한 서비스와 서비스 기능만 사용합니다.

코드형 인프라

ADDF는 프레임워크로서 ADDF 모듈을 코드형 인프라(IaC)로 배포하도록 설계되었습니다. IaC는 수동 배포 프로세스를 없애고 수동 프로세스로 인해 발생할 수 있는 오류와 구성 오류를 방지하는 데 도움이 됩니다.

ADDF는 일반적인 IaC 프레임워크를 사용하여 모듈을 오케스트레이션하고 배포하도록 설계되었습니다. 여기에는 다음이 포함되지만 이에 국한되지는 않습니다.

- AWS Cloud Development Kit (AWS CDK)
- AWS CloudFormation
- Hashicorp Terraform

다양한 IaC 프레임워크를 사용하여 다양한 모듈을 작성한 다음 ADDF를 사용하여 모듈을 배포할 수 있습니다.

ADDF 모듈에서 사용하는 기본 IaC 프레임워크는 입니다 AWS CDK. AWS CDK 는 AWS 리소스를 강제로 정의하는 데 사용할 수 있는 상위 수준의 객체 지향 추상화입니다.는 AWS CDK 이미 다양한 서비스 및 시나리오에 대해 기본적으로 보안 모범 사례를 적용하고 있습니다. AWS CDK를 사용하면 보안 구성 오류의 위험이 줄어듭니다.

IaC에 대한 자동 보안 검사

오픈 소스 cdk-nag 유틸리티(GitHub)가 ADDF에 통합되었습니다. 이 유틸리티는 AWS CDK 를 기반으로 하는 ADDF 모듈을 자동으로 확인하여 일반 및 보안 모범 사례를 준수하는지 확인합니다. cdk-nag 유틸리티는 규칙과 규칙 팩을 사용하여 모범 사례를 위반하는 코드를 감지하고 보고합니다. 규칙과 전체 목록에 대한 자세한 내용은 cdk-nag rules(GitHub)를 참조하세요.

AWS CDK 배포 역할에 대한 사용자 지정 최소 권한 정책

ADDF는 AWS CDK v2를 광범위하게 사용합니다. 모든 ADDF를 부트스트랩해야 AWS 계정 합니다 AWS CDK. 자세한 내용은 Bootstrapping(AWS CDK 설명서)을 참조하세요.

기본적으로는 부트스트랩된 계정에서 생성된 AWS CDK 배포 역할에 허용 AdministratorAccess AWS 관리형 정책을 AWS CDK 할당합니다. 이 역할의 전체 이름은 입니다`cdk-[CDK_QUALIFIER]-cfn-exec-role-[AWS_ACCOUNT_ID]-[REGION]`.는이 역할을 AWS CDK 사용하여 배포 프로세스의 AWS 계정 일부로 부트스트래핑된에 리소스를 배포합니다 AWS CDK .

조직의 보안 요구 사항에 따라 `AdministratorAccess` 정책이 너무 허용적일 수 있습니다. AWS CDK 부트스트랩 프로세스의 일부로 필요에 따라 정책과 권한을 사용자 지정할 수 있습니다. `--cloudformation-execution-policies` 파라미터를 사용하여 새로 정의된 정책으로 계정을 다시 부트스트랩하면 정책을 변경할 수 있습니다. 자세한 내용은 부트스트래핑 사용자 지정(AWS CDK 문서화)을 참조하세요.

> **Note**
>
> 이 보안 기능은 ADDF에만 국한된 것은 아니지만 ADDF 배포의 전반적인 보안을 강화할 수 있으므로 이 섹션에 나열되어 있습니다.

모듈 deployspec 파일에 대한 최소 권한 정책

각 모듈에는 deployspec.yaml이라는 배포 사양 파일이 포함되어 있습니다. 이 파일은 모듈의 배포 지침을 정의합니다. CodeSeeder는 이를 사용하여를 사용하여 대상 계정에 정의된 모듈을 배포합니다 AWS CodeBuild. CodeSeeder는 배포 사양 파일의 지침에 따라 리소스를 배포하기 위해 CodeBuild에 기본 서비스 역할을 할당합니다 이 서비스 역할은 최소 권한 원칙에 따라 설계되었습니다. 모든 ADDF 제공 모듈은 AWS CDK 애플리케이션으로 생성되므로 AWS CDK 애플리케이션을 배포하는 데 필요한 모든 권한이 포함됩니다.

그러나 외부에서 스테이지 명령을 실행해야 하는 경우 CodeBuild의 기본 서비스 역할을 사용하는 대신 사용자 지정 IAM 정책을 생성 AWS CDK해야 합니다. 예를 들어 Terraform AWS CDK과 같은 이외의 IaC 배포 프레임워크를 사용하는 경우 해당 특정 프레임워크가 작동할 수 있는 충분한 권한을 부여하는 IAM 정책을 생성해야 합니다. 전용 IAM 정책이 필요한 또 다른 시나리오는 `install`, `pre_buildbuild`, 또는 `post_build` 스테이지 명령 AWS 서비스 에 다른에 대한 직접 AWS Command Line Interface (AWS CLI) 호출을 포함하는 경우입니다. 예를 들어, S3 버킷에 파일을 업로드하는 Amazon Simple Storage Service(S3) 명령이 모듈에 포함되어 있는 경우 사용자 지정 정책이 필요합니다. 사용자 지정 IAM 정책은 AWS CDK 배포 외부의 모든 AWS 명령에 대한 세분화된 제어를 제공합니다. ADDF 모듈에 대한 사용자 지정 IAM 정책을 생성할 때 최소 권한을 적용해야 합니다.

데이터 암호화

ADDF는 잠재적으로 민감한 데이터를 저장하고 처리합니다. 이 데이터를 보호하기 위해 SeedFarmer, CodeSeeder 및 ADDF 제공 모듈은 사용된 모든에 대해 저장 및 전송 중인 데이터를 암호화합니다 AWS 서비스 (`demo-only`폴더의 모듈에 대해 달리 명시적으로 명시되지 않는 한).

Secrets Manager를 사용하는 보안 인증 스토리지

ADDF는 Docker Hub, JupyterHub 및 Amazon Redshift와 같은 다양한 서비스에 대한 다양한 시크릿을 처리하며, AWS Secrets Manager를 사용하여 ADDF 관련 시크릿을 저장합니다. 이렇게 하면 소스 코드에서 민감한 데이터를 제거할 수 있습니다.

Secrets Manager 시크릿은 대상 계정이 제대로 작동하는 데 필요한 경우에만 대상 계정에 저장됩니다. 기본적으로 도구 체인 계정에는 시크릿이 포함되어 있지 않습니다.

SeedFarmer 및 CodeSeeder의 보안 검토

SeedFarmer와 CodeSeeder(GitHub 리포지토리)는 ADDF 및 해당 ADDF 모듈을 배포하는 데 사용됩니다. 이러한 오픈 소스 프로젝트는에 설명된 대로 ADDF와 동일한 정기적인 AWS 내부 보안 검토 프로세스를 거칩니다ADDF 보안 검토 프로세스.

CodeSeeder AWS CodeBuild 역할에 대한 권한 경계 지원

IAM 권한 경계는 ID 기반 정책이 IAM 엔터티에 부여할 수 있는 최대 권한을 정의하는 일반 보안 메커니즘입니다. SeedFarmer와 CodeSeeder는 각 대상 계정에 대한 IAM 권한 경계 연결을 지원합니다. 권한 경계는 CodeSeeder가 모듈을 배포할 때 CodeBuild에서 사용하는 모든 서비스 역할의 최대 권한을 제한합니다. 보안 팀이 ADDF 외부에서 IAM 권한 경계를 생성해야 합니다. IAM 권한 경계 정책 연결은 ADDF 배포 매니페스트 파일인 deployment.yaml 내의 속성으로 허용됩니다. 자세한 내용은 권한 경계(SeedFarmer 설명서)를 참조하세요.SeedFarmer

워크플로는 다음과 같습니다.

1. 보안 팀이 보안 요구 사항에 따라 IAM 권한 경계를 정의하고 생성합니다. IAM 권한 경계는 각 ADDF에서 개별적으로 생성해야 합니다 AWS 계정. 출력은 권한 경계 정책 Amazon 리소스 이름(ARN) 목록입니다.
2. 보안 팀이 ADDF 개발자 팀과 정책 ARN 목록을 공유합니다.
3. ADDF 개발자 팀이 매니페스트 파일에 정책 ARN 목록을 통합합니다. 이 통합의 예는 sample-permissionboundary.yaml(GitHub)을 참조하세요.
4. 배포에 성공하면 CodeBuild가 모듈을 배포하는 데 사용하는 모든 서비스 역할에 권한 경계가 연결됩니다.
5. 보안 팀이 필요에 따라 권한 경계가 적용되는지 모니터링합니다.

AWS 다중 계정 아키텍처

AWS Well-Architected Framework의 보안 원칙에 정의된 대로 조직의 요구 사항에 AWS 계정따라 리소스와 워크로드를 여러 로 분리하는 것이 모범 사례로 간주됩니다. 이는가 격리 경계 AWS 계정 역할을 하기 때문입니다. 자세한 내용은 AWS 계정 관리 및 분리를 참조하세요. 이 개념을 구현한 것이 다중 계정 아키텍처입니다. 적절하게 설계된 AWS 다중 계정 아키텍처는 워크로드 분류를 제공하며 보안 침해 발생 시 단일 계정 아키텍처보다 영향 범위가 작습니다.

ADDF는 기본적으로 AWS 다중 계정 아키텍처를 지원합니다. 조직의 보안 및 separation-of-duties 요구 사항에 필요한 AWS 계정 만큼 ADDF 모듈을 배포할 수 있습니다. 도구 체인과 대상 계정 기능을 결합하여 단일 AWS 계정에 ADDF를 배포할 수 있습니다. 또는 ADDF 모듈이나 모듈 그룹에 대한 개별 대상 계정을 생성할 수 있습니다.

고려해야 할 유일한 제한 사항은 ADDF 모듈이 각 모듈에 대한 가장 작은 배포 단위를 나타낸다는 것입니다 AWS 계정.

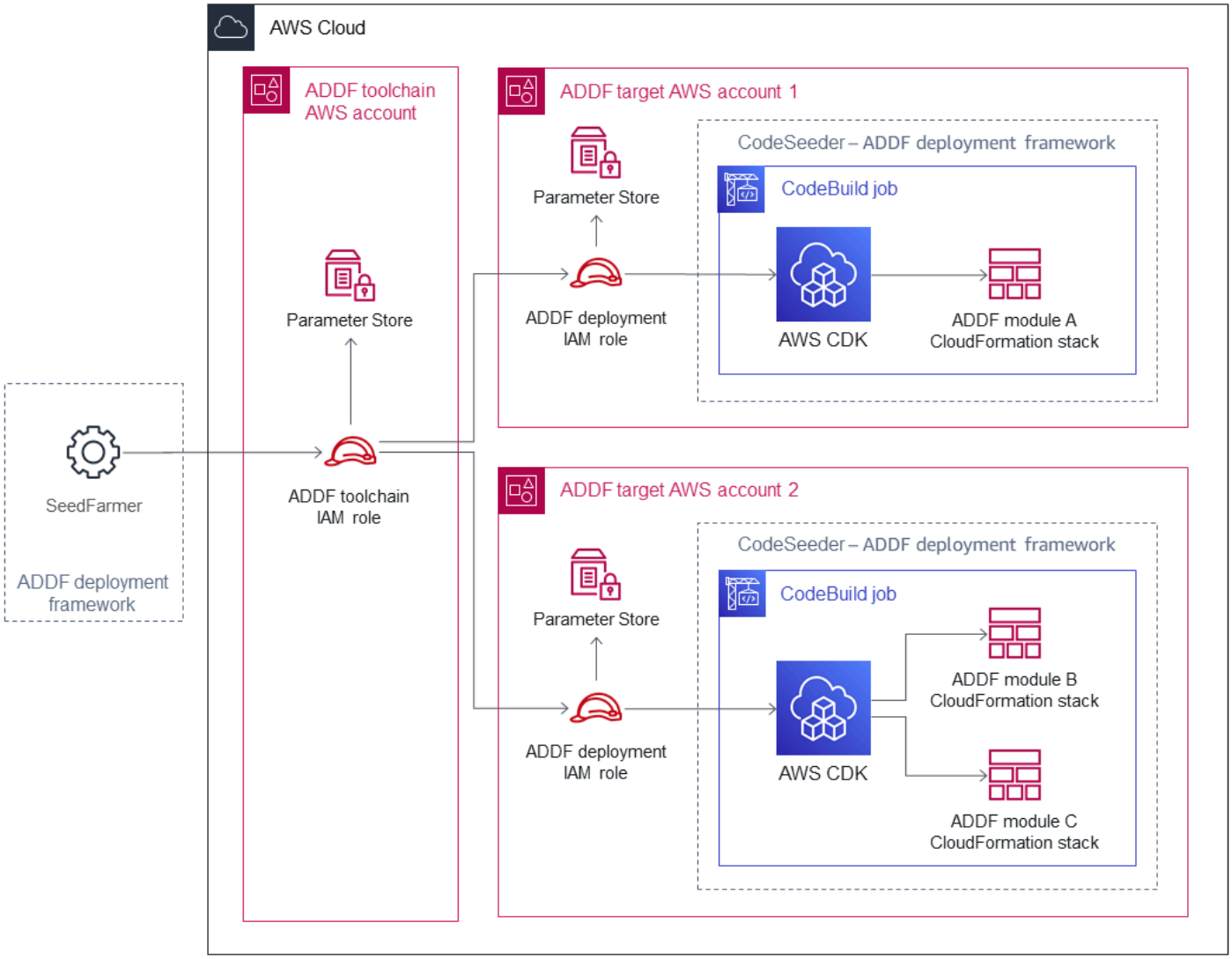
프로덕션 환경에서는 도구 체인 계정과 하나 이상의 대상 계정으로 구성된 다중 계정 아키텍처를 사용하는 것이 좋습니다. 자세한 내용은 ADDF 아키텍처 단원을 참조하십시오.

다중 계정 배포에 대한 최소 권한

다중 계정 아키텍처를 사용하는 경우 SeedFarmer는 대상 계정에 액세스하여 다음 세 가지 작업을 수행해야 합니다.

1. 도구 체인 계정과 대상 계정에 ADDF 모듈 메타데이터를 씁니다.
2. 도구 체인 계정과 대상 계정에서 현재 ADDF 모듈 메타데이터를 읽습니다.
3. 모듈을 배포하거나 업데이트할 목적으로 대상 계정에서 AWS CodeBuild 작업을 시작합니다.

다음 그림은 ADDF별 AWS Identity and Access Management (IAM) 역할을 수임하기 위한 작업을 포함하여 교차 계정 관계를 보여줍니다.



이러한 크로스 계정 작업은 잘 정의된 역할 수임 작업을 통해 수행됩니다.

- ADDF 도구 체인 IAM 역할이 도구 체인 계정에 배포됩니다. SeedFarmer가 이 역할을 수임합니다. 이 역할에는 `iam:AssumeRole` 작업을 수행할 수 있는 권한이 있으며 각 대상 계정에서 ADDF 배포 IAM 역할을 수임할 수 있습니다. 또한 ADDF 도구 체인 IAM 역할은 로컬 AWS Systems Manager 파라미터 스토어 작업을 실행할 수 있습니다.
- ADDF 배포 IAM 역할이 각 대상 계정에 배포됩니다. 이 역할은 ADDF 도구 체인 IAM 역할을 사용하여 도구 체인 계정에서만 수임할 수 있습니다. 이 역할에는 로컬 AWS Systems Manager 파라미터 스토어 작업을 실행할 수 있는 권한이 있으며 CodeSeeder를 통해 CodeBuild AWS CodeBuild 작업을 시작하고 설명하는 작업을 실행할 수 있는 권한이 있습니다.

이러한 ADDF 관련 IAM 역할은 ADDF 부트스트래핑 프로세스의 일부로 생성됩니다. 자세한 내용은 ADDF 배포 가이드 AWS 계정(GitHub)의 Bootstrap(s)을 참조하세요.GitHub

모든 크로스 계정 권한은 최소 권한 원칙에 따라 설정됩니다. 한 대상 계정이 손상되어도 다른 ADDF AWS 계정에 미치는 영향은 거의 없거나 전혀 없습니다.

ADDF에 대한 단일 계정 아키텍처의 경우 역할 관계는 동일하게 유지되며, 단일 AWS 계정으로 축소될 뿐입니다.

ADDF 보안 설정 및 운영

ADDF(Autonomous Driving Data Framework)는 조직의 전담 DevOps 및 보안 팀의 지속적 유지 보수와 관리가 필요한 맞춤형 소프트웨어로 취급되어야 합니다. 이 섹션에서는 수명 주기 전반에 걸쳐 ADDF를 설정하고 운영하는 데 도움이 되는 보안 관련 일반 작업을 설명합니다.

이 섹션에는 다음 태스크가 포함되어 있습니다.

- ADDF 아키텍처 정의
- 초기 설정
- ADDF 배포 프레임워크 코드 사용자 지정
- ADDF에서 사용자 지정 모듈 작성
- 반복되는 ADDF 배포
- 반복되는 보안 감사
- ADDF 업데이트
- 서비스 해제

ADDF 아키텍처 정의

ADDF 인스턴스는 배포된 AWS 계정 환경만큼만 안전합니다. 이 AWS 계정 환경은 특정 사용 사례의 보안 및 운영 요구 사항을 충족하도록 설계되어야 합니다. 예를 들어, 개념 증명(PoC) 환경에서 ADDF 인스턴스를 설정하기 위한 보안 및 운영 관련 작업과 고려 사항은 프로덕션 환경에서 ADDF를 설정하는 경우와 다릅니다.

PoC 환경에서 ADDF 실행

PoC 환경에서 ADDF를 사용하려는 경우 다른 워크로드가 포함되지 않은 ADDF AWS 계정 전용를 생성하는 것이 좋습니다. 이렇게 하면 ADDF와 해당 기능을 탐색하는 동안 계정을 안전하게 유지할 수 있습니다. 이 접근 방식은 다음과 같은 이점이 있습니다.

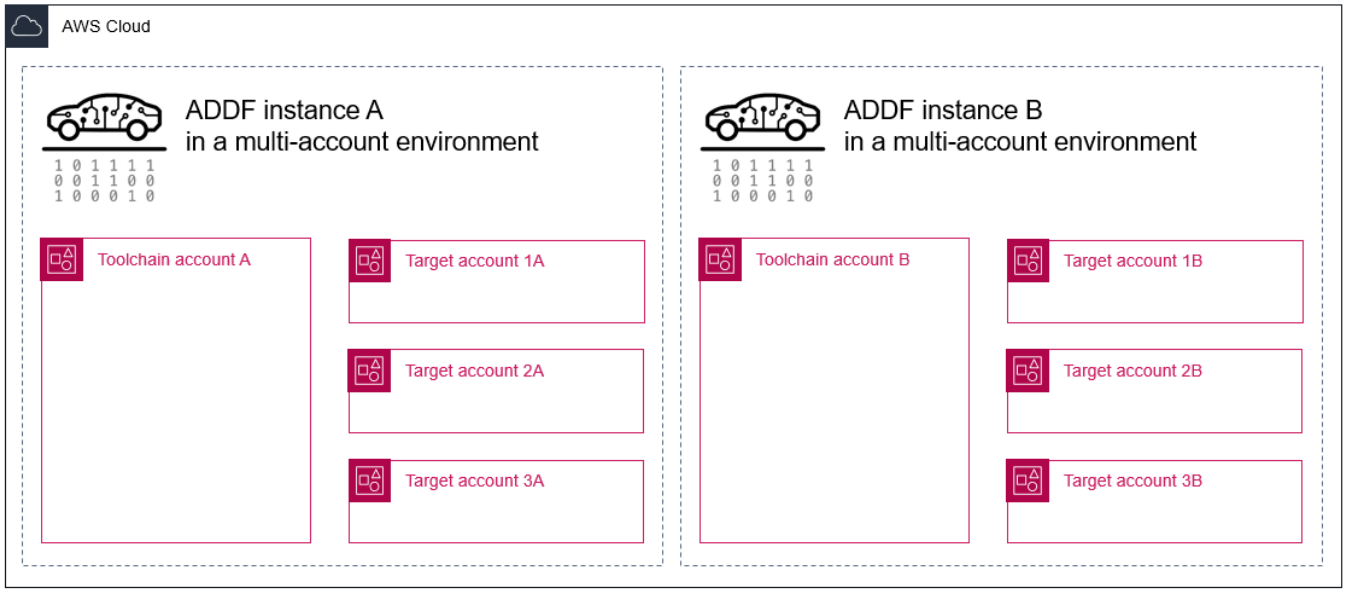
- 심각한 ADDF 구성 오류가 발생하더라도 다른 워크로드에 부정적인 영향이 없습니다.
- ADDF 설정에 부정적인 영향을 줄 수 있는 다른 워크로드 구성 오류의 위험은 없습니다.

PoC 환경의 경우에도 프로덕션 환경에서 ADDF 실행에 나열된 모범 사례를 최대한 많이 따르는 것이 좋습니다.

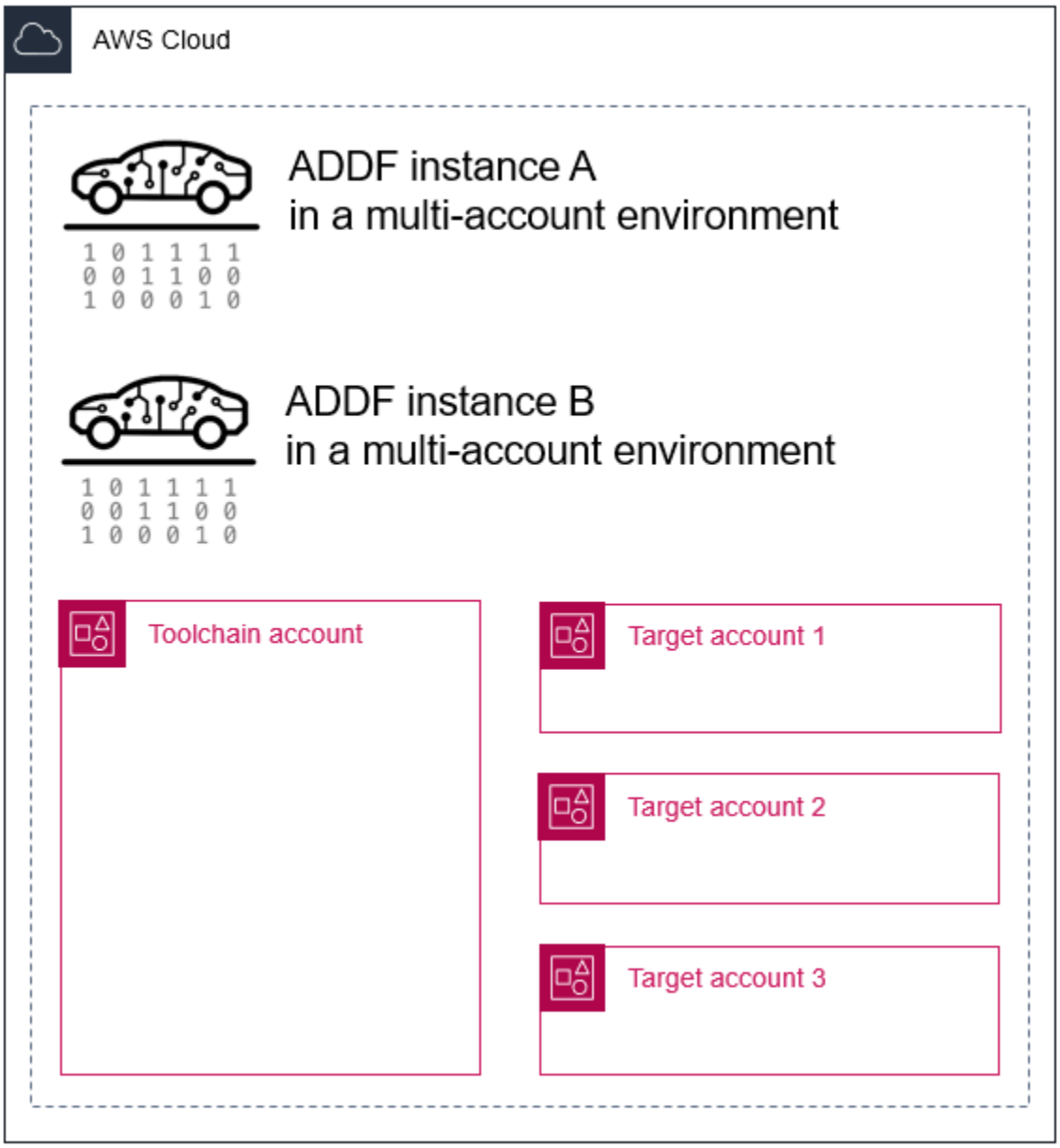
프로덕션 환경에서 ADDF 실행

엔터프라이즈 프로덕션 환경에서 ADDF를 사용하려는 경우 조직의 보안 모범 사례를 고려하여 그에 따라 ADDF를 구현하는 것이 좋습니다. 조직의 보안 모범 사례 외에도 다음을 구현하는 것이 좋습니다.

- 장기적이고 헌신적인 ADDF DevOps 팀 구성 – ADDF를 맞춤형 소프트웨어로 취급해야 합니다. 전담 DevOps 팀의 지속적 유지 보수 및 관리가 필요합니다. 프로덕션 환경에서 ADDF 실행을 시작하기 전에 ADDF 배포 수명이 끝날 때까지 전체 리소스를 투입하여 충분한 규모와 기능을 갖춘 DevOps 팀을 정의해야 합니다.
- 다중 계정 아키텍처 사용 – 관련 없는 다른 워크로드 없이 자체 전용 AWS 다중 계정 환경에 각 ADDF 인스턴스를 배포해야 합니다. AWS 계정 관리 및 분리(AWS Well-Architected Framework)에 정의된 대로 조직의 요구 사항에 AWS 계정따라 리소스와 워크로드를 여러 로 분리하는 것이 모범 사례로 간주됩니다. 이는가 격리 경계 AWS 계정 역할을 하기 때문입니다. 적절하게 설계된 AWS 다중 계정 아키텍처는 워크로드 분류를 제공하며 보안 침해 발생 시 단일 계정 아키텍처보다 영향 범위가 작습니다. 다중 계정 아키텍처를 사용하면 계정을 AWS 서비스 할당량 내로 유지할 수 있습니다 조직의 보안 및 업무 분리 요구 사항을 충족하는 데 필요한 만큼의 AWS 계정 에 ADDF 모듈을 배포합니다.
- 여러 ADDF 인스턴스 배포 – 조직의 소프트웨어 개발 프로세스에 따라 ADDF 모듈을 적절히 개발, 테스트 및 배포하기 위해 필요한 만큼 별도의 ADDF 인스턴스를 설정합니다. 여러 ADDF 인스턴스를 설정할 때 다음 접근 방식 중 하나를 사용할 수 있습니다.
  - 서로 다른 AWS 다중 계정 환경의 여러 ADDF 인스턴스 - 별도의를 사용하여 서로 다른 ADDF 인스턴스 AWS 계정 를 격리할 수 있습니다. 예를 들어, 조직에 전용 개발, 테스트 및 프로덕션 단계가 있는 경우 각 단계에 대해 별도의 ADDF 인스턴스와 전용 계정을 생성할 수 있습니다. 이렇게 하면 오류가 여러 단계로 전파될 위험이 줄고, 승인 프로세스를 구현하는 데 도움이 되고, 사용자 액세스가 특정 환경으로만 제한되는 등 많은 이점이 있습니다. 다음 이미지는 별도의 다중 계정 환경에 배포된 2개의 ADDF 인스턴스를 보여줍니다.



- 동일한 AWS 다중 계정 환경의 여러 ADDF 인스턴스 - 동일한 AWS 다중 계정 환경을 공유하는 여러 ADDF 인스턴스를 생성할 수 있습니다. 이렇게 하면 동일한 AWS 계정에 격리된 브랜치가 효과적으로 생성됩니다. 예를 들어, 여러 개발자가 동시에 작업하는 경우 개발자는 동일한 AWS 계정에 전용 ADDF 인스턴스를 생성할 수 있습니다. 이를 통해 개발자는 개발 및 테스트 목적으로 격리된 브랜치에서 작업할 수 있습니다. 이 접근 방식을 사용하는 경우 각 ADDF 인스턴스마다 ADDF 리소스에 고유한 리소스 이름이 있어야 합니다. 이는 ADDF 사전 제공 모듈에서 기본적으로 지원됩니다. AWS 서비스 할당량을 초과하지 않는 한 이 접근 방식을 사용할 수 있습니다. 다음 이미지는 공유 다중 계정 환경에 배포된 2개의 ADDF 인스턴스를 보여줍니다.



- 동일한 AWS 단일 계정 환경의 여러 ADDF 인스턴스 – 이 아키텍처는 이전 예와 매우 유사합니다. 차이점은 여러 ADDF 인스턴스가 다중 계정 환경 대신 단일 계정 환경에 배포된다는 것입니다. 이 아키텍처는 범위가 매우 제한적이고 여러 개발자가 동시에 여러 브랜치에서 작업하는 매우 간단한 ADDF 사용 사례에 적합합니다.

SeedFarmer는 ADDF 인스턴스의 배포를 제어하는 단일 도구이므로 조직의 배포 전략 및 CI/CD 프로세스에 맞는 모든 환경과 계정 아키텍처를 구축할 수 있습니다.

- 조직의 보안 요구 사항에 따라 AWS Cloud Development Kit (AWS CDK) 부트스트랩 프로세스 사용자 지정 - 기본적으로 부트스트랩 프로세스 중에 AdministratorAccess AWS 관리형 정책을 AWS CDK 할당합니다. 이 정책은 전체 관리 권한을 부여합니다. 이 정책이 조직의 보안 요구 사항에 비해 너무 허용적인 경우 적용되는 정책을 사용자 지정할 수 있습니다. 자세한 내용은 AWS CDK 배포 역할에 대한 사용자 지정 최소 권한 정책 단원을 참조하십시오.
- IAM에서 액세스를 설정할 때 모범 사례 준수 - 사용자가 ADDF에 액세스할 수 있는 구조화된 AWS Identity and Access Management (IAM) 액세스 솔루션을 설정합니다 AWS 계정. ADDF의 프레임워크는 최소 권한 원칙을 준수하도록 설계되었습니다. IAM 액세스 패턴은 최소 권한 원칙을 따르고 조직의 요구 사항과 IAM의 보안 모범 사례(IAM 설명서)를 준수해야 합니다.
- 조직의 모범 사례에 따라 네트워킹 설정 - ADDF에는 기본 퍼블릭 또는 프라이빗 Virtual Private Cloud(VPC)를 생성하는 선택적 네트워킹 AWS CloudFormation 스택이 포함되어 있습니다. 조직의 구성에 따라 이 VPC는 리소스를 인터넷에 직접 노출할 수 있습니다. 조직의 네트워킹 모범 사례를 따르고 보안이 강화된 사용자 지정 네트워크 모듈을 생성하는 것이 좋습니다.

- 보안 예방, 탐지 및 완화 조치를 AWS 계정 수준에서 배포 - AWS 는 Amazon GuardDuty, AWS Security Hub CSPM Amazon Detective 및와 같은 다양한 보안 서비스를 제공합니다 AWS Config. ADDF에서 이러한 서비스를 활성화 AWS 계정 하고 조직의 보안 방지, 탐지, 완화 및 인시던트 처리 프로세스를 통합합니다. 보안, 자격 증명 및 규정 준수 모범 사례(AWS 아키텍처 센터)와 해당 서비스의 설명서에 포함된 서비스별 권장 사항을 따르는 것이 좋습니다. 자세한 내용은 AWS 보안 설명서를 참조하세요.

구현 및 구성 세부 정보는 조직의 특정 요구 사항 및 프로세스에 따라 크게 달라지기 때문에 ADDF에서는 이러한 주제를 다루지 않습니다. 대신 이러한 주제를 다루는 것이 조직의 핵심적인 책임입니다. 일반적으로 AWS 랜딩 존을 관리하는 팀은 ADDF 환경을 계획하고 구현하는 데 도움을 줍니다.

초기 설정

ADDF Deployment Guide(GitHub)에 따라 ADDF를 설정합니다. 모든 배포의 시작점은 autonomous-driving-data-framework Git Hub 리포지토리의 `/manifest` 폴더입니다. `/manifest/example-dev` 폴더에는 데모용 샘플 배포가 포함되어 있습니다. 이 샘플을 자신의 배포를 설계하는 시작점으로 사용합니다. 해당 디렉터리에는 deployment.yaml이라는 ADDF 배포 매니페스트 파일이 있습니다. 여기에는 SeedFarmer가 AWS 클라우드에서 ADDF 및 해당 리소스를 관리, 배포 또는 삭제하는 데 필요한 모든 정보가 포함되어 있습니다. 전용 파일에 ADDF 모듈 그룹을 생성할 수 있습니다. core-modules.yaml은 코어 모듈 그룹의 한 예로 ADDF에서 제공하는 모든 코어 모듈을 포함합니다. 요약하면 deployment.yaml 파일은 대상 계정에 배포될 그룹과 모듈에 대한 모든 참조를 포함하며 배포 순서를 지정합니다.

특히 개념 증명을 위한 것이 아닌 환경에서는 안전하고 규정을 준수하는 구성의 경우 배포하려는 각 모듈의 소스 코드를 검토하는 것이 좋습니다. 보안 강화 모범 사례에 따르면 의도한 사용 사례에 필요한 모듈만 배포해야 합니다.

> **Note**
>
> `modules/demo-only/` 폴더의 ADDF 모듈은 보안이 강화되지 않았으므로 프로덕션 환경 또는 민감하거나 보호되는 데이터가 있는 환경에 배포해서는 안 됩니다. 이러한 모듈은 시스템 기능을 보여주기 위해 포함되어 있으며, 보안이 강화된 사용자 지정 자체 모듈을 생성하기 위한 기반으로 사용할 수 있습니다.

ADDF 배포 프레임워크 코드 사용자 지정

ADDF 배포 프레임워크와 해당 오케스트레이션 및 배포 로직은 모든 요구 사항에 맞게 완전히 사용자 지정할 수 있습니다. 하지만 다음과 같은 이유로 사용자 지정을 삼가거나 변경 내용을 최소화하는 것이 좋습니다.

- 업스트림 호환성 유지 - 업스트림 호환성으로 최신 기능 및 보안 업데이트를 위해 ADDF를 보다 쉽게 업데이트할 수 있습니다. 프레임워크를 변경하면 SeedFarmer, CodeSeeder 및 모든 ADDF 코어 모듈과의 기본 하위 버전 호환성이 중단됩니다.
- 보안 결과 – ADDF 배포 프레임워크를 변경하는 것은 의도하지 않은 보안 결과를 초래할 수 있는 복잡한 작업일 수 있습니다. 최악의 시나리오에서는 프레임워크 변경으로 인해 보안이 취약해질 수 있습니다.

가능하면 ADDF 배포 프레임워크와 ADDF 코어 모듈 코드를 수정하는 대신 자체 모듈 코드를 빌드하고 사용자 지정합니다.

> **Note**
>
> ADDF 배포 프레임워크의 특정 부분에 개선이나 추가 보안 강화가 필요하다고 생각되면 풀 요청을 통해 ADDF 리포지토리에 변경 사항을 제공하세요. 자세한 내용은 오픈 소스 보안 검토 및 기여 단원을 참조하십시오.

ADDF에서 사용자 지정 모듈 작성

새로운 ADDF 모듈을 생성하거나 기존 모듈을 확장하는 것이 ADDF의 핵심 개념입니다. 모듈을 생성하거나 사용자 지정할 때 일반적인 AWS 보안 모범 사례와 안전한 코딩을 위한 조직의 모범 사례를 따르는 것이 좋습니다. 또한 조직의 보안 요구 사항에 따라 초기 및 주기적인 내부 또는 외부 기술 보안 검토를 수행하여 보안 문제의 위험을 더욱 줄이는 것이 좋습니다.

반복되는 ADDF 배포

ADDF Deployment Guide(GitHub)에 설명된 대로 ADDF와 해당 모듈을 배포합니다. 대상 계정에서 리소스를 추가, 업데이트 또는 제거하는 반복적인 ADDF 배포를 지원하기 위해 SeedFarmer는 도구 체인 및 대상 계정의 Parameter Store에 저장된 MD5 해시를 사용하여 현재 배포된 인프라를 로컬 코드 베이스의 매니페스트 파일에 정의된 인프라와 비교합니다.

이 접근 방식은 소스 리포지토리(SeedFarmer를 운영하는 로컬 코드 베이스)가 신뢰할 수 있는 소스이고 여기에 명시적으로 선언된 인프라가 배포에서 원하는 결과인 GitOps 패러다임을 따릅니다. GitOps에 대한 자세한 내용은 What is GitOps(GitLab 웹 사이트)를 참조하세요.

반복되는 보안 감사

조직의 다른 소프트웨어와 마찬가지로 ADDF와 사용자 지정 ADDF 모듈 코드를 보안 위험 관리, 보안 검토 및 보안 감사 주기에 통합합니다.

ADDF 업데이트

ADDF는 지속적인 개발 노력의 일환으로 정기적인 업데이트를 받습니다. 여기에는 기능 업데이트, 보안 관련 개선 및 수정이 포함됩니다. 정기적으로 새 프레임워크 릴리스를 확인하고 적시에 업데이트를 적용하는 것이 좋습니다. 자세한 내용은 Steps to update ADDF(ADDF 설명서)를 참조하세요.

서비스 해제

더 이상 필요 없는 경우 AWS 계정에서 ADDF와 모든 관련 리소스를 삭제합니다. 사용되지 않는 무인 인프라는 불필요한 비용과 잠재적인 보안 위험을 초래합니다. 자세한 내용은 Steps to destroy ADDF(ADDF 설명서)를 참조하세요.

다음 단계

이 가이드에서는 AWS 클라우드 환경에 자율 주행 데이터 프레임워크(ADDF)를 배포할 때의 보안 및 운영 모범 사례와 고려 사항을 검토했습니다. 이 안내서에서는 ADDF 사용자, ADDF 코어 팀 및 간의 공동 책임 모델을 검토하여 ADDF를 안전하게 설정하고 운영 AWS 하기 위한 역할과 책임을 이해합니다. 환경별 권장 사항을 비롯하여 수명 주기 전반에 걸쳐 ADDF를 안전하게 운영하기 위한 권장 사항도 포함되어 있습니다.

리소스 섹션의 리소스를 숙지하는 것이 좋습니다. 준비가 되면 ADDF Deployment Guide(GitHub)의 지침에 따라 ADDF를 설정할 수 있습니다.

ADDF를 설정하고 운영하면서 배포 프레임워크의 개선이나 추가 보안 강화가 필요하다고 생각되면 풀 요청을 통해 ADDF 리포지토리에 변경 사항을 제공하세요. 자세한 내용은 오픈 소스 보안 검토 및 기여 단원을 참조하십시오.

리소스

AWS 설명서

- 에서 ADDF를 사용하여 사용자 지정 워크플로 개발 및 배포 AWS(AWS 블로그 게시물)
- AWS 보안 서비스 설명서
- IAM의 보안 모범 사례
- AWS 계정 관리 및 분리
- 에 대한 부트스트래핑 AWS CDK
- AWS 공동 책임 모델
- AWS Well-Architected 프레임워크

오픈 소스 리소스

- ADDF 리포지토리(GitHub)
- ADDF Deployment Guide(GitHub)
- CodeSeeder 리포지토리(GitHub)
- SeedFarmer 리포지토리(GitHub)

고지 사항

고객은 본 문서의 정보를 독립적으로 평가할 책임이 있습니다. 이 문서: (a) 정보 제공 목적으로만 사용되며, (b) 예고 없이 변경될 수 있는 현재 AWS 제품 제공 및 관행을 나타내며, (c) AWS 및 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 약속이나 보장도 생성하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 표현 또는 조건 없이 “있는 그대로” 제공됩니다.

고객에 AWS 대한의 책임과 책임은 계약에 의해 AWS 관리되며,이 문서는 AWS 와 고객 간의 계약의 일부이거나 수정하지 않습니다.

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
최초 게시	—	2022년 11월 15일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.