사용자 가이드

AWS CloudShell

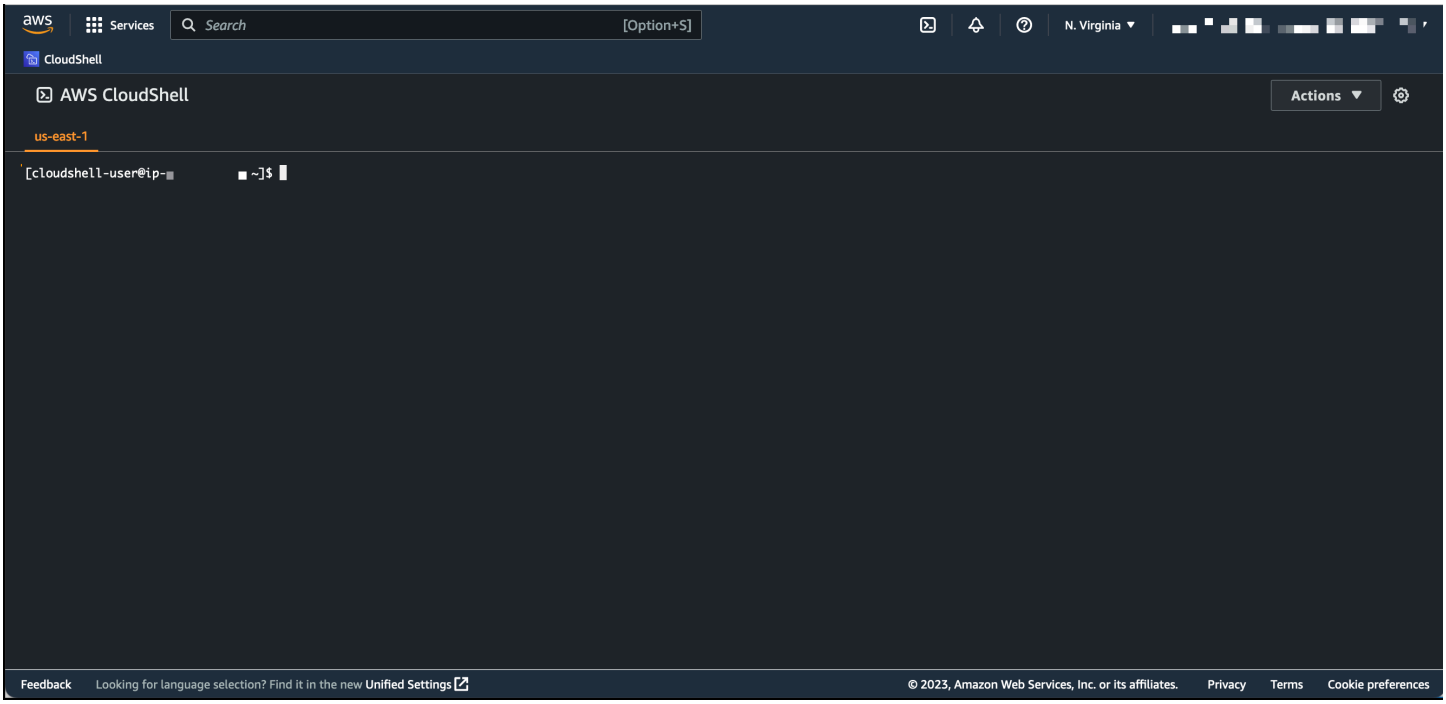
AWS CloudShell: 사용자 가이드

Table of Contents

이게 뭐야 AWS CloudShell?

AWS CloudShell 에서 직접 실행할 수 있는 브라우저 기반의 사전 인증된 셸입니다. AWS Management Console몇 가지 방법으로 탐색할 CloudShell 수 AWS Management Console 있습니다. 자세한 내용은 AWS CloudShell시작하는 방법을 참조하세요.

선호하는 셸 (예: Bash PowerShell, 또는) 을 사용하여 AWS CLI 명령을 실행할 수 Z shell 있습니다. 명령줄 도구를 다운로드하거나 설치할 필요 없이 이 작업을 수행할 수 있습니다.



시작하면 AWS CloudShell Amazon Linux 2023을 기반으로 하는 컴퓨팅 환경이 생성됩니다. 이 환경에서는 광범위한 사전 설치 개발 도구, 파일 업로드 및 다운로드 옵션, 세션 간 영구 파일 스토리지에 액세스할 수 있습니다.

(지금 사용해 보십시오: AWS CloudShell 시작하기)

AWS CloudShell features

이 주제에서는 CloudShell 콘솔에서 시작하고, 선호하는 명령줄 셸 간에 원활하게 전환하고, 원하는 대로 사용자 지정하는 CloudShell 방법을 설명합니다. 또한 각각 최대 1GB의 영구 스토리지를 사용할 수 있으며 특정 보안 AWS 리전기능으로 CloudShell 환경을 보호하는 방법도 확인할 수 있습니다.

AWS Command Line Interface

AWS CloudShell 에서 실행할 수 있습니다 AWS Management Console. 콘솔에 로그인하는 데 사용한 AWS 자격 증명은 새 셸 세션에서 자동으로 사용할 수 있습니다. AWS CloudShell 사용자는 사전 인증을 받았기 때문에 버전 2를 AWS 서비스 사용하여 AWS CLI 상호 작용할 때 자격 증명을 구성하지 않아도 됩니다. 셸의 컴퓨팅 환경에 사전 AWS CLI 설치되어 있습니다.

명령줄 인터페이스 AWS 서비스 사용과 상호 작용하는 방법에 대한 자세한 내용은 을 참조하십시오. 다음 지역AWS 서비스 이용AWS CloudShell

쉘 및 개발 도구

AWS CloudShell 세션용으로 생성된 셸을 사용하면 원하는 명령줄 셸 간에 원활하게 전환할 수 있습니다. 좀 더 구체적으로 말하자면,, Bash PowerShell, 사이를 전환할 수 있습니다. Z shell 다음과 같은 다른 도구 및 유틸리티에 대한 액세스 권한도 있습니다. 이러한 클레임에는 git, make, pip, sudo,tar, tmux, vim, wget, zip이 포함됩니다.

셸 환경은 Node.js, Python 등 여러 주요 소프트웨어 언어를 지원하도록 사전 구성되어 있습니다. 즉, 예를 들어 런타임 설치를 먼저 수행하지 않고도 Node.js 및 Python 프로젝트를 실행할 수 있습니다. PowerShell 사용자는 .NET Core 런타임을 사용할 수 있습니다.

로컬 리포지토리에서 만들거나 로컬 리포지토리에 업로드한 파일을 에서 관리하는 원격 리포지토리로 푸시하기 전에 커밋할 수 있습니다. AWS CloudShell AWS CodeCommit

자세한 설명은 AWS CloudShell 컴퓨팅 환경: 사양 및 소프트웨어 섹션을 참조하세요.

영구 스토리지

를 사용하면 추가 비용 없이 각각 AWS 리전 최대 1GB의 영구 스토리지를 사용할 수 있습니다. AWS CloudShell영구 스토리지는 홈 디렉터리($HOME)에 있으며 사용자만 이용할 수 있습니다. 각 쉘 세션이 종료된 후 삭제되는 임시 환경 리소스와 달리, 홈 디렉터리의 데이터는 세션 간에 유지됩니다.

영구 스토리지의 데이터 보존에 대한 자세한 정보는 영구 스토리지에서 확인하십시오.

보안

AWS CloudShell 환경과 사용자는 특정 보안 기능으로 보호됩니다. 여기에는 IAM 권한 관리, 쉘 세션 제한, 텍스트 입력 시 안전한 붙여넣기 등의 기능이 포함됩니다.

IAM을 통한 권한 관리

관리자는 IAM 정책을 사용하여 AWS CloudShell 사용자에게 권한을 부여하거나 거부할 수 있습니다. 또한 사용자가 쉘 환경에서 수행할 수 있는 특정 작업을 지정하는 정책을 생성할 수 있습니다. 자세한 내용은 IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리 섹션을 참조하십시오.

쉘 세션 관리

비활성 및 장기 실행 세션은 자동으로 중단되고 재활용됩니다. 자세한 내용은 셸 세션 섹션을 참조하십시오.

텍스트 입력용 안전한 붙여넣기

안전한 붙여넣기는 기본적으로 활성화됩니다. 이 보안 기능을 사용하려면 쉘에 붙여넣으려는 여러 줄 텍스트에 악성 스크립트가 포함되어 있는지 확인해야 합니다. 자세한 내용은 여러 줄 텍스트에 안전 붙여넣기 사용 섹션을 참조하십시오.

사용자 지정 옵션

AWS CloudShell 환경을 원하는 대로 사용자 지정할 수 있습니다. 예를 들어, 화면 레이아웃(다중 탭), 표시된 텍스트 크기 변경이 가능하고, 밝은 인터페이스 테마와 어두운 인터페이스 테마 간 전환이 가능합니다. 자세한 내용은 나만의 커스터마이징AWS CloudShell경험 섹션을 참조하십시오.

자체 소프트웨어 설치 및 시작 쉘 스크립트 수정을 통해 쉘 환경을 확장할 수도 있습니다.

세션 복원

세션 복원 기능은 CloudShell 터미널의 하나 또는 여러 브라우저 탭에서 실행하던 세션을 복원합니다. 최근에 닫은 브라우저 탭을 새로고침하거나 다시 열면 이 기능은 비활성 세션으로 인해 쉘이 중단될 때까지 세션을 재개합니다. CloudShell 세션을 계속 사용하려면 터미널 창에서 아무 키나 누르십시오. 쉘 세션에 대한 자세한 정보는 쉘 세션에서 확인하십시오.

또한 세션 복원은 개별 터미널 탭에서 최신 터미널 출력 및 실행 중인 프로세스를 복원합니다.

> **Note**
>
> 모바일 애플리케이션에서는 세션 복원을 사용할 수 없습니다.

AWS CloudShell요금

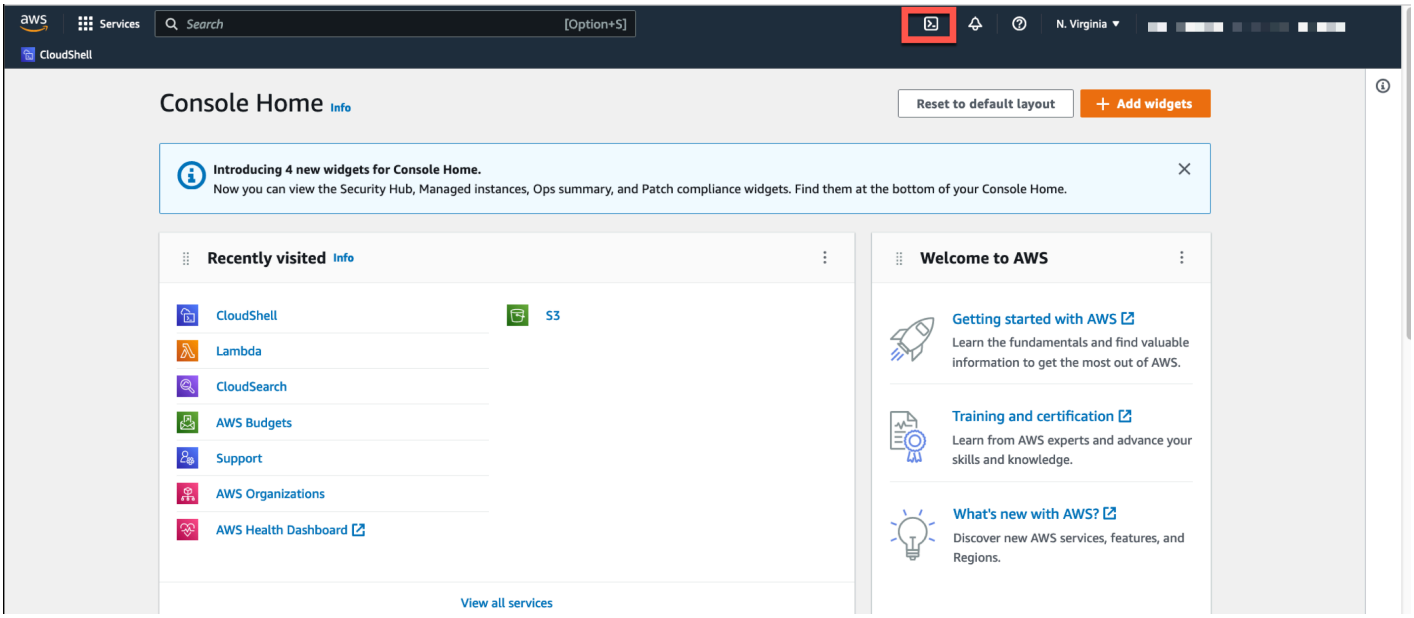
AWS CloudShell AWS 서비스 는 추가 비용 없이 이용할 수 있습니다. 하지만 함께 실행하는 다른 AWS 리소스에 대해서는 비용을 지불해야 AWS CloudShell합니다. 또한, 표준 데이터 전송 요금도 적용됩니다. 자세한 내용은 AWS CloudShell 요금을 참조하십시오.

자세한 설명은 에 대한 서비스 할당량 및 제한AWS CloudShell 섹션을 참조하세요.

어떻게 시작해야 AWS CloudShell할까요?

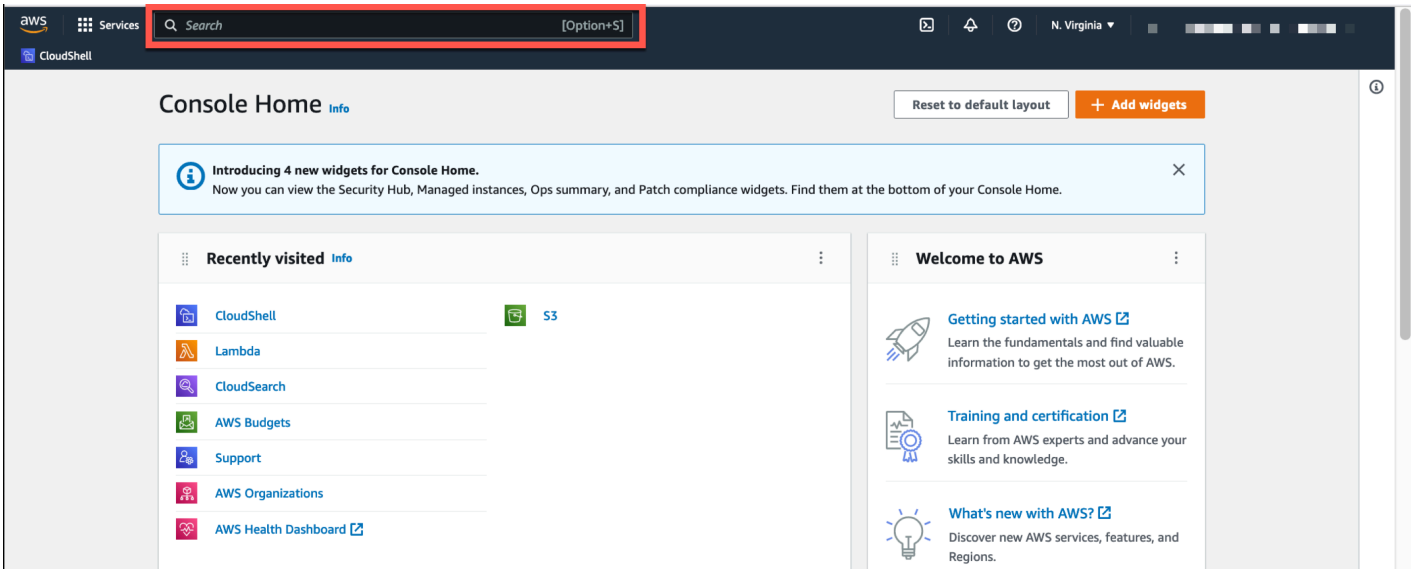
셸 작업을 시작하려면 에 AWS Management Console 로그인하고 다음 옵션 중 하나를 선택하십시오.

- 탐색 표시줄에서 CloudShell아이콘을 선택합니다.



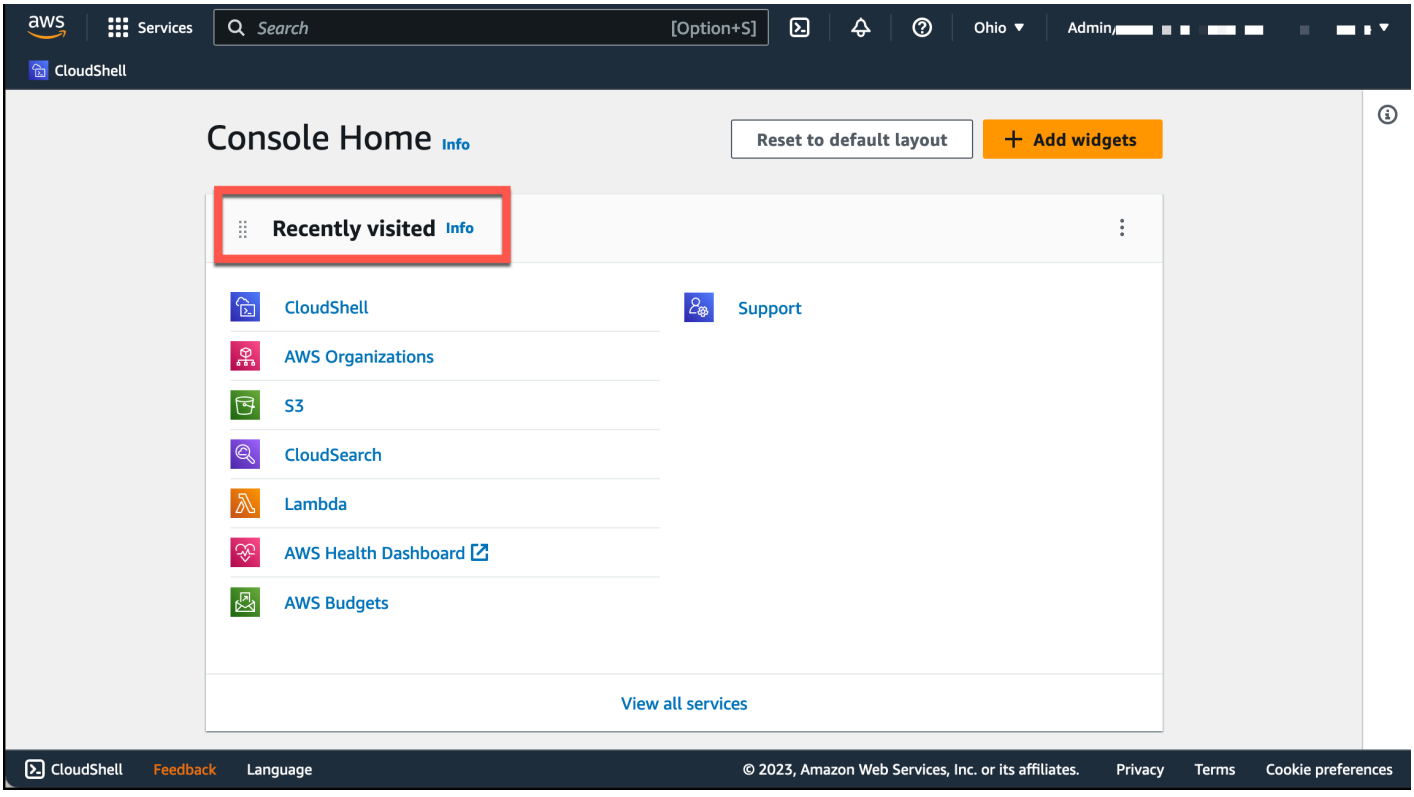
- 검색 상자에 “CloudShell" 을 입력한 다음 선택합니다 CloudShell.

  이 단계를 수행하면 CloudShell 세션이 전체 화면으로 열립니다.

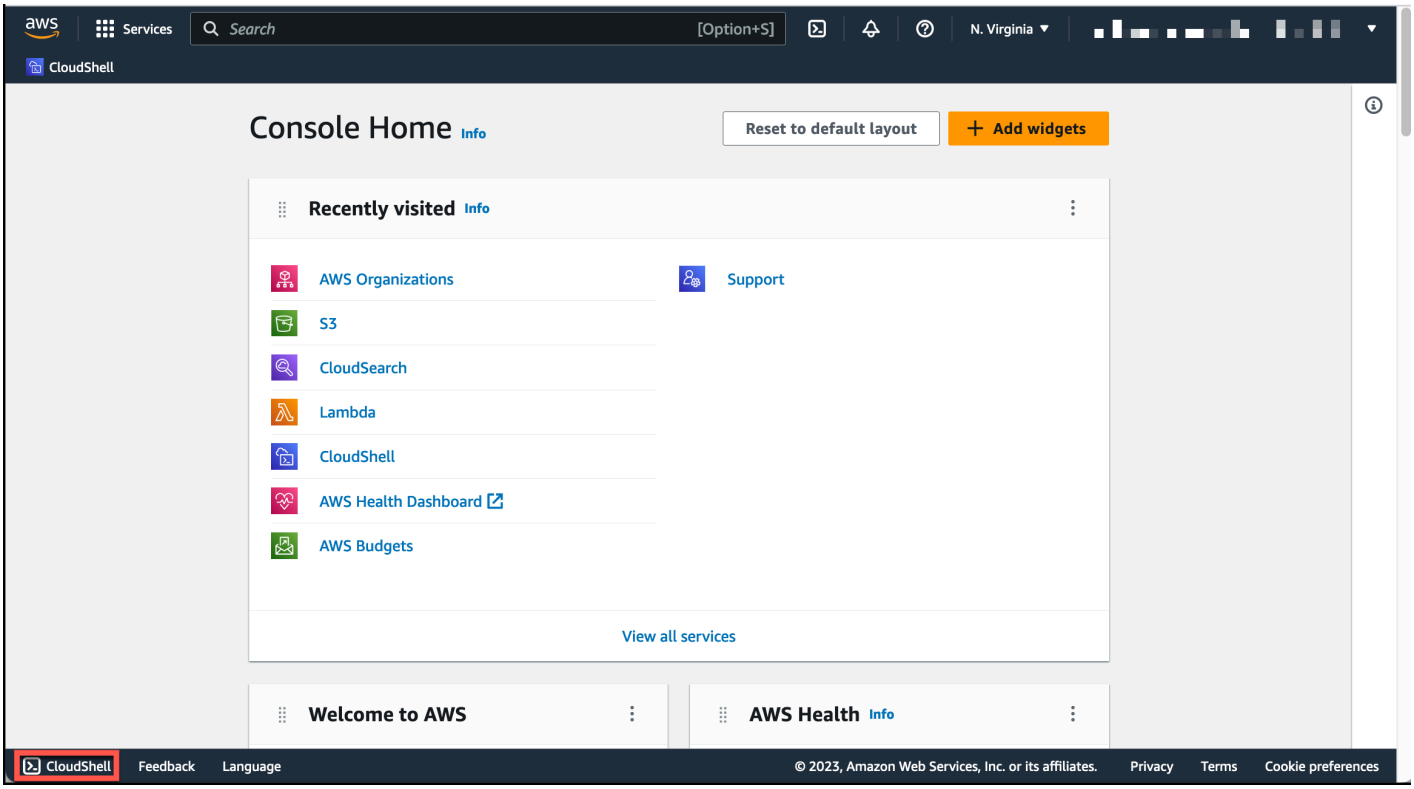


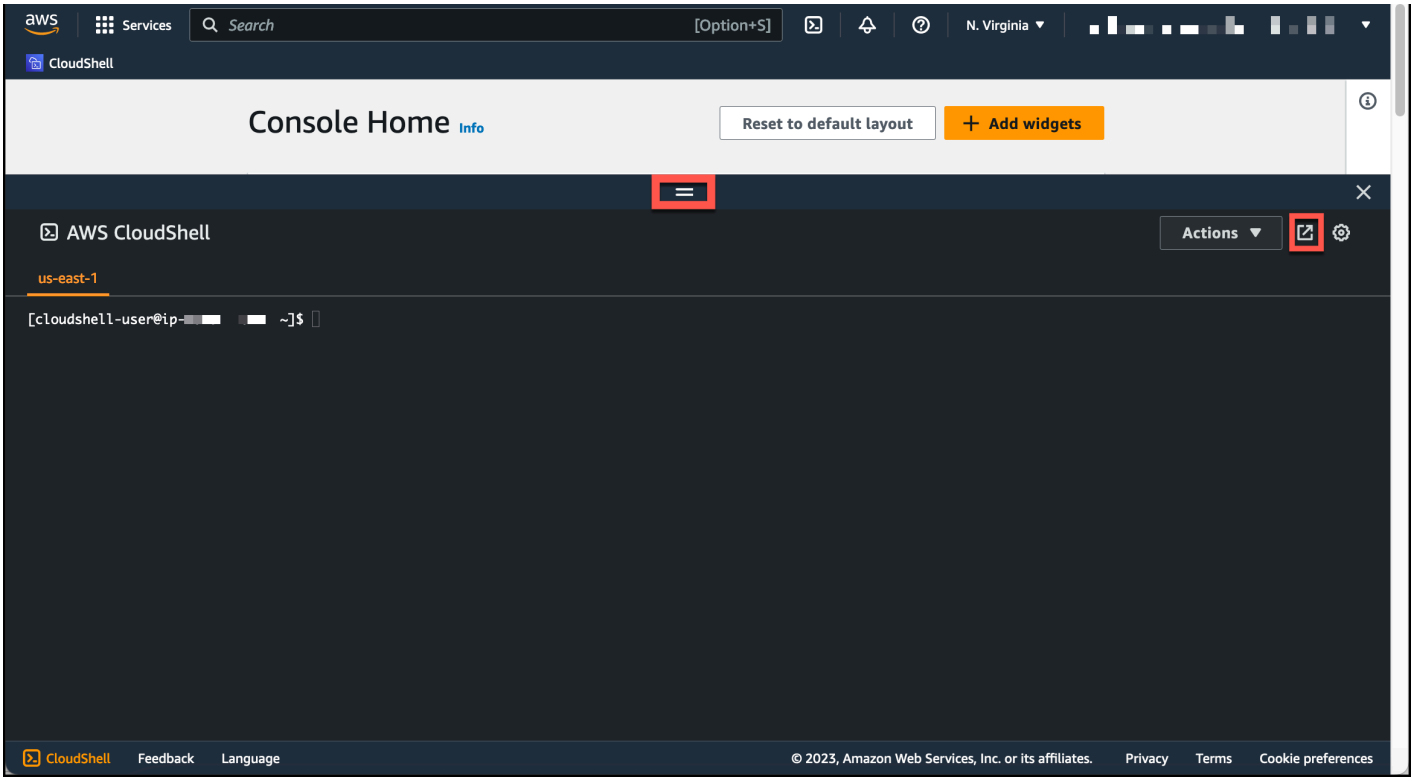
- 최근 방문한 위젯에서 선택합니다 CloudShell.

  이 단계를 수행하면 CloudShell 세션이 전체 화면으로 열립니다.



- 콘솔 왼쪽 하단에 있는 에서 선택하세요 CloudShell. Console Toolbar =드래그하여 CloudShell 세션 높이를 조정할 수 있습니다.





새 브라우저 탭에서 열기를 클릭하여 CloudShell 세션을 전체 화면으로 전환할 수도 있습니다.

에 AWS Management Console 로그인하고 주요 작업을 수행하는 방법에 대한 지침은 시작하기를 참조하십시오 AWS CloudShell. AWS CloudShell

주요 AWS CloudShell 주제

- AWS CloudShell 시작하기
- AWS CloudShell 작업
- 다음 지역AWS 서비스 이용AWS CloudShell
- 나만의 커스터마이징AWS CloudShell경험
- AWS CloudShell 컴퓨팅 환경: 사양 및 소프트웨어

AWS CloudShell 자주 묻는 질문

다음은 에 대한 몇 가지 일반적인 질문에 대한 AWS CloudShell답변입니다.

보안 중심의 상세한 FAQ는 AWS CloudShell 보안 FAQ에서 확인하십시오.

- 사용을 시작하려면 어떻게 해야 하나요 AWS CloudShell?
- 액세스하려면 무엇이 필요한가요 AWS CloudShell?
- AWS CloudShell 안에 뭐가 있어요Console Toolbar?
- AWS CloudShell 에서 어떻게 실행하나요Console Toolbar?
- 환경을 만들고 관리하려면 어떻게 해야 하나요? AWS CloudShell
- AWS 리전 어느 버전에서 AWS CloudShell 사용할 수 있나요?
- CloudShell 에서 실행할 때 선택한 지역에서 사용할 수 AWS CloudShell 없는 경우 어떤 AWS 리전 것이 Console Toolbar 할당되나요?
- AWS CloudShell에서 어떤 종류의 쉘을 사용할 수 있나요?
- 어떤 웹 브라우저와 함께 사용할 수 있나요 AWS CloudShell?
- AWS CloudShell 에서 실행할 때 어떤 웹 브라우저를 사용할 수 있나요Console Toolbar?
- AWS CloudShell 을(를) Console Toolbar에서 시작할 때 파일을 다운로드할 수 있나요?
- 쉘 환경에는 어떤 소프트웨어가 사전 설치되어 있나요?
- 쉘 환경에서 사용할 수 없는 소프트웨어를 설치할 수 있나요?

- AWS CloudShell에서 사용자가 할 수 있는 작업을 제한할 수 있나요?
- 사용 AWS CloudShell중인 AWS 리전 위치를 변경하려는 경우 홈 디렉터리에서 데이터를 이동하려면 어떻게 해야 합니까?
- 사용자 비활성으로 인한 AWS CloudShell 시간 초과 시점을 결정하는 한도를 늘릴 수 있나요?
- 홈 화면에서 AWS CloudShell 접속할 수 있나요? AWS Console Mobile Application
- AWS CloudShell 에서 어떻게 실행할 수 있나요 AWS Console Mobile Application?
- iOS와 Android 키보드에서 보조 키를 사용할 때 보조 키를 사용할 AWS CloudShell 수 있나요? AWS Console Mobile Application
- 에서 AWS CloudShell 탭 디스플레이를 여러 탭으로 분할할 수 있나요? AWS Console Mobile Application
- 모바일 AWS CloudShell 장치의 콘솔 툴바에서 액세스할 수 있나요?

사용을 시작하려면 어떻게 해야 하나요 AWS CloudShell?

AWS CloudShell 에서 몇 단계만 거치면 시작할 수 AWS Management Console있습니다. 이렇게 하려면 https://console.aws.amazon.com/console/home 에서 사용자 AWS 계정 또는 IAM 자격 증명을 사용하여 콘솔에 로그인합니다.

자세한 내용은 AWS CloudShell시작하기를 참조하십시오.

AWS CloudShell액세스하려면 무엇이 필요한가요?

AWS CloudShell 에서 액세스하므로 유효한 계정 별칭 또는 ID AWS Management Console, 사용자 이름 및 암호를 제공할 수 있는 IAM 사용자여야 합니다.

AWS CloudShell 콘솔에서 실행하려면 연결된 정책이 제공하는 IAM 권한이 필요합니다. 자세한 설명은 IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리 섹션을 참조하세요.

에는 어떤 내용이 AWS CloudShell 포함되어 있나요? Console Toolbar

왼쪽 하단에 있는 CloudShell AWS Management Console아이콘입니다.

AWS CloudShell 에서 실행하려면 어떻게 해야 하나요Console Toolbar?

콘솔 왼쪽 아래에 있는 AWS CloudShell CloudShell아이콘을 Console Toolbar 선택하여 에서 실행할 수 있습니다.

AWS 리전 어느 버전에서 AWS CloudShell 사용할 수 있나요?

지원되는 서비스 엔드포인트 AWS 리전 및 관련 서비스 엔드포인트 목록은 의 AWS CloudShellAmazon Web Services 일반 참조페이지를 참조하십시오.

CloudShell 에서 시작할 때 선택한 지역에서 사용할 수 AWS CloudShell 없는 경우 어떤 AWS 리전 것이 할당됩니까? Console Toolbar

기본 리전은 선택한 리전과 가장 가까운 리전으로 할당됩니다. 자세한 내용은 지역 선택 AWS CloudShell, 실행 및 셸 선택을 참조하십시오.

기본 리전이 아닌 다른 리전의 리소스를 관리할 권한을 제공하는 명령을 실행할 수 있습니다. 자세한 내용은 작업하기를 참조하십시오 AWS 리전.

AWS CloudShell에서 어떤 종류의 쉘을 사용할 수 있나요?

에서는 AWS CloudShellBash shell PowerShell, 또는 를 사용하여 명령을 실행할 수 Z shell 있습니다. 쉘을 전환하려면 명령 프롬프트에 다음 형식으로 사용하려는 쉘 이름을 입력합니다.

- `bash`: 사용 Bash shell
- `pwsh`: 사용 PowerShell
- `zsh`: 사용 Z shell

어떤 웹 브라우저와 함께 사용할 수 있나요 AWS CloudShell?

AWS CloudShell 구글 크롬, 모질라 파이어폭스, 마이크로소프트 엣지, 애플 사파리 브라우저의 최신 버전을 지원합니다.

환경을 만들고 관리하려면 어떻게 해야 하나요? AWS CloudShell

AWS CloudShell 환경은 지역별 IAM 사용자 ID를 기준으로 생성되고 관리됩니다. 를 `UserId` `aws sts get-caller-identity` 실행하여 확인할 수 있습니다. 환경은 해당 지역의 IAM 사용자 ID가 소유합니다. IAM `UserId` 또는 지역을 변경하면 다른 AWS CloudShell 환경에 액세스할 수 있습니다.

AWS CloudShell 에서 실행할 때 어떤 웹 브라우저를 사용할 수 있습니까? Console Toolbar

최신 버전의 구글 크롬, 마이크로소프트 엣지, 모질라 파이어폭스, 애플 사파리 브라우저를 Console Toolbar 사용하여 실행할 CloudShell 수 있습니다.

에서 실행할 AWS CloudShell 때 파일을 다운로드할 수 있나요? Console Toolbar

예, CloudShell 에서 실행하면 파일을 다운로드할 수 Console Toolbar 있습니다. 최신 버전의 Google 크롬 및 Microsoft Edge 브라우저를 사용하여 파일을 다운로드할 수 있습니다.

현재는 Mozilla Firefox 및 Apple Safari 브라우저를 사용하여 파일을 다운로드할 수 없습니다.

쉘 환경에는 어떤 소프트웨어가 사전 설치되어 있나요?

AWS CloudShell 세션용으로 만든 셸을 사용하면 원하는 명령줄 셸 (, 및) 간에 원활하게 전환할 수 있습니다. Bash PowerShell Z shell Make, pip, sudo, tar, tmux, Vim, Wget, Zip 등의 사전 설치 도구와 유틸리티에도 액세스할 수 있습니다.

쉘 환경은 대부분의 주요 소프트웨어 언어를 지원하도록 사전 구성되어 있습니다. 예를 들어 런타임 설치를 먼저 Python 수행하지 않고도 실행 Node.js 및 프로젝트에 사용할 수 있습니다. PowerShell 사용자는 .NET Core 런타임을 사용할 수 있습니다.

쉘을 사용하여 생성하거나 쉘 인터페이스로 업로드한 파일을 git의 사전 설치된 버전을 사용하여 관리되는 버전 제어 저장소에 추가할 수 있습니다.

자세한 설명은 사전 설치 소프트웨어 섹션을 참조하세요.

쉘 환경에서 사용할 수 없는 소프트웨어를 설치할 수 있나요?

예. AWS CloudShell 사용자는 sudo 권한이 있으며 명령줄에서 소프트웨어를 설치할 수 있습니다. 자세한 설명은 쉘 환경에 타사 소프트웨어 설치 섹션을 참조하세요.

AWS CloudShell에서 사용자가 할 수 있는 작업을 제한할 수 있나요?

예. AWS CloudShell에서 사용자가 수행할 수 있는 작업을 제어할 수 있습니다. 예를 들어, 사용자가 액세스하도록 AWS CloudShell 허용하되 셸 환경 내에서 파일을 업로드하거나 다운로드하는 것은 차단할 수 있습니다. 또는 사용자의 AWS CloudShell액세스를 완전히 차단할 수도 있습니다. 자세한 설명은 IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리 섹션을 참조하세요.

사용 AWS CloudShell중인 AWS 리전 위치를 변경하려는 경우 홈 디렉터리에서 데이터를 이동하려면 어떻게 해야 합니까?

한 지역에서 다른 지역으로 AWS CloudShell 데이터를 AWS 리전 이동하려면 먼저 한 지역의 홈 디렉터리 콘텐츠를 로컬 컴퓨터에 다운로드한 다음 다른 지역의 홈 디렉터리로 업로드하십시오. 자세한 설명은 자습서: 로컬 컴퓨터와 컴퓨터 간에 여러 파일 복사AWS CloudShell 섹션을 참조하세요.

사용자 비활성으로 인한 AWS CloudShell 시간 초과 시점을 결정하는 한도를 늘릴 수 있나요?

키보드 또는 포인터로 AWS CloudShell 조작하지 않으면 셸 세션은 약 20~30분 후에 자동으로 종료됩니다. 실행 중인 프로세스는 상호 작용으로 계산되지 않습니다. CloudShell 집중적인 작업 기반 활동을 위해 설계되었으므로 현재로서는 이 제한 시간을 늘릴 계획이 없습니다.

보다 유연한 제한 시간을 사용하여 터미널 기반 작업을 수행하려면 클라우드 기반 IDE를 사용하거나 Amazon EC2 인스턴스를 AWS 서비스 시작하고 연결하는 것이 좋습니다. AWS Cloud9

홈 화면에서 액세스할 AWS CloudShell 수 있습니까? AWS Console Mobile Application

예. Console Mobile AWS Console Mobile Application Application에 로그인하여 에 액세스할 AWS CloudShell 수 있습니다. 자세한 내용은 AWS Console Mobile Application 사용 설명서를 참조하십시오.

AWS CloudShell 에서 어떻게 실행할 수 있나요 AWS Console Mobile Application?

다음 방법 중 하나를 AWS CloudShell 사용하여 시작할 수 있습니다.

1. 탐색 모음 하단에 있는 AWS CloudShell 아이콘을 선택합니다.
2. 서비스 메뉴에서 AWS CloudShell을(를) 선택합니다.

에서 사용할 때 iOS와 Android 키보드에서 보조키를 사용할 AWS CloudShell 수 있나요? AWS Console Mobile Application

예. iOS 및 Android 키보드에서 보조키를 사용할 수 있습니다. 자세한 정보는 AWS Console 모바일 애플리케이션 사용 설명서에서 확인하십시오.

에서 AWS CloudShell 탭 디스플레이를 여러 탭으로 분할할 수 있나요? AWS Console Mobile Application

아니요. 현재 모바일 애플리케이션에서는 여러 AWS CloudShell 탭을 실행할 수 없습니다.

모바일 AWS CloudShell 디바이스에서도 접속할 수 있나요? Console Toolbar

아니요. 현재 모바일 AWS CloudShell Console Toolbar 장치에서는 액세스할 수 없습니다.

AWS CloudShell 시작하기

이 입문 자습서에서는 셸 명령줄 인터페이스를 사용하여 주요 작업을 AWS CloudShell 시작하고 수행하는 방법을 보여줍니다.

먼저 에 AWS Management Console 로그인하고 를 AWS 리전 선택합니다. 그런 다음 새 브라우저 CloudShell 창에서 실행하고 사용할 셸 유형을 선택합니다.

다음으로 홈 디렉터리에 새 폴더를 만들고 로컬 시스템에서 이 폴더에 파일을 업로드합니다. 명령줄에서 프로그램으로 실행하기 전에 사전 설치된 편집기를 사용하여 해당 파일을 작업합니다. 마지막으로 AWS CLI 명령을 호출하여 Amazon S3 버킷을 생성하고 파일을 객체로 버킷에 추가합니다.

필수 조건

IAM 권한

다음과 같은 AWS 관리형 정책을 IAM ID (예: 사용자, 역할 또는 그룹) 에 AWS CloudShell 연결하여 권한을 얻을 수 있습니다.

- AWSCloudShellFullAccess: 사용자에게 해당 기능에 대한 전체 액세스 권한을 AWS CloudShell 제공합니다.

이 자습서에서는 다음과 같은 작업을 수행할 수도 있습니다AWS 서비스. 좀 더 구체적으로 말하자면, S3 버킷을 만들고 해당 버킷에 객체를 추가하여 Amazon S3와 상호 작용합니다. IAM 자격 증명에는 최소한 `s3:CreateBucket` 및 `s3:PutObject` 권한을 부여하는 정책이 필요합니다.

자세한 내용은 Amazon 심플 스토리지 서비스 사용 설명서의 Amazon S3 작업을 참조하십시오.

연습 파일

또한 이 연습에는 명령줄 인터페이스에서 프로그램으로 실행되는 파일을 업로드하고 편집하는 작업도 포함됩니다. 로컬 컴퓨터에서 텍스트 편집기를 열고 다음 코드 스니펫을 추가합니다.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
sum=x+y
print("The sum is",sum)
```

add_prog.py 이름으로 파일을 저장합니다.

목차

1단계: 로그인 AWS Management Console

이 단계에는 액세스를 위한 IAM 사용자 정보를 입력하는 작업이 포함됩니다. AWS Management Console 이미 콘솔을 사용하고 있다면 2단계로 건너뛰세요.

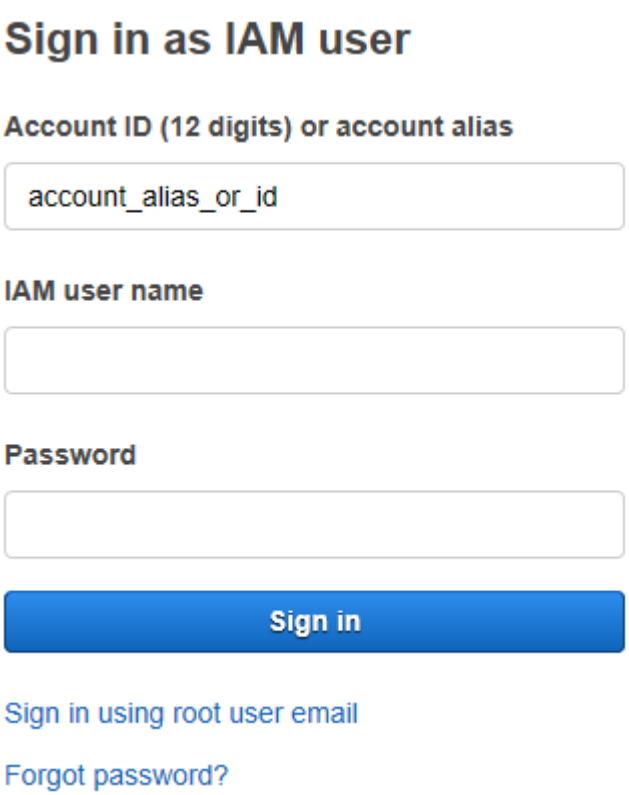
- IAM 사용자 로그인 URL을 AWS Management Console 사용하거나 기본 로그인 페이지로 이동하여 액세스할 수 있습니다.

  IAM user sign-in URL

  - 브라우저를 열고 다음 로그인 URL을 입력합니다. 관리자가 제공한 계정 별칭 또는 계정 account_alias_or_id ID로 바꾸십시오.

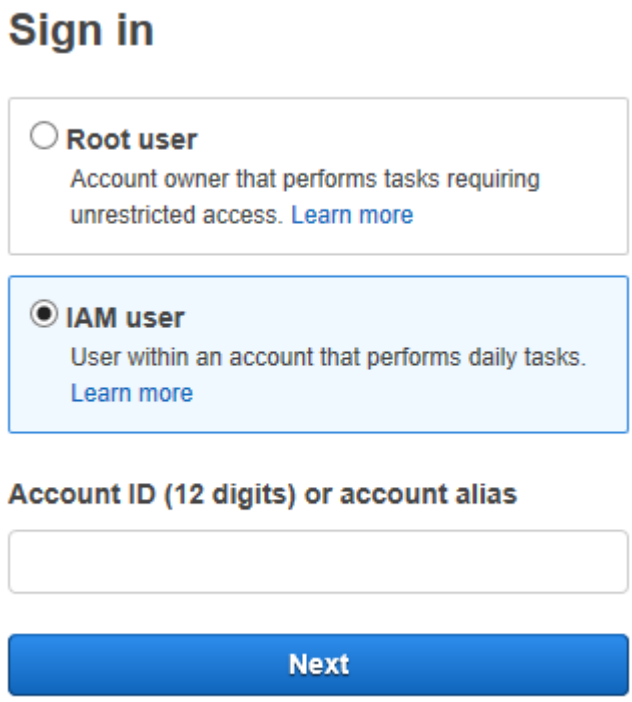
    ```
    https://account_alias_or_id.signin.aws.amazon.com/console/
    ```

  - IAM 로그인 자격 증명을 입력하고 로그인을 선택합니다.

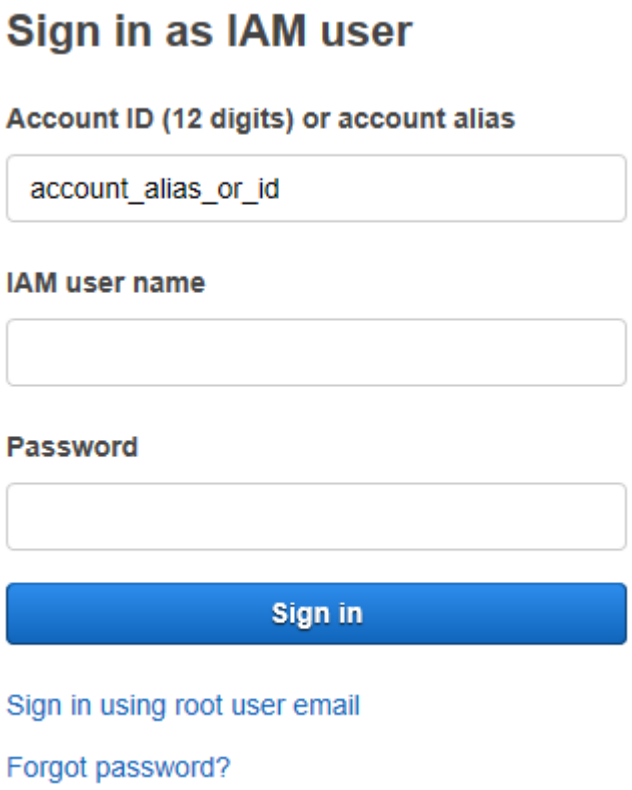


Main sign-in page

- https://aws.amazon.com/console/ 을 엽니다.
- 이전에 이 브라우저를 사용하여 로그인하지 않은 경우 기본 로그인 페이지가 나타납니다. IAM 사용자를 선택하고 계정 별칭 또는 계정 ID를 입력한 후 다음을 선택합니다.



- 이전에 이미 IAM 사용자로 로그인한 경우. 브라우저가 의 계정 별칭 또는 계정 ID를 기억할 수 있습니다. AWS 계정 그렇다면 IAM 로그인 자격 증명을 입력하고 로그인을 선택합니다.



> **Note**
>
> 루트 사용자로 로그인할 수도 있습니다. 이 ID는 계정 내 모든 AWS 서비스 리소스와 리소스에 대한 완전한 액세스 권한을 가집니다. 일상적인 작업, 심지어 관리 작업에도 루트 사용자를 사용하지 않는 것이 좋습니다. 대신, IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례를 준수합니다.

2단계: 지역 선택AWS CloudShell, 실행, 셸 선택

이 단계에서는 콘솔 AWS CloudShell 인터페이스에서 실행하고 사용 가능한 AWS 리전 셸을 선택한 다음 원하는 셸 (예: Bash PowerShell, 또는) 로 Z shell 전환합니다.

1. 작업할 영역을 AWS 리전 선택하려면 지역 선택 메뉴로 이동하여 작업할 지원 AWS 지역을 선택합니다. (사용 가능한 지역은 강조 표시됩니다.)

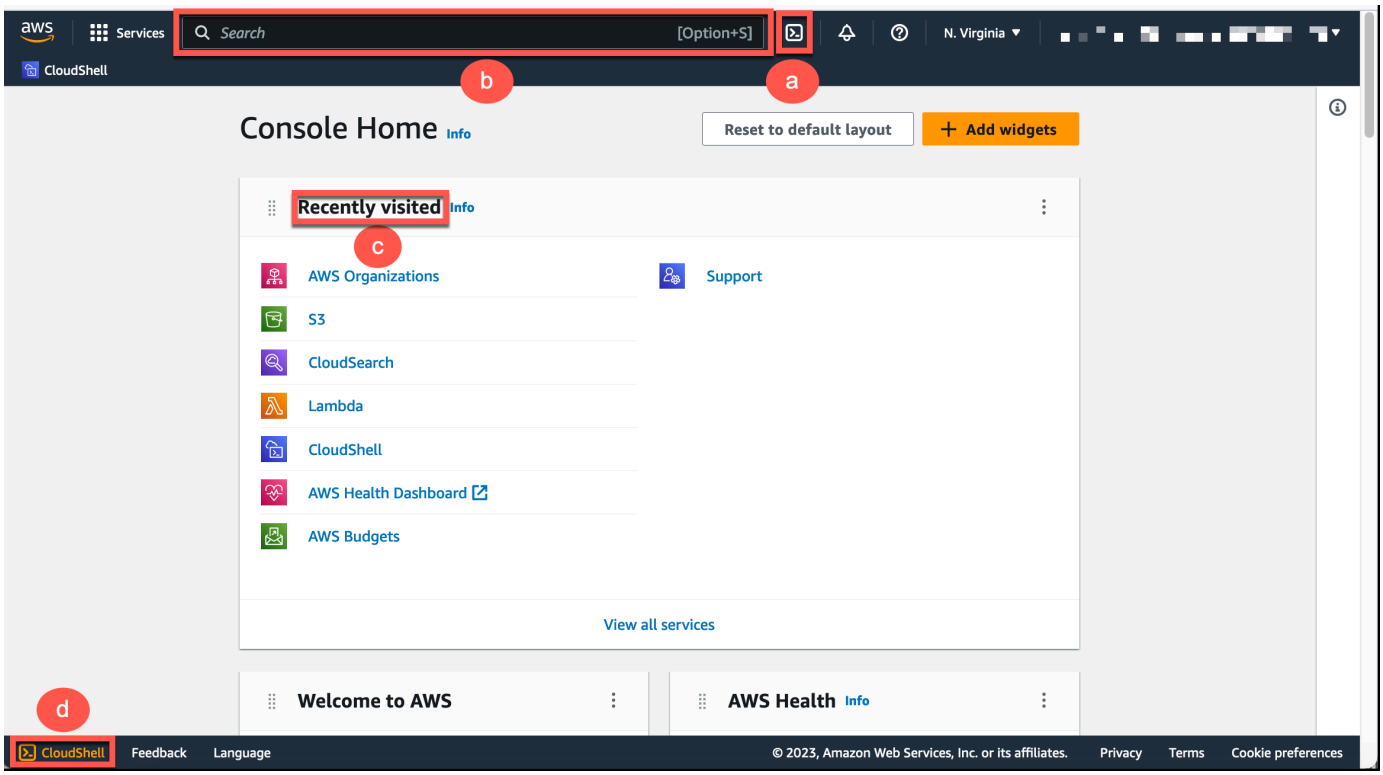
> **Important**
>
> 지역을 전환하면 인터페이스가 새로 고쳐지고 선택한 AWS 리전 영역의 이름이 명령줄 텍스트 위에 표시됩니다. 영구 스토리지에 추가하는 모든 파일은 이 스토리지에서만 사용할 수 있습니다. AWS 리전 지역을 변경하면 다른 스토리지와 파일에 액세스할 수 있습니다.

> **Important**
>
> 콘솔 왼쪽 아래에 있는 CloudShell 에서 실행할 때 선택한 지역에서 사용할 수 CloudShell 없는 경우 기본 지역은 선택한 지역과 가장 가까운 지역으로 설정됩니다. Console Toolbar 기본 지역이 아닌 다른 지역의 리소스를 관리할 권한을 제공하는 명령을 실행할 수 있습니다. 자세한 내용은 작업하기를 참조하십시오AWS 리전.
>
> Example
>
> 예
>
> 유럽 (스페인) 을 eu-south-2 선택했지만 유럽 (스페인) eu-south-2 에서는 사용할 수 CloudShell 없는 경우 기본 지역은 유럽 (스페인) eu-west-1 과 가장 가까운 유럽 (아일랜드) 으로 설정됩니다eu-south-2.

기본 지역인 유럽 (아일랜드) eu-west-1 의 서비스 할당량을 사용하면 모든 지역에서 동일한 CloudShell 세션이 복원됩니다. 기본 지역은 변경될 수 있으며 CloudShell 브라우저 창에 알림이 표시됩니다.

2. 에서 다음 AWS Management Console 옵션 중 하나를 CloudShell 선택하여 시작할 수 있습니다.

   1. 탐색 막대에서 CloudShell아이콘을 선택합니다.
   2. 검색 상자에 “CloudShell" 을 입력한 다음 선택합니다 CloudShell.
   3. 최근 방문 위젯에서 선택합니다 CloudShell.
   4. 콘솔 왼쪽 하단에 있는 에서 선택합니다 CloudShell. Console Toolbar
      - CloudShell 세션의 높이를 조정하려면 드래그하십시오=.
      - CloudShell 세션을 전체 화면으로 전환하려면 새 브라우저 탭 아이콘에서 열기를 클릭합니다.



명령 프롬프트가 표시되면 셸이 상호 작용할 준비가 된 것입니다.

> **Note**
>
> 성공적으로 시작하거나 상호 작용하는 AWS CloudShell 데 방해가 되는 문제가 발생하는 경우 에서 해당 문제를 식별하고 해결하는 데 필요한 정보를 확인하세요. AWS CloudShell 문제 해결

3. 사용할 사전 설치된 셸을 선택하려면 명령줄 프롬프트에 해당 프로그램 이름을 입력합니다.

   Bash

   ```
   bash
   ```

   로 Bash 전환하면 명령 프롬프트의 기호가 로 $ 업데이트됩니다.

   > **Note**
   >
   > Bash시작 시 실행되는 기본 AWS CloudShell 셸입니다.

   PowerShell

   ```
   pwsh
   ```

   로 PowerShell 전환하면 명령 프롬프트의 기호가 로 PS> 업데이트됩니다.

   Z shell

   ```
   zsh
   ```

   로 Z shell 전환하면 명령 프롬프트의 기호가 로 % 업데이트됩니다.

   셸 환경에 사전 설치된 버전에 대한 자세한 내용은 AWS CloudShell 컴퓨팅 환경 섹션의 셸 표를 참조하십시오.

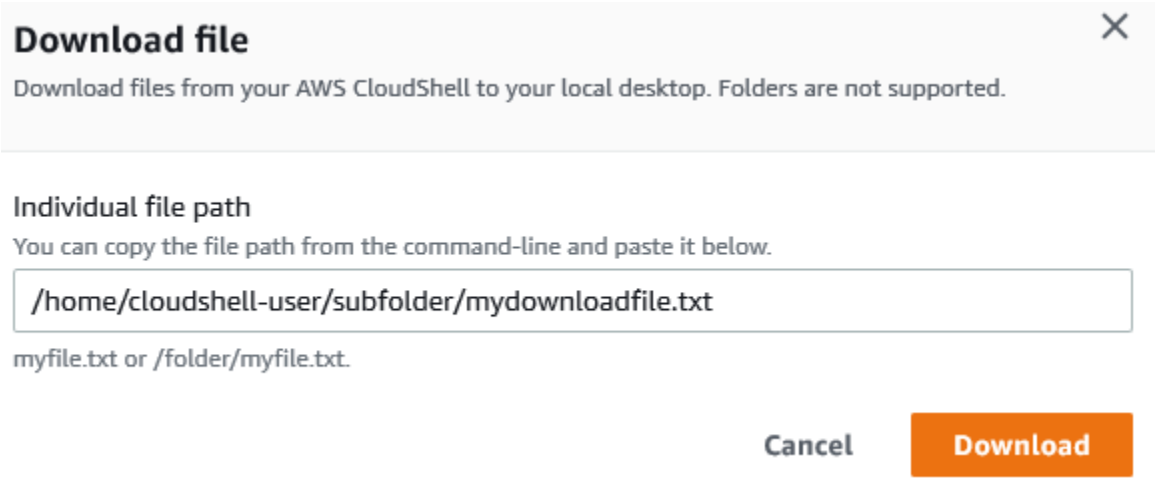
3단계: 에서 파일 다운로드 AWS CloudShell

이 단계는 파일 다운로드 프로세스를 안내합니다.

1. 파일을 다운로드하려면 작업으로 이동하여 메뉴에서 파일 다운로드를 선택합니다.

파일 다운로드 대화 상자가 표시됩니다.

2. 파일 다운로드 대화 상자에 다운로드할 파일의 경로를 입력합니다.



> **Note**
>
> 다운로드할 파일을 지정할 때 절대 또는 상대 경로를 사용할 수 있습니다. 상대 경로 이름을 사용하면 기본적으로 시작 부분에 자동으로 추가됩니다. `/home/cloudshell-user/mydownload-file`따라서 이라는 파일을 다운로드하려면 다음 두 경로가 모두 유효한 경로입니다.
>
> - 절대 경로: `/home/cloudshell-user/subfolder/mydownloadfile.txt`
> - 상대 경로: `subfolder/mydownloadfile.txt`

3. 다운로드를 선택합니다.

파일 경로가 올바르면 대화 상자가 표시됩니다. 이 대화 상자를 사용하여 기본 응용 프로그램으로 파일을 열 수 있습니다. 또는 로컬 컴퓨터의 폴더에 파일을 저장할 수 있습니다.

> **Note**
>
> CloudShell 에서 실행할 때는 다운로드 옵션을 사용할 수 없습니다Console Toolbar. CloudShell 콘솔에서 또는 Chrome 웹 브라우저를 사용하여 파일을 다운로드할 수 있습니다. 파일 다운로드 방법에 대한 자세한 내용은 3단계: 파일 다운로드를 참조하십시오AWS CloudShell.

4단계: 파일 업로드 AWS CloudShell

이 단계에서는 파일을 업로드한 다음 홈 디렉터리의 새 디렉터리로 이동하는 방법을 설명합니다.

1. 현재 작업 디렉터리를 확인하려면 프롬프트에 다음 명령을 입력합니다.

   pwd

   Enter 키를 누르면 셸은 현재 작업 디렉토리 (예:/home/cloudshell-user) 를 반환합니다.
2. 이 디렉터리에 파일을 업로드하려면 작업으로 이동하여 메뉴에서 파일 업로드를 선택합니다.

   파일 업로드 대화 상자가 표시됩니다.
3. 찾아보기를 선택합니다.
4. 시스템의 파일 업로드 대화 상자에서 이 자습서를 위해 만든 텍스트 파일 (add_prog.py) 을 선택하고 [열기] 를 선택합니다.
5. 파일 업로드 대화 상자에서 업로드를 선택합니다.

   진행률 표시줄은 업로드를 추적합니다. 업로드가 성공하면 홈 디렉터리의 루트에 add_prog.py 추가되었다는 확인 메시지가 나타납니다.
6. 파일용 디렉터리를 만들려면 디렉터리 만들기 명령어:를 입력합니다. mkdir mysub_dir
7. 업로드된 파일을 홈 디렉터리의 루트에서 새 디렉터리로 이동하려면 다음 mv 명령을 사용합니다.

   mv add_prog.py mysub_dir.
8. 작업 디렉터리를 새 디렉터리로 변경하려면 를 입력합니다cd mysub_dir.

   명령 프롬프트가 업데이트되어 작업 디렉토리가 변경되었음을 알립니다.
9. 현재 디렉터리의 내용을 보려면 ls 명령을 입력합니다. mysub_dir

   작업 디렉토리의 내용이 나열됩니다. 여기에는 방금 업로드한 파일이 포함됩니다.

5단계: 에서 파일 제거 AWS CloudShell

이 단계에서는 에서 파일을 제거하는 방법을 설명합니다AWS CloudShell.

1. 에서 AWS CloudShell 파일을 제거하려면 rm (제거) 와 같은 표준 셸 명령을 사용합니다.

   rm my-file-for-removal

2. 지정된 기준에 맞는 여러 파일을 제거하려면 find 명령을 실행합니다.

   다음 예제에서는 이름에 접미사 “.pdf”가 포함된 모든 파일을 제거합니다.

```
find -type f -name '*.pdf' -delete
```

> **Note**
>
> 특정 버전에서 사용을 AWS CloudShell 중단한다고 가정해 보겠습니다. AWS 리전 그러면 해당 지역의 영구 저장소에 있는 데이터가 지정된 기간이 지나면 자동으로 제거됩니다. 자세한 내용은 영구 저장소를 참조하십시오.

6단계: 홈 디렉터리 백업 생성

1. 백업 파일 생성

   홈 디렉터리 외부에 임시 폴더를 생성합니다.

```
HOME_BACKUP_DIR=$(mktemp --directory)
```

   다음 옵션 중 하나를 사용하여 백업을 만들 수 있습니다.

   a. tar를 사용하여 백업 파일을 생성합니다.

      tar를 사용하여 백업 파일을 만들려면 다음 명령을 입력합니다.

```
tar \
    --create \
    --gzip \
    --verbose \
    --file=${HOME_BACKUP_DIR}/home.tar.gz \
    [--exclude ${HOME}/.cache] \ // Optional
    ${HOME}/
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.tar.gz"
```

   b. zip을 사용하여 백업 파일을 생성합니다.

      zip을 사용하여 백업 파일을 만들려면 다음 명령을 입력합니다.

```
zip \
    --recurse-paths \
    ${HOME_BACKUP_DIR}/home.zip \
    ${HOME} \
    [--exclude ${HOME}/.cache/\*] // Optional
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.zip"
```

2. 백업 파일을 외부로 전송 CloudShell

   다음 옵션 중 하나를 사용하여 백업 파일을 외부로 전송할 수 있습니다 CloudShell.

   a. 로컬 컴퓨터에 백업 파일을 다운로드합니다.

      이전 단계에서 만든 파일을 다운로드할 수 있습니다. 에서 CloudShell 파일을 다운로드하는 방법에 대한 자세한 내용은 파일 다운로드 위치를 참조하십시오AWS CloudShell.

      파일 다운로드 대화 상자에 다운로드할 파일의 경로 (예:`/tmp/tmp.iA99tD9L98/home.tar.gz`) 를 입력합니다.

   b. 백업 파일을 S3로 전송합니다.

      버킷을 생성하려면 다음 명령을 입력합니다.

      ```
      aws s3 mb s3://${BUCKET_NAME}
      ```

      AWS CLI를 사용하여 파일을 S3 버킷에 복사합니다.

      ```
      aws s3 cp ${HOME_BACKUP_DIR}/home.tar.gz s3://${BUCKET_NAME}
      ```

      > **Note**
      > 데이터 전송 요금이 부과될 수 있습니다.

3. S3 버킷에 직접 백업

   S3 버킷에 직접 백업하려면 다음 명령을 입력합니다.

   ```
   aws s3 cp \
       ${HOME}/ \
       s3://${BUCKET_NAME} \
   ```

```
    --recursive \
    [--exclude .cache/\*] // Optional
```

7단계: 셸 세션 재시작

> **Note**
>
> 보안 조치로 오랫동안 키보드나 포인터를 사용하여 셸과 상호 작용하지 않으면 세션이 자동으로 중지됩니다. 장기 실행 세션도 자동으로 중지됩니다. 자세한 정보는 셸 세션을 참조하세요.

1. 셸 세션을 다시 시작하려면 [작업], [다시 시작 AWS CloudShell] 을 선택합니다.

   다시 시작하면 현재 AWS 리전 활성 세션이 모두 AWS CloudShell 중지된다는 알림이 표시됩니다.
2. 확인하려면 [Restart] 를 선택합니다.

   인터페이스에 CloudShell 컴퓨팅 환경이 중지되고 있다는 메시지가 표시됩니다. 환경이 중지되었다가 다시 시작된 후 새 세션에서 명령줄 작업을 시작할 수 있습니다.

   > **Note**
   >
   > 환경을 다시 시작하는 데 몇 분 정도 걸릴 수 있는 경우도 있습니다.

8단계: 셸 세션 홈 디렉터리 삭제

> **Warning**
>
> 홈 디렉터리 삭제는 홈 디렉터리에 저장된 모든 데이터가 영구적으로 삭제되는 되돌릴 수 없는 작업입니다. 하지만 다음과 같은 상황에서는 이 옵션을 고려해 볼 수 있습니다.
>
> - 파일을 잘못 수정하여 AWS CloudShell 컴퓨팅 환경에 액세스할 수 없습니다. 홈 디렉터리를 삭제하면 기본 설정으로 AWS CloudShell 돌아갑니다.
> - 모든 데이터를 AWS CloudShell 즉시 제거하고 싶을 것입니다. 특정 AWS CloudShell 지역에서 사용을 중지하면 AWS 해당 지역에서 AWS CloudShell 다시 시작하지 않는 한 보존 기간 종료 시 영구 스토리지가 자동으로 삭제됩니다.

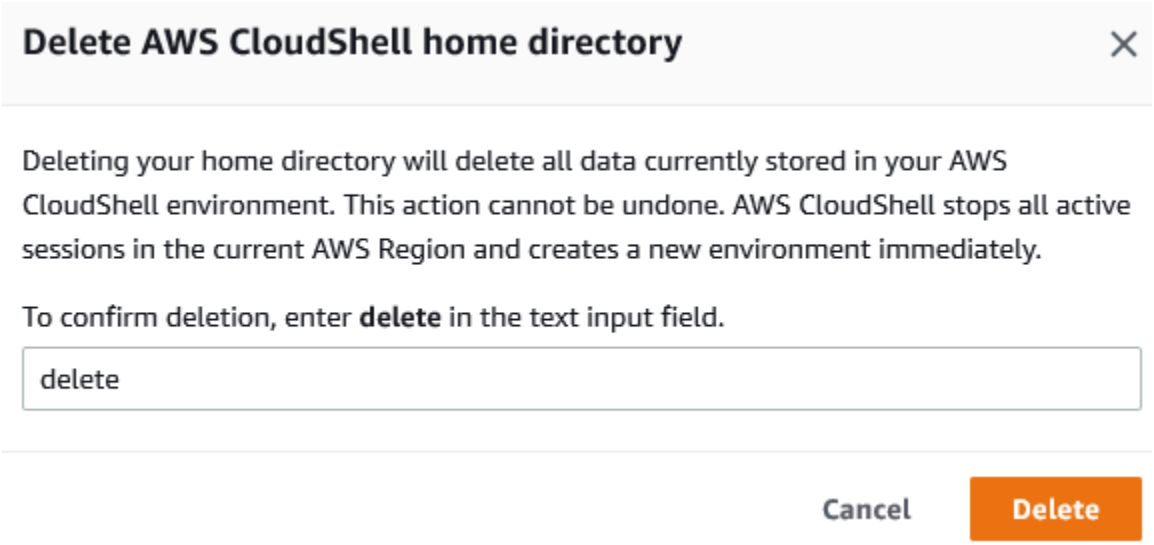
파일을 장기간 보관해야 하는 경우 Amazon S3 또는 같은 서비스를 고려해 보십시오 CodeCommit.

1. 셸 세션을 삭제하려면 작업, AWS CloudShell 홈 디렉터리 삭제를 선택합니다.

   AWS CloudShell홈 디렉터리를 삭제하면 현재 AWS CloudShell 환경에 저장된 모든 데이터가 삭제된다는 알림이 표시됩니다.

   > **Note**
   >
   > 이 작업을 취소할 수 없습니다.

2. 삭제를 확인하려면 텍스트 입력 필드에 delete를 입력한 다음 삭제를 선택합니다.



   AWS는 현재의 모든 활성 세션을 CloudShell AWS 리전 중지하고 즉시 새 환경을 만듭니다.

   셸 세션을 수동으로 종료합니다.

   명령줄을 사용하면 셸 세션에서 나가서 `exit` 명령을 사용하여 로그아웃할 수 있습니다. 그런 다음 아무 키나 눌러 다시 연결하고 계속 사용할 AWS CloudShell 수 있습니다.

9단계: 파일 코드를 편집하고 명령줄을 사용하여 실행

이 단계에서는 사전 설치된 Vim 편집기를 사용하여 파일 작업을 수행하는 방법을 보여줍니다. 그런 다음 명령줄에서 해당 파일을 프로그램으로 실행합니다.

1. 이전 단계에서 업로드한 파일을 편집하려면 다음 명령을 입력합니다.

   ```
   vim add_prog.py
   ```

   셸 인터페이스가 새로 고쳐져 Vim 편집기가 표시됩니다.

2. 에서 파일을 Vim 편집하려면 I 키를 누릅니다. 이제 프로그램에서 두 개가 아닌 세 개의 숫자를 더하도록 내용을 편집하십시오.

   ```
   import sys
   x=int(sys.argv[1])
   y=int(sys.argv[2])
   z=int(sys.argv[3])
   sum=x+y+z
   print("The sum is",sum)
   ```

   > **Note**
   >
   > 텍스트를 편집기에 붙여넣고 안전 붙여넣기 기능을 활성화한 경우 경고가 표시됩니다. 복사된 여러 줄 텍스트에는 악성 스크립트가 포함될 수 있습니다. 안전 붙여넣기 기능을 사용하면 텍스트를 붙여넣기 전에 전체 텍스트를 확인할 수 있습니다. 텍스트가 안전하다고 생각되면 [붙여넣기] 를 선택합니다.

3. 프로그램을 편집한 후 키를 Esc 눌러 Vim 명령 모드로 들어갑니다. 그런 다음 `:wq` 명령을 입력하여 파일을 저장하고 편집기를 종료합니다.

   > **Note**
   >
   > Vim명령 모드를 처음 사용하는 경우 처음에는 명령 모드와 삽입 모드 사이를 전환하는 것이 어려울 수 있습니다. 명령 모드는 파일을 저장하고 응용 프로그램을 종료할 때 사용됩니다. 새 텍스트를 삽입할 때는 삽입 모드가 사용됩니다. 삽입 모드로 들어가려면 및 키를 누르고I, 명령 모드로 들어가려면 를 누릅니다Esc. 에서 AWS CloudShell 사용할 수 있는

기타 도구 Vim 및 기타 도구에 대한 자세한 내용은 을 참조하십시오개발 도구 및 쉘 유틸리티.

4. 기본 명령줄 인터페이스에서 다음 프로그램을 실행하고 입력에 사용할 세 개의 숫자를 지정합니다. 구문은 다음과 같습니다.

   ```
   python3 add_prog.py 4 5 6
   ```

   명령줄에는 프로그램 출력이 표시됩니다`The sum is 15.`

10단계: 파일을 Amazon S3 버킷의 객체로 AWS CLI 추가하는 데 사용합니다.

이 단계에서는 Amazon S3 버킷을 만든 다음 PutObject메서드를 사용하여 코드 파일을 해당 버킷의 객체로 추가합니다.

> **Note**
>
> 대부분의 경우 소프트웨어 파일을 버전 관리 리포지토리에 CodeCommit 커밋하는 등의 서비스를 사용할 수 있습니다. 이 자습서에서는 in을 사용하여 AWS CLI 다른 AWS 서비스와 상호 작용하는 AWS CloudShell 방법을 보여줍니다. 이 방법을 사용하면 추가 리소스를 다운로드하거나 설치할 필요가 없습니다. 또한 셸 내에서 이미 인증되었기 때문에 직접 호출을 하기 전에 보안 인증을 구성하지 않아도 됩니다.

1. 지정된 AWS 리전 버킷에 버킷을 만들려면 다음 명령을 입력합니다.

   ```
   aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
   ```

   > **Note**
   >
   > `us-east-1`지역 외부에서 버킷을 생성하는 경우 `LocationConstraint` 파라미터를 추가하여 `create-bucket-configuration` 지역을 지정하십시오. 다음은 구문의 예제입니다.

```
$ aws s3api create-bucket --bucket my-bucket --region eu-west-1 --create-
bucket-configuration LocationConstraint=eu-west-1
```

호출이 성공하면 명령줄에 다음 출력과 유사한 서비스의 응답이 표시됩니다.

```
{
    "Location": "/insert-unique-bucket-name-here"
}
```

> **Note**
>
> 버킷 이름 지정 규칙을 준수하지 않으면 다음 오류가 표시됩니다. CreateBucket 작업을 호출하는 동안 오류가 발생했습니다 (InvalidBucketName): 지정된 버킷이 유효하지 않습니다.

2. 파일을 업로드하고 방금 만든 버킷에 파일을 객체로 추가하려면 메서드를 호출합니다PutObject.

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body
 add_prog.py
```

객체가 Amazon S3 버킷에 업로드되면 명령줄에 다음 출력과 유사한 서비스의 응답이 표시됩니다.

```
{"ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebeea56\""}
```

ETag는 저장된 객체의 해시입니다. 이 해시를 사용하여 Amazon S3에 업로드된 객체의 무결성을 확인할 수 있습니다.

관련 주제

- 다음 지역AWS 서비스 이용AWS CloudShell
- 자습서: 로컬 컴퓨터와 컴퓨터 간에 여러 파일 복사AWS CloudShell
- 튜토리얼: CodeCommit 에서 사용AWS CloudShell
- AWS CloudShell 작업

- 나만의 커스터마이징AWS CloudShell경험

AWS CloudShell 자습서

다음 자습서를 통해 다양한 기능과 통합을 사용하여 실험하고 테스트할 수 있습니다. AWS CloudShell

주제

- 자습서: 로컬 컴퓨터와 컴퓨터 간에 여러 파일 복사AWS CloudShell
- 튜토리얼: CodeCommit 에서 사용AWS CloudShell
- 자습서: Amazon S3 객체에 대해 미리 서명된 URL 생성AWS CloudShell
- 자습서: 내부에 AWS CloudShell Docker 컨테이너를 구축하여 Amazon ECR 리포지토리로 푸시하기
- 자습서: 를 사용하여 Lambda 함수 배포하기 AWS CDK

자습서: 로컬 컴퓨터와 컴퓨터 간에 여러 파일 복사AWS CloudShell

CloudShell 인터페이스를 사용하면 로컬 시스템과 셸 환경 간에 한 번에 단일 파일을 업로드하거나 다운로드할 수 있습니다. 로컬 시스템 간에 CloudShell 여러 파일을 동시에 복사하려면 다음 옵션 중 하나를 사용합니다.

- Amazon S3: 로컬 시스템과 간에 파일을 복사할 때 S3 버킷을 중개자로 사용하십시오 CloudShell.
- Zip 파일: CloudShell 인터페이스를 사용하여 업로드하거나 다운로드할 수 있는 단일 압축 폴더에 여러 파일을 압축합니다.

> **Note**
>
> 들어오는 인터넷 트래픽을 허용하지 CloudShell 않기 때문에 현재는 로컬 시스템과 CloudShell 컴퓨팅 환경 간에`scp` 또는 여러 파일을`rsync` 복사하는 등의 명령을 사용할 수 없습니다.

Amazon S3를 사용하여 여러 파일 업로드 및 다운로드

사전 조건

버킷과 객체를 사용하려면 다음과 같은 Amazon S3 API 작업을 수행할 권한을 부여하는 IAM 정책이 필요합니다.

- `s3:CreateBucket`
- `s3:PutObject`
- `s3:GetObject`

Amazon S3 작업의 전체 목록은 Amazon 심플 스토리지 서비스 API 참조의 작업을 참조하십시오.

AmazonAWS CloudShell S3를 사용하여 여러 파일을 업로드합니다.

1. 에서AWS CloudShell 다음`s3` 명령을 실행하여 S3 버킷을 생성합니다.

   ```
   aws s3api create-bucket --bucket your-bucket-name --region us-east-1
   ```

   호출이 성공하면 명령줄에 S3 서비스의 응답이 표시됩니다.

   ```
   {
       "Location": "/your-bucket-name"
   }
   ```

2. 로컬 머신의 디렉터리에 있는 파일을 버킷으로 업로드합니다. 다음 옵션 중 하나를 선택하여 파일을 업로드합니다.

   - AWS Management Console: 파일 및 폴더를 버킷에 업로드하는 drag-and-drop 데 사용합니다.
   - AWS CLI: 로컬 컴퓨터에 설치된 도구 버전을 사용하여 명령줄을 사용하여 파일 및 폴더를 버킷에 업로드합니다.

   Using the console

   - https://s3.console.aws.amazon.com/s3/ 에서 Amazon S3 콘솔을 엽니다.

     (를 사용하는AWS CloudShell 경우 이미 콘솔에 로그인되어 있어야 합니다.)
   - 왼쪽 탐색 창에서 버킷을 선택한 다음 폴더 또는 파일을 업로드할 버킷 이름을 선택합니다. Create bucket (Create bucket) 을 선택하여 원하는 버킷을 만들 수도 있습니다.
   - 업로드할 파일 및 폴더를 선택하려면 업로드를 선택합니다. 그런 다음 선택한 파일 및 폴더를 대상 버킷의 객체가 나열되어 있는 콘솔 창으로 끌어서 놓습니다. 또는 파일 추가 또는 폴더 추가를 선택합니다.

     선택한 파일이 업로드 페이지에 나열됩니다.

- 추가할 파일을 지정하려면 확인란을 선택합니다.
- 선택한 파일을 버킷에 추가하려면 [Upload] 를 선택합니다.

> **Note**
>
> 콘솔을 사용할 때의 전체 구성 옵션에 대한 자세한 내용은 S3 버킷에 파일 및 폴더를 업로드하려면 어떻게 해야 합니까? 를 참조하십시오. Amazon Storage Console 에서 확인할 수 있습니다.

Using AWS CLI

> **Note**
>
> 이 옵션을 사용하려면 로컬 컴퓨터에AWS CLI 도구를 설치하고AWS 서비스 호출을 위한 자격 증명을 구성해야 합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서를 참조하세요.

- AWS CLI도구를 실행하고 다음aws s3 명령을 실행하여 지정된 버킷을 로컬 시스템의 현재 디렉토리 콘텐츠와 동기화합니다.

```
aws s3 sync folder-path s3://your-bucket-name
```

동기화가 성공하면 버킷에 추가된 모든 객체에 대한 업로드 메시지가 표시됩니다.

3. CloudShell 명령줄로 돌아가서 다음 명령을 입력하여 셸 환경의 디렉터리를 S3 버킷의 콘텐츠와 동기화합니다.

```
aws s3 sync  s3://your-bucket-name folder-path
```

> **Note**
>
> sync명령에--exclude "<value>" 및--include "<value>" 매개 변수를 추가하여 패턴 일치를 수행하여 특정 파일이나 객체를 제외하거나 포함할 수도 있습니다.
> 자세한 내용은 AWS CLI명령 참조서에서 제외 및 포함 필터 사용을 참조하십시오.

동기화에 성공하면 버킷에서 디렉터리로 다운로드된 모든 파일에 대한 다운로드 메시지가 표시됩니다.

> **Note**
>
> sync 명령을 사용하면 새 파일과 업데이트된 파일만 소스 디렉터리에서 대상으로 재귀적으로 복사됩니다.

AmazonAWS CloudShell S3를 사용하여 여러 파일을 다운로드합니다.

1. AWS CloudShell명령줄을 사용하여 다음aws s3 명령을 입력하여 셸 환경의 현재 디렉토리 콘텐츠와 S3 버킷을 동기화합니다.

```
aws s3 sync folder-path s3://your-bucket-name
```

> **Note**
>
> sync명령에--exclude "<value>" 및--include "<value>" 매개 변수를 추가하여 패턴 일치를 수행하여 특정 파일이나 객체를 제외하거나 포함할 수도 있습니다. 자세한 내용은 AWS CLI명령 참조서에서 제외 및 포함 필터 사용을 참조하십시오.

   동기화가 성공하면 버킷에 추가된 모든 객체에 대한 업로드 메시지가 표시됩니다.

2. 버킷의 콘텐츠를 로컬 시스템에 다운로드 합니다. Amazon S3 콘솔은 여러 객체 다운로드를 지원하지 않으므로 로컬 컴퓨터에 설치된AWS CLI 도구를 사용해야 합니다.

   AWS CLI도구의 명령줄 프롬프트에 다음 명령을 실행합니다.

```
aws s3 sync s3://your-bucket-name folder-path
```

   동기화가 성공하면 명령줄에 대상 디렉터리에 업데이트되거나 추가된 각 파일에 대한 다운로드 메시지가 표시됩니다.

> **Note**
>
> 이 옵션을 사용하려면 로컬 컴퓨터에AWS CLI 도구를 설치하고AWS 서비스 호출을 위한 자격 증명을 구성해야 합니다. 자세한 내용은 AWS Command Line Interface 사용 설명서를 참조하세요.

압축 폴더를 사용하여 여러 파일 업로드 및 다운로드

zip/unzip 유틸리티를 사용하면 단일 파일로 취급될 수 있는 아카이브의 여러 파일을 압축할 수 있습니다. 유틸리티는 CloudShell 컴퓨팅 환경에 사전 설치되어 있습니다.

사전 설치된 도구에 대한 자세한 내용은 을 참조하십시오개발 도구 및 쉘 유틸리티.

압축 폴더를AWS CloudShell 사용하여 여러 파일을 업로드합니다.

1. 로컬 시스템에서 업로드할 파일을 압축 폴더에 추가합니다.
2. 시작한 다음 작업 CloudShell, 파일 업로드를 선택합니다.
3. 파일 업로드 대화 상자에서 파일 선택을 선택한 다음 방금 만든 압축 폴더를 선택합니다.
4. 파일 업로드 대화 상자에서 업로드를 선택하여 선택한 파일을 셸 환경에 추가합니다.
5. CloudShell 명령줄에서 다음 명령을 실행하여 zip 아카이브의 내용을 지정된 디렉터리에 압축 해제합니다.

```
unzip zipped-files.zip -d my-unzipped-folder
```

압축 폴더를AWS CloudShell 사용하여 여러 파일 다운로드

1. CloudShell 명령줄에서 다음 명령을 실행하여 현재 디렉터리의 모든 파일을 압축된 폴더에 추가합니다.

```
zip -r zipped-archive.zip *
```

2. 작업, 파일 다운로드를 선택합니다.
3. 파일 다운로드 대화 상자에서 압축된 폴더의 경로 (`/home/cloudshell-user/zip-folder/zipped-archive.zip`예:) 를 입력한 다음 다운로드를 선택합니다.

경로가 올바르면 브라우저 대화 상자에 압축된 폴더를 열거나 로컬 컴퓨터에 저장할지 선택할 수 있습니다.

4. 이제 로컬 컴퓨터에서 다운로드한 압축 폴더의 내용을 압축 해제할 수 있습니다.

튜토리얼: CodeCommit 에서 사용AWS CloudShell

CodeCommit 프라이빗 Git 리포지토리를 호스팅하는 안전하고 확장성이 뛰어난 관리형 소스 관리 서비스입니다. AWS CloudShell를 사용하면 git-remote-codecommit유틸리티를 사용하여 CodeCommit 명령줄에서 작업할 수 있습니다. 이 유틸리티는AWS CloudShell 컴퓨팅 환경에 사전 설치되어 있으며 CodeCommit 리포지토리에서 코드를 푸시하고 가져오는 간단한 방법을 제공합니다. 이 유틸리티는 Git을 확장하여 이 작업을 수행합니다. 자세한 내용은 AWS CodeCommit 사용 설명서를 참조하세요.

이 자습서에서는 CodeCommit 리포지토리를 생성하고AWS CloudShell 컴퓨팅 환경에 복제하는 방법을 설명합니다. 또한AWS 클라우드에서 관리되는 원격 리포지토리로 파일을 푸시하기 전에 파일을 스테이징하고 복제된 리포지토리에 커밋하는 방법도 알아봅니다.

사전 조건

IAM 사용자가 사용해야AWS CloudShell 하는 권한에 대한 자세한 내용은 시작하기 자습서의 사전 요구 사항 섹션을 참조하십시오. 작업하려면 IAM 권한도 필요합니다 CodeCommit.

또한 시작하기 전에 다음 사항을 확인하십시오.

- Git 명령어 및 버전 제어 개념에 대한 기본적인 이해
- 로컬 및 원격 리포지토리에 커밋할 수 있는 셸의 홈 디렉터리에 있는 파일입니다. 이 자습서에서는 다음과 같이 지칭합니다`my-git-file`.

1단계: CodeCommit 리포지토리를 만들고 복제하기

1. CloudShell 명령줄 인터페이스에서 다음`codecommit` 명령을 입력하여 라는 CodeCommit 리포지토리를 생성합니다`MyDemoRepo`.

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-
description "My demonstration repository"
```

리포지토리가 성공적으로 생성되면 명령줄에 서비스의 응답이 표시됩니다.

```
{
    "repositoryMetadata": {
        "accountId": "111122223333",
        "repositoryId": "0dcd29a8-941a-1111-1111-11111111111a",
        "repositoryName": "MyDemoRepo",
        "repositoryDescription": "My demonstration repository",
        "lastModifiedDate": "2020-11-23T20:38:23.068000+00:00",
        "creationDate": "2020-11-23T20:38:23.068000+00:00",
        "cloneUrlHttp": "https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/
MyDemoRepo",
        "cloneUrlSsh": "ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/
MyDemoRepo",
        "Arn": "arn:aws:codecommit:eu-west-1:111111111111:MyDemoRepo"
     }
 )
```

2. 명령줄을 사용하여 로컬 리포지토리를 위한 새 디렉터리를 만들고 이 디렉터리를 작업 디렉터리로 지정합니다.

```
mkdir my-shell-repo
cd my-shell-repo
```

3. 원격 리포지토리를 복제하려면`git clone` 명령을 사용합니다. (git-remote-codecommit작업할 때는 HTTPS (GRC) URL 스타일을 사용하십시오.

```
git clone codecommit::eu-west-1://MyDemoRepo
```

리포지토리가 성공적으로 복제되면 명령줄에 서비스 응답이 표시됩니다.

```
Cloning into 'MyDemoRepo'...
warning: You appear to have cloned an empty repository.
```

4. 복제된 리포지토리로 이동하려면`cd` 명령을 사용합니다.

```
cd MyDemoRepo
```

2단계: 파일을 CodeCommit 저장소로 푸시하기 전에 파일을 스테이징하고 커밋합니다.

1. Vim 에디터 또는 의 파일 업로드 기능을 사용하여MyDemoRepo 폴더에my-git-file 호출된 파일을 추가합니다AWS CloudShell. 두 가지 방법을 모두 사용하는 방법을 알아보려면 시작하기 튜토리얼을 참조하세요.
2. 리포지토리에 파일을 스테이징하려면 gitadd 명령을 실행합니다.

```
git add my-git-file
```

3. 파일이 스테이징되어 커밋할 준비가 되었는지 확인하려면 gitstatus 명령을 실행합니다.

```
git status
```

   my-git-file새 파일로 나열되고 녹색 텍스트로 표시되어 커밋할 준비가 되었음을 나타냅니다.

4. 스테이징된 파일의 이 버전을 리포지토리에 커밋합니다.

```
git commit -m "first commit to repo"
```

> **Note**
>
> 커밋을 완료하기 위한 구성 정보를 묻는 메시지가 표시되면 다음 형식을 사용하십시오.
>
> ```
> $ git config --global user.name "Jane Doe"
> $ git config --global user.email janedoe@example.com
> ```

5. 원격 리포지토리를 로컬 리포지토리의 변경 사항과 동기화하려면 변경 사항을 업스트림 브랜치로 푸시하세요.

```
git push
```

자습서: Amazon S3 객체에 대해 미리 서명된 URL 생성AWS CloudShell

이 자습서에서는 Amazon S3 객체를 다른 사람과 공유하기 위해 미리 서명된 URL을 생성하는 방법을 보여줍니다. 객체 소유자는 공유할 때 자체 보안 자격 증명을 지정하므로 미리 서명된 URL을 받는 사람은 누구나 제한된 시간 동안 객체에 액세스할 수 있습니다.

사전 조건

- AWSCloudShellFullAccess정책에서 제공하는 액세스 권한을 가진 IAM 사용자입니다.
- 미리 서명된 URL을 생성하는 데 필요한 IAM 권한은 Amazon Simple Storage Service 사용 설명서의 다른 사람과 객체 공유를 참조하십시오.

1단계: Amazon S3 버킷에 액세스할 수 있는 IAM 역할 생성

1. 공유할 수 있는 IAM 세부 정보를 가져오려면 에서`get-caller-identity` 명령을AWS CloudShell 호출하십시오.

   ```
   aws sts get-caller-identity
   ```

   명령줄에 다음과 비슷한 응답이 표시됩니다.

   ```
   {
       "Account": "123456789012",
       "UserId": "AROAXXOZUUOTTWDCVIDZ2:redirect_session",
       "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
   }
   ```

2. 이전 단계에서 가져온 사용자 정보를 가져와서AWS CloudFormation 템플릿에 추가합니다. 이 템플릿은 IAM 역할 생성. 이 역할은 공동 작업자에게 공유 리소스에 대한 최소 권한 권한을 부여합니다.

   ```
   Resources:
     CollaboratorRole:
       Type: AWS::IAM::Role
       Properties:
         AssumeRolePolicyDocument:
           Version: 2012-10-17
   ```

```
        Statement:
          - Effect: Allow
            Principal:
              AWS: "arn:aws:iam::531421766567:role/Feder08"
            Action: "sts:AssumeRole"
      Description: Role used by my collaborators
      MaxSessionDuration: 7200
  CollaboratorPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 's3:*'
            Resource: 'arn:aws:s3:::<YOUR_BUCKET_FOR_FILE_TRANSFER>'
            Condition:
              StringEquals:
                s3:prefix:
                 - "myfolder/*"
      PolicyName: S3ReadSpecificFolder
      Roles:
        - !Ref CollaboratorRole
Outputs:
  CollaboratorRoleArn:
    Description: Arn for the Collaborator's Role
    Value: !GetAtt CollaboratorRole.Arn
```

3. 이름이 지정된 파일에AWS CloudFormation 템플릿을 저장합니다`template.yaml`.
4. 템플릿을 사용하여 스택을 배포하고`deploy` 명령을 호출하여 IAM 역할을 생성합니다.

```
aws cloudformation deploy --template-file ./template.yaml --stack-name
 CollaboratorRole --capabilities CAPABILITY_IAM
```

미리 서명된 URL 생성

1. 에서AWS CloudShell 편집기를 사용하여 다음 코드를 추가합니다. 이 코드는 페더레이션 사용자에게 직접 액세스할 수 있는 URL을 생성합니다AWS Management Console.

```
import urllib, json, sys
```

```
import requests
import boto3
import os

def main():
  sts_client = boto3.client('sts')
  assume_role_response = sts_client.assume_role(
      RoleArn=os.environ.get(ROLE_ARN),
      RoleSessionName="collaborator-session"
  )
  credentials = assume_role_response['Credentials']
  url_credentials = {}
  url_credentials['sessionId'] = credentials.get('AccessKeyId')
  url_credentials['sessionKey'] = credentials.get('SecretAccessKey')
  url_credentials['sessionToken'] = credentials.get('SessionToken')
  json_string_with_temp_credentials = json.dumps(url_credentials)
  print(f"json string {json_string_with_temp_credentials}")

  request_parameters = f"?
Action=getSigninToken&Session={urllib.parse.quote(json_string_with_temp_credentials)}"
  request_url = "https://signin.aws.amazon.com/federation" + request_parameters
  r = requests.get(request_url)
  signin_token = json.loads(r.text)
  request_parameters = "?Action=login"
  request_parameters += "&Issuer=Example.org"
  request_parameters += "&Destination=" + urllib.parse.quote("https://us-
west-2.console.aws.amazon.com/cloudshell")
  request_parameters += "&SigninToken=" + signin_token["SigninToken"]
  request_url = "https://signin.aws.amazon.com/federation" + request_parameters

  # Send final URL to stdout
  print (request_url)

if __name__ == "__main__":
  main()
```

2. 라는 파일에 코드를 저장합니다`share.py`.
3. 명령줄에서 다음을 실행하여 IAM 역할의 Amazon 리소스 이름 (ARN) 을AWS CloudFormation 검색합니다. 그런 다음Python 스크립트에서 이를 사용하여 임시 보안 자격 증명을 획득합니다.

```
ROLE_ARN=$(aws cloudformation describe-stacks --stack-name CollaboratorRole --query
 "Stacks[*].Outputs[?OutputKey=='CollaboratorRoleArn'].OutputValue" --output text)
 python3 ./share.py
```

스크립트는 공동 작업자가 클릭하여 가져올 수 있는 URL을 반환합니다AWS Management Console.AWS CloudShell 협력자는 향후 3,600초 (1시간) 동안 Amazon S3 버킷의`myfolder/` 폴더를 완전히 제어할 수 있습니다. 자격 증명은 한 시간 후에 만료됩니다. 이 시간이 지나면 협력자는 더 이상 버킷에 액세스할 수 없습니다.

자습서: 내부에 AWS CloudShell Docker 컨테이너를 구축하여 Amazon ECR 리포지토리로 푸시하기

이 자습서에서는 Docker 컨테이너를 정의 및 구축하고 Amazon ECR AWS CloudShell 리포지토리로 푸시하는 방법을 보여줍니다.

사전 조건

- Amazon ECR 리포지토리를 생성하고 푸시하려면 필요한 권한이 있어야 합니다. Amazon ECR을 사용하는 리포지토리에 대한 자세한 내용은 Amazon ECR 사용 설명서의 Amazon ECR 프라이빗 리포지토리를 참조하십시오. Amazon ECR을 사용하여 이미지를 푸시하는 데 필요한 권한에 대한 자세한 내용은 Amazon ECR 사용 설명서의 이미지 푸시에 필요한 IAM 권한을 참조하십시오.

자습서 절차

다음 자습서에서는 CloudShell 인터페이스를 사용하여 Docker 컨테이너를 구축하고 Amazon ECR 리포지토리로 푸시하는 방법을 간략하게 설명합니다.

1. 홈 디렉터리에 새 폴더를 생성합니다.

```
mkdir ~/docker-cli-tutorial
```

2. 생성한 폴더로 이동합니다.

```
cd ~/docker-cli-tutorial
```

3. 빈 Dockerfile을 생성합니다.

```
touch Dockerfile
```

4. 예를 nano Dockerfile 들어 텍스트 편집기를 사용하여 파일을 열고 다음 내용을 붙여넣습니다.

```
# Dockerfile

# Base this container on the latest Amazon Linux version
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install the cowsay binary
RUN dnf install --assumeyes cowsay

# Default entrypoint binary
ENTRYPOINT [ "cowsay" ]

# Default argument for the cowsay entrypoint
CMD [ "Hello, World!" ]
```

5. 이제 Dockerfile을 빌드할 준비가 되었습니다. 를 실행하여 컨테이너를 빌드합니다. docker build 향후 명령에 사용할 easy-to-type 이름으로 컨테이너에 태그를 지정합니다.

```
docker build --tag test-container .
```

뒤에 마침표 (.) 를 포함해야 합니다.

6. 이제 컨테이너를 테스트하여 컨테이너가 제대로 실행되고 있는지 확인할 수 있습니다AWS CloudShell.

```
docker container run test-container
```

7. 이제 작동하는 Docker 컨테이너를 확보했으므로 Amazon ECR 리포지토리로 푸시해야 합니다. 기존 Amazon ECR 리포지토리가 있는 경우 이 단계를 건너뛰어도 됩니다.

다음 명령을 실행하여 이 자습서를 위한 Amazon ECR 리포지토리를 생성합니다.

```
ECR_REPO_NAME=docker-tutorial-repo
aws ecr create-repository --repository-name ${ECR_REPO_NAME}
```

8. Amazon ECR 리포지토리를 생성한 후 Docker 컨테이너를 해당 리포지토리로 푸시할 수 있습니다.

   다음 명령을 실행하여 Docker용 Amazon ECR 로그인 자격 증명을 가져옵니다.

   ```
   AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
   ECR_URL=${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_REGION}.amazonaws.com
   aws ecr get-login-password | docker login --username AWS --password-stdin
    ${ECR_URL}
   ```

9. 대상 Amazon ECR 리포지토리로 이미지에 태그를 지정한 다음 해당 리포지토리로 푸시합니다.

   ```
   docker tag test-container ${ECR_URL}/${ECR_REPO_NAME}
   docker push ${ECR_URL}/${ECR_REPO_NAME}
   ```

   이 자습서를 완료하려고 할 때 오류가 발생하거나 문제가 발생하는 경우 이 안내서의 문제 해결 섹션을 참조하십시오.

정리

이제 Docker 컨테이너를 Amazon ECR 리포지토리에 성공적으로 배포했습니다. 이 자습서에서 생성한 파일을 AWS CloudShell 환경에서 제거하려면 다음 명령을 실행합니다.

- ```
  cd ~
  rm -rf ~/docker-cli-tutorial
  ```
- Amazon ECR 리포지토리를 삭제합니다.

  ```
  aws ecr delete-repository --force --repository-name ${ECR_REPO_NAME}
  ```

자습서: 를 사용하여 Lambda 함수 배포하기 AWS CDK

이 자습서에서는 를 사용하여 계정에 Lambda 함수를 배포하는 방법을 보여줍니다. AWS Cloud Development Kit (AWS CDK)
  ```

사전 조건

- 에서 사용할 수 있도록 계정을 부트스트랩하십시오. AWS CDK 를 통한 부트스트래핑에 대한 자세한 내용은 v2 개발자 AWS CDK 안내서의 부트스트래핑을 참조하십시오. AWS CDK 계정을 부트스트랩하지 않았다면 에서 실행할 수 있습니다. `cdk bootstrap` CloudShell

- 계정에 리소스를 배포할 수 있는 적절한 권한이 있는지 확인하세요. 관리자 권한을 사용하는 것이 좋습니다.

튜토리얼 절차

다음 자습서에서는 를 사용하여 Docker 컨테이너 기반 Lambda 함수를 배포하는 방법을 간략하게 설명합니다. AWS CDK

1. 홈 디렉터리에 새 폴더를 생성합니다.

```
mkdir ~/docker-cdk-tutorial
```

2. 생성한 폴더로 이동합니다.

```
cd ~/docker-cdk-tutorial
```

3. AWS CDK종속성을 로컬에 설치합니다.

```
npm install aws-cdk aws-cdk-lib
```

4. 생성한 폴더에 스켈레톤 AWS CDK 프로젝트를 생성합니다.

```
touch cdk.json
mkdir lib
touch lib/docker-tutorial.js lib/Dockerfile lib/hello.js
```

5. 예를 들어`nano cdk.json`, 텍스트 편집기를 사용하여 파일을 열고 다음 내용을 붙여넣습니다.

```
{
  "app": "node lib/docker-tutorial.js"
}
```

6. `lib/docker-tutorial.js`파일을 열고 다음 내용을 붙여넣습니다.

```
// this file defines the CDK constructs we want to deploy

const { App, Stack } = require('aws-cdk-lib');
const { DockerImageFunction, DockerImageCode } = require('aws-cdk-lib/aws-lambda');
const path = require('path');

// create an application
const app = new App();

// define stack
class DockerTutorialStack extends Stack {
  constructor(scope, id, props) {
    super(scope, id, props);

    // define lambda that uses a Docker container
    const dockerfileDir = path.join(__dirname);
    new DockerImageFunction(this, 'DockerTutorialFunction', {
      code: DockerImageCode.fromImageAsset(dockerfileDir),
      functionName: 'DockerTutorialFunction',
    });
  }
}

// instantiate stack
new DockerTutorialStack(app, 'DockerTutorialStack');
```

7. 를 lib/Dockerfile 열고 다음 내용을 붙여넣습니다.

```
# Use a NodeJS 20.x runtime
FROM public.ecr.aws/lambda/nodejs:20

# Copy the function code to the LAMBDA_TASK_ROOT directory
# This environment variable is provided by the lambda base image
COPY hello.js ${LAMBDA_TASK_ROOT}

# Set the CMD to the function handler
CMD [ "hello.handler" ]
```

8. lib/hello.js파일을 열고 다음 내용을 붙여넣습니다.

```
// define the handler
exports.handler = async (event) => {
```

```
  // simply return a friendly success response
  const response = {
    statusCode: 200,
    body: JSON.stringify('Hello, World!'),
  };
  return response;
};
```

9. AWS CDKCLI를 사용하여 프로젝트를 합성하고 리소스를 배포합니다. 계정을 부트스트랩해야 합니다.

```
npx cdk synth
npx cdk deploy --require-approval never
```

10. Lambda 함수를 호출하여 확인 및 확인합니다.

```
aws lambda invoke --function-name DockerTutorialFunction out.json
jq . out.json
```

이제 를 사용하여 Docker 컨테이너 기반 Lambda 함수를 성공적으로 배포했습니다. AWS CDK 에 대한 자세한 내용은 v2 개발자 AWS CDK 안내서를 참조하십시오. AWS CDK 이 자습서를 완료하려고 할 때 오류가 발생하거나 문제가 발생하는 경우 이 가이드의 문제 해결 섹션을 참조하여 도움을 받으십시오.

정리

이제 를 사용하여 Docker 컨테이너 기반 Lambda 함수를 성공적으로 배포했습니다. AWS CDK AWS CDK프로젝트 내에서 다음 명령을 실행하여 관련 리소스를 삭제합니다. 삭제를 확인하라는 메시지가 표시됩니다.

-
```
npx cdk destroy DockerTutorialStack
```

- 이 자습서에서 만든 파일 및 리소스를 AWS CloudShell 환경에서 제거하려면 다음 명령을 실행합니다.

```
cd ~
rm -rf ~/docker-cli-tutorial
```

AWS CloudShell 작업

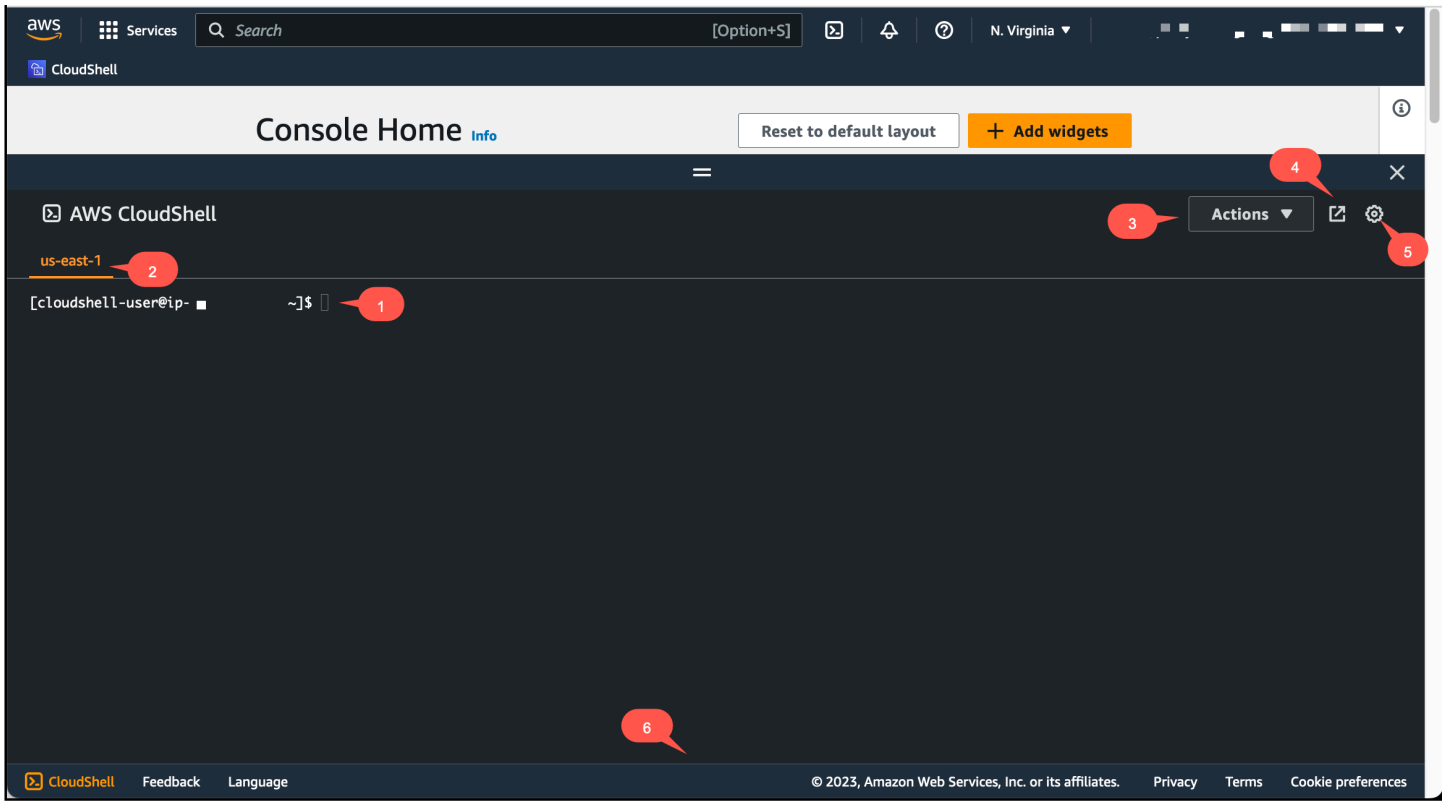
이 섹션에서는 지원되는 AWS CloudShell 애플리케이션과 상호 작용하고 특정 작업을 수행하는 방법을 설명합니다.

주제

AWS CloudShell 인터페이스 탐색

AWS Management Console및 에서 CloudShell 인터페이스 기능을 탐색할 수 Console Toolbar 있습니다.

다음 스크린샷은 주요 AWS CloudShell 인터페이스 기능을 나타냅니다.



1. AWS CloudShell 명령줄 인터페이스는 선호하는 쉘을 사용하여 명령을 실행하는 데 사용됩니다. 현재 쉘 유형은 명령 프롬프트에 표시됩니다.
2. 터미널 탭은 AWS CloudShell이(가) 현재 실행되는 AWS 리전을(를) 사용합니다.
3. 작업 메뉴에는 화면 레이아웃 변경, 파일 다운로드 및 업로드, 홈 디렉터리 재시작AWS CloudShell 및 삭제AWS CloudShell 옵션이 있습니다.

> **Note**
>
> CloudShell 에서 실행할 때는 다운로드 옵션을 사용할 수 없습니다Console Toolbar.

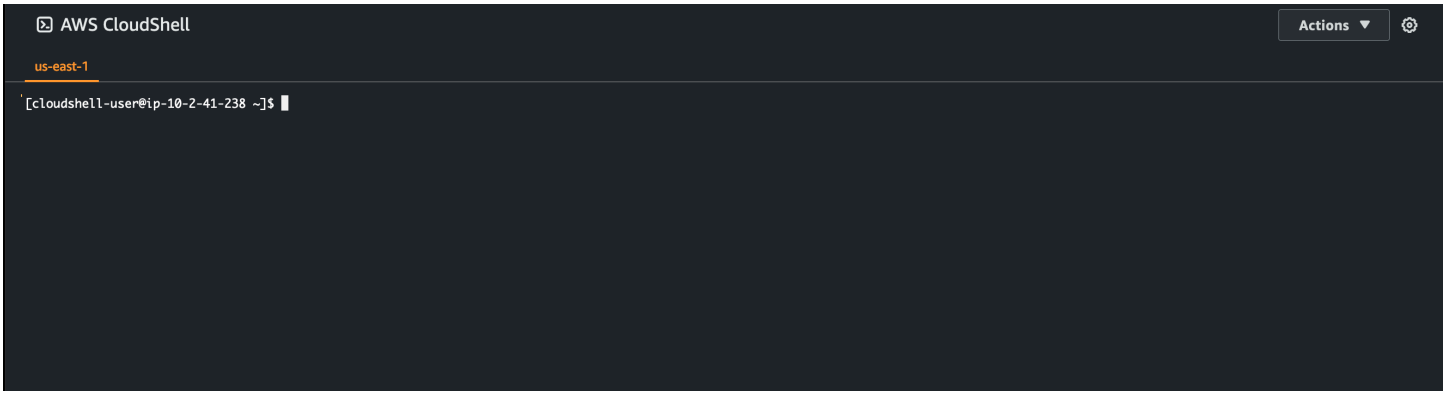
4. 전체 화면에서 CloudShell 세션에 접근할 수 있는 옵션을 제공하는 새 브라우저에서 열기 탭입니다.
5. 환경 설정 옵션은 쉘 환경을 사용자 지정할 때 사용할 수 있습니다.
6. 하단 표시줄에는 다음과 같은 옵션이 있습니다.
   - CloudShell CloudShell아이콘에서 실행합니다.
   - 피드백 아이콘에서 피드백을 제공합니다. 제출하고자 하는 피드백 유형을 선택하고 의견을 추가한 다음 제출을 선택합니다.
     - 피드백을 제출하려면 다음 옵션 중 하나를 선택하십시오. CloudShell
       - 콘솔에서 실행하고 CloudShell 피드백을 선택합니다. 의견을 추가한 다음 제출을 선택합니다.
       - 콘솔 왼쪽 하단에서 을 선택한 다음 새 브라우저에서 열기 탭 아이콘인 피드백을 선택합니다. CloudShellConsole Toolbar 의견을 추가한 다음 제출을 선택합니다.

     > **Note**
     >
     > CloudShell 에서 실행할 때는 피드백 옵션을 사용할 수 없습니다Console Toolbar.

   - 개인정보 처리방침과 이용 약관을 살펴보고 쿠키 환경을 사용자 지정합니다.

AWS 리전에서 작업하기

현재 실행 중인 AWS 리전이(가) 명령줄 인터페이스 위에 표시됩니다.



리전 선택기로 특정 리전을 선택하여 AWS 리전에서 작업할 수 있습니다. 리전 변경 후, 쉘 세션이 선택된 리전에서 실행 중인 다른 컴퓨팅 환경에 연결되면서 인터페이스가 새로고침됩니다.

> **Important**
>
> AWS 리전에서 각각 최대 1GB의 영구 스토리지를 사용할 수 있습니다. 영구 스토리지는 홈 디렉터리($HOME)에 저장됩니다. 따라서 홈 디렉터리에 저장된 개인 파일, 디렉터리, 프로그램, 스크립트가 모두 하나의 AWS 리전에 위치하게 됩니다. 또한, 홈 디렉터리에 위치하고 다른 리전에 저장되어 있는 것과는 상이합니다.
> 영구 스토리지 내 장기간 파일 보관 역시 리전 단위로 관리됩니다. 자세히 알아보려면 영구 스토리지의 내용을 참조하세요.

AWS CLI에 대한 AWS 리전 기본값 지정하기

환경 변수를 사용하여 AWS CLI을(를) 사용하여 AWS 서비스에 액세스하는 데 필요한 구성 옵션과 보안 인증 정보를 지정할 수 있습니다. 쉘 세션 기본값 AWS 리전을(를) 지정하는 환경 변수는 AWS Management Console에 있는 특정 리전에서 AWS CloudShell를 시작할 때, 또는 리전 선택기에서 옵션을 선택할 때 설정됩니다.

환경 변수는 `aws configure`에서 업데이트되는 AWS CLI 보안 인증 정보 파일보다 우선합니다. 따라서 `aws configure` 명령을 실행하여 환경 변수로 지정된 리전을 변경할 수 없습니다. 대신, AWS CLI 명령의 기본 리전을 변경하려면 `AWS_REGION` 환경 변수에 값을 할당하세요. 다음 예시에서 `us-east-1`을(를) 현재의 리전으로 교체합니다.

Bash or Zsh

```
$ export AWS_REGION=us-east-1
```

환경 변수를 설정하면 사용되는 값이 변경되어 셸 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 셸의 스타트업 스크립트에서 변수를 설정하면 해당 변수가 향후 세션에서도 영구적으로 적용되도록 할 수 있습니다.

PowerShell

```
PS C:\> $Env:AWS_REGION="us-east-1"
```

PowerShell 프롬프트에서 환경 변수를 설정하는 경우 환경 변수는 현재 세션 기간 동안만 값을 저장합니다. 또는 PowerShell 프로필에 변수를 추가하여 향후 모든 PowerShell 세션에 사용할 변수를 설정할 수 있습니다. 환경 변수 저장에 대한 자세한 내용은 PowerShell 설명서를 참조하십시오.

기본 리전을 변경했는지 확인하려면 `aws configure list` 명령을 실행하여 현재의 AWS CLI 구성 데이터를 표시합니다.

> **Note**
>
> 특정 AWS CLI 명령의 경우 명령줄 옵션 `--region`을(를) 사용하여 기본 리전을 재정의할 수 있습니다. 자세한 내용은 AWS Command Line Interface 사용 설명서에서 명령줄 옵션을 참조하세요.

파일 및 스토리지 작업

AWS CloudShell 인터페이스를 사용하여 쉘 환경에서 파일을 업로드하고 다운로드할 수 있습니다. 파일 다운로드 및 업로드에 대한 자세한 정보는 AWS CloudShell 시작하기에서 확인하세요.

추가한 파일을 세션 종료 후 사용할 수 있게 하려면 영구 스토리지와 임시 스토리지의 차이점을 알아야 합니다.

- 영구 스토리지: AWS 리전별로 각각 1GB의 영구 스토리지가 있습니다. 영구 스토리지는 홈 디렉터리에 있습니다.
- 임시 스토리지: 임시 스토리지는 세션 종료 시 재활용됩니다. 임시 스토리지는 홈 디렉터리 외부 디렉터리에 있습니다.

> **Important**
>
> 향후 쉘 세션에서 사용할 파일은 홈 디렉터리에 남겨 두세요. 예를 들어, `mv` 명령을 실행하여 파일을 홈 디렉터리 밖으로 옮긴다고 가정해 보겠습니다. 그러면 현재 쉘 세션이 종료될 때 해당 파일이 재활용됩니다.

도커 사용 작업

AWS CloudShell설치 또는 구성 없이 Docker를 완벽하게 지원합니다. 내부에서 Docker 컨테이너를 정의, 빌드 및 실행할 수 있습니다. AWS CloudShell AWS CDK툴킷을 통해 Docker 컨테이너 기반 Lambda 함수와 같은 Docker 기반 리소스를 배포하고 Docker CLI를 통해 Docker 컨테이너를 빌드하고 Amazon ECR 리포지토리로 푸시할 수 있습니다. 이 두 배포를 모두 실행하는 방법에 대한 자세한 단계는 다음 자습서를 참조하십시오.

- 자습서: 를 사용하여 Lambda 함수 배포하기 AWS CDK
- 자습서: 내부에 AWS CloudShell Docker 컨테이너를 구축하여 Amazon ECR 리포지토리로 푸시하기

Docker를 다음과 함께 사용할 때는 몇 가지 제한 및 제한이 있습니다. AWS CloudShell

- Docker는 환경 내 공간이 제한되어 있습니다. 개별 이미지가 크거나 기존 Docker 이미지가 너무 많으면 문제가 발생하여 추가 이미지를 가져오거나 빌드하거나 실행하지 못할 수 있습니다. Docker에 대한 자세한 내용은 Docker 설명서 가이드를 참조하세요.
- Docker는 특정 지역에서만 지원됩니다. Docker에서 지원되는 지역에 대한 자세한 내용은 Docker 지역을 참조하십시오.
- Docker와 함께 AWS CloudShell 사용할 때 문제가 발생하는 경우 이 가이드의 문제 해결 섹션에서 이러한 문제를 잠재적으로 해결하는 방법에 대한 정보를 참조하세요.

에 대한 접근성 기능 사용하기AWS CloudShell

이 항목에서는 다음과 같은 접근성 기능을 사용하는 방법을 설명합니다.CloudShell. 키보드를 사용하여 페이지에서 초점을 맞출 수 있는 요소를 탐색할 수 있습니다. 모양을 사용자 정의할 수도 있습니다.CloudShell글꼴 크기 및 인터페이스 테마를 포함합니다.

내 키보드 내비게이션CloudShell

페이지에서 초점을 맞출 수 있는 요소를 탐색하려면 키를 누릅니다.Tab.

CloudShell터미널 접근성 기능

다음을 사용할 수 있습니다.Tab다음 모드에서 키를 사용하십시오.

- 터미널 모드 (기본값)— 이 모드에서는 터미널이 다음을 캡처합니다.Tab키 입력. 터미널에 초점이 맞춰진 후 키를 누릅니다.Tab터미널 기능에만 액세스하려면
- 내비게이션 모드— 이 모드에서는 단말기가 사용자의 데이터를 캡처하지 않습니다.Tab키 입력. 버튼을 누릅니다.Tab페이지에서 초점을 맞출 수 있는 요소를 탐색하려면

터미널 모드와 내비게이션 모드 사이를 전환하려면 키를 누릅니다.Ctrl+M. 다시 전환하고 나면탭: 내비게이션헤더에 표시되며 다음을 사용할 수 있습니다.Tab키를 사용하여 페이지를 탐색할 수 있습니다.

터미널 모드로 돌아가려면 키를 누릅니다.Ctrl+M. 아니면 선택하세요X다음탭: 내비게이션.

> **Note**
>
> 현재,CloudShell모바일 장치에서는 터미널 접근성 기능을 사용할 수 없습니다.

에서 글꼴 크기 및 인터페이스 테마 선택CloudShell

모양을 사용자 정의할 수 있습니다.CloudShell시각적 선호도에 맞게 조정할 수 있습니다.

- 글꼴 크기— 다음 중에서 선택가장 작음,스몰,미디엄,라지, 그리고가장 큰터미널의 글꼴 크기. 글꼴 크기 변경에 대한 자세한 내용은 을 참조하십시오.the section called “글꼴 크기 변경”.
- 테마— 다음 중에서 선택하세요라이트과다크인터페이스 테마. 인터페이스 테마 변경에 대한 자세한 내용은 을 참조하십시오.the section called “인터페이스 테마 변경”.

다음 지역AWS 서비스 이용AWS CloudShell

의AWS CloudShell 주요 이점은 이를 사용하여 명령줄 인터페이스에서AWS 서비스를 관리할 수 있다는 것입니다. 따라서 먼저 로컬에서 도구를 다운로드 및 설치하거나 자격 증명을 구성할 필요가 없습니다. AWS CloudShell실행하면 다음과 같은AWS 명령줄 도구가 이미 설치된 컴퓨팅 환경이 생성됩니다.

- AWS CLI
- AWS Elastic Beanstalk CLI
- 아마존 ECS CLI
- AWS SAM

이미AWS 로그인했으므로 서비스를 사용하기 전에 자격 증명을 로컬에서 구성할 필요가 없습니다. 에 로그인하는 데 사용한 자격 증명이 으로AWS Management Console 전달됩니다AWS CloudShell.

에 사용되는 기본AWS 지역을 변경하려면`AWS_REGION` 환경 변수에 할당된 값을 변경할 수 있습니다.AWS CLI (자세한 내용은 AWS CLI에 대한 AWS 리전 기본값 지정하기 섹션을 참조하세요.)

이 항목의 나머지 부분에서는 명령줄에서 를AWS CloudShell 사용하여 선택한AWS 서비스와 상호 작용하는 방법을 보여 줍니다.

AWS CLI선택한AWS 서비스의 명령줄 예제

다음 예는AWS CLI 버전 2에서 사용할 수 있는 명령을 사용하여 작업할 수 있는 수많은AWS 서비스 중 일부만을 나타냅니다. 전체 목록은 AWS CLI 명령 참조를 참조하십시오.

- DynamoDB
- AWS Cloud9
- Amazon EC2
- S3 글래이셔

DynamoDB

DynamoDB는 완전관리형 NoSQL 데이터베이스 서비스로서 원활한 확장성과 함께 빠르고 예측 가능한 성능을 제공합니다. 이 서비스의 NoSQL 모드 구현은 키-값 및 문서 데이터 구조를 지원합니다.

다음create-table 명령을 실행하면AWS 계정에 이름이 지정된 NoSQL 스타일MusicCollection 테이블이 생성됩니다.

```
aws dynamodb create-table \
    --table-name MusicCollection \
    --attribute-definitions AttributeName=Artist,AttributeType=S
 AttributeName=SongTitle,AttributeType=S \
    --key-schema AttributeName=Artist,KeyType=HASH
 AttributeName=SongTitle,KeyType=RANGE \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
    --tags Key=Owner,Value=blueTeam
```

자세한 내용은 사용AWS Command Line Interface 설명서의AWS CLI 과 함께 DynamoDB 사용을 참조하십시오.

AWS Cloud9

AWS Cloud9은 브라우저 창에서 코드를 작성, 실행 및 디버깅하는 데 사용할 수 있는 클라우드 기반 통합 개발 환경 (IDE) 입니다. 환경에는 코드 편집기, 디버거 및 터미널이 있습니다.

다음create-environment-ec2 명령은 지정된 설정으로AWS Cloud9 EC2 개발 환경을 생성합니다. Amazon EC2 인스턴스를 시작한 후 인스턴스에서 환경으로 연결합니다.

```
aws cloud9 create-environment-ec2 --name my-demo-env --description "My demonstration
 development environment." --instance-type t2.micro --subnet-id subnet-1fab8aEX --
automatic-stop-time-minutes 60 --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

자세한 내용은 AWS Cloud9명령줄 참조를 참조하세요.

Amazon EC2

Amazon Ek (Amazon EC2) 는 클라우드에서 안전하고 크기 조정 가능한 컴퓨팅 용량을 제공하는 웹 서비스입니다. 웹 규모 컴퓨팅 작업을 보다 쉽고 쉽게 사용할 수 있도록 설계되었습니다.

다음run-instances 명령은 지정한 VPC 서브넷에서 t2.micro 인스턴스를 시작합니다.

```
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --key-
name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

자세한 내용은 사용AWS Command Line Interface 설명서의AWS CLI 과 함께 Amazon EC2 사용을 참조하십시오.

S3 Glacier

S3 Glacier 및 S3 Glacier 및 S3 Glacier 및 S3 Glacier 및 S3 Glacier 및 S3 Glacier 및 S3 Glacier 및 Amazon S3 Glacier 및 S3 Glacier 및 S3 Glacier

다음`create-vault` 명령은 아카이브를 저장하기 위한 컨테이너인 볼트를 생성합니다.

```
aws glacier create-vault --vault-name my-vault --account-id -
```

자세한 내용은 사용AWS Command Line Interface 설명서의AWS CLI 과 함께 Amazon S3 Glacier 사용을 참조하십시오.

AWSElastic Beanstalk

AWS Elastic BeanstalkCLI는 로컬 리포지토리에서 환경 생성, 업데이트 및 모니터링을 간소화하는 명령줄 인터페이스를 제공합니다. 이 컨텍스트에서 환경 은 애플리케이션 버전을 실행 중인AWS 리소스 컬렉션을 의미합니다.

다음`create` 명령은 사용자 지정 Amazon Virtual Private Cloud (VPC) 에서 새 환경을 생성합니다.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --
vpc.securitygroup sg-70cff265
```

자세한 내용은 AWS Elastic Beanstalk개발자 안내서의 EB CLI 명령 참조를 참조하십시오.

Amazon ECS CLI

Amazon Ek (Amazon ECS) 명령줄 인터페이스 (Amazon ECS) 명령줄 인터페이스 (CLI) 는 몇 가지 상위 수준 명령을 제공합니다. 로컬 개발 환경에서 클러스터 생성, 업데이트 및 모니터링 프로세스를 단순화하도록 설계되었습니다. Amazon ECS 클러스터 는 태스크 또는 서비스의 논리적 그룹입니다.

다음`configure` 명령은 라는 클러스터 구성을 생성하도록 Amazon ECS CLI를 구성합니다`ecs-cli-demo`. 이 클러스터 구성은 에서`ecs-cli-demo` 클러스터의 기본 시작`FARGATE` 유형으로 사용합니다`us-east-1 region`.

```
ecs-cli configure --region us-east-1 --cluster ecs-cli-demo --default-launch-type
 FARGATE --config-name ecs-cli-demo
```

자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 Amazon ECS 명령줄 참조를 참조하세요.

AWS SAM CLI

AWS SAMCLI는AWS Serverless Application Model 템플릿 및 애플리케이션 코드에서 작동하는 명령줄 도구입니다. 이를 사용하여 여러 작업을 수행할 수 있습니다. 여기에는 Lambda 함수를 로컬에서 호출하고, 서버리스 애플리케이션을 위한 배포 패키지를 생성하고, 서버리스 애플리케이션을AWS 클라우드에 배포하는 것이 포함됩니다.

다음`init` 명령은 필수 매개 변수를 매개 변수로 전달하여 새 SAM 프로젝트를 초기화합니다.

```
sam init --runtime python3.7 --dependency-manager pip --app-template hello-world --name
 sam-app
```

자세한 내용은 AWS Serverless Application Model개발자 안내서의AWS SAM CLI 명령 참조를 참조하십시오.

나만의 커스터마이징AWS CloudShell경험

다음과 같은 측면을 사용자 지정할 수 있습니다.AWS CloudShell경험:

- 탭 레이아웃: 명령줄 인터페이스를 여러 열과 행으로 분할합니다.
- 글꼴 크기: 명령줄 텍스트의 크기를 조정합니다.
- 색상 테마: 밝은 테마와 어두운 테마 사이를 전환합니다.
- 세이프 붙여넣기: 붙여넣기 전에 여러 줄 텍스트를 확인해야 하는 기능을 켜거나 끕니다.
- Tmux를 세션으로 복원: tmux를 사용하면 세션이 비활성화될 때까지 세션이 복원됩니다.

다음과 같이 셸 환경을 확장할 수도 있습니다.자체 소프트웨어 설치과시작 셸 스크립트 수정.

명령줄 디스플레이를 여러 탭으로 분할

명령줄 인터페이스를 여러 창으로 분할하여 여러 명령을 실행합니다.

> **Note**
>
> 탭을 여러 개 연 후 선택한 창의 아무 곳이나 클릭하여 작업하려는 탭을 선택할 수 있습니다. 를 선택하여 탭을 닫을 수 있습니다.x기호: 지역 이름 옆에 있습니다.

- 선택액션그리고 다음 옵션 중 하나는탭 레이아웃:
  - 새 탭: 현재 활성화된 탭 옆에 새 탭을 추가합니다.
  - 행으로 나누기: 현재 활성 탭 아래에 있는 행에 새 탭을 추가합니다.
  - 열로 나누기: 현재 활성화된 탭 옆에 있는 열에 새 탭을 추가합니다.

  공간이 충분하지 않아 각 탭을 완전히 표시할 수 없는 경우 스크롤하여 전체 탭을 확인하세요. 창을 구분하는 분할 막대를 선택하고 포인터로 드래그하여 창 크기를 늘리거나 줄일 수도 있습니다.

글꼴 크기 변경

명령줄 인터페이스에 표시되는 텍스트 크기를 늘리거나 줄입니다.

1. 변경하려면AWS CloudShell터미널 설정을 보려면 다음으로 이동하십시오.설정,기본 설정.
2. 글자 크기를 선택하세요. 옵션은 다음과 같습니다.가장 작음,스몰,미디엄,라지, 그리고가장 큰.

인터페이스 테마 변경

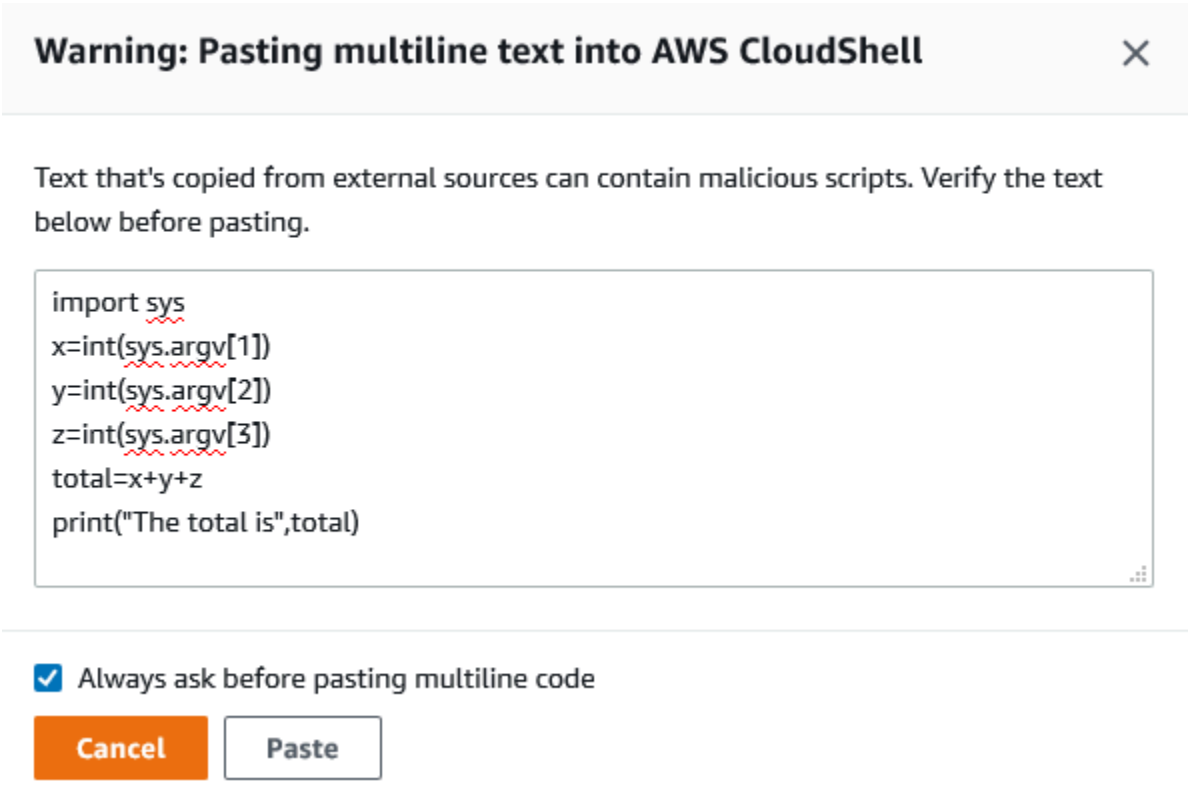
명령줄 인터페이스의 밝은 테마와 어두운 테마 사이를 전환합니다.

1. 변경하려면AWS CloudShell테마를 보려면 다음으로 이동하십시오.설정,기본 설정.
2. 고르세요라이트또는다크.

여러 줄 텍스트에 안전 붙여넣기 사용

안전 붙여넣기는 셸에 붙여넣으려는 여러 줄 텍스트에 악성 스크립트가 포함되어 있지 않은지 확인하라는 메시지를 표시하는 보안 기능입니다. 타사 사이트에서 복사한 텍스트에는 셸 환경에서 예상치 못한 동작을 유발하는 숨겨진 코드가 포함되어 있을 수 있습니다.

안전 붙여넣기 대화 상자에는 클립보드에 복사한 전체 텍스트가 표시됩니다. 보안 위험이 없다고 생각되면 다음을 선택하십시오.붙여넣기.



세이프 붙여넣기를 활성화하여 스크립트의 잠재적 보안 위험을 파악하는 것이 좋습니다. 선택하여 이 기능을 켜거나 끌 수 있습니다.환경설정,안전 붙여넣기 활성화과안전 붙여넣기 비활성화.

사용tmux세션 복원으로

AWS CloudShelltmux를 사용하여 단일 또는 다중 브라우저 탭에서 세션을 복원합니다. 브라우저 탭을 새로 고치면 세션이 비활성화될 때까지 세션이 재개됩니다. 자세한 내용은 을 참조하십시오.세션 복원.

에 대한 보안 AWS CloudShell

Amazon Web Services(AWS)에서 가장 우선순위가 높은 것이 클라우드 보안입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처를 활용할 수 있습니다. 보안은 기업과 기업 간의 AWS 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

클라우드 보안 — AWS 클라우드에서 제공되는 모든 서비스를 실행하는 인프라를 보호하고 안전하게 사용할 수 있는 서비스를 제공하는 역할을 합니다. AWS 당사의 보안 책임은 최우선 과제이며 AWS, AWS 규정 준수 프로그램의 일환으로 타사 감사자가 보안 효과를 정기적으로 테스트하고 검증합니다.

클라우드에서의 보안 — 사용자의 책임은 사용 중인 AWS 서비스와 데이터의 민감도, 조직의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요인에 따라 결정됩니다.

AWS CloudShell 지원하는 특정 AWS 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 AWS 규정 준수 프로그램의 규정 준수 노력 범위에 속하는 AWS 서비스를 참조하십시오.

다음 항목에서는 보안 및 규정 준수 목표를 AWS CloudShell 충족하도록 구성하는 방법을 보여줍니다.

주제

- 데이터 보호: AWS CloudShell
- AWS용 ID 및 액세스 관리 CloudShell
- 로그인 및 모니터링 AWS CloudShell
- 규정 준수 검증: AWS CloudShell
- 레질리언스: AWS CloudShell
- 의 인프라 보안 AWS CloudShell
- 의 구성 및 취약성 분석 AWS CloudShell
- 에 대한 보안 모범 사례 AWS CloudShell
- AWS CloudShell 보안 FAQ

데이터 보호: AWS CloudShell

AWS 공동 책임 모델 의 데이터 보호에 적용됩니다 AWS CloudShell. 이 모델에 설명된 대로 AWS 는 모든 데이터를 실행하는 글로벌 인프라를 보호하는 역할을 AWS 클라우드합니다. 이 인프라에서 호스

팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center OR AWS Identity and Access Management (IAM) 을 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이 방식을 사용하면 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 리소스와 통신할 수 있습니다. AWS TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하여 Amazon S3에 저장된 민감한 데이터를 검색하고 보호합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해 액세스할 때 FIPS 140-2로 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용하십시오. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조하십시오.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS CloudShell 또는 AWS 서비스 SDK를 사용하거나 다른 방법으로 작업하는 경우가 포함됩니다. AWS CLI AWS 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 보안 인증를 URL에 포함시켜서는 안 됩니다.

데이터 암호화

데이터 암호화란 저장된 데이터 (저장된 데이터 AWS CloudShell) 와 전송 중인 데이터 (AWS CloudShell 및 서비스 엔드포인트 간 이동 시) 를 보호하는 것을 말합니다.

미사용 시 암호화는 다음을 사용합니다. AWS KMS

유휴 데이터 암호화는 저장된 데이터를 암호화하여 무단 액세스로부터 데이터를 보호하는 것을 의미합니다. 사용 AWS CloudShell시 AWS 지역당 1GB의 영구 스토리지가 무료로 제공됩니다. 영구 스토

리지는 홈 디렉터리($HOME)에 있으며 사용자만 이용할 수 있습니다. 각 쉘 세션이 종료된 후 재활용되는 임시 환경 리소스와 달리, 홈 디렉터리의 데이터는 세션 간에 유지됩니다.

저장된 데이터의 암호화는 AWS Key Management Service (AWS KMS) 에서 AWS CloudShell 제공하는 암호화 키를 사용하여 구현됩니다. 이 AWS 서비스는 환경에 저장된 고객 데이터를 암호화하는 데 사용되는 암호화 키인 고객 마스터 키 (CMK) 를 생성하고 제어하는 관리형 서비스입니다. AWS CloudShell AWS CloudShell 고객을 대신하여 데이터 암호화를 위한 암호화 키를 생성하고 관리합니다.

전송 중 암호화

전송 중 데이터 암호화는 데이터가 통신 엔드포인트 간을 이동하는 동안 데이터를 가로채기에서 보호하는 것을 의미합니다.

기본적으로 클라이언트의 웹 브라우저 컴퓨터와 클라우드 기반 컴퓨터 간의 모든 데이터 AWS CloudShell 통신은 HTTPS/TLS 연결을 통해 모든 데이터를 전송하여 암호화됩니다.

통신에 HTTPS/TLS 사용을 활성화하기 위해 어떤 조치도 필요하지 않습니다.

AWS용 ID 및 액세스 관리 CloudShell

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 AWS 도와줍니다. IAM 관리자는 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. CloudShell IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

- 고객
- 자격 증명을 통한 인증
- 정책을 사용한 액세스 관리
- AWS가 IAM과 CloudShell 협력하는 방법
- AWS의 자격 증명 기반 정책 예제 CloudShell
- AWS CloudShell 자격 증명 및 액세스 문제 해결
- IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리

고객

사용하는 방식 AWS Identity and Access Management (IAM) 은 수행하는 작업에 따라 다릅니다. CloudShell

서비스 사용자 - CloudShell 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 CloudShell 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. CloudShell의 기능에 액세스할 수 없는 경우 AWS CloudShell 자격 증명 및 액세스 문제 해결 섹션을 참조하세요.

서비스 관리자 — 회사에서 CloudShell 리소스를 담당하는 경우 전체 액세스 권한이 있을 수 CloudShell 있습니다. 서비스 사용자가 액세스해야 하는 CloudShell 기능과 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요. 회사에서 IAM을 어떻게 사용할 수 있는지 자세히 CloudShell 알아보려면 을 참조하십시오AWS가 IAM과 CloudShell 협력하는 방법.

IAM 관리자 - IAM 관리자라면 CloudShell에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 CloudShell ID 기반 정책의 예를 보려면 을 참조하십시오. AWS의 자격 증명 기반 정책 예제 CloudShell

자격 증명을 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 연동 자격 증명으로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK) 와 명령줄 인터페이스 (CLI) 를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용자 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하세요.

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 태스크의 전체 목록은 IAM 사용자 설명서의 루트 사용자 보안 인증이 필요한 태스크를 참조하세요.

연동 보안 인증

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS Directory Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. AWS 계정 IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 사용자 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용자 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 보안 인증입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용

자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증 정보만 제공합니다. 자세한 정보는 IAM 사용자 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하세요.

IAM 역할

IAM 역할은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용자 설명서의 IAM 역할 사용을 참조하세요.

임시 보안 인증 정보가 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 연동 자격 증명에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 연동 자격 증명이 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용자 설명서의 Creating a role for a third-party Identity Provider(서드 파티 자격 증명 공급자의 역할 만들기) 부분을 참조하세요. IAM 자격 증명 센터를 사용하는 경우 권한 집합을 구성합니다. 인증 후 아이덴티티가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관 짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트 섹션을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 교차 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용자 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.
- 서비스 간 액세스 — 일부는 다른 AWS 서비스서비스의 기능을 AWS 서비스 사용합니다. 예를 들어 서비스에서 직접 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.

- 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.
- 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용자 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.
- 서비스 연결 역할 — 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증 정보를 얻을 수 있습니다. 자세한 정보는 IAM 사용자 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용자 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하세요.

정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책의 권한이 요청 허용 또는 거부 여부를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용자 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

보안 인증 기반 정책

보안 인증 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 보안 인증에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용자 설명서의 IAM 정책 생성을 참조하세요.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용자 설명서의 관리형 정책과 인라인 정책의 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 설명서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 ACL(액세스 제어 목록) 개요 섹션을 참조하세요.

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 특성입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 보안 인증 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용자 설명서의 IAM 엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU) 에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 특성을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함) 에 대한 권한을 제한합니다. AWS 계정 루트 사용자조직 및 SCP에 대한 자세한 정보는 AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 자격 증명 기반 정책의 교집합과 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용자 설명서의 세션 정책을 참조하세요.

여러 정책 타입

여러 정책 타입이 요청에 적용되는 경우 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련되어 있을 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

AWS가 IAM과 CloudShell 협력하는 방법

IAM을 사용하여 액세스를 CloudShell 관리하기 전에 어떤 IAM 기능과 함께 사용할 수 있는지 알아보십시오. CloudShell

AWS에서 사용할 수 있는 IAM 기능 CloudShell

IAM 특성	CloudShell 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책 내 태그)	아니요
임시 보안 인증	예
전달 액세스 세션(FAS)	아니요
서비스 역할	아니요
서비스 링크 역할	아니요

CloudShell 및 기타 AWS 서비스가 대부분의 IAM 기능과 어떻게 작동하는지 자세히 알아보려면 IAM 사용 설명서의 IAM과 함께 작동하는AWS 서비스를 참조하십시오.

ID 기반 정책은 다음과 같습니다. CloudShell

ID 기반 정책 지원	예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용자 설명서의 IAM 정책 생성(Creating IAM policies)을 참조합니다.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 태스크와 리소스 뿐만 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 자격 증명 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용자 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

다음에 대한 ID 기반 정책 예제 CloudShell

CloudShell ID 기반 정책의 예를 보려면 을 참조하십시오. AWS의 자격 증명 기반 정책 예제 CloudShell

내 리소스 기반 정책 CloudShell

리소스 기반 정책 지원	아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

크로스 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념합니다. 보안 주체와 리소스가 다른 AWS 계정경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 개체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 개체에 자격 증명 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 정보는 IAM 사용자 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

에 대한 정책 조치 CloudShell

정책 작업 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

CloudShell 작업 목록을 보려면 서비스 권한 부여 참조의 CloudShell AWS에서 정의한 작업을 참조하십시오. 일부 작업에는 API가 두 개 이상 있을 수 있습니다.

정책 조치는 조치 앞에 다음 접두사를 CloudShell 사용합니다.

```
cloudshell
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "cloudshell:action1",
      "cloudshell:action2"
         ]
```

CloudShell ID 기반 정책의 예를 보려면 을 참조하십시오. AWS의 자격 증명 기반 정책 예제 CloudShell

에 대한 정책 리소스 CloudShell

정책 리소스 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 보고서에는 `Resource` 또는 `NotResource` 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 작업을 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

CloudShell 리소스 유형 및 ARN 목록을 보려면 서비스 권한 부여 참조의 CloudShell AWS에서 정의한 리소스를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업에 대해 알아보려면 AWS에서 정의한 작업을 참조하십시오. CloudShell

CloudShell 자격 증명 기반 정책의 예를 보려면 을 참조하십시오. AWS의 자격 증명 기반 정책 예제 CloudShell

에 대한 정책 조건 키 CloudShell

서비스별 정책 조건 키 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 적음 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우 AWS (은)는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 `OR` 연산을 사용하여 조건을 AWS 평가합니다. 명령문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 정보는 IAM 사용자 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 AWS 설명서의 글로벌 조건 컨텍스트 키를 참조하십시오.

CloudShell 조건 키 목록을 보려면 서비스 권한 부여 CloudShell 참조의 AWS용 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업 및 리소스에 대해 알아보려면 AWS에서 정의한 작업을 참조하십시오 CloudShell.

CloudShell 자격 증명 기반 정책의 예를 보려면 을 참조하십시오. AWS의 자격 증명 기반 정책 예제 CloudShell

내 ACL CloudShell

ACL 지원	아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 설명서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ABAC 포함 CloudShell

ABAC 지원(정책의 태그)	아니요

ABAC(속성 기반 액세스 제어)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 개체 (사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그를 기반으로 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우 값은 서비스에 대해 예(Yes)입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우 값은 부분적(Partial)입니다.

ABAC에 대한 자세한 정보는 IAM 사용자 설명서의 ABAC란 무엇인가요?를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용자 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

다음과 같은 임시 자격 증명 사용 CloudShell

임시 보안 인증 정보 지원	예

임시 자격 증명을 사용하여 로그인하면 일부 자격 증명이 AWS 서비스 작동하지 않습니다. 임시 자격 증명을 사용하는 방법을 AWS 서비스 비롯한 추가 정보는 IAM 사용 설명서의 IAM과AWS 서비스 연동되는 내용을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하면 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 SSO (Single Sign-On) 링크를 AWS 사용하여 액세스하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증 정보를 자동으로 생성합니다. 역할 전환에 대한 자세한 정보는 IAM 사용자 설명서의 역할로 전환(콘솔)을 참조하세요.

또는 API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS CLI . AWS 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 인증 정보 섹션을 참조하세요.

역할을 전환하면 다른 환경을 사용하게 됩니다. 동일한 AWS CloudShell 환경 내에서는 역할을 전환할 수 없습니다.

포워드 액세스 세션 대상 CloudShell

전달 액세스 세션(FAS) 지원	아니요

IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 사용자는 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

CloudShell의 서비스 역할

서비스 역할 지원	아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM role(IAM 역할)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용자 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할의 권한을 변경하면 CloudShell 기능이 중단될 수 있습니다. 서비스 역할을 편집하기 위한 지침이 CloudShell 제공되는 경우에만 서비스 역할을 편집하십시오.

서비스 연결 역할은 다음과 같습니다. CloudShell

서비스 연결 역할 지원	아니요

서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

AWS의 자격 증명 기반 정책 예제 CloudShell

기본적으로 사용자 및 역할에는 CloudShell 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용자 설명서의 IAM 정책 생성을 참조하세요.

각 리소스 유형의 ARN 형식을 비롯하여 CloudShell 에서 정의한 CloudShell 작업 및 리소스 유형에 대한 자세한 내용은 서비스 권한 부여 참조의 AWS용 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- CloudShell 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

ID 기반 정책은 누군가가 계정에서 CloudShell 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. 자격 증명 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하여 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용자 설명서의 AWS managed policies(관리형 정책) 또는 AWS managed policies for job functions(직무에 대한 관리형 정책)를 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 least-privilege permissions(최소 권한)으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용자 설명서에 있는 Policies and permissions in IAM(IAM의 정책 및 권한)을 참조하세요.
- Use conditions in IAM policies to further restrict access(IAM 정책의 조건을 사용하여 액세스 추가 제한) – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 생성할 수 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 정보는 IAM 사용자 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 권장 사항을 제공하여 안전하고 기능적인 정책을 생성하도록 돕습니다. 자세한 정보는 IAM 사용자 설명서의 IAM Access Analyzer policy validation(IAM Access Analyzer 정책 검증)을 참조하세요.
- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용자 설명서의 Configuring MFA-protected API access(MFA 보호 API 액세스 구성)를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용자 설명서의 IAM의 보안 모범 사례를 참조하세요.

CloudShell 콘솔 사용

AWS CloudShell 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 내 CloudShell 리소스의 세부 정보를 나열하고 볼 수 있어야 AWS 계정합니다. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 CloudShell 콘솔을 계속 사용할 수 있도록 하려면 엔티티에 CloudShell *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 연결하세요. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
```

```
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS CloudShell 자격 증명 및 액세스 문제 해결

다음 정보를 사용하면 및 IAM을 사용할 때 발생할 수 있는 일반적인 문제를 CloudShell 진단하고 해결하는 데 도움이 됩니다.

주제

- 저는 다음과 같은 작업을 수행할 권한이 없습니다. CloudShell
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 외부 사용자가 내 CloudShell 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

저는 다음과 같은 작업을 수행할 권한이 없습니다. CloudShell

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 awes:*GetWidget* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

이 경우 awes:*GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요하면 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 CloudShell에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 AWS 서비스 수 있는 기능도 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 CloudShell에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우 Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 CloudShell 리소스에 액세스할 수 있도록 AWS 계정 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 이러한 기능의 CloudShell 지원 여부를 알아보려면 을 참조하십시오AWS가 IAM과 CloudShell 협력하는 방법.
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 IAM 사용 설명서에서 자신이 소유한 다른 AWS 계정 IAM 사용자에게 액세스 권한 제공을 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정AWS 계정 소유에 대한 액세스 제공을 참조하십시오.
- 자격 증명 연동을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용자 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 연동)을 참조하세요.

- 교차 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용자 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리

AWS Identity and Access Management (IAM) 에서 제공할 수 있는 액세스 관리 리소스를 사용하여 관리자는 IAM 사용자에게 권한을 부여할 수 있습니다. 이런 방식으로 사용자들은 AWS CloudShell 에 액세스하고 환경 기능을 사용할 수 있습니다. 관리자는 또한 해당 사용자가 쉘 환경에서 수행할 수 있는 작업을 세부적으로 지정하는 정책을 만들 수 있습니다.

관리자가 사용자에게 액세스 권한을 부여하는 가장 빠른 방법은 관리형 AWS 정책을 사용하는 것입니다. AWS 관리형 정책은 AWS에서 생성 및 관리하는 독립 실행형 정책입니다. 다음과 같은 AWS 관리형 정책을 IAM ID에 연결할 AWS CloudShell 수 있습니다.

- AWS CloudShellFullAccess: 모든 기능에 대한 전체 액세스 권한과 함께 AWS CloudShell 을 사용할 수 있는 권한을 부여합니다.

AWS CloudShellFullAccess정책은 와일드카드 (*) 문자를 사용하여 IAM ID (사용자, 역할 또는 그룹) 에 및 기능에 대한 전체 액세스 권한을 부여합니다. CloudShell 이 정책에 대한 자세한 내용은 AWS 관리형 정책 사용 설명서를 참조하십시오 AWS CloudShellFullAccess.

> **Note**
>
> 다음과 같은 AWS 관리형 정책을 사용하는 IAM ID도 시작할 수 있습니다. CloudShell 단, 이러한 정책은 광범위한 권한을 제공합니다. 따라서 IAM 사용자의 직무에 필수적인 경우에만 이런 정책을 부여할 것을 권장합니다.
>
> - 관리자: IAM 사용자에게 전체 액세스 권한을 제공하고 해당 사용자가 모든 서비스 및 리소스에 권한을 위임할 수 있도록 합니다. AWS
> - 개발자 파워 유저: IAM 사용자가 애플리케이션 개발 작업을 수행하고 AWS 인식 가능한 애플리케이션 개발을 지원하는 리소스 및 서비스를 생성 및 구성할 수 있도록 합니다.
>
> 관리 정책을 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가 (콘솔)을 참조하십시오.

사용자 지정 정책 AWS CloudShell 사용 시 허용 가능한 조치 관리

IAM 사용자가 수행할 수 있는 작업을 관리하려면 CloudShellPolicy 관리형 정책을 템플릿으로 CloudShell 사용하는 사용자 지정 정책을 생성하십시오. 아니면 관련 IAM 자격 증명(사용자, 그룹, 역할)에 내장된 인라인 정책을 편집합니다.

예를 들어 IAM 사용자의 액세스는 CloudShell 허용하지만 로그인에 사용되는 CloudShell 환경 자격 증명은 전달하지 못하게 할 수 있습니다. AWS Management Console

> **Important**
>
> AWS CloudShell 에서 시작하려면 IAM 사용자에게 다음 AWS Management Console작업에 대한 권한이 필요합니다.
>
> - `CreateEnvironment`
> - `CreateSession`
> - `GetEnvironmentStatus`
> - `StartEnvironment`
>
> 연결된 정책에서 이러한 작업 중 하나를 명시적으로 허용하지 않는 경우, 시작하려고 하면 IAM 권한 오류가 반환됩니다. CloudShell

AWS CloudShell 권한

명칭	부여된 권한에 대한 설명	시작하는 데 CloudShell 필요한가요?
`cloudshell:CreateEnvironment`	CloudShell 환경을 만들고, CloudShell 세션 시작 시 레이아웃을 검색하고, 웹 애플리케이션의 현재 레이아웃을 백엔드에 저장합니다. 이 권한은 에 *	예

명칭	부여된 권한에 대한 설명	시작하는 데 CloudShell 필요한가요?
	설명된 Resource 값으로만 예상됩니다. the section called “에 대한 IAM 정책의 예 CloudShell”	
cloudshell:CreateSession	에서 CloudShell 환경에 연결합니다. AWS Management Console	예
cloudshell:GetEnvironmentStatus	CloudShell 환경 상태를 읽어보십시오.	예
cloudshell:DeleteEnvironment	CloudShell 환경을 삭제합니다.	아니요
cloudshell:GetFileDownloadUrls	CloudShell 웹 인터페이스를 CloudShell 사용하여 파일을 다운로드하는 데 사용되는 사전 서명된 Amazon S3 URL을 생성합니다.	아니요
cloudshell:GetFileUploadUrls	CloudShell 웹 인터페이스를 CloudShell 사용하여 파일을 업로드하는 데 사용되는 사전 서명된 Amazon S3 URL을 생성합니다.	아니요
cloudshell:PutCredentials	로그인하는 데 사용된 자격 증명을 to에 전달합니다. AWS Management Console CloudShell	아니요
cloudshell:StartEnvironment	중지된 CloudShell 환경을 시작합니다.	예

명칭	부여된 권한에 대한 설명	시작하는 데 CloudShell 필요한가요?
`cloudshell:StopEnvironment`	실행 중인 CloudShell 환경을 중지합니다.	아니요

에 대한 IAM 정책의 예 CloudShell

다음 예는 액세스 CloudShell 권한을 제한하는 정책을 생성하는 방법을 보여줍니다. 또한 쉘 환경에서 수행할 수 있는 작업을 알 수 있습니다.

다음 정책은 액세스 CloudShell 및 해당 기능에 대한 완전한 액세스 거부를 시행합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "DenyCloudShell",
        "Effect": "Deny",
        "Action": [
            "cloudshell:*"
        ],
        "Resource": "*"
    }]
}
```

다음 정책은 IAM 사용자가 액세스할 수 있도록 CloudShell 허용하지만 파일 업로드 및 다운로드를 위해 미리 서명된 URL을 생성하는 것은 차단합니다. 예를 들어, `wget`과 같은 클라이언트를 사용하여 환경 간 파일 전송은 계속할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "AllowUsingCloudshell",
        "Effect": "Allow",
        "Action": [
            "cloudshell:*"
        ],
        "Resource": "*"
    },
    {
```

```
        "Sid": "DenyUploadDownload",
        "Effect": "Deny",
        "Action": [
            "cloudshell:GetFileDownloadUrls",
            "cloudshell:GetFileUploadUrls"
        ],
        "Resource": "*"
    }]
}
```

다음 정책은 IAM 사용자의 액세스를 허용합니다. CloudShell 하지만 이 정책에서는 로그인하는 데 사용한 자격 증명이 환경으로 전달되지 AWS Management Console 않도록 합니다 CloudShell . 이 정책을 사용하는 IAM 사용자는 내에서 자격 증명을 수동으로 구성해야 합니다. CloudShell

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "AllowUsingCloudshell",
        "Effect": "Allow",
        "Action": [
            "cloudshell:*"
        ],
        "Resource": "*"
    },
    {
        "Sid": "DenyCredentialForwarding",
        "Effect": "Deny",
        "Action": [
            "cloudshell:PutCredentials"
        ],
        "Resource": "*"
    }]
}
```

다음 정책은 IAM 사용자가 환경을 생성할 AWS CloudShell 수 있도록 허용합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "CloudShellUser",
        "Effect": "Allow",
```

```
        "Action": [
            "cloudshell:CreateEnvironment",
            "cloudshell:CreateSession",
            "cloudshell:GetEnvironmentStatus",
            "cloudshell:StartEnvironment"
        ],
        "Resource": "*"
    }]
}
```

액세스 권한 AWS 서비스

CloudShell 로그인할 때 사용한 IAM 자격 증명을 사용합니다. AWS Management Console

> **Note**
>
> 에 로그인할 때 사용한 IAM 자격 증명을 사용하려면 권한이 `cloudshell:PutCredentials` 있어야 합니다. AWS Management Console

의 이 사전 인증 기능을 CloudShell 사용하면 편리하게 사용할 수 있습니다. AWS CLI하지만 IAM 사용자에게는 여전히 명령줄에서 AWS 서비스 호출되는 것에 대한 명시적 권한이 필요합니다.

예를 들어, IAM 사용자가 Amazon S3 버킷을 생성하고 파일을 객체로 업로드해야 한다고 가정해 보겠습니다. 이러한 작업을 명시적으로 허용하는 정책을 생성할 수 있습니다. IAM 콘솔에는 JSON 형식의 정책 문서를 작성하는 프로세스를 안내하는 대화형 비주얼 편집기가 있습니다. 정책 생성 후, 관련 IAM 자격 증명(사용자, 그룹 또는 역할)에 연결할 수 있습입니다.

관리 정책을 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명 권한 추가(콘솔)을 참조하십시오.

로그인 및 모니터링 AWS CloudShell

이 항목에서는 를 사용하여 AWS CloudShell 활동 및 성능을 기록하고 모니터링하는 방법을 설명합니다 CloudTrail.

를 사용하여 활동을 모니터링합니다. CloudTrail

AWS CloudShell 사용자 AWS CloudTrail, 역할 또는 역할 AWS 서비스 내에서 수행한 작업의 기록을 제공하는 서비스와 통합됩니다 AWS CloudShell. CloudTrail 모든 API 호출을 AWS CloudShell 이벤트

로 캡처합니다. 캡처된 호출에는 AWS CloudShell 콘솔에서의 호출과 AWS CloudShell API에 대한 코드 호출이 포함됩니다.

트레일을 생성하면 Amazon Simple Storage Service (Amazon S3) 버킷으로 CloudTrail 이벤트를 지속적으로 전송할 수 있습니다. 여기에는 에 대한 이벤트가 포함됩니다 AWS CloudShell.

추적을 구성하지 않은 경우 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 요청에 대한 다양한 정보를 찾아볼 수 있습니다. 예를 들어, AWS에 어떤 요청이 이루어졌는지 CloudShell, 어떤 IP 주소를 통해 요청했는지, 누가 언제 요청했는지 알 수 있습니다.

AWS CloudShell 에서 CloudTrail

다음 표에는 CloudTrail 로그 파일에 저장된 AWS CloudShell 이벤트가 나열되어 있습니다.

> **Note**
>
> AWS CloudShell 다음을 포함하는 이벤트:
>
> - *변경하지 않는 (읽기 전용) API 호출임을 나타냅니다.
> - 이 단어는 `Environment` 셸 경험을 호스팅하는 컴퓨팅 환경의 수명 주기와 관련이 있습니다.
> - 이 단어는 CloudShell 터미널의 모든 브라우저 탭을 `Layout` 복원합니다.

CloudShell 이벤트: CloudTrail

이벤트 이름	설명
`createEnvironment`	CloudShell 환경이 생성될 때 발생합니다.
`createSession`	CloudShell 환경이 에서 연결될 때 발생합니다 AWS Management Console.
`deleteEnvironment`	CloudShell 환경이 삭제될 때 발생합니다.
`deleteSession`	현재 브라우저 CloudShell 탭에서 실행 중인 탭의 세션이 삭제될 때 발생합니다.

이벤트 이름	설명
getEnvironmentStatus*	CloudShell 환경 상태를 검색할 때 발생합니다.
getFileDownloadUrls*	CloudShell 웹 인터페이스를 CloudShell 사용하여 파일을 다운로드하는 데 사용되는 사전 서명된 Amazon S3 URL이 생성될 때 발생합니다.
getFileUploadUrls*	CloudShell 웹 인터페이스를 CloudShell 사용하여 파일을 업로드하는 데 사용되는 사전 서명된 Amazon S3 URL이 생성될 때 발생합니다.
getLayout*	세션 시작 시 CloudShell 레이아웃을 검색할 때 발생합니다.
putCredentials	에 로그인하는 데 사용된 자격 증명이 전달될 AWS Management Console 때 발생합니다. CloudShell
redeemCode*	CloudShell환경에서 새로 고침 토큰을 검색하는 워크플로가 시작될 때 발생합니다. 나중에 putCredentials 명령에서 이 토큰을 사용하여 CloudShell 환경에 액세스할 수 있습니다.
sendHeartBeat	CloudShell 세션이 활성 상태인지 확인하기 위해 발생합니다.
startEnvironment	CloudShell 환경이 시작될 때 발생합니다.
stopEnvironment	실행 중인 CloudShell 환경이 중지될 때 발생합니다.
updateLayout	백엔드에 있는 웹 애플리케이션의 현재 레이아웃이 저장될 때 발생합니다.

“레이아웃”이라는 단어가 포함된 이벤트는 CloudShell 터미널의 모든 브라우저 탭을 복원합니다.

EventBridge AWS CloudShell 액션 규칙

EventBridge 규칙을 사용하여 규칙과 일치하는 이벤트를 EventBridge 수신할 때 수행할 대상 작업을 지정합니다. CloudTrail 로그 파일에 이벤트로 기록된 작업을 기반으로 수행할 대상 AWS CloudShell 작업을 지정하는 규칙을 정의할 수 있습니다.

예를 들어, `put-rule` 명령을 AWS CLI사용하여 EventBridge 규칙을 만들 수 있습니다. `put-rule`호출에는 최소한 EventPattern or가 포함되어야 ScheduleExpression 합니다. 일치하는 이벤트가 관찰되면 EventPatterns 규칙이 트리거됩니다. EventPattern for AWS CloudShell 이벤트:

```
{ "source": [ "aws.cloudshell" ], "detail-type": [ "AWS API Call via CloudTrail" ],
 "detail": { "eventSource": [ "cloudshell.amazonaws.com" ] } }
```

자세한 내용은 Amazon EventBridge 사용 설명서의 이벤트 및 이벤트 패턴을 참조하십시오. EventBridge

규정 준수 검증: AWS CloudShell

제3자 감사자는 여러 규정 AWS 준수 프로그램의 일환으로 AWS 서비스의 보안 및 규정 준수를 평가합니다.

AWS CloudShell 는 다음 규정 준수 프로그램의 적용 범위에 포함됩니다.

SOC

AWS 시스템 및 조직 규제 (SOC) 보고서는 주요 규정 준수 제어 항목 및 AWS 목표를 달성하는 방법을 보여주는 독립적인 제3자 검토 보고서입니다.

Service	SDK	SOC 1,2,3
AWS CloudShell	CloudShell	✓

PCI

결제 카드 산업 데이터 보안 표준 (PCI DSS) 은 아메리칸 익스프레스, 디스커버 파이낸셜 서비스, JCB 인터내셔널, 월드와이드 및 비자 주식회사가 설립한 PCI 보안 표준 위원회에서 관리하는 독점적인 정보 보안 표준입니다. MasterCard

Service	SDK	PCI
AWS CloudShell	CloudShell	✓

ISO 및 CSA STAR 인증 및 서비스

AWS ISO/IEC 27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015 및 CSA STAR CCM v4.0을 준수하기 위한 인증을 받았습니다.

Service	SDK	ISO 및 CSA STAR 인증 및 서비스
AWS CloudShell	CloudShell	✓

FedRamp

연방정부 위험 및 권한 부여 관리 프로그램(FedRAMP)은 클라우드 제품 및 서비스의 보안 평가, 권한 부여 및 지속적인 모니터링에 대한 표준 접근 방식을 제공하는 정부 차원의 프로그램입니다.

Service	SDK	FedRAMP Moderate(동부/서부)	페드램프 하이 () GovCloud
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

국방부(DoD) 클라우드 컴퓨팅 보안 요구 사항 안내서(SRG)에서는 DoD 고객에게 서비스를 제공하기 위해 클라우드 서비스 제공업체(CSP)가 DoD 임시 인증을 획득할 수 있도록 표준화된 평가 및 인증 프로세스를 제공합니다.

DoD CC SRG 평가 및 권한 부여를 거치는 서비스는 다음과 같은 상태를 갖습니다.

- 서드 파티 평가 조직(3PAO) 평가: 이 서비스는 현재 서드 파티 평가자에 의해 평가를 받는 중입니다.
- 공동 권한 부여 위원회(JAB) 검토: 이 서비스는 현재 JAB의 검토를 받는 중입니다.
- 국방 정보 시스템 기관(DISA) 검토: 이 서비스는 현재 DISA의 검토를 받는 중입니다.

Service	SDK	DoD CC SRG IL2(동부/서부)	DoD CC SRG IL2 () GovCloud	DoD CC SRG IP4 () GovCloud	DoD CC SRG IP5 () GovCloud	DoD CC SRG IL6 (시크릿 리전)AWS
AWS CloudShell	CloudShell	3PAO 평가	N/A	해당 사항 없음	해당 사항 없음	N/A

HIPAA BAA

1996년에 제정된 미국 건강 보험 양도 및 책임에 관한 법(HIPAA)은 환자의 동의나 지식 없이 민감한 환자 건강 정보가 공개되지 않도록 방지하기 위해 국가 표준의 수립을 요구한 연방법입니다.

AWS HIPAA의 적용을 받는 피보험 대상 단체 및 비즈니스 동료는 보호 대상 의료 정보 (PHI) 를 안전하게 처리, 저장 및 전송할 수 있습니다. 또한 2013년 7월부터 이러한 고객을 위해 표준화된 비즈니스 제휴 부록 (BAA) 을 AWS 제공합니다.

Service	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

IRAP(Information Security Registered Assessors Program)를 통해 호주 정부 고객은 적절한 제어가 이루어지는지 검증하고 호주 사이버 보안 센터(ACSC)에서 제작한 호주 정부 ISM(Information Security Manual)의 요구 사항을 해결하기 위한 적절한 책임 모델을 결정할 수 있습니다.

Service	네임스페이스*	IRAP 보호
AWS CloudShell	N/A	✓

*네임스페이스는 환경 전반의 서비스를 식별하는 데 도움이 됩니다. AWS 예를 들어, IAM 정책을 생성하고 Amazon 리소스 이름 (ARN) 으로 작업하고 로그를 읽을 AWS CloudTrail 때.

MTCS

멀티티어 클라우드 보안(MTCS)은 ISO 27001/02 정보 보안 관리 시스템(ISMS) 표준을 기반으로 하는 싱가포르 운영 보안 관리 표준(SPRING SS 584)입니다.

Service	SDK	미국 동부(오하이오)	미국 동부(버지니아 북부)	미국 서부(오레곤)	미국 서부(캘리포니아 북부)	싱가포르	서울
AWS CloudShell	CloudShell	✓	✓	✓	N/A	해당 사항 없음	N/A

C5

C5(Cloud Computing Compliance Controls Catalogue)는 연방 정보 보안 사무소(BSI)가 독일에서 도입한 독일 정부 지원 증명 체계로서, 독일 정부의 '클라우드 공급자를 위한 보안 권장 사항'에 따라 클라우드 서비스를 사용할 때 일반적인 사이버 공격에 대한 운영 보안을 입증할 수 있도록 돕습니다.

Service	SDK	C5
AWS CloudShell	CloudShell	✓

ENS High

ENS(Esquema Nacional de Seguridad) 인증 제도는 재무행정부와 CCN(국립 암호학 센터)에서 개발했습니다. 여기에는 적절한 정보 보호에 필요한 기본 원칙과 최소 요구 사항이 포함됩니다.

Service	SDK	ENS High
AWS CloudShell	CloudShell	✓

FINMA

FINMA는 스위스의 독립적인 금융 시장 규제 기관입니다. AWS이러한 GSMA 요구 사항 준수는 핀란드 교통 및 통신 기관인 Traficom이 설정한 클라우드 서비스 공급자에 대한 높아진 기대치에 부응하려는 지속적인 노력을 보여줍니다.

Service	SDK	FINMA
AWS CloudShell	CloudShell	✓

PiTuKri

AWS PiTuKri 요구 사항을 준수한다는 것은 핀란드 교통통신국인 Traficom이 설정한 클라우드 서비스 공급자에 대한 높아진 기대치를 충족하기 위한 당사의 지속적인 노력을 보여줍니다.

Service	SDK	PiTuKri
AWS CloudShell	CloudShell	✓

특정 규정 준수 프로그램의 범위에 속하는 AWS 서비스 목록은 규정 준수 프로그램별 AWS 범위 내 서비스 규정 준수 프로그램별 참조하십시오. 일반 정보는 AWS 규정 준수 프로그램AWS 보증 프로그램 규정AWS 참조하십시오.

를 사용하여 AWS Artifact타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드를 참조하세요.

사용 시 규정 준수 AWS CloudShell 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 퀵 스타트 가이드 - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 중심의 기본 환경을 배포하기 위한 단계를 제공합니다. AWS
- HIPAA 보안 및 규정 준수를 위한 설계 백서 — 이 백서는 기업이 HIPAA 준수 애플리케이션을 개발하는 데 사용할 수 있는 방법을 설명합니다. AWS
- AWS 규정 준수 리소스 규정 — 이 통합 문서 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS Config 개발자 안내서의 규칙을 통한 리소스 평가 — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.

- AWS Security Hub— 이 AWS 서비스는 보안 업계 표준 및 모범 사례를 준수하는지 확인하는 데 도움이 AWS 되는 내부 보안 상태를 종합적으로 보여줍니다.

레질리언스: AWS CloudShell

AWS 글로벌 인프라는 AWS 지역 및 가용 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며, 이러한 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 지역 및 가용 영역에 대한 자세한 내용은 글로벌 인프라를 참조하십시오AWS .

AWS 글로벌 인프라 외에도 데이터 복원력 및 백업 요구 사항을 AWS CloudShell 지원하는 특정 기능을 지원합니다.

- 직접 만든 커밋 파일을 만들고 추가합니다. AWS CodeCommit클라우드에서 비공개로 자산을 저장하고 관리할 때 사용할 수 있는 Amazon Web Services에서 호스팅하는 버전 관리 서비스입니다. 이러한 자산은 문서, 소스 코드 및 바이너리 파일로 구성될 수 있습니다. 자세한 설명은 튜토리얼: CodeCommit 에서 사용AWS CloudShell 섹션을 참조하세요.
- AWS CLI 호출을 사용하여 홈 디렉터리의 파일을 지정하고 Amazon S3 버킷에 객체로 추가할 수 있습니다. AWS CloudShell 예시는 시작하기 자습서에서 확인하십시오.

의 인프라 보안 AWS CloudShell

관리형 서비스로서 AWS 글로벌 네트워크 보안으로 AWS CloudShell 보호됩니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 액세스할 AWS CloudShell 수 있습니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증 정보를 생성하여 요청에 서명할 수 있습니다.

> **Note**
>
> 기본적으로 컴퓨팅 환경의 시스템 패키지용 보안 패치를 AWS CloudShell 자동으로 설치합니다.

의 구성 및 취약성 분석 AWS CloudShell

컴퓨팅 환경에 설치한 모든 소프트웨어를 패치하고 최신 상태로 유지하는 것은 AWS CloudShell 사용자의 책임입니다.

에 대한 보안 모범 사례 AWS CloudShell

다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 환경에 적절하지 않거나 충분하지 않을 수 있으므로 참고용으로만 사용해 주십시오.

에 대한 몇 가지 보안 모범 사례 AWS CloudShell

- IAM 권한 및 정책을 사용하여 액세스를 AWS CloudShell 제어하고 사용자가 역할에 필요한 작업(예: 파일 다운로드 및 업로드) 만 수행할 수 있도록 합니다. 자세한 설명은 IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리 섹션을 참조하세요.
- 사용자, 역할, 세션 이름과 같은 민감한 데이터를 IAM 엔터티에 포함시키지 않습니다.
- 안전한 붙여넣기 기능을 계속 활성화한면 외부 소스에서 복사한 텍스트의 잠재적 보안 위험을 포착할 수 있습니다. 안전한 붙여넣기는 기본적으로 활성화됩니다. 자세한 설명은 여러 줄 텍스트에 안전 붙여넣기 사용 섹션을 참조하세요.
- 공동 보안 책임 모델을 숙지한 다음 AWS CloudShell컴퓨팅 환경에 타사 애플리케이션을 설치하십시오.
- 사용자의 쉘 환경에 악영향을 주는 쉘 스크립트를 편집하기 전에 롤백 메커니즘을 준비합니다. 자세한 내용은 스크립트로 쉘 수정 섹션을 참조하십시오.
- 버전 제어 시스템(예: AWS CodeCommit)에 안전하게 코드를 저장합니다.

AWS CloudShell 보안 FAQ

이 AWS 서비스의 보안에 대한 FAQ에 대한 답변.

셸 세션을 시작하고 CloudShell 시작할 때 사용되는 AWS 프로세스와 기술은 무엇입니까?

AWS Management Console로그인할 때 IAM 사용자 자격 증명을 입력합니다. 그리고 콘솔 CloudShell 인터페이스에서 시작하면 서비스를 위한 컴퓨팅 환경을 만드는 CloudShell API를 호출할 때 이러한 자격 증명이 사용됩니다. 그런 다음 컴퓨팅 환경을 위한 AWS Systems Manager 세션이 생성되고 CloudShell 해당 세션에 명령을 보냅니다.

보안 FAQ 목록으로 돌아가기

네트워크 액세스를 제한할 수 CloudShell 있습니까?

네트워크 공급자와 CloudShell 연결하여 네트워크 액세스를 제한할 수 있습니다. 또는 IAM 권한을 사용하거나, IAM 권한에 대한 액세스를 명시적으로 CloudShell 거부하거나 제공하지 않고, 암시적 IAM 거부 기능을 사용할 수 있습니다. 자세한 내용은 IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리를 참조하십시오.

보안 FAQ 목록으로 돌아가기

CloudShell 환경을 사용자 지정할 수 있습니까?

사용자 CloudShell 환경에 맞는 유틸리티 및 기타 타사 소프트웨어를 다운로드하고 설치할 수 있습니다. $HOME 디렉터리에 설치된 소프트웨어만 세션 간에 유지됩니다.

AWS 공동 책임 모델에 정의된 대로, 설치한 애플리케이션의 필수 구성과 관리는 사용자의 책임입니다.

보안 FAQ 목록으로 돌아가기

내 **$HOME** 디렉토리는 실제로 어디에 저장되나요? AWS 클라우드

$HOME에 데이터를 저장하기 위한 인프라는 Amazon S3에서 제공합니다.

보안 FAQ 목록으로 돌아가기

내 **$HOME** 디렉터리를 암호화할 수 있나요?

$HOME 디렉터리의 데이터는 Amazon S3 암호화를 사용하여 이미 암호화되어 있습니다.

보안 FAQ 목록으로 돌아가기

내 **$HOME** 디렉터리에서 바이러스 검사를 실행할 수 있나요?

현재는 $HOME 디렉터리의 바이러스 검사를 실행할 수 없습니다. 이 기능에 대한 지원은 검토 중입니다.

보안 FAQ 목록으로 돌아가기

AWS CloudShell 컴퓨팅 환경: 사양 및 소프트웨어

시작하면 AWS CloudShell Amazon Linux 2023을 기반으로 하는 컴퓨팅 환경이 생성되어 셸 환경을 호스팅합니다. 이 환경은 컴퓨팅 리소스(vCPU 및 메모리)로 구성되며 명령줄 인터페이스에서 액세스할 수 있는 다양한 사전 설치 소프트웨어가 있습니다. 소프트웨어를 설치하고 쉘 스크립트를 수정하여 기본 환경을 구성할 수도 있습니다.

컴퓨팅 환경 리소스

개별 AWS CloudShell 컴퓨팅 환경에는 다음과 같은 CPU 및 메모리 리소스가 할당됩니다.

- 1 vCPU(가상 중앙 처리 장치)
- 2-GiB RAM

또한 환경은 다음과 같은 스토리지 구성으로 프로비저닝됩니다.

- 1-GB 영구 스토리지(세션 종료 후에도 스토리지 유지)

자세히 알아보려면 영구 스토리지의 내용을 참조하세요.

CloudShell 네트워크 요구 사항

WebSockets

CloudShell WebSocket 프로토콜에 따라 달라지며, 프로토콜에 따라 사용자의 웹 브라우저와 AWS 클라우드 CloudShell 서비스 간에 양방향 대화식 통신이 가능합니다. 사설 네트워크에서 브라우저를 사용하는 경우 프록시 서버와 방화벽을 통해 인터넷에 안전하게 액세스할 수 있을 것입니다. WebSocket 통신은 일반적으로 프록시 서버를 통해 문제 없이 이루어질 수 있습니다. 하지만 프록시 서버가 제대로 작동하지 않는 경우도 WebSockets 있습니다. 이 문제가 발생하면 CloudShell 인터페이스에서 다음 오류를 `Failed to open sessions : Timed out while opening the session` 보고합니다.

이 오류가 반복해서 발생하는 경우 프록시 서버 설명서를 참조하여 허용하도록 구성되어 있는지 확인하십시오 WebSockets. 아니면 네트워크 시스템 관리자에게 문의하시기 바랍니다.

> **Note**
>
> 특정 URL을 허용 목록에 추가하여 세분화된 권한을 정의하려는 경우 AWS Systems Manager 세션에서 입력 및 출력을 보내기 위한 WebSocket 연결을 여는 데 사용하는 URL의 일부를 추가할 수 있습니다. (AWS CloudShell 명령은 해당 Systems Manager 세션으로 전송됩니다.) Systems Manager에서 StreamUrl 사용하는 이 형식의 형식은 입니다`wss://ssmmessages.`**`region`**`.amazonaws.com/v1/data-channel/`**`session-id`**`?stream=(input|output)`.
> 리전은 미국 동부(오하이오) 리전의 `us-east-2` 같이 AWS Systems Manager이 지원하는 AWS 리전의 리전 식별자를 나타냅니다.
> 세션 ID는 특정 Systems Manager 세션이 정상적으로 시작된 후 생성되므로 URL 허용 목록을 업데이트할 때만 `wss://ssmmessages.region.amazonaws.com` 지정이 가능합니다. 자세한 내용은 AWS Systems ManagerAPI 참조의 StartSession작업을 참조하십시오.

사전 설치 소프트웨어

> **Note**
>
> AWS CloudShell 개발 환경은 최신 소프트웨어에 액세스할 수 있도록 정기적으로 업데이트되므로 이 설명서에는 특정 버전 번호가 표시하지 않습니다. 그 대신, 설치된 버전을 확인할 수 있는 방법을 알려 드립니다. 설치된 버전을 확인하려면 프로그램명을 입력하고 `--version` 옵션(예: `git --version`)을 입력합니다.

쉘

사전 설치 쉘

명칭	Description	Version information
Bash	Bash 쉘은 AWS CloudShell용 기본 쉘 애플리케이션입니다.	`bash --version`
PowerShell (pwsh)	명령줄 인터페이스와 스크립팅 언어 지원을 제공하는 PowerShell 것은 Microsoft의 .NET 명령 언어 런타임을	`pwsh --version`

명칭	Description	Version information
	기반으로 구축되었습니다. PowerShell .NET 객체를 수락하고 `cmdlets` 반환하는 간단한 명령을 사용합니다.	
Z 쉘(zsh)	Z 쉘, 또는 `zsh`은(는) 일명 Bourne 쉘의 확장 버전으로 테마 및 플러그인에 대한 향상된 사용자 지정 지원을 제공합니다.	`zsh --version`

AWS 명령줄 인터페이스(CLI)

CLI

명칭	Description	Version information
AWS CDK 툴킷 CLI	AWS CDK 툴킷, CLI 명령어 `cdk`은(는) AWS CDK 앱과 상호 작용하는 기본 도구입니다. 앱을 실행하고, 정의한 애플리케이션 모델 정보를 얻고, AWS CloudFormation 템플릿(AWS CDK에서 생성)을 배포합니다. 자세한 내용은 AWS CDK 툴킷 단원을 참조하세요.	`cdk --version`
AWS CLI	AWS CLI은(는) 명령줄에서 여러 AWS 서비스를 관리하고 스크립트를 사용하여 자동화하는 데 사용할 수 있는 명령줄 인터페이스입니다. 자세히 알아보려면 다음 지역AWS 서비스 이용AWS CloudShell의 내용을 참조하세요.	`aws --version`

명칭	Description	Version information
	up-to-date AWS CLI버전 2를 가장 많이 사용하고 있는지 확인하는 방법에 대한 자세한 내용은 을 참조하십시오홈 디렉터리에 AWS CLI 설치하기.	
EB CLI	AWS Elastic Beanstalk CLI는 로컬 리포지토리에서 환경 생성, 업데이트 및 모니터링을 단순화하는 대화형 명령을 제공하는 명령줄 인터페이스입니다. 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서에서 Elastic Beanstalk 명령줄 인터페이스(EB CLI) 사용을 참조하세요.	`eb --version`
Amazon ECS CLI	Amazon Elastic Container Service(Amazon ECS) 명령줄 인터페이스(CLI)는 로컬 개발 환경에서 클러스터 및 작업 모니터링을 간소화하는 상위 수준 명령을 제공합니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 Amazon ECS 명령줄 참조 사용을 참조하세요.	`ecs-cli --version`

명칭	Description	Version information
AWS SAM CLI	AWS SAM CLI는 AWS Serverless Application Model 템플릿 및 애플리케이션에서 작동하는 명령줄 도구입니다. 여러 작업을 수행할 수 있습니다. 여기에는 로컬에서 람다 함수 호출, 서버리스 애플리케이션을 위한 배포 패키지 생성, 서버리스 애플리케이션을AWS 클라우드에 배포하는 방법이 포함됩니다. 자세한 내용은 AWS Serverless Application Model 개발자 가이드의 AWS SAM CLI 명령 참조를 참조하세요.	`sam --version`
AWS Tools for PowerShell	에서 제공하는 기능을 기반으로 구축된 AWS Tools for PowerShell PowerShell 모듈입니다AWS SDK for .NET. 를 사용하면 PowerShell 명령줄에서 AWS 리소스에 대한 작업을 스크립팅할 수 있습니다. AWS Tools for PowerShell AWS CloudShell은(는) AWS Tools for PowerShell의 모듈화된 버전(AWS.tools)을 사전 설치합니다. 자세한 내용은 사용 AWS Tools for PowerShell설명서의 AWS 도구 사용을 참조하십시오. PowerShell	`pwsh --Command ' Get-Module -ListAvailable -Name AWS.Tools.Common'`

런타임 및 AWS SDK: Node.js 및 Python 3

런타임 및 AWS SDK

명칭	Description	Version information
Node.js (npm 포함)	Node.js 는 비동기 프로그래밍 기술을 더 쉽게 적용할 수 있도록 설계된 JavaScript 런타임입니다. 자세한 정보는 공식 Node.js 사이트에 있는 설명서에서 확인하세요. npm은 온라인 모듈 레지스트리에 대한 액세스를 제공하는 패키지 관리자입니다. JavaScript 자세한 내용은 공식 npm 사이트의 설명서를 참조하십시오.	• Node.js: `node --version` • npm: `npm --version`
Node.js 용 JavaScript SDK	소프트웨어 개발 키트 (SDK)는 Amazon S3, Amazon EC2, DynamoDB 및 Amazon SWF를 비롯한 AWS 서비스에 JavaScript 객체를 제공하여 코딩을 단순화하는 데 도움이 됩니다. 자세한 정보는 AWS SDK for JavaScript 개발자 안내서를 참조하세요.	`npm -g ls --depth 0 2>/dev/null | grep aws-sdk`
Python	Python 3은 셸 환경에서 사용할 준비가 되었습니다. 현재 Python 3이 프로그래밍 언어 기본 버전으로 간주됩니다(Python 2 지원은 2020년 1월에 종료). 자세한 정보는 Python 공식 사이트에 있는 설명서에서 확인하세요.	• Python 3: `python3 --version` • pip: `pip3 --version`

명칭	Description	Version information
	또한 사전 설치된 pip는 Python 용 패키지 설치 프로그램입니다. 이 명령줄 프로그램을 사용하여 Python 패키지 색인과 같은 온라인 색인에서 Python 패키지를 설치할 수 있습니다. 자세한 정보는 Python Packaging Authority 제공 설명서에서 확인하세요.	
SDK for Python (Boto3)	Boto는 Python 개발자가 Amazon EC2, Amazon S3처럼 AWS 서비스의 생성, 구성, 관리 시 사용하는 소프트웨어 개발 키트(SDK)입니다. SDK는 객체 easy-to-use 지향 API와 저수준 액세스를 제공합니다. AWS 서비스 자세한 내용은Boto3 설명서를 참조하십시오.	`pip3 list | grep boto3`

개발 도구 및 쉘 유틸리티

개발 도구 및 쉘 유틸리티

명칭	Description	Version information
bash-completion	bash-completion은 Tab 키를 눌러 부분적으로 입력된 명령이나 인수를 자동으로 완성할 수 있는 쉘 함수 모음입니다. bash-completion이 지원하는 패키지는 `/usr/share/bash-completion/`	`dnf info bash-completion`

명칭	Description	Version information
	`completions` 에서 찾을 수 있습니다. 패키지 명령에 대한 자동 완성을 설정하려면 프로그램 파일을 소싱해야 합니다. 예를 들어, Git 명령의 자동 완성을 설정하려면 다음 줄을 `.bashrc`에 추가하여 AWS CloudShell 세션이 시작될 때마다 기능을 사용할 수 있게 합니다. `source /usr/share/bash-completion/completions/git` 사용자 지정 완성 스크립트를 사용하려면 영구 홈 디렉터리(`$HOME`)에 추가하고 `.bashrc`에서 직접 소싱합니다. 자세한 내용은 의 프로젝트의 README 페이지를 참조하십시오. GitHub	

명칭	Description	Version information
CodeCommit Git용 유틸리티	git-remote-codecommit CodeCommit Git을 확장하여 리포지토리에서 코드를 푸시하고 가져오는 간단한 방법을 제공하는 유틸리티입니다. 이는 페더레이션 액세스, ID 공급자 및 임시 자격 증명을 사용한 연결을 지원하는 데 권장되는 방법입니다. 자세한 내용은 사용 설명서의 witter에 대한 HTTPS 연결 설정 단계를 참조하십시오. AWS CodeCommit git-remote-codecommit AWS CodeCommit	`pip3 list | grep git-remote-codecommit`
Git	Git는 브랜치 워크플로와 콘텐츠 스테이징을 통해 최신 소프트웨어 개발 방식을 지원하는 분산 버전 제어 시스템입니다. 자세한 정보는 Git 공식 사이트에 있는 설명서에서 확인하세요.	`git --version`
iputils	iputils 패키지에는 Linux 네트워킹용 유틸리티가 들어 있습니다. 제공된 유틸리티에 대한 자세한 내용은 의 iputils 저장소를 참조하십시오. GitHub	iputils 도구 예시: `arping -V`

명칭	Description	Version information
jq	jq 유틸리티는 JSON 형식의 데이터를 구문 분석하여 명령줄 필터로 수정된 출력을 생성합니다. 자세한 내용은 에서 호스팅되는 jq 설명서를 참조하십시오. GitHub	`jq --version`
kubectl	kubectl은 Kubernetes API를 사용하여 Kubernetes 클러스터의 컨트롤 플레인과 통신하는 명령줄 도구입니다.	`kubectl --version`
make	make 유틸리티는 `makefiles` 으로 작업 세트를 자동화하고 코드 컴파일을 구성합니다. 자세한 내용은 GNU Make 설명서를 참조하십시오.	`make --version`
man	man 명령은 명령줄 유틸리티 및 도구에 대한 매뉴얼 페이지를 제공합니다. 예를 들어, `man ls`은(는) 디렉토리 콘텐츠를 나열하는 `ls` 명령의 매뉴얼 페이지를 반환합니다. 자세한 내용은 man 페이지에 대한 Wikipedia 페이지를 참조하십시오.	`man --version`
nano	nano는 텍스트 기반 인터페이스용의 작고 사용자 친화적인 편집기입니다. 자세한 내용은 GNU 나노 설명서를 참조하십시오.	`nano --version`

명칭	Description	Version information
procps	procps는 현재 실행 중인 프로세스를 모니터링하고 중지하는 데 사용하는 시스템 관리 유틸리티입니다. 자세한 정보는 procps로 실행 가능한 프로그램 목록이 수록되어 있는 README 파일에서 확인하세요.	`ps --version`
SSH 클라이언트	SSH 클라이언트는 보안 쉘 프로토콜로 원격 컴퓨터와의 암호화된 통신을 합니다. OpenSSH는 사전 설치된 SSH 클라이언트입니다. 자세한 정보는 OpenBSD가 유지관리하는 OpenSSH 사이트에서 확인하세요.	`ssh -V`
sudo	sudo 유틸리티가 있으면 다른 사용자(일반적으로 수퍼유저)의 보안 권한으로 프로그램을 실행할 수 있습니다. Sudo는 시스템 관리자로서 애플리케이션을 설치해야 할 때 유용합니다. 자세한 정보는 Sudo 매뉴얼에서 확인하세요.	`sudo --version`
tar	tar는 여러 파일을 단일 아카이브 파일(tarball)로 그룹화할 때 사용하는 명령줄 유틸리티입니다. 자세한 내용은 GNU 타르 설명서를 참조하세요.	`tar --version`

명칭	Description	Version information
tmux	tmux는 여러 창에서 여러 프로그램을 동시에 실행할 때 사용하는 터미널 멀티플렉서입니다. 자세한 정보는 tmux를 간단하게 소개하는 내용의 블로그에서 확인하세요.	`tmux -V`
unzip	자세한 정보는 zip/unzip에서 확인하세요.	
vim	vim은 텍스트 기반 인터페이스를 통해 상호 작용할 수 있는 사용자 지정 가능한 편집기입니다. 자세한 내용은 vim.org에서 제공되는 설명서 리소스를 참조하십시오.	`vim --version`
wget	wget은 명령줄의 엔드포인트로 지정된 웹 서버에서 콘텐츠를 검색할 때 사용하는 컴퓨터 프로그램입니다. 자세한 내용은 GNU Wget 설명서를 참조하십시오.	`wget --version`

zip/unzip	zip/unzip 유틸리티는 데이터 손실 없이 무손실 데이터 압축을 제공하는 아카이브 파일 형식을 사용합니다. zip 명령을 호출하면 파일을 단일 아카이브로 그룹화하고 압축합니다. unzip을 사용하면 아카이브에서 지정된 디렉터리로 파일을 추출합니다.	`unzip --version`
		
		`zip --version`

명칭	Description	Version information
Docker	Docker는 애플리케이션을 개발, 배송 및 실행하기 위한 개방형 플랫폼입니다. Docker를 사용하면 애플리케이션을 인프라에서 분리하여 소프트웨어를 신속하게 제공할 수 있습니다. 이를 통해 내부에 Dockerfile을 빌드하고 CDK를 사용하여 AWS CloudShell Docker 자산을 구축할 수 있습니다. Docker에서 지원되는 지역에 대한 자세한 내용은 Docker 지역을 참조하십시오. Docker 환경에는 공간이 제한되어 있다는 점에 유의해야 합니다. 개별 이미지가 크거나 기존 Docker 이미지가 너무 많으면 문제가 발생할 수 있습니다. Docker에 대한 자세한 내용은 Docker 설명서 가이드를 참조하십시오.	`docker --version`

홈 디렉터리에 AWS CLI 설치하기

사용자 CloudShell 환경에 사전 설치된 다른 소프트웨어와 마찬가지로 이 AWS CLI 도구도 예정된 업그레이드 및 보안 패치를 통해 자동으로 업데이트됩니다. 최신 up-to-date 버전을 사용하고 싶다면 셸의 AWS CLI 홈 디렉터리에 도구를 수동으로 설치하도록 선택할 수 있습니다.

> **Important**
>
> 의 AWS CLI 사본을 홈 디렉터리에 수동으로 설치해야 다음에 CloudShell 세션을 시작할 때 사용할 수 있습니다. 수동 설치가 필요한 이유는 `$HOME` 외부 디렉터리에 추가된 파일이 쉘 세션 종료 시 삭제되기 때문입니다. 또한 AWS CLI 복사본 설치 후에는 자동으로 업데이트되지 않습니다. 다시 말해, 업데이트와 보안 패치 관리는 사용자의 책임입니다.
> AWS 공동 책임 모델데이터 보호: AWS CloudShell에 대한 자세한 내용은 을(를) 참조하세요.

AWS CLI을(를) 설치하려면

1. 명령줄에서 CloudShell 명령을 사용하여 AWS CLI 설치된 파일의 압축된 사본을 셸로 전송합니다. `curl`

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. 폴더의 압축을 풉니다.

```
unzip awscliv2.zip
```

3. 도구를 지정 폴더에 추가하려면 AWS CLI 설치 프로그램을 실행합니다.

```
sudo ./aws/install --install-dir /home/cloudshell-user/usr/local/aws-cli --bin-
dir /home/cloudshell-user/usr/local/bin
```

   정상적으로 설치되면 명령줄에 다음 메시지가 표시됩니다.

```
You can now run: /home/cloudshell-user/usr/local/bin/aws --version
```

4. PATH 환경 변수도 업데이트하면 aws 명령을 실행할 때 도구 설치 경로를 지정하지 않아도 되므로 편리합니다.

```
export PATH=/home/cloudshell-user/usr/local/bin:$PATH
```

> **Note**
>
> PATH 변경을 취소하면 지정된 경로가 없는 aws 명령에는 기본적으로 AWS CLI의 사전 설치된 버전이 사용됩니다.

쉘 환경에 타사 소프트웨어 설치

> **Note**
>
> 공동 보안 책임 모델을 검토한 다음 AWS CloudShell 컴퓨팅 환경에 타사 애플리케이션을 설치할 것을 권장합니다.

기본적으로 전체 AWS CloudShell 사용자에게는 sudo 권한이 있습니다. 따라서 sudo 명령으로 쉘의 컴퓨팅 환경에서 아직 사용할 수 없는 소프트웨어를 설치할 수 있습니다. 예를 들어 DNF 패키지 관리 sudo 유틸리티와 함께 사용하여 설치할 수 있습니다. 그러면 다음과 cowsay 같은 메시지가 있는 소의 ASCII 아트 사진이 생성됩니다.

```
sudo dnf install cowsay
```

그리고 echo "Welcome to AWS CloudShell" | cowsay을(를) 입력하면 새로 설치된 프로그램이 열립니다.

> **Important**
>
> Package 관리 유틸리티 (예: 디렉터리) 의 dnf install 프로그램과 같은 유틸리티 (/usr/bin예: 셸 세션이 종료될 때 재활용됨). 즉 세션별로 추가 소프트웨어가 설치되고 사용됩니다.

스크립트로 쉘 수정

기본 쉘 환경을 수정하려면 쉘 환경이 시작될 때마다 실행되는 쉘 스크립트를 편집해야 합니다. .bashrc 스크립트는 기본 bash 쉘이 시작될 때마다 실행됩니다.

> **Warning**
>
> .bashrc 파일을 잘못 수정하면 이후에 쉘 환경에 액세스하지 못할 수 있습니다. 편집하기 전 파일 사본을 만들어 두는 것이 좋습니다. .bashrc 편집 시 쉘을 두 개 열어 위험을 줄일 수도 있습니다. 그러면 한 쉘에서 액세스 권한을 잃더라도 다른 쉘에는 계속 로그인한 상태이므로 변경 내용을 롤백할 수 있습니다.
> .bashrc 또는 다른 파일을 잘못 수정하여 액세스 권한을 잃은 경우, 홈 디렉터리 삭제를 통해 AWS CloudShell을(를) 기본 설정으로 복원시키면 됩니다.

이 절차에서는 쉘 환경이 Z 쉘 실행으로 자동 전환되도록 .bashrc 스크립트를 수정합니다.

1. 텍스트 편집기(예: Vim)로 .bashrc을(를) 엽니다.

   ```
   vim .bashrc
   ```

2. 편집기 인터페이스에서 I 키를 눌러 편집을 시작하고 다음을 추가합니다.

```
zsh
```

3. 종료하고 편집한 `.bashrc` 파일을 저장하려면 Esc 키를 눌러 Vim 명령 모드로 전환하고 다음을 입력합니다.

   ```
   :wq
   ```

4. `source` 명령으로 `.bashrc` 파일을 재로드합니다.

   ```
   source .bashrc
   ```

   명령줄 인터페이스를 다시 사용할 수 있게 되면 프롬프트 기호가 %로 바뀌어 현재 Z 쉘을 사용하고 있음을 나타냅니다.

Amazon Linux 2에서 Amazon Linux 2023으로 AWS CloudShell 마이그레이션

AWS CloudShell아마존 리눅스 2 (AL2) 를 기반으로 하던 가 아마존 리눅스 2023 (AL2023) 로 마이그레이션되었습니다. AL2023에 대한 자세한 정보는 Amazon Linux 2023 사용 설명서의 Amazon Linux 2023(AL2023)란 무엇인가요?를 참조하세요.

AL2023 를 사용하면 에서 제공하는 모든 도구를 사용하여 기존 CloudShell 환경에 계속 액세스할 수 있습니다. CloudShell 사용 가능한 도구에 대한 자세한 정보는 사전 설치 소프트웨어에서 확인하세요.

AL2023은 Node.js 18, Python 3.9 등 최신 버전의 패키지를 포함하여 개발 도구에 대한 몇 가지 개선 사항을 제공합니다.

> **Note**
>
> AL2023 버전에서는 Python 2가 사용자 CloudShell 환경과 함께 더 이상 제공되지 않습니다.

AL2와 AL2023의 주요 차이점에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서에서 Amazon Linux 2와 Amazon Linux 2023 비교를 참조하세요.

이에 대한 문의 사항이 있으면 AWS Support에 연락하십시오. AWS re:Post 포럼에서 답을 검색하고 질문을 올릴 수도 있습니다. AWS re:Post에 들어갈 때 AWS로 로그인해야 할 수 있습니다.

AWS CloudShell 마이그레이션 FAQ

다음은 AWS CloudShell를 통한 AL2에서 AL2023로의 마이그레이션과 관련된 일반적인 질문에 대한 답변입니다.

이번 마이그레이션이 AL2에서 실행되는 Amazon EC2 인스턴스와 같은 다른 AWS 리소스에도 영향을 미칠까요?

사용자의 AWS CloudShell 환경 이외의 서비스나 리소스는 이번 마이그레이션의 영향을 받지 않습니다. AWS CloudShell에서 생성했거나 액세스했을 수 있는 리소스도 마찬가지입니다. 예를 들어, AL2에서 실행되도록 생성된 Amazon EC2 인스턴스는 AL2023 인스턴스로 마이그레이션되지 않습니다.

AL2023 마이그레이션과 함께 변경된 패키지는 무엇입니까?

AWS CloudShell 환경에는 현재 사전 설치된 소프트웨어가 포함되어 있습니다. 사전 설치된 소프트웨어의 전체 목록에 대한 자세한 내용은 사전 설치된 소프트웨어를 참조하십시오. AWS CloudShellPython 2를 제외하고 이러한 패키지를 계속 제공합니다. AL2와 AL2023에서 제공하는 패키지 간의 전체 차이점은 AL2와 AL2023 비교에서 확인하세요. AL2023 마이그레이션 후 특정 패키지 및 버전 요구사항이 더 이상 충족되지 않는 고객의 경우 AWS Support에 문의할 것을 권장합니다.

마이그레이션을 거부할 수 있나요?

답은 '아니오'입니다. AWS CloudShell환경은 AL2023 에 의해 AWS 관리되므로 모든 환경이 버전으로 업그레이드되었습니다.

내 AWS CloudShell 환경의 백업 생성이 가능한가요?

AWS CloudShell은(는) 사용자 홈 디렉터리에서 계속 지속됩니다. 자세한 내용은 Service Quotas 및 제한AWS CloudShell을 참조하세요. 홈 폴더에 저장되어 있는 파일이나 구성을 백업하려면 6단계: 홈 디렉터리 백업 생성을 수행하세요.

AWS CloudShell 문제 해결

를 사용하는 동안 셸 명령줄 인터페이스를 사용하여 AWS CloudShell 주요 작업을 CloudShell 시작하거나 수행할 때와 같은 문제가 발생할 수 있습니다. 이 장에서는 일반적으로 접할 수 있는 문제를 해결하는 방법을 설명하겠습니다.

에 대한 다양한 질문에 대한 CloudShell 답변은 AWS CloudShellFAQ를 참조하십시오. AWS CloudShell 토론 포럼에서 답을 검색하고 질문을 올릴 수도 있습니다. 이 포럼에 들어갈 때 AWS에 로그인해야 할 수 있습니다. 또한 직접 당사에 문의할 수도 있습니다.

오류 해결

다음 색인에 있는 오류를 접한 경우, 아래 해결 방법에 따라 오류를 해결할 수 있습니다.

주제

환경을 시작할 수 없음. 다시 시도하려면 작업, 재시작을 선택하여 브라우저를 새로고침하거나 재시작합니다 AWS CloudShell

문제: AWS CloudShell 에서 시작하려고 하면 IAM 관리자로부터 필요한 권한을 받은 후 브라우저를 새로 고치거나 다시 시작한 후에도 액세스가 거부됩니다. AWS Management Console CloudShell

해결 방법: AWS지원에 문의하세요.

(맨 위로 이동)

환경을 시작할 수 없음. 필요한 권한이 없습니다. IAM 관리자에게 AWS CloudShell 액세스 권한을 요청합니다.

문제: AWS Management Console에서 AWS CloudShell을(를) 시작하려고 하면 액세스가 거부되고 필수 권한이 없다는 메시지가 표시됩니다.

원인: AWS CloudShell 액세스 시 사용하는 IAM 자격 증명에 필수 IAM 권한이 없습니다.

해결 방법: IAM 관리자에게 필수 권한을 요청하세요. 연결된 AWS 관리형 정책 (AWSCloudShellFullAccess) 또는 내장된 인라인 정책을 추가하여 이 작업을 수행할 수 있습니다. 자세히 알아보려면 IAM 정책을 통한 AWS CloudShell 액세스 및 사용 관리의 내용을 참조하세요.

(맨 위로 이동)

AWS CloudShell 명령줄에 액세스할 수 없음

문제: 컴퓨팅 환경에서 사용하는 파일을 수정한 후, AWS CloudShell에서 명령줄에 액세스할 수 없습니다.

해결 방법: `.bashrc` 또는 다른 파일을 잘못 수정하여 액세스 권한을 잃은 경우, 홈 디렉터리 삭제를 통해 AWS CloudShell을(를) 기본 설정으로 복원시키면 됩니다.

(맨 위로 이동)

외부 IP 주소를 ping할 수 없음

문제: 명령줄에서 ping 명령(예: `ping amazon.com`)을 실행하면 다음 메시지가 나타납니다.

```
ping: socket: Operation not permitted
```

원인: ping 유틸리티는 인터넷 제어 메시지 프로토콜(ICMP)을 사용하여 에코 요청 패킷을 대상 호스트로 보냅니다. 대상에서 응답할 때까지 에코를 기다립니다. AWS CloudShell에서는 ICMP 프로토콜이 활성화되지 않으므로, ping 유틸리티는 쉘 컴퓨팅 환경에서 작동하지 않습니다.

해결 방법: 에서는 AWS CloudShell ICMP가 지원되지 않으므로 다음 명령을 실행하여 Netcat을 설치할 수 있습니다. Netcat은 TCP 또는 UDP를 사용하여 네트워크 연결을 읽고 쓸 수 있는 컴퓨터 네트워킹 유틸리티입니다.

```
sudo yum install nc
nc -zv www.amazon.com 443
```

(맨 위로 이동)

터미널 준비 시 문제가 발생했습니다.

문제: Microsoft Edge 브라우저로 AWS CloudShell에 액세스하려고 하면 쉘 세션을 시작할 수 없고 브라우저에 오류 메시지가 표시됩니다.

원인: AWS CloudShell은(는) 이전 버전 Microsoft Edge와 호환되지 않습니다. AWS CloudShell에 액세스하려면 지원되는 브라우저의 최신 네 개 주요 버전을 사용해야 합니다.

해결 방법: Microsoft 사이트에서 최신 버전 Edge 브라우저를 설치하세요.

(맨 위로 이동)

에서 화살표 키가 제대로 작동하지 않음 PowerShell

문제: 정상 작동 중에는 화살표 키로 명령줄 인터페이스를 탐색하고 명령 이력을 앞뒤로 스캔할 수 있습니다. 하지만 특정 버전의 PowerShell AWS CloudShell on에서 화살표 키를 누르면 문자가 잘못 출력될 수 있습니다.

원인: 화살표 키가 문자를 잘못 출력하는 상황은 Linux에서 실행되는 PowerShell 7.2.x 버전에서 알려진 문제입니다.

해결 방법: 화살표 키의 동작을 수정하는 이스케이프 시퀀스를 제거하려면 PowerShell 프로필 파일을 편집하고 변수를 로 설정하십시오. `$PSStyle` `PlainText`

1. AWS CloudShell 명령줄에 다음 명령을 입력하여 파일을 생성합니다.

   ```
   vim ~/.config/powershell/Microsoft.PowerShell_profile.ps1
   ```

   > **Note**
   >
   > 이미 PowerShell 로그인했다면 다음 명령을 사용하여 편집기에서 프로필 파일을 열 수도 있습니다.
   >
   > ```
   > vim $PROFILE
   > ```

2. 편집기에서 파일의 기존 텍스트 끝으로 이동한 다음 i를 눌러 삽입 모드로 전환한 후 다음 명령문을 추가합니다.

```
$PSStyle.OutputRendering = 'PlainText'
```

3. 편집한 후 Esc을(를) 눌러 명령 모드로 들어갑니다. 그리고 다음 명령을 입력하여 파일을 저장하고 편집기를 종료합니다.

```
:wq
```

> **Note**
>
> 변경 사항은 다음에 시작할 때 적용됩니다 PowerShell.

(맨 위로 이동)

지원되지 않는 웹 소켓으로 인해 세션이 시작되지 않습니다. CloudShell

문제: AWS CloudShell을(를) 시작하려고 하면 `Failed to open sessions : Timed out while opening the session` 메시지가 반복적으로 표시됩니다.

원인: 프로토콜에 CloudShell 따라 다릅니다. WebSocket 프로토콜에 따라 웹 브라우저와 웹 브라우저 간에 양방향 대화식 통신이 가능합니다. AWS CloudShell 사설망에서 브라우저를 사용하는 경우 프록시 서버와 방화벽을 통해 인터넷에 안전하게 액세스할 수 있을 수 있습니다. WebSocket 통신은 일반적으로 프록시 서버를 통해 문제 없이 이루어질 수 있습니다. 하지만 프록시 서버가 제대로 작동하지 않는 경우도 WebSockets 있습니다. 이 문제가 발생하면 셸 세션을 시작할 CloudShell 수 없고 결국 연결 시도 시간이 초과됩니다.

해결 방법: 지원되지 않는 WebSockets 문제가 아닌 다른 문제로 인해 연결 시간 초과가 발생할 수 있습니다. 이런 경우에는 먼저 CloudShell 명령줄 인터페이스가 있는 브라우저 창을 새로 고치십시오.

새로고침한 후에도 시간 초과 오류가 계속 발생하는 경우 프록시 서버 설명서를 참조하세요. 또한 프록시 서버가 Web Socket을 허용하도록 구성되어 있는지 확인하세요. 아니면 네트워크 시스템 관리자에게 문의하세요.

> **Note**
>
> 특정 URL을 허용 목록에 넣어 세분화된 권한을 정의하고자 한다고 가정해 보겠습니다. AWS Systems Manager세션에서 입력 전송 및 출력 수신을 위한 WebSocket 연결을 여는 데 사용하는 URL의 일부를 추가할 수 있습니다. AWS CloudShell 명령은 해당 Systems Manager 세션으로 전송됩니다.
> Systems Manager에서 StreamUrl 사용하는 이 형식의 형식은 다음과 같습니다 `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.
> 리전은 AWS Systems Manager에서 지원되는 AWS 리전에 대한 리전 식별자를 나타냅니다. 예를 들어, `us-east-2`은(는) 미국 동부(오하이오) 리전의 지역 식별자입니다.
> 세션 ID는 특정 Systems Manager 세션이 성공적으로 시작된 후 생성되므로 URL 허용 목록을 업데이트할 때만 `wss://ssmmessages.region.amazonaws.com`을(를) 지정할 수 있습니다. 자세한 내용은 AWS Systems ManagerAPI 참조의 StartSession작업을 참조하십시오.

(맨 위로 이동)

`AWSPowerShell.NetCore` 모듈을 가져올 수 없음

문제: 를 가져올 때 AWSPowerShell. NetCore모듈을 PowerShell by로 `Import-Module -Name AWSPowerShell.NetCore` 가져오면 다음과 같은 오류 메시지가 나타납니다.

임포트 모듈: 지정된 모듈 '. AWSPowerShell NetCore어떤 모듈 디렉터리에도 유효한 모듈 파일이 없어서 '이 (가) 로드되지 않았습니다.

원인: `AWSPowerShell.NetCore` 모듈이 AWS CloudShell에서 서비스별 AWS.Tools 모듈로 교체되었습니다.

해결 방법: 명시적 가져오기 명령문이 더 이상 필요하지 않거나 관련 서비스별 AWS.Tools 모듈로 변경해야 할 수 있습니다.

Example

Example

- 대부분의 경우 .Net 형식을 사용하지 않는 한 명시적인 가져오기 명령문은 필요하지 않습니다. 다음은 가져오기 명령문의 예제입니다.
  - `Get-S3Bucket`

  - `(Get-EC2Instance).Instances`

- .Net 유형을 사용하는 경우, 서비스 수준 모듈(`AWS.Tools.<Service>`)을 가져오기 하세요. 다음은 구문의 예제입니다.

```
Import-Module -Name AWS.Tools.EC2
$InstanceTag = [Amazon.EC2.Model.Tag]::new("Environment","Dev")
```

```
Import-Module -Name AWS.Tools.S3
$LifecycleRule = [Amazon.S3.Model.LifecycleRule]::new()
```

자세한 정보는 버전 4 공지(AWS Tools for PowerShell)에서 확인하세요.

(맨 위로 이동)

를 사용할 때 Docker가 실행되지 않습니다. AWS CloudShell

문제: 사용 시 Docker가 제대로 실행되지 않습니다. AWS CloudShell 다음과 같은 오류 메시지가 나타납니다. `docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?`

해결 방법: 환경을 다시 시작해 보십시오. 이 오류 메시지는 지원되지 않는 AWS CloudShell 지역에서 Docker를 실행할 때 발생할 수 있습니다. 지원되는 지역에서 Docker를 실행하고 있는지 확인하세요. Docker 컨테이너 사용을 지원하는 지역에 대한 자세한 내용은 Docker 지역을 참조하세요. AWS CloudShell

Docker의 디스크 공간이 부족합니다.

문제: 다음과 같은 오류 메시지가 나타납니다. `ERROR: failed to solve: failed to register layer: write [...]: no space left on device`

원인: Dockerfile이 사용 가능한 디스크 공간을 초과했습니다. AWS CloudShell 이는 개별 이미지가 크거나 기존 Docker 이미지가 너무 많기 때문일 수 있습니다.

해결 방법: `df -h` 를 실행하여 디스크 사용량을 확인합니다. `sudo du -sh /folder/folder1`실행하여 크기가 클 것으로 생각되는 특정 폴더의 크기를 확인하고 다른 파일을 삭제하여 공간을 확보해 보세요. 한 가지 옵션은 를 실행하여 사용하지 않는 Docker 이미지를 제거하는 것입니다. `docker rmi`

Docker의 환경 공간이 제한되어 있다는 점에 유의해야 합니다. Docker에 대한 자세한 내용은 Docker 설명서 가이드를 참조하세요.

`docker push`시간이 초과되어 계속 재시도하고 있습니다.

문제: 실행 시간이 `docker push` 초과되어 계속 재시도해도 성공하지 못합니다.

원인: 권한 누락, 잘못된 리포지토리로 푸시 또는 인증 부족으로 인해 발생할 수 있습니다.

해결 방법: 이 문제를 해결하려면 올바른 저장소로 푸시해야 합니다. 제대로 `docker login` 인증하려면 실행하십시오. Amazon ECR 리포지토리로 푸시하는 데 필요한 모든 권한이 있는지 확인하십시오.

AWS CloudShell에 지원되는 브라우저

다음 표에는 AWS CloudShell에 지원되는 브라우저가 나와 있습니다.

웹 브라우저 지원

브라우저	버전
Google Chrome	최신 3개 주요 버전
Mozilla Firefox	최신 3개 주요 버전
Microsoft Edge	최신 3개 주요 버전
macOS용 Apple Safari	최신 2개 주요 버전

AWS CloudShell용 AWS 리전 지원

이 섹션에는 지원되는 AWS 리전과 AWS CloudShell 옵트인 리전의 목록이 있습니다. 에 대한 AWS 서비스 엔드포인트 및 할당량 목록은 의 AWS CloudShell페이지를 참조하십시오. CloudShell Amazon Web Services 일반 참조

AWS CloudShell용 AWS 리전 지원은 다음과 같습니다.

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(캘리포니아 북부)
- 미국 서부(오레곤)
- 아시아 태평양(뭄바이)
- 아시아 태평양(오사카)
- 아시아 태평양(서울)
- 아시아 태평양(시드니)
- 아시아 태평양(싱가포르)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 유럽(스톡홀름)
- 남아메리카(상파울루)

GovCloud 지역

지원되는 GovCloud 지역은 다음과 CloudShell 같습니다.

- AWS GovCloud (미국 동부)
- AWS GovCloud (미국 서부)

옵트인 리전

옵트인 리전에서 기본 지원 리전으로 사용하려면 수동으로 활성화해야 합니다. 자세한 내용은 AWS 리전 관리를 참조하십시오. 지원되는 옵트인 지역은 다음과 같습니다. CloudShell

- 아프리카(케이프타운)
- 아시아 태평양(홍콩)
- 아시아 태평양(자카르타)
- 유럽(밀라노)
- 중동(바레인)
- 중동(UAE)

Docker가 지원되는 지역

AWS CloudShell컴퓨팅 환경은 다음 지역의 Docker 컨테이너만 지원합니다.

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(오레곤)
- 아시아 태평양(뭄바이)
- 아시아 태평양(시드니)
- 아시아 태평양(싱가포르)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 남아메리카(상파울루)

에 대한 서비스 할당량 및 제한AWS CloudShell

이 페이지에서는 다음 영역에 적용되는 서비스 할당량 및 제한에 대해 설명합니다.

- 퍼시스턴트 스토리지
- 월간 사용량
- 명령 크기
- 동시 사용 가능 셸
- 셸 세션
- 네트워크 액세스 및 데이터 전송
- 시스템 파일 및 페이지 새로고침

영구 스토리지

를 사용하면 각각 1GB의 영구 스토리지를 무료로 사용할AWS 리전 수 있습니다.AWS CloudShell 영구 스토리지는 홈 디렉터리 ($HOME) 에 있으며 사용자만 사용할 수 있습니다. 각 셸 세션이 종료된 후 재활용되는 임시 환경 리소스와 달리 홈 디렉터리의 데이터는 세션 간에 유지됩니다.

AWS CloudShell에서 사용을 중단하면 데이터는 마지막 세션이 끝난 후 120일 동안 해당 지역의 영구 저장소에 보관됩니다.AWS 리전 120일이 지나면 사용자가 조치를 취하지 않는 한 데이터가 해당 지역의 영구 저장소에서 자동으로 삭제됩니다. 거기서AWS CloudShell 다시 실행하면 제거를 방지할 수 AWS 리전 있습니다. 자세한 내용은 2단계: 지역 선택CloudShell, AWS 시작 및 셸 선택을 참조하십시오.

> **Note**
>
> 사용 시나리오
> Márcia는 미국 동부 (버니지아 북부) 및 유럽 (아일랜드) 의AWS 리전 두 개의 디렉터리에 파일을 저장하곤AWS CloudShell 했습니다. 그 후 그녀는 유럽 (아일랜드)AWS CloudShell 에서만 사용하기 시작했고 미국 동부 (버지니아 북부) 에서 Shell 세션을 시작하는 것을 중단했습니다. 미국 동부 (버지니아) 의 데이터 삭제 기한이 되기 전에 Márcia는 미국 동부 (버지니아 북부) 리전을AWS CloudShell 시작하고 선택하여 홈 디렉터리가 재활용되지 않도록 하기로 결정했습니다. 그녀는 셸 세션에 지속적으로 유럽 (아일랜드) 을 사용했기 때문에 해당 지역의 영구 스토리지는 영향을 받지 않습니다.

월별 사용량

각 항목에AWS CloudShell 대한 월별 사용량AWS 리전 할당량이 있습니다AWS 계정. 해당 지역의 월간 할당량에AWS CloudShell 도달한 후 액세스를 시도하면 셸 환경을 시작할 수 없는 이유를 설명하는 메시지가 표시됩니다.

> **Note**
>
> 월간 사용량 할당량을 늘려야 하는 경우 AWS Support에 문의하십시오.

명령 크기

명령 크기는 65412자를 초과할 수 없습니다.

> **Note**
>
> 65412자를 초과하는 명령을 실행하려면 선택한 언어로 스크립트를 만든 다음 명령줄 인터페이스에서 실행하십시오. 명령줄 인터페이스에서 액세스할 수 있는 사전 설치된 소프트웨어의 범위에 대한 자세한 내용은 사전 설치된 소프트웨어를 참조하십시오.
> 스크립트를 만든 다음 명령줄 인터페이스에서 실행하는 방법에 대한 예를 보려면 자습서: 시작하기를 참조하십시오AWS CloudShell.

동시 사용 가능 셸

- 동시 셸: 계정당AWS 리전 최대 10개의 셸을 동시에 실행할 수 있습니다.

셸 세션

- 비활성 세션:AWS CloudShell 대화형 셸 환경으로 20~30분 동안 키보드나 포인터를 사용하여 상호 작용하지 않으면 셸 세션이 종료됩니다. 실행 중인 프로세스는 상호 작용으로 계산되지 않습니다.
- 장기 실행 세션: 약 12시간 동안 지속적으로 실행되는 셸 세션은 사용자가 해당 기간 동안 정기적으로 이 세션과 상호 작용하더라도 자동으로 종료됩니다.

네트워크 액세스 및 데이터 전송

다음 제한은 사용자AWS CloudShell 환경의 인바운드 및 아웃바운드 트래픽 모두에 적용됩니다.

- 아웃바운드: 퍼블릭 인터넷에 액세스할 수 있습니다.
- 인바운드: 인바운드 포트에 액세스할 수 없습니다. 퍼블릭 IP 주소를 사용할 수 없습니다.

> **Warning**
>
> 공용 인터넷에 접속하면 특정 사용자가AWS CloudShell 환경에서 데이터를 내보낼 위험이 있습니다. IAM 관리자는 IAM 도구를 통해 신뢰할 수 있는AWS CloudShell 사용자의 허용 목록을 관리하는 것이 좋습니다. 특정 사용자의 액세스를 명시적으로 거부하는 방법에 대한 자세한 내용은 을 참조하십시오사용자 지정 정책 AWS CloudShell 사용 시 허용 가능한 조치 관리.

데이터 전송: 대용량 파일의 경우 파일 업로드 및 다운로드가AWS CloudShell 느릴 수 있습니다. 또는 셸의 명령줄 인터페이스를 사용하여 Amazon S3 버킷에서 환경으로 파일을 전송할 수 있습니다.

시스템 파일 및 페이지 재로드에 대한 제한

- 시스템 파일: 컴퓨팅 환경에 필요한 파일을 잘못 수정하면 환경에 액세스하거나 사용할 때 문제가 발생할 수 있습니다.AWS CloudShell 이 경우 다시 액세스하려면 홈 디렉터리를 삭제해야 할 수 있습니다.
- 페이지 새로고침:AWS CloudShell 인터페이스를 다시 로드하려면 운영 체제의 기본 단축키 시퀀스 대신 브라우저의 새로 고침 버튼을 사용하십시오.

AWS CloudShell 사용 설명서 기록

최신 업데이트

아래 표에 AWS CloudShell 사용 설명서의 주요 변경 사항이 설명되어 있습니다.

변경 사항	설명	날짜
사용 AWS CloudShell 설명서에 새 자습서가 추가되었습니다.	내부에 AWS CloudShell Docker 컨테이너를 구축하여 Amazon ECR 리포지토리로 푸시하는 방법과 이를 통해 Lambda 함수를 배포하는 방법을 자세히 설명하는 두 개의 새로운 자습서가 추가되었습니다. AWS CDK	2023년 12월 27일
Docker 컨테이너는 특정 지역에서 지원됩니다. AWS CloudShell	Docker 컨테이너에 대한 지원이 특정 지역에 AWS CloudShell 추가되었습니다.	2023년 12월 27일
AWS CloudShell이제 아마존 리눅스 2023 (AL2023) 를 사용하도록 마이그레이션했습니다.	AWS CloudShell현재 AL2023을 사용하고 있으며 아마존 리눅스 2에서 마이그레이션했습니다.	2023년 12월 4일
새로운 AWS CloudShell AWS 리전	AWS CloudShell은(는) 이제 일반적으로 다음 AWS 리전에서 사용할 수 있습니다. • 미국 서부(캘리포니아 북부) • 아프리카(케이프타운) • 아시아 태평양(홍콩) • 아시아 태평양(오사카) • 아시아 태평양(서울) • 아시아 태평양(자카르타) • 아시아 태평양(싱가포르)	2023년 6월 16일

	• 유럽(파리) • 유럽(스톡홀름) • 유럽(밀라노) • 중동(바레인) • 중동(UAE)	
Console Toolbar에서 AWS CloudShell시작	콘솔의 Console Toolbar 왼쪽 하단에 있는 를 선택하여 CloudShell 실행합니다 CloudShell .	2023년 3월 28일
새로운 AWS CloudShellAWS 리전	AWS CloudShell는 이제 다음 AWS 리전에서 사용 가능합니다. • 캐나다(중부) • 유럽(런던) • 남아메리카(상파울루)	2022년 10월 6일
AWS CloudShell미국 AWS에서 지원 GovCloud	AWS CloudShell이제 AWS GovCloud (미국) 리전에서 지원됩니다.	2022년 6월 29일
보안 FAQ	보안 문제 중심의 추가 FAQ.	2022년 4월 14일
Web Socket	네트워크 요구 사항에 WebSocket 프로토콜 사용을 설명하는 CloudShell 섹션이 추가되었습니다.	2022년 3월 21일
화살표 키 문제 해결: PowerShell	화살표 키를 눌렀을 때 글자가 잘못 출력되는 문제를 다음 단계에 따라 해결하세요.	2022년 2월 7일

탭 키 자동 완성	Tab 키를 눌러 부분적으로 입력된 명령이나 인수를 자동 완성하는 bash-completion을 사용하는 방법을 설명하는 새 설명서입니다.	2021년 9월 24일
AWS 리전 지정	AWS CLI 명령 기본값 AWS 리전에 대한 설명서.	2021년 5월 11일
PDF 및 Kindle 버전에서 서식 지정하기	테이블 셀의 이미지 크기와 텍스트를 수정했습니다.	2021년 3월 10일
선택된 AWS 리전 내 AWS CloudShell 정식 출시(GA)	AWS CloudShell은(는) 이제 일반적으로 다음 AWS 리전에서 사용할 수 있습니다. • 미국 동부(오하이오) • 미국 동부(버지니아 북부) • 미국 서부(오레곤) • 아시아 태평양(도쿄) • 유럽(아일랜드) • 아시아 태평양(뭄바이) • 아시아 태평양(시드니) • 유럽(프랑크푸르트)	2020년 12월 15일

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