시작 안내서

# AWS Management Console

버전 1.0

# AWS Management Console: 시작 안내서

# Table of Contents

# 이게 뭐야 AWS Management Console?

리소스 AWS Management Console관리를 위한 광범위한 서비스 콘솔 컬렉션으로 구성되며 이를 참조하는 웹 애플리케이션입니다. AWS 처음 로그인하면 콘솔 홈 페이지가 나타납니다. 홈 페이지는 각 서비스 콘솔에 대한 액세스와 AWS 관련 태스크를 수행하는 데 필요한 정보에 액세스할 수 있는 단일 위치를 제공합니다. 또한 최근 방문, AWS 건강 등과 같은 위젯을 추가, 제거 및 재정렬하여 콘솔 홈 환경을 사용자 지정할 수 있습니다.

> **Note**
>
> 언어 선택 옵션이 새로운 통합 설정(United Settings) 페이지로 이동했습니다. 자세한 내용은 AWS Management Console의 언어 변경을 참조하세요.

반면, 개별 서비스 콘솔은 클라우드 컴퓨팅을 위한 다양한 도구와 계정 및 결제에 대한 정보를 제공합니다.

## 원하는 디바이스 사용

AWS Management Console은 태블릿뿐만 아니라 다른 종류의 디바이스에서도 작동하도록 설계되었습니다.

- 화면에 더 많은 정보를 표시하도록 가로 및 세로 공간이 최대화되었습니다.

- 더 나은 터치 경험을 위해 버튼과 선택기가 더 커집니다.

안드로이드 및 iOS용 앱으로도 제공됩니다. AWS Management Console 이 앱은 전체 웹 경험에서 유용한 모바일 관련 태스크를 제공합니다. 예를 들어 휴대폰에서 기존 Amazon EC2 인스턴스와 Amazon CloudWatch 경보를 쉽게 보고 관리할 수 있습니다.

AWS 콘솔 모바일 앱은 Amazon 앱스토어, 구글 플레이 또는 iTunes에서 다운로드할 수 있습니다.

# 구성하기 AWS Management Console

이 항목에서는 구성 방법 AWS Management Console 및 통합 설정 페이지를 사용하여 모든 서비스 콘솔에 적용되는 기본값을 설정하는 방법에 대해 설명합니다. 또한 서비스와 리소스에 대한 정보를 추적하는 사용자 지정 구성 요소를 추가할 수 있는 콘솔 홈 대시보드의 기능인 위젯에 대해서도 설명합니다. AWS

주제



## 위젯 작업

콘솔 홈 대시보드에는 사용자 AWS 환경에 대한 중요한 정보를 표시하고 서비스 바로 가기를 제공하는 위젯이 포함되어 있습니다. 위젯을 추가 및 제거하거나, 다시 정렬 또는 크기를 조정하여 환경을 사용자 지정할 수 있습니다.

위젯을 추가하려면

1. 콘솔 홈 대시보드 오른쪽 상단 또는 하단에 있는 +위젯 추가 버튼을 선택합니다.
2. 위젯 제목 표시줄 왼쪽 상단에 있는 6개의 세로 점으로 표시된 드래그 표시기를 선택한 다음 콘솔 홈 대시보드로 드래그합니다.

위젯을 제거하려면

1. 위젯 제목 표시줄 오른쪽 상단에 있는 3개의 세로 점으로 표시된 줄임표를 선택합니다.
2. 위젯 제거(Remove widget)를 선택합니다.

위젯을 다시 정렬하려면

- 위젯 제목 표시줄 왼쪽 상단에 있는 6개의 세로 점으로 표시된 드래그 표시기를 선택한 다음 콘솔 홈 대시보드의 새 위치로 위젯을 드래그합니다.

위젯의 크기를 조정하려면

- 위젯 오른쪽 하단의 크기 조정 아이콘을 선택한 다음 위젯을 드래그하여 크기를 조정합니다.

위젯 구성 및 설정부터 다시 시작하려면 콘솔 홈 대시보드를 기본 레이아웃으로 재설정하면 됩니다. 이렇게 하면 변경 사항이 콘솔 홈 대시보드 레이아웃으로 되돌아가며 모든 위젯이 기본 위치 및 크기로 복원됩니다.

페이지를 기본 레이아웃으로 재설정하려면

1. 페이지 오른쪽 상단에서 기본 레이아웃으로 재설정을 선택합니다.
2. 확인하려면 재설정을 선택합니다.

> **Note**
>
> 그러면 모든 변경 사항이 콘솔 홈 대시보드의 레이아웃으로 되돌아갑니다.

콘솔 홈 대시보드에서 새 위젯을 요청하려면

1. 콘솔 홈 대시보드 왼쪽 하단에서 다른 위젯을 보고 싶다면 알려주세요!를 선택합니다.

   콘솔 홈 대시보드에 추가되었으면 하는 위젯을 설명해 주세요.
2. 제출을 선택합니다.

> **Note**
>
> 제안은 정기적으로 검토되어, 향후 업데이트 시 새로운 위젯이 AWS Management Console에 추가될 수 있습니다.

# 통합 설정 구성

AWS Management Console 통합 설정 페이지에서 표시, 언어, 지역과 같은 설정 및 기본값을 구성할 수 있습니다. 시각적 모드와 기본 언어는 탐색 모음에서 직접 설정할 수도 있습니다. 이러한 변경 사항은 모든 서비스 콘솔에 적용됩니다.

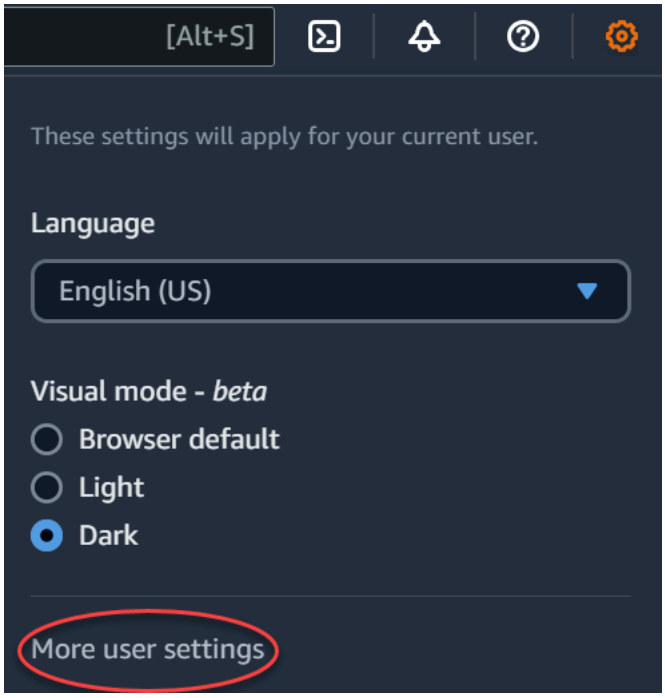
> **Important**
>
> 설정, 즐겨 찾는 서비스 및 최근에 방문한 서비스가 전 세계에 유지되도록 이 데이터는 기본적으로 비활성화된 지역을 포함하여 모든 AWS 리전지역에 저장됩니다. 이러한 리전은 아프리카(케이프타운), 아시아 태평양(홍콩), 아시아 태평양(하이데라바드), 아시아 태평양(자카르타), 유럽(밀라노), 유럽(스페인), 유럽(취리히), 중동(바레인), 중동(UAE)입니다. 그래도 여전히 수동으로 리전에 액세스할 수 있도록 설정한 다음 해당 리전에서 리소스를 생성하고 관리해야 합니다. 이 데이터를 모두 저장하지 않으려면 모두 AWS 리전재설정을 선택하여 설정을 지운 다음 설정 관리에서 최근에 방문한 서비스를 기억하지 않도록 설정하세요.

## 통합 설정에 액세스하기

다음 절차는 통합 설정에 액세스하는 방법을 설명합니다.

통합 설정에 액세스하는 방법

1. AWS Management Console에 로그인합니다.
2. 내비게이션 바에서 기어 아이콘을 선택합니다.
3. 통합 설정 페이지를 열려면 더 많은 사용자 설정을 선택합니다.




## 통합 설정 재설정

통합 설정을 재설정하여 모든 통합 설정 구성을 삭제하고 기본 설정을 복원할 수 있습니다.

> **Note**
>
> 이는 탐색 및 서비스 메뉴의 AWS즐겨찾는 서비스, 콘솔 홈 위젯과 에서 최근에 방문한 서비스, 기본 언어 AWS Console Mobile Application, 기본 지역 및 시각 모드와 같이 서비스에 적용되는 모든 설정을 포함하여 의 여러 영역에 영향을 줍니다.

모든 통합 설정을 재설정하려면

1. AWS Management Console에 로그인합니다.
2. 내비게이션 바에서 기어 아이콘을 선택합니다.
3. 추가 사용자 설정을 선택하여 통합 설정 페이지를 엽니다.
4. 모두 재설정을 선택합니다.

## 통합 설정 편집

다음 절차는 기본 설정을 편집하는 방법을 설명합니다.

통합 설정을 편집하려면

1. AWS Management Console에 로그인합니다.
2. 내비게이션 바에서 기어 아이콘을 선택합니다.
3. 추가 사용자 설정을 선택하여 통합 설정 페이지를 엽니다.
4. 선호하는 설정 옆의 편집(Edit)을 선택합니다.

   - 현지화 및 기본 지역:(Localization and default Region:)
     - 언어에서는 콘솔 텍스트의 기본 언어를 선택할 수 있습니다.
     - 기본 리전(Default Region)에서는 로그인할 때마다 적용되는 기본 리전을 선택할 수 있습니다. 계정에 사용할 수 있는 리전 중 하나를 선택할 수 있습니다. 마지막으로 사용한 리전을 기본값으로 선택할 수도 있습니다.

       AWS Management Console의 리전 라우팅에 대한 자세한 내용은 리전 선택을 참조하세요.
   - 표시:
     - Visual mode(시각적 모드)에서는 콘솔을 라이트 모드, 다크 모드 또는 브라우저의 기본 표시 모드로 설정할 수 있습니다.

       다크 모드는 베타 기능이며 일부 AWS 서비스 콘솔에는 적용되지 않을 수 있습니다.
     - 즐겨찾기 모음 표시는 해당 아이콘과 함께 전체 서비스 이름을 표시하거나 서비스의 아이콘만 표시하도록 즐겨찾기 모음 표시를 전환합니다.
     - 즐겨찾기 모음 아이콘 크기는 즐겨찾기 모음 표시의 서비스 아이콘 크기를 소형(16x16 픽셀)과 대형(24x24 픽셀) 간에 전환합니다.
   - 설정 관리:
     - 최근 방문한 서비스 기억을 사용하면 최근에 방문한 서비스를 AWS Management Console 기억할지 여부를 선택할 수 있습니다. 이 기능을 끄면 최근에 방문한 서비스 기록도 삭제되므로 서비스 메뉴나 콘솔 홈 위젯에 최근에 방문한 서비스가 더 이상 표시되지 않습니다. AWS Console Mobile Application
5. 변경 사항 저장를 선택합니다.

## 비주얼 모드 변경 AWS Management Console

비주얼 모드는 본체를 조명 모드, 다크 모드 또는 브라우저의 기본 디스플레이 모드로 설정합니다.

탐색 모음에서 시각적 모드를 변경하려면

1. AWS Management Console에 로그인합니다.
2. 내비게이션 바에서 기어 아이콘을 선택합니다.
3. 시각적 모드에서 밝은 모드를 원하면 라이트, 어두운 모드를 원하면 다크, 브라우저의 기본 표시 모드를 사용하려면 브라우저 기본값을 선택합니다.

## 통합 설정에서 기본 언어 변경

다음 절차는 탐색 막대를 사용하여 기본 언어를 변경하는 방법을 설명합니다.

탐색 모음에서 기본 언어를 변경하려면

1. AWS Management Console에 로그인합니다.
2. 내비게이션 바에서 기어 아이콘을 선택합니다.
3. 언어에서, 브라우저 기본값을 선택하거나 드롭다운 목록에서 원하는 언어를 선택합니다.

## 리전 선택

대부분의 서비스에서는 리소스 관리 위치를 AWS 리전 지정하는 서비스를 선택할 수 있습니다. 지역은 동일한 지리적 영역에 위치한 AWS 리소스 집합입니다. 일부 서비스 (예:) 의 경우 AWS Management Console또는 지역을 선택할 필요가 없습니다. AWS Identity and Access Management AWS 리전에 대해 자세히 알아보려면 AWS 일반 참조의 AWS 리전관리 섹션을 참조하세요.

리전을 선택하려면

1. AWS Management Console에 로그인합니다.
2. 해당 서비스의 콘솔로 이동할 서비스를 선택합니다.
3. 탐색 모음에서 현재 표시된 리전의 이름을 선택합니다. 그런 다음 전환하려는 리전을 선택합니다.

기본 리전을 선택하려면

1. 탐색 모음에서 설정 아이콘을 선택한 다음 더 많은 사용자 설정을 선택하여 통합 설정 페이지로 이동합니다.
2. 현지화 및 기본 지역(Localization and default Region) 옆의 편집(Edit)을 선택합니다.

3. 기본 지역을 선택한 다음 설정 저장을 선택합니다. 기본 리전을 선택하지 않으면 마지막으로 방문한 리전이 기본값이 됩니다.
4. (선택 사항) 새 기본 지역으로 바로 이동하려면 새 기본 지역으로 이동을 선택합니다.

> **Note**
>
> AWS 리소스를 생성했지만 콘솔에 해당 리소스가 표시되지 않는 경우 콘솔에 다른 지역의 리소스가 표시되고 있을 수 있습니다. 일부 리소스(예: Amazon EC2 인스턴스)는 해당 리소스가 생성된 리전에 한정됩니다. 해당 리소스를 보려면 리전 선택기를 사용하여 리소스가 포함된 리전을 선택합니다.

# 즐겨찾기 추가 및 제거

자주 사용하는 서비스에 보다 빠르게 액세스하려면 서비스 콘솔을 즐겨찾기(Favorites) 목록에 저장하면 됩니다.

즐겨찾기(Favorites)의 목록에 서비스 추가

1. AWS Management Console에 로그인합니다.
2. 페이지 오른쪽 상단 또는 하단에 있는 위젯 추가(Add widgets) 버튼을 선택합니다.
3. 위젯 추가 메뉴에서 콘솔에 추가할 즐겨찾기를 선택한 다음 추가를 선택합니다.

   즐겨찾기는 콘솔 홈의 하단에 추가됩니다. 위젯 상단의 제목 표시줄을 선택한 다음 위젯을 페이지의 새 위치로 끌어 놓아 즐겨찾기를 끌어서 놓을 수 있습니다.
4. 탐색 모음에서 서비스(Services)를 선택합니다.
5. 최근 방문 목록 또는 모든 서비스 목록에서 즐겨찾기로 추가하려는 서비스의 이름을 찾습니다.
6. 해당 서비스 이름 왼쪽에 있는 별을 선택합니다.
7. 이전 두 단계를 반복하여 즐겨찾기 목록에 서비스를 더 추가합니다.

즐겨찾기(Favorites)의 목록에서 서비스 제거

1. 탐색 모음에서 서비스(Services)를 선택합니다.
2. 다음 중 하나를 수행하십시오.

- 즐겨찾기 목록에서 서비스 이름 위에 마우스 포인터를 올려 놓습니다. 그런 다음 서비스 이름 오른쪽에 있는 ×를 선택합니다.
- [최근 방문(Recently visited)] 목록 또는 [모든 서비스(All services)] 목록에서 [즐겨찾기(Favorites)] 목록에 포함된 서비스의 이름 옆에 있는 별표를 선택 취소합니다.

# 암호 변경

계정 소유자인 경우 에서 AWS 계정 암호를 변경할 수 AWS Management Console있습니다.

암호를 변경하려면

1. AWS Management Console에 로그인합니다.
2. 탐색 모음에서 계정 이름을 선택합니다.
3. Security credentials(보안 자격 증명)를 선택합니다.
4. 표시되는 옵션은 AWS 계정 유형에 따라 달라집니다. 콘솔에 표시되는 지침에 따라 암호를 변경합니다.
5. 현재 암호를 한 번 입력하고 새 암호를 두 번 입력합니다.

   새 암호는 8자 이상이어야 하며 다음을 포함해야 합니다.

   - 하나 이상의 기호
   - 하나 이상의 숫자
   - 하나 이상의 대문자
   - 하나 이상의 소문자
6. 암호 변경(Change Password) 또는 변경 사항 저장(Save changes)을 선택합니다.

# 의 언어 변경 AWS Management Console

이 AWS Console Home 환경에는 AWS 서비스의 기본 언어를 변경할 수 있는 통합 설정 페이지가 포함됩니다 AWS Management Console. 탐색 모음에서 액세스할 수 있는 설정 메뉴에서 기본 언어를 빠르게 변경할 수도 있습니다. 전체 콘솔의 어느 위치에서나 이를 변경할 수 있습니다.

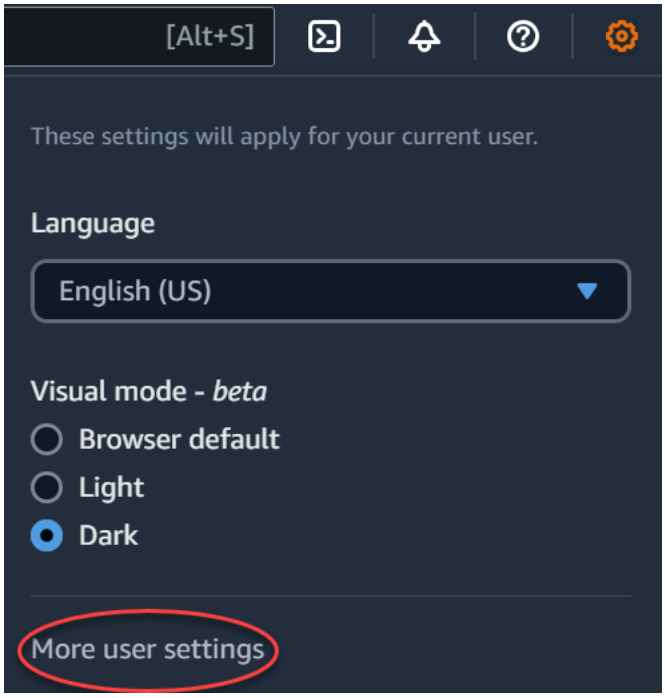
> **Note**
>
> 이 절차에 따라 모든 콘솔의 언어를 변경할 수 있지만, AWS 설명서의 언어는 변경할 수 없습니다. 설명서에 사용된 언어를 변경하려면 설명서 페이지의 오른쪽 상단에 있는 언어 메뉴를 사용합니다.

는 AWS Management Console 현재 다음 언어를 지원합니다.

- 영어(미국)
- 영어(영국)
- 인도네시아어
- 독일어
- 프랑스어
- 일본어
- 스페인어
- 이탈리아어
- 포르투갈어
- 한국어
- 중국어 간체
- 중국어 번체

통합 설정에서 기본 언어를 변경하려면

1. AWS Management Console에 로그인합니다.
2. 탐색 모음에서 설정 아이콘을 선택합니다.
3. 통합 설정 페이지를 열려면 더 많은 사용자 설정을 선택합니다.




4. 통합 설정(Unified Settings)에서 현지화 및 기본 지역(Localization and default Region) 옆의 편집(Edit)을 선택합니다.
5. 콘솔에 사용할 언어를 선택하려면 다음 옵션 중 하나를 선택합니다.

   - 드롭다운 목록에서 브라우저 기본값을 선택한 다음 설정 저장을 선택합니다.

     모든 AWS 서비스의 콘솔 텍스트는 브라우저 설정에서 설정한 기본 언어로 표시됩니다.

     > **Note**
     > 브라우저 기본값은 AWS Management Console에서 지원하는 언어만 지원합니다.

   - 드롭다운 목록에서 브라우저 기본값을 선택한 다음 설정 저장을 선택합니다.

     모든 AWS 서비스의 콘솔 텍스트가 기본 설정 언어로 표시됩니다.

탐색 모음에서 기본 언어를 변경하려면

1. AWS Management Console에 로그인합니다.
2. 탐색 모음에서 설정 아이콘을 선택합니다.
3. 언어에서, 브라우저 기본값을 선택하거나 드롭다운 목록에서 원하는 언어를 선택합니다.

# 서비스 시작하기

AWS Management Console에서는 여러 가지 방법으로 개별 서비스 콘솔로 이동할 수 있습니다.

특정 서비스의 콘솔을 열려면

다음 중 하나를 수행하세요.

- 탐색 모음의 검색 상자에 서비스 이름의 전체 또는 일부를 입력합니다. [서비스(Services)]의 검색 결과 목록에서 원하는 서비스를 선택합니다. 자세한 정보는 통합 검색을 사용하여 제품, 서비스, 기능 등 검색을 참조하십시오.
- 최근 방문한 서비스(Recently visited services) 위젯에서 서비스 이름을 선택합니다.
- 최근 방문한 서비스(Recently visited services) 위젯에서 모든 AWS 서비스 보기(View all AWS services)를 선택합니다. 그런 다음 모든 AWS 서비스(All AWS services) 페이지에서 서비스 이름을 선택합니다.
- 탐색 모음에서 [서비스(Services)]를 선택하여 서비스의 전체 목록을 엽니다. 그런 다음 [최근 방문(Recently visited)] 또는 [모든 서비스(All services)]에서 서비스를 선택합니다.

# 통합 검색을 사용하여 제품, 서비스, 기능 등 검색

탐색 모음의 검색 상자는 AWS 서비스 및 기능, 서비스 설명서, AWS Marketplace를 추적하기 위한 통합 검색 도구를 제공합니다. 몇 글자를 입력하기만 하면 이 모든 범주의 결과를 볼 수 있습니다. 문자를 많이 입력할수록 검색 결과가 더 구체화됩니다.

서비스, 기능, 설명서 또는 AWS Marketplace 제품을 검색하려면

1. 의 탐색 표시줄에 있는 검색 상자에 검색어 전체 또는 일부를 입력합니다. AWS Management Console
2. 다음 중 하나를 수행하여 검색을 구체화하고 세부 정보를 확인합니다.
   - 결과의 범위를 원하는 콘텐츠 유형으로 좁히려면 왼쪽의 범주 중 하나를 선택합니다.
   - 특정 범주에 대한 더 많은 결과를 보려면 각 범주 머리글 옆에 있는 [*n*개 결과 모두 보기(See all n results)]를 선택합니다. 기본 결과 목록으로 돌아가려면 왼쪽 상단에서 [뒤로(Back)]를 선택합니다.
   - 특정 서비스의 많이 사용하는 기능으로 빠르게 이동하려면 결과에서 서비스 이름을 찾아 링크를 선택합니다.
   - 문서 또는 AWS Marketplace 결과에 대한 자세한 내용을 보려면 검색결과 제목에서 일시 중지하십시오.
3. 링크를 선택하여 원하는 서비스, 주제 또는 AWS Marketplace 페이지로 이동합니다.

> **Tip**
>
> 키보드를 사용하여 상위 검색 결과로 빠르게 이동할 수도 있습니다. 먼저 Alt+s(Windows) 또는 Option+s(macOS)를 눌러 검색 창에 액세스합니다. 그런 다음 검색어를 입력합니다. 의도한 결과가 목록의 상단에 표시되면 Enter 키를 누릅니다. 예를 들어 Amazon EC2 콘솔로 빠르게 이동하려면 ec2를 입력하고 Enter 키를 누릅니다.

# Amazon Q 개발자와 채팅

Amazon Q Developer는 애플리케이션을 이해, 구축, 확장 및 운영하는 데 도움이 되는 생성적 인공 지능 (AI) 기반 대화형 도우미입니다. AWS AWS 아키텍처, AWS 리소스, 모범 사례 AWS, 설명서 등에 대한 질문을 포함하여 Amazon Q에 질문을 할 수 있습니다. 지원 사례를 생성하고 실제 상담원으로부터 지원을 받을 수도 있습니다. 자세한 내용은 Amazon Q란 무엇입니까? 를 참조하십시오. Amazon Q 개발자 사용 설명서에서 확인할 수 있습니다.

## Amazon Q로 시작하세요

육각형 Amazon Q 아이콘을 선택하여 AWS 설명서 웹 사이트 AWS Management Console, AWS 웹 사이트 또는 AWS 콘솔 모바일 애플리케이션에서 Amazon Q와 채팅을 시작할 수 있습니다. 자세한 내용은 Amazon Q 개발자 사용 설명서의 Amazon Q Developer 시작하기를 참조하십시오.

## 예시 질문

다음은 Amazon Q에 질문할 수 있는 몇 가지 예시 질문입니다.

- `How do I get billing support?`
- `How do I create an EC2 instance?`
- `How do I troubleshoot a "Failed to load" error?`
- `How do I close an AWS account?`
- `Can you connect me with a person?`

# 내 애플리케이션은 어디에 있나요? AWS

myApplications는 AWS에 구축된 애플리케이션의 비용, 상태, 보안 태세 및 성능을 관리하고 모니터링할 수 있도록 하는 콘솔 홈의 확장입니다. 계정의 모든 애플리케이션, 모든 애플리케이션의 주요 지표, 여러 서비스 콘솔의 비용, 보안, 운영 지표 및 인사이트 개요에 액세스할 수 있습니다 AWS Management Console. myApplications에는 다음이 포함됩니다.

- 콘솔 홈 페이지의 애플리케이션 위젯
- 애플리케이션 리소스 비용 및 보안 조사 결과를 보는 데 사용할 수 있는 myApplications
- 비용, 성능, 보안 조사 결과 같은 주요 애플리케이션 지표를 볼 수 있는 myApplications 대시보드

## myApplications의 기능

- 애플리케이션 생성 - 새 애플리케이션을 생성하고 리소스를 구성합니다. 애플리케이션은 내 애플리케이션에 자동으로 표시되므로, API AWS Management Console, CLI 및 SDK에서 조치를 취할 수 있습니다. 애플리케이션을 생성할 때 코드형 인프라(IAC)가 생성되며 myApplication 대시보드에서 액세스할 수 있습니다. IaC는 테라폼을 포함한 IaC 도구에서 사용할 수 있습니다. AWS CloudFormation
- 애플리케이션 액세스 - myApplications 위젯에서 원하는 애플리케이션을 선택하여 빠르게 액세스할 수 있습니다.
- 애플리케이션 지표 비교 - myApplications를 사용하여 애플리케이션 리소스 비용, 여러 애플리케이션에 대한 중요한 보안 조사 결과 수 등 애플리케이션의 주요 지표를 비교할 수 있습니다.
- 애플리케이션 모니터링 및 관리 — 알람, 카나리아, 검색 결과 Amazon CloudWatch, 비용 추세를 통해 얻은 서비스 수준 목표를 사용하여 애플리케이션 상태 및 성능을 평가합니다. AWS Security Hub AWS Cost Explorer Service또한 에서 컴퓨팅 메트릭 요약 및 최적화를 찾고 리소스 규정 준수 및 구성 상태를 관리할 수 있습니다. AWS Systems Manager

## 관련 서비스

myApplications는 다음과 같은 서비스를 사용합니다.

- AppRegistry
- AppManager
- Amazon CloudWatch

- Amazon EC2
- AWS Lambda
- AWS 리소스 탐색기
- AWS Security Hub
- Systems Manager
- AWS Service Catalog
- 태그 지정

## myApplications 액세스

AWS Management Console의 왼쪽 사이드바에서 myApplications를 선택하여 myApplications에 액세스할 수 있습니다.

## 요금

MyApplications AWS on은 추가 비용 없이 제공됩니다. 설정 요금이나 사전 약정은 없습니다. myApplications 대시보드에 요약된 기본 리소스 및 서비스의 사용 요금은 해당 리소스에 대해 게시된 요금으로 여전히 적용됩니다.

## 지원되는 리전

MyApplications는 다음에서 사용할 수 있습니다. AWS 리전

- 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(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)

- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 유럽(스톡홀름)
- 남아메리카(상파울루)

## 옵트인 리전

옵트인 리전에서 기본 지원 리전으로 이러한 리전에서 myApplications를 사용하려면 해당 리전을 수동으로 활성화해야 합니다. 에 대한 AWS 리전자세한 내용은 관리를 AWS 리전 참조하십시오. 다음과 같은 옵트인 리전이 지원됩니다.

- 아프리카(케이프타운)
- 아시아 태평양(홍콩)
- 아시아 태평양(하이데라바드)
- 아시아 태평양(자카르타)
- 아시아 태평양(멜버른)
- 유럽(밀라노)
- 유럽(스페인)
- 유럽(취리히)
- 중동(바레인)
- 중동(UAE)
- 이스라엘(텔아비브)

# myApplications 시작하기

myApplications를 사용하여 애플리케이션을 생성, 모니터링 및 관리하려면 다음 단계를 사용하세요.

## 1단계: 애플리케이션 생성

새 애플리케이션을 생성하거나 2023년 11월 8일 이전에 생성된 기존 AppRegistry 애플리케이션을 온보딩하여 MyApplications를 시작하십시오.

## Create an application

애플리케이션을 생성하려면

1. AWS Management Console에 로그인합니다.
2. 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 애플리케이션 생성을 선택합니다.
4. 애플리케이션 이름을 입력합니다.
5. (선택 사항) 애플리케이션 설명을 입력합니다.
6. (선택 사항) 태그를 추가합니다. 태그는 리소스에 대한 메타데이터를 유지하기 위해 리소스에 적용되는 키-값 쌍입니다.

> **Note**
>
> AWS 애플리케이션 태그는 새로 생성된 애플리케이션에 자동으로 적용되며 애플리케이션과 관련된 리소스를 식별하는 데 사용할 수 있습니다. 자세한 내용은 AWS Service Catalog AppRegistry 관리자 안내서의 AWS 애플리케이션 태그를 참조하십시오.

7. (선택 사항) 속성 그룹을 추가합니다. 속성 그룹을 사용하여 애플리케이션 메타데이터를 저장할 수 있습니다.
8. 다음을 선택합니다.
9. (선택 사항) 기존 리소스를 추가합니다.

> **Note**
>
> 리소스를 검색하고 추가하려면 AWS 리소스 탐색기를 켜야 합니다. 자세한 내용은 시작하기를 참조하십시오 AWS 리소스 탐색기.
> 추가된 모든 리소스에는 AWS 애플리케이션 태그가 지정됩니다.

   a. 리소스 선택을 선택합니다.
   b. (선택 사항) 보기를 선택합니다.
   c. 리소스를 검색합니다. 키워드, 이름 또는 유형별로 검색하거나 리소스 유형을 선택할 수 있습니다.

> **Note**
>
> 찾고 있는 리소스를 찾을 수 없는 경우 문제를 해결하세요. AWS 리소스 탐색 기자세한 내용은 Resource Explorer User Guide의 Troubleshooting Resource Explorer search issues를 참조하세요.

d. 추가할 리소스 옆에 있는 확인란을 선택합니다.

e. 추가를 선택합니다.

f. 다음을 선택합니다.

10. 선택 사항을 검토합니다.
11. AWS CloudFormation 스택을 연결하는 경우 페이지 하단의 체크박스를 선택하세요.

> **Note**
>
> 애플리케이션에 추가되는 AWS CloudFormation 모든 리소스에 애플리케이션 태그가 지정되므로 애플리케이션에 스택을 추가하려면 스택 업데이트가 필요합니다. AWS 스택이 마지막으로 업데이트된 이후에 수행된 수동 구성은 이 업데이트 후에 반영되지 않을 수 있습니다. 따라서 다운타임이나 기타 애플리케이션 문제가 발생할 수 있습니다. 자세한 정보에 대해서는AWS CloudFormation 사용 설명서의 스택 리소스의 업데이트 동작을 참조하세요.

12. 애플리케이션 생성을 선택합니다.

Onboard existing application

기존 애플리케이션을 온보딩하려면 AppRegistry

1. AWS Management Console에 로그인합니다.
2. 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 검색 창을 사용하여 애플리케이션을 찾습니다.
4. 애플리케이션을 선택합니다.
5. ****** ## 온보딩을 선택합니다.
6. CloudFormation 스택을 연결하는 경우 알림 상자에서 확인란을 선택합니다.
7. 애플리케이션 온보딩을 선택합니다.

## 2단계: 애플리케이션 보기

모든 리전 또는 특정 리전의 애플리케이션과 관련 정보를 카드 보기 또는 테이블 보기로 볼 수 있습니다.

애플리케이션을 보려면

1. 왼쪽 사이드바에서 myApplications를 선택합니다.
2. 리전에서 현재 리전 또는 지원되는 리전을 선택합니다.
3. 특정 애플리케이션을 찾으려면 검색 창에 이름, 키워드 또는 설명을 입력합니다.
4. (선택 사항) 기본 보기는 카드 보기입니다. 애플리케이션 페이지를 사용자 지정하려면 다음을 수행합니다.

    a. 기어 모양 아이콘을 선택합니다.
    b. (선택 사항) 페이지 크기를 선택합니다.
    c. (선택 사항) 카드 보기 또는 테이블 보기를 선택합니다.
    d. (선택 사항) 페이지 크기를 선택합니다.
    e. (선택 사항) 테이블 보기를 사용하는 경우 테이블 보기의 속성을 선택합니다.
    f. (선택 사항) 표시되는 애플리케이션 속성과 나타나는 순서를 전환합니다.
    g. 확인을 선택합니다.

# 애플리케이션 관리

이 주제에서는 애플리케이션을 관리하는 방법을 다룹니다.

## 애플리케이션 편집

응용 프로그램 편집이 열리고 설명을 업데이트할 AppRegistry 수 있습니다. 를 AppRegistry 사용하여 애플리케이션의 태그와 속성 그룹을 편집할 수도 있습니다.

애플리케이션을 편집하려면

1. AWS Management Console을 엽니다.
2. 콘솔의 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 편집하려는 애플리케이션을 선택합니다.
4. myApplication 대시보드에서 작업을 선택한 다음 애플리케이션 편집을 선택합니다.

5. 애플리케이션 설명 편집에서 설명을 업데이트한 다음 변경 내용 저장 선택합니다.

태그를 편집하려면

- AWS Service Catalog AppRegistry 관리자 안내서의 태그 관리에 나와 있는 단계를 따르십시오.

속성 그룹을 편집하려면

- AWS Service Catalog AppRegistry 관리자 안내서의 속성 그룹 편집에 나와 있는 단계를 따르십시오.

## 애플리케이션 삭제

더 이상 필요하지 않은 애플리케이션은 삭제할 수 있습니다.

애플리케이션 삭제

1. AWS Management Console을 엽니다.
2. 콘솔의 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 삭제하려는 애플리케이션을 선택합니다.
4. myApplication 대시보드에서 작업을 선택합니다.
5. 애플리케이션 삭제를 선택합니다.
6. 삭제를 선택합니다.
7. 삭제를 확인한 다음 애플리케이션 삭제를 선택합니다.

## 코드 스니펫 생성

myApplications는 모든 애플리케이션에 대한 코드 스니펫을 생성합니다. 코드 스니펫을 사용하면 코드형 인프라(IaC) 도구를 통해 새로 생성된 리소스를 애플리케이션에 자동으로 추가할 수 있습니다. 추가된 모든 리소스에는 AWS 애플리케이션 태그가 지정되어 애플리케이션과 연결됩니다.

애플리케이션에 대한 코드 스니펫을 생성하려면

1. AWS Management Console을 엽니다.
2. 콘솔의 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 애플리케이션을 검색하고 선택합니다.

4. 작업을 선택합니다.
5. 코드 스니펫 받기를 선택합니다.
6. 코드 스니펫 유형을 선택합니다.
7. 복사를 선택하여 코드를 클립보드로 복사합니다.
8. 코드를 IaC 도구에 붙여 넣습니다.

# 리소스 관리

이 주제에서는 리소스 관리 방법을 설명합니다.

## 리소스 추가

애플리케이션에 리소스를 추가하면 리소스를 그룹화하고 보안, 성능 및 규정 준수를 관리할 수 있습니다.

리소스를 추가하려면

1. AWS Management Console을 엽니다.
2. 콘솔의 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 애플리케이션을 검색하고 선택합니다.
4. 리소스 관리를 선택합니다.
5. 리소스 추가(Add resources)를 선택합니다.
6. (선택 사항) 보기를 선택합니다.
7. 리소스를 검색합니다. 키워드, 이름 또는 유형별로 검색하거나 리소스 유형을 선택할 수 있습니다.

> **Note**
>
> 찾고 있는 리소스를 찾을 수 없는 경우 문제를 해결하세요. AWS 리소스 탐색기자세한 내용은 Resource Explorer User Guide의 Troubleshooting Resource Explorer search issues를 참조하세요.

8. 추가할 리소스 옆에 있는 확인란을 선택합니다.
9. 추가를 선택합니다.

# 리소스 제거

리소스를 제거하여 애플리케이션에서 리소스 연결을 해제할 수 있습니다.

리소스를 제거하려면

1. AWS Management Console을 엽니다.
2. 콘솔의 왼쪽 사이드바에서 myApplications를 선택합니다.
3. 애플리케이션을 검색하고 선택합니다.
4. 리소스 관리를 선택합니다.
5. (선택 사항) 보기를 선택합니다.
6. 리소스를 검색합니다. 키워드, 이름 또는 유형별로 검색하거나 리소스 유형을 선택할 수 있습니다.

> **Note**
>
> 찾고 있는 리소스를 찾을 수 없는 경우 문제를 해결하세요. AWS 리소스 탐색기자세한 내용은 Resource Explorer User Guide의 Troubleshooting Resource Explorer search issues를 참조하세요.

7. 제거를 선택합니다.
8. 리소스 제거를 선택하여 리소스를 제거하려고 함을 확인합니다.

# myApplications 대시보드

생성하거나 온보딩하는 각 애플리케이션에는 고유한 myApplications 대시보드가 있습니다. MyApplications 대시보드에는 여러 서비스의 통찰력을 제공하는 비용, 보안 및 운영 위젯이 포함되어 있습니다. AWS 또한 각 위젯은 즐겨찾기에 추가하거나, 재정렬하거나, 제거하거나, 크기를 조정할 수 있습니다. 자세한 정보는 위젯 작업을 참조하세요.

## 애플리케이션 대시보드 설정 위젯

이 위젯에는 애플리케이션 리소스 관리를 AWS 서비스 위한 구성을 지원하는 데 사용할 수 있는 추천 시작 활동 목록이 포함되어 있습니다.

# 애플리케이션 요약 위젯

이 위젯은 애플리케이션의 이름, 설명, AWS 애플리케이션 태그를 보여줍니다. 코드형 인프라(IAC)에서 애플리케이션 태그를 액세스하고 복사하여 리소스에 수동으로 태그를 지정할 수 있습니다.

# 컴퓨팅 위젯

이 위젯은 애플리케이션에 추가하는 컴퓨팅 리소스에 대한 정보와 지표를 표시합니다. 여기에는 총 경보 수 및 총 컴퓨팅 리소스 유형 수가 포함됩니다. 이 위젯에는 Amazon EC2 인스턴스 CPU 사용률 및 Lambda Amazon CloudWatch 호출에 대한 리소스 성능 지표 추세 차트도 표시됩니다.

## 컴퓨팅 위젯 구성

컴퓨팅 위젯에 데이터를 채우려면 애플리케이션에 대해 하나 이상의 Amazon EC2 인스턴스 또는 Lambda 함수를 설정합니다. 자세한 내용은 AWS Lambda 개발자 안내서의 Amazon Elastic Compute 클라우드 문서 및 Lambda 시작하기를 참조하세요.

# 비용 및 사용량 위젯

이 위젯은 애플리케이션 AWS 리소스의 비용 및 사용 데이터를 보여줍니다. 이 데이터를 사용하여 월별 비용을 비교하고 AWS 서비스별 비용을 볼 수 있습니다. 이 위젯은 리소스와 직접 연관되지 않은 세금, 수수료 및 기타 공유 비용을 제외하고 AWS 애플리케이션 태그로 태그가 지정된 리소스의 비용만 요약합니다. 표시된 비용은 일반 비용이며 24시간마다 최소 1회 이상 업데이트됩니다. 자세한 내용은 AWS Cost Management 사용 설명서의 AWS 리소스 탐색기를 사용한 비용 분석을 참조하세요.

## 비용 및 사용량 위젯 구성

비용 및 사용량 위젯을 구성하려면 애플리케이션과 AWS Cost Explorer Service 계정에서 활성화하십시오. 이 서비스는 추가 비용 없이 제공되며 설정 비용이나 사전 약정이 없습니다. 자세한 내용은 AWS Cost Management 사용 설명서의 Cost Explorer 활성화를 참조하세요.

# AWS 보안 위젯

이 위젯은 애플리케이션에 대한 AWS 보안의 보안 결과를 표시합니다. AWS 보안은 애플리케이션의 보안 결과를 종합적으로 보여줍니다 AWS. 심각도별로 최근 우선순위 조사 결과에 액세스하고, 보안 태세를 모니터링하고, 중요하거나 심각도가 높은 최근 조사 결과에 액세스하고, 다음 단계를 위한 인사이트를 얻을 수 있습니다. 자세한 정보는 AWS Security Hub을 참조하세요.

## AWS 보안 위젯 구성

AWS 보안 위젯을 구성하려면 애플리케이션과 계정을 AWS Security Hub 설정해야 합니다. 자세한 내용은 What is AWS Security Hub? 를 참조하십시오. AWS Security Hub 사용 설명서에서 요금 정보는 AWS Security Hub 사용 설명서의 AWS Security Hub 무료 평가판, 사용량 및 가격을 참조하세요.

AWS Security Hub AWS Config 기록을 구성해야 합니다. 이 서비스는 AWS 계정과 관련된 리소스를 자세히 보여줍니다. 자세한 내용은AWS Systems Manager 사용 설명서의 AWS Systems Manager를 참조하십시오.

# DevOps 위젯

이 위젯은 운영 인사이트를 보여주므로 규정 준수를 평가하고 애플리케이션에 대한 조치를 취할 수 있습니다. 이러한 인사이트에는 다음이 포함됩니다.

- 집합 관리
- 상태 관리
- 패치 관리
- 구성 및 OpsItems 관리

## DevOps 위젯 구성

DevOps 위젯을 구성하려면 애플리케이션과 계정을 AWS Systems Manager OpsCenter 활성화하세요. 자세한 내용은 Systems Manager 탐색기 시작하기 및 OpsCenter AWS Systems Manager 사용 설명서를 참조하십시오. OpsCenter 활성화하면 일반적으로 사용되는 규칙 AWS Config 및 Amazon CloudWatch 이벤트를 OpsItems 기반으로 이벤트가 자동으로 AWS Systems Manager Explorer 생성되도록 구성할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 설정을 OpsCenter 참조하십시오.

Systems Manager 에이전트가 실행되도록 인스턴스를 구성하고 권한을 적용하여 패치 스캔을 활성화할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager 빠른 설정을 참조하세요.

또한 Patch Manager를 설정하여 애플리케이션에 대한 Amazon EC2 인스턴스의 자동 패치를 AWS Systems Manager 설정할 수 있습니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 빠른 설정 패치 정책 사용을 참조하세요.

요금 정보는 AWS Systems Manager 요금을 참조하세요.

# 모니터링 및 운영 위젯

이 위젯은 다음을 보여줍니다.

- 애플리케이션과 관련된 리소스에 대한 경보 및 알림
- 애플리케이션 서비스 수준 목표(SLO) 및 지표
- 사용 가능한 AWS 애플리케이션 신호 지표

## 모니터링 및 운영 위젯 구성

모니터링 및 운영 위젯을 구성하려면 계정에서 CloudWatch 경보와 카나리아를 생성하십시오. AWS 자세한 내용은 Amazon 사용  CloudWatch 설명서의 Amazon CloudWatch  알람 사용 및 카나리아 생성을 참조하십시오. CloudWatch 알람 요금과 가상 카나리아 요금에 대해서는 각각 Amazon CloudWatch 요금과AWS 클라우드 운영 및 마이그레이션 블로그를 참조하십시오.

CloudWatch 애플리케이션 신호에 대한 자세한 내용은 Amazon 사용  CloudWatch 설명서의 Amazon CloudWatch 애플리케이션 인사이트 활성화를 참조하십시오.

## 태그 위젯

이 위젯은 애플리케이션과 관련된 모든 태그를 표시합니다. 이 위젯을 사용하여 애플리케이션 메타데이터(중요도, 환경, 비용 센터)를 추적하고 관리할 수 있습니다. 자세한 내용은 태그란 무엇입니까? 를 참조하십시오.  AWS 리소스 태깅 모범 사례 AWS 백서에서

# AWS Management Console 프라이빗 액세스

AWS Management Console 프라이빗 액세스는 에 대한 액세스를 제어하는 고급 보안 기능입니다. AWS Management Console AWS Management Console 사용자가 네트워크 내에서 예기치 AWS 계정 않게 로그인하는 것을 방지하려는 경우 사설 액세스가 유용합니다. 이 기능을 사용하면 네트워크 내에서 트래픽이 AWS 계정 언제 시작되는지 알려진 특정 AWS Management Console 집합에만 액세스를 제한할 수 있습니다.

주제



## 지원 AWS 리전, 서비스 콘솔 및 기능

AWS Management Console 프라이빗 액세스는 일부 지역 및 서비스만 지원합니다. AWS 지원되지 않는 서비스 콘솔은 AWS Management Console에서 비활성화됩니다. 또한 AWS Management Console 프라이빗 액세스를 사용할 때는 통합 설정의 기본 지역 선택과 같은 특정 AWS Management Console 기능이 비활성화될 수 있습니다.

지원되는 리전 및 서비스 콘솔은 아래와 같습니다.

지원되는 리전

- 미국 동부(오하이오)
- 미국 동부(버지니아 북부)
- 미국 서부(캘리포니아 북부)
- 미국 서부(오레곤)
- 아시아 태평양(하이데라바드)
- 아시아 태평양(뭄바이)

- 아시아 태평양(서울)
- 아시아 태평양(오사카)
- 아시아 태평양(싱가포르)
- 아시아 태평양(시드니)
- 아시아 태평양(도쿄)
- 캐나다(중부)
- 유럽(프랑크푸르트)
- 유럽(아일랜드)
- 유럽(런던)
- 유럽(파리)
- 유럽(스톡홀름)
- 남아메리카(상파울루)
- 아프리카(케이프타운)
- 아시아 태평양(홍콩)
- 아시아 태평양(자카르타)
- 아시아 태평양(멜버른)
- 캐나다 서부(캘거리)
- 유럽(밀라노)
- 유럽(스페인)
- 유럽(취리히)
- 중동(바레인)
- 중동(UAE)
- 이스라엘(텔아비브)

### 지원되는 서비스 콘솔

- Amazon API Gateway
- AWS App Mesh
- AWS Application Migration Service
- Amazon Athena

- AWS Auto Scaling
- AWS Billing Conductor
- AWS Certificate Manager
- AWS Cloud Map
- 아마존 CloudFront
- 아마존 CloudWatch
- AWS CodeArtifact
- AWS CodeBuild
- 아마존 CodeGuru
- Amazon Comprehend
- Amazon Comprehend Medical
- AWS Compute Optimizer
- AWS Console Home
- AWS Database Migration Service
- AWS DeepRacer
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon EC2
- Amazon EC2 Global View
- EC2 Image Builder
- Amazon EC2 Instance Connect
- Amazon Elastic 컨테이너 레지스트리
- Amazon Elastic Container Service
- AWS Elastic Disaster Recovery
- Amazon Elastic File System
- Amazon Elastic Kubernetes 서비스
- 아마존 ElastiCache
- Amazon EMR
- 아마존 EventBridge
- 아마존 GameLift

- AWS Global Accelerator
- AWS Glue DataBrew
- AWS Ground Station
- 아마존 GuardDuty
- AWS Identity and Access Management
- AWS Identity and Access Management Access Analyzer
- Amazon Inspector
- Amazon Kendra
- AWS Key Management Service
- Amazon Kinesis
- Amazon Managed Service for Apache Flink
- Amazon Data Firehose
- Amazon Kinesis Video Streams
- AWS Lambda
- Amazon Lex
- AWS License Manager
- Amazon Managed Grafana
- Amazon Managed Streaming for Apache Kafka
- Amazon Managed Workflows for Apache Airflow(MWAA)
- AWS Migration Hub 전략 권장 사항
- Amazon MQ
- Network Access Analyzer
- AWS Network Manager
- 아마존 OpenSearch 서비스
- AWS Organizations
- Outposts에서의 Amazon S3
- 아마존 SageMaker 런타임
- 아마존 SageMaker 합성 데이터
- AWS Secrets Manager
- Service Quotas

- AWS Signer
- Amazon Simple Email Service
- Amazon Simple Queue Service
- Amazon Simple Storage Service(S3)
- AWS SQL Workbench
- AWS Step Functions
- AWS Support
- AWS Systems Manager
- AWS Transfer Family
- 통합 설정
- Amazon VPC IP 주소 관리자

# AWS Management Console 프라이빗 액세스 보안 제어 개요

## 네트워크에서 AWS Management Console 에 대한 계정 제한

AWS Management Console Private Access는 네트워크에서의 액세스를 조직에 알려진 AWS 계정 특정 집합으로만 제한하려는 경우에 유용합니다. AWS Management Console 이렇게 하면 사용자가 네트워크 내에서 예기치 못한 AWS 계정 에 로그인하는 것을 방지할 수 있습니다. AWS Management Console VPC 엔드포인트 정책을 사용하여 이러한 제어를 구현할 수 있습니다. 자세한 정보는 서비스 제어 정책 및 VPC 엔드포인트 정책 구현을 참조하세요.

## 네트워크에서 인터넷으로 연결

정적 콘텐츠 (, CSS AWS Management Console JavaScript, 이미지) 와 같이 에서 사용하는 자산에 액세스하려면 네트워크를 통한 인터넷 연결이 여전히 필요하며 이를 통해 AWS 서비스 활성화되지 않은 모든 자산에 액세스할 수 AWS PrivateLink있습니다. 에서 사용하는 최상위 도메인 목록은 AWS Management Console을 참조하십시오문제 해결.

> **Note**
>
> 현재 AWS Management Console 프라이빗 액세스는, `status.aws.amazon.comhealth.aws.amazon.com`, 등의 엔드포인트를 지원하지 않습니다. `docs.aws.amazon.com` 이러한 도메인을 공용 인터넷으로 라우팅해야 합니다.

# 필수 VPC 엔드포인트 및 DNS 구성

AWS Management Console 프라이빗 액세스에는 지역당 다음과 같은 두 개의 VPC 엔드포인트가 필요합니다. *region*을 현재의 해당하는 리전 정보로 바꿉니다.

1. com.amazonaws. *## ## ##* AWS Management Console
2. com.amazonaws. *##. ### ##* AWS 로그인

> **Note**
>
> AWS Management Console과 함께 사용하는 기타 리전과 상관없이, 인프라 및 네트워킹 연결을 항상 미국 동부(버지니아 북부)(us-east-1) 리전으로 프로비저닝합니다. AWS Transit Gateway 를 사용하여 미국 동부(버지니아 북부) 리전과 다른 모든 리전 간의 연결을 설정할 수 있습니다. 자세한 내용은 Amazon VPC Transit Gateway 가이드의 전송 게이트웨이 시작하기를 참조하세요. Amazon VPC 피어링도 사용할 수 있습니다. 자세한 내용은 Amazon VPC Peering Guide의 VPC 피어링이란?을 참조하세요. 이러한 옵션을 비교하려면 Amazon Virtual Private Cloud(VPC) 연결 옵션 백서에서 Amazon VPC 간 연결 옵션을 참조하세요.

## DNS및 에 대한 구성 AWS Management ConsoleAWS 로그인

네트워크 트래픽을 각 VPC 엔드포인트로 라우팅하려면 사용자가 AWS Management Console에 액세스할 네트워크에서 DNS 레코드를 구성하세요. 이러한 DNS 레코드는 사용자의 브라우저 트래픽을 생성된 VPC 엔드포인트로 이동합니다.

단일 호스팅 영역을 생성할 수 있습니다. 그러나 `health.aws.amazon.com` 및 `docs.aws.amazon.com`과 같은 엔드포인트는 VPC 엔드포인트가 없으므로 액세스할 수 없습니다. 이러한 도메인을 공용 인터넷으로 라우팅해야 합니다. 다음 CNAME 레코드를 사용하여 리전별로 `signin.aws.amazon.com`용과 `console.aws.amazon.com`용으로 각각 하나씩 두 개의 프라이빗 호스팅 영역을 생성하는 것이 좋습니다.

- 리전 CNAME 레코드(모든 리전)
- region.signin.aws.amazon.com은 로그인 영역의 VPC 엔드포인트를 가리킵니다. AWS 로그인 DNS
- region.console.aws.amazon.com은 콘솔 영역의 VPC 엔드포인트를 가리킵니다. AWS Management Console DNS
- 리전이 없는 CNAME 레코드는 미국 동부(버지니아 북부) 리전에서만 사용할 수 있습니다. 항상 미국 동부(버지니아 북부) 리전을 설정해야 합니다.

- signin.aws.amazon.com은 미국 동부 (버지니아 북부) 의 AWS 로그인 VPC 엔드포인트를 가리키고 있습니다 (us-east-1)
- 미국 동부 (버지니아 북부) 의 AWS Management Console VPC 엔드포인트를 가리키는 console.aws.amazon.com (us-east-1)

CNAME 레코드 생성에 대한 지침은 Amazon Route 53 개발자 안내서의 레코드 작업을 참조하세요.

Amazon S3를 비롯한 일부 AWS 콘솔은 이름에 다른 패턴을 사용합니다. DNS 다음은 두 가지 예제입니다.

- support.console.aws.amazon.com
- s3.console.aws.amazon.com

이 트래픽을 AWS Management Console VPC 엔드포인트로 보내려면 해당 이름을 개별적으로 추가해야 합니다. 완전한 프라이빗 환경을 위해 모든 엔드포인트에 대해 라우팅을 구성하는 것이 좋습니다. 하지만 AWS Management Console 프라이빗 액세스를 사용하는 데 반드시 필요한 것은 아닙니다.

다음 `json` 파일에는 지역별로 구성할 콘솔 엔드포인트 및 콘솔 엔드포인트의 전체 목록이 포함되어 있습니다. AWS 서비스DNS 이름의 `com.amazonaws.`*`region`*`.console` 엔드포인트 아래에 있는 `PrivateIpv4DnsNames` 필드를 사용하세요.

- https://configuration.private-access.console.amazonaws.com/us-east-1.config.json
- https://configuration.private-access.console.amazonaws.com/us-east-2.config.json
- https://configuration.private-access.console.amazonaws.com/us-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-2.config.json
- https://configuration.private-access.console.amazonaws.com/ca-central-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/il-central-1.config.json

> **Note**
>
> AWS Management Console 프라이빗 액세스 범위에 엔드포인트가 수시로 추가되므로 이 목록은 매달 업데이트됩니다. 프라이빗 호스팅 영역을 최신 상태로 유지하려면 이전 파일 목록을 주기적으로 가져오세요.

Route 53을 사용하여 DNS를 구성할 경우 https://console.aws.amazon.com/route53/v2/hostedzones# 로 이동하여 DNS 설정을 확인하세요. Route 53의 각 프라이빗 호스팅 영역에 대해 다음과 같은 레코드 세트가 있는지 확인합니다.

- console.aws.amazon.com
- signin.aws.amazon.com
- region.console.aws.amazon.com
- region.signin.aws.amazon.com
- support.console.aws.amazon.com
- global.console.aws.amazon.com
- 이전에 목록에 등록된 JSON 파일에 있는 추가 레코드

## VPC 엔드포인트 및 DNS 서비스 구성 AWS

직접 브라우저 요청과 웹 서버에서 프록시되는 요청의 조합을 AWS 서비스 통한 AWS Management Console 호출. 이 트래픽을 VPC 엔드포인트로 보내려면 AWS Management Console VPC 엔드포인트를 추가하고 각 종속 서비스에 DNS 대해 구성해야 합니다. AWS

다음 json 파일에는 사용 가능한 AWS PrivateLink 지원 항목이 AWS 서비스 나열되어 있습니다. 서비스가 통합되지 않는 경우 해당 서비스는 이러한 파일에 포함되지 않습니다. AWS PrivateLink

- https://configuration.private-access.console.amazonaws.com/us-east-1.config.json
- https://configuration.private-access.console.amazonaws.com/us-east-2.config.json
- https://configuration.private-access.console.amazonaws.com/us-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-2.config.json

- https://configuration.private-access.console.amazonaws.com/ap-south-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-2.config.json
- https://configuration.private-access.console.amazonaws.com/ca-central-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/il-central-1.config.json

해당하는 서비스의 VPC 엔드포인트에 대한 `ServiceName` 필드를 사용하여 VPC에 추가합니다.

> **Note**
>
> AWS Management Console Private Access에 대한 지원이 더 많은 서비스 콘솔에 추가됨에 따라 매달 이 목록을 업데이트합니다. 최신 상태를 유지하려면 이전 파일 목록을 주기적으로 가져와서 VPC 엔드포인트를 업데이트하세요.

# 서비스 제어 정책 및 VPC 엔드포인트 정책 구현

프라이빗 액세스를 AWS Management Console 위한 서비스 제어 정책 (SCP) 및 VPC 엔드포인트 정책을 사용하여 VPC 및 연결된 온프레미스 네트워크 내에서 사용할 AWS Management Console 수 있는 계정 세트를 제한할 수 있습니다.

## 서비스 제어 정책과 함께 AWS Management Console 프라이빗 액세스 사용 AWS Organizations

AWS 조직에서 특정 서비스를 허용하는 서비스 제어 정책 (SCP) 을 사용하는 경우 허용된 작업에 `signin:*` 추가해야 합니다. Private Access VPC 엔드포인트를 AWS Management Console 통해 로그인하면 권한 없이 SCP가 차단하는 IAM 인증이 수행되기 때문에 이 권한이 필요합니다. 예를 들어, 다음 서비스 제어 정책은 AWS Management Console 프라이빗 액세스 엔드포인트를 사용하여 액세스하는 경우를 포함하여 조직에서 Amazon EC2 및 CloudWatch 서비스를 사용할 수 있도록 허용합니다.

```
{
  "Effect": "Allow",
  "Action": [
    "signin:*",
    "ec2:*",
```

```
    "cloudwatch:*",
    ... Other services allowed
  },
  "Resource": "*"
}
```

SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서에서 서비스 제어 정책(SCP)을 참조하세요.

## 예상 계정 및 조직에만 AWS Management Console 사용 허용 (신뢰할 수 있는 ID)

AWS Management Console 로그인한 계정의 ID를 구체적으로 제어하는 VPC 엔드포인트 정책을 AWS 로그인 지원합니다.

다른 VPC 엔드포인트 정책과 달리, 이 정책은 인증 전에 평가됩니다. 따라서 인증된 세션의 로그인 및 사용만 특별히 제어하고 세션에서 수행하는 AWS 서비스별 작업은 제어하지 않습니다. 예를 들어 세션이 Amazon EC2 콘솔과 같은 AWS 서비스 콘솔에 액세스하는 경우 이러한 VPC 엔드포인트 정책은 해당 페이지를 표시하기 위해 취해진 Amazon EC2 작업에 대해 평가되지 않습니다. 대신 로그인한 IAM 주체와 연결된 IAM 정책을 사용하여 서비스 작업에 대한 권한을 제어할 수 있습니다. AWS

> **Note**
>
> 및 VPC 엔드포인트에 대한 AWS Management Console SignIn VPC 엔드포인트 정책은 정책 공식의 제한된 하위 집합만 지원합니다. 모든 `Principal` 및 `Resource`는 `*`로 설정해야 하며 `Action`은 `*` 또는 `signin:*` 중 하나여야 합니다. `aws:PrincipalOrgId` 및 `aws:PrincipalAccount` 조건 키를 사용하여 VPC 엔드포인트에 대한 액세스를 제어할 수 있습니다.

다음 정책은 콘솔 엔드포인트와 SignIn VPC 엔드포인트 모두에 권장됩니다.

이 VPC 엔드포인트 정책은 지정된 AWS 계정 AWS 조직에서의 로그인을 허용하고 다른 계정으로의 로그인을 차단합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgId": "o-xxxxxxxxxxx"
        }
      }
    }
  ]
}
```

이 VPC 엔드포인트 정책은 로그인을 특정 목록으로 AWS 계정 제한하고 다른 계정으로의 로그인을 차단합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalAccount": [ "111122223333", "222233334444" ]
        }
      }
    }
  ]
}
```

로그인 VPC 엔드포인트 AWS Management Console 및 로그인 VPC 엔드포인트에서 조직을 제한하는 AWS 계정 정책은 로그인 시 평가되며 기존 세션에 대해 정기적으로 재평가됩니다.

# 자격 증명 기반 정책 및 기타 정책 유형 구현

정책을 만들고 이를 IAM ID (사용자, 사용자 그룹 또는 역할) 또는 리소스에 연결하여 액세스를 관리합니다. AWS AWS 이 페이지에서는 정책을 Private Access와 AWS Management Console 함께 사용할 경우 어떻게 작동하는지 설명합니다.

# 지원되는 AWS 글로벌 조건 컨텍스트 키

AWS Management Console 프라이빗 액세스는 `aws:VpcSourceIp` AWS 글로벌 조건 컨텍스트 키를 지원하지 `aws:SourceVpce` 않습니다. AWS Management Console 프라이빗 액세스를 사용할 경우 정책에서 `aws:SourceVpc` IAM 조건을 대신 사용할 수 있습니다.

## AWS Management Console 프라이빗 액세스가 AWS와 함께 작동하는 방식: SourceVpc

이 섹션에서는 사용자가 생성한 요청이 AWS Management Console 도달할 수 있는 다양한 네트워크 경로를 설명합니다 AWS 서비스. 일반적으로 AWS 서비스 콘솔은 직접 브라우저 요청과 웹 서버에서 프록시되는 요청을 혼합하여 구현됩니다. AWS Management Console AWS 서비스이러한 구현은 사전 고지 없이 변경될 수 있습니다. 보안 요구 사항에 VPC 엔드포인트 AWS 서비스 사용에 대한 액세스가 포함되는 경우 직접 또는 프라이빗 액세스를 통해 VPC에서 사용하려는 모든 서비스에 대해 VPC 엔드포인트를 구성하는 것이 좋습니다. AWS Management Console 또한 정책에서는 프라이빗 액세스 기능의 특정 `aws:SourceVpce` 값 대신 `aws:SourceVpc` IAM 조건을 사용해야 합니다. AWS Management Console 이 섹션에서는 다양한 네트워크 경로의 작동 방식에 대한 세부 정보를 제공합니다.

사용자는 에 로그인한 후 직접 브라우저 요청과 AWS Management Console 웹 서버가 서버로 프록시하는 요청을 조합하여 요청합니다. AWS Management Console AWS 서비스 AWS 예를 들어, CloudWatch 그래프 데이터 요청은 브라우저에서 직접 이루어집니다. 반면 Amazon S3와 같은 일부 AWS 서비스 콘솔 요청은 웹 서버에서 Amazon S3로 프록시됩니다.

직접 브라우저 요청의 경우, AWS Management Console 프라이빗 액세스를 사용해도 아무 것도 변경되지 않습니다. 이전과 마찬가지로, 요청은 VPC가 monitoring.region.amazonaws.com 에 도달하도록 구성한 네트워크 경로를 통해 서비스에 도달합니다. VPC가 에 대한 com.amazonaws.region.monitoring VPC 엔드포인트로 구성된 경우 요청은 CloudWatch 해당 VPC 엔드포인트를 통해 전달됩니다. CloudWatch 에 대한 CloudWatch VPC 엔드포인트가 없는 경우 요청은 VPC의 Internet Gateway를 통해 퍼블릭 엔드포인트에 CloudWatch 도달합니다. CloudWatch VPC 엔드포인트를 CloudWatch 통해 들어오는 요청에는 IAM 조건이 `aws:SourceVpc` 적용되며 해당 값으로 `aws:SourceVpce` 설정됩니다. 퍼블릭 엔드포인트를 CloudWatch 통해 도달하는 요청은 요청의 소스 IP 주소로 `aws:SourceIp` 설정됩니다. IAM 조건 키에 대한 자세한 정보는 IAM 사용 설명서의 전역 조건 키를 참조하세요.

Amazon S3 콘솔을 방문할 때 Amazon S3 콘솔이 버킷을 나열하도록 요청하는 것과 같이 AWS Management Console 웹 서버에서 프록시되는 요청의 경우 네트워크 경로가 다릅니다. 이러한 요청은 VPC에서 시작되지 않으므로 해당 서비스에 대해 VPC에 구성된 VPC 엔드포인트를 사용하지 않습

니다. 이 경우 Amazon S3에 대한 VPC 엔드포인트가 있더라도, 버킷 목록을 나열해달라는 Amazon S3에 대한 세션의 요청은 Amazon S3 VPC 엔드포인트를 사용하지 않습니다. 하지만 지원되는 서비스와 함께 AWS Management Console 프라이빗 액세스를 사용하는 경우 이러한 요청 (예: Amazon S3) 에는 요청 컨텍스트에 `aws:SourceVpc` 조건 키가 포함됩니다. `aws:SourceVpc`조건 키는 로그인 및 콘솔용 AWS Management Console 프라이빗 액세스 엔드포인트가 배포되는 VPC ID로 설정됩니다. 따라서 자격 증명 기반 정책에서 `aws:SourceVpc` 제한을 사용할 경우, AWS Management Console 프라이빗 액세스 로그인과 콘솔 엔드포인트를 호스팅하는 이 VPC의 VPC ID를 추가해야 합니다. `aws:SourceVpce` 조건은 각각의 로그인 또는 콘솔 VPC 엔드포인트 ID로 설정됩니다.

> **Note**
>
> 사용자가 AWS Management Console 프라이빗 액세스에서 지원되지 않는 서비스 콘솔에 액세스해야 하는 경우, `aws:SourceIP` 조건 키를 사용하여 사용자의 ID 기반 정책에 예상 퍼블릭 네트워크 주소(예: 온프레미스 네트워크 범위) 목록을 포함해야 합니다.

## 다양한 네트워크 경로가 반영되는 방식 CloudTrail

사용자가 생성한 요청에 사용된 다양한 네트워크 경로가 CloudTrail 이벤트 기록에 AWS Management Console 반영됩니다.

직접 브라우저 요청의 경우, AWS Management Console 프라이빗 액세스를 사용해도 아무 것도 변경되지 않습니다. CloudTrail 이벤트에는 서비스 API 호출에 사용된 VPC 엔드포인트 ID와 같은 연결에 대한 세부 정보가 포함됩니다.

AWS Management Console 웹 서버에 의해 프록시되는 요청의 경우, CloudTrail 이벤트에 VPC 관련 세부 정보가 포함되지 않습니다. 하지만 브라우저 세션을 설정하는 데 필요한 초기 요청 (예: `AwsConsoleSignIn` 이벤트 유형) 에는 이벤트 세부 정보에 AWS 로그인 VPC 엔드포인트 ID가 포함됩니다. AWS 로그인

# AWS Management Console 프라이빗 액세스를 사용해 보세요.

이 섹션에서는 새 계정에서 AWS Management Console 프라이빗 액세스를 설정하고 테스트하는 방법을 설명합니다.

AWS Management Console 프라이빗 액세스는 고급 보안 기능이며 네트워킹 및 VPC 설정에 대한 사전 지식이 필요합니다. 이번 주제에서는 대규모 인프라 없이 AWS Management Console 프라이빗 액세스를 사용해 볼 수 있는 방법을 설명합니다.

주제

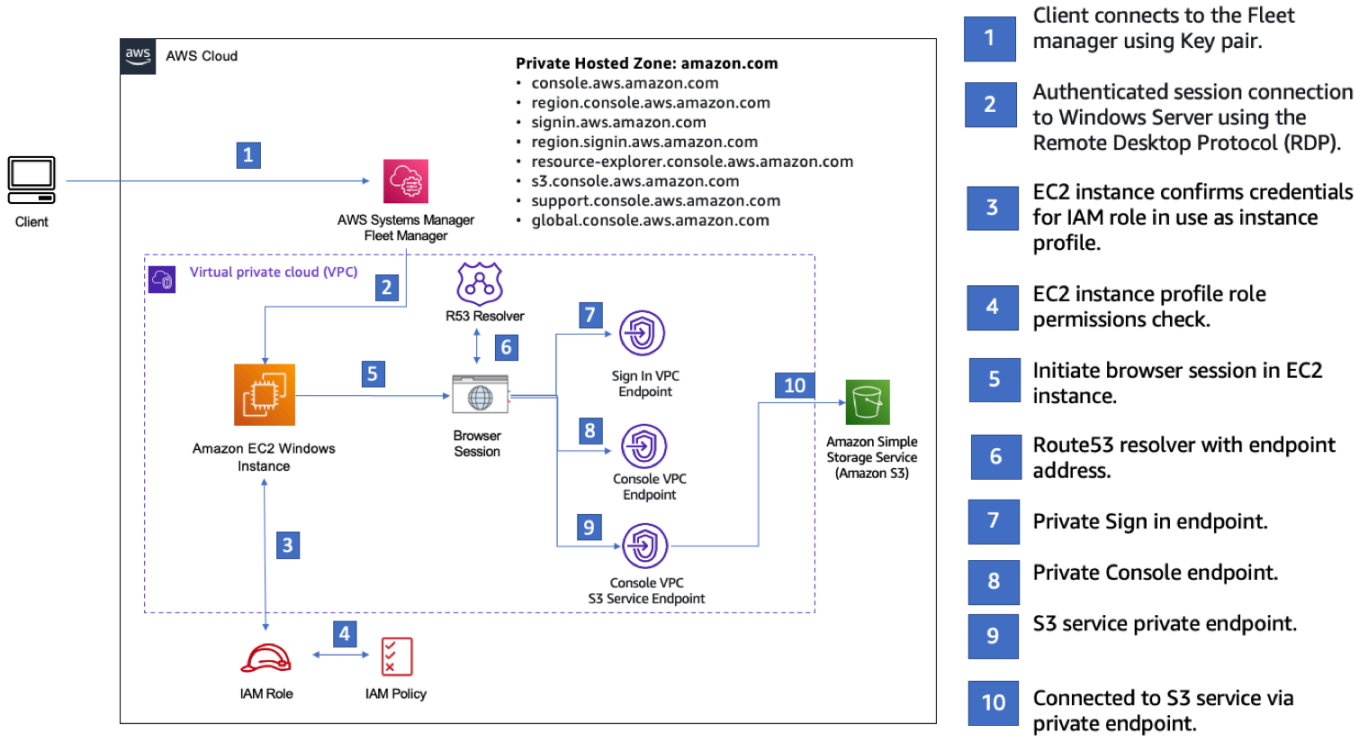


# Amazon EC2를 사용한 테스트 설정

Amazon Elastic Compute Cloud(Amazon EC2)는 Amazon Web Services 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2를 사용하여 원하는 수의 가상 서버를 구축하고 보안 및 네트워킹을 구성하며 스토리지를 관리할 수 있습니다. 이 설정에서는 AWS Systems Manager의 기능인 Fleet Manager를 사용하여 원격 데스크톱 프로토콜(RDP)을 사용하는 Amazon EC2 Windows 인스턴스에 연결합니다.

이 안내서는 Amazon EC2 인스턴스에서 Amazon 심플 스토리지 서비스에 대한 AWS Management Console 프라이빗 액세스 연결을 설정하고 체험하기 위한 테스트 환경을 보여줍니다. 이 자습서에서는 Amazon EC2에서 이 기능을 시각화하는 데 사용할 네트워크 설정을 생성하고 구성하는 데 사용합니다 AWS CloudFormation .

아래의 다이어그램은 Amazon EC2를 사용하여 AWS Management Console 프라이빗 액세스 설정에 액세스하는 워크플로를 설명합니다. 여기에서는 사용자가 프라이빗 엔드포인트를 사용하여 Amazon S3에 연결하는 방법을 보여줍니다.




다음 AWS CloudFormation 템플릿을 복사하여 네트워크 설정 절차의 3단계에서 사용할 파일에 저장합니다.

> **Note**
>
> 이 AWS CloudFormation 템플릿은 현재 이스라엘 (텔아비브) 지역에서 지원되지 않는 구성을 사용합니다.

AWS Management Console 프라이빗 액세스 환경 Amazon EC2 템플릿 AWS CloudFormation

```
Description: |
  AWS Management Console Private Access.
Parameters:
  VpcCIDR:
    Type: String
    Default: 172.16.0.0/16
    Description: CIDR range for VPC

  Ec2KeyPair:
    Type: AWS::EC2::KeyPair::KeyName
```

```
    Description: The EC2 KeyPair to use to connect to the Windows instance

  PublicSubnet1CIDR:
    Type: String
    Default: 172.16.1.0/24
    Description: CIDR range for Public Subnet A

  PublicSubnet2CIDR:
    Type: String
    Default: 172.16.0.0/24
    Description: CIDR range for Public Subnet B

  PublicSubnet3CIDR:
    Type: String
    Default: 172.16.2.0/24
    Description: CIDR range for Public Subnet C

  PrivateSubnet1CIDR:
    Type: String
    Default: 172.16.4.0/24
    Description: CIDR range for Private Subnet A

  PrivateSubnet2CIDR:
    Type: String
    Default: 172.16.5.0/24
    Description: CIDR range for Private Subnet B

  PrivateSubnet3CIDR:
    Type: String
    Default: 172.16.3.0/24
    Description: CIDR range for Private Subnet C

  LatestWindowsAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base'

  InstanceTypeParameter:
    Type: String
    Default: 't2.medium'


Resources:

##########################
```

```
# VPC AND SUBNETS
#########################

  AppVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: !Ref VpcCIDR
      InstanceTenancy: default
      EnableDnsSupport: true
      EnableDnsHostnames: true

  PublicSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ""

  PublicSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ""

  PublicSubnetC:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet3CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone:
        Fn::Select:
          - 2
          - Fn::GetAZs: ""
```

```
  PrivateSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet1CIDR
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ""

  PrivateSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet2CIDR
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ""

  PrivateSubnetC:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet3CIDR
      AvailabilityZone:
        Fn::Select:
          - 2
          - Fn::GetAZs: ""

  InternetGateway:
    Type: AWS::EC2::InternetGateway

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref AppVPC

  NatGatewayEIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment

  NatGateway:
```

```
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGatewayEIP.AllocationId
      SubnetId: !Ref PublicSubnetA

########################
# Route Tables
########################

  PrivateRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref AppVPC

  DefaultPrivateRoute:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway

  PrivateSubnetRouteTableAssociation1:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetA

  PrivateSubnetRouteTableAssociation2:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetB

  PrivateSubnetRouteTableAssociation3:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetC

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref AppVPC
```

```
  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnetARouteTableAssociation1:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetA

  PublicSubnetBRouteTableAssociation2:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetB

  PublicSubnetBRouteTableAssociation3:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetC


##########################
# SECURITY GROUPS
##########################

  VPCEndpointSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Allow TLS for VPC Endpoint
      VpcId: !Ref AppVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: !GetAtt AppVPC.CidrBlock

  EC2SecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
```

```
    Properties:
      GroupDescription: Default EC2 Instance SG
      VpcId: !Ref AppVPC

#########################
# VPC ENDPOINTS
#########################

  VPCEndpointGatewayS3:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.s3'
      VpcEndpointType: Gateway
      VpcId: !Ref AppVPC
      RouteTableIds:
        - !Ref PrivateRouteTable

  VPCEndpointInterfaceSSM:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.ssm'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceEc2messages:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.ec2messages'
      VpcId: !Ref AppVPC
```

```
  VPCEndpointInterfaceSsmmessages:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.ssmmessages'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceSignin:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.signin'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceConsole:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.console'
      VpcId: !Ref AppVPC

##########################
# ROUTE53 RESOURCES
```

```
#########################

  ConsoleHostedZone:
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Console VPC Endpoint Hosted Zone'
      Name: 'console.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  ConsoleRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  GlobalConsoleRecord:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'global.console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleS3ProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 's3.console.aws.amazon.com'
      AliasTarget:
```

```
       DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
       HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
     Type: A

 ConsoleSupportProxyRecordGlobal:
   Type: AWS::Route53::RecordSet
   Properties:
     HostedZoneId: !Ref 'ConsoleHostedZone'
     Name: "support.console.aws.amazon.com"
     AliasTarget:
       DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
       HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
     Type: A

 ExplorerProxyRecordGlobal:
   Type: AWS::Route53::RecordSet
   Properties:
     HostedZoneId: !Ref 'ConsoleHostedZone'
     Name: "resource-explorer.console.aws.amazon.com"
     AliasTarget:
       DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
       HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
     Type: A

 ConsoleRecordRegional:
   Type: AWS::Route53::RecordSet
   Properties:
     HostedZoneId: !Ref 'ConsoleHostedZone'
     Name: !Sub "${AWS::Region}.console.aws.amazon.com"
     AliasTarget:
       DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
       HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
     Type: A

 SigninHostedZone:
   Type: "AWS::Route53::HostedZone"
```

```
    Properties:
      HostedZoneConfig:
        Comment: 'Signin VPC Endpoint Hosted Zone'
      Name: 'signin.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  SigninRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: 'signin.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

  SigninRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: !Sub "${AWS::Region}.signin.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

#########################
# EC2 INSTANCE
#########################

  Ec2InstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          -
```

```
            Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore

  Ec2InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: /
      Roles:
       - !Ref Ec2InstanceRole

  EC2WinInstance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: !Ref LatestWindowsAmiId
      IamInstanceProfile: !Ref Ec2InstanceProfile
      KeyName: !Ref Ec2KeyPair
      InstanceType:
        Ref: InstanceTypeParameter
      SubnetId: !Ref PrivateSubnetA
      SecurityGroupIds:
        - Ref: EC2SecurityGroup
      BlockDeviceMappings:
        - DeviceName: /dev/sda1
          Ebs:
            VolumeSize: 50
      Tags:
      - Key: "Name"
        Value: "Console VPCE test instance"
```

네트워크를 설정하려면

1. 조직의 관리 계정으로 로그인하고 AWS CloudFormation 콘솔을 엽니다.
2. 스택 생성을 선택합니다.

3. 새 리소스 사용(표준)(With new resources (standard))을 선택합니다. 이전에 만든 AWS CloudFormation 템플릿 파일을 업로드하고 다음을 선택합니다.
4. **PrivateConsoleNetworkForS3** 같은 스택 이름을 입력한 후 다음을 선택합니다.
5. VPC 및 서브넷의 경우, 원하는 IP CIDR 범위를 입력하거나 제공된 기본값을 사용합니다. 기본값을 사용하는 경우 해당 값이 내 기존 VPC 리소스와 겹치지 않는지 확인하세요. AWS 계정
6. Ec2 KeyPair 파라미터의 경우 계정의 기존 Amazon EC2 키 페어에서 하나를 선택합니다. 기존 Amazon EC2 키 페어가 없다면 다음 단계로 진행하기 전에 이를 새로 생성해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2를 사용하여 키 페어 생성을 참조하십시오.
7. 스택 생성을 선택합니다.
8. 스택이 생성된 후 리소스 탭을 선택하여 생성된 리소스를 확인합니다.

Amazon EC2 인스턴스에 연결하려면

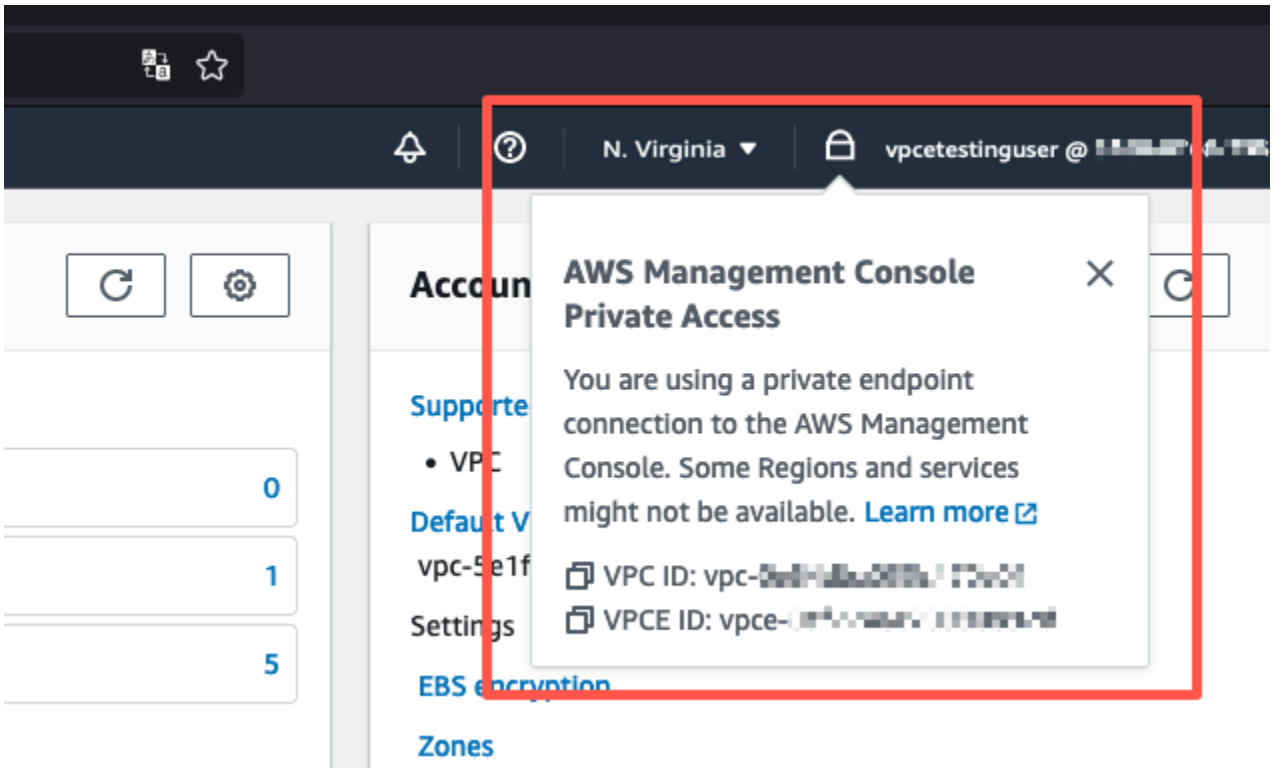
1. 조직의 관리 계정으로 로그인하고 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 인스턴스 페이지에서 템플릿으로 만든 콘솔 VPCE 테스트 인스턴스를 선택합니다. AWS CloudFormation 그런 다음 연결을 선택합니다.

> **Note**
>
> 이 예제에서는 의 AWS Systems Manager Explorer기능인 플릿 관리자를 사용하여 Windows 서버에 연결합니다. 연결이 시작되기까지 몇 분 정도 걸릴 수 있습니다.

4. 인스턴스에 연결 페이지에서 RDP 클라이언트를 선택한 다음, Fleet Manager를 사용하여 연결을 선택합니다.
5. Fleet Manager 원격 데스크톱을 선택합니다.
6. 웹 인터페이스를 사용하여 Amazon EC2 인스턴스의 관리 암호를 얻고 Windows 데스크톱에 액세스하려면 템플릿을 생성할 때 AWS CloudFormation 사용한 Amazon EC2 키 쌍과 연결된 개인 키를 사용하십시오.
7. Amazon EC2 Windows 인스턴스에서 브라우저에서 를 AWS Management Console 엽니다.
8. AWS 자격 증명으로 로그인한 후 Amazon S3 콘솔을 열고 AWS Management Console 프라이빗 액세스를 사용하여 연결되어 있는지 확인합니다.

AWS Management Console 프라이빗 액세스 설정을 테스트하려면

1. 조직의 관리 계정으로 로그인하고 Amazon S3 콘솔을 엽니다.
2. 탐색 메뉴에서 잠금-프라이빗 아이콘을 선택하면 사용 중인 VPC 엔드포인트를 볼 수 있습니다. 아래의 스크린샷은 잠금-프라이빗 아이콘의 위치와 VPC 정보를 보여줍니다.



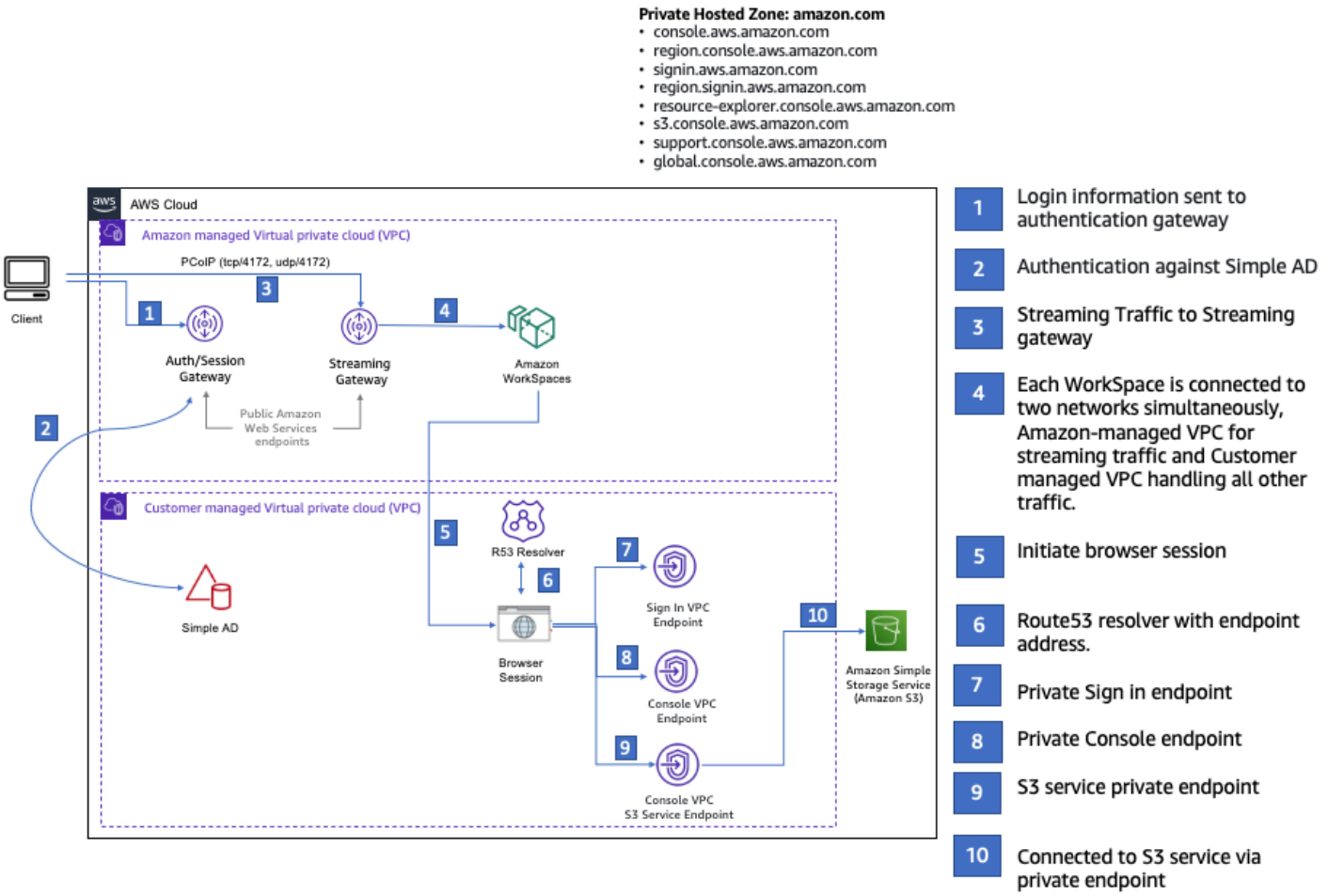

## Amazon을 사용한 테스트 설정 WorkSpaces

Amazon을 WorkSpaces 사용하면 사용자에게 가상 클라우드 기반 Windows, Amazon Linux 또는 Ubuntu Linux 데스크톱 (일명) 을 프로비저닝할 수 있습니다. WorkSpaces 필요에 따라 신속하게 사용자를 추가 또는 제거할 수 있습니다. 사용자는 여러 디바이스 또는 웹 브라우저에서 가상 데스크톱에 액세스할 수 있습니다. 자세히 WorkSpaces 알아보려면 Amazon WorkSpaces 관리 안내서를 참조하십시오.

이 섹션의 예제는 사용자 환경에서 에서 실행되는 웹 브라우저를 사용하여 AWS Management Console Private Access에 WorkSpace 로그인하는 테스트 환경을 설명합니다. 그런 다음 사용자가 Amazon Simple Storage Service 콘솔을 방문합니다. WorkSpace 이는 기업 사용자가 VPC로 연결된 네트워크에서 랩톱을 사용하여 브라우저에서 액세스하는 경험을 시뮬레이션하기 위한 것입니다. AWS Management Console

이 자습서에서는 네트워크 설정 및 사용할 단순 Active Directory를 만들고 구성하는 방법과 WorkSpaces 함께 사용하는 AWS CloudFormation 방법을 단계별로 설정하는 방법을 설명합니다. WorkSpace AWS Management Console

다음 다이어그램은 를 사용하여 AWS Management Console 프라이빗 액세스 설정을 WorkSpace 테스트하는 워크플로를 설명합니다. 클라이언트 WorkSpace, Amazon 관리형 VPC, 고객 관리형 VPC 간의 관계를 보여줍니다.




다음 AWS CloudFormation 템플릿을 복사하여 네트워크 설정 절차의 3단계에서 사용할 파일에 저장합니다.

AWS Management Console 프라이빗 액세스 환경 AWS CloudFormation 템플릿

```
Description: |
  AWS Management Console Private Access.
Parameters:
```

```
  VpcCIDR:
    Type: String
    Default: 172.16.0.0/16
    Description: CIDR range for VPC

  PublicSubnet1CIDR:
    Type: String
    Default: 172.16.1.0/24
    Description: CIDR range for Public Subnet A

  PublicSubnet2CIDR:
    Type: String
    Default: 172.16.0.0/24
    Description: CIDR range for Public Subnet B

  PrivateSubnet1CIDR:
    Type: String
    Default: 172.16.4.0/24
    Description: CIDR range for Private Subnet A

  PrivateSubnet2CIDR:
    Type: String
    Default: 172.16.5.0/24
    Description: CIDR range for Private Subnet B

# Amazon WorkSpaces is available in a subset of the Availability Zones for each
 supported Region.
# https://docs.aws.amazon.com/workspaces/latest/adminguide/azs-workspaces.html
Mappings:
  RegionMap:
    us-east-1:
      az1: use1-az2
      az2: use1-az4
      az3: use1-az6
    us-west-2:
      az1: usw2-az1
      az2: usw2-az2
      az3: usw2-az3
    ap-south-1:
      az1: aps1-az1
      az2: aps1-az2
      az3: aps1-az3
    ap-northeast-2:
      az1: apne2-az1
```

```
      az2: apne2-az3
    ap-southeast-1:
      az1: apse1-az1
      az2: apse1-az2
    ap-southeast-2:
      az1: apse2-az1
      az2: apse2-az3
    ap-northeast-1:
      az1: apne1-az1
      az2: apne1-az4
    ca-central-1:
      az1: cac1-az1
      az2: cac1-az2
    eu-central-1:
      az1: euc1-az2
      az2: euc1-az3
    eu-west-1:
      az1: euw1-az1
      az2: euw1-az2
    eu-west-2:
      az1: euw2-az2
      az2: euw2-az3
    sa-east-1:
      az1: sae1-az1
      az2: sae1-az3

Resources:

  iamLambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
              - lambda.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
      Policies:
        - PolicyName: describe-ec2-az
```

```
        PolicyDocument:
          Version: "2012-10-17"
          Statement:
            - Effect: Allow
              Action:
                - 'ec2:DescribeAvailabilityZones'
              Resource: '*'
    MaxSessionDuration: 3600
    Path: /service-role/

 fnZoneIdtoZoneName:
   Type: AWS::Lambda::Function
   Properties:
     Runtime: python3.8
     Handler: index.lambda_handler
     Code:
       ZipFile: |
         import boto3
         import cfnresponse

         def zoneId_to_zoneName(event, context):
             responseData = {}
             ec2 = boto3.client('ec2')
             describe_az = ec2.describe_availability_zones()
             for az in describe_az['AvailabilityZones']:
                 if event['ResourceProperties']['ZoneId'] == az['ZoneId']:
                     responseData['ZoneName'] = az['ZoneName']
                     cfnresponse.send(event, context, cfnresponse.SUCCESS,
responseData, str(az['ZoneId']))

         def no_op(event, context):
             print(event)
             responseData = {}
             cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
str(event['RequestId']))

         def lambda_handler(event, context):
             if event['RequestType'] == ('Create' or 'Update'):
                 zoneId_to_zoneName(event, context)
             else:
                 no_op(event,context)
     Role: !GetAtt iamLambdaExecutionRole.Arn

 getAZ1:
```

```
    Type: "Custom::zone-id-zone-name"
    Properties:
      ServiceToken: !GetAtt fnZoneIdtoZoneName.Arn
      ZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', az1 ]
  getAZ2:
    Type: "Custom::zone-id-zone-name"
    Properties:
      ServiceToken: !GetAtt fnZoneIdtoZoneName.Arn
      ZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', az2 ]

########################
# VPC AND SUBNETS
########################

  AppVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: !Ref VpcCIDR
      InstanceTenancy: default
      EnableDnsSupport: true
      EnableDnsHostnames: true

  PublicSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone: !GetAtt getAZ1.ZoneName

  PublicSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone: !GetAtt getAZ2.ZoneName

  PrivateSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet1CIDR
      AvailabilityZone: !GetAtt getAZ1.ZoneName
```

```
  PrivateSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet2CIDR
      AvailabilityZone: !GetAtt getAZ2.ZoneName

  InternetGateway:
    Type: AWS::EC2::InternetGateway

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref AppVPC

  NatGatewayEIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment

  NatGateway:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGatewayEIP.AllocationId
      SubnetId: !Ref PublicSubnetA

##########################
# Route Tables
##########################

  PrivateRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref AppVPC

  DefaultPrivateRoute:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway

  PrivateSubnetRouteTableAssociation1:
```

```
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetA

  PrivateSubnetRouteTableAssociation2:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetB

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref AppVPC

  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnetARouteTableAssociation1:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetA

  PublicSubnetBRouteTableAssociation2:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetB


##########################
# SECURITY GROUPS
##########################

  VPCEndpointSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
```

```
      GroupDescription: Allow TLS for VPC Endpoint
      VpcId: !Ref AppVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: !GetAtt AppVPC.CidrBlock

#########################
# VPC ENDPOINTS
#########################

  VPCEndpointGatewayS3:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.s3'
      VpcEndpointType: Gateway
      VpcId: !Ref AppVPC
      RouteTableIds:
        - !Ref PrivateRouteTable

  VPCEndpointInterfaceSignin:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.signin'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceConsole:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
```

```
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.console'
      VpcId: !Ref AppVPC

#########################
# ROUTE53 RESOURCES
#########################

  ConsoleHostedZone:
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Console VPC Endpoint Hosted Zone'
      Name: 'console.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  ConsoleRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  GlobalConsoleRecord:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'global.console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleS3ProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
```

```
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 's3.console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleSupportProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: "support.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ExplorerProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: "resource-explorer.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: !Sub "${AWS::Region}.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
```

```
      Type: A

  SigninHostedZone:
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Signin VPC Endpoint Hosted Zone'
      Name: 'signin.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  SigninRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: 'signin.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

  SigninRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: !Sub "${AWS::Region}.signin.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

##########################
# WORKSPACE RESOURCES
##########################
  ADAdminSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Name: "ADAdminSecret"
```

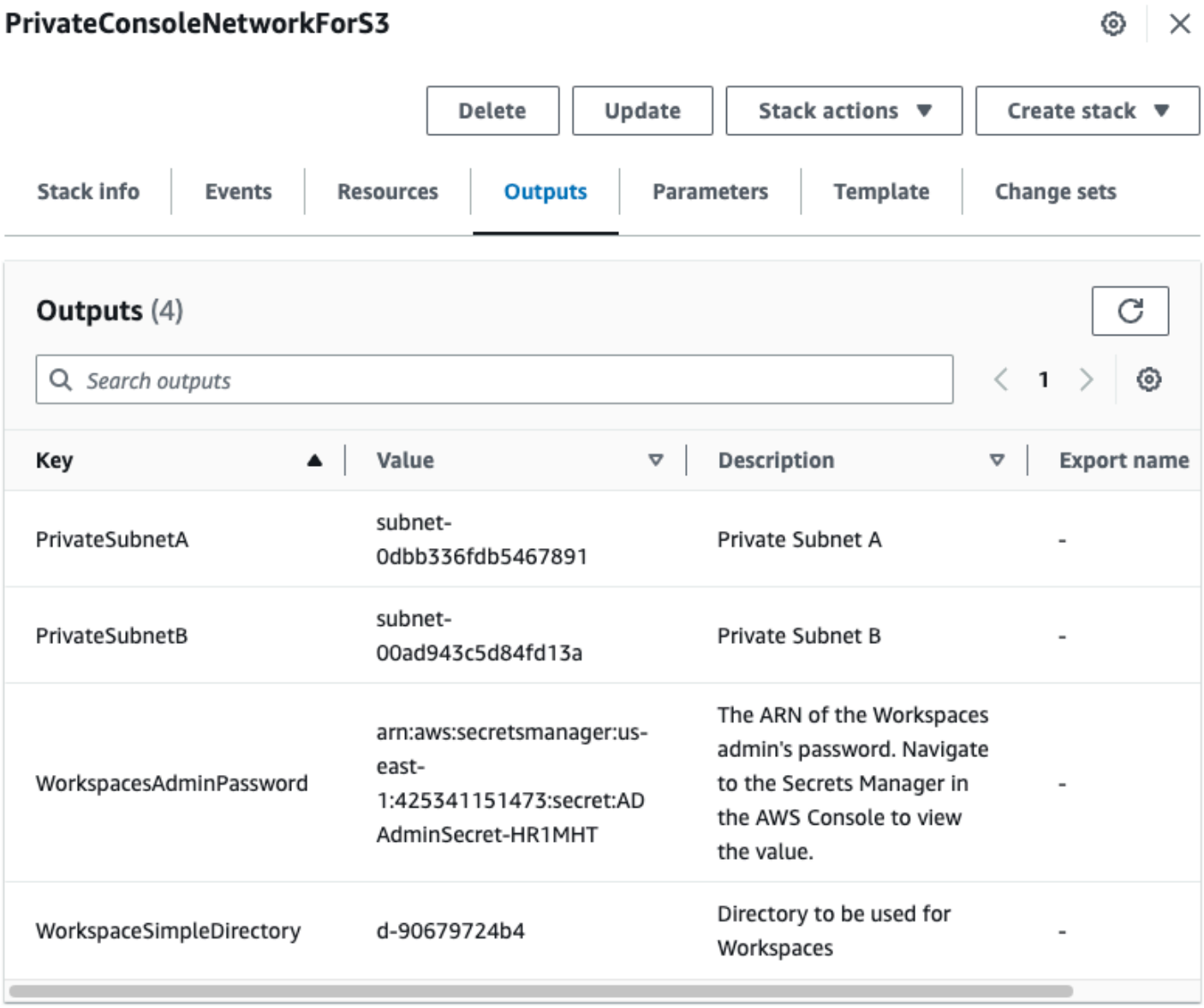
```
      Description: "Password for directory services admin"
      GenerateSecretString:
        SecretStringTemplate: '{"username": "Admin"}'
        GenerateStringKey: password
        PasswordLength: 30
        ExcludeCharacters: '"@/\'

  WorkspaceSimpleDirectory:
    Type: AWS::DirectoryService::SimpleAD
    DependsOn: AppVPC
    DependsOn: PrivateSubnetA
    DependsOn: PrivateSubnetB
    Properties:
      Name: "corp.awsconsole.com"
      Password: '{{resolve:secretsmanager:ADAdminSecret:SecretString:password}}'
      Size: "Small"
      VpcSettings:
        SubnetIds:
          - Ref: PrivateSubnetA
          - Ref: PrivateSubnetB

        VpcId:
          Ref: AppVPC


Outputs:
  PrivateSubnetA:
    Description: Private Subnet A
    Value: !Ref PrivateSubnetA

  PrivateSubnetB:
    Description: Private Subnet B
    Value: !Ref PrivateSubnetB

  WorkspaceSimpleDirectory:
    Description: Directory to be used for Workspaces
    Value: !Ref WorkspaceSimpleDirectory

  WorkspacesAdminPassword:
    Description : "The ARN of the Workspaces admin's password.  Navigate to the Secrets
 Manager in the AWS Console to view the value."
    Value: !Ref ADAdminSecret
```

> **Note**
>
> 이 테스트 설정은 미국 동부(버지니아 북부)(us-east-1) 리전에서 실행되도록 설계되었습니다.

네트워크를 설정하려면

1. 조직의 관리 계정으로 로그인하고 AWS CloudFormation 콘솔을 엽니다.
2. 스택 생성을 선택합니다.
3. 새 리소스 사용(표준)(With new resources (standard))을 선택합니다. 이전에 만든 AWS CloudFormation 템플릿 파일을 업로드하고 다음을 선택합니다.
4. **`PrivateConsoleNetworkForS3`** 같은 스택 이름을 입력한 후 다음을 선택합니다.
5. VPC 및 서브넷의 경우, 원하는 IP CIDR 범위를 입력하거나 제공된 기본값을 사용합니다. 기본값을 사용하는 경우 해당 값이 내 기존 VPC 리소스와 겹치지 않는지 확인하세요. AWS 계정
6. 스택 생성을 선택합니다.
7. 스택이 생성된 후 리소스 탭을 선택하여 생성된 리소스를 확인합니다.
8. 출력 탭을 선택하여 프라이빗 서브넷과 Workspace Simple Directory의 값을 확인합니다. 다음 생성 및 구성 절차의 4단계에서 이 값을 사용하므로 이 값을 기록해 두십시오. WorkSpace

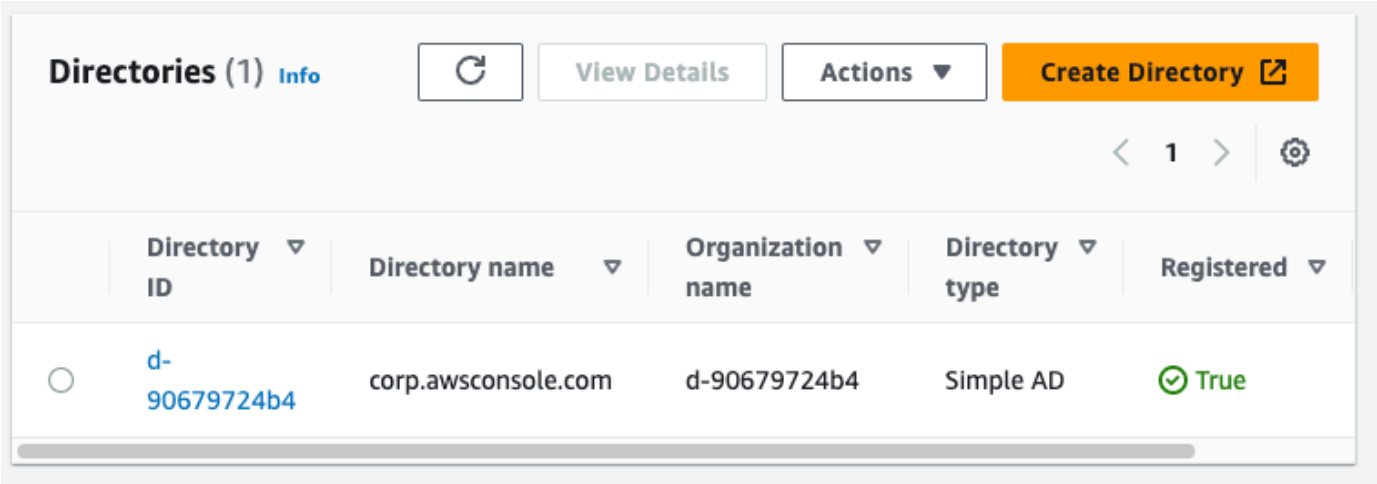
아래의 스크린샷은 프라이빗 서브넷과 Workspace Simple Directory의 값이 표시된 출력 탭의 보기를 보여줍니다.




이제 네트워크를 만들었으니 다음 절차를 사용하여 네트워크를 만들고 액세스하십시오 WorkSpace.

생성하려면 WorkSpace

1. WorkSpaces 콘솔을 엽니다.
2. 탐색 창에서 디렉터리를 선택합니다.
3. 디렉터리 페이지에서 디렉터리 상태가 활성인지 확인합니다. 아래의 스크린샷은 활성 디렉터리가 있는 디렉터리 페이지를 보여줍니다.




4. 에서 WorkSpaces 디렉토리를 사용하려면 디렉토리를 등록해야 합니다. 탐색 창에서 을 선택한 WorkSpaces다음 [Create] 를 선택합니다 WorkSpaces.
5. 디렉터리 선택의 경우, 이전 프로시저에서 AWS CloudFormation 에 의해 생성된 디렉터리를 선택합니다. 작업 메뉴에서 등록을 선택합니다.
6. 서브넷 선택의 경우, 이전 프로시저의 9단계에서 설명한 두 개의 프라이빗 서브넷을 선택합니다.
7. 셀프 서비스 권한 활성화를 선택한 다음 등록을 선택합니다.
8. 디렉토리를 등록한 후 계속해서 디렉토리를 WorkSpace 생성하십시오. 등록된 디렉터리를 선택한 후 다음을 선택합니다.
9. 사용자 생성 페이지에서 추가 사용자 생성을 선택합니다. 을 (를) 사용할 수 있도록 이름과 이메일을 입력합니다 WorkSpace. WorkSpace 로그인 정보가 이 이메일 주소로 전송되므로 이메일 주소가 유효한지 확인하십시오.
10. 다음을 선택합니다.
11. 사용자 식별 페이지에서, 9단계에서 생성한 사용자를 선택한 후 다음을 선택합니다.
12. 번들 선택 페이지에서 Standard with Amazon Linux 2를 선택한 후 다음을 선택합니다.
13. 실행 모드 및 사용자 지정에 대해 기본 설정을 사용하고 WorkSpace 생성을 선택합니다. `Pending`상태가 WorkSpace 시작되어 약 20분 `Available` 내에 상태가 전환됩니다.
14. 사용할 수 있게 WorkSpace 되면 9단계에서 제공한 이메일 주소로 액세스 지침이 포함된 이메일을 받게 됩니다.

에 로그인한 후 AWS Management Console 프라이빗 액세스를 사용하여 액세스하고 있는지 테스트할 수 있습니다. WorkSpace

액세스하려면 WorkSpace

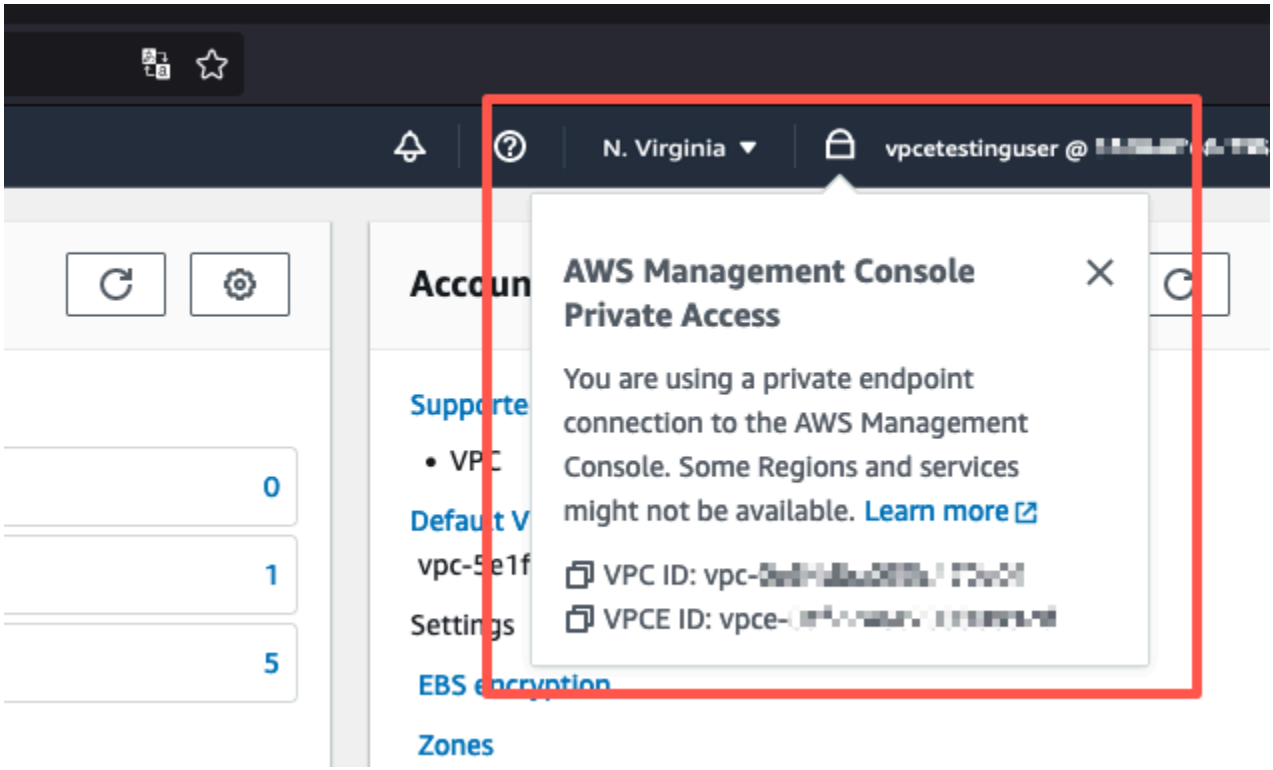
1. 이전 프로시저의 14단계에서 받은 이메일을 엽니다.
2. 이메일에서 제공된 고유 링크를 선택하여 프로필을 설정하고 WorkSpaces 클라이언트를 다운로드합니다.
3. 암호를 설정합니다.
4. 선택한 클라이언트를 다운로드합니다.
5. 클라이언트를 설치하고 실행합니다. 이메일에 제공된 등록 코드를 입력한 다음 등록을 선택합니다.
6. 3단계에서 생성한 자격 증명을 WorkSpaces 사용하여 Amazon에 로그인합니다.

AWS Management Console 프라이빗 액세스 설정을 테스트하려면

1. WorkSpace에서 브라우저를 엽니다. 그런 다음, AWS Management Console로 이동하고 보안 인증 정보를 사용하여 로그인합니다.

   > **Note**
   >
   > Firefox를 브라우저로 사용하는 경우 브라우저 설정에서 HTTPS를 통한 DNS 활성화 옵션이 꺼져 있는지 확인합니다.

2. Amazon S3 콘솔을 열고 AWS Management Console 프라이빗 액세스를 사용하여 연결되어 있는지 확인할 수 있습니다.
3. 탐색 메뉴에서 잠금-프라이빗 아이콘을 선택하면 사용 중인 VPC 및 VPC 엔드포인트를 볼 수 있습니다. 아래의 스크린샷은 잠금-프라이빗 아이콘의 위치와 VPC 정보를 보여줍니다.




## IAM 정책을 사용하여 VPC 설정 테스트

Amazon EC2를 WorkSpaces 사용하거나 액세스를 제한하는 IAM 정책을 배포하여 설정한 VPC를 추가로 테스트할 수 있습니다.

아래의 정책은 지정된 VPC를 Amazon S3가 사용하지 않는 한 Amazon S3에 대한 액세스를 거부합니다.

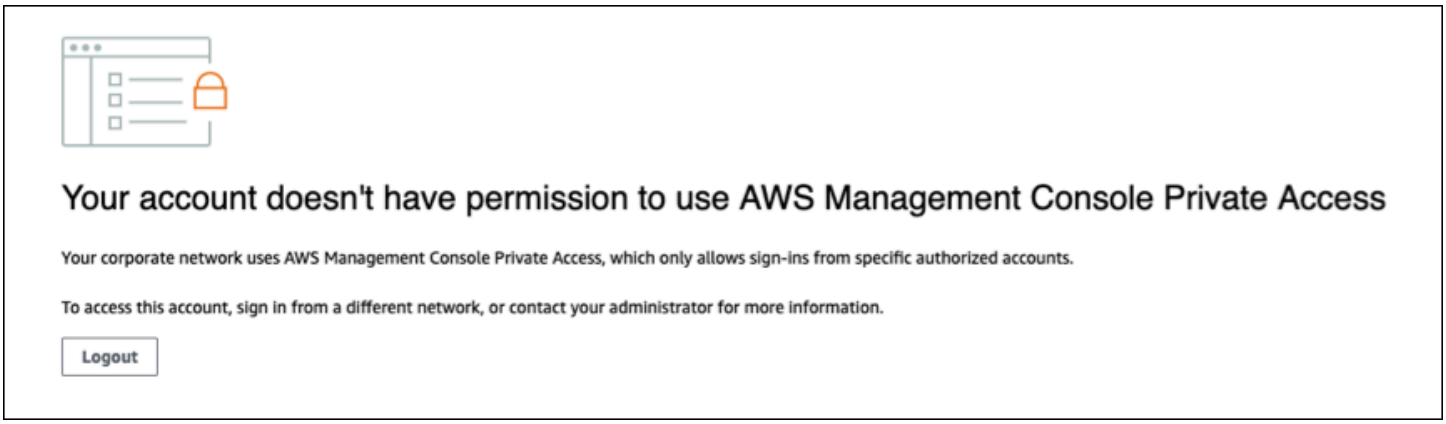
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "S3:*",
            "Resource": "*",
            "Condition": {
                "StringNotEqualsIfExists": {
                    "aws:SourceVpc": "sourceVPC"
                },
                "Bool": {
                    "aws:ViaAwsService": "false"
                }
```

```
            }
        }
    ]
}
```

다음 정책은 로그인 엔드포인트에 대한 AWS Management Console 프라이빗 액세스 정책을 사용하여 선택한 AWS 계정 ID에 대한 로그인을 제한합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": [
                        "AWSAccountID"
                    ]
                }
            }
        }
    ]
}
```

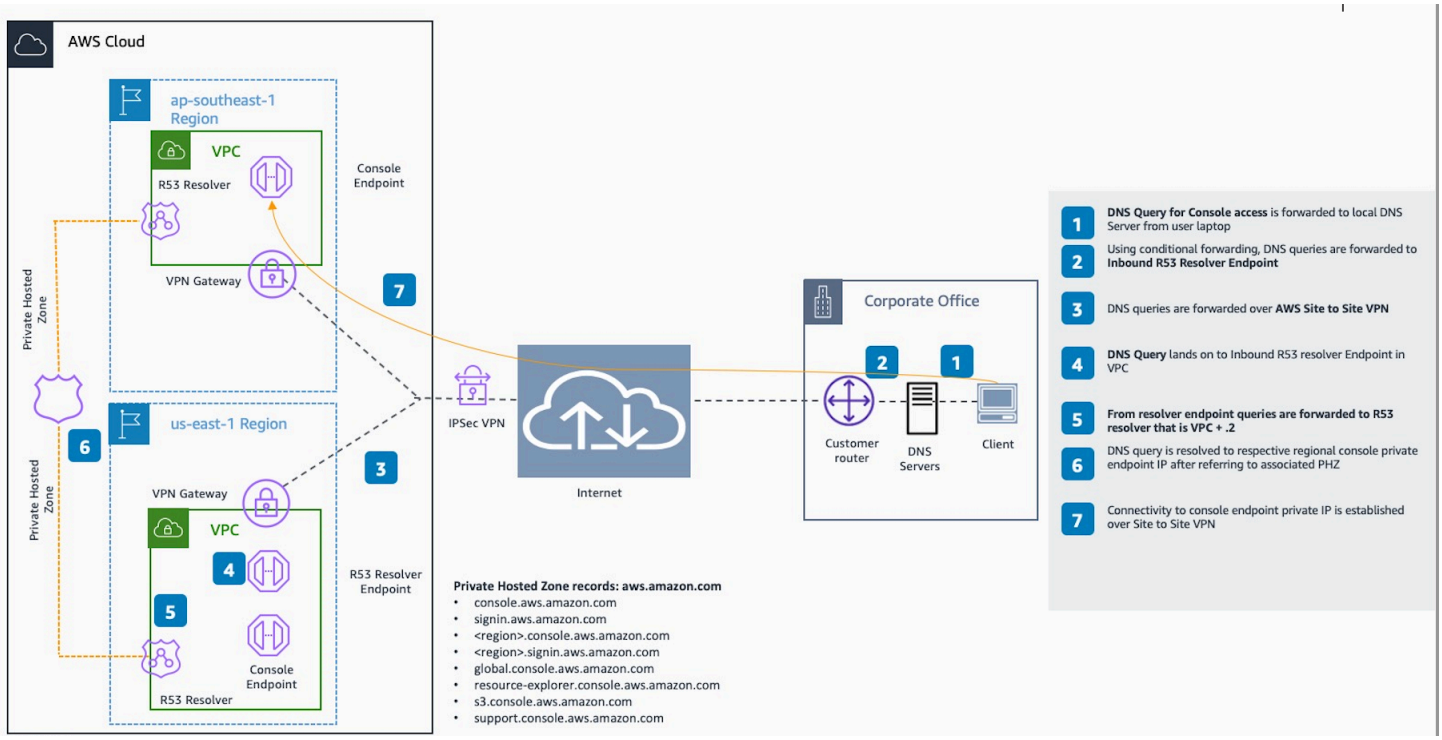
계정에 속하지 않는 ID로 연결할 경우 다음과 같은 오류 페이지가 표시됩니다.



# 참조 아키텍처

온프레미스 네트워크에서 AWS Management Console 프라이빗 액세스에 비공개로 AWS Site-to-Site VPN 연결하려면 AWS 가상 사설 게이트웨이 (VGW) 연결 옵션을 활용할 수 있습니다. AWS Site-to-Site VPN 연결을 생성하고 연결을 통해 트래픽을 전달하도록 라우팅을 구성하여 VPC에서 원격 네트워크에 액세스할 수 있도록 합니다. 자세한 내용은 사이트 간 VPN 사용 AWS 설명서의 사이트 간 VPN 이란 무엇입니까? 를 참조하십시오.AWS  AWS 가상 사설 게이트웨이 (VGW) 는 VPC와 온프레미스 네트워크 간의 게이트웨이 역할을 하는 가용성이 높은 지역 서비스입니다.

AWS Site-to-Site VPN AWS 가상 프라이빗 게이트웨이 (VGW) 로




이 레퍼런스 아키텍처 설계의 핵심 구성 요소는 특히 인바운드 Amazon Route 53 Resolver리졸버입니다. AWS Management Console 프라이빗 액세스 엔드포인트가 생성되는 VPC에서 설정하면 지정된 서브넷에 리졸버 엔드포인트 (네트워크 인터페이스) 가 생성됩니다. 그런 다음 온프레미스 DNS 서버의 조건부 전달자에서 해당 IP 주소를 참조하여 프라이빗 호스팅 영역에 있는 레코드를 쿼리할 수 있습니다. 온프레미스 클라이언트가 에 연결되면 프라이빗 액세스 엔드포인트의 AWS Management Console프라이빗 IP로 라우팅됩니다. AWS Management Console

AWS Management Console 프라이빗 액세스 엔드포인트에 대한 연결을 설정하기 전에 미국 동부 (버지니아 북부) 지역뿐만 아니라 액세스하려는 모든 지역에 AWS Management Console 프라이빗 액세스 엔드포인트를 설정하고 프라이빗 호스팅 영역을 구성하는 사전 요구 사항 단계를 완료하십시오. AWS Management Console

# 콘솔 툴바에서 AWS CloudShell 실행

AWS CloudShell은 브라우저 기반의 사전 인증된 쉘로, 콘솔의 AWS Management Console에서 직접 시작할 수 있습니다. 원하는 쉘(Bash, PowerShell 또는 Z 쉘)을 사용하여 서비스에 대해 AWS CLI 명령을 실행할 수 있습니다.

다음 두 가지 방법 중 하나를 사용하여 Console Toolbar에서 CloudShell을 실행할 수 있습니다.

- 콘솔 왼쪽 하단에서 CloudShell 아이콘을 선택합니다.
- 콘솔 탐색 모음에서 CloudShell 아이콘을 선택합니다.

이 서비스에 대한 자세한 내용은 AWS CloudShell 사용 설명서를 참조하세요.

AWS CloudShell 이용 가능한 AWS 리전에 대한 자세한 내용은 AWS 리전 서비스 목록을 참조하세요. 콘솔 지역 선택은 CloudShell 리전과 동기화됩니다. 선택한 리전에서 CloudShell을 사용할 수 없는 경우 CloudShell은 가장 가까운 리전에서 실행됩니다.

# 청구서 정보 가져오기

필요한 권한이 있는 경우 콘솔에서 AWS 요금에 대한 정보를 얻을 수 있습니다.

결제 정보를 보려면

1. 탐색 모음에서 계정 이름을 선택합니다.
2. 결제 대시보드(Billing Dashboard)를 선택합니다.
3. AWS Billing and Cost Management 대시보드를 사용하여 월간 지출에 대한 요약 및 분석 정보를 찾습니다. 자세한 내용은 AWS Billing 사용 설명서를 참조하세요.

# 콘솔에서 마크다운 사용

CloudWatchAmazon과 같은 일부 서비스는 특정 필드에서 Markdown 사용을 지원합니다. AWS Management Console이 단원에서는 콘솔에서 지원되는 마크다운 서식 유형에 대해 설명합니다.

내용



## 단락, 행 간격, 수평 행

단락은 빈 행으로 구분합니다. HTML로 변환할 때 문단 사이의 빈 줄이 렌더링되도록 하려면 줄 바꿈하지 않는 공백( )이 있는 새 줄과 빈 줄을 차례로 추가하십시오. 다음 예제와 같이 빈 줄을 여러 개 삽입하려면 이 선의 쌍을 반복합니다.

```
 

 
```

문단을 구분하는 수평 규칙을 만들려면 하이픈이 3개씩(---) 있는 새 줄을 추가합니다.

```
Previous paragraph.
---
Next paragraph.
```

고정 너비 유형의 텍스트 블록을 생성하려면 3개의 백틱(`)이 있는 줄을 추가합니다. 고정 너비 유형으로 표시할 텍스트를 입력합니다. 그런 다음 3개의 백틱이 있는 새 줄을 추가합니다. 다음 예제에서는 표시 시 고정 너비 유형으로 서식이 지정되는 텍스트를 보여 줍니다.

```
```
This appears in a text box with a background shading.
The text is in monospace.
```
```

```
```

# 제목

제목을 생성하려면 파운드 기호(#)를 사용합니다. 단일 파운드 기호와 공백은 최상위 제목을 나타냅니다. 2개의 파운드 기호는 두 번째 수준의 제목을 생성하며 3개의 파운드 기호는 세 번째 수준의 제목을 생성합니다. 다음 예제에서는 최상위 수준, 두 번째 수준 및 세 번째 수준의 제목을 보여 줍니다.

```
# Top-level heading
```

```
## Second-level heading
```

```
### Third-level heading
```

# 텍스트 서식 지정

텍스트를 기울임꼴로 지정하려면 해당 텍스트 앞 뒤에 1개의 밑줄( _ )이나 별표(*)를 입력하여 텍스트를 묶습니다.

```
*This text appears in italics.*
```

텍스트를 굵은체로 지정하려면 해당 텍스트 앞 뒤에 두 개의 밑줄이나 별표를 입력하여 텍스트를 묶습니다.

```
**This text appears in bold.**
```

텍스트에 취소선 서식을 지정하려면 해당 텍스트 앞 뒤에 2개의 물결 기호(~)를 입력하여 텍스트를 묶습니다.

```
~~This text appears in strikethrough.~~
```

# Links

텍스트 하이퍼링크를 추가하려면 다음 예제와 같이 대괄호([ ])로 둘러싸인 링크 텍스트를 입력한 다음 괄호(( ))에 전체 URL을 입력합니다.

```
Choose [link_text](http://my.example.com).
```

## List

글머리표 목록으로 행 서식을 지정하려면 다음 예제와 같이 하나의 별표(*)와 함께 별도의 줄에 입력한 후 공백을 입력합니다.

```
Here is a bulleted list:
* Ant
* Bug
* Caterpillar
```

번호가 매겨진 목록으로 행 서식을 지정하려면 다음 예제와 같이 숫자, 마침표(.), 공백과 함께 별도의 줄에 입력합니다.

```
Here is a numbered list:
1. Do the first step
2. Do the next step
3. Do the final step
```

## 테이블 및 버튼 (CloudWatch 대시보드)

CloudWatch 대시보드 텍스트 위젯은 마크다운 테이블 및 버튼을 지원합니다.

테이블을 생성하려면 세로 막대(|)를 사용하여 열을 구분하고 새 줄을 사용하여 행을 구분합니다. 첫 번째 행을 헤더 행으로 생성하려면 헤더 행과 값의 첫 번째 행 사이에 줄을 삽입합니다. 그런 다음 표의 각 열에 대해 최소 3개의 하이픈(-)을 추가합니다. 세로 막대를 사용하여 열을 구분합니다. 다음 예제에서는 2개의 열, 헤더 행 및 2개의 데이터 행이 있는 테이블의 마크다운을 보여 줍니다.

```
Table | Header
----|-----
Amazon Web Services | AWS
1 | 2
```

이전 예제의 마크다운 텍스트는 다음 테이블을 생성합니다.

| 표 | 헤더 |
| --- | --- |
| Amazon Web Services | AWS |
| 1 | 2 |

CloudWatch 대시보드 텍스트 위젯에서 하이퍼링크가 버튼으로 표시되도록 형식을 지정할 수도 있습니다. 버튼을 만들려면 다음 예제와 같이 [button:*Button text*]를 사용한 다음 괄호(( ))에 전체 URL을 입력합니다.

```
[button:Go to AWS](http://my.example.com)
[button:primary:This button stands out even more](http://my.example.com)
```

# 문제 해결

와 관련된 일반적인 문제에 대한 해결책을 찾으려면 이 섹션을 참조하십시오 AWS Management Console.

또한 Amazon Q Developer를 사용하여 일부 AWS 서비스의 일반적인 오류를 진단하고 해결할 수 있습니다. 자세한 내용은 Amazon Q 개발자 사용 설명서의 Amazon Q Developer와 함께 콘솔에서 발생하는 일반적인 오류 진단을 참조하십시오.

주제



## 페이지가 정상적으로 로드되지 않음

- 이 문제가 가끔 발생하는 경우 인터넷 연결을 확인합니다. 다른 네트워크를 통해 연결하거나, VPN을 사용하거나 사용하지 않고 연결해 보거나, 다른 웹 브라우저를 사용해 보십시오.
- 영향을 받는 모든 사용자가 같은 팀에 속해 있다면 개인 정보 보호 브라우저 확장 또는 보안 방화벽 문제일 수 있습니다. 개인 정보 보호 브라우저 확장 프로그램 및 보안 방화벽이 에서 사용하는 도메인에 대한 액세스를 차단할 수 있습니다. AWS Management Console이러한 확장을 끄거나 방화벽 설정을 조정해 봅니다. 연결 문제를 확인하려면 브라우저 개발자 도구(Chrome, Firefox)를 열고 콘솔(Console) 탭에서 오류를 조사합니다. AWS Management Console 는 다음 목록을 포함한 도메인의 접미사를 사용합니다. 단, 이 목록이 전부는 아니며 추후 변경될 수 있습니다. 이러한 도메인의 접미사는 AWS에서 독점적으로 사용하는 것은 아닙니다.
  - .a2z.com
  - .amazon.com
  - .amazonaws.com
  - .aws
  - .aws.com
  - .aws.dev
  - .awscloud.com

- .awsplayer.com
- .awsstatic.com
- .cloudfront.net
- .live-video.net

> **Warning**
>
> 2022년 7월 31일부터 더 AWS 이상 인터넷 익스플로러 11을 지원하지 않습니다. 지원되는 다른 AWS Management Console 브라우저와 함께 사용하는 것이 좋습니다. 자세한 내용은 AWS News 블로그를 참조하세요.

## 브라우저에 연결할 때 내 브라우저에 '액세스 거부' 오류가 표시됩니다. AWS Management Console

다음을 모두 사용하는 경우 콘솔의 최근 변경 사항이 액세스에 영향을 미칠 수 있습니다.

- VPC 내의 브라우저.
- VPC 엔드포인트.
- `aws:SourceIp`글로벌 조건 키가 포함된 IAM 정책.

콘솔에서 IAM 정책 페이지로 이동합니다. `aws:SourceIp`글로벌 조건 키 및 추가 `aws:SourceVpc` 키가 포함된 IAM 정책을 검토하는 것이 좋습니다.

또는 VPC 엔드포인트를 AWS Management Console 통해 액세스하고 정책의 조건을 `aws:SourceVpc` 사용하기 위해 AWS Management Console Private Access 기능에 온보딩하는 것을 고려할 수 있습니다. 자세한 정보는 AWS Management Console 프라이빗 액세스을 참조하세요.

## 내 브라우저에 연결할 때 시간 초과 오류가 표시됩니다. AWS Management Console

기본값에서 서비스 중단이 발생한 경우 AWS 리전, 연결하려고 할 때 브라우저에 504 Gateway Timeout 오류가 표시될 수 있습니다. AWS Management Console다른 지역에서 에 로그인하려면 URL에 AWS Management Console 대체 지역 엔드포인트를 지정하십시오. 예를 들어 `us-west-1`(캘리포

니아 북부) 리전에서 중단이 발생한 경우 us-west-2(오레곤) 리전에 액세스하려면 다음 템플릿을 사용합니다.

```
https://region-code.console.aws.amazon.com
```

자세한 내용은 AWS 일반 참조의 AWS Management Console 서비스 엔드포인트를 참조하세요.

를 포함한 모든 AWS 서비스상태를 AWS Management Console보려면 을 참조하십시오 AWS Health Dashboard.

## AWS Management Console 의 언어를 변경하고 싶지만 페이지 하단에서 언어 선택 메뉴를 찾을 수 없음

언어 선택 메뉴가 새로운 통합 설정(United Settings) 페이지로 이동했습니다. 의 AWS Management Console언어를 변경하려면 통합 설정 페이지로 이동한 다음 콘솔의 언어를 선택합니다.

자세한 내용은 AWS Management Console의 언어 변경을 참조하세요.

# 문서 기록

다음 표에서는 2021년 3월을 기준으로 AWS Management Console 시작 가이드에서 변경된 중요 사항에 대해 설명합니다.

| 변경 사항 | 설명 | 날짜 |
| --- | --- | --- |
| 아마존 Q와 채팅하기 | 사용자가 Amazon Q Developer에 AWS 질문할 수 있는 방법을 자세히 설명하는 새 설정 페이지. 자세한 내용은 Amazon Q 개발자와의 채팅을 참조하십시오. | 2024년 5월 29일 |
| 내 애플리케이션 | 내 애플리케이션을 소개하는 새 페이지입니다. 자세한 내용은 내 애플리케이션이란 무엇입니까? 를 참조하십시오. AWS. | 2023년 11월 29일 |
| 통합 설정 구성 | 언어 및 리전을 포함하여 현재 사용자에게 적용되는 설정 및 기본값을 구성하는 데 사용되는 새 설정 페이지입니다. 자세한 내용은 통합 설정 구성을 참조하세요. | 2022년 4월 6일 |
| 새 AWS Console Home UI | 새 AWS Console Home UI에는 중요한 사용 정보를 표시하는 위젯과 서비스 바로 가기가 AWS 포함되어 있습니다. 자세한 내용은 위젯 작업을 참조하세요. | 2022년 2월 25일 |
| 콘솔 언어 변경 | AWS Management Console에 대해 다른 언어를 선택합니다. 자세한 내용은 AWS | 2021년 4월 1일 |

| 변경 사항 | 설명 | 날짜 |
| --- | --- | --- |
| | Management Console의 언어 변경을 참조하세요. | |
| 런칭 CloudShell | AWS CloudShell 에서 AWS Management Console 열고 AWS CLI 명령을 실행합니다. 자세한 내용은 시작을 AWS CloudShell 참조하십시오. | 2021년 3월 22일 |

# AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.

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