



manTech
better solution

Installing Accordion v2.12.1 on AWS

목차

1. 제품 개요	3
1.1 소개	3
1.1.1 필수 구성 요소 및 요구 사항	3
1.1.2 지원하는 리전	4
1.1.3 아키텍처	4
1.1.4 사용 사례	6
2. 계획 지침	6
2.1 보안	6
2.1.1 IAM 정책 설정	7
2.2 비용 및 라이선스	9
2.3 인스턴스 크기	10
3. 배포 절차	11
3.1 1단계, 아코디언 인프라 리소스 배포	11
3.1.1 VPC 및 서브넷 생성	11
3.1.2 보안 그룹 생성	12
3.1.3 인스턴스 생성	14
3.2 2단계, 아코디언 설치 세팅	25
3.2.1 Bastion 인스턴스	25
3.2.2 Node 인스턴스	31
3.3 3단계, 아코디언 설치	32
4. 운영 지침	32
4.1 아코디언 백업과 복구 지원	32
4.1.1 AWS에서 아코디언 백업과 복구	32
4.2 정기적인 유지보수	34
4.3 비상 유지보수	35

4.3.1 kubelet 프로세스 시작.....	35
4.3.2 상태 점검.....	35
4.3.3 아코디언 오류 유형.....	37
4.3.4 아코디언 오류 복구 절차.....	37
4.3.5 아코디언 복구 실패 시 복구 절차.....	38
4.4 RTO.....	38
4.5 AWS 서비스 한도 관리.....	38
5. 시스템 관리.....	39
5.1 로그인.....	39
5.2 화면 구성.....	39
5.2.1 헤더영역.....	40
5.2.2 메뉴영역.....	41
5.2.3 이벤트영역.....	45
5.3 라이선스 관리.....	46
5.4 패치 및 업데이트 관리.....	46
6. 지원.....	46
6.1 기술지원.....	46
6.2 지원 비용.....	47
6.3 SLA.....	47

1. 제품 개요

이 문서에서는 AWS를 이전에 사용한 적이 있으며 AWS 서비스에 대해 잘 알고 있다고 가정합니다. 만약 AWS의 새로운 기능은 AWS 설명서의 시작 섹션을 참조하십시오. 또한 다음과 같은 AWS 기술을 숙지해야 합니다.

- Amazon VPC - Amazon VPC(Amazon Vprivate Cloud) 서비스를 사용하면 사용자가 정의한 가상 네트워크에서 AWS 서비스 및 기타 리소스를 시작할 수 있는 AWS Cloud의 분리된 전용 섹션을 프로비저닝할 수 있습니다. 고유한 IP 주소 범위 선택, 서브넷 생성, 경로 테이블 및 네트워크 게이트웨이 구성 등 가상 네트워킹 환경을 완벽하게 제어할 수 있습니다.
- Amazon EC2 - Amazon Elastic Compute Cloud(Amazon EC2) 서비스를 사용하면 다양한 운영 체제에서 가상 시스템 인스턴스를 시작할 수 있습니다. 기존 AMI(Amazon 시스템 이미지)에서 선택하거나 자체 가상 시스템 이미지를 가져올 수 있습니다.

1.1 소개

맨택의 아코디언은 쿠버네티스를 기반으로 애플리케이션 배포 및 운영을 위한 최적의 환경을 제공하며, 클라우드 네이티브 애플리케이션 실현을 위한 All-in-One 통합 관리 플랫폼입니다. 클라우드 컴퓨팅이 IT 환경의 표준이 되어 가고 있는 가운데 변화하는 고객의 요구에 빠르게 대응하고 클라우드 인프라 환경에 맞는 애플리케이션 운영 및 클라우드 통합 관리를 지원합니다.

1.1.1 필수 구성 요소 및 요구 사항

이 항목에서는 AWS에 아코디언을 설치하기 위한 필수 구성 요소 및 리소스 요구 사항에 대해 설명합니다.

- 필수 구성 요소

아코디언을 사용하기 위해서는 두 가지의 AMI를 모두 사용해야 합니다. 제품의 설치본을 담고 있는 Bastion AMI와 클러스터를 구성 할 Node AMI를 EC2로 배포해야 합니다.

평가판 활성화 키를 받으려면 (<https://accordions.co.kr/>)에서 문의하기를 통해 활성화 키를 받으셔야 합니다.

- 요구 사항

아코디언을 설치하려면 다음 수준의 가상 머신이 필요합니다.

가상 머신 이름 (태그)	가상 머신 타입	가상 머신 개수(기본)
마스터 노드	C4.2xlarge, C5a.2xlarge, C5.2xlarge	3
인프라 노드	C4.2xlarge, C5a.2xlarge, C5.2xlarge	2
워커 노드	C4.2xlarge, C5a.2xlarge, C5.2xlarge	2

1.1.2 지원하는 리전

코드	이름
ap-northeast-2	아시아 태평양(서울)

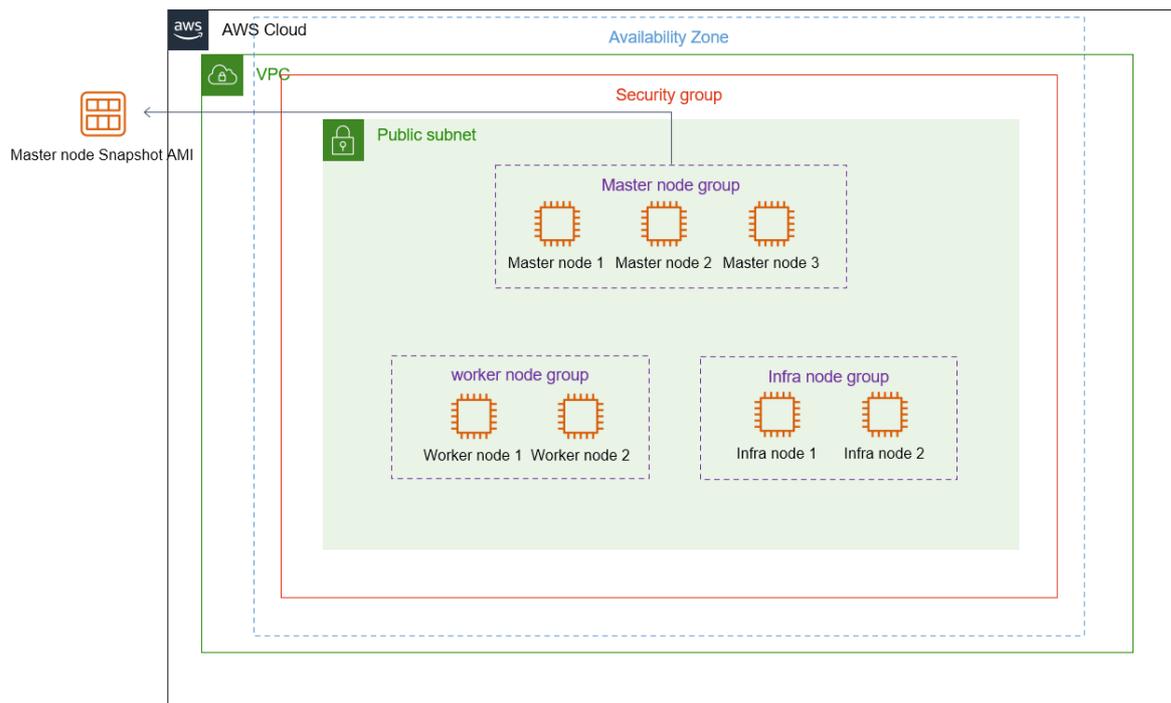
1.1.3 아키텍처

이 아키텍처는 VPC에 다음에 구성 요소를 포함하여, 아코디언을 배포합니다.

- 공유 서브넷에 Bastion, Node EC2 인스턴스

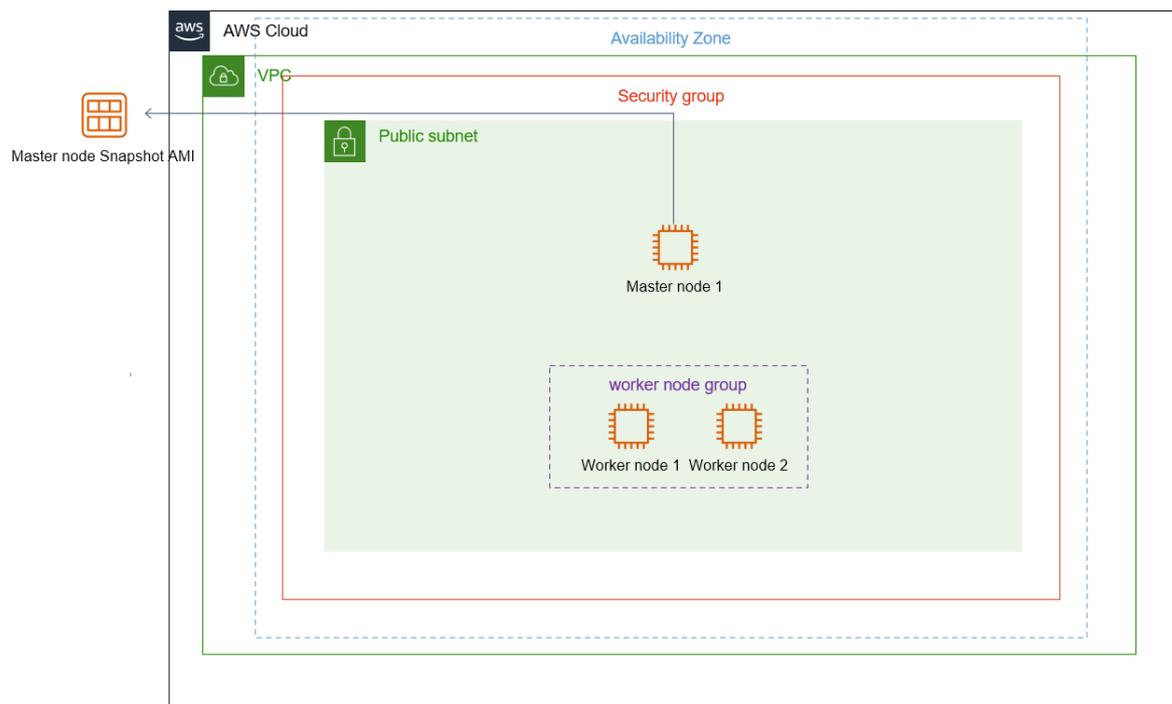
A. 고가용성 구성

이 구성은 마스터 노드의 고가용성을 보장 합니다. 아코디언의 핵심 컴포넌트들의 서비스 지속성이 보장됩니다.



B. 싱글 구성

비용 절감 및 테스트 환경 구성 목적으로 싱글 구성이 가능합니다. 싱글 구성 시에는 마스터 노드 다운타임 중에 아코디언 기능을 이용하는 웹페이지에 접속 할 수 없게 됩니다.



1.1.4 사용 사례

아코디언 사용사례는 아래 링크를 참조하십시오.

- <https://www.youtube.com/watch?v=LKZTspNYbVo&list=PL6vKW169zYSeqG6XCfeBkhBWGWIoVlvqp>

2. 계획 지침

2.1 보안

배포된 AMI에 아코디언을 설치하는데 필요한 것은 SSH 접근입니다.

- ✓ SSH 접근에 AWS 루트 자격 증명을 사용하지 않습니다.

2.1.1 IAM 정책 설정

아코디언을 배포 및 서비스를 위해 VPC, EC2, Subnet, SG 대해 생성 및 조회 할 수 있는 권한이 필요합니다. IAM 사용자에게 아래 상자에 권한이 있어야 합니다.

- 1) AWS Management Console에 IAM 대쉬보드로 접속합니다.
- 2) 좌측 메뉴에서 정책 > [정책 생성] 선택
- 3) 정책 생성 화면에서 JSON 탭에서 아래 내용을 입력 후 생성

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CopyImage",
        "ec2:CreateImage",
        "ec2:CreateKeypair",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2>DeleteKeypair",
        "ec2>DeleteSecurityGroup",
        "ec2>DeleteSnapshot",
        "ec2>DeleteVolume",
        "ec2:DeregisterImage",
        "ec2:DescribeImageAttribute",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeRegions",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSnapshots",
        "ec2:DescribeSubnets",
        "ec2:DescribeTags",
        "ec2:DescribeVolumes",

```

```
"ec2:DetachVolume",
"ec2:GetPasswordData",
"ec2:ModifyImageAttribute",
"ec2:ModifyInstanceAttribute",
"ec2:ModifySnapshotAttribute",
"ec2:RegisterImage",
"ec2:RunInstances",
"ec2:StopInstances",
"ec2:TerminateInstances",
"ec2:AcceptVpcPeeringConnection",
"ec2:AcceptVpcEndpointConnections",
"ec2:AllocateAddress",
"ec2:AssignIpv6Addresses",
"ec2:AssignPrivateIpAddresses",
"ec2:AssociateAddress",
"ec2:AssociateDhcpOptions",
"ec2:AssociateRouteTable",
"ec2:AssociateSubnetCidrBlock",
"ec2:AssociateVpcCidrBlock",
"ec2:AttachClassicLinkVpc",
"ec2:AttachInternetGateway",
"ec2:AttachNetworkInterface",
"ec2:AttachVpnGateway",
"ec2:AuthorizeSecurityGroupEgress",
"ec2:AuthorizeSecurityGroupIngress",
"ec2:CreateCarrierGateway",
"ec2:CreateCustomerGateway",
"ec2:CreateDefaultSubnet",
"ec2:CreateDefaultVpc",
"ec2:CreateDhcpOptions",
"ec2:CreateEgressOnlyInternetGateway",
"ec2:CreateFlowLogs",
"ec2:CreateInternetGateway",
"ec2:CreateLocalGatewayRouteTableVpcAssociation",
"ec2:CreateNatGateway",
"ec2:CreateNetworkAcl",
"ec2:CreateNetworkAclEntry",
"ec2:CreateNetworkInterface",
"ec2:CreateNetworkInterfacePermission",
"ec2:CreateRoute",
"ec2:CreateRouteTable",
```

```

    "ec2:CreateSecurityGroup",
    "ec2:CreateSubnet",
    "ec2:CreateTags",
    "ec2:CreateVpc",
    "ec2:CreateVpcEndpoint",
    "ec2:CreateVpcEndpointConnectionNotification",
    "ec2:CreateVpcEndpointServiceConfiguration",
    "ec2:CreateVpcPeeringConnection",
    "ec2:CreateVpnConnection",
    "ec2:CreateVpnConnectionRoute",
    "ec2:CreateVpnGateway"
  ],
  "Resource": "*"
}
]
}

```

2.2 비용 및 라이선스

아코디언은 BYOL을 지원합니다. BYOL은 귀사의 파트너사 또는 총판에서 제공되며, 플랫폼에 관계없이 모든 프라이빗 및 퍼블릭 클라우드에서 동일한 주문 방식을 제공합니다.

라이선스	가격(1vCPU)	내용
Accordion container platform Basic	80,000	<ul style="list-style-type: none"> - Container Orchestration based on Kubernetes - Web based Admin portal - Private Docker registry - Event monitoring & alarm - Roll based access control - Support auto scale out/in - Bundled L2/L3 overlay network - Support CI/CD pipeline based
Accordion container platform Enterprise	140,000	<ul style="list-style-type: none"> - Container Orchestration based on Kubernetes - Web based Admin portal - Private Docker registry - Event monitoring & alarm - Roll based access control - Support auto scale out/in - Bundled L2/L3 overlay network

		<ul style="list-style-type: none"> - Support CI/CD pipeline based - Support Tomcat, Wildfly middleware service - Support APM & Web log analysis
--	--	--

1. 청구 가능한 AWS 서비스의 전체 목록

AWS 서비스 비용은 사용자가 부담해야 합니다. 사용하려는 클러스터 크기에 따라 리소스의 비용이 달라집니다. 자세한 내용은 본 안내서에서 사용할 각 AWS 서비스에 대한 페이지(<https://aws.amazon.com/pricing/>)를 참조하십시오.

A. EC2 인스턴스(필수)

B. EBS(필수)

2.3 인스턴스 크기

아코디언 AMI는 AWS에서 다음과 같은 크기의 인스턴스 사양을 지원합니다. Node AMI의 경우 사용하고자 하는 노드 타입의 따라 인스턴스 타입을 지정하십시오. 최신 정보는 다음 링크(<https://aws.amazon.com/ko/ec2/instance-types/>)를 참조하십시오.

1. Bastion AMI

노드 타입	인스턴스 타입	vCPU	Memory(GiB)	EBS Volume	EBS Type
마스터 노드	C4.2xlarge	8	15	300GB	범용 SSD(gp2)
마스터 노드	C5a.2xlarge 또는 C5.2xlarge	8	16	300GB	범용 SSD(gp2)

2. Node AMI

노드 타입	인스턴스 타입	vCPU	Memory(GiB)	EBS Volume	EBS Type
워커 노드	C4.2xlarge	8	15	500GB	범용 SSD(gp2)

워커 노드	C5a.2xlarge 또는 C5.2xlarge	8	16	500GB	범용 SSD(gp2)
마스터 노드	C4.2xlarge	8	15	300GB	범용 SSD(gp2)
마스터 노드	C5a.2xlarge 또는 C5.2xlarge	8	16	300GB	범용 SSD(gp2)
인프라 노드	C4.2xlarge	8	15	200GB	범용 SSD(gp2)
인프라 노드	C5a.2xlarge 또는 C5.2xlarge	8	16	200GB	범용 SSD(gp2)

3. 배포 절차

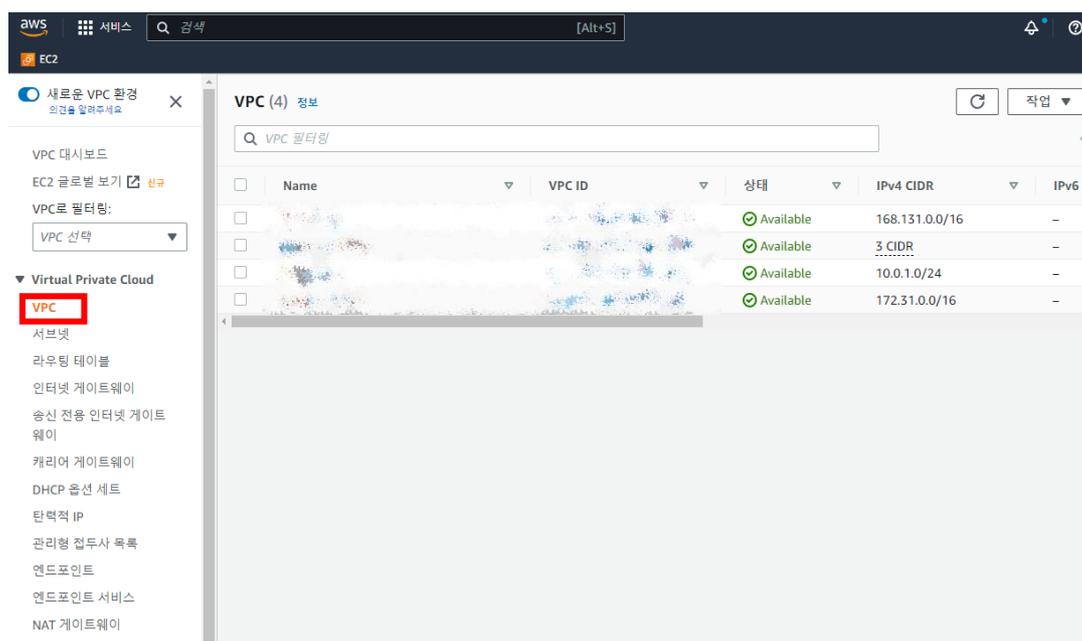
3.1 1단계, 아코디언 인프라 리소스 배포

3.1.1 VPC 및 서브넷 생성

아코디언을 설치하기 위해서는 VPC와 서브넷을 생성하거나 기존의 생성되어 있는 것을 사용 할 수 있습니다. 아코디언을 구성하기 위해서는 각각 하나의 VPC와 서브넷이면 충분합니다.

A. 기존 VPC 및 서브넷 확인

1. 기존 VPC는 다음 경로에서 확인 할 수 있습니다. [관리자 콘솔 > 서비스 > 네트워킹 및 콘텐츠 전송 > VPC]

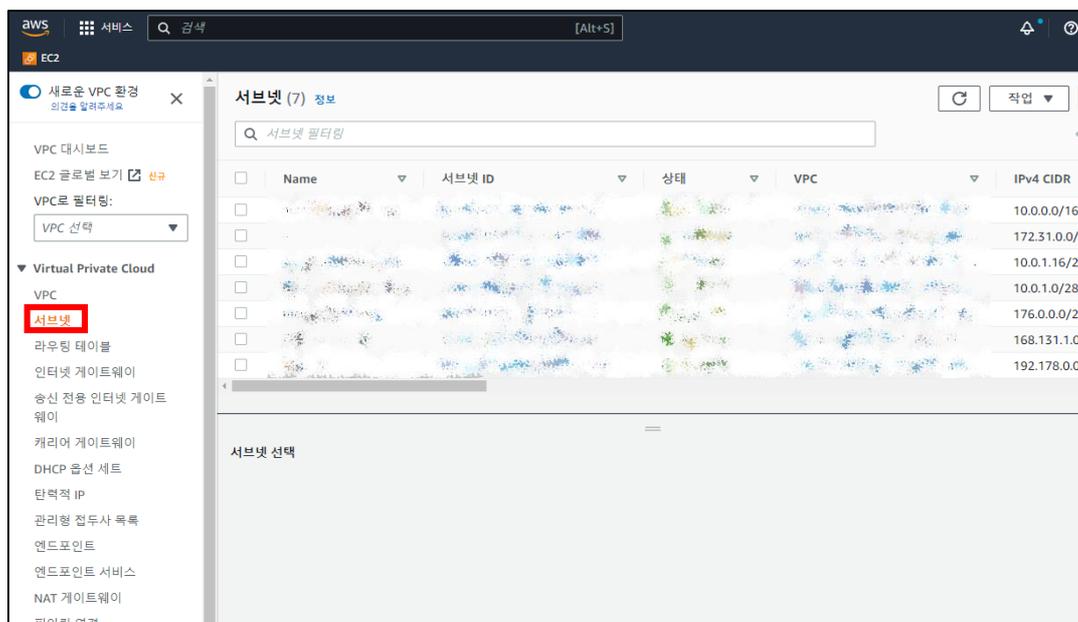


2. 기존 서브넷은 다음 경로에서 확인 할 수 있습니다. [관리자 콘솔 > 서비스 > 네트워킹 및 콘텐츠 전송 > 서브넷]

B. 아코디언 서브넷 생성

1. 아마존 VPC 콘솔에 접속합니다. (<https://console.aws.amazon.com/vpc/>)
2. 좌측 메뉴에서 '서브넷 > 서브넷 생성'을 선택합니다.
3. 필요에 따라 서브넷 세부 정보를 지정하고 서브넷 생성을 선택합니다.

메뉴	값
서브넷 이름	Accordion subnet
VPC	구성하고자 하는 VPC 선택
VPC CIDRs	-
가용 영역	-
IPv4 CIDR block	-



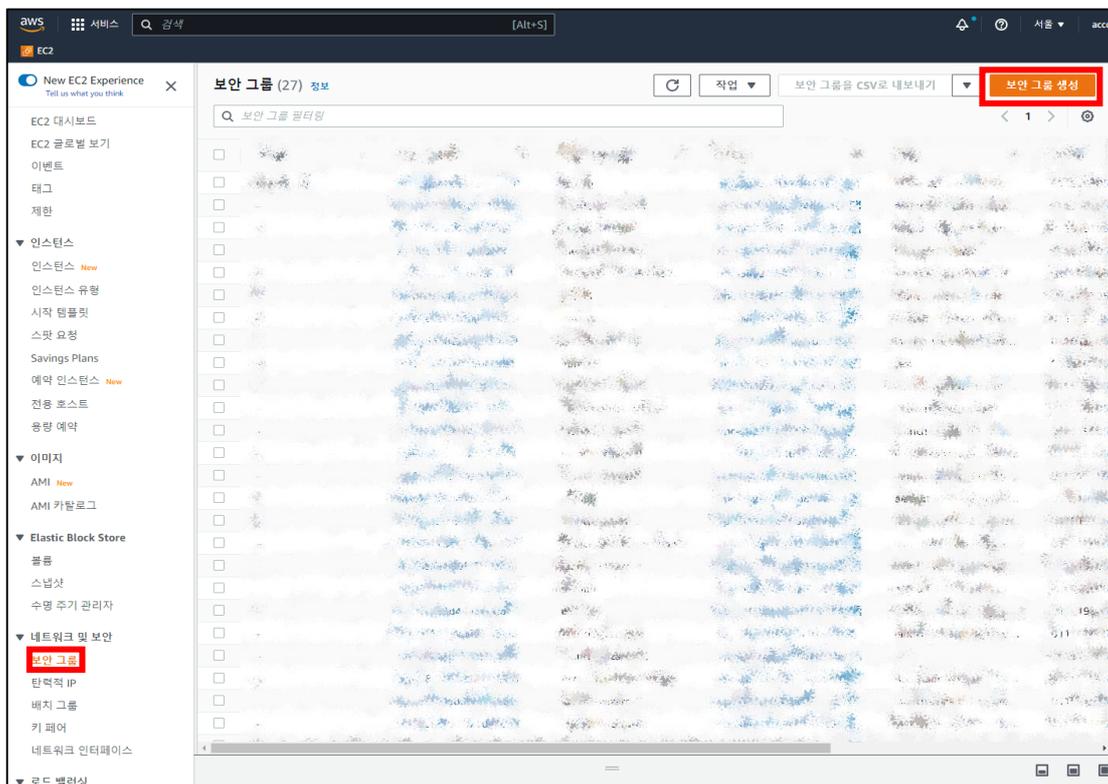
3.1.2 보안 그룹 생성

아코디언 클러스터 내 구성 컴포넌트들이 서로 통신할 수 있도록 보안 그룹을 설정해야 합니다.

A. 아코디언 보안 그룹 생성

1. AWS EC2 관리 콘솔의 접속합니다.

2. [네트워크 및 보안 > 보안 그룹] 을 선택하고 보안 그룹 생성을 선택합니다.



3. 아래와 같이 인바운드 규칙에 새 규칙을 추가합니다.

A. Inbound Rule

유형	포트 범위	소스 IP	설명
TCP	30000~32767	0.0.0.0/0	NodePort Services
모든 트래픽	모든 포트	Accordion SG	Calico networking IP in IP (encapsulation)
SSH	전체	고객 관리자 IP	SSH

B. Outbound Rule

유형	포트 범위	소스 IP	설명
모든 트래픽	전체	0.0.0.0/0	-

4. 다음과 같이 Name 태그 추가

메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion SG		

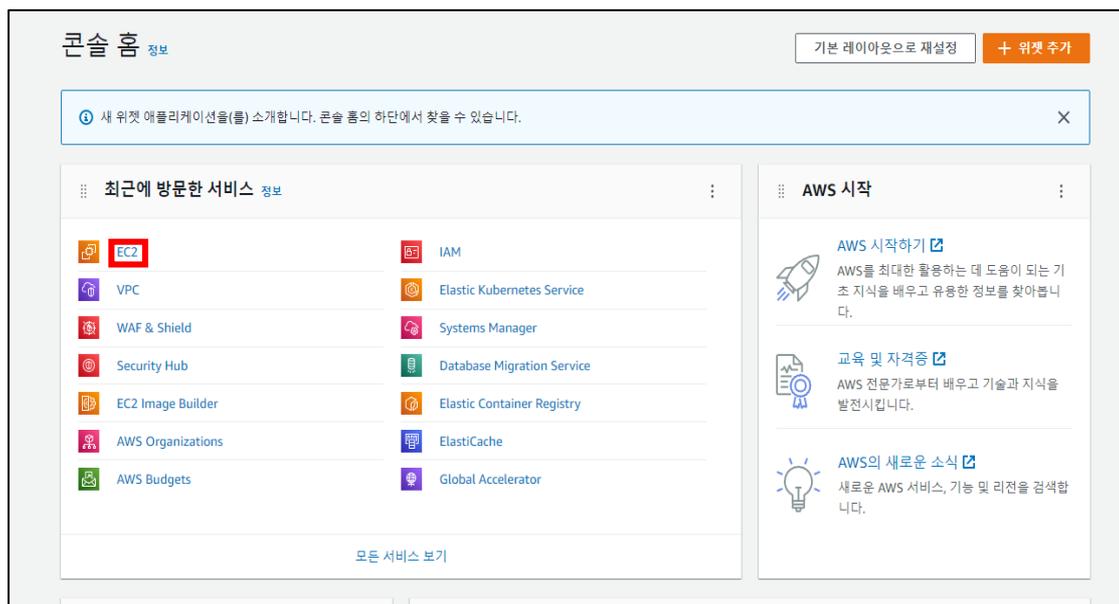
3.1.3 인스턴스 생성

A. 고가용성 구성 배포

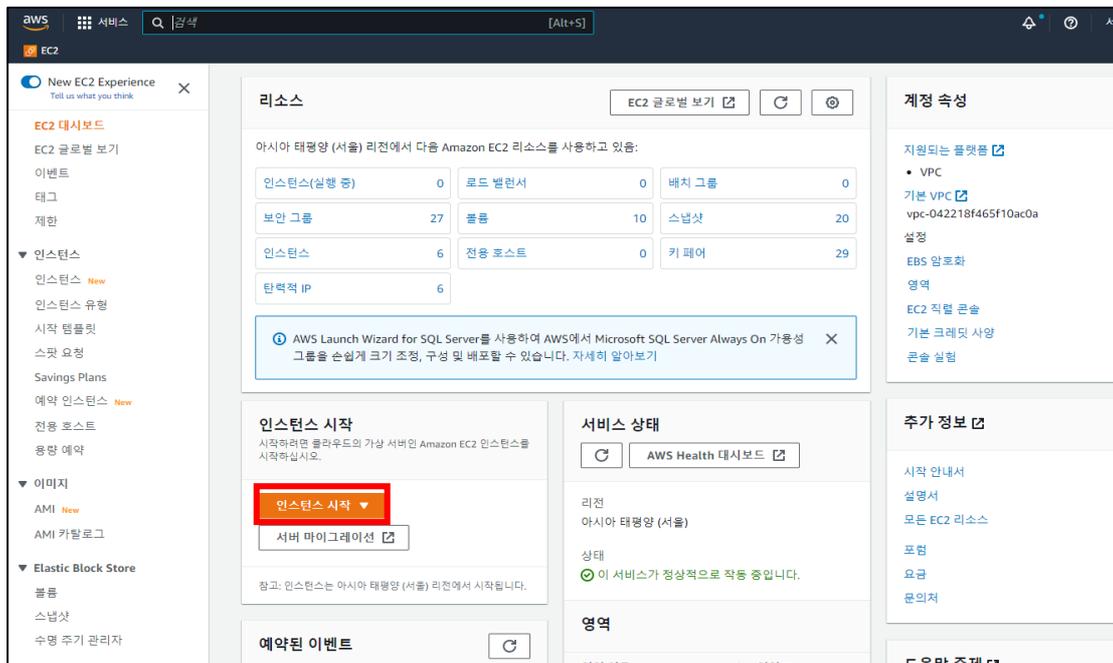
고가용성 구성은 마스터 노드 3중화, 인프라 노드를 포함한 구성이 권장됩니다.

가. Bastion AMI

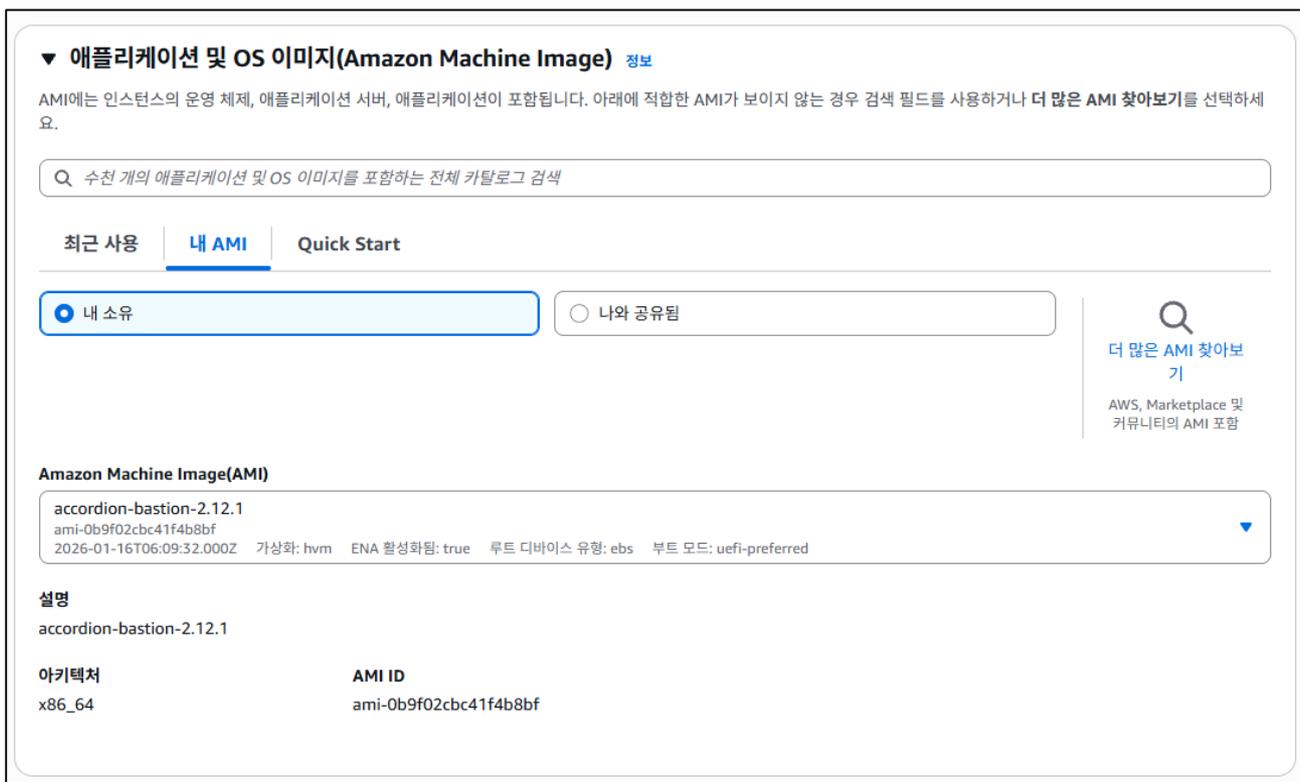
1. AWS 관리 콘솔에 로그인하고 EC2를 클릭합니다.



2. 인스턴스 시작을 클릭합니다.



3. 공유 AMI를 사용하여 인스턴스를 생성합니다.



4. 인스턴스 타입을 선택합니다.

- ✓ 인스턴스 타입은 [2.3 인스턴스 크기]를 참조하십시오.

5. 인스턴스 세부 정보를 설정합니다.

메뉴	값
구성 세부사항	<ul style="list-style-type: none"> - 인스턴스 수 : 1 - 네트워크 : VPC에 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 서브넷 : 서브넷에 대한 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 퍼블릭 IP 자동 할당 : 활성화
기타	- 추가적인 알림이 없다면 기본 옵션 사용

6. 스토리지 추가

- ✓ 스토리지는 [2.3 인스턴스 크기]를 참조하십시오.

7. 태그 추가

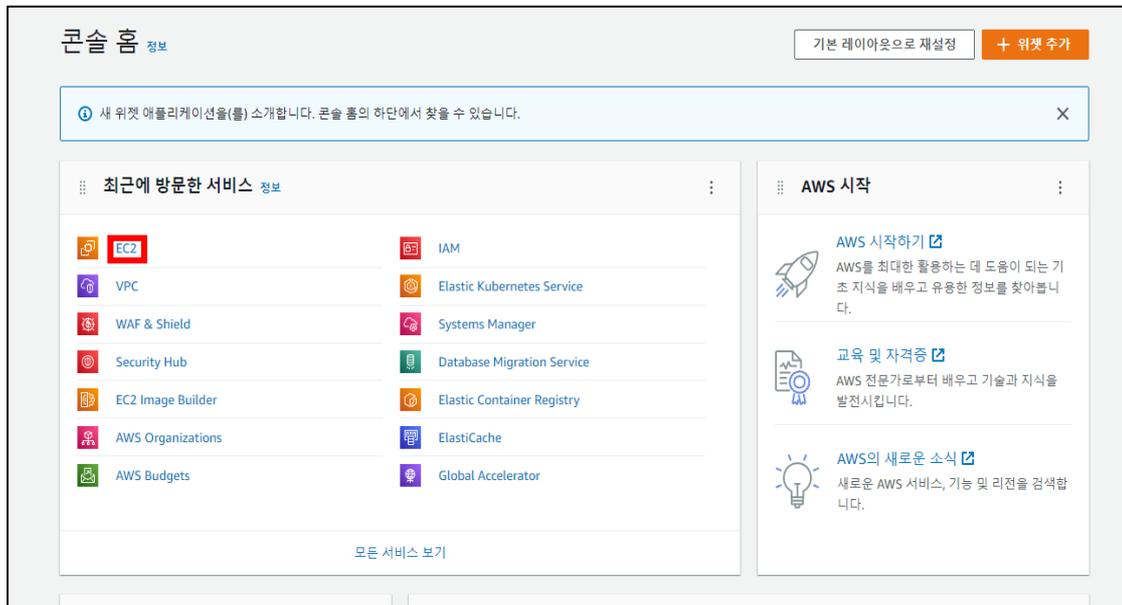
메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion Mastser		

8. 보안 그룹 설정

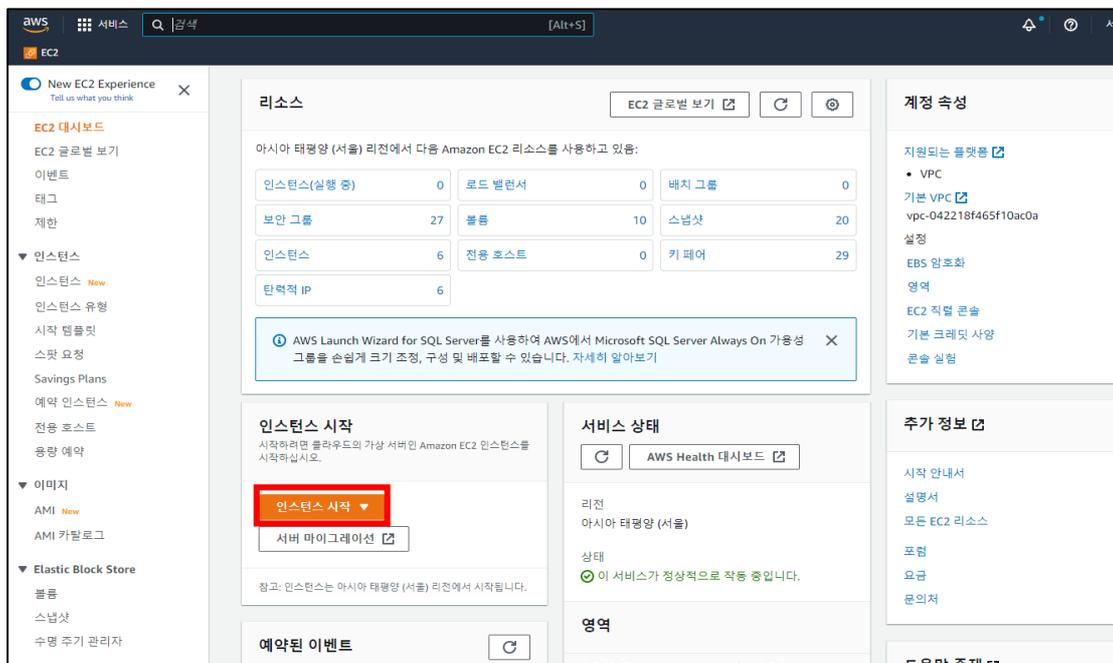
- ✓ 기존 보안 그룹을 선택하십시오. [Accordion SG]

나. Node AMI

1. AWS 관리 콘솔에 로그인하고 EC2를 클릭합니다.



2. 인스턴스 시작을 클릭합니다.



3. 공유 AMI를 사용하여 인스턴스를 생성합니다.

The screenshot shows the AWS console interface for selecting an AMI and instance type. The '내 AMI' (My AMI) tab is active, and the 'amazon-node-2.12.1' AMI is selected. The instance type 't3.xlarge' is chosen. A warning message is displayed: '프리 티어: AWS 계정을 개설한 것해에 프리 티어 AMI, 매월 750 시간의 퍼블릭 IPv4 주소 사용량, 30GiB의 EBS 스토리지, 2백만 I/O, 1GB의 스냅샷 및 100GB의 인터넷 대역폭과 함께 사용하면 매월 750시간의 t2.micro 인스턴스 사용량(또는 t2.micro를 사용할 수 없는 경우 t3.micro)을 제공받습니다. 데이터 전송 요금은 프리티어 허용 범위에 포함되지 않습니다.'

4. 인스턴스 타입을 선택합니다.

- 사용하고자 하는 노드 타입의 따라 인스턴스 타입을 지정합니다.
- ✓ 인스턴스 타입은 [2.3 인스턴스 크기]를 참조하십시오.

5. 인스턴스 세부 정보를 설정합니다.

- 마스터 노드

메뉴	값
구성 세부사항	<ul style="list-style-type: none"> - 인스턴스 수 : 2 - 네트워크 : VPC에 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 서브넷 : 서브넷에 대한 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 퍼블릭 IP 자동 할당 : 활성화
기타	- 추가적인 알림이 없다면 기본 옵션 사용

- 인프라 노드

메뉴	값
구성 세부사항	<ul style="list-style-type: none"> - 인스턴스 수 : 2 - 네트워크 : VPC에 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 서브넷 : 서브넷에 대한 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 퍼블릭 IP 자동 할당 : 활성화
기타	- 추가적인 알림이 없다면 기본 옵션 사용

- 워커 노드

메뉴	값
구성 세부사항	<ul style="list-style-type: none"> - 인스턴스 수 : 2 - 네트워크 : VPC에 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 서브넷 : 서브넷에 대한 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 퍼블릭 IP 자동 할당 : 활성화
기타	- 추가적인 알림이 없다면 기본 옵션 사용

6. 스토리지 추가

- ✓ 스토리지는 [2.3 인스턴스 크기]를 참조하십시오.

7. 태그 추가

- 마스터 노드

메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion Master		

- 인프라 노드

메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion Infra		

- 워커 노드

메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion worker		

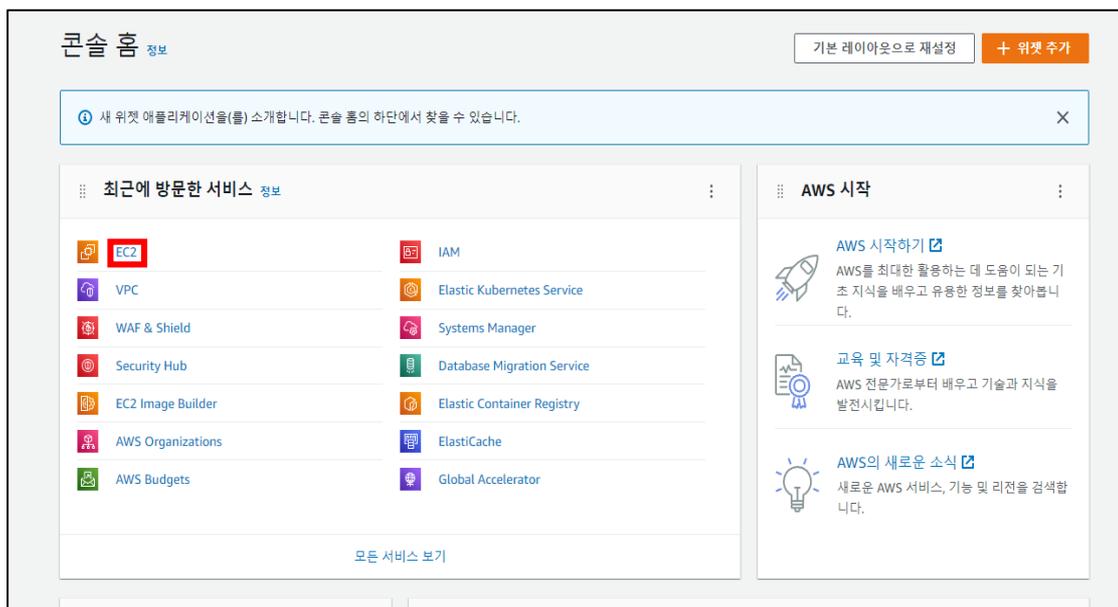
8. 보안 그룹 설정

- ✓ 기존 보안 그룹을 선택하십시오. [Accordion SG]

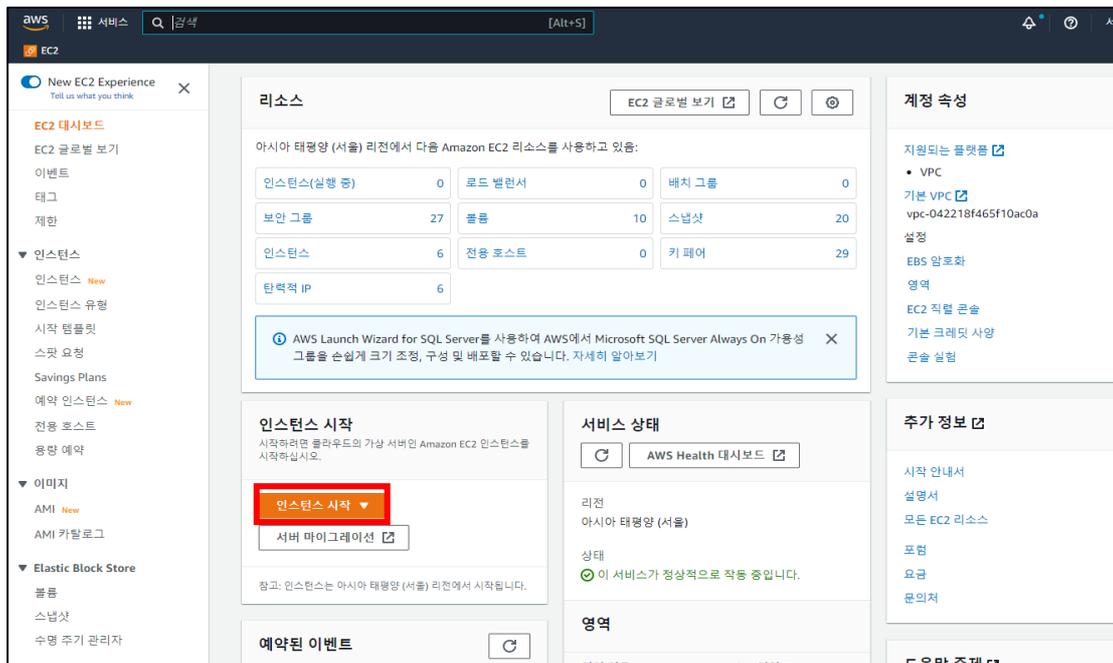
B. 싱글 구성 배포

가. Bastion AMI

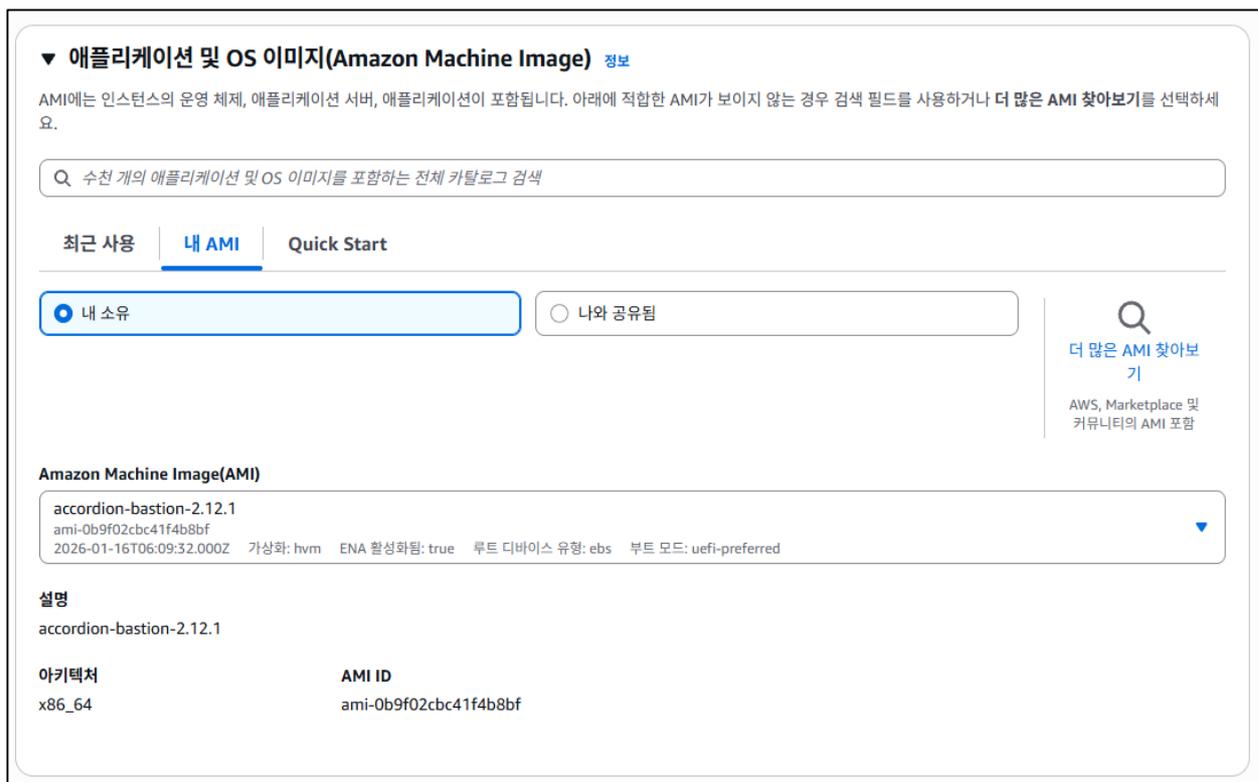
1. AWS 관리 콘솔에 로그인하고 EC2를 클릭합니다.



2. 인스턴스 시작을 클릭합니다.



3. 공유 AMI를 사용하여 인스턴스를 생성합니다.



4. 인스턴스 타입을 선택합니다.

✓ 인스턴스 타입은 [2.3 인스턴스 크기]를 참조하십시오.

5. 인스턴스 세부 정보를 설정합니다.

메뉴	값
구성 세부사항	<ul style="list-style-type: none"> - 인스턴스 수 : 1 - 네트워크 : VPC에 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 서브넷 : 서브넷에 대한 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 퍼블릭 IP 자동 할당 : 활성화
기타	- 추가적인 알림이 없다면 기본 옵션 사용

6. 스토리지 추가

✓ 스토리지는 [2.3 인스턴스 크기]를 참조하십시오.

7. 태그 추가

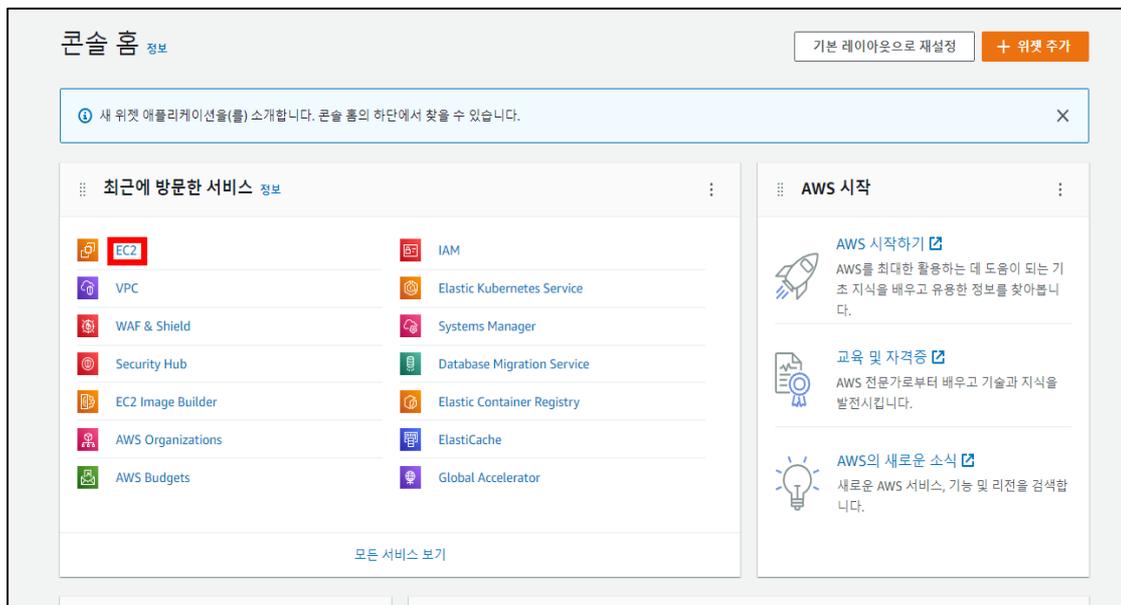
메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion Mastser		

8. 보안 그룹 설정

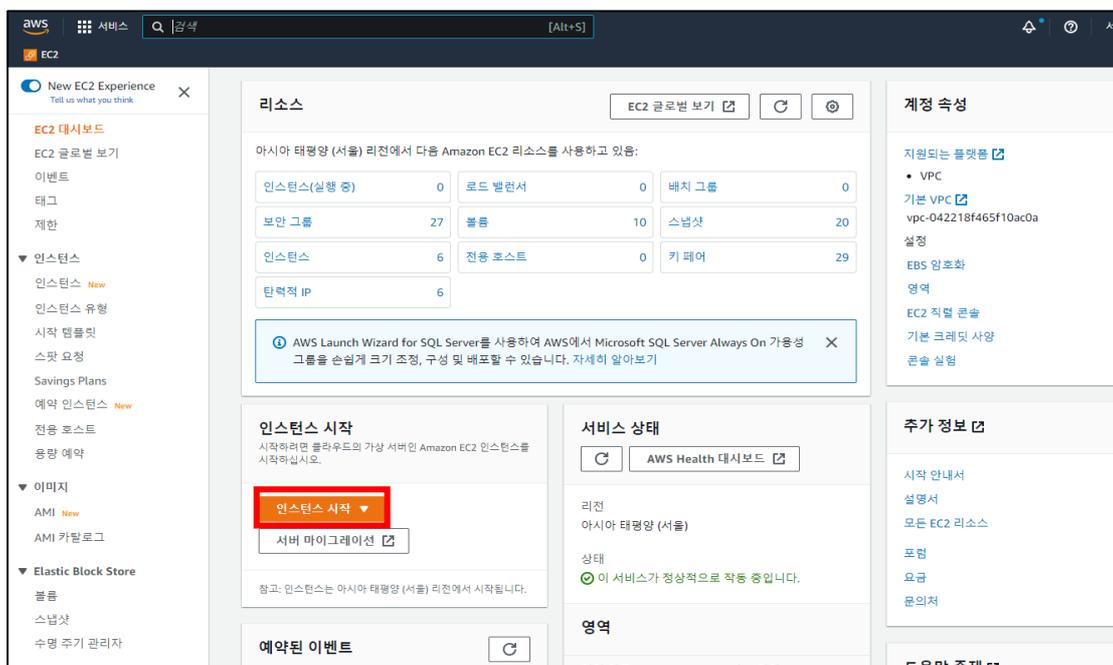
✓ 기존 보안 그룹을 선택하십시오. [Accordion SG]

다. Node AMI

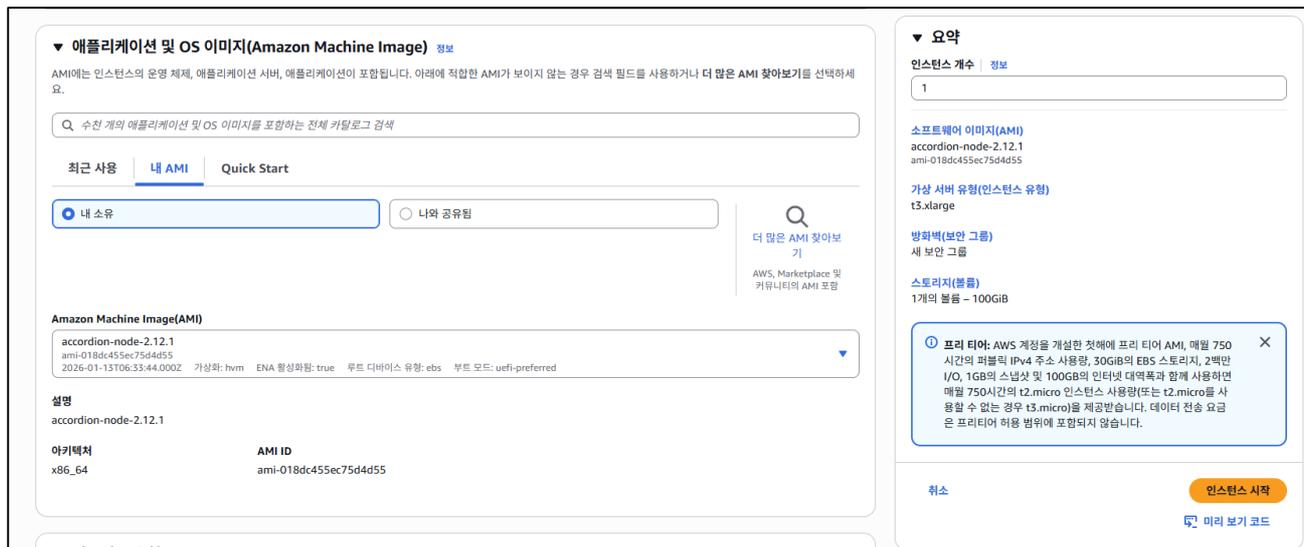
1. AWS 관리 콘솔에 로그인하고 EC2를 클릭합니다.



2. 인스턴스 시작을 클릭합니다.



3. 공유 AMI를 사용하여 인스턴스를 생성합니다.



4. 인스턴스 타입을 선택합니다.

- 사용하고자 하는 노드 타입의 따라 인스턴스 타입을 지정합니다.
- ✓ 인스턴스 타입은 [2.3 인스턴스 크기]를 참조하십시오.

5. 인스턴스 세부 정보를 설정합니다.

- 워커 노드

메뉴	값
구성 세부사항	<ul style="list-style-type: none"> - 인스턴스 수 : 2 - 네트워크 : VPC에 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 서브넷 : 서브넷에 대한 자세한 내용은 다음 링크를 참조 VPC 작업 - Amazon Virtual Private Cloud - 퍼블릭 IP 자동 할당 : 활성화
기타	- 추가적인 알림이 없다면 기본 옵션 사용

6. 스토리지 추가

- ✓ 스토리지는 [2.3 인스턴스 크기]를 참조하십시오.

7. 태그 추가

- 워커 노드

메뉴	값	메뉴	값
태그 키	Name	설명	식별하기 위한 태그 지정
값	Accordion worker		

8. 보안 그룹 설정

- ✓ 기존 보안 그룹을 선택하십시오. [Accordion SG]

3.2 2단계, 아코디언 설치 세팅

설치 세팅 및 설치에 대해서는 모두 'accordion' 일반유저로 진행합니다.

3.2.1 Bastion 인스턴스

A. 시스템 환경 설정

1. 시스템 호스트 이름을 변경합니다.(①~②)
2. 설치를 위한 ssh key를 생성합니다.(③)
3. Bastion 인스턴스를 포함한 모든 인스턴스의 키를 배포합니다.(④)
4. 키 배포가 정상적으로 성공했는지 확인합니다.(⑤)
5. 모든 인스턴스의 대한 hostname 정보를 hosts의 기입합니다.

```

① [accordion@localhost ~]sudo hostnamectl set-hostname acc-master
② [accordion@localhost ~] hostname
acc-master
③ [accordion@localhost ~] ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/accordion/.ssh/id_rsa): [enter]
Created directory '/home/accordion/.ssh'.
Enter passphrase (empty for no passphrase): [enter]
Enter same passphrase again: [enter]

```

```

Your identification has been saved in /home/accordion/.ssh/id_rsa.
Your public key has been saved in /home/accordion/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:uxvbkM90NbZofE4XlhNb8nNek+MlldKa5+XxJh3oEVo    accordion@ip-10-0-90-112.ap-northeast-
2.compute.internal
The key's randomart image is:
+---[RSA 3072]-----+
|
| . |
| .ooE. |
| +O.o.|
| S o=. *BBo|
| o ++ *BBB|
| = . *.*o=|
| X o + + |
| +. + . |
+----[SHA256]-----+
④ [accordion@localhost ~] sudo ssh-copy-id -i /home/accordion/.ssh/id_rsa.pub accordion@[노드 IP]
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/accordion/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already
installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the
new keys
accordion@10.140.0.3's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'accordion@10.0.99.160'"
and check to make sure that only the key(s) you wanted were added.
⑤ [accordion@localhost ~] ssh accordion@[노드 IP]
⑥ [accordion@localhost ~] sudo vi /etc/hosts
10.140.0.2 acc-master
10.140.0.3 acc-node1
...
10.140.0.8 acc-infra2

```

B. 앤서블 인벤토리 설정

- 고가용성 구성 배포

1. /mantech 디렉토리에 설치 스크립트를 옮깁니다. (①)

2. /mantech/accordion-installer/hosts 파일을 vi 편집기로 수정합니다. (2)

```

① [accordion@localhost ~] mv accordion-installer rpms /mantech
② [accordion@localhost mantech] vi /mantech/accordion-installer/hosts
=====
# host-cluster List
=====
acc-host-master ansible_host=10.140.0.2 ansible_connection=ssh node_role=infra
acc-host-node1 ansible_host=10.140.0.3 ansible_connection=ssh node_role=infra
acc-host-node2 ansible_host=10.140.0.4 ansible_connection=ssh node_role=infra
acc-host-master2 ansible_host=10.140.0.5 ansible_connection=ssh node_role=infra
acc-host-master3 ansible_host=10.140.0.6 ansible_connection=ssh node_role=infra
acc-host-infra1 ansible_host=10.140.0.7 ansible_connection=ssh node_role=infra
acc-host-infra2 ansible_host=10.140.0.8 ansible_connection=ssh node_role=infra

...

# Manager Group

[host-master]
acc-host-master

[host-master-cluster]
acc-host-master2
acc-host-master3

[host-minions]
acc-host-node1
acc-host-node2
[host-infra]
acc-host-infra1
acc-host-infra2

[host-etcd]

...

[all:vars]
ansible_ssh_port=22
ansible_user=accordion
ansible_group=accordion

```

```

ansible_password_option=no
#ansible_ssh_pass=ROOT_PASSWORD
keyfile_option=no
#ansible_ssh_private_key_file=/mantech/accordion.pem
ansible_ssh_common_args=-oPubkeyAuthentication=yes

```

- 싱글 구성 배포

1. /mantech 디렉토리에 설치 스크립트를 옮깁니다. (①)
2. /mantech/accordion-installer/hosts 파일을 vi 편집기로 수정합니다. (②)

```

① [accordion@localhost ~] mv accordion-installer rpms /mantech
② [accordion@localhost mantech] vi /mantech/accordion-installer/hosts
=====
# host-cluster List
=====
acc-host-master ansible_host=10.140.0.2 ansible_connection=ssh node_role=infra
acc-host-node1 ansible_host=10.140.0.3 ansible_connection=ssh node_role=infra
acc-host-node2 ansible_host=10.140.0.4 ansible_connection=ssh node_role=infra
...

# Manager Group

[host-master]
acc-host-master

[host-master-cluster]

[host-minions]
acc-host-node1
acc-host-node2
...

[all:vars]
ansible_ssh_port=22
ansible_user=accordion
ansible_group=accordion
ansible_password_option=no

```

```
#ansible_ssh_pass=ROOT_PASSWORD
keyfile_option=no
#ansible_ssh_private_key_file=/mantech/accordion.pem
ansible_ssh_common_args=-oPubkeyAuthentication=yes
```

C. 앤서블 설치 변수 지정

- 고가용성 구성 배포

1. /mantech/accordion-installer/group_vars/hosts 파일을 vi 편집기로 수정합니다.

```
① [accordion@localhost ~] vi /mantech/accordion-installer/group_vars/hosts
#- setup variable for cluster installation

#####
## Master configuration
#####
# cluster_name
cluster_name: "host-cluster"
# master isolation ( yes / no )
master_isolation: "yes"
master_host_name: "acc-master"
master_ip: 10.140.0.2

acc_interface: ens192
#####
# 3master mode( yes / no )
#####
master_mode: "yes"
master2_ip: 10.140.0.5
master3_ip: 10.140.0.6
master2_hostname: "acc-master2"
master3_hostname: "acc-master3"
# LB_mode ( L4 / haproxy )
LB_mode: "haproxy"
LB_port: 8443LB_vip: 10.140.0.10

# LB_domain_option ( yes / no )
LB_domain_option: "no"
```

```
LB_domain_address: "l4.domain.accordions.co.kr"
```

```
# Set it up if you want to add the master server later
```

```
# If master_mode is "no", it will not work.
```

```
# single_option( yes / no )
```

```
single_option: "no"
```

```
...
```

```
# nfs_setup ( internal / external )
```

```
nfs_setup: "internal"
```

```
nfs_server_ip: 10.140.0.2
```

- 싱글 구성 배포

1. /mantech/accordion-installer/group_vars/hosts.yml 파일을 vi 편집기로 수정합니다.

```
① [accordion@localhost ~] vi /mantech/accordion-installer/group_vars/hosts.yml
```

```
##- setup variable for cluster installation
```

```
#####
```

```
## Master configuration
```

```
#####
```

```
# cluster_name
```

```
cluster_name: "host-cluster"
```

```
# master isolation ( yes / no )
```

```
master_isolation: "yes"
```

```
master_host_name: "acc-master"
```

```
master_ip: 10.140.0.2
```

```
acc_interface: ens192
```

```
#####
```

```
# 3master mode( yes / no )
```

```
#####
```

```
master_mode: "no"
```

```
...
```

```
# nfs_setup ( internal / external )
```

```
nfs_setup: "internal"
```

```

nfs_server_ip: 10.140.0.2
...
#####
# base registry
#####
# accordion_registry_option ( local / external )
base_registry_option: "local"
base_registry_addr: 10.140.0.10
...
#####
# user registry
#####
# registry_option ( registry / harbor )
user_registry_option: "registry"
user_registry_addr: 10.140.0.10

```

3.2.2 Node 인스턴스

A. 시스템 환경 설정

1. 시스템 호스트 이름을 변경합니다.(①~②)
2. 모든 인스턴스의 대한 hostname 정보를 hosts의 기입합니다.(③)

```

① [accordion@localhost ~]sudo hostnamectl set-hostname acc-node1
② [accordion@localhost ~] hostname
acc-node1
③ [accordion@localhost ~] sudo vi /etc/hosts
10.140.0.2 acc-master
10.140.0.3 acc-node1
...
10.140.0.8 acc-infra2

```

3.3 3단계, 아코디언 설치

1. /mantech/rpms/dpkg22/0_ansible 으로 이동해 install.sh 를 실행하여 ansible을 설치합니다. (①)
2. /mantech/accordion-installer 로 이동하여 install.sh를 실행합니다. (②)
3. Status가 Ready 또는 Running로 표시되는지 확인합니다.. (③)

```

① [accordion@localhost ~] cd /mantech/rpms/dpkg22/0_ansible && sudo sh ./install.sh
② [accordion@localhost ~] cd /mantech/accordion-installer && sudo sh ./install.sh
③ [accordion@localhost accordion-installer] sh ./status.sh
=====
nodes
=====
NAME          STATUS  ROLES                AGE   VERSION
host1-master  Ready  control-plane,master  9m   v1.23.8
host1-node1   Ready  worker                9m   v1.23.8
host1-node2   Ready  worker                9m   v1.23.8
=====
deployments/po/svc
=====
...

```

4. 운영 지침

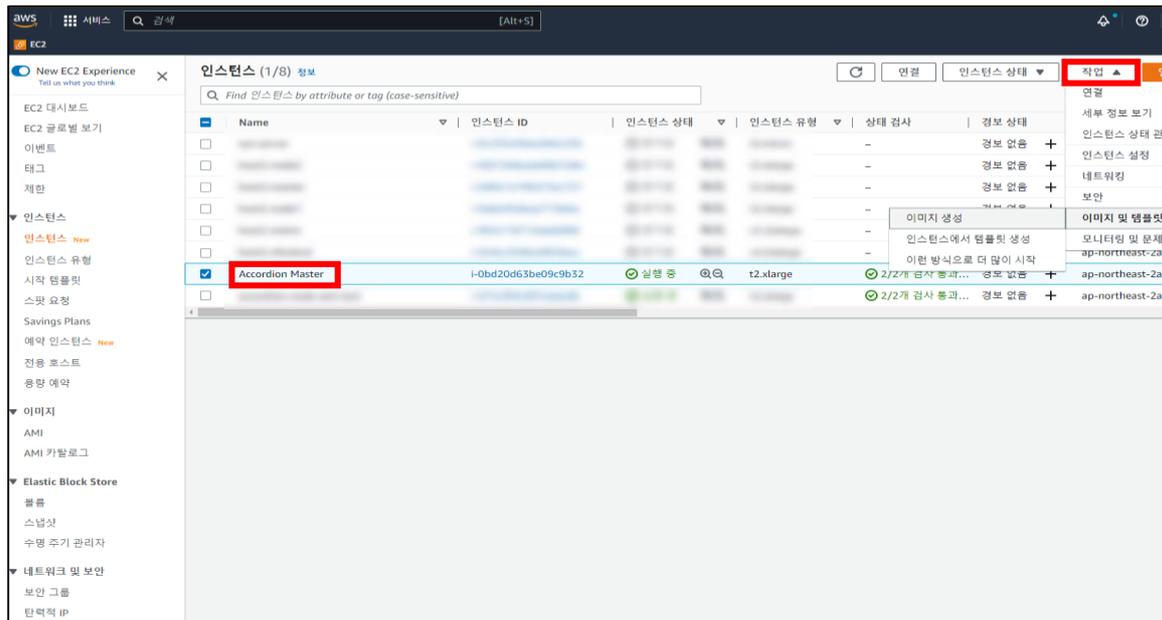
4.1 아코디언 백업과 복구 지원

4.1.1 AWS에서 아코디언 백업과 복구

[다이어그램]

A. 백업(스냅샷)

1. Accordion Master의 AMI를 생성합니다.

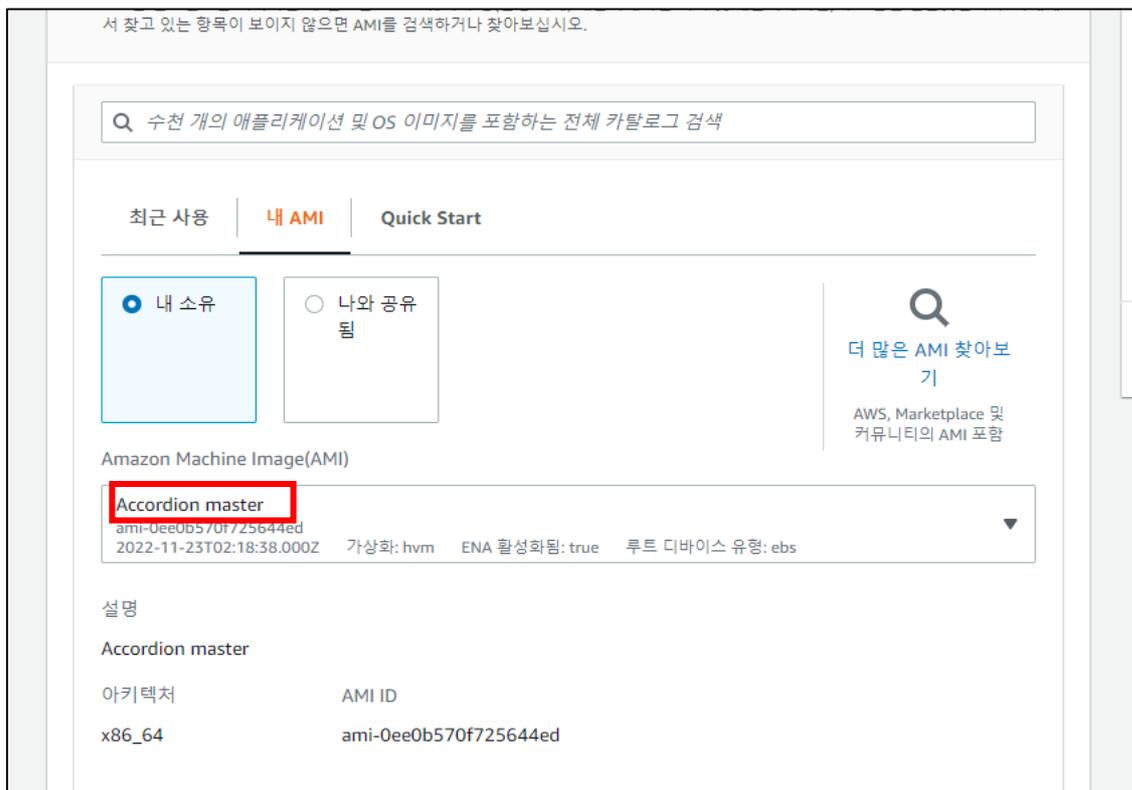


2. 이미지를 생성합니다.

이름	값
이미지 이름	Accordion master
이미지 설명	Accordion master
재부팅 안 함	미체크
인스턴스 볼륨	기본 설정

B. 복구

1. Accordion Master 이미지(AMI)를 생성합니다.



2. [3.1.5 인스턴스 생성을] 참조하여 AMI(snapshot)을 생성합니다.

4.2 정기적인 유지보수

최신 릴리스 및 기술 지원 서비스를 사용하여 제품을 최대한 활용할 수 있습니다.

유지보수 비용은 개발자의 정책에 따라 결정되며 최신 릴리스 개발 및 업그레이드와 관련된 서비스를 포함합니다.

유지보수 및 기술지원의 세부사항은 사용권 계약에 따라 달라질 수 있습니다.

유지보수는 크게 다음과 같이 나뉩니다.

- 정기 유지보수: 정기적인 유지보수는 유지보수 계약에 따라 수행한다.

- 비상 유지 관리: 유지보수 계약에 따라 비상 유지보수를 수행한다.

유지보수 범위는 다음과 같습니다.

- 제품 정기 점검
- 인증서 관리
- 패치 및 업그레이드

4.3 비상 유지보수

4.3.1 kubelet 프로세스 시작

1) kubelet 프로세스

A. 시작

순서	명령어	설명
1	-	ssh로 원격 접속을 합니다.
2	systemctl start kubelet	kubelet 프로세스를 시작합니다.

B. 중지

순서	명령어	설명
1	-	ssh로 원격 접속을 합니다.
2	systemctl stop kubelet	kubelet 프로세스를 중지합니다.

4.3.2 상태 점검

1) 주요 프로세스 점검

A. 아코디언 파드 점검

순서	명령어	설명
1	-	ssh로 원격 접속을 합니다.
2	kubectl get pod -n acc-global grep -v 'Running'	acc-global에서 'Running' 상태가 아닌 파드 조회합니다.
3	kubectl get pod -n acc-system grep -v 'Running'	acc-system에서 'Running' 상태가 아닌 파드 조회합니다.

정상적인 예1)

- 비정상적인 파드가 조회되지 않습니다.

```
[root@host1-master ~]# k get pod -n acc-global | grep -v 'Running'
NAME                READY   STATUS    RESTARTS   AGE
[root@host1-master ~]#
```

정상적인 예2)

- 비정상적인 파드가 조회되지 않습니다.

```
[root@host1-master ~]# k get pod -n acc-system | grep -v 'Running'
NAME                READY   STATUS    RESTARTS   AGE
[root@host1-master ~]#
```

B. 쿠버네티스 주요 파드 점검

순서	명령어	설명
1	-	ssh로 원격 접속을 합니다.
2	kubectl get pod -n kube-system grep -v 'Running'	kube-system 에서 'Running' 상태가 아닌 파드 조회합니다.

정상적인 예1)

- 비정상적인 파드가 조회되지 않습니다.

```
[root@host1-master ~]# k get pod -n kube-system | grep -v 'Running'
NAME                READY   STATUS    RESTARTS   AGE
[root@host1-master ~]#
```

C. kubelet 프로세스 점검

순서	명령어	설명
1	-	ssh로 원격 접속을 합니다.
2	systemctl status kubelet	kubelet 프로세스를 점검합니다.

정상적인 예1)

```
[root@host1-master ~]# systemctl status kubelet
● kubelet.service - kubelet: The Kubernetes Node Agent
   Loaded: loaded (/usr/lib/systemd/system/kubelet.service; enabled; vendor preset: disabled)
   Drop-In: /etc/systemd/system/kubelet.service.d
            └─10-kubeadm.conf
   Active: active (running) since ㄱ 2022-10-21 14:50:01 KST; 1 months 2 days ago
     Docs: https://kubernetes.io/docs/
   Main PID: 7164 (kubelet)
    Tasks: 27
   Memory: 145.2M
```

D. 아코디언 노드 점검

순서	명령어	설명
1	-	ssh로 원격 접속을 합니다.
2	kubectl get node	노드 상태를 조회합니다.

정상적인 예1)

- STATUS가 'Ready' 상태로 표시됩니다.

```
[root@host1-master ~]# kubectl get node
NAME                STATUS    ROLES
host1-master        Ready    control-plane,master
host1-node1         Ready    worker
host1-node2         Ready    worker
[root@host1-master ~]#
```

4.3.3 아코디언 오류 유형

- 서버 리소스 부족 - EBS 용량
- 네트워크 방화벽으로 인한 차단

4.3.4 아코디언 오류 복구 절차

Case1. 서버 리소스 부족 - EBS 용량

1. 용량 최적화

"/" 가 100%인 경우 조치가 필요합니다.

불필요한 데이터를 검토해 삭제합니다.

Case2. 네트워크 방화벽으로 인한 차단

1. 네트워크와 보안 그룹을 체크합니다.

소스	목적지	포트	목적
Accordion SG	Accordion SG	6443	Kubernetes API server
Accordion SG	Accordion SG	8443	Kubernetes API server(마스터노드 3중화)
Accordion SG	Accordion SG	2379~2380	etcd server client API
Accordion SG	Accordion SG	10250	Kubelet API

Accordion SG	Accordion SG	10251	kube-scheduler
Accordion SG	Accordion SG	10255	kube-controller-manager
Accordion SG	Accordion SG	179	Calico networking (BGP)
Accordion SG	Accordion SG	4789	Calico networking with VXLAN enabled
워커노드	마스터노드	5473	Calico networking with Typha enable
워커노드	마스터노드	9100	node exporter
워커노드	마스터노드	5000	Accordion local Infra-registry
클라이언트	마스터노드	30000~32767	NodePort Services

2. 정상화되었는지 확인합니다.

4.3.5 아코디언 복구 실패 시 복구 절차

복구 실패 시 스냅샷이 여부에 따라 복구하거나 다시 설치할 수 있습니다.

- A. 기존 Accordion의 스냅샷을 사용하여 AMI 재생성
- B. 아코디언 재설치
 - [3.1.3 인스턴스 생성] 절차에 따라 재설치

4.4 RTO

단일 구성이 배포되면 관리자 오류가 발생하면 RTO가 발생합니다. 기존 설치 관리자의 스냅샷을 사용하여 AMI를 재생성하고 AMI를 다시 설치해야 합니다. 최소 10분에서 최대 30분까지 발생합니다.

4.5 AWS 서비스 한도 관리

AWS 는 AWS 리소스의 가용성을 보장하고 필요 이상으로 많은 리소스를 실수로 프로비저닝하는 것을 방지하기 위해 각 계정에 대한 서비스 할당량(이전의 서비스 한도)을 유지합니다.

- A. 서비스 할당량 증가

Service Quotas 콘솔을 사용하여 대부분의 AWS 할당량 증가를 확인하고 요청할 수 있습니다.

중요: 대부분의 서비스 할당량은 AWS 리전에만 해당합니다. 할당량 증가가 필요한 AWS 리전을 선택합니다.

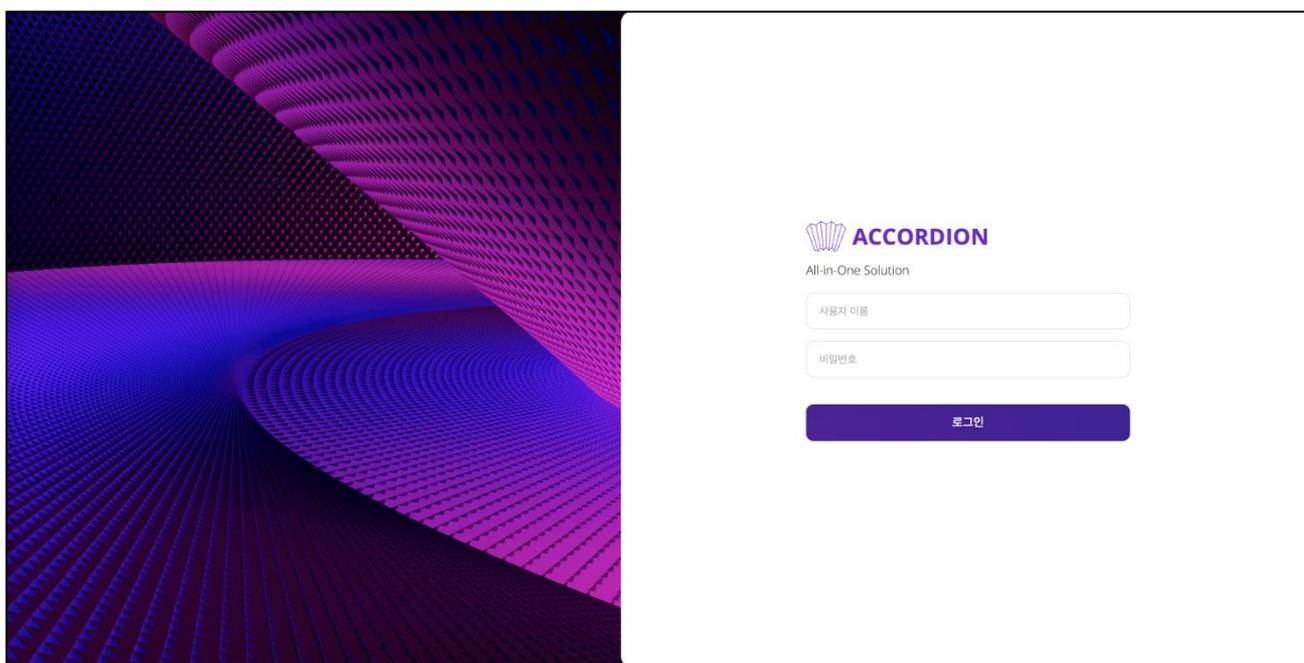
다른 모든 할당량 증가 요청의 경우 AWS Support 콘솔에서 사례를 생성한 다음 서비스 한도 증가를 선택합니다.

참고: 서비스 할당량이 증가해도 요금은 부과되지 않습니다. AWS 리소스 또는 서비스를 시작하거나 사용하는 경우에만 요금이 부과됩니다.

5. 시스템 관리

5.1 로그인

관리자 PC에서 웹 브라우저(파이어폭스, 크롬, Internet Explorer 10 이상)를 실행합니다. 그런 다음 웹 브라우저의 주소 창에 'https://<아코디언 마스터 퍼블릭 IP>:30000'를 입력하고 [Enter]를 누르거나 [로그인] 버튼을 클릭합니다.



아코디언은 미리 정해진 관리자 계정과 초기 암호를 가지고 있습니다.

메뉴	설명
사용자 이름	사용자 이름은 'admin'으로 입력합니다.
비밀번호	초기 암호는 'kubectl get secret init-secret -n acc-global -o jsonpath='{.data.init-key}' base64 -d' 명령어를 통해 확인할 수 있습니다. 처음 로그인 후 반드시 변경해야 합니다.

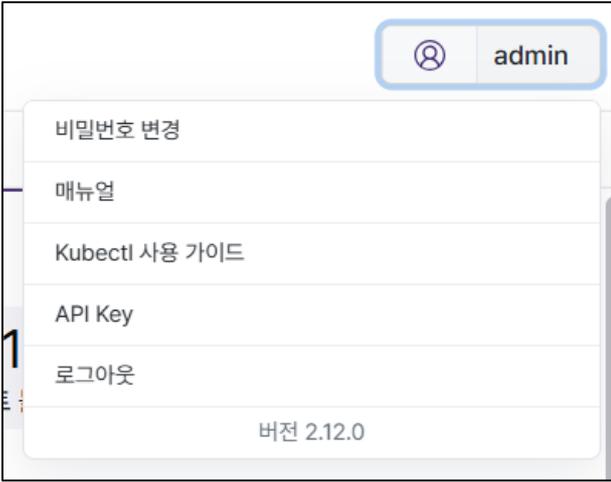
최초 로그인 시, 암호를 변경할 수 있는 화면으로 이동합니다. 새 암호를 두 번 입력하여 변경합니다. 암호는 8자리 이상이어야 하며 특수 문자 및 숫자를 포함해야 합니다.

5.2 화면 구성

화면은 다음과 같이 3개의 영역으로 나누어집니다.

영역	설명
헤더 영역	화면의 상단에 위치한다. 스코프(클러스터/네임스페이스) 를 선택할 수 있고 사용자 정보가 표시된다.
메뉴 영역	화면의 좌측에 위치한다. 각 스코프(글로벌/클러스터/네임스페이스) 에 대한 메뉴를 표시한다. 메뉴는 사용자 권한에 따라 사용가능한 항목이 표시된다.
이벤트 영역	화면의 하단에 위치하여 멀티 클러스터에서 발생하는 이벤트를 표시한다.
메인 영역	각 메뉴에 대한 화면을 표시한다. 메뉴별 정보를 표시하고 기능을 설정, 제어, 모니터링 할 수 있다.

5.2.1 헤더영역

기능	설명
클러스터	네임스페이스, 노드 관리와 같이 클러스터 스코프에서 필요한 작업을 수행하기 위해 클러스터를 선택할 수 있다.
네임스페이스	애플리케이션, 빌드와 같이 네임스페이스 스코프에서 필요한 작업을 수행하기 위해 네임스페이스를 선택할 수 있다.
사용자 설정	<p>사용자 비밀번호 설정 및 로그아웃을 수행하거나 매뉴얼 조회 등을 할 수 있다.</p> <ul style="list-style-type: none"> - 비밀번호 변경: 사용자 비밀번호를 변경 할 수 있다. - 매뉴얼: 아코디언 매뉴얼을 확인 할 수 있다. - Kubectl 사용 가이드: kubectl 다운로드 링크 및 설정 파일을 받을 수 있다. - 로그아웃: 사용자 로그아웃을 한다. 

비밀번호 변경을 선택하면 사용자의 비밀번호를 재설정할 수 있다.

비밀번호 변경

현재 비밀번호

새 비밀번호

비밀번호 확인

5.2.2 메뉴영역

메뉴는 스코프별로 분류되며 스코프는 글로벌, 클러스터, 네임스페이스 3개로 구분됩니다.

A. 글로벌 메뉴

메뉴	설명
글로벌 대시보드	멀티 클러스터 모니터링 대시보드로 멀티 클러스터와 리소스의 상태 정보를 조회합니다.
클러스터	호스트 클러스터 및 호스트 클러스터와 연결된 멤버 클러스터의 정보를 조회하고 관리합니다.
헬름	헬름 차트를 사용하기 위해 헬름 저장소를 조회하고 관리합니다.
계정	<p>아코디언 전반에 걸쳐 계정 정보를 관리합니다. 사용자, 그룹 및 그리고 글로벌/클러스터/네임 스페이스 스코프별 메뉴에 대한 권한을 관리하고 사용자/그룹과 글로벌 권한을 바인딩하여 글로벌 멤버로 설정한다. 그 외에 사용자의 접속 로그를 조회할 수 있습니다.</p> <ul style="list-style-type: none"> - 사용자: 사용자 정보를 조회, 관리합니다. - 그룹: 사용자 그룹 정보를 조회, 관리합니다. - 글로벌 권한: 글로벌 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리합니다. - 클러스터 권한: 클러스터 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리합니다.

	<ul style="list-style-type: none"> - 네임스페이스 권한: 네임스페이스 스코프의 메뉴에 대해 사용 권한을 집합하여 단일 권한으로 관리합니다. - 글로벌 멤버: 사용자/그룹과 글로벌 권한을 바인딩한다. 사용자 접속 로그: 사용자의 접속 로그를 조회합니다. - 사용자 접속 로그: 사용자의 접속 로그를 조회합니다.
글로벌 설정	<p>활성화 키와 알림 발송을 위한 메일 서버를 관리합니다.</p> <ul style="list-style-type: none"> - 활성화 키: 활성화 키 정보를 조회, 등록합니다. - 메일서버: 알림 발송을 위해 메일 서버를 관리합니다.

B. 클러스터 메뉴

클러스터 메뉴는 개별 클러스터 스코프에서 서비스 배포와 운영을 위한 메뉴를 제공합니다.

메뉴	설명
클러스터 대시보드	해당 클러스터의 리소스 상태 정보를 제공합니다.
네임스페이스	해당 클러스터의 네임스페이스 정보를 관리합니다.
노드	해당 클러스터를 구성하는 노드의 정보를 관리한다. 그리고 노드를 스케줄 대상에서 제외하거나 노드에 배포된 파드를 다른 노드로 이동시킬 수 있습니다.
애플리케이션	헬름을 이용해 애플리케이션을 클러스터에 배포합니다.
워크로드	<p>워크로드 리소스 정보를 관리하기 위한 대시보드를 제공하고 편집기 등으로 관리합니다.</p> <ul style="list-style-type: none"> - 워크로드 대시보드: 배포한 파드 정보를 차트로 표시합니다. - 파드: 파드 정보를 제공하고 관리한다. 컨테이너 로그 및 터미널을 이용해 접속할 수 있습니다. - 디플로이먼트/스태이트풀셋: 파드를 관리하는 리소스로 편집기로 관리하며 오토스케일 을 설정할 수 있습니다. - 데몬셋: 노드마다 파드를 실행하도록 제어합니다. - 잡/크론잡: 특정 작업을 일시적/주기적으로 실행할 수 있도록 관리합니다.
구성	<p>시스템과 애플리케이션 구성 및 설정에 필요한 정보를 관리합니다.</p> <ul style="list-style-type: none"> - 컨피드맵/시크릿: 일반 정보 또는 기밀성 정보를 저장합니다. - HPA/리미트레인저: 수평적 오토스케일 및 리소스 할당을 제한합니다.
네트워크	애플리케이션 연동을 위해 네트워크 구성에 필요한 리소스를 관리합니다.

	<ul style="list-style-type: none"> - 서비스: 애플리케이션을 네트워크 서비스로 노출합니다. - 인그레스: 서비스를 http 또는 https 경로로 노출합니다. - 네트워크폴리시: OSI 3 또는 4 계층에서 트래픽 흐름을 제어합니다.
스토리지	<p>스토리지 목록을 확인 할 수 있으며 YAML/생성/수정/삭제 할 수 있습니다.</p> <ul style="list-style-type: none"> - 퍼시스턴트볼륨: NFS 와 같이 스토리지 구현에 필요한 정보를 관리합니다. - 퍼시스턴트볼륨클레임: 스토리지에 대해 요청을 하는 리소스 정보를 관리합니다. - 스토리지클래스: 퍼시스턴트볼륨을 동적으로 배포합니다.
커스텀 리소스	쿠버네티스 커스텀 리소스를 관리합니다.
접근제어	<p>쿠버네티스의 역할 기반 접근 제어를 관리하고 구성합니다.</p> <ul style="list-style-type: none"> - 서비스어카운트: 주로 파드에 인증 정보를 제공한다. (사용자 역할) - 롤/클러스터롤: API나 리소스에 대한 권한을 설정합니다. - 롤바인딩/클러스터롤바인딩: 롤/클러스터롤을 사용자 또는 서비스어카운트 등에 연결합니다. - 파드시큐리티폴리시: 파드가 실행해야하는 조건 셋을 정의합니다. (쿠버네티스에 관련 설정이 사전에 적용되어 있어야 한다.)
모니터링	<p>시스템 상태 및 각종 로그와 알림정보에 대해서 조회합니다.</p> <ul style="list-style-type: none"> - 시스템: 클러스터 내 CPU, 메모리와 같은 리소스 사용량 정보를 제공합니다. - 이벤트 로그: 쿠버네티스 이벤트 로그를 조회합니다. - 컨테이너 로그: 쿠버네티스에 배포되었던 컨테이너에서 발생한 로그를 조회합니다. - 감사 로그: 클러스터 내 감사 로그를 조회합니다. - 서비스메시: 클러스터 내 서비스 간의 통신 정보를 표시합니다. - 알림: 클러스터 내에서 발생한 알림에 대해 알림 수단 별 알림 횟수 정보를 제공합니다.

C. 네임스페이스 메뉴

네임스페이스 메뉴는 네임스페이스 스코프에서 서비스의 배포와 운영을 위한 메뉴를 제공합니다.

메뉴 중 일부는 클러스터 메뉴와 중복됩니다. 중복되는 메뉴의 경우 동일한 기능에 대해 접근하는 스코프의 차이를 제외하고 대부분 동일한 기능을 제공합니다.

메뉴	설명
네임스페이스 대시보드	해당 네임스페이스의 리소스 상태 정보를 제공합니다.
애플리케이션	<p>헬름과 카탈로그를 이용해 애플리케이션을 클러스터에 배포합니다.</p> <ul style="list-style-type: none"> - 헬름: 헬름 차트로 애플리케이션을 배포합니다. 주로 빌드를 수행하지 않는 서비스형 애플리케이션을 배포합니다. - 카탈로그: 카탈로그로 애플리케이션을 배포합니다. 빌드 파이프라인을 지원합니다.
빌드	<p>애플리케이션 빌드에 필요한 정보를 제어하고 관리합니다. 컨테이너 이미지를 빌드할 수 있고 빌드에 필요한 파이프라인 또는 승인에 대해 설정합니다.</p> <ul style="list-style-type: none"> - 파이프라인: 태스크를 조합하여 빌드를 수행할 파이프라인을 작성합니다. - 승인: 파이프라인에서 발생한 승인 요청을 처리합니다. - 태스크 템플릿: 파이프라인 작성시 사용하는 태스크에 대한 템플릿을 관리합니다.
워크로드	클러스터 메뉴의 워크로드 메뉴와 동일한 기능을 네임스페이스 스코프에서 수행합니다.
구성	클러스터 메뉴의 구성 메뉴와 동일한 기능을 네임스페이스 스코프에서 수행합니다.
네트워크	클러스터 메뉴의 네트워크 메뉴와 동일한 기능을 네임스페이스 스코프에서 수행합니다.
스토리지	퍼시스턴트볼륨클레임 리소스 정보를 관리합니다.
접근제어	<p>쿠버네티스의 역할 기반 접근 제어를 관리하고 구성합니다.</p> <p>서비스어카운트: 주로 파드에 인증 정보를 제공한다. (사용자 역할) 룰: API나 리소스에 대한 권한을 설정합니다.</p> <p>롤바인딩: 룰을 사용자 또는 서비스어카운트 등에 연결합니다.</p>
모니터링	<p>시스템 및 애플리케이션 상태 및 각종 로그를 조회합니다.</p> <ul style="list-style-type: none"> - 시스템: 네임스페이스 내 CPU, 메모리와 같은 리소스 사용량 정보를 제공합니다. - 애플리케이션: 스카우터로 수집한 애플리케이션에 대한 메트릭 정보를 제공합니다. - 이벤트 로그: 네임스페이스에서 발생한 쿠버네티스 이벤트 로그를 조회합니다. - 컨테이너 로그: 네임스페이스에 배포되었던 컨테이너에서 발생한 로그를 조회합니다. - 감사 로그: 네임스페이스 내 감사 로그를 조회합니다.

설정	<p>네임스페이스를 사용하는 멤버를 구성합니다. 그리고 알림 정책을 세우고 알림을 위한 슬랙 정보를 설정합니다.</p> <ul style="list-style-type: none"> - 네임스페이스 멤버: 네임스페이스 권한과 사용자/그룹을 연결합니다. - 알림 정책: 네임스페이스 알림 발송 규칙 및 발송 주기/수신자 등을 설정합니다. - 슬랙: 슬랙을 통해 알림을 받을 수 있도록 슬랙 정보를 설정합니다.
----	--

5.2.3 이벤트영역

이벤트는 멀티 클러스터 쿠버네티스 실시간 이벤트 정보를 제공하며 기본으로 최소화 되어있으며 이벤트 영역을 클릭하면 상세 조회할 수 있다.

A. 이벤트 영역의 최소화

이벤트 영역에 우측 톨 팁으로 현재 실시간으로 발생한 이벤트 수를 확인 할 수 있다.



B. 이벤트 영역의 최대화

이벤트 영역의 최대화 시 이벤트 상세 확인 및 검색 할 수 있다.

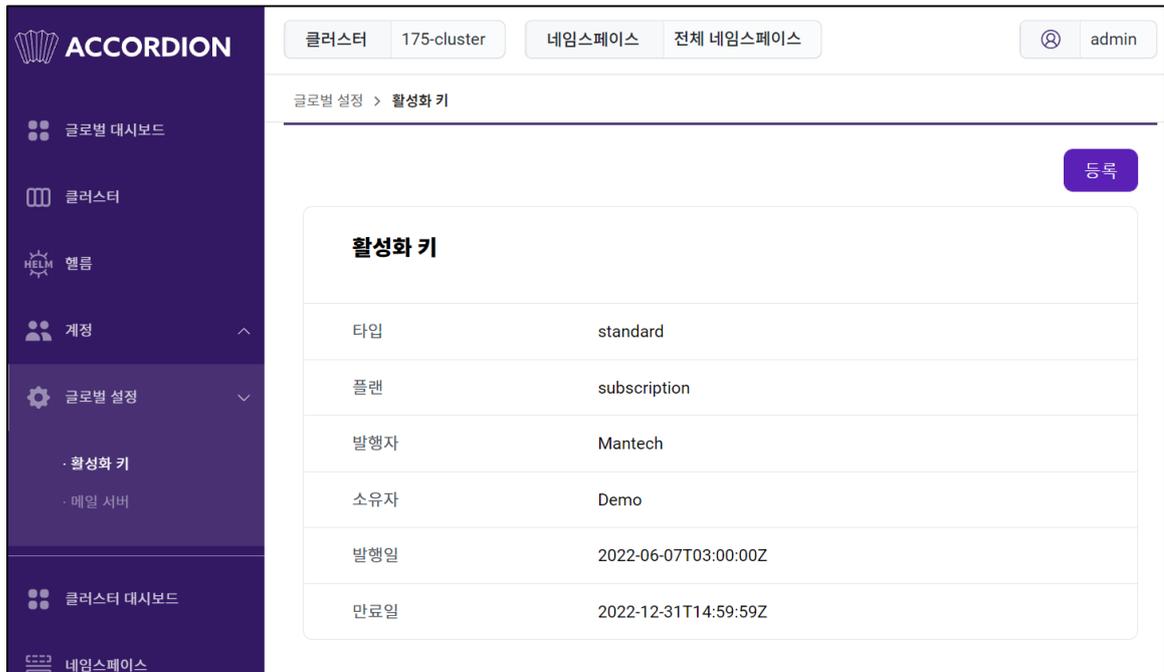
Age	클러스터	네임스페이스	타입	원인	오브젝트	개수	메세지
right now	host-cluster	mjtest	Normal	Pulling	pod/test-27500090-1-t2thk		Pulling image "busybox"
just now	host-cluster	mjtest	Normal	Scheduled	pod/test-27500090-1-t2thk		Successfully assigned mjtest/test-27500090-1-t2thk to gpu-accordion1
just now	host-cluster	mjtest	Normal	SuccessfulCreate	job/test-27500090		Created pod: test-27500090-1-t2thk
57 seconds ago	host-cluster	mjtest	Normal	Pulled	pod/test-27500089-1-rn6nt		Successfully pulled image "busybox" in 1.835204267s
57 seconds ago	host-cluster	mjtest	Normal	Created	pod/test-27500089-1-rn6nt		Created container hello

항목	설명
Age	이벤트가 발생한 시간
클러스터	클러스터 명
네임스페이스	네임스페이스 명
타입	<p>이벤트 타입</p> <ul style="list-style-type: none"> - Normal: 일반적인 작업으로 발생된 이벤트 - Warning: 오류에 의해 발생된 이벤트
원인	이벤트 발생 이유
오브젝트	오브젝트 정보
메세지	이벤트 상세 메시지

5.3 라이선스 관리

1. 라이선스 확인

글로벌 설정에서 활성화 키 메뉴에서 사용 중인 활성화 키를 확인하고 등록할 수 있습니다.



활성화 키	
타입	standard
플랜	subscription
발행자	Mantech
소유자	Demo
발행일	2022-06-07T03:00:00Z
만료일	2022-12-31T14:59:59Z

2. 라이선스 등록 및 편집

'활성화 키' 메뉴에서 키를 등록할 수 있습니다. 등록된 활성화 키는 삭제할 수 없습니다. '등록' 버튼 클릭 시 활성화 키 등록 팝업에 키를 입력하고 등록 버튼을 클릭합니다.

5.4 패치 및 업데이트 관리

패치/업데이트는 라이선스 계약에 따라 분기별로 자동 처리됩니다.

6. 지원

6.1 기술지원

기술지원의 범위는 매뉴얼 등 문서에 명시된 기능에 한해 제공됩니다.

기술 지원 범위는 다음과 같습니다.

- 장애/기술 지원
- 설치 지원: 설치 안내서 및 설치 매뉴얼

기술 지원 및 문의는 아래 메일 주소로 요청하십시오.

- oss-tech@mantech.co.kr

6.2 지원 비용

기술 지원은 라이선스 계약에 따라 제공됩니다.

6.3 SLA

SLA는 라이선스 계약에 따라 제공됩니다.