

# Rancher에 AKS 구축하여 Azure Pipeline으로 Application 빌드/배포하기

(그리고 Rancher에서 모니터링 환경 구성하기)

# Who am I?

김영대 / 제로빅 / zerobig

소속 및 직책

SPTek 스토어기술운영팀 팀장

주요 활동내역 및 역할

Kubernetes 문서 한글화팀 공식 Member (2018.12~ )

슬기로운 Azure 생활 공동 운영자 (2020.1~ )

Azure Solutions Architect Expert – 2019.10

Azure DevOps Engineer Expert – 2019.12

제로빅 블로그

<https://zerobig-k8s.tistory.com/>

제로빅 페이스북

<https://www.facebook.com/addicted2k8s>

제로빅 이메일

[zerobig.kim@gmail.com](mailto:zerobig.kim@gmail.com) zerobig@sptek.co.kr



## 사용하게 될 주요 서비스/솔루션



Azure DevOps

이전 Team Foundation Server(TFS)의 새로운 이름으로 버전제어 시스템, Application Lifecycle Management(ALM), CI/CD 파이프라인, Test 및 Artifact 등을 모두 통합한 웹 서비스



AKS

Azure에서 제공하는 가용성과 보안이 높은 완전 관리형 Kubernetes 서비스



ACR

Azure에서 제공하는 컨테이너 레지스트리



Kubernetes 통합관리 솔루션



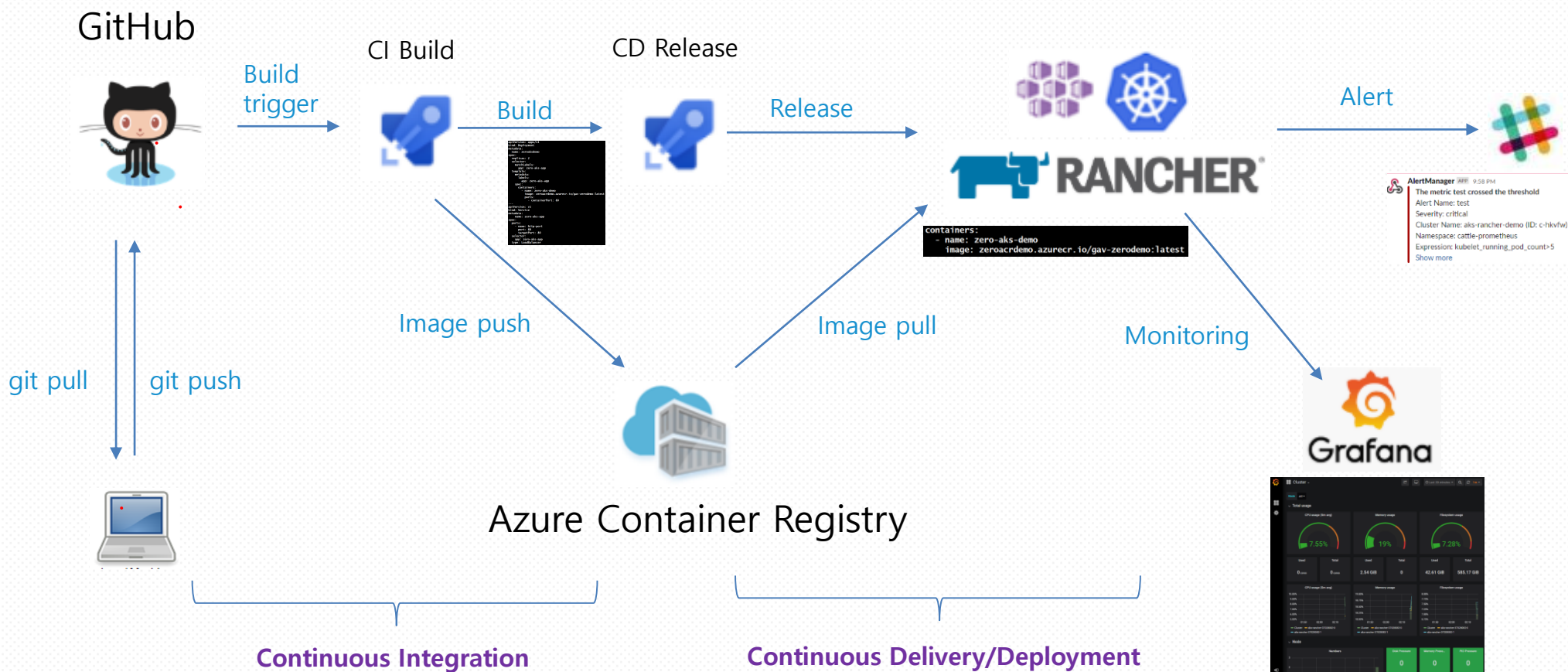
오픈소스 시각화 도구

# Topic

## Rancher에 AKS 구축하여 Azure Pipeline으로 Application 빌드/배포하기 (그리고 Rancher에서 모니터링 환경 구성하기)

### Azure Pipelines

### AKS on Rancher



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: zeroaksdemo
spec:
  replicas: 2
  selector:
    matchLabels:
      app: zero-aks-app
  template:
    metadata:
      labels:
        app: zero-aks-app
    spec:
      containers:
        - name: zero-aks-demo
          image: zeroacrdemo.azurecr.io/gav-zero-demo:latest
          ports:
            - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: zero-aks-app
spec:
  ports:
    - name: http-port
      port: 80
      targetPort: 80
  selector:
    app: zero-aks-app
  type: LoadBalancer
```

## 데모 진행 관련 참고 사항

√ 약 50분 내 많은 서비스/솔루션을 다루게 되므로 녹화방식으로 진행이 이루어 지며 개별적 Deep Dive 보다는 전체 하나의 큰 그림을 그려보는데 의미를 둠

√ 데모 진행과정에서 설치 및 동작 처리 등에 필요한 시간 동안 동영상 녹화를 중지 후 처리 완료 후 다시 재개하는 방식으로 진행됨

√ 데모의 간결함을 위해 보안관련 세밀한 설정 등은 배제됨

## 사전 요구 사항

실습 진행을 위해서는 다음과 같은 준비가 필요

- Azure 구독 : [Azure 체험 계정 만들기](#)
- Azure DevOps 계정 : [Azure DevOps 계정 만들기](#)
- Azure CLI : [Azure CLI 설치](#) (2.3.1)
- Git (2.17.1) : [Git 다운로드 및 설치](#)
- Docker (19.03.8) : [Docker 다운로드 및 설치](#)
- ASP.NET Core Runtime 3.0 : [다운로드 및 설치](#)



## 서비스 이미지 준비 및 ACR 생성하기

To Do

+ New item

612 Rancher 모니터링 환경 구축하기

zerobig.devops

State ● To Do

0/2

611 Azure DevOps 릴리즈 파이프라인 구성하기

zerobig.devops

State ● To Do

610 Rancher 및 AKS 설치하기

zerobig.devops

State ● To Do

609 Azure DevOps 빌드 파이프라인 구성하기

zerobig.devops

State ● To Do

Doing

605 서비스 이미지 준비 및 Azure Container Registry 생성하기

zerobig.devops

State ● Doing

0/2

+ Add Task

☒ Local 환경 Docker 빌드 및 이미지 검증하기

☐ ACR 생성 및 이미지 태그, 푸시하기

1/5

Done



# 서비스 이미지 준비 및 ACR 생성하기

## Local 환경 Docker 빌드 및 이미지 검증하기

# 소스 다운로드

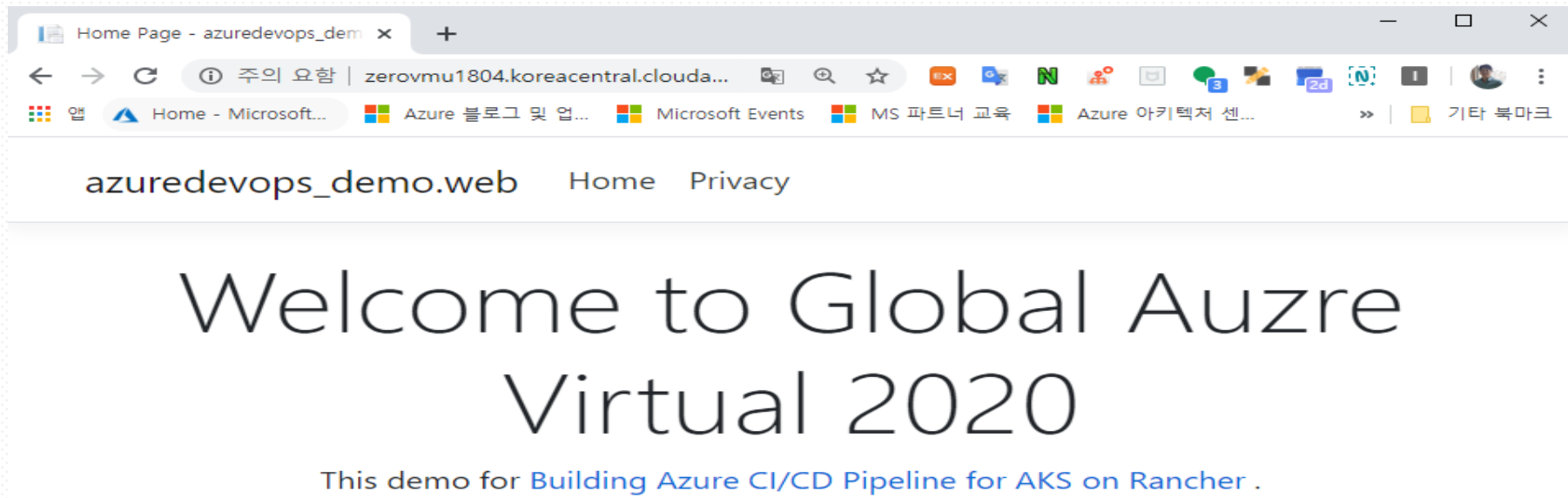
```
git clone https://github.com/zer0big/gav-zero demo.git
```

# 도커 빌드 : 이미지명은 각각 자신의 상황에 맞게 수정

```
docker build -t gav-zero demo .
```

# 생성 이미지 검증

```
docker run -d -p 80:80 gav-zero demo
```



## 서비스 이미지 준비 및 ACR 생성하기

### ACR 생성 및 이미지 태그, 푸시하기

```
# Azure Login  
az login
```

```
# 변수선언 : 각각 자신의 상황에 맞게 수정  
rgName=0demo4gav2020  
acrName=zeroacrdemo
```

```
# 리소스 그룹 생성  
az group create -n $rgName -l koreacentral
```

```
# ACR 생성  
az acr create -n $acrName -g $rgName --sku Standard
```

```
# ACR 로그인 및 결과 확인  
az acr login --name zeroacrdemo
```

```
root@zerovmu1804:~/gav-zero demo# az acr login --name zeroacrdemo  
Login Succeeded
```

# 서비스 이미지 준비 및 ACR 생성하기

## ACR 생성 및 이미지 태그, 푸시하기 (계속)

# 도커 태그 : 태그명은 각각 자신의 상황에 맞게 수정

```
docker tag gav-zero demo zeroacrdemo.azurecr.io/gav-zero demo
```

# 도커 푸시 to ACR

```
docker push zeroacrdemo.azurecr.io/gav-zero demo
```

Home > zeroacrdemo | Repositories > gav-zero demo

zeroacrdemo | Repositories  
Container registry

Search (Ctrl+ /)

Services

- Repositories
- Webhooks
- Replications
- Tasks

Repository permissions

Refresh

Search to filter repositories ...

Repositories ↑↓

- gav-zero demo

gav-zero demo  
Repository

Refresh Delete

|                   |                            |                |   |
|-------------------|----------------------------|----------------|---|
| Repository        | gav-zero demo              | Tag count      | 1 |
| Last updated date | 2020. 4. 20. 오후 3:10 GMT+9 | Manifest count | 1 |

Search to filter tags ...

Tags ↑↓

- latest

# Azure DevOps 빌드 파이프라인 구성하기

To Do

+ New item

612 Rancher 모니터링 환경 구축하기

zerobig.devops

State To Do

0/2

611 Azure DevOps 릴리즈 파이프라인 구성하기

zerobig.devops

State To Do

610 Rancher 및 AKS 설치하기

zerobig.devops

State To Do

Doing

609 Azure DevOps 빌드 파이프라인 구성하기

zerobig.devops

State Doing

1/5

Done

605 서비스 이미지 준비 및 Azure Container Registry 생성하기

zerobig.devops

State Done

2/2

+ Add Task

☒ ☒ Local 환경 Docker 빌드 및 이미지 검증하기

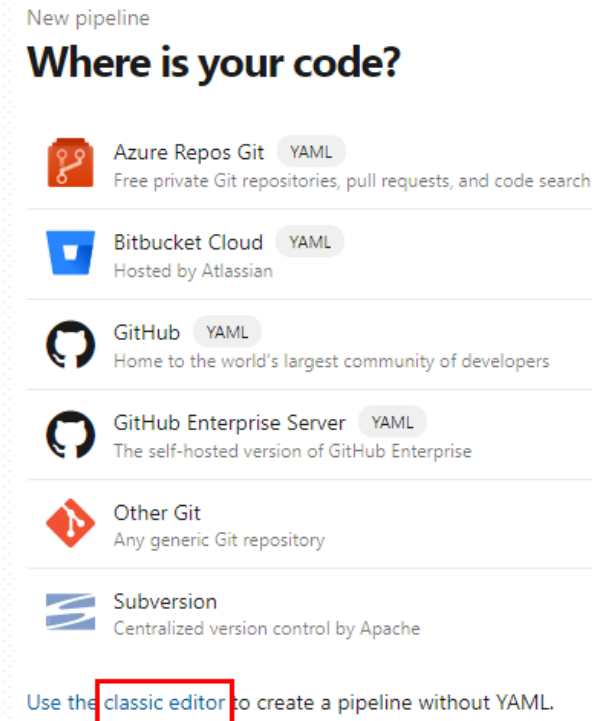
☒ ☒ ACR 생성 및 이미지 태그, 푸시하기

# Azure DevOps 빌드 파이프라인 구성하기 (계속)



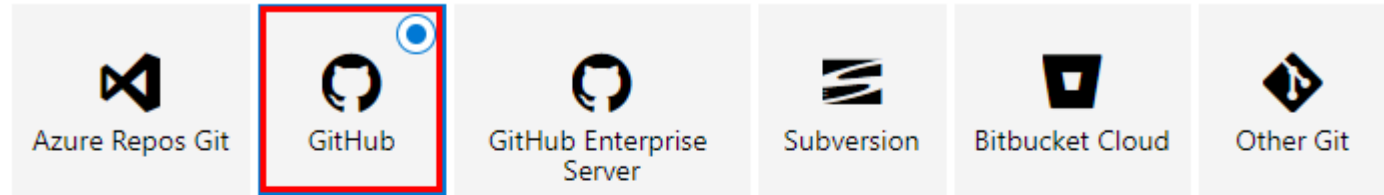
Pipeline – Pipelines 선택

Classic editor 선택



## Azure DevOps 빌드 파이프라인 구성하기 (계속)

Select a source



Authorized using connection: [GitHub connection 1](#) [Change](#) ▾

Repository \* | [Manage on GitHub](#) ↗

zer0big/gav-zero demo

...

Default branch for manual and scheduled builds \*

master


...

Continue


소스 위치 (GitHub) 및 Repository 선택 후 Continue 클릭

## Azure DevOps 빌드 파이프라인 구성하기 (계속)


Select a template

Or start with an  Empty job

Configuration as code

 **YAML**  
Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. [Learn more](#)

Others

 **ASP.NET Core**  
Build and test an ASP.NET Core web application.

**Apply**

템플릿 검색 창에 “ASP.NET Core” 입력 후 검색 결과 중 “ASP.NET Core” 선택  
Apply 클릭

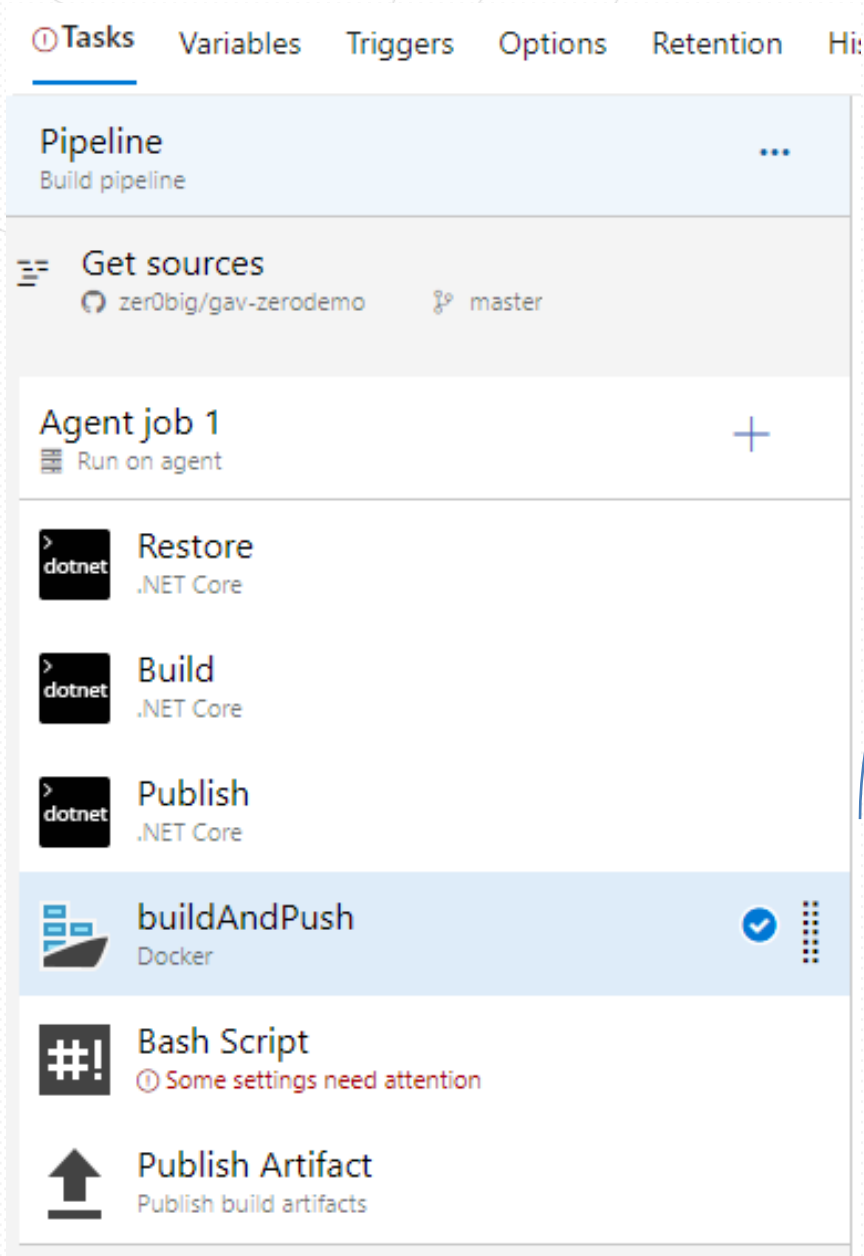


## Azure DevOps 빌드 파이프라인 구성하기 (계속)

The screenshot shows the Azure DevOps Pipeline editor. On the left, the 'Pipeline' view shows a build pipeline with the following steps: 'Get sources' (zer0big/gav-zero demo, master), 'Agent job 1' (Run on agent), 'Restore' (.NET Core), 'Build' (.NET Core), 'Publish' (.NET Core), and 'Publish Artifact' (Publish build artifacts). A red box highlights a '+' button next to 'Agent job 1'. On the right, the 'Add tasks' panel is open, showing a search bar with 'docker' entered. Below the search bar, the 'Docker' task is highlighted, with a red box around the 'Add' button. Other tasks listed include 'Docker CLI installer' and 'Docker Compose'.

+ 버튼을 눌러 "docker" 입력 후 검색 결과 중 "Docker" 선택 후 Add 클릭  
동일한 방식으로 "bash" 입력 후 검색 결과 중 "Bash" 선택 후 Add 클릭

# Azure DevOps 빌드 파이프라인 구성하기 (계속)



Tasks Variables Triggers Options Retention Hi

**Pipeline** ...  
Build pipeline

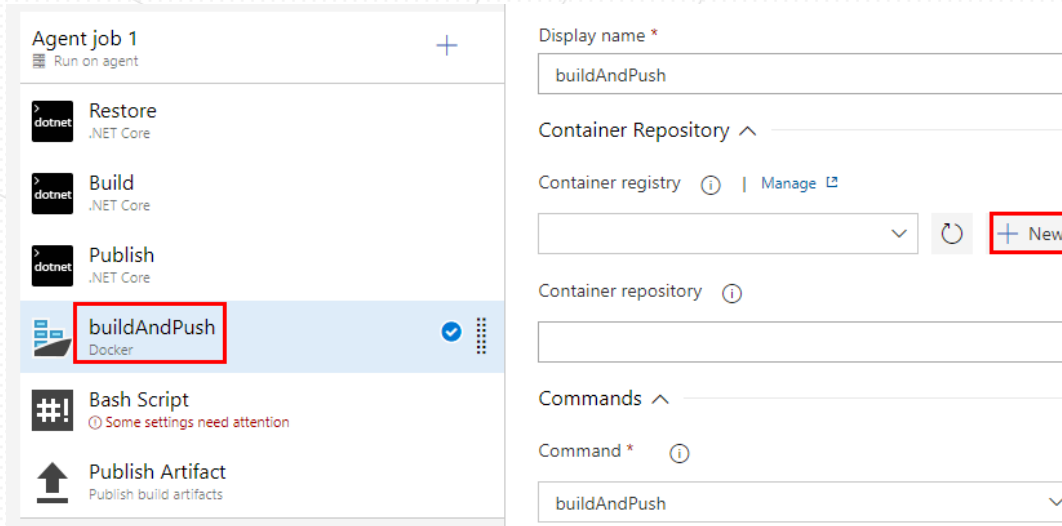
Get sources  
zer0big/gav-zerodemo master

Agent job 1  
Run on agent

- Restore  
dotnet .NET Core
- Build  
dotnet .NET Core
- Publish  
dotnet .NET Core
- buildAndPush  
Docker
- Bash Script  
Some settings need attention
- Publish Artifact  
Publish build artifacts

최종 다음과 같은 모습으로  
Task가 배열 (신속하고 간결한 데  
모를 위해 "Test"(.NET Core) 태스크는  
Removed 된 상태임)

## Azure DevOps 빌드 파이프라인 구성하기 (계속)



Agent job 1  
Run on agent

dotnet Restore .NET Core

dotnet Build .NET Core

dotnet Publish .NET Core

**buildAndPush** Docker

Bash Script  
Some settings need attention

Publish Artifact  
Publish build artifacts

Display name \*  
buildAndPush

Container Repository ^

Container registry ⓘ | Manage

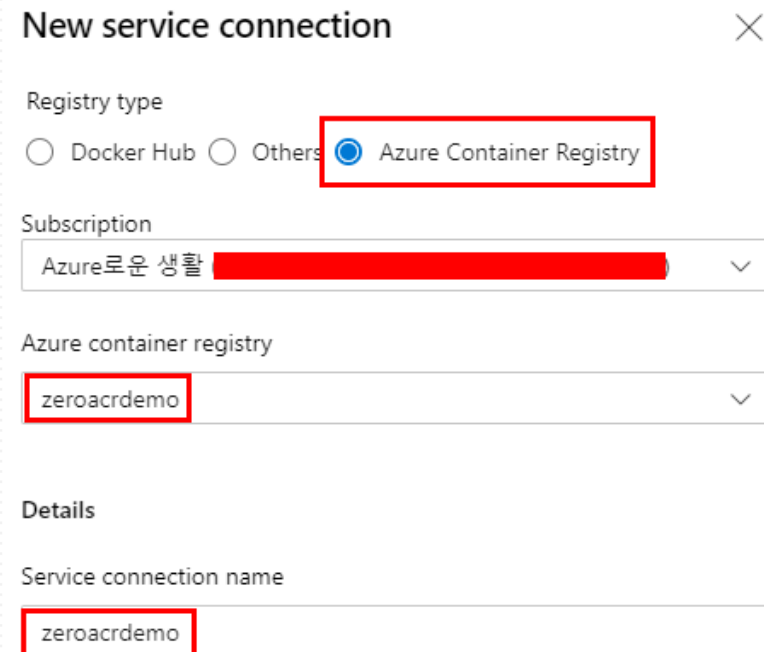
Container repository ⓘ

Commands ^

Command \* ⓘ  
buildAndPush

buildAndPush 태스크를 선택  
하고 +New를 클릭하여  
service connection 생성

Azure Container Registry 선택 후  
로그인 과정을 거친 후 등록된 ACR  
선택 및 Service connection name  
입력하여 Save



New service connection

Registry type  
☐ Docker Hub ☐ Others ☒ Azure Container Registry

Subscription  
Azure로운 생활

Azure container registry  
zeroacrdemo

Details  
Service connection name  
zeroacrdemo

## Azure DevOps 빌드 파이프라인 구성하기 (계속)

The screenshot displays the Azure DevOps build pipeline configuration interface. On the left, a sidebar lists the build steps: **Restore** (.NET Core), **Build** (.NET Core), **Publish** (.NET Core), **buildAndPush** (Docker) (selected), **Bash Script** (Some settings need attention), and **Publish Artifact** (Publish build artifacts). The main configuration area for the **buildAndPush** step is shown on the right. It includes a **Container Repository** section with a dropdown menu set to **zeroacrdemo** and a **+ New** button. Below this, the **Container repository** field is set to **gav-zerodemo**, which is highlighted with a red box. The **Commands** section shows a dropdown menu set to **buildAndPush**. The **Dockerfile** section shows a dropdown menu set to **\*\*/Dockerfile**. A blue arrow points from the **buildAndPush** step in the sidebar to the **Container repository** field.

Container repository 필드에 해당 이름 입력

## Azure DevOps 빌드 파이프라인 구성하기 (계속)

Agent job 1  
Run on agent

- Restore  
.NET Core
- Build  
.NET Core
- Publish  
.NET Core
- buildAndPush  
Docker
- Bash Script**  
Bash
- Publish Artifact  
Publish build artifacts

Display name \*

Bash Script

Type ⓘ

☐ File Path ☒ Inline

Script \* ⓘ

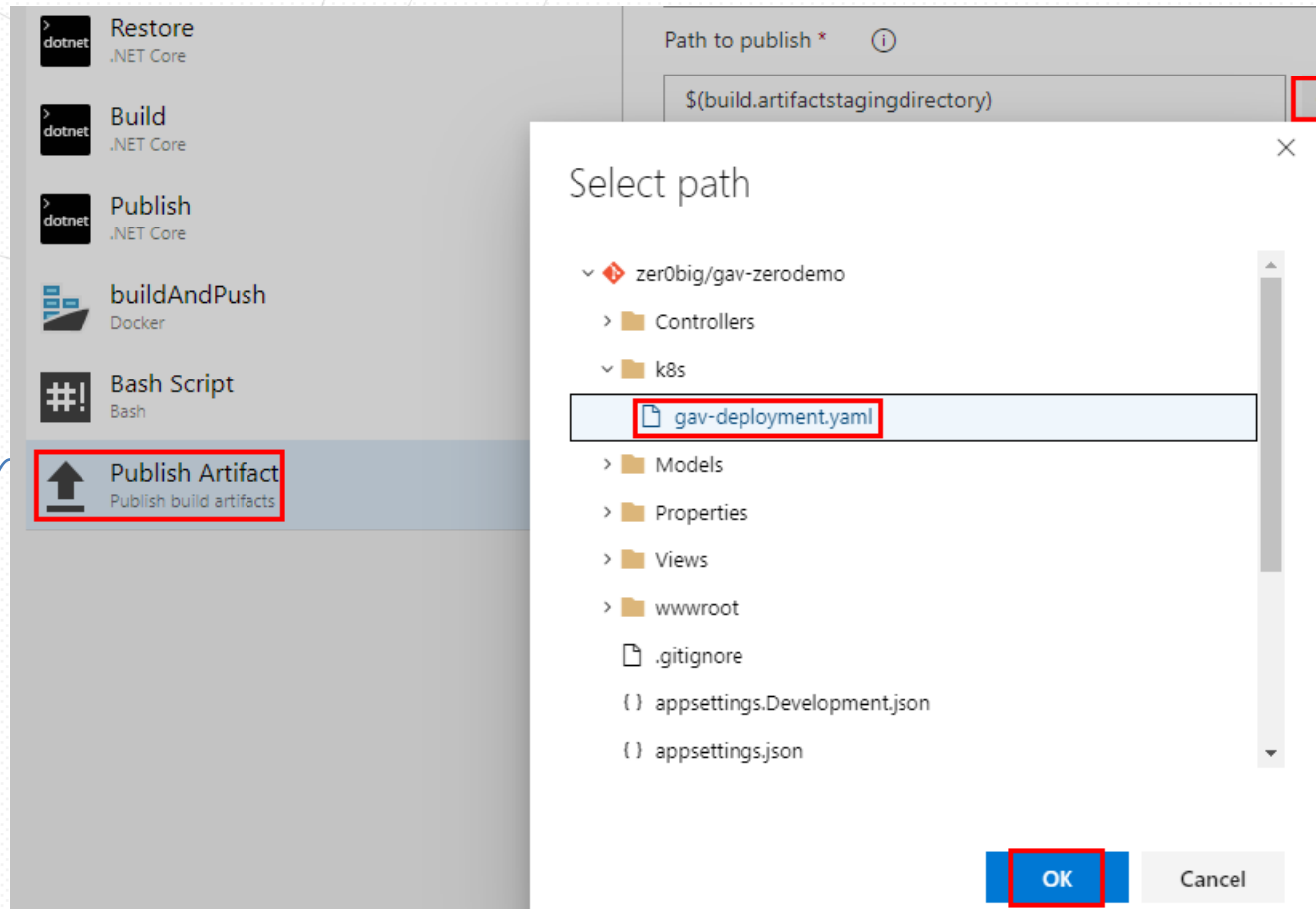
# Write your commands here

```
sed -i "s/latest/$(Build.BuildId)/g" k8s/gav-deployment.yaml
```

Advanced ▾

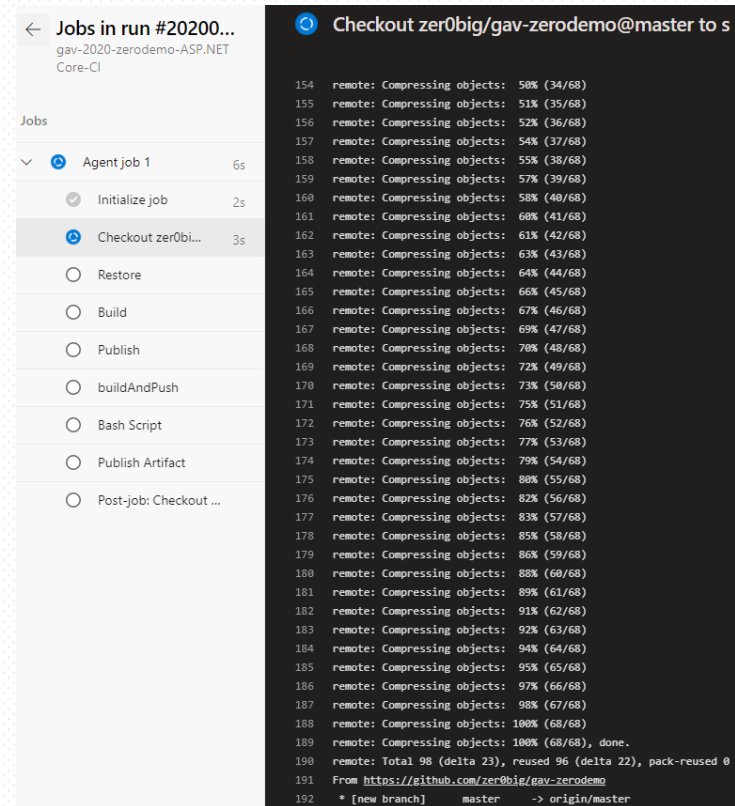
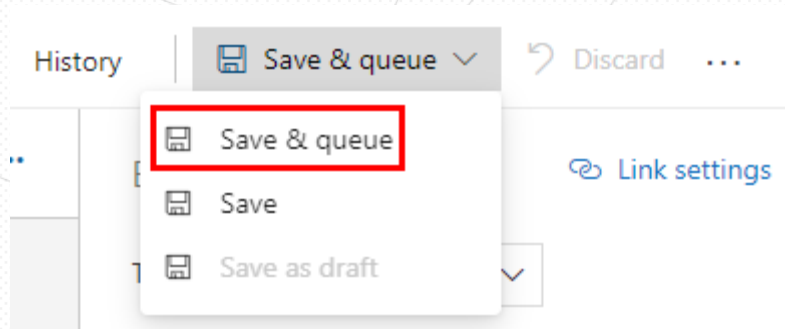
Bash Script 태스크에서 Inline 유형을 선택 후 "sed -i  
"s/latest/\$(Build.BuildId)/g" k8s/gav-deployment.yaml" 명령 입력

## Azure DevOps 빌드 파이프라인 구성하기 (계속)



**Publish 태스크에서 Path to publish의 패스를 k8s/gav-deployment.yaml로 선택하고 OK 클릭**

# Azure DevOps 빌드 파이프라인 구성하기 (계속)



구성한 빌드 파이프라인을 Save & queue를 선택하고 Save and run을 다시 선택하여 빌드 진행 및 결과 검증



# Azure DevOps 빌드 파이프라인 구성하기 (계속)

← Jobs in run #20200...

gav-2020-zerodemo-ASP.NET  
Core-CI

Jobs

|   |                    |        |
|---|--------------------|--------|
| ✓ | Agent job 1        | 1m 24s |
| ✓ | Initialize job     | 3s     |
| ✓ | Checkout zerObi... | 4s     |
| ✓ | Restore            | 5s     |
| ✓ | Build              | 7s     |
| ✓ | Publish            | 2s     |
| ✓ | buildAndPush       | 57s    |
| ✓ | Bash Script        | <1s    |
| ✓ | Publish Artifact   | 1s     |
| ✓ | Post-job: Check... | <1s    |
| ✓ | Finalize Job       | <1s    |

✓ Agent job 1

View raw log

```
1 Pool: Azure Pipelines
2 Image: ubuntu-16.04
3 Queued: Just now [manage_parallel_jobs]
4 Agent: Hosted Agent
5 Started: Just now
6 Duration: 1m 24s
7
8 The agent request is already running or has already completed.
9 ▶ Job preparation parameters
10 ▶ fx 3 queue time variables used
11 📁 1 artifact produced
12 Job live console data:
13 Starting: Agent job 1
14 Finishing: Agent job 1
```

← Artifacts

Published Consumed

Name

|   |                       |
|---|-----------------------|
| ✓ | 📁 drop                |
|   | 📄 gav-deployment.yaml |

빌드 완료 후 Artifacts – drop 위치에 gav-deployment.yaml 파일 생성 확인

# Rancher 및 AKS 설치하기

To Do

+ New item

i 612 Rancher 모니터링 환경 구축하기

Z zerobig.devops

State ● To Do

0/2

i 611 Azure DevOps 릴리즈 파이프라인 구성하기

Z zerobig.devops

State ● To Do

Doing

i 610 Rancher 및 AKS 설치하기

Z zerobig.devops

State ● Doing

1/5

Done

i 609 Azure DevOps 빌드 파이프라인 구성하기

Z zerobig.devops

State ● Done

i 605 서비스 이미지 준비 및 Azure Container Registry 생성하기

Z zerobig.devops

State ● Done

2/2

+ Add Task

☒ ☒ Local 환경 Docker 빌드 및 이미지 검증하기

☒ ☒ ACR 생성 및 이미지 태그 푸시하기

24

CSPTek

## Rancher 및 AKS 설치하기 (계속)

# 서비스 주체 생성

az ad sp create-for-rbac --name demo4gav

```
root@zerovmu1804:~/gav-zero-demo# az ad sp create-for-rbac --name demo4gav
Changing "demo4gav" to a valid URI of "http://demo4gav", which is the required format used for service principal names
Found an existing application instance of "db389d64-6370-4861-af24-8a8f772fb470". We will patch it
Creating a role assignment under the scope of "/subscriptions/a791ad38-[REDACTED]-460ee318d917"
Role assignment already exists.
```

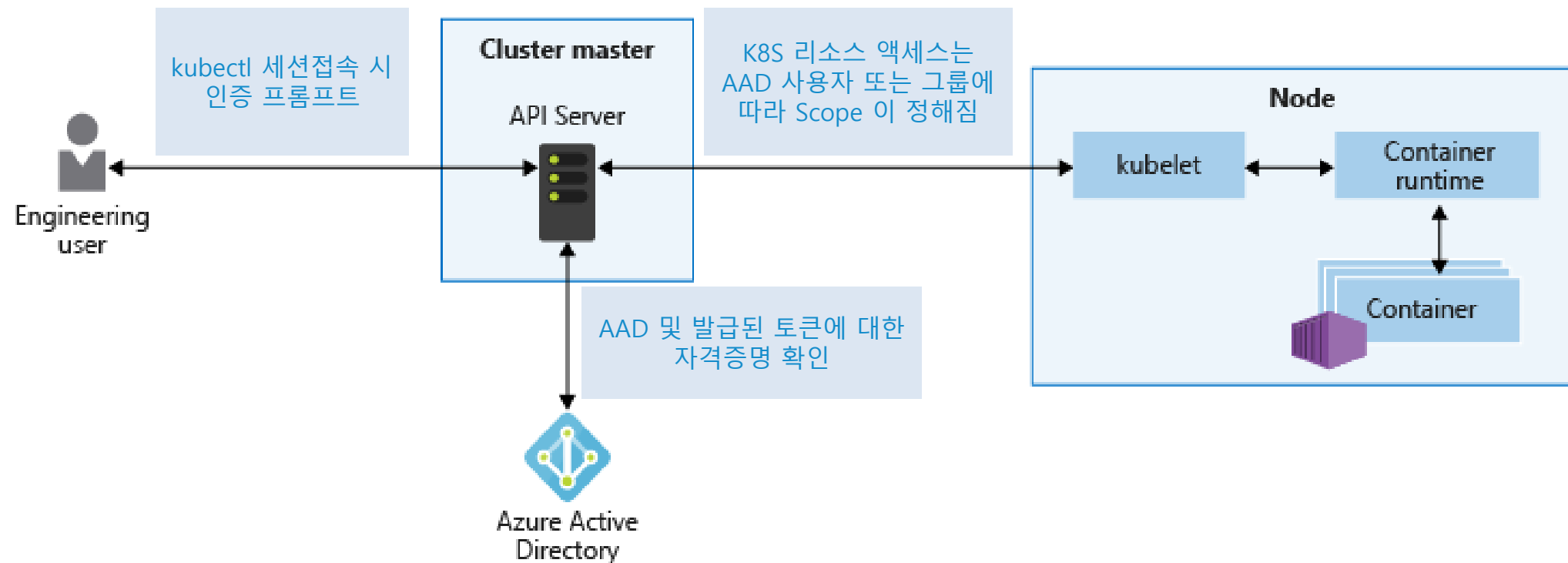
```
{
  "appId": "db389d64-6370-4861-af24-8a8f772fb470",
  "displayName": "demo4gav",
  "name": "http://demo4gav",
  "password": "737d6b9d-[REDACTED]-71ccbe1d4f3e",
  "tenant": "bb43721a-680c-4c9f-b234-72c8fe6c8e3c"
}
```

명령 수행 결과는 AKS 구성 시 필요하므로 따로 복사하여 보관

## Rancher 및 AKS 설치하기 (계속)

### AKS - Azure Active Directory 통합

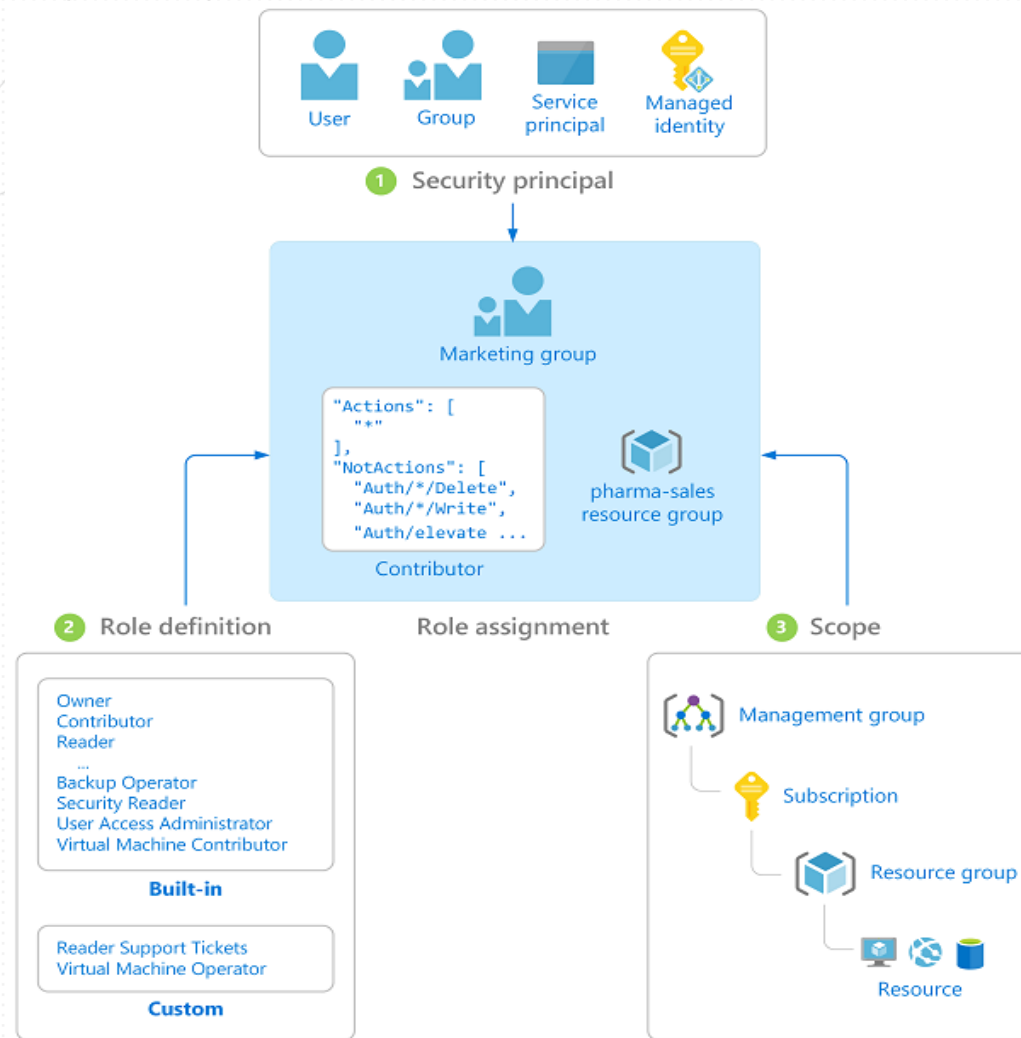
- AKS는 Azure AD(Active Directory) 통합으로 클러스터의 보안을 강화할 수 있음
- Azure AD를 사용하면 온-프레미스 ID를 AKS 클러스터에 통합하여 계정 관리 및 보안을 위한 단일 소스를 제공



\* 상세 내용 : <https://docs.microsoft.com/ko-kr/azure/aks/concepts-identity> 참조

# Rancher 및 AKS 설치하기 (계속)

## RBAC 역할할당 예시



\* 상세 내용 : <https://docs.microsoft.com/ko-kr/azure/role-based-access-control/overview> 참조

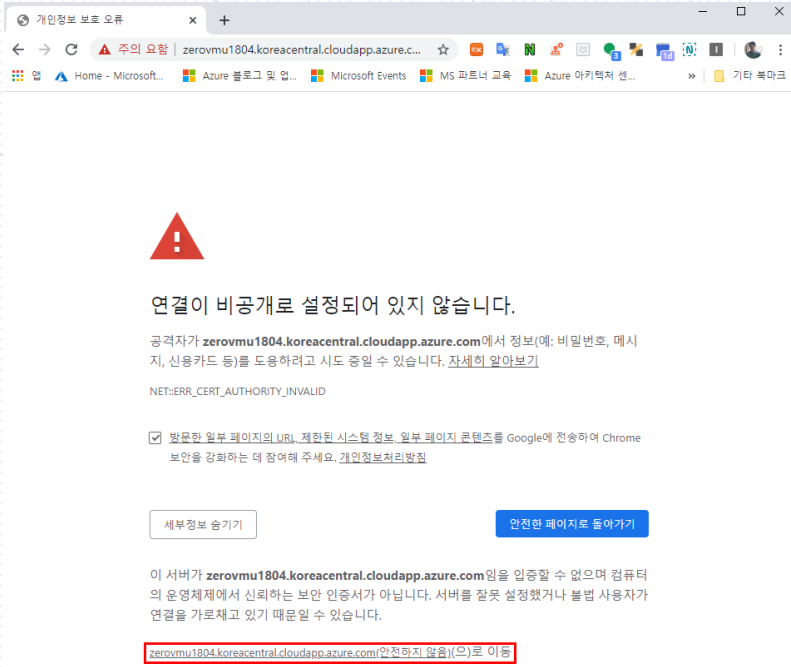
## Rancher 및 AKS 설치하기 (계속)

# Rancher 실행

```
docker run -d -v /home/zerobig/rancher-data:/var/lib/rancher -p 8080:80 -p 443:443  
rancher/rancher:v2.3.5
```

```
root@zerovmu1804:~/gav-zerodemo# docker run -d -v /home/zerobig/rancher-data:/var/lib/rancher -p 8080:80 -p 443:443 ra  
ncher/rancher:v2.3.5  
Unable to find image 'rancher/rancher:v2.3.5' locally  
v2.3.5: Pulling from rancher/rancher  
5c939e3a4d10: Pull complete  
c63719cdbe7a: Pull complete  
19a861ea6baf: Pull complete  
651c9d2d6c4f: Pull complete  
6d1c86a401db: Pull complete  
c7d485afd256: Pull complete  
285d247a5c2b: Pull complete  
e7acc0299fc2: Pull complete  
7346f9f2da73: Pull complete  
91f1fe4c3d21: Pull complete  
8b36bf060d06: Pull complete  
1e46e34177f8: Pull complete  
08aff7247104: Pull complete  
bf9b23e67888: Pull complete  
Digest: sha256:cdffc4e0d1aff6adb606b0f9f3033bf7667bcda69aa43293355b6bc2701b5c6a  
Status: Downloaded newer image for rancher/rancher:v2.3.5  
f0eefd992ff194795da1a9200b6d480afb2a0f06f5f14536037e80a8db609ec2
```

# Rancher 및 AKS 설치하기 (계속)



Welcome to Rancher

The first order of business is to set a strong password for the default `admin` user.

☒ Allow collection of anonymous statistics [Learn More](#)

☒ Set a specific password to use:

New Password

.....

Confirm Password

.....

☐ Use a new randomly generated password:

[Continue](#)

Rancher Server URL

What URL should be used for this installation? All the nodes in your clusters will need to be able to reach this.

URL

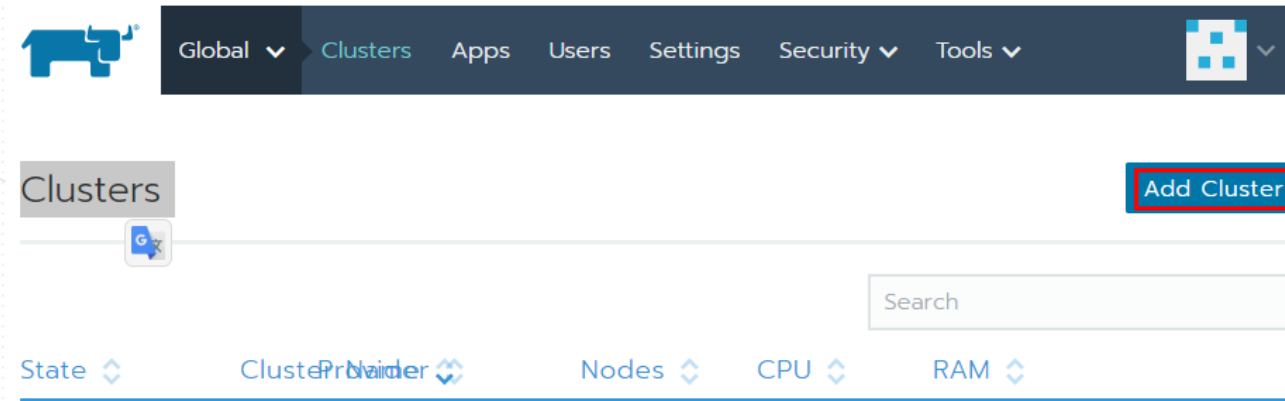
[https://](#)

[Save URL](#)

브라우저 창에 작업수행 VM의 주소 및 도메인 네임 입력 후 Rancher에 접속하고 초기 패스워드 및 Server URL 정보 Save 수행

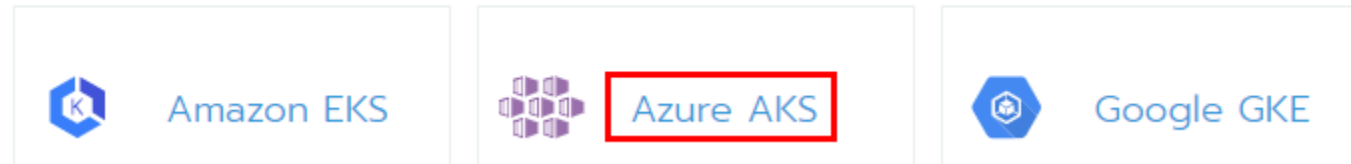


## Rancher 및 AKS 설치하기 (계속)



우측 상단 Add Cluster 선택

With a hosted Kubernetes provider



화면 하단 Azure AKS 선택

## Rancher 및 AKS 설치하기 (계속)

Cluster Name 및 하단  
Subscription ID, Tenant ID,  
Client ID(=AppId), Client  
Secret(=password),  
Location 정보 입력 후 "Next:  
Authenticate & configure  
nodes" 선택

Cluster Name \*

[Add a Description](#)

zero-aks-rancher-demo



Note: Currently Amazon EKS will not create an ingress controller when launching a new cluster. If you need this functionality you will have to create an ingress controller manually after cluster creation.



### Member Roles

Control who has access to the cluster and what permission they have to change it.



### Labels & Annotations

Configure labels and annotations for the cluster.

None

### Account Access

Configure the Azure Active Directory (AD) Service Principal that will be used to talk to Azure

Subscription ID \*

a791ad38-██████████-460ee318d917

Tenant ID \*

bb43721a-680c-4c9f-b234-72c8fe6c8e3c

Client ID \*

db389d64-6370-4861-af24-8a8f772fb470

Client Secret \*

.....

Location \*

Korea Central



# Rancher 및 AKS 설치하기 (계속)

Kubernetes Options

Customize the kubernetes cluster options

Kubernetes Version

v1.15.10

DNS Prefix

e.g. example

HTTP Application Routing

☐ Enabled ☒ Disabled

Monitoring

☒ Enabled ☐ Disabled

Tags

+ Add Tag

Node Options

Customize the nodes that will be created

Admin Username

azureuser

Cluster Resource Group \*

Odemo4gav2020

Node Count

2

OS Disk Size

100

GB

VM Size

Standard\_D2\_v2

Networking

☒ Default ☐ Advanced

SSH Public Key \*

Read from a file

ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQCA4sAldn0/AQAiuCVNVdhtvLVpWgMzqXNbtv-CHFqPQ9Q9P2htUNBKozFLQarE0CWtrMamIDsnYIDJpSHqPSHYzVoDU/yjufbIRBAUWr0grO  
689j2HVdTI0lqnyl8qhWkxfZtHHA7HHf7v7IVKg4sqWGScjkBHmx5MydiqDNKQl/lqbcNH5F-RvRk5IZOZzsq/P9U-CXd2BQw3Gc-0aN/Mrkb+IOUKYbplEbdayECarDauTaeAwlPM+Oh  
SfptpU9A/NZxfa6bs-LdxncndVvd0VEolKQ8xRdbiUrMIJoJ9v/gsT6gwVy-mMDzbapzibY9g-6wDVAuGbrB8ITCq8v

Kubernetes version (v1.15.10), Cluster Resource Group(ACR 생성한 그룹과 동일), Node Count 및 SSH Key 정보 입력 후 “Create” 선택

## Rancher 및 AKS 설치하기 (계속)

Clusters

State Cluster Name Add Cluster

Delete

Provisioning zero-aks-rancher-demo Search

| State        | Cluster Name          | Provider  | Nodes | CPU | RAM |
|--------------|-----------------------|-----------|-------|-----|-----|
| Provisioning | zero-aks-rancher-demo | Azure AKS | 0     | n/a | n/a |

| State  | Cluster Name          | Provider  | Nodes | CPU | RAM |
|--------|-----------------------|-----------|-------|-----|-----|
| Active | zero-aks-rancher-demo | Azure AKS | 2     | n/a | n/a |

State Cluster Name

Active zero-aks-rancher-demo

최초 Provisioning 상태에서 약 7~8분 경과 후  
이상이 없다면 Active 상태로 전환

## Rancher 및 AKS 설치하기 (계속)

State: Active Cluster Name: zero-aks-rancher-demo

Tools

Dashboard: zero-aks-rancher-demo

Provider: Azure AKS | Kubernetes Version: v1.15.10 | Created: 4:50 PM

Enable Monitoring to see live metrics

22% CPU 0.8 of 3.8 Reserved

11% Memory 0.9 of 8.9 GiB Reserved

6% Pods 14 of 220 Used

Etcd Controller Manager Scheduler Nodes

Events  
Events of current Cluster

생성된 클러스터를 선택하여 초기 대시보드 화면으로 이동 후 Tools - Monitoring 선택, 클러스터 모니터링 구성화면 하단에서 "Enable" 클릭

# Azure DevOps 릴리즈 파이프라인 구성하기

To Do

+ New item

612 Rancher 모니터링 환경 구축하기

zerobig.devops

State To Do

0/2

Doing 1/5

611 Azure DevOps 릴리즈 파이프라인 구성하기

zerobig.devops

State Doing

Done

610 Rancher 및 AKS 설치하기

zerobig.devops

State Done

609 Azure DevOps 빌드 파이프라인 구성하기

zerobig.devops

State Done

605 서비스 이미지 준비 및 Azure Container Registry 생성하기

zerobig.devops

State Done

2/2

## Azure DevOps 릴리즈 파이프라인 구성하기 (계속)




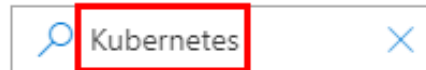
### No release pipelines found

Automate your release process in a few easy steps with a new pipeline

New pipeline

### Select a template

Or start with an  Empty job



### Featured



#### Deploy to a Kubernetes cluster

Deploy, configure, update your containerized applications to a Kubernetes cluster.

Apply

Pipelines – Releases 선택하여 New pipeline 선택 후 템플릿 선택 화면에서 “Kubernetes” 입력, 검색 결과 중 “Deploy to a Kubernetes cluster” 선택하여 Apply



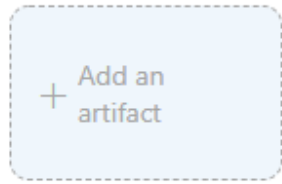
## Azure DevOps 릴리즈 파이프라인 구성하기 (계속)

Stage  
Demo

Properties ^  
Name and owners of the stage

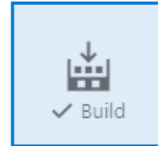
Stage name  
Demo

Stage owner  
zerobig.devops



### Add an artifact

Source type



5 more artifact types v

Project \* ⓘ

gav-2020-zerodemo

Source (build pipeline) \* ⓘ

gav-2020-zerodemo-ASP.NET Core-CI

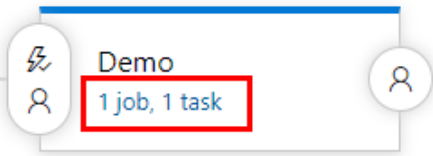
Add

Stage 이름을 임의로 입력 후 해당 창을 닫은 뒤 +Add an artifact를 선택하여 Source(build pipeline)를 빌드 파이프라인의 결과인 "gav-2020-zerodemo-ASP.NET Core-CI"로 지정하여 선택 후 Add

## Azure DevOps 릴리즈 파이프라인 구성하기 (계속)

1 job, 1 task를 선택하여 kubectl 태스크 구성을 시작

Task version을 1.\*로 선택하고 Service connection type을 Azure Resource Manager로 지정한 뒤 Azure subscription을 선택하여 Authorize 수행



Kubectl ⓘ

View YAML Remove

Task version

1.\*

Display name \*

kubectl

Kubernetes Cluster ^

Service connection type \*



Azure Resource Manager

Azure subscription \*



[Manage](#)

Azure로운 생활 (a791ad38--460ee318d917)

Authorize



ⓘ Click Authorize to configure an Azure service connection. A new Azure service principal will be created and added to the Contributor role, having access to all resources in the selected subscription. To restrict the scope of the service principal to a specific resource group, see [connect to Microsoft Azure](#)

## Azure DevOps 릴리즈 파이프라인 구성하기 (계속)

Resource group \* ⓘ  
0demo4gav2020

Kubernetes cluster \* ⓘ  
c-bbnkg

☐ Use cluster admin credentials ⓘ

Namespace ⓘ  
default

Commands ^

Command ⓘ  
apply

☒ Use configuration ⓘ

자동 감지된 Resource group,  
Kubernetes cluster를 선택하고  
Namespace(default),  
command(apply)를 입력 후 “Use  
configuration”을 체크하여 drop 이하  
yaml 파일 위치를 지정,  
추가로 Advanced – version spec에서  
“Check for latest version” 체크

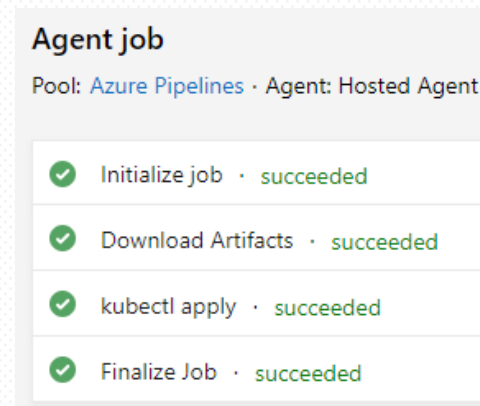
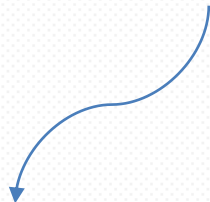
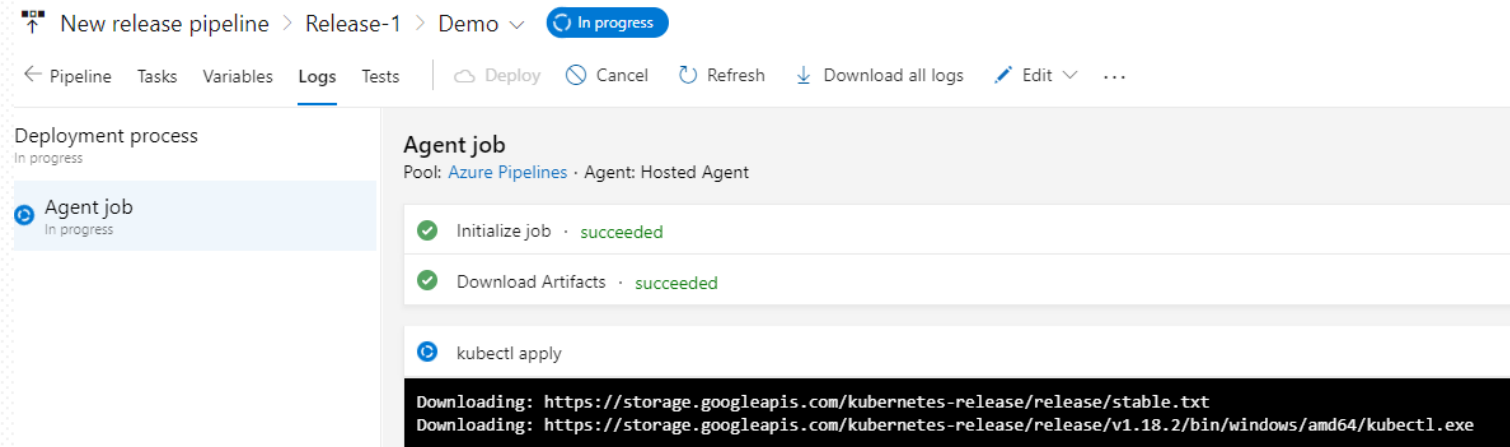
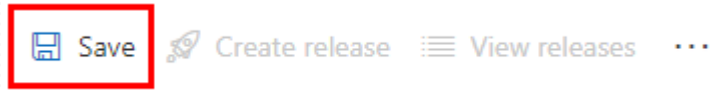
Select a file or folder

- Linked artifacts
  - \_gav-2020-zerodemo-ASP.NET Core-CI (Build)
    - drop
      - gav-deployment.yaml

Version spec ⓘ  
1.7.0

☒ Check for latest version ⓘ

# Azure DevOps 릴리즈 파이프라인 구성하기 (계속)



구성한 릴리즈 파이프라인을 Save,  
Create release 후 Create하여 실행하  
여 결과 확인

## Azure DevOps 릴리즈 파이프라인 구성하기 (계속)

# AKS 자격증명 획득 및 배포 결과 모니터링

az aks get-credentials -g 0demo4gav2020 -n c-bbnkg

```
root@zerovmu1804:~/gav-zero demo# az aks get-credentials -g 0demo4gav2020 -n c-bbnkg
Merged "c-bbnkg" as current context in /root/.kube/config
root@zerovmu1804:~/gav-zero demo# kubectl get nodes
```

| NAME                   | STATUS | ROLES | AGE | VERSION  |
|------------------------|--------|-------|-----|----------|
| aks-rancher-40569460-0 | Ready  | agent | 47m | v1.15.10 |
| aks-rancher-40569460-1 | Ready  | agent | 47m | v1.15.10 |

kubectl get pod

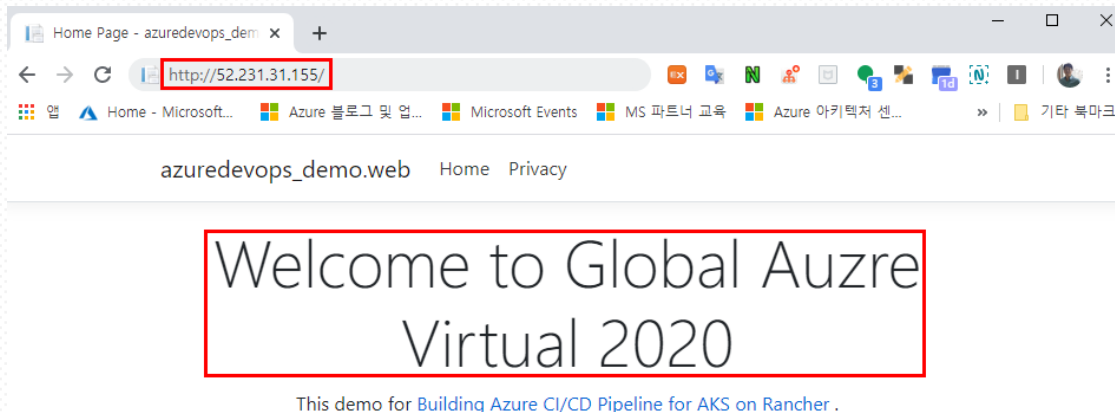
kubectl get svc

```
root@zerovmu1804:~/gav-zero demo# kubectl get pod
```

| NAME                         | READY | STATUS  | RESTARTS | AGE |
|------------------------------|-------|---------|----------|-----|
| zeroaksdemo-797b5d8d47-2nwcs | 1/1   | Running | 0        | 36s |
| zeroaksdemo-797b5d8d47-b87jc | 1/1   | Running | 0        | 17s |

```
root@zerovmu1804:~/gav-zero demo# kubectl get svc
```

| NAME         | TYPE         | CLUSTER-IP   | EXTERNAL-IP   | PORT(S)      | AGE |
|--------------|--------------|--------------|---------------|--------------|-----|
| kubernetes   | ClusterIP    | 10.0.0.1     | <none>        | 443/TCP      | 67m |
| zero-aks-app | LoadBalancer | 10.0.171.155 | 52.231.31.155 | 80:32504/TCP | 20m |



LoadBalancer EXTERNAL-IP  
웹 접속 시 로컬 검증 결과와  
동일 결과 확인

# Azure DevOps CI/CD 구성 및 결과 검증

gav-2020-zerodemo-ASP.NET Core-CI

Tasks Variables **Triggers** Options Retention History | Save & queue Discard Summary Queue ...

Continuous integration

zer0big/gav-zerodemo  
Enabled

Pull request validation

zer0big/gav-zerodemo

☒ Enable continuous integration

☐ Batch changes while a build is in progress

Artifacts | + Add

\_gav-2020-zerodemo-

Continuous deployment trigger

Build: \_gav-2020-zerodemo-ASP.NET Core-CI

☒ Enabled

Creates a release every time a new build is available.

CI 빌드 파이프라인 Edit – Triggers에서 “Enable continuous integration” 체크  
CD 릴리즈 파이프라인 Edit 에서 “Continuous deployment trigger” 활성화

## Azure DevOps CI/CD 구성 및 결과 검증 (계속)

# 소스 변경 및 CI/CD 검증

```
root@zerovmu1804:~/gav-zero-demo# vi Views/Home/Index.cshtml
```

```
<div class="text-center">
  <h1 class="display-4">Korea will overcome COVID-19 crisis soon~~~!!!</h1>
  <p>This demo for <a href="https://docs.microsoft.com/aspnet/core"> Building
r </a>.</p>
</div>
```

```
root@zerovmu1804:~/gav-zero-demo# git add .
```

```
root@zerovmu1804:~/gav-zero-demo# git commit -m "Updated Index.cshtml"
```

```
root@zerovmu1804:~/gav-zero-demo# git push
```

소스 내용을 수정하여 CI와 CD에 대한 트리거가 유발되는지 확인하고 소스 변경이 정상적으로 반영되는지 확인



# Azure DevOps 릴리즈 파이프라인 구성하기 (계속)

## Recently run pipelines

Pipeline Last run

gav-2020-zerodemo-ASP.NET Core-CI #20200420.4 • Updated Index.cshhtml  
Individual CI for master

Jobs in run #20200420.4  
gav-2020-zerodemo-ASP.NET Core-CI

| Jobs                               | Duration |
|------------------------------------|----------|
| Agent job 1                        | 1m 28s   |
| Initialize job                     | 4s       |
| Checkout zer0big/gav-zerodemo...   | 3s       |
| Restore                            | 5s       |
| Build                              | 6s       |
| Publish                            | 2s       |
| buildAndPush                       | 1m 3s    |
| Bash Script                        | <1s      |
| Publish Artifact                   | <1s      |
| Post-job: Checkout zer0big/gav-... | <1s      |
| Finalize Job                       | <1s      |

Agent job 1

```
1 Pool: Azure Pipelines
2 Image: ubuntu-16.04
3 Agent: Hosted Agent
4 Started: Just now
5 Duration: 1m 28s
6
7 Job preparation parameters
8 1 artifact produced
9 Job live console data:
10 Finishing: Agent job 1
```

Release-2  
20200420.4 master 2020. 4. 20. 오후 6:21:52 Demo

## Agent job

Pool: Azure Pipelines · Agent: Hosted Agent

- Initialize job · succeeded
- Download Artifacts · succeeded
- kubectl apply · succeeded
- Finalize Job · succeeded

Home Page - azuredevops\_demo x +

주의 요함 52.231.31.155

azuredevops\_demo.web Home Privacy

Korea will overcome COVID-19 crisis soon ~ ~ ~ !!!

This demo for Building Azure CI/CD Pipeline for AKS on Rancher .

CI 빌드가 유발되어 처리되고 연이어 CD가 유발되어 최종 웹 페이지에 정상 배포 이루어 짐



# Rancher 모니터링 환경 구축하기

To Do

+ New item

<

Doing

1/5

>

612 Rancher 모니터링 환경 구축하기

zerobig.devops

State ● Doing

0/2

+ Add Task

☐ ☒ Grafana 구성하기

☐ ☒ Slack 연동하기

611 Azure DevOps 릴리즈 파이프라인 구성하기

zerobig.devops

State ● Done

610 Rancher 및 AKS 설치하기

zerobig.devops

State ● Done

609 Azure DevOps 빌드 파이프라인 구성하기

zerobig.devops

State ● Done

605 서비스 이미지 준비 및 Azure Container Registry 생성하기

zerobig.devops

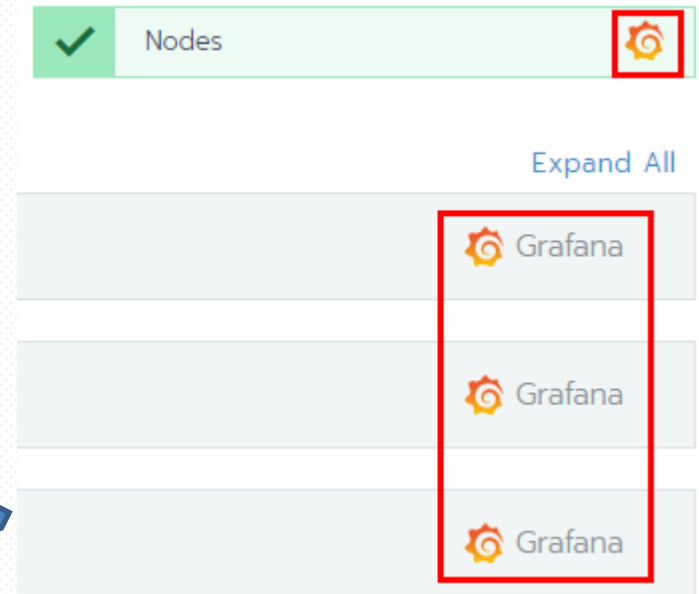
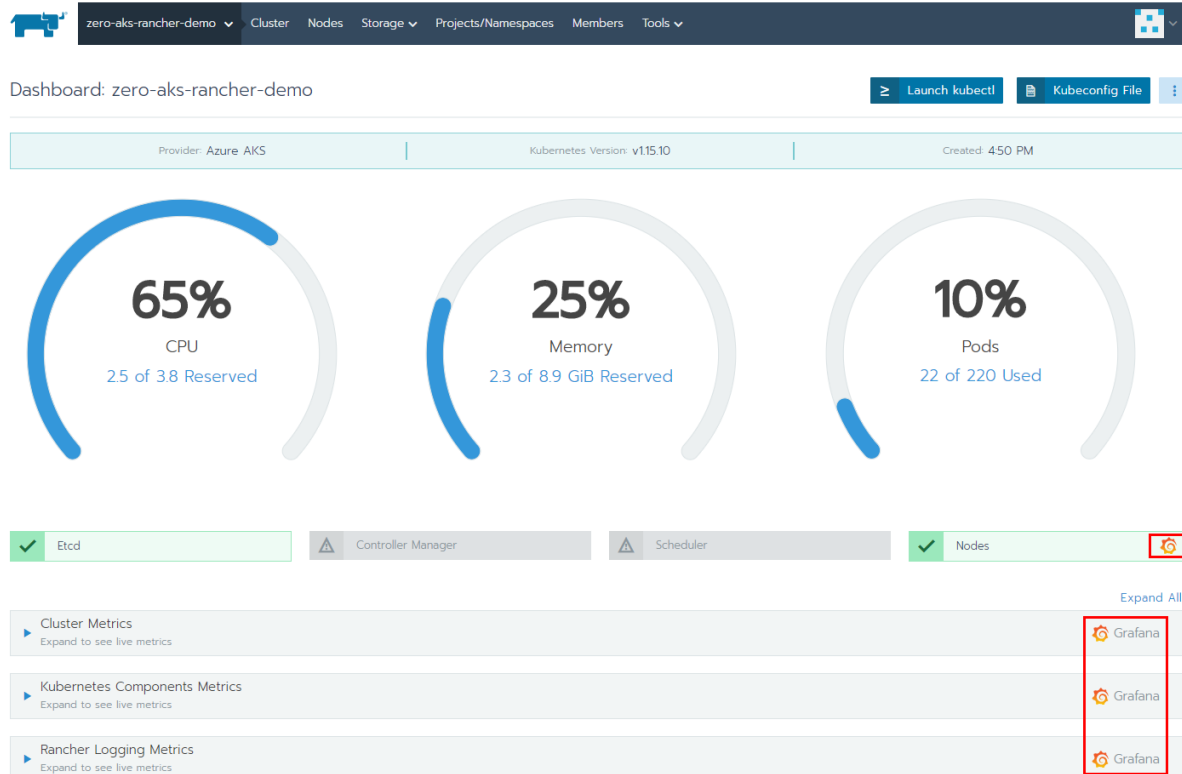
State ● Done

2/2

Click Authorize to configure an Azure service connection. A new Azure service principal will be created and added to the Contributor role,  
① having access to all resources in the selected subscription. To restrict the scope of the service principal to a specific resource group, see  
[connect to Microsoft Azure](#)

# Rancher 모니터링 환경 구축하기 (계속)

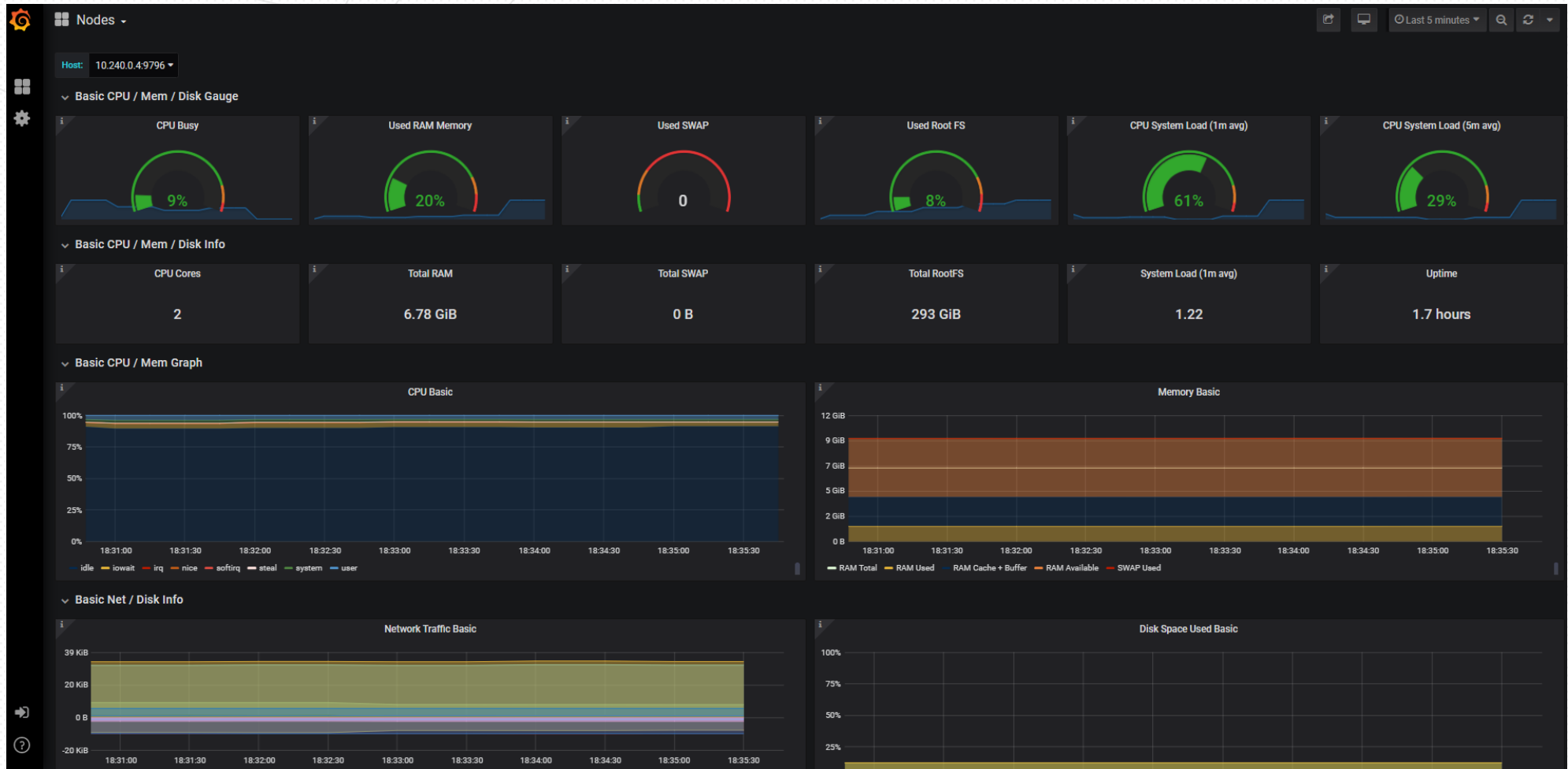
## Grafana 구성하기



이전 모니터링 활성화 결과로 Grafana 아이콘이 활성화 되었으며 해당 아이콘을 선택

# Rancher 모니터링 환경 구축하기 (계속)

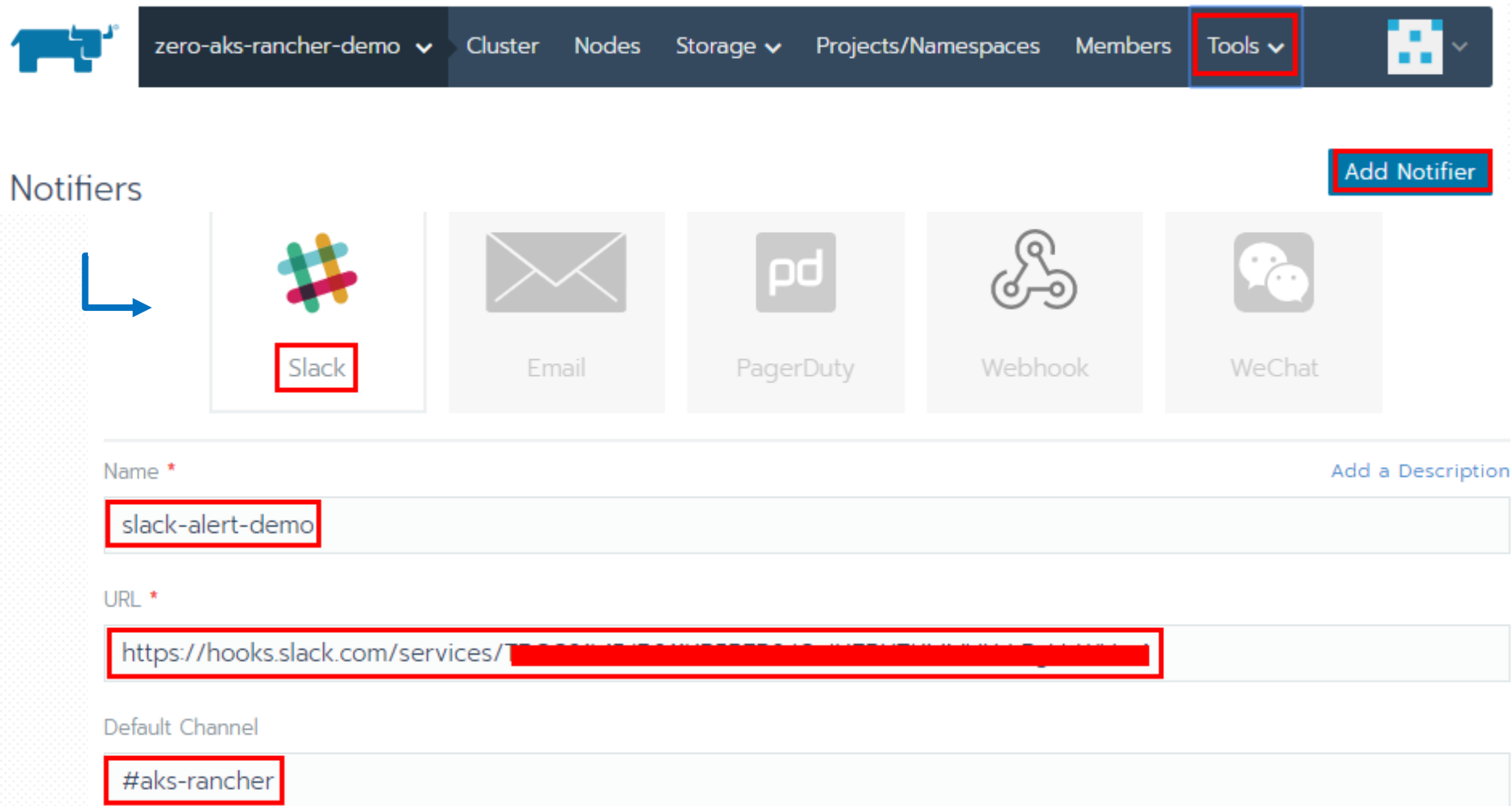
## Grafana 구성하기



Grafana에 의한 모니터링이 정상적으로 확인됨

## Rancher 모니터링 환경 구축하기 (계속)

### Slack 연동하기



The screenshot shows the Rancher UI interface. At the top, the navigation bar includes the Rancher logo, the cluster name 'zero-aks-rancher-demo', and several menu items: 'Cluster', 'Nodes', 'Storage', 'Projects/Namespaces', 'Members', 'Tools', and a user profile icon. The 'Tools' menu is highlighted with a red box. Below the navigation bar, the 'Notifiers' section is visible. A blue arrow points to the 'Slack' notifier icon, which is also highlighted with a red box. To the right of the notifier icons is a red 'Add Notifier' button. Below the notifier selection, there are three input fields: 'Name', 'URL', and 'Default Channel'. The 'Name' field contains 'slack-alert-demo', the 'URL' field contains a Slack hook URL, and the 'Default Channel' field contains '#aks-rancher'. Each of these three input fields is highlighted with a red box.

zero-aks-rancher-demo Cluster Nodes Storage Projects/Namespaces Members Tools

Notifiers

Add Notifier

Slack Email PagerDuty Webhook WeChat

Name \* Add a Description

slack-alert-demo

URL \*

https://hooks.slack.com/services/

Default Channel

#aks-rancher

우측 상단 Tools – Notifier 선택 후 “Add Notifier”를 클릭  
Slack 선택 후 이름, hook URL 및 채널이름을 입력

## Rancher 모니터링 환경 구축하기 (계속)

### Slack 연동하기

TEST

#### #aks-rancher

You created this channel on April 13th. This is the very

[Add an app](#) [Add people](#)



**zerobig** 4:24 PM

added an integration to this channel: incoming-webhook



**incoming-webhook** APP 6:42 PM

Slack setting validated

하단의 "Test"를 선택하여 Slack 연동에 대한 검증 수행

# Rancher 모니터링 환경 구축하기 (계속)

## Slack 연동하기



zero-aks-rancher-demo ▾

Cluster

Nodes

Storage ▾

Projects/Namespaces

Members

Tools ▾



### Cluster Alerts

Add Alert Group

Name

alert-group-demo

Alert Rule

Name demo1

When ☐ System Service

scheduler ▾

And

☐ Normal

Deployment ▾

☐ Warning

☐ Node

aks-rancher-40569460-0 ▾

☐ Node Selector

+ Add Selector

☒ Expression

kube\_deployment\_created ▾

Event

Is Greater Than ▾

1

For

10 seconds ▾

Add Alert Group을 선택  
하여 그룹 이름을 입력하  
고 Alert Rule을 추가

# Rancher 모니터링 환경 구축하기 (계속)

## Slack 연동하기

Name Add

alert-group-demo

Alert Rule

Name demo1

When An

☐ System Service scheduler

☐ Normal Deployment

☐ Warning

☐ Node aks-rancher-40569460-0

☐ Node Selector + Add Selector

☒ Expression kube\_deployment\_created

Is Greater Than 1

For 10 seconds

Alert Rule 이름(demo1),  
System service(Scheduler),  
Event (Deployment), Node,  
Expression(kube\_deployment\_created), Greater Than  
(1), For(10 seconds)에 각각  
의 값을 입력

→ 노드1 스케줄러에 10초 동안 1개 이상의 Deployment가 create되는 이벤트가 감지되면 Alert를 생성하는 규칙

## Rancher 모니터링 환경 구축하기 (계속)

### Slack 연동하기

0 17:08 04-20 17:30 04-20 18:00 04-20 18:30 04-20 19:00 04-20 19:18 04-20

Send a Critical **Warning** Info

Show advanced options

Send a (Warning) 선택 후  
"Show advanced options"  
선택하여 Group Interval  
Time과 Repeat Interval  
Time 입력

→ Group Interval Time : 이미 발송된 alert를 포함하는 그룹에 추가된 alert를 전송 전 대기시간, Repeat Interval Time : 이미 전송된 지정된 alert를 재전송하기 전 대기시간

Group Inherited

☐ Enabled - Inheriting the advanced options from group

☒ Disabled

Group Wait Time

H 0 M 0 S 30

How long to wait to buffer alerts of the same group before sending initially.

Group Interval Time

H 0 M 0 S **3**

How long to wait before sending an alert that has been added to a group which contains already fired alerts.

Repeat Interval Time

H 0 M **1** S **30**

How long to wait before re-sending a given alert that has already been sent.



# Rancher 모니터링 환경 구축하기 (계속)

## Slack 연동하기

Alert To **slack-alert-demo (Slack)** Channel **#aks-rancher**

The value set in the Notifier is used by default. You can override this with another value.

**+ Add Recipient**

**Show advanced options**

**Create** Cancel

Group Wait Time  
H 0 M 0 S 30  
How long to wait to buffer alerts of the same group before sending initially.

Group Interval Time  
H 0 M 0 S **5**  
How long to wait before sending an alert that has been added to a group which contains already fired alerts.

Repeat Interval Time  
H 0 M **1** S **30**  
How long to wait before re-sending a given alert that has already been sent.

**Create** Cancel

|                          |        |       |        |                |                          |  |
|--------------------------|--------|-------|--------|----------------|--------------------------|--|
| alert-group-demo         |        |       |        |                | Add Alert Rule           |  |
| <input type="checkbox"/> | Active | demo1 | Metric | Greater Than 1 | slack-alert-demo (Slack) |  |

Alert To를 생성한 Notifiers  
정보로 지정 선택 후 "Show  
advanced options" 선택하  
여 Group Interval Time과  
Repeat Interval Time 입력

→ Group Interval Time : 이미 발송된  
alert를 포함하는 그룹에 추가된 alert  
를 전송 전 대기시간, Repeat Interval  
Time : 이미 전송된 지정된 alert를 재  
전송하기 전 대기시간

## Rancher 모니터링 환경 구축하기 (계속)

### Slack 연동하기

# Deployment creation 이벤트 유발

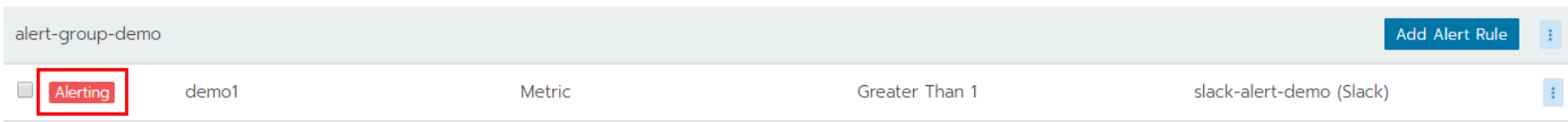
```
kubectl create deployment --image=nginx nginx1  
kubectl create deployment --image=nginx nginx2  
kubectl create deployment --image=nginx nginx3  
kubectl create deployment --image=nginx nginx4  
kubectl create deployment --image=nginx nginx5
```

```
root@zerovmu1804:~/gav-zero-demo# kubectl get deployment
```

| NAME    | READY | UP-TO-DATE | AVAILABLE | AGE |
|---------|-------|------------|-----------|-----|
| nginx1  | 1/1   | 1          | 1         | 92s |
| nginx10 | 1/1   | 1          | 1         | 79s |
| nginx2  | 1/1   | 1          | 1         | 91s |
| nginx3  | 1/1   | 1          | 1         | 89s |
| nginx4  | 1/1   | 1          | 1         | 88s |
| nginx5  | 1/1   | 1          | 1         | 87s |

# Rancher 및 AKS 설치하기 (계속)

## Slack 연동하기



잠시 후 구성했던 조건에 따라  
(Deployment 생성 1개 이상)  
demo1에 Alerting이 발생하여  
Slack에 Alert 생성

### #aks-rancher

You created this channel on April 13th. This is the very beginning of the #aks-rancher

[Add an app](#) [Add people](#)

**zerobig** 4:24 PM  
added an integration to this channel: [incoming-webhook](#)

**incoming-webhook** APP 6:42 PM  
Slack setting validated

**AlertManager** APP 7:41 PM  
The metric demo1 crossed the threshold  
Alert Name: demo1  
Severity: warning  
Cluster Name: zero-aks-rancher-demo (ID: c-bbnkg)  
Namespace: cattle-systemPod Name: exporter-kube-state-cluster-monitoring-587cbdfd9b-5c7xb  
Expression: kube\_deployment\_created>1  
[Show more](#)

The metric demo1 crossed the threshold  
Alert Name: demo1  
Severity: warning  
Cluster Name: zero-aks-rancher-demo (ID: c-bbnkg)  
Namespace: cattle-systemPod Name: exporter-kube-state-cluster-monitoring-587cbdfd9b-5c7xb  
Expression: kube\_deployment\_created>1  
[Show more](#)

# 전체 일감 처리 완료

To Do

+ New item

<

Doing

0/5

Done

<

612 Rancher 모니터링 환경 구축하기

zerobig.devops

State Done

2/2

+ Add Task

Grafana 구성하기

Slack 연동하기

611 Azure DevOps 릴리즈 파이프라인 구성하기

zerobig.devops

State Done

610 Rancher 및 AKS 설치하기

zerobig.devops

State Done

609 Azure DevOps 빌드 파이프라인 구성하기

zerobig.devops

State Done

605 서비스 이미지 준비 및 Azure Container Registry 생성하기

zerobig.devops

State Done

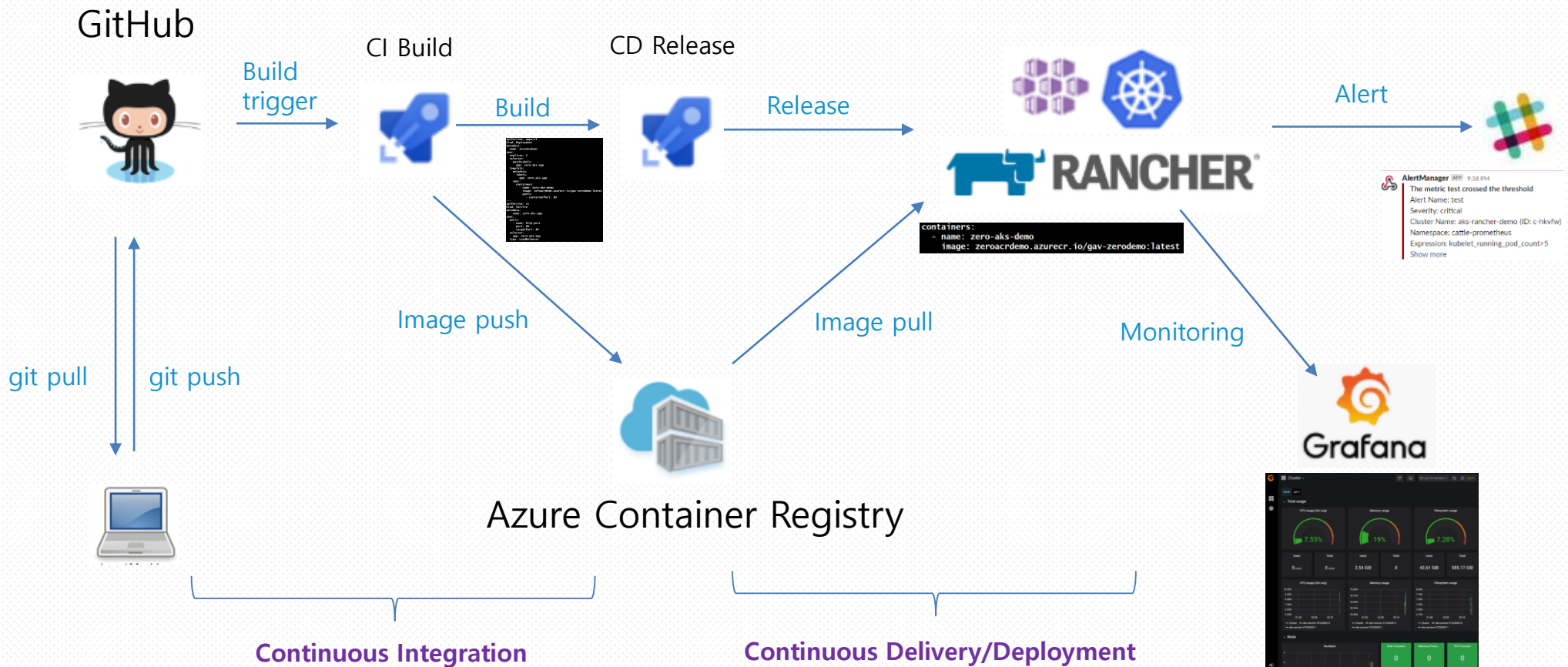
2/2

## RECAP.

Rancher에 AKS 구축하여 Azure Pipeline으로 Application 빌드/배포하기  
(그리고 Rancher에서 모니터링 환경 구성하기)

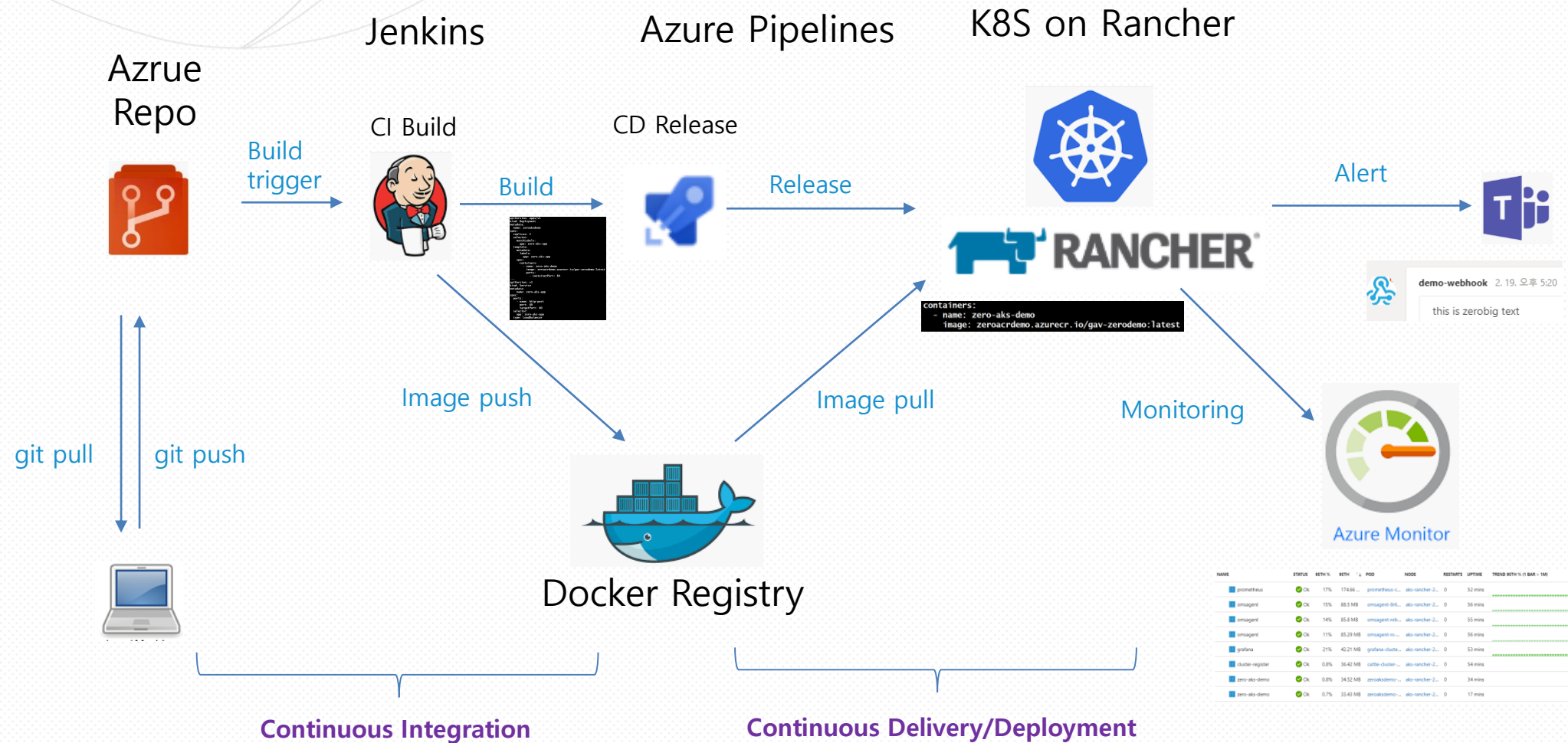
Azure Pipelines

AKS on Rancher



# RECAP.

## Rancher에 AKS 구축하여 Azure Pipeline으로 Application 빌드/배포하기 (그리고 Rancher에서 모니터링 환경 구성하기)

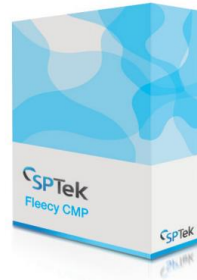




# SPTek 클라우드 솔루션 소개

## 프라이빗 클라우드 솔루션 제품

하이브리드 멀티 클라우드 환경에서 클라우드 서비스를 배포/운영/관리하는 통합관리 솔루션으로, 기업의 현 IT 업무환경을 기반으로 멀티 클라우드 환경을 자연스럽게 수용할 수 있도록 합니다. 개발 및 운영(DevOps)에 다양한 툴체인을 활용하여 VM/Docker/Kubernetes/Physical Machine 환경에 Application(Microservices)을 배포하기 위한 자동화된 CI/CD Pipeline을 제공합니다.



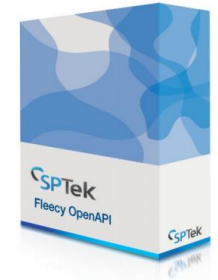
### Fleecy CMP

하이브리드 멀티 클라우드를 지원하는 클라우드 통합관리 솔루션



### Fleecy DevOps

개발에서 운영까지 Application 배포를 위한 CI/CD Pipeline 솔루션



### Fleecy OpenAPI

기업의 데이터나 서비스를 OpenAPI로 디지털 전환을 지원하는 솔루션

## 데이터센터 인프라 통합관리 솔루션 제품

데이터센터의 모든 리소스들을 소프트웨어 기반으로 관리하며, 물리 및 가상자원의 복잡한 관리를 자동화 해주는 차세대 데이터 센터 통합관리 운영환경을 제공합니다.

- MD Edition : Multi DC 지원
- SC Edition : 스케줄링 기반의 자원 예약관리 지원



### Cloud Portal Solution

사용자의 간편 신청/이용 관리자의 수준 높은 운영관리 포털



### DevOps Solution

컨테이너 기반의 DevOps, 개발에서 운영까지 자동화된 프로세스



### Automation Solution

H/W, S/W 정보수집, S/W 설치/배포/설정 관리 등 자동화

**DevOps Gold Partner** 역량 보유 **SPTek** <http://www.sptek.co.kr/>

## 즐거로운 Azure 생활



<https://www.facebook.com/groups/azurouslife/>



# 쿠버네티스 문서 한글화 참여 방법

**누구나 이슈, PR을 등록하고 리뷰 할 수 있습니다!**

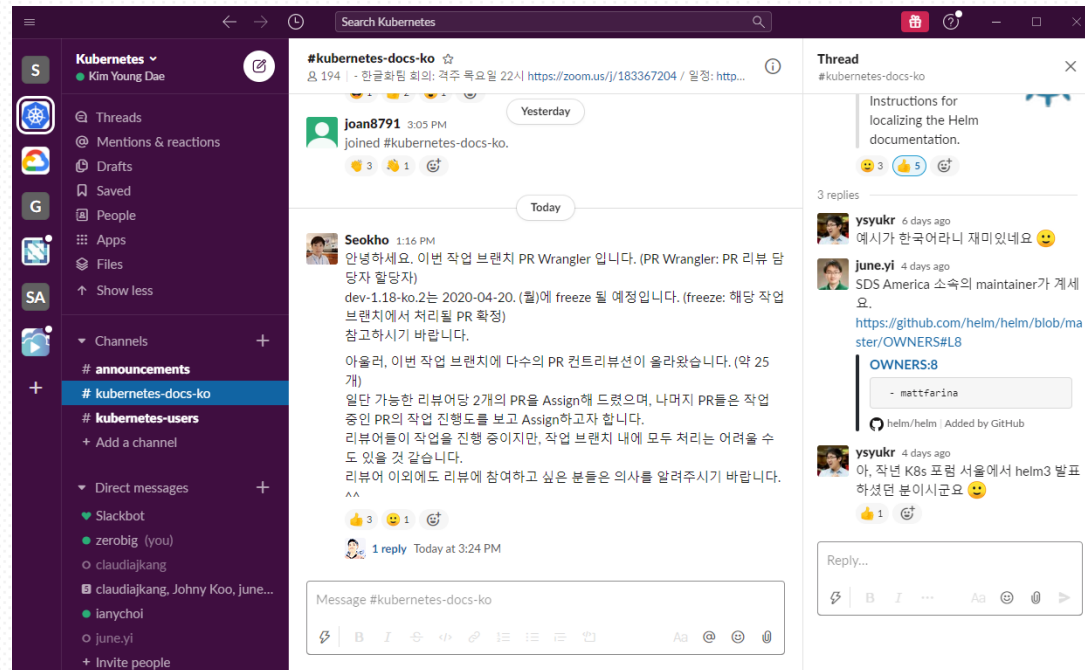
(1) 한글화 팀 Slack 채널: Kubernetes Slack 가입 후,  
[#kubernetes-docs-ko](#) 채널 참여

(2) 한글화 팀 회의: 격주 목요일 22시 ~ 23시. Zoom([#kubernetes-docs-ko](#)에 링크 공유)에  
서 열립니다. (회의록: <https://goo.gl/1gZ45u>)

(3) 문서 기여 방법

<https://github.com/kubernetes/website/blob/master/README-ko.md>

(4) 한글화 스타일 가이드 [https://kubernetes.io/ko/docs/contribute/localization\\_ko/](https://kubernetes.io/ko/docs/contribute/localization_ko/)





Thank you | Mobile & Convergence Leading Company SPTek