Serverless Vulnerabilities Analysis

2023.07.03.

Team. 구름빵

임우협(PM) 고동현

유지예 이성주

최보민 최하늘

www.CodeEngn.com

2023 CodeEngn Conference 19

CodeEngn

발표자 소개

- 이름
  - 임우협 (brwook)
- 소속
  - 세종대학교 Security Factorial 동아리
- 관심 분야
  - Pwnable
  - Cloud
  - Microservice Architecture
- 이메일
  - brwook@naver.com

목차

프로젝트 소개

1. 프로젝트 배경
2. 프로젝트 주제

프로젝트 진행

1. 진행 상황 요약
2. Azure Container Instances 취약점 분석
3. 데모 영상

프로젝트 결론

1. 버그바운티 결과

Ⅰ 프로젝트 소개

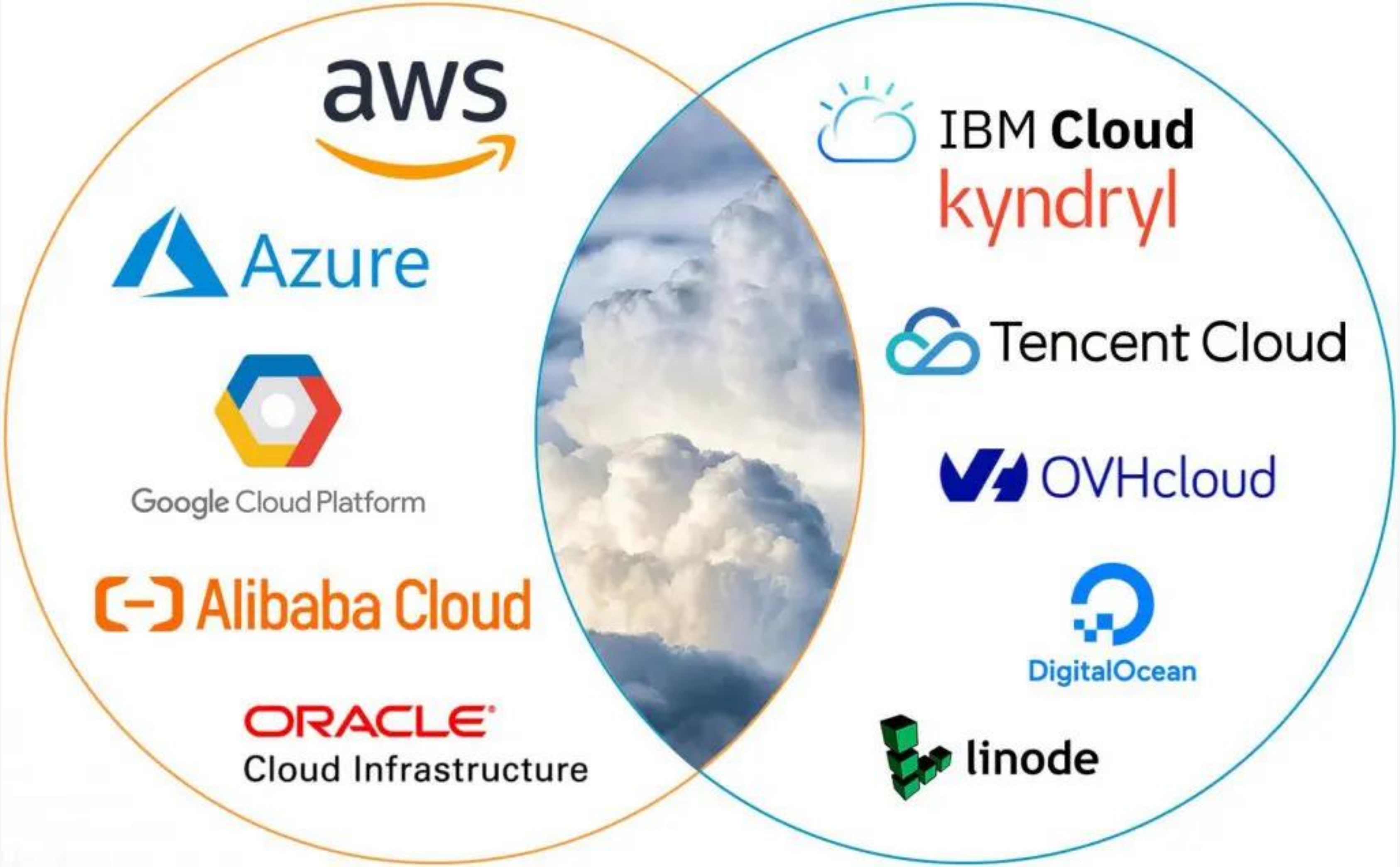
- 프로젝트 배경

- 프로젝트 주제

프로젝트 배경

▶ 클라우드란?

- Public Cloud Service Provider(CSP)

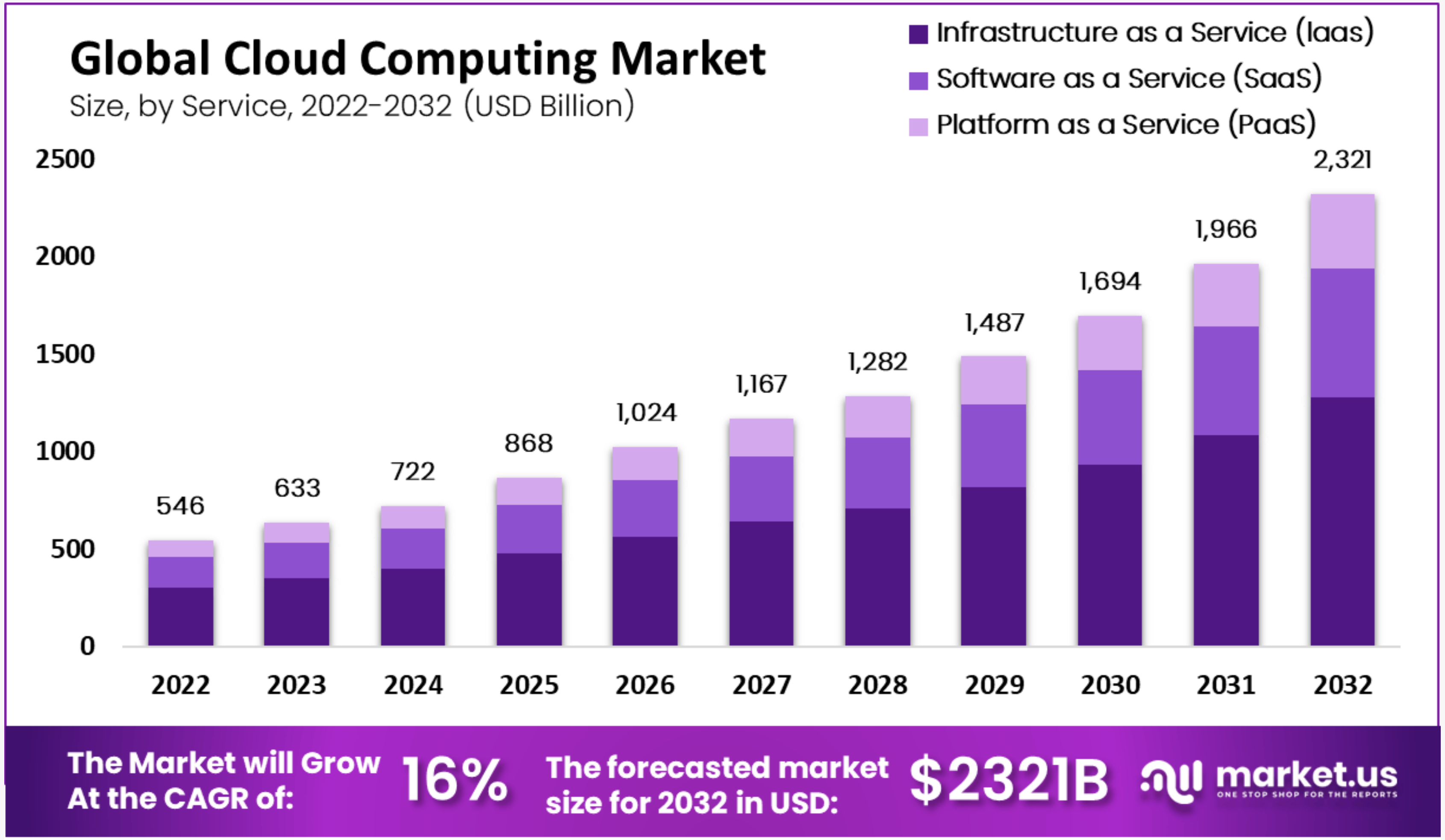


1) "https://dgtlinfra.com/top-10-cloud-service-providers-2022", Dgtl Infra

프로젝트 배경

▶ 날로 커져가는 클라우드 시장

- 클라우드의 가용성, 탄력성, 비용 절감, 다양한 서비스, 보안성, …

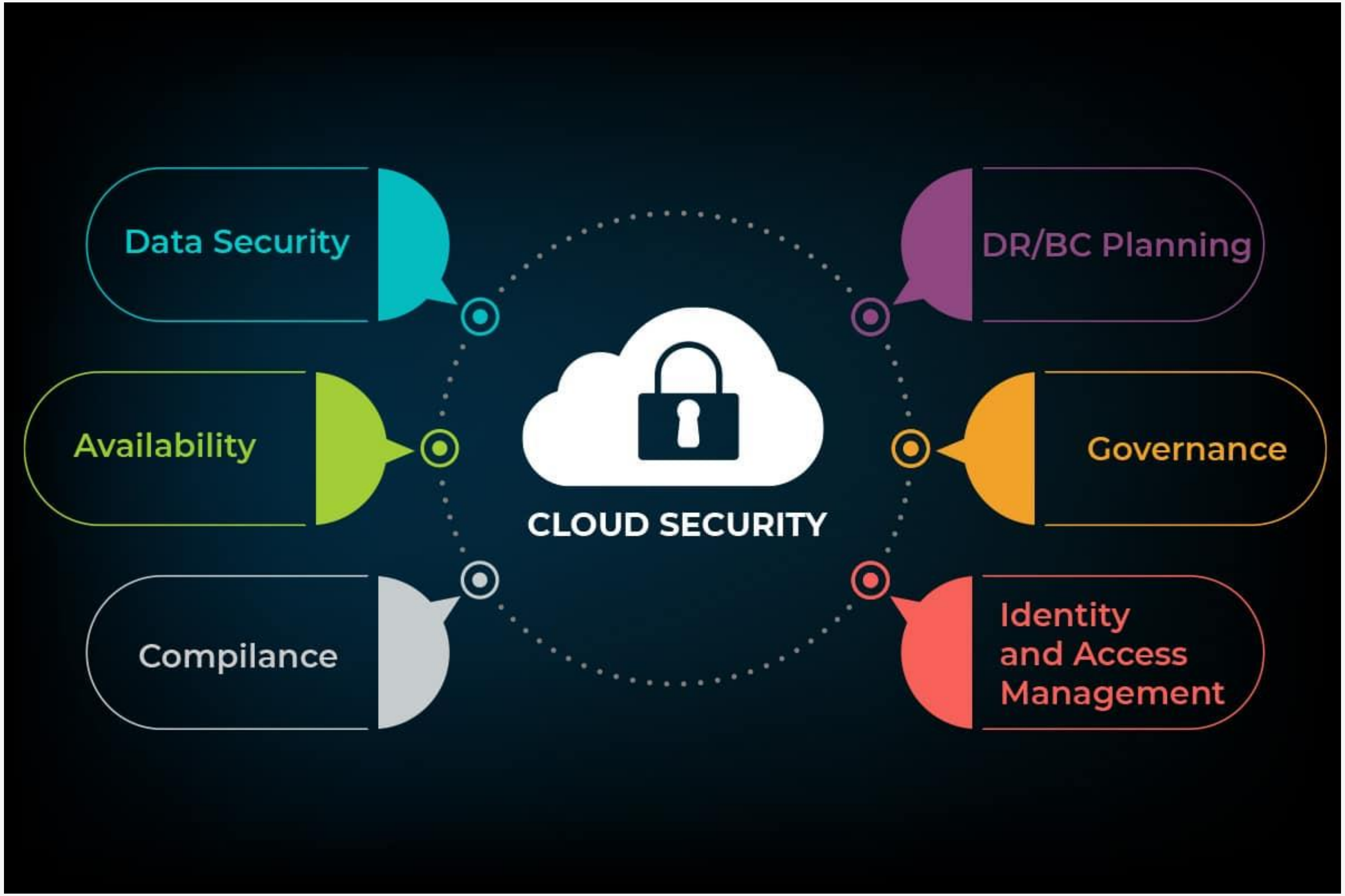


1) "https://www.globenewswire.com/en/news-release/2023/03/17/2629610/0/en/Cloud-Computing-Market-to-Reach-USD-2-321-1-Billion-by-2032-Exploring-the-Diverse-Applications-of-Cloud-Computing.html", Globe Newsire

프로젝트 배경

▶ 결국 중요한 것은 보안

- CSP에서 취약점이 발견될 경우,
  해당 서비스를 이용하는
  모든 클라이언트에게 영향을 미침

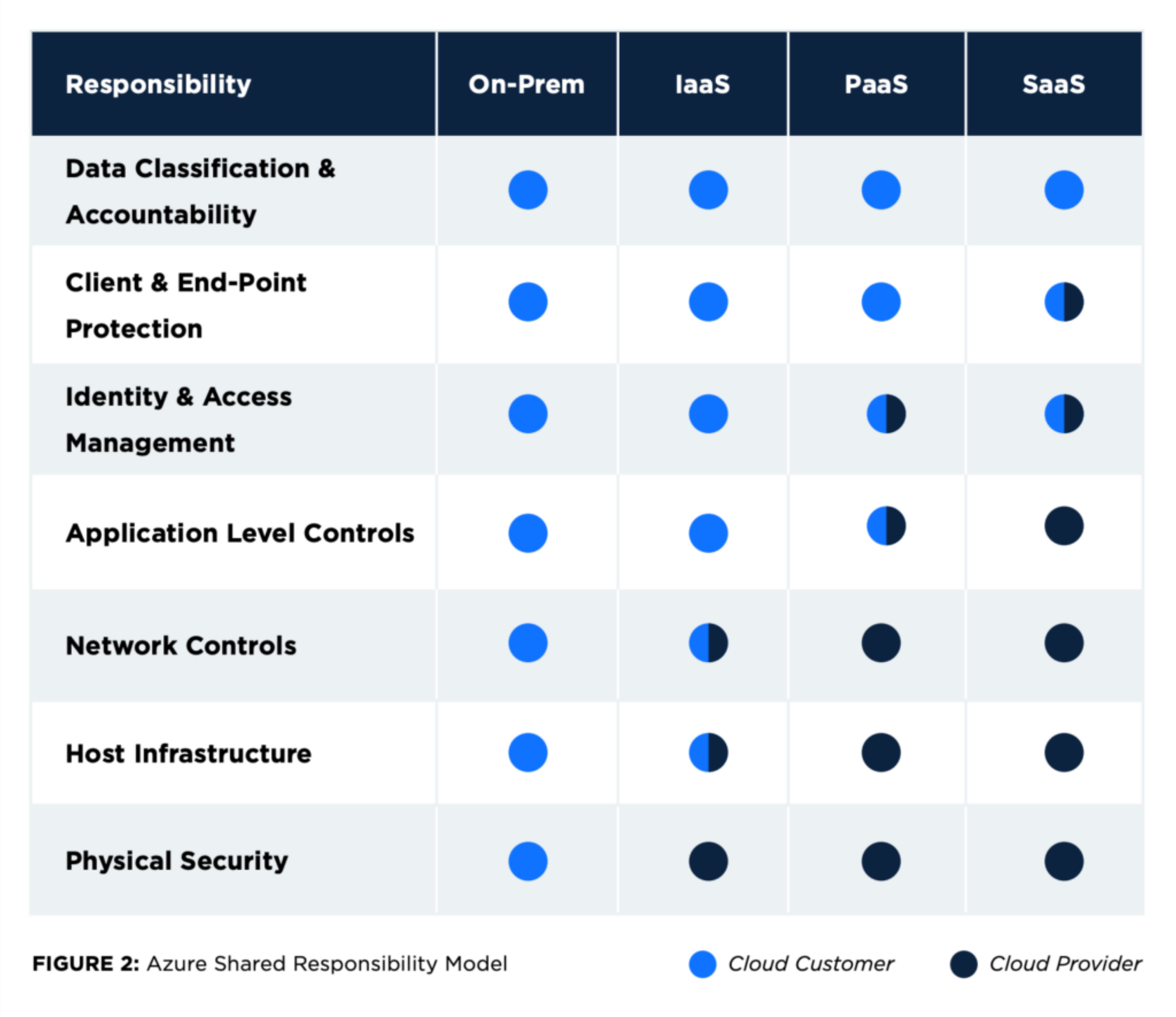


1) “https://www.eescorporation.com/cloud-security-a-detailed-guide”, Enterprise Engineering Solutions

프로젝트 배경

▶ CSP 취약점?

- Shared Responsibility Model(SRM)에서 CSP가 책임지는 부분에서 발생하는 취약점



Responsibility	On-Prem	IaaS	PaaS	SaaS
Data Classification & Accountability	🔵	🔵	🔵	🔵
Client & End-Point Protection	🔵	🔵	🔵	◐
Identity & Access Management	🔵	🔵	◐	◐
Application Level Controls	🔵	🔵	◐	●
Network Controls	🔵	◐	●	●
Host Infrastructure	🔵	◐	●	●
Physical Security	🔵	●	●	●

FIGURE 2: Azure Shared Responsibility Model

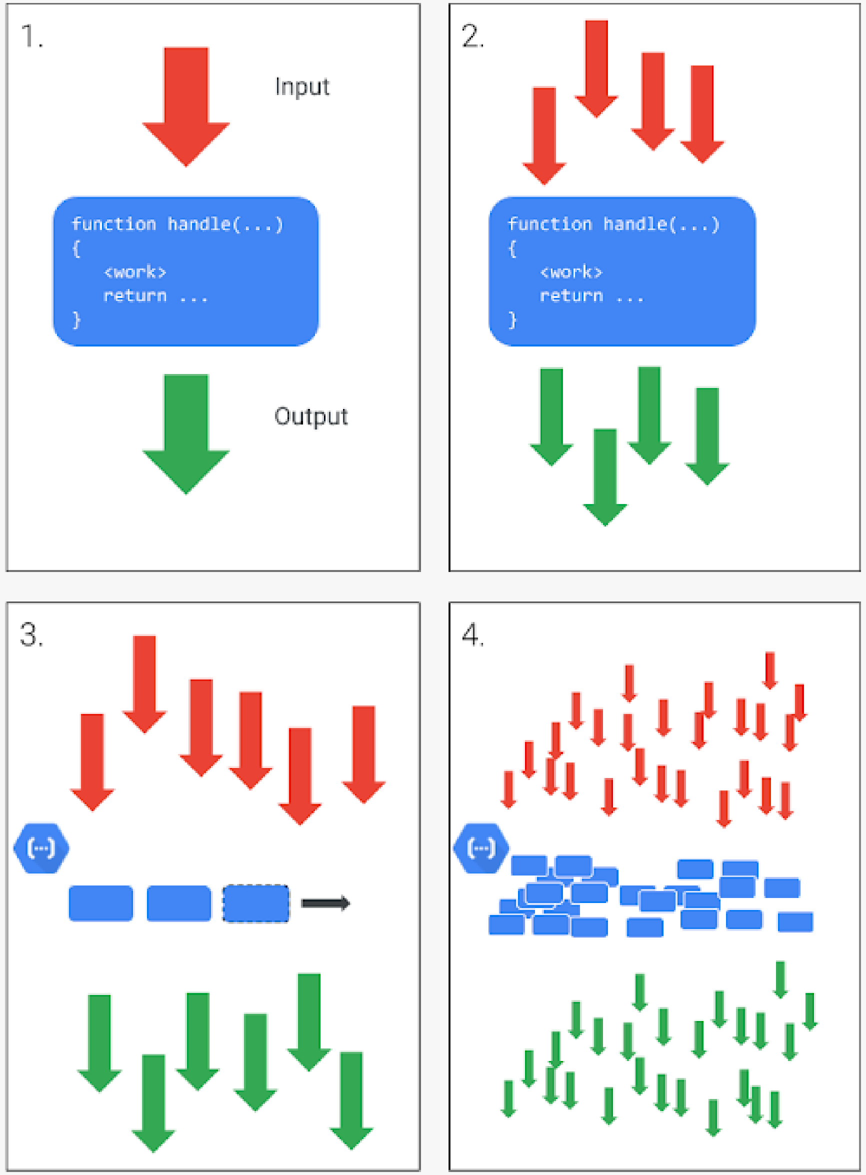
🔵 *Cloud Customer* ● *Cloud Provider*

1) “https://cloudcheckr.com/cloud-security/shared-responsibility-model/”, CloudCheckr

프로젝트 배경

▶ CSP 취약점 이해를 위한 예시

- FaaS는 요청에 따라 스케일링
- FaaS에서 발생하는 Denial of Wallet(DoW) 공격



```
function handle(...)
{
   <work>
   return ...
}
```

```
function handle(...)
{
   <work>
   return ...
}
```

1) “https://cloud.google.com/blog/products/serverless/6-strategies-for-scaling-your-serverless-applications?hl=en”, Google Cloud

2) Unsplash, Jp Valery

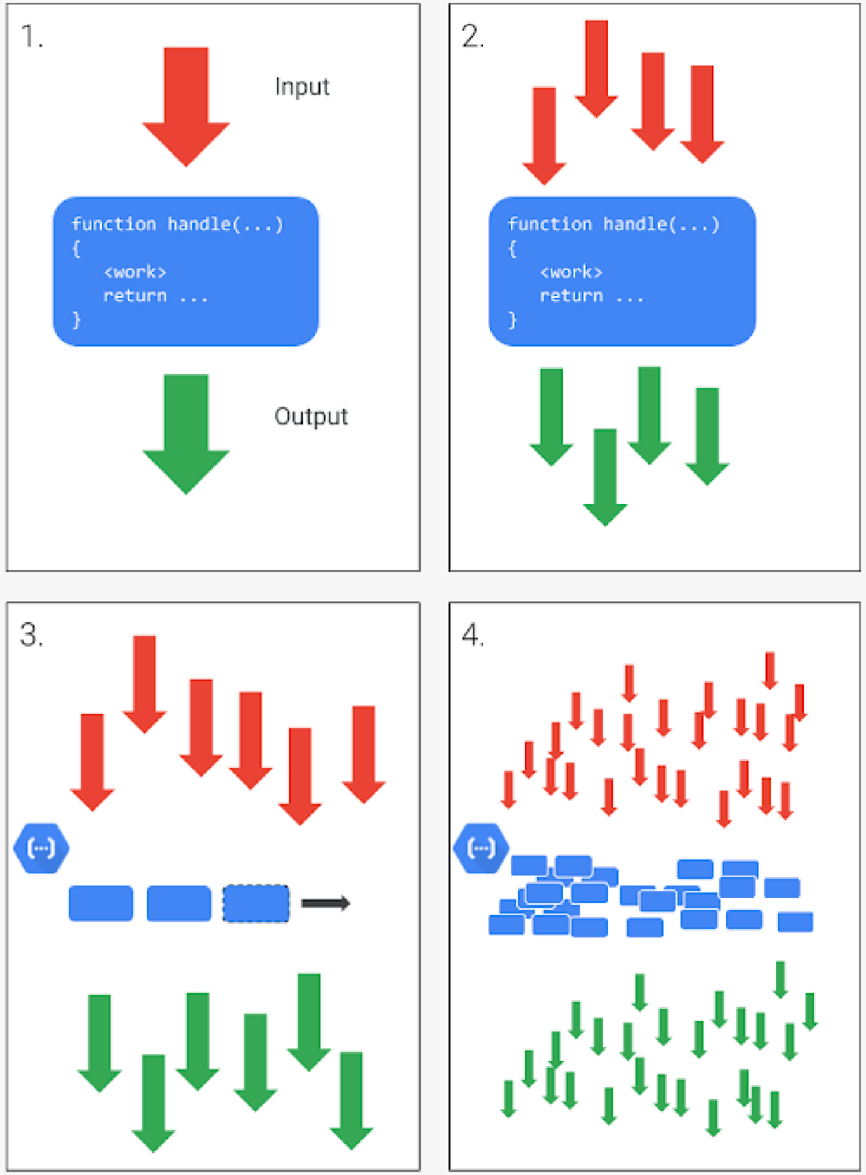
프로젝트 배경

▶ CSP 취약점 이해를 위한 예시

- (1) CSP에서 FaaS의 요청 수에 대해 제한을 두지 않았다면?
- (2) 기본 제한을 클라이언트가 의도적으로 해제했다면?

1) “https://cloud.google.com/blog/products/serverless/6-strategies-for-scaling-your-serverless-applications?hl=en”, Google Cloud

2) Unsplash, Jp Valery



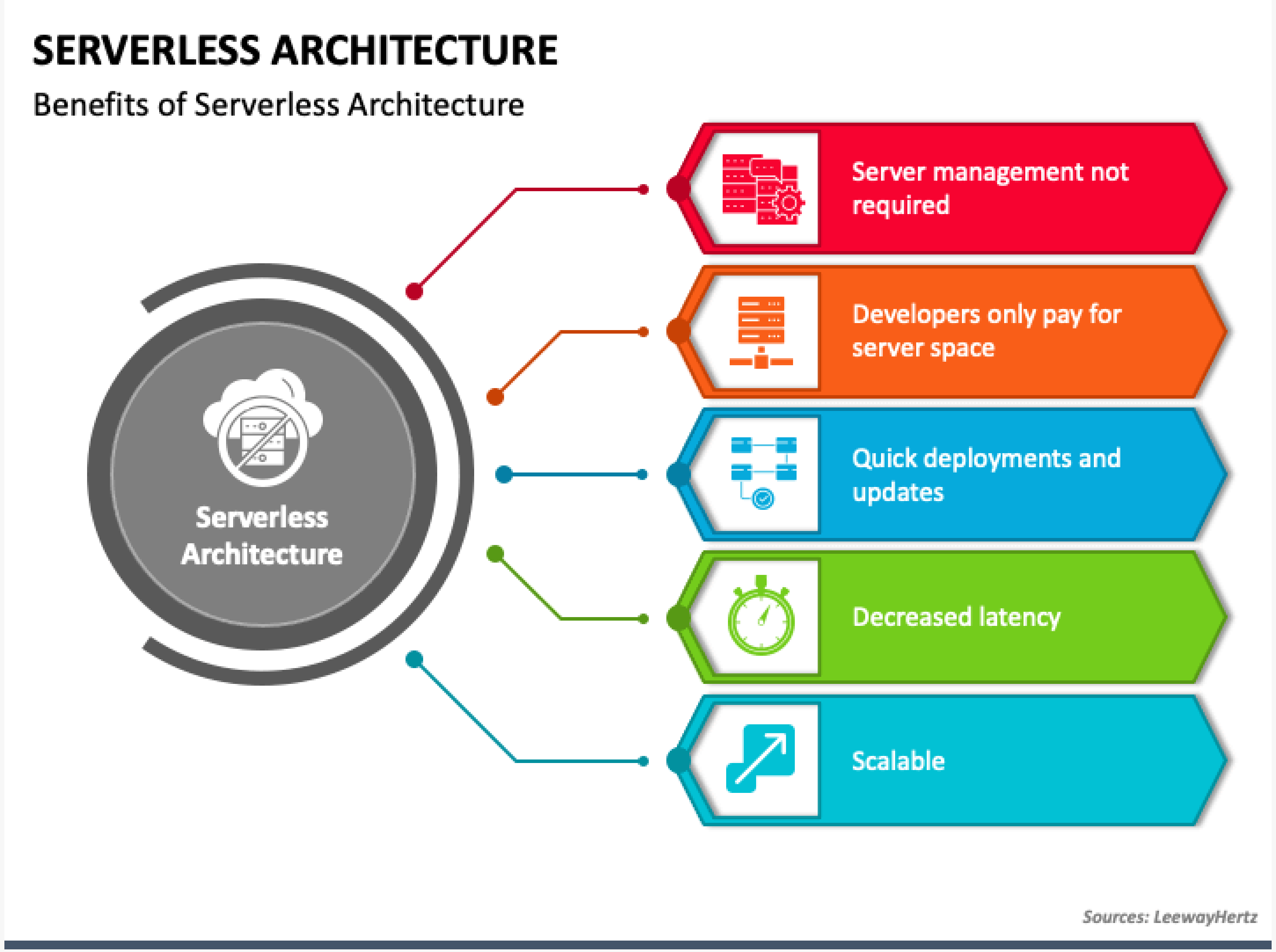
```
function handle(...)
{
   <work>
   return ...
}
```

```
function handle(...)
{
   <work>
   return ...
}
```

프로젝트 주제

▶ Serverless?

- 서버를 관리할 필요 없이 서비스를 배포 및 운영



SERVERLESS ARCHITECTURE

Benefits of Serverless Architecture

Serverless Architecture

Server management not required

Developers only pay for server space

Quick deployments and updates

Decreased latency

Scalable

Sources: LeewayHertz

1) “https://www.sketchbubble.com/en/presentation-serverless-architecture.html”, SketchBubble

프로젝트 주제

프로젝트(Serverless Vulnerabilities Analysis)의 주제

Serverless Computing Service 분석을 통해

클라우드 환경을 사용자에게 가시화하고 취약점을 찾아보자!

Ⅱ 프로젝트 진행

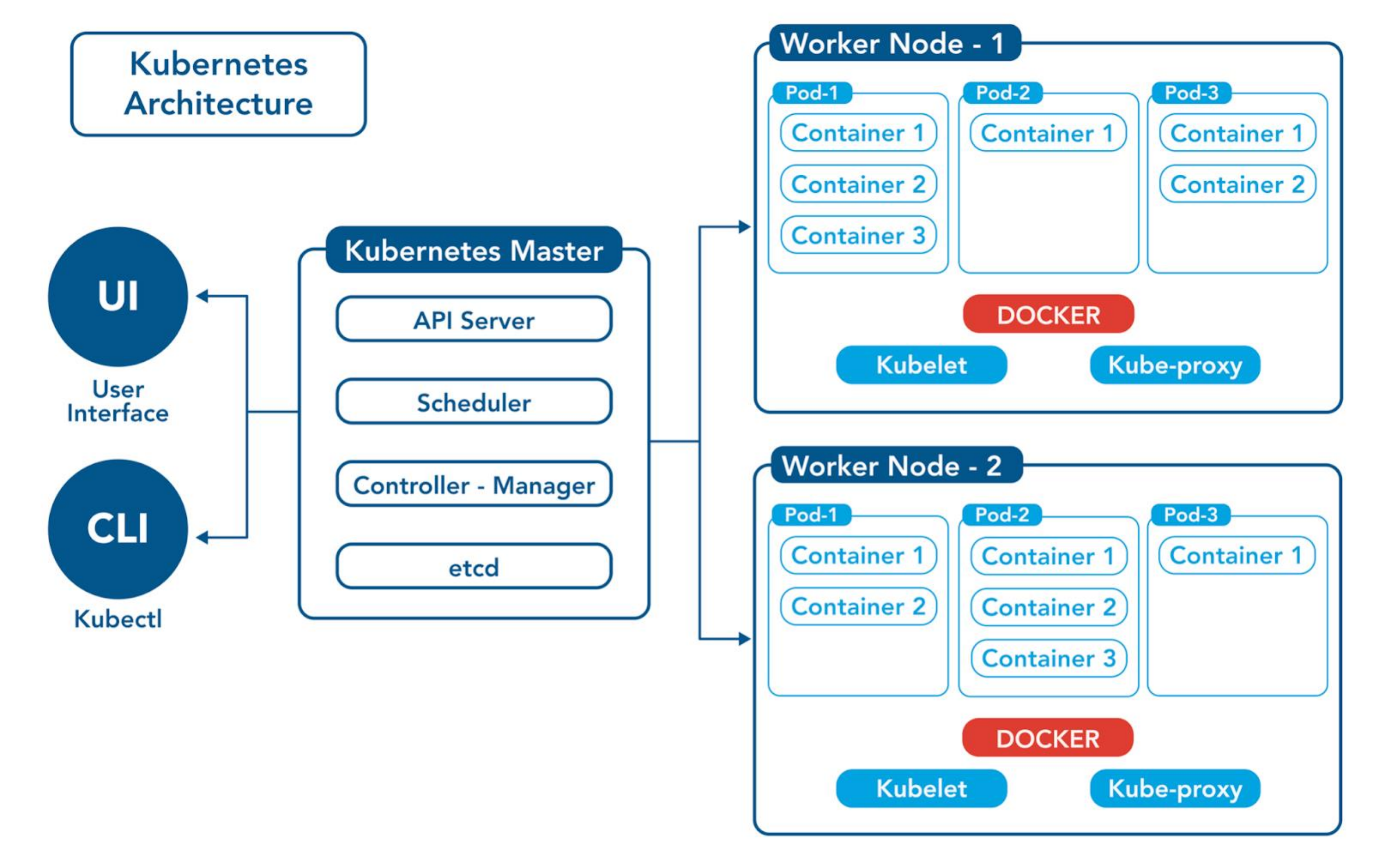
- CVE 분석 내용 요약

- Azure Container Instances 취약점 분석

[참고1] Serverless 구조

▶ 마이크로서비스 아키텍처

- 독립성, 확장성, 배포 용이



1) “https://www.opsramp.com/guides/why-kubernetes/kubernetes-architecture”, OpsRamp

CVE 분석 내용 요약

Docker, Kubernetes, Serverless 1-Day 분석

1. 사전 지식 습득

- Docker 구축 및 기능 연구
- Kubernetes 구축 및 기능 연구
- Serverless Service 서비스 사용 및 기능 연구

2. 원데이 분석

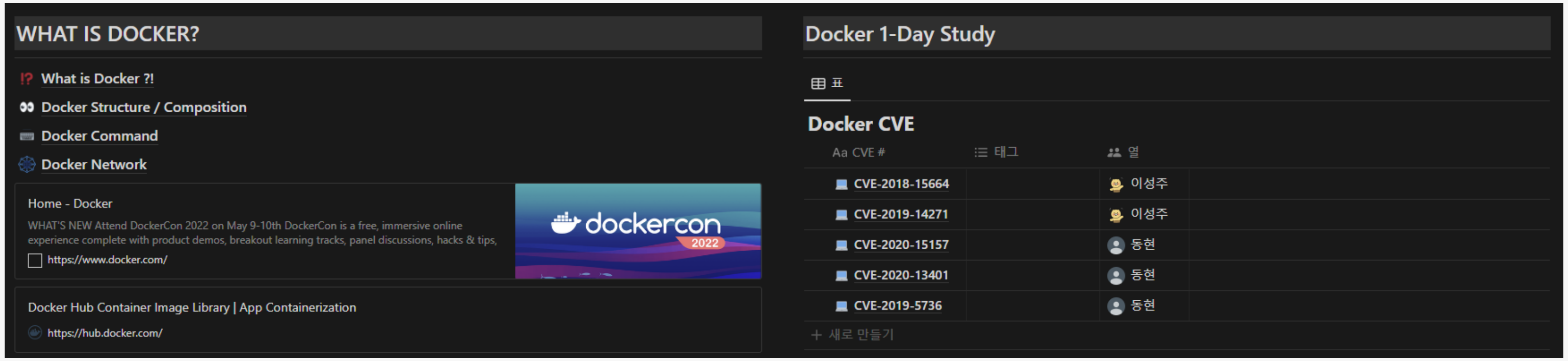
- Docker, Kubernetes 원데이 분석
- 클라우드 서비스 원데이 분석

  ex. OMIGOD, AzureEscape

CVE 분석 내용 요약

▶ Docker, k8s CVE

- Root Cause 분석 및 재연



WHAT IS DOCKER?

- What is Docker ?!
- Docker Structure / Composition
- Docker Command
- Docker Network

Home - Docker

WHAT'S NEW Attend DockerCon 2022 on May 9-10th DockerCon is a free, immersive online experience complete with product demos, breakout learning tracks, panel discussions, hacks & tips,

https://www.docker.com/

Docker Hub Container Image Library | App Containerization

https://hub.docker.com/

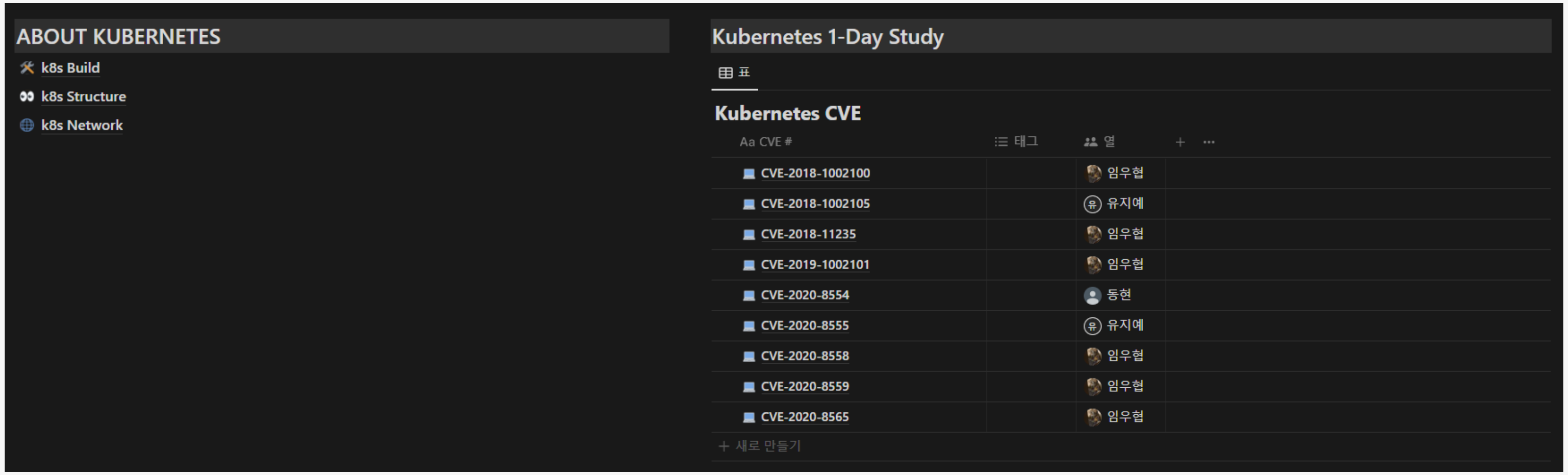
Docker 1-Day Study

표

Docker CVE

Aa CVE #	태그	열
CVE-2018-15664		이성주
CVE-2019-14271		이성주
CVE-2020-15157		동현
CVE-2020-13401		동현
CVE-2019-5736		동현

+ 새로 만들기



ABOUT KUBERNETES

- k8s Build
- k8s Structure
- k8s Network

Kubernetes 1-Day Study

표

Kubernetes CVE

Aa CVE #	태그	열
CVE-2018-1002100		임우협
CVE-2018-1002105		유지예
CVE-2018-11235		임우협
CVE-2019-1002101		임우협
CVE-2020-8554		동현
CVE-2020-8555		유지예
CVE-2020-8558		임우협
CVE-2020-8559		임우협
CVE-2020-8565		임우협

+ 새로 만들기

CVE 분석 내용 요약

Docker, Kubernetes, Serverless 1-Day 분석

1. 사전 지식 습득

- Docker 구축 및 기능 연구
- Kubernetes 구축 및 기능 연구
- Serverless Service 서비스 사용 및 기능 연구

2. 원데이 분석

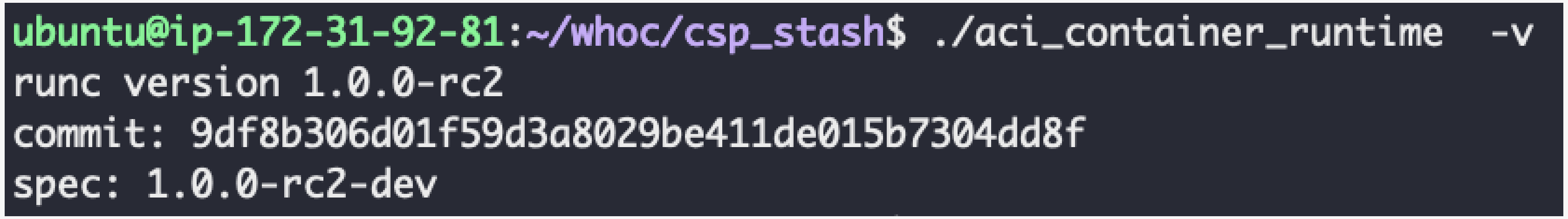
- Docker, Kubernetes 원데이 분석
- 클라우드 서비스 원데이 분석

ex. OMIGOD, AzureEscape

CVE 분석 내용 요약

▶ AzureEscape

- 2021년 9월 패치된 Azure Container Instances 취약점
- CVE-2019-5736을 활용한 컨테이너 이스케이프



```
ubuntu@ip-172-31-92-81:~/whoc/csp_stash$ ./aci_container_runtime  -v
runc version 1.0.0-rc2
commit: 9df8b306d01f59d3a8029be411de015b7304dd8f
spec: 1.0.0-rc2-dev
```

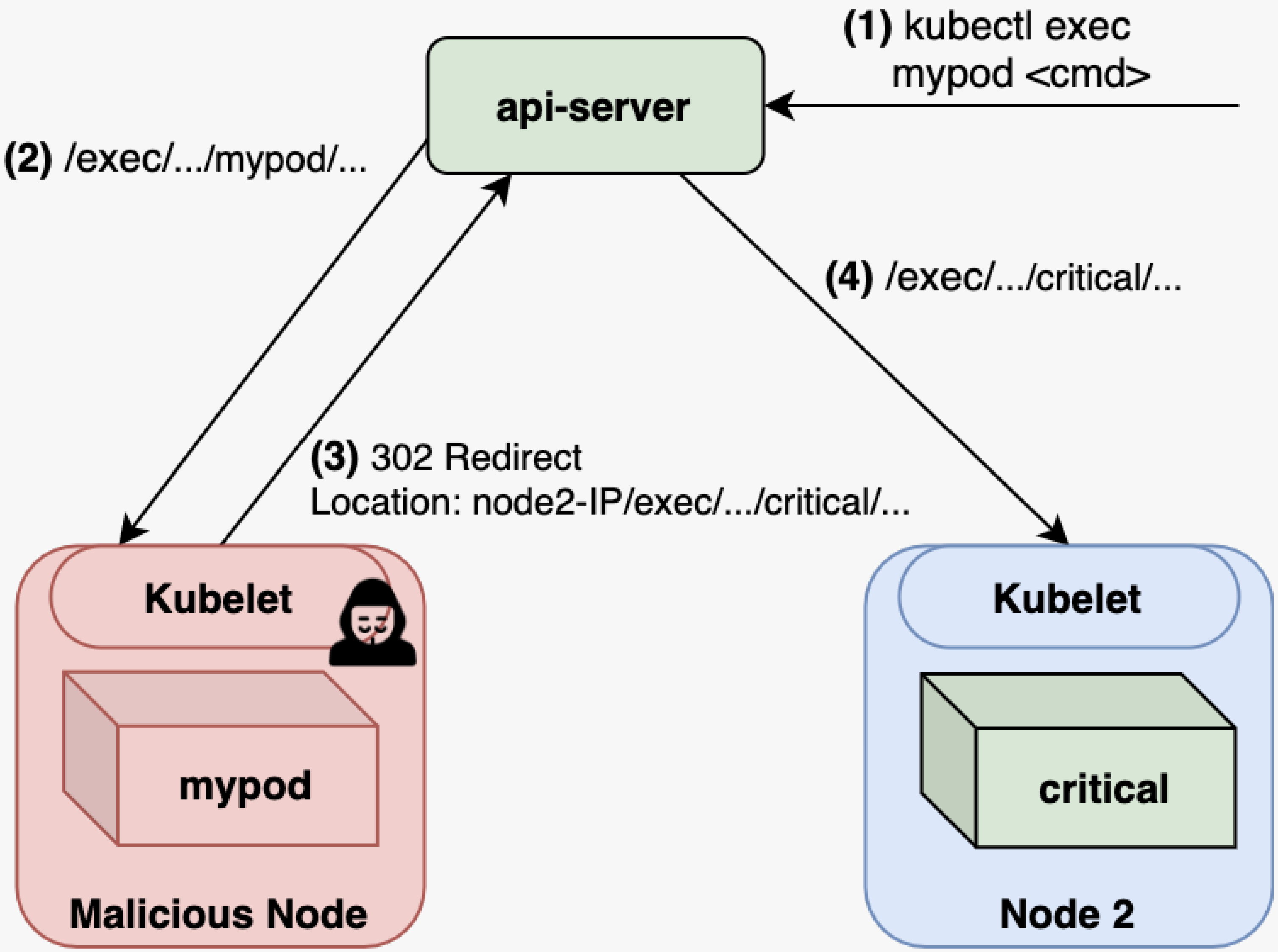
1) “https://blog.aquasec.com/azurescape-azure-container-instances”, aqua

2) https://unit42.paloaltonetworks.com/azure-container-instances/, unit42

CVE 분석 내용 요약

▶ AzureEscape (1)

- CVE-2018-1002102를 활용한 클러스터 전체 장악



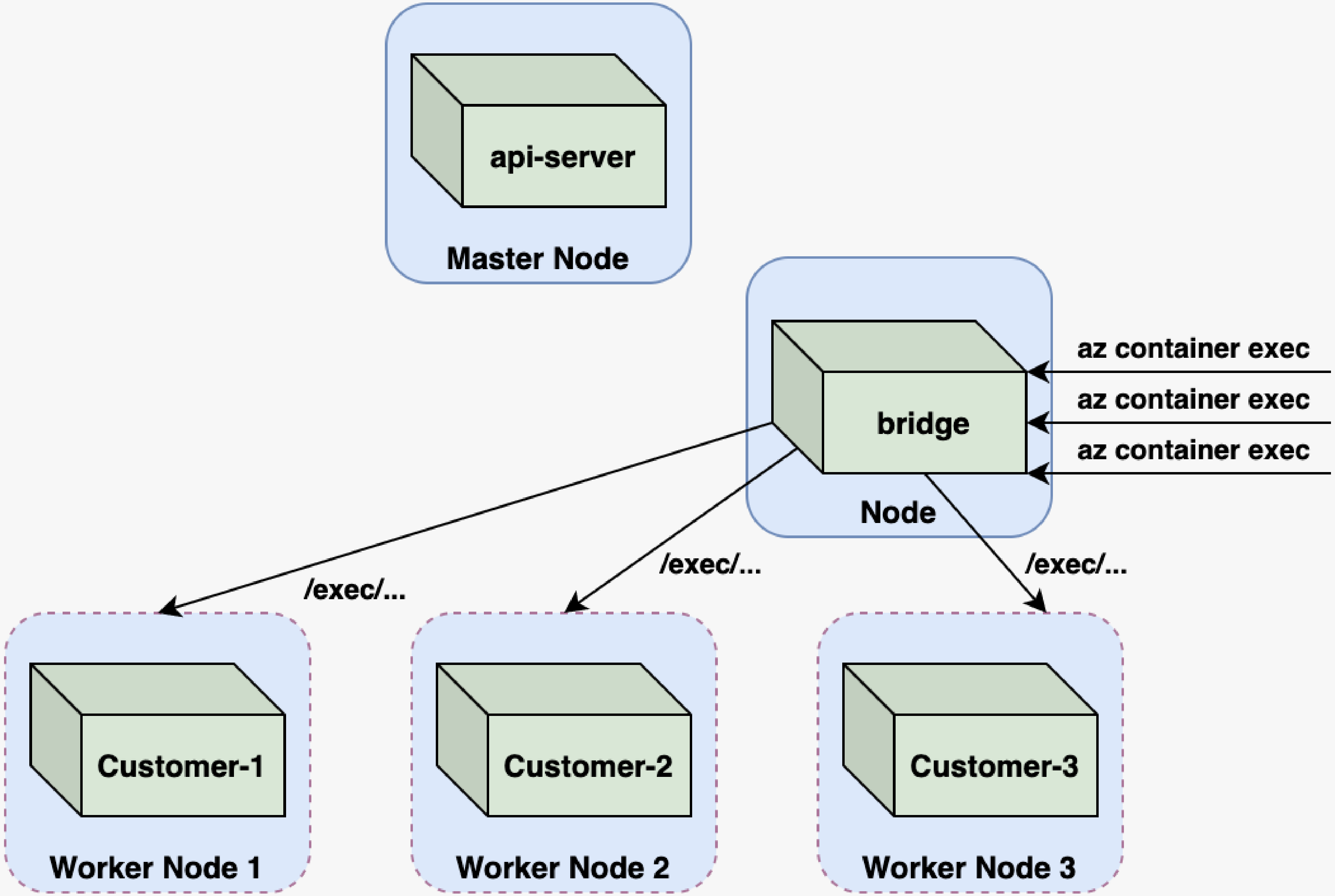
1) https://unit42.paloaltonetworks.com/azure-container-instances/, unit42

CVE 분석 내용 요약

▶ AzureEscape (2)

- Bridge Pod에서 발생하는 Credential Leak
- Credential을 활용한 클러스터 전체 장악

1) https://unit42.paloaltonetworks.com/azure-container-instances/, unit42



[참고2]
블랙박스 테스팅

▶ CSP 자체 제작 보안 기술

- 화이트페이퍼 혹은 오픈소스로 공개되어 있는 경우가 왕왕 있음

1) “https://aws.amazon.com/ko/blogs/opensource/firecracker-open-source-secure-fast-microvm-serverless”, AWS

2) https://en.wikipedia.org/wiki/Gvisor, Wikipedia

3) https://contextere.com/Blog/experiences-microsoft-azure-service-fabric/, contextere

진행 상황 요약

AWS Lambda, Azure Functions·Container Instances, GCP Cloud Functions·Cloud Run 취약점 분석

3-1. 취약점 분석 (AWS Lambda)

- AWS Lambda에서 WHOC 이미지를 배포하여 컨테이너 런타임 획득 시도
- AWS Lambda에서 Symlink Exchange Attack을 시도해 TOC-TOU 유도

3-2. 취약점 분석 (Azure Functions, GCP Functions)

- Azure Functions, GCP Functions: MITM attack을 통한 Information Disclosure 시도, Kernel Exploitation을 통한 Privilege Escalation 시도

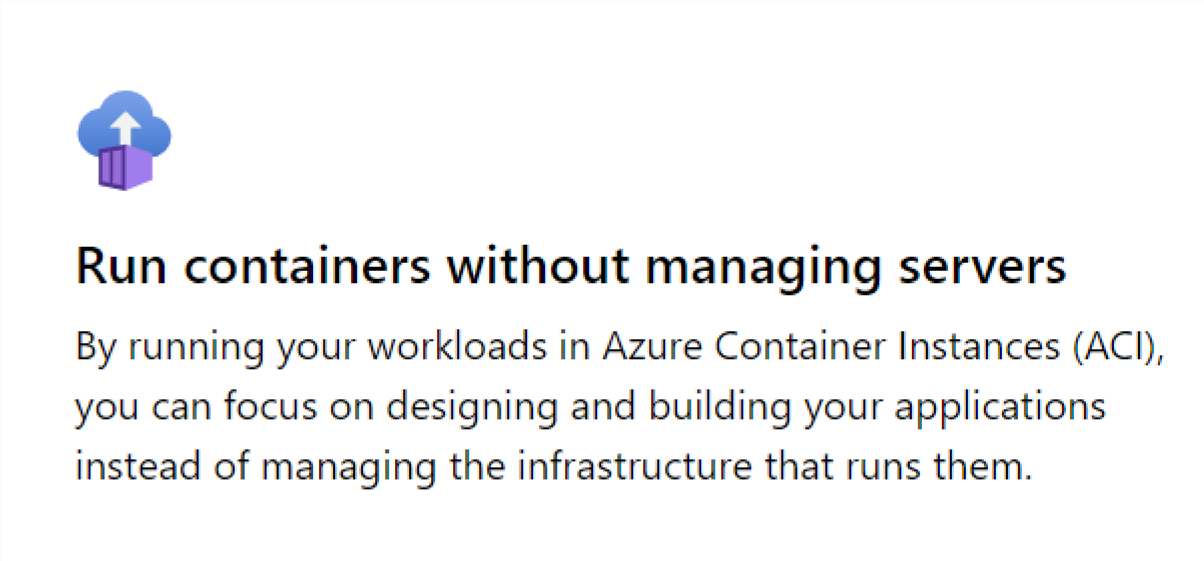
3-3. 취약점 분석 (Azure Container Instances, GCP Run)

- ACI: Container Escape 및 Worker node 장악
- GCP Run: Process Injection을 통한 Privilege Escalation 시도

Azure Container Instances 취약점 분석

▶ Azure Container Instances

- 컨테이너 설정 파일을 입력으로 주면, 컨테이너들을 직접 부팅 시켜주는 서비스



Run containers without managing servers

By running your workloads in Azure Container Instances (ACI), you can focus on designing and building your applications instead of managing the infrastructure that runs them.

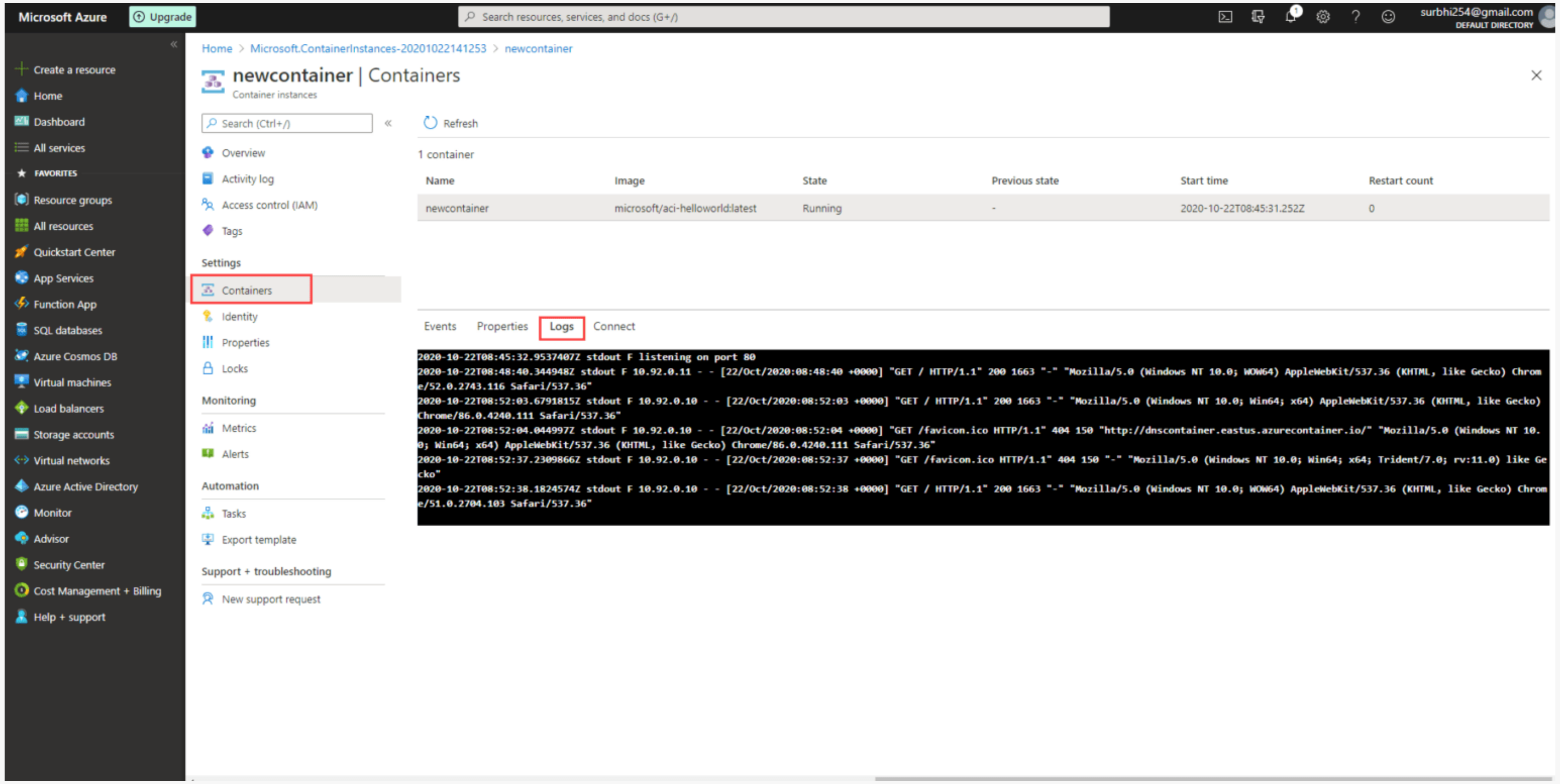
1) “https://azure.microsoft.com/en-us/products/container-instances”, Azure

Azure Container Instances 취약점 분석

▶ Azure Container Instances

- 웹 인터페이스를 통해 컨테이너 메타 정보와 출력 로그 확인

1) “https://k21academy.com/microsoft-azure/case-study-deploy-a-container-instance-in-azure-using-the-azure-portal”, K21Academy



1 container

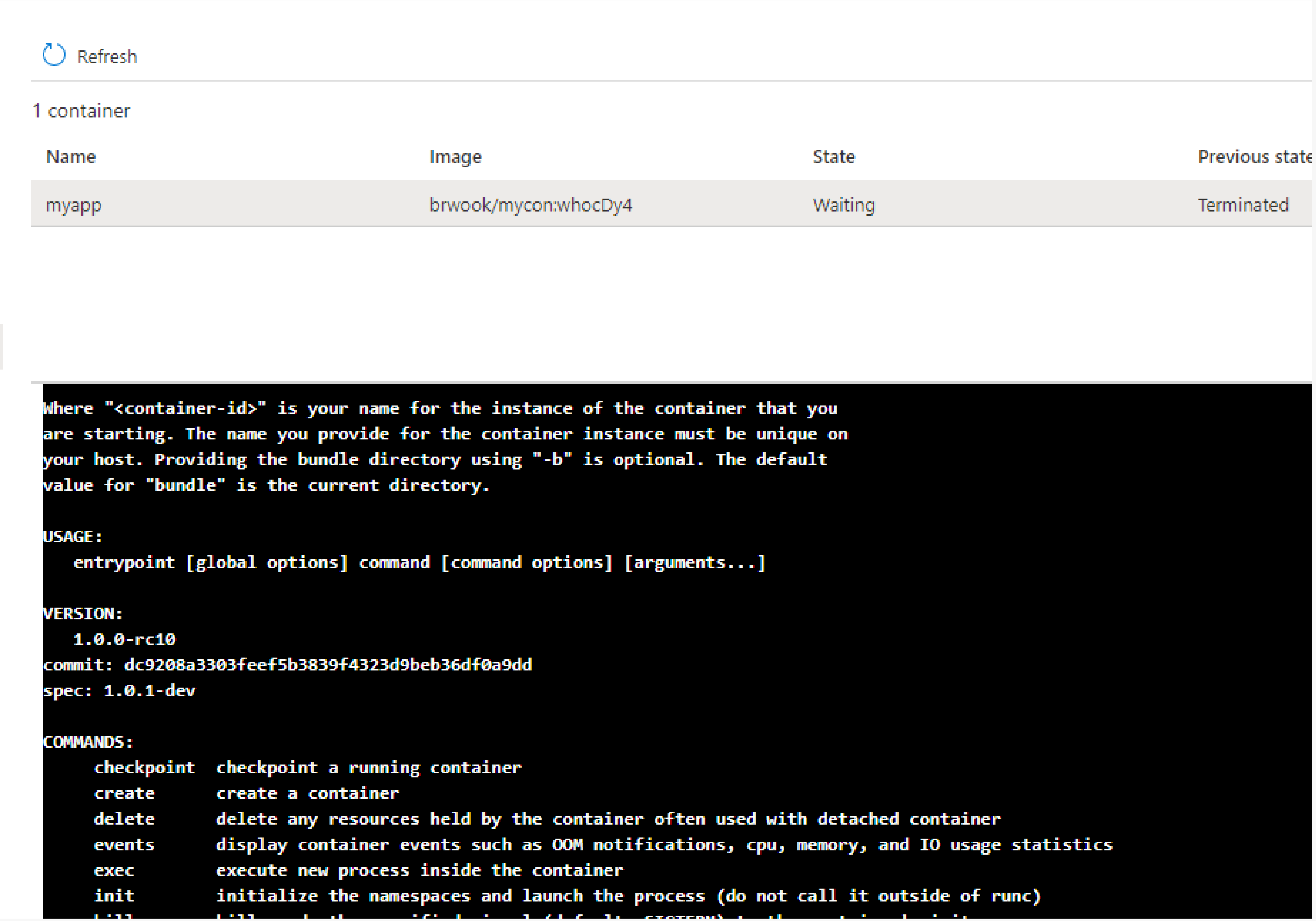
Name	Image	State	Previous state	Start time	Restart count
newcontainer	microsoft/aci-helloworld:latest	Running	-	2020-10-22T08:45:31.252Z	0

```
2020-10-22T08:45:32.9537407Z stdout F listening on port 80
2020-10-22T08:48:40.344948Z stdout F 10.92.0.11 - - [22/Oct/2020:08:48:40 +0000] "GET / HTTP/1.1" 200 1663 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/52.0.2743.116 Safari/537.36"
2020-10-22T08:52:03.6791815Z stdout F 10.92.0.10 - - [22/Oct/2020:08:52:03 +0000] "GET / HTTP/1.1" 200 1663 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.111 Safari/537.36"
2020-10-22T08:52:04.044997Z stdout F 10.92.0.10 - - [22/Oct/2020:08:52:04 +0000] "GET /favicon.ico HTTP/1.1" 404 150 "http://dnscontainer.eastus.azurecontainer.io/" "Mozilla/5.0 (Windows NT 10.
0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.111 Safari/537.36"
2020-10-22T08:52:37.2309866Z stdout F 10.92.0.10 - - [22/Oct/2020:08:52:37 +0000] "GET /favicon.ico HTTP/1.1" 404 150 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; Trident/7.0; rv:11.0) like Ge
cko"
2020-10-22T08:52:38.1824574Z stdout F 10.92.0.10 - - [22/Oct/2020:08:52:38 +0000] "GET / HTTP/1.1" 200 1663 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/51.0.2704.103 Safari/537.36"
```

Azure Container Instances 취약점 분석

▶ 컨테이너 런타임 버전 확인

- “/proc/self/exe” ENTRYPOINT로 지정
- 컨테이너 런타임 바이너리 실행 및 웹을 통한 출력 확인



1 container

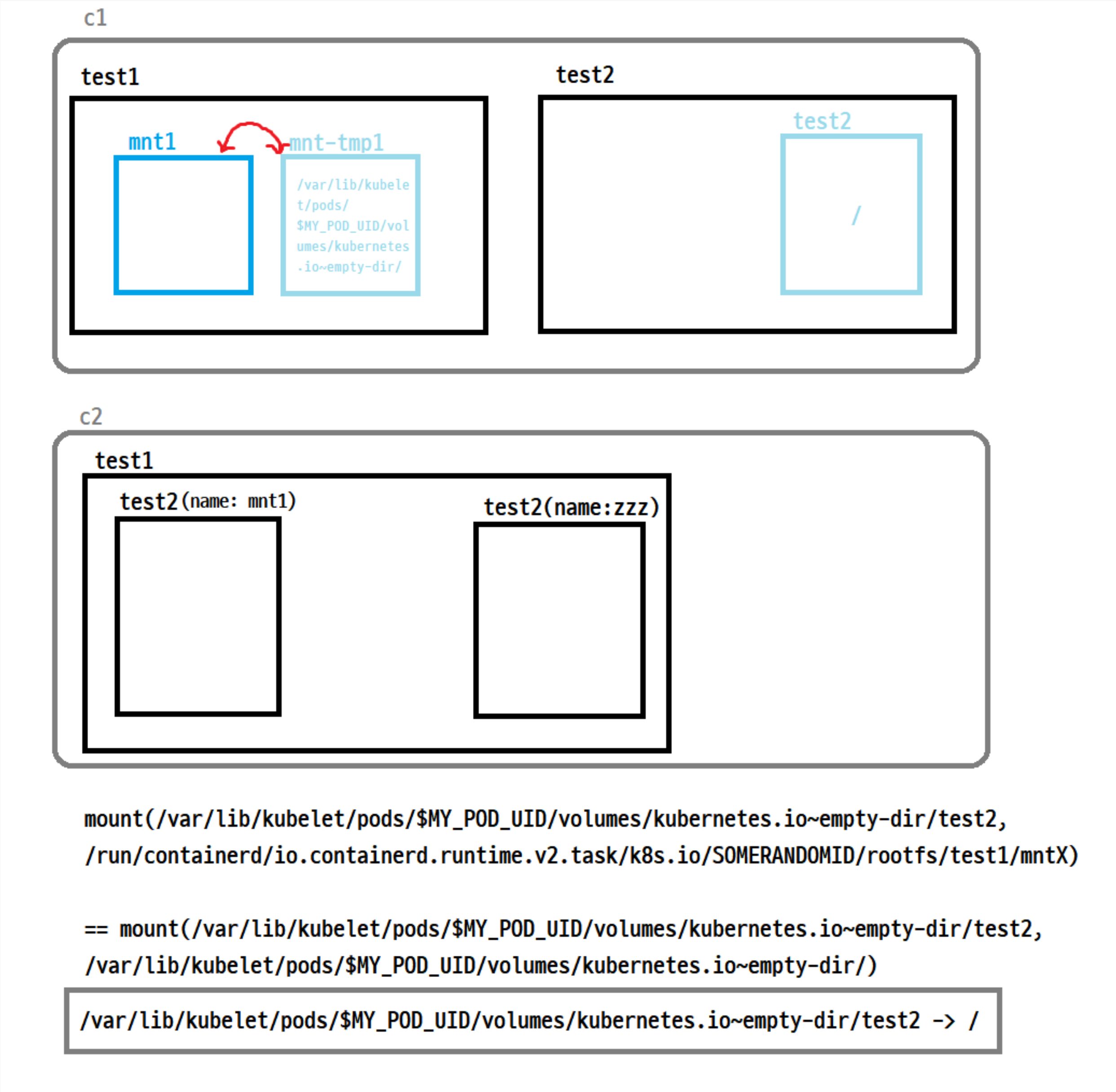
Name	Image	State	Previous state
myapp	brwook/mycon:whocDy4	Waiting	Terminated

```
Where "<container-id>" is your name for the instance of the container that you
are starting. The name you provide for the container instance must be unique on
your host. Providing the bundle directory using "-b" is optional. The default
value for "bundle" is the current directory.

USAGE:
   entrypoint [global options] command [command options] [arguments...]

VERSION:
   1.0.0-rc10
commit: dc9208a3303feef5b3839f4323d9beb36df0a9dd
spec: 1.0.1-dev

COMMANDS:
     checkpoint  checkpoint a running container
     create      create a container
     delete      delete any resources held by the container often used with detached container
     events      display container events such as OOM notifications, cpu, memory, and IO usage statistics
     exec        execute new process inside the container
     init        initialize the namespaces and launch the process (do not call it outside of runc)
```

Azure Container Instances 취약점 분석

▶ CVE-2021-30465 적용가능

- “1.0.0-rc94 <=“ 해당 버전일 경우, symlink exchange attack(TOCTOU) 수행 가능



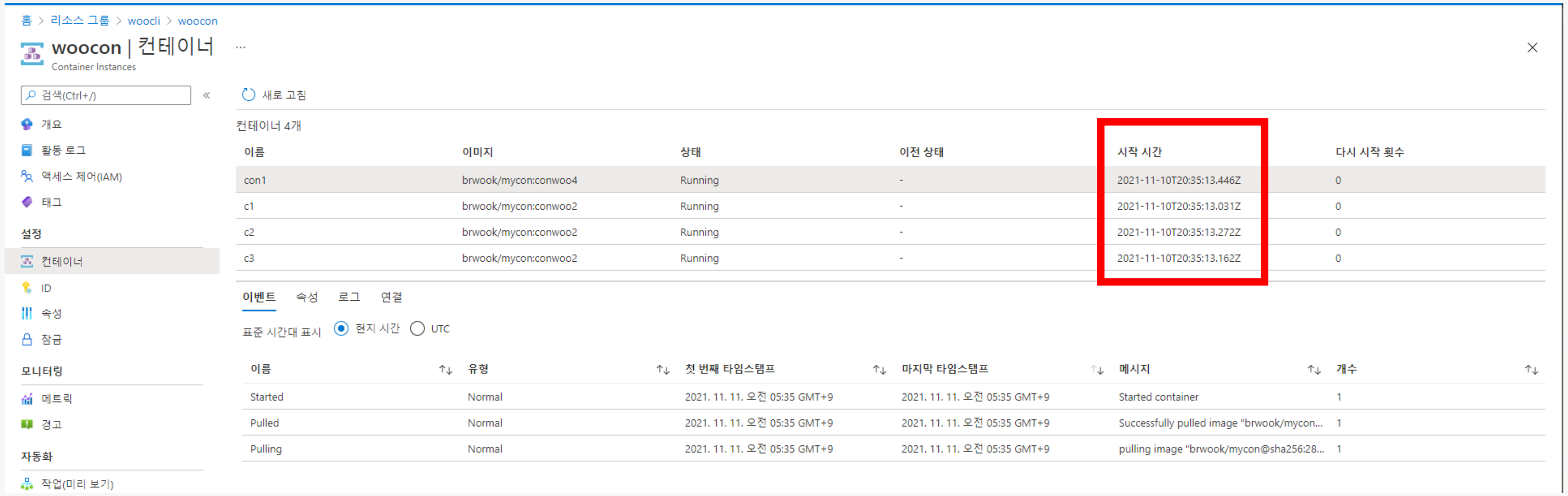
```
mount(/var/lib/kubelet/pods/$MY_POD_UID/volumes/kubernetes.io~empty-dir/test2,
/run/containerd/io.containerd.runtime.v2.task/k8s.io/SOMERANDOMID/rootfs/test1/mntX)

== mount(/var/lib/kubelet/pods/$MY_POD_UID/volumes/kubernetes.io~empty-dir/test2,
/var/lib/kubelet/pods/$MY_POD_UID/volumes/kubernetes.io~empty-dir/)
```

```
/var/lib/kubelet/pods/$MY_POD_UID/volumes/kubernetes.io~empty-dir/test2 -> /
```

Azure Container Instances 취약점 분석

▶ ACI 제약 사항 (1)

- 컨테이너들이 동시에 시작해서 symlink exchange attack 을 수행하기까지 시간 필요
- Start와 Stop을 반복하다가, 특정 조건에 따라 Running 상태를 유지하는 컨테이너 개발



컨테이너 4개

이름	이미지	상태	이전 상태	시작 시간	다시 시작 횟수
con1	brwook/mycon:conwoo4	Running	-	2021-11-10T20:35:13.446Z	0
c1	brwook/mycon:conwoo2	Running	-	2021-11-10T20:35:13.031Z	0
c2	brwook/mycon:conwoo2	Running	-	2021-11-10T20:35:13.272Z	0
c3	brwook/mycon:conwoo2	Running	-	2021-11-10T20:35:13.162Z	0

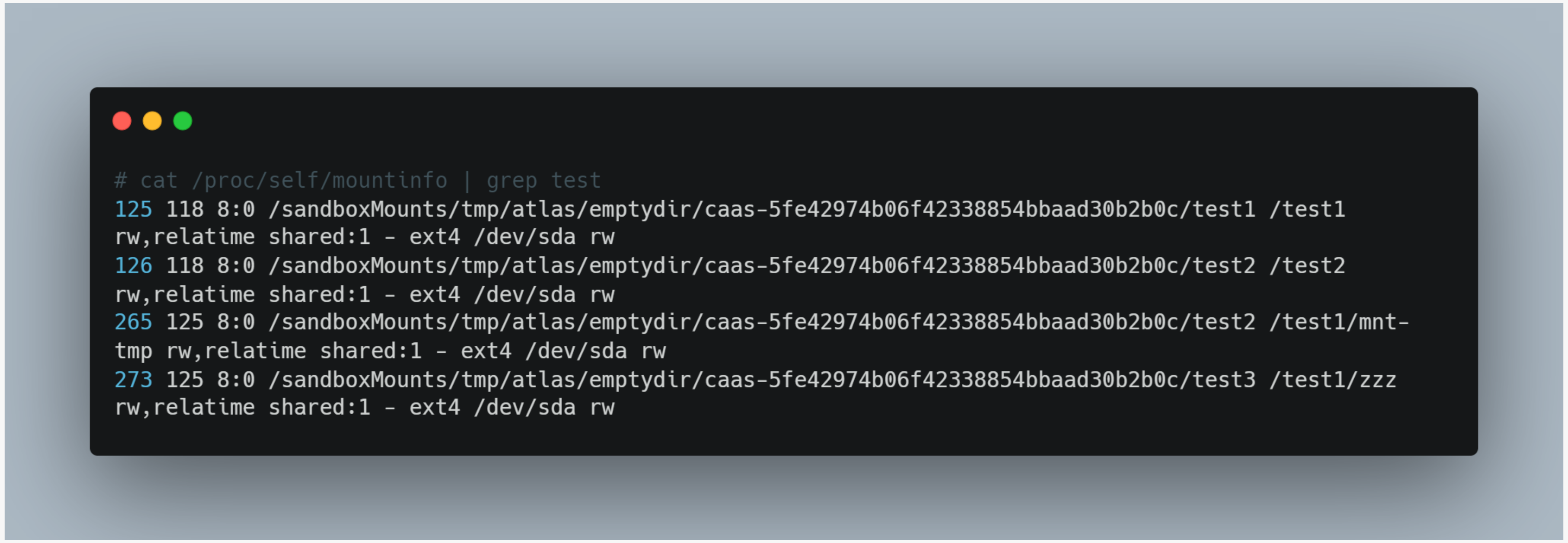
표준 시간대 표시 ● 현지 시간 ○ UTC

이름	유형	첫 번째 타임스탬프	마지막 타임스탬프	메시지	개수
Started	Normal	2021. 11. 11. 오전 05:35 GMT+9	2021. 11. 11. 오전 05:35 GMT+9	Started container	1
Pulled	Normal	2021. 11. 11. 오전 05:35 GMT+9	2021. 11. 11. 오전 05:35 GMT+9	Successfully pulled image "brwook/mycon...	1
Pulling	Normal	2021. 11. 11. 오전 05:35 GMT+9	2021. 11. 11. 오전 05:35 GMT+9	pulling image "brwook/mycon@sha256:28...	1

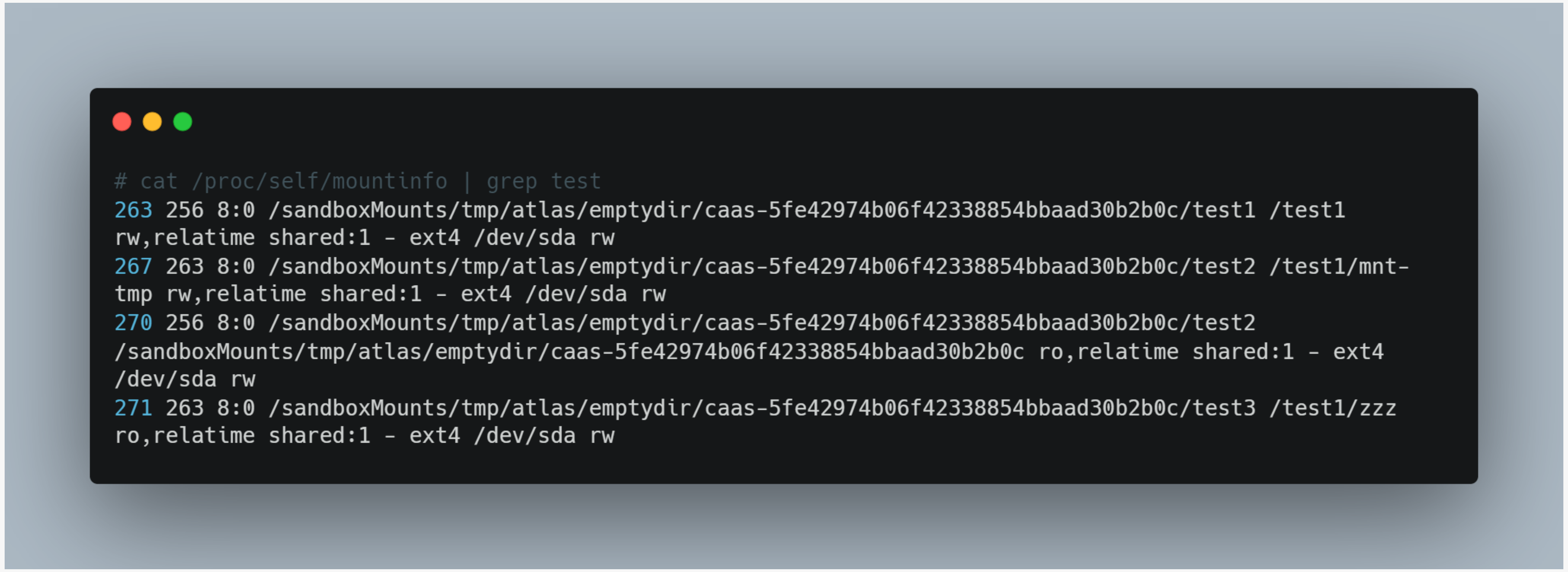
Azure Container Instances 취약점 분석

▶ ACI 제약 사항 (2)

- emptyDir 볼륨을 생성할 수 있을뿐, emptyDir.medium 필드를 설정할 수 없음
- tmpfs가 아닌, ext4로 할당



```
# cat /proc/self/mountinfo | grep test
125 118 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test1 /test1
rw,relatime shared:1 - ext4 /dev/sda rw
126 118 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test2 /test2
rw,relatime shared:1 - ext4 /dev/sda rw
265 125 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test2 /test1/mnt-
tmp rw,relatime shared:1 - ext4 /dev/sda rw
273 125 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test3 /test1/zzz
rw,relatime shared:1 - ext4 /dev/sda rw
```

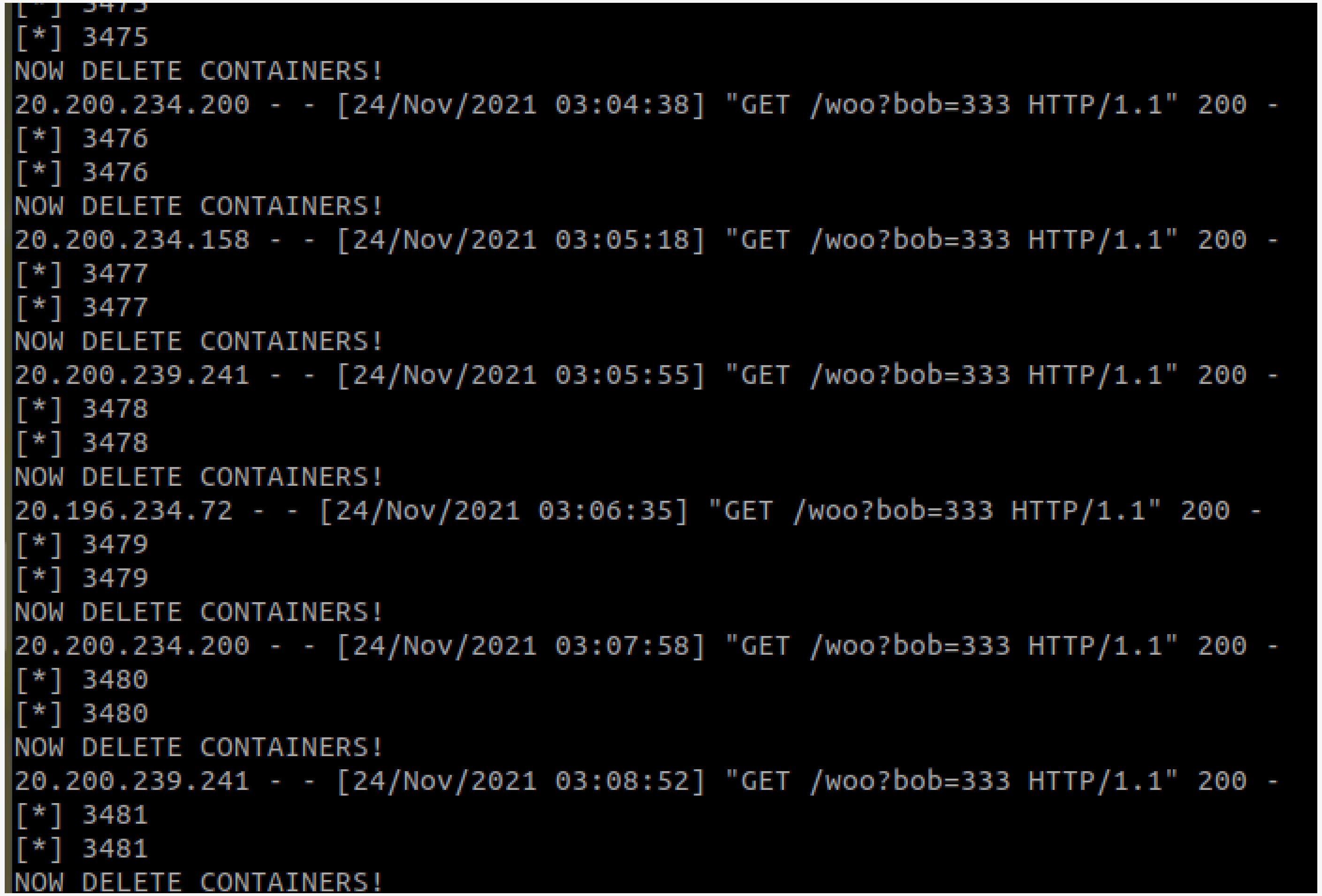


```
# cat /proc/self/mountinfo | grep test
263 256 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test1 /test1
rw,relatime shared:1 - ext4 /dev/sda rw
267 263 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test2 /test1/mnt-
tmp rw,relatime shared:1 - ext4 /dev/sda rw
270 256 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test2
/sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c ro,relatime shared:1 - ext4
/dev/sda rw
271 263 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-5fe42974b06f42338854bbaad30b2b0c/test3 /test1/zzz
ro,relatime shared:1 - ext4 /dev/sda rw
```

Azure Container Instances 취약점 분석

▶ 로컬은 되는데… 리모트는?

- 자동화 코드를 만들어서 이틀 동안 돌렸으나 실패

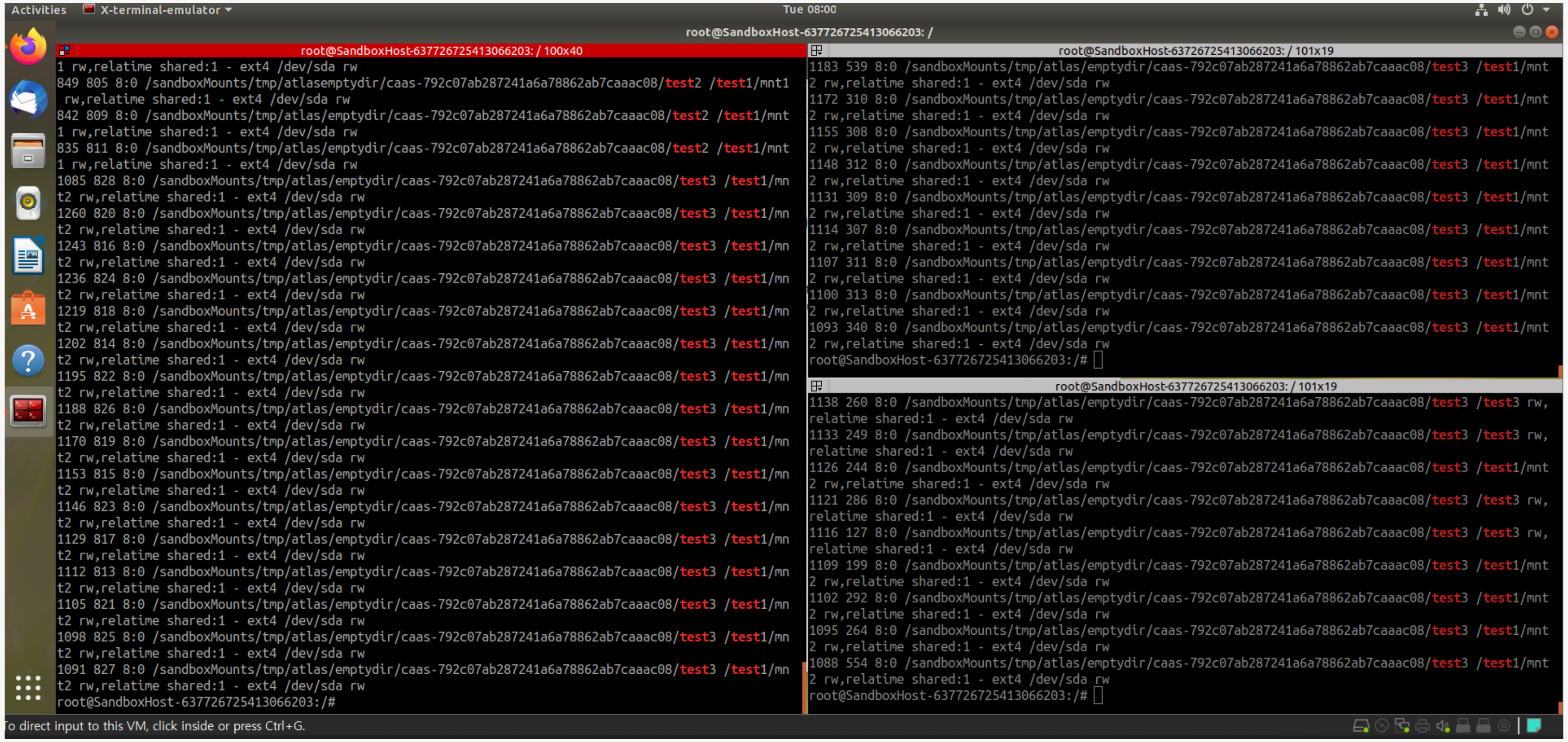


```
[*] 3475
[*] 3475
NOW DELETE CONTAINERS!
20.200.234.200 - - [24/Nov/2021 03:04:38] "GET /woo?bob=333 HTTP/1.1" 200 -
[*] 3476
[*] 3476
NOW DELETE CONTAINERS!
20.200.234.158 - - [24/Nov/2021 03:05:18] "GET /woo?bob=333 HTTP/1.1" 200 -
[*] 3477
[*] 3477
NOW DELETE CONTAINERS!
20.200.239.241 - - [24/Nov/2021 03:05:55] "GET /woo?bob=333 HTTP/1.1" 200 -
[*] 3478
[*] 3478
NOW DELETE CONTAINERS!
20.196.234.72 - - [24/Nov/2021 03:06:35] "GET /woo?bob=333 HTTP/1.1" 200 -
[*] 3479
[*] 3479
NOW DELETE CONTAINERS!
20.200.234.200 - - [24/Nov/2021 03:07:58] "GET /woo?bob=333 HTTP/1.1" 200 -
[*] 3480
[*] 3480
NOW DELETE CONTAINERS!
20.200.239.241 - - [24/Nov/2021 03:08:52] "GET /woo?bob=333 HTTP/1.1" 200 -
[*] 3481
[*] 3481
NOW DELETE CONTAINERS!
```

Azure Container Instances 취약점 분석

▶ 취약점 익스 실패 원인 분석

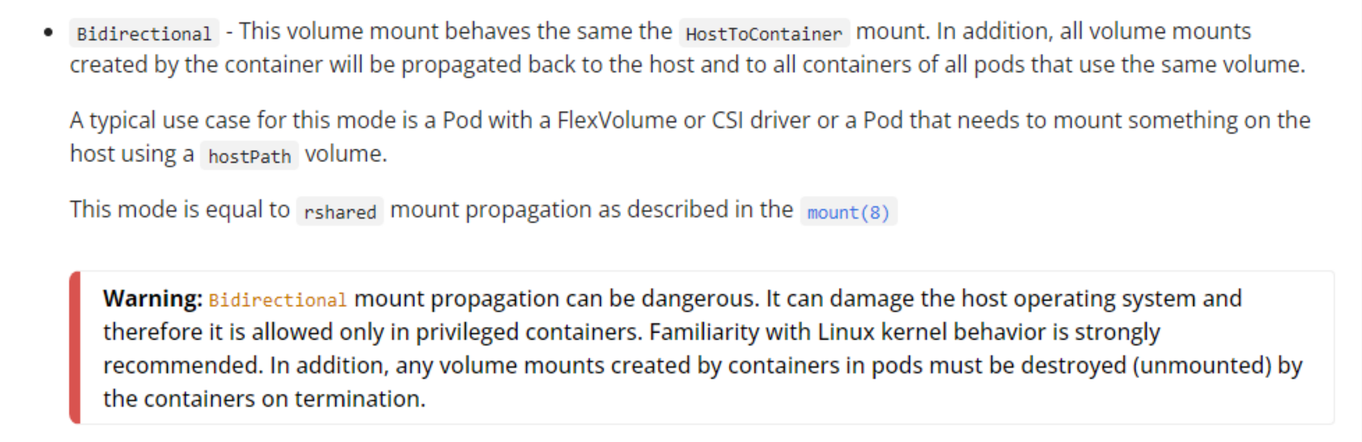
- ACI는 Azure Kubernetes Service 위에서 돌아감
- 할당된 컨테이너에 접속해 마운트 정보를 확인해 보니 옵션이 좀 다름


Activities
X-terminal-emulator
Tue 08:00
root@SandboxHost-637726725413066203: /
root@SandboxHost-637726725413066203: / 100x40
root@SandboxHost-63726725413066203: / 101x19
root@SandboxHost-63726725413066203: / 101x19
To direct input to this VM, click inside or press Ctrl+G.

```
1 rw,relatime shared:1 - ext4 /dev/sda rw
849 805 8:0 /sandboxMounts/tmp/atlasemptydir/caas-792c07ab287241a6a78862ab7caaac08/test2 /test1/mnt1
 rw,relatime shared:1 - ext4 /dev/sda rw
842 809 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test2 /test1/mnt
1 rw,relatime shared:1 - ext4 /dev/sda rw
835 811 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test2 /test1/mnt
1 rw,relatime shared:1 - ext4 /dev/sda rw
1085 828 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1260 820 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1243 816 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1236 824 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1219 818 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1202 814 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1195 822 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1188 826 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1170 819 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1153 815 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1146 823 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1129 817 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1112 813 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1105 821 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1098 825 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
1091 827 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mn
t2 rw,relatime shared:1 - ext4 /dev/sda rw
root@SandboxHost-637726725413066203:/#
```

```
1183 539 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1172 310 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1155 308 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1148 312 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1131 309 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1114 307 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1107 311 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1100 313 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1093 340 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
root@SandboxHost-637726725413066203:/#
```

```
1138 260 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test3 rw,
relatime shared:1 - ext4 /dev/sda rw
1133 249 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test3 rw,
relatime shared:1 - ext4 /dev/sda rw
1126 244 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1121 286 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test3 rw,
relatime shared:1 - ext4 /dev/sda rw
1116 127 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test3 rw,
relatime shared:1 - ext4 /dev/sda rw
1109 199 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1102 292 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1095 264 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
1088 554 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-792c07ab287241a6a78862ab7caaac08/test3 /test1/mnt
2 rw,relatime shared:1 - ext4 /dev/sda rw
root@SandboxHost-637726725413066203:/#
```



- `Bidirectional` - This volume mount behaves the same the `HostToContainer` mount. In addition, all volume mounts created by the container will be propagated back to the host and to all containers of all pods that use the same volume.

  A typical use case for this mode is a Pod with a FlexVolume or CSI driver or a Pod that needs to mount something on the host using a `hostPath` volume.

  This mode is equal to `rshared` mount propagation as described in the `mount(8)`

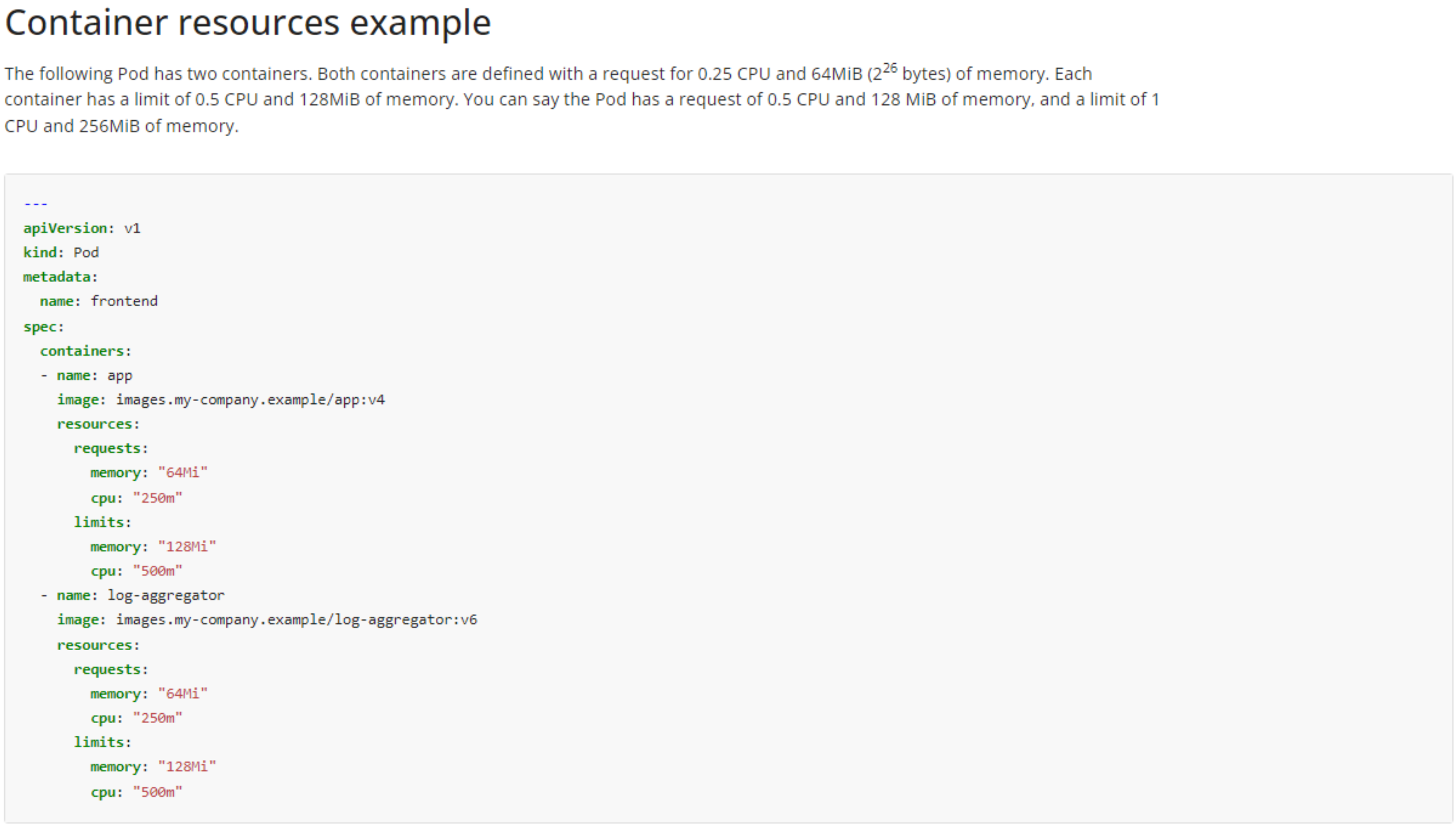
> **Warning:** `Bidirectional` mount propagation can be dangerous. It can damage the host operating system and therefore it is allowed only in privileged containers. Familiarity with Linux kernel behavior is strongly recommended. In addition, any volume mounts created by containers in pods must be destroyed (unmounted) by the containers on termination.

1) "https://kubernetes.io/docs/concepts/storage/volumes/#mount-propagation", kubernetes

Azure Container Instances 취약점 분석

Azure Container Instances Escape 취약점 분석 배경: 컨테이너 수에 따라 달라지는 환경

차라리 컨테이너 개수를 훨씬 늘려서
익스 확률을 높여보자!



Container resources example

The following Pod has two containers. Both containers are defined with a request for 0.25 CPU and 64MiB (2^{26} bytes) of memory. Each container has a limit of 0.5 CPU and 128MiB of memory. You can say the Pod has a request of 0.5 CPU and 128 MiB of memory, and a limit of 1 CPU and 256MiB of memory.

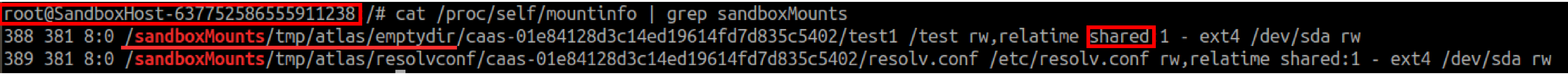
```
---
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image: images.my-company.example/app:v4
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
  - name: log-aggregator
    image: images.my-company.example/log-aggregator:v6
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

1) “https://m.cafe.daum.net/dotax/Elgq/3634016?svc=topRank”

2) “https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/”, Kubernetes

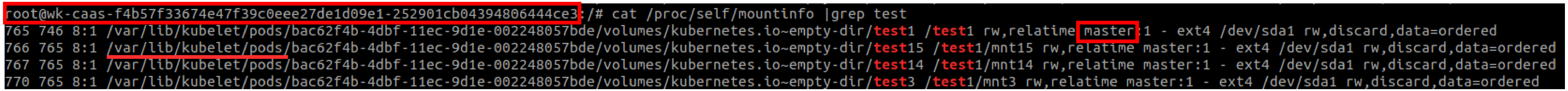
Azure Container Instances 취약점 분석

Azure Container Instances Escape 취약점 분석 배경: 컨테이너 수에 따라 달라지는 환경



```
root@SandboxHost-637752586555911238:/# cat /proc/self/mountinfo | grep sandboxMounts
388 381 8:0 /sandboxMounts/tmp/atlas/emptydir/caas-01e84128d3c14ed19614fd7d835c5402/test1 /test rw,relatime shared:1 - ext4 /dev/sda rw
389 381 8:0 /sandboxMounts/tmp/atlas/resolvconf/caas-01e84128d3c14ed19614fd7d835c5402/resolv.conf /etc/resolv.conf rw,relatime shared:1 - ext4 /dev/sda rw
```

- 4개 이하의 컨테이너를 할당한 경우
  - hostname: SandboxHost
  - volume 위치: /sandboxMounts/tmp/atlas/emptydir/…
  - mount 옵션: shared



```
root@wk-caas-f4b57f33674e47f39c0eee27de1d09e1-252901cb04394806444ce3:/# cat /proc/self/mountinfo |grep test
765 746 8:1 /var/lib/kubelet/pods/bac62f4b-4dbf-11ec-9d1e-002248057bde/volumes/kubernetes.io~empty-dir/test1 /test1 rw,relatime master:1 - ext4 /dev/sda1 rw,discard,data=ordered
766 765 8:1 /var/lib/kubelet/pods/bac62f4b-4dbf-11ec-9d1e-002248057bde/volumes/kubernetes.io~empty-dir/test15 /test1/mnt15 rw,relatime master:1 - ext4 /dev/sda1 rw,discard,data=ordered
767 765 8:1 /var/lib/kubelet/pods/bac62f4b-4dbf-11ec-9d1e-002248057bde/volumes/kubernetes.io~empty-dir/test14 /test1/mnt14 rw,relatime master:1 - ext4 /dev/sda1 rw,discard,data=ordered
770 765 8:1 /var/lib/kubelet/pods/bac62f4b-4dbf-11ec-9d1e-002248057bde/volumes/kubernetes.io~empty-dir/test3 /test1/mnt3 rw,relatime master:1 - ext4 /dev/sda1 rw,discard,data=ordered
```

- 5개 이상의 컨테이너를 할당한 경우
  - hostname: wk-caas
  - volume 위치: /var/lib/kubelet/pods/…
  - mount 옵션: master

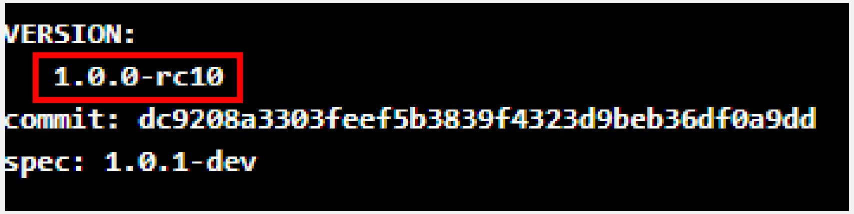
Azure Container Instances 취약점 분석

Azure Container Instances Escape 취약점 발생 환경: 컨테이너 수에 따라 달라지는 runC 버전

4개 이하의 컨테이너

- Azure Container Instances의 runC version

: 1.0.0-rc10 (2020)



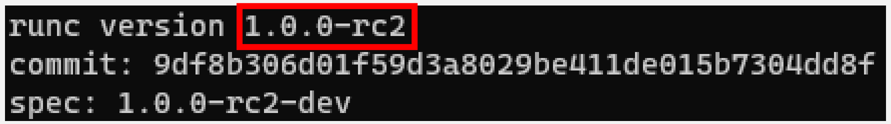
```
VERSION:
  1.0.0-rc10
commit: dc9208a3303feef5b3839f4323d9beb36df0a9dd
spec: 1.0.1-dev
```

5개 이상의 컨테이너

- Azure Container Instances의 runC version

: 1.0.0-rc2 (2016)

⇒ 훨씬 취약한 컨테이너 환경 구성



```
runc version 1.0.0-rc2
commit: 9df8b306d01f59d3a8029be411de015b7304dd8f
spec: 1.0.0-rc2-dev
```

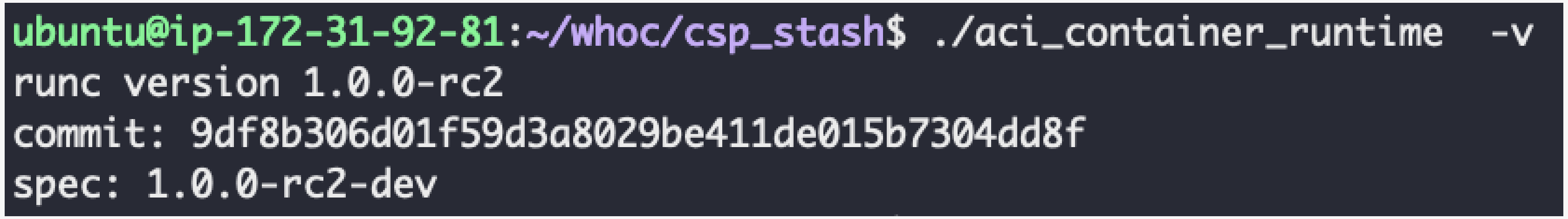
Azure Container Instances 취약점 분석

▶ AzureEscape

- 2021년 9월 패치된 Azure Container Instances 취약점
- CVE-2019-5736을 활용한 컨테이너 이스케이프

1) "https://blog.aquasec.com/azurescape-azure-container-instances", aqua

2) https://unit42.paloaltonetworks.com/azure-container-instances/, unit42

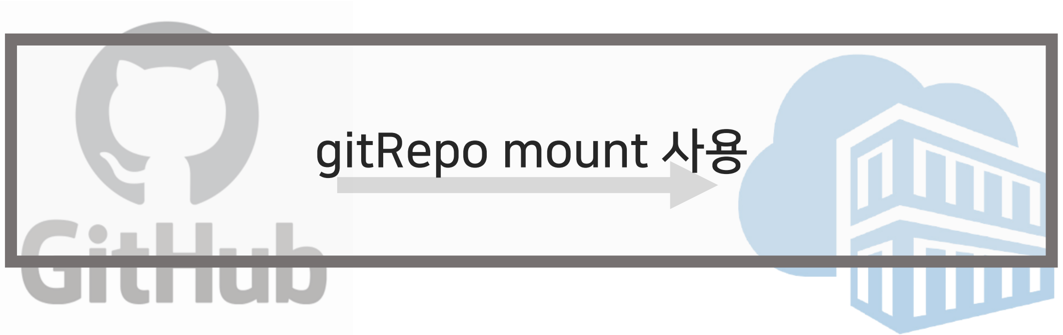


```
ubuntu@ip-172-31-92-81:~/whoc/csp_stash$ ./aci_container_runtime  -v
runc version 1.0.0-rc2
commit: 9df8b306d01f59d3a8029be411de015b7304dd8f
spec: 1.0.0-rc2-dev
```

Azure Container Instances 취약점 분석

Azure Container Instances Escape 취약점 발생 조건: 5개 이상의 container, gitRepo mount

5개 이상의 Container 생성



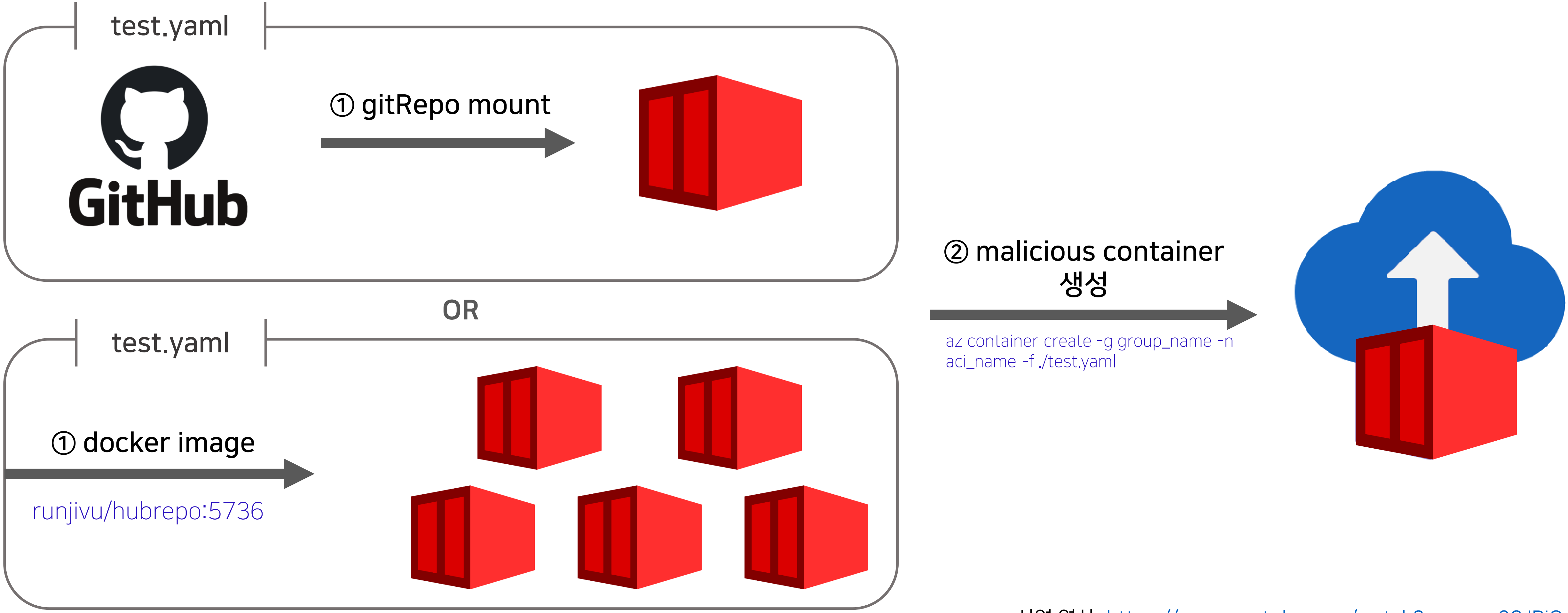
gitRepo mount 사용

링크: https://github.com/GroomPang/Research/blob/main/ACI_POC/poc.sh

링크: https://github.com/GroomPang/Research/blob/main/ACI_POC/test.yaml

Azure Container Instances – Container Escape

Azure Container Instances의 Container Escape 과정 (1) : create malicious container



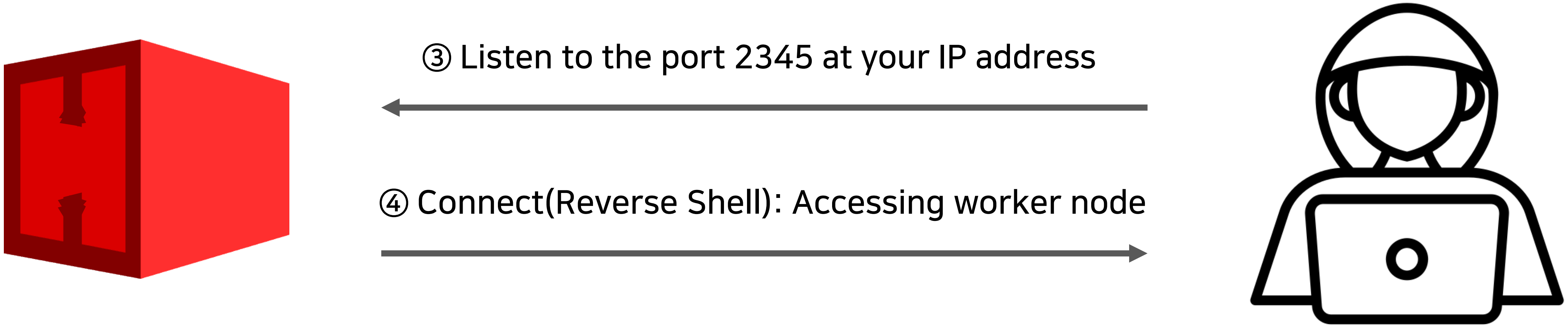
runjivu/hubrepo:5736

```
az container create -g group_name -n aci_name -f ./test.yaml
```

시연 영상: https://www.youtube.com/watch?v=mgoc90JPjOw

Azure Container Instances – Container Escape

Azure Container Instances의 Container Escape 과정 (2): container escape



③ Listen to the port 2345 at your IP address

④ Connect(Reverse Shell): Accessing worker node

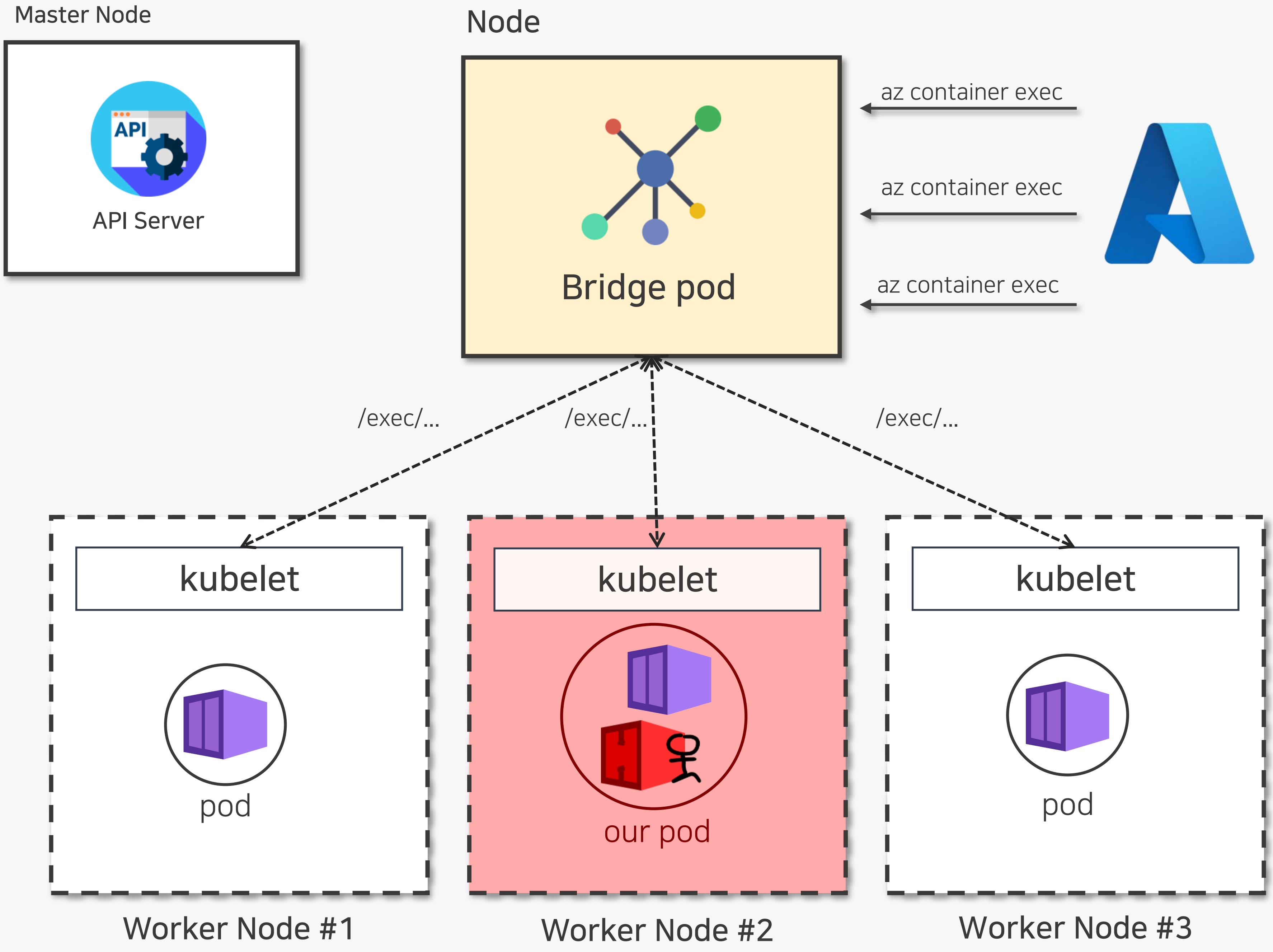
시연 영상: https://www.youtube.com/watch?v=mgoc90JPjOw

Azure Container Instances 취약점 분석

▶ ACI 워커노드 장악

▷ malicious container를 통해 container escape

▷ 이를 통한, worker node 장악



Azure Container Instances 취약점 분석 - 결론

Azure Container Instances 취약점 분석 내용 정리

- **취약한 환경**

  특정 상황에서 구버전의 runc 사용
  대부분의 Kubernetes Component가 hyperkube 컨테이너를 바탕으로 동작

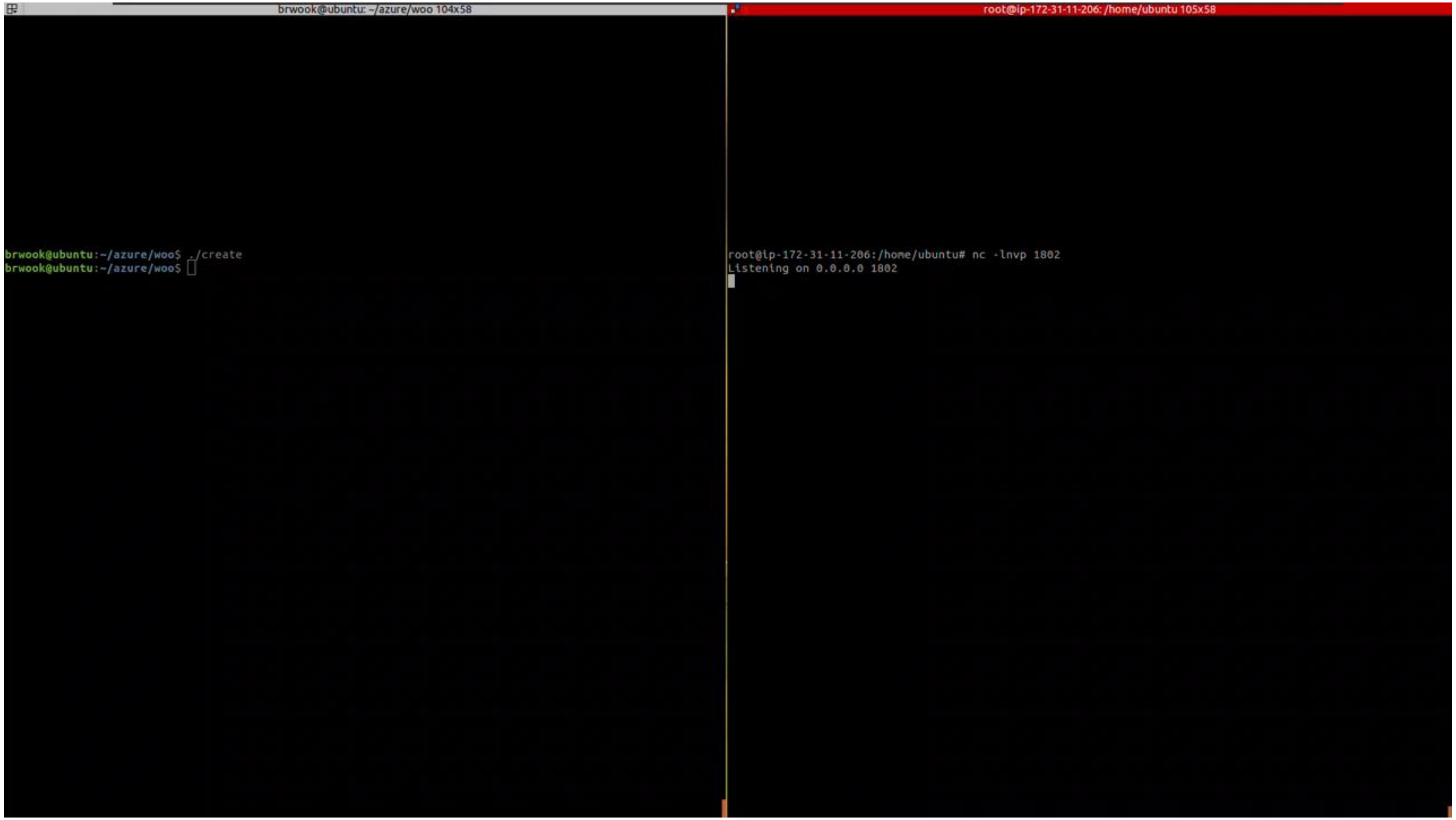
- **취약점 분석 내용**

  - Container Escape
  ⇒ Azure Container Instances Escape 취약점 제보!

  - 노드 환경 정보 파악
  - kubelet 변조를 통한 정보 유출

데모 영상

Azure Container Instances Container Escape



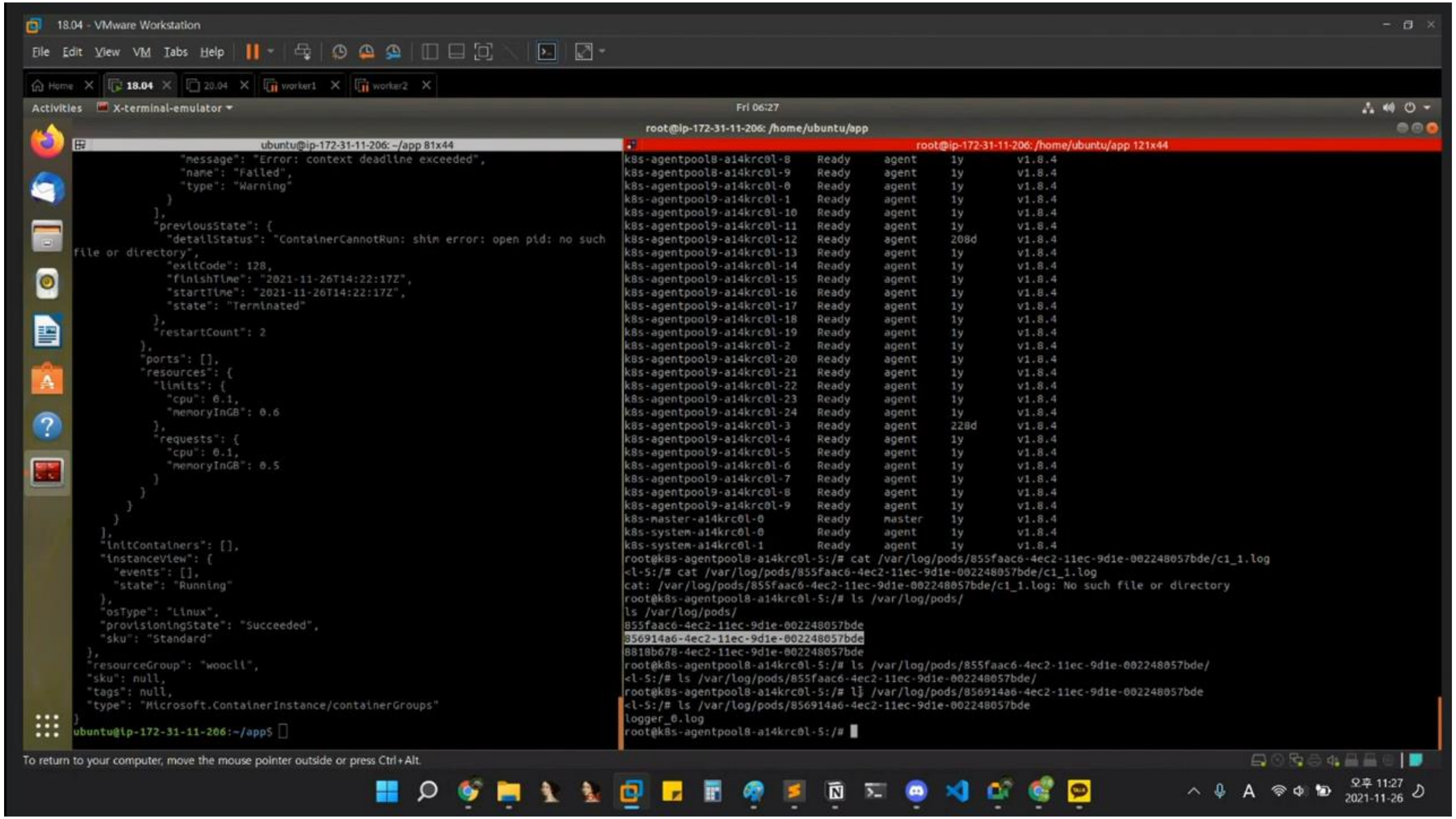
brwook@ubuntu: ~/azure/woo 104x58
root@ip-172-31-11-206: /home/ubuntu 105x58

```
brwook@ubuntu:~/azure/woo$ ./create
brwook@ubuntu:~/azure/woo$ 
```

```
root@ip-172-31-11-206:/home/ubuntu# nc -lnvp 1802
Listening on 0.0.0.0 1802
```

데모 영상

Azure Container Instances Log Modification


18.04 - VMware Workstation
File Edit View VM Tabs Help
Home
18.04
20.04
worker1
worker2
Activities
X-terminal-emulator
Fri 06:27
root@ip-172-31-11-206: /home/ubuntu/app
ubuntu@ip-172-31-11-206: ~/app 81x44
root@ip-172-31-11-206: /home/ubuntu/app 121x44
오후 11:27
2021-11-26

```
              "message": "Error: context deadline exceeded",
              "name": "Failed",
              "type": "Warning"
            }
          ],
          "previousState": {
            "detailStatus": "ContainerCannotRun: shim error: open pid: no such
file or directory",
            "exitCode": 128,
            "finishTime": "2021-11-26T14:22:17Z",
            "startTime": "2021-11-26T14:22:17Z",
            "state": "Terminated"
          },
          "restartCount": 2
        },
        "ports": [],
        "resources": {
          "limits": {
            "cpu": 0.1,
            "memoryInGB": 0.6
          },
          "requests": {
            "cpu": 0.1,
            "memoryInGB": 0.5
          }
        }
      }
    }
  ],
  "initContainers": [],
  "instanceView": {
    "events": [],
    "state": "Running"
  },
  "osType": "Linux",
  "provisioningState": "Succeeded",
  "sku": "Standard"
},
"resourceGroup": "woocli",
"sku": null,
"tags": null,
"type": "Microsoft.ContainerInstance/containerGroups"
}
ubuntu@ip-172-31-11-206:~/app$ 
```

```
k8s-agentpool8-a14krc0l-8    Ready    agent    1y      v1.8.4
k8s-agentpool8-a14krc0l-9    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-0    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-1    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-10   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-11   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-12   Ready    agent    208d    v1.8.4
k8s-agentpool9-a14krc0l-13   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-14   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-15   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-16   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-17   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-18   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-19   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-2    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-20   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-21   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-22   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-23   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-24   Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-3    Ready    agent    228d    v1.8.4
k8s-agentpool9-a14krc0l-4    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-5    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-6    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-7    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-8    Ready    agent    1y      v1.8.4
k8s-agentpool9-a14krc0l-9    Ready    agent    1y      v1.8.4
k8s-master-a14krc0l-0        Ready    master   1y      v1.8.4
k8s-system-a14krc0l-0        Ready    agent    1y      v1.8.4
k8s-system-a14krc0l-1        Ready    agent    1y      v1.8.4
root@k8s-agentpool8-a14krc0l-5:/# cat /var/log/pods/855faac6-4ec2-11ec-9d1e-002248057bde/c1_1.log
<l-5:/# cat /var/log/pods/855faac6-4ec2-11ec-9d1e-002248057bde/c1_1.log
cat: /var/log/pods/855faac6-4ec2-11ec-9d1e-002248057bde/c1_1.log: No such file or directory
root@k8s-agentpool8-a14krc0l-5:/# ls /var/log/pods/
ls /var/log/pods/
855faac6-4ec2-11ec-9d1e-002248057bde
856914a6-4ec2-11ec-9d1e-002248057bde
8818b678-4ec2-11ec-9d1e-002248057bde
root@k8s-agentpool8-a14krc0l-5:/# ls /var/log/pods/855faac6-4ec2-11ec-9d1e-002248057bde/
<l-5:/# ls /var/log/pods/855faac6-4ec2-11ec-9d1e-002248057bde/
root@k8s-agentpool8-a14krc0l-5:/# ls /var/log/pods/856914a6-4ec2-11ec-9d1e-002248057bde
<l-5:/# ls /var/log/pods/856914a6-4ec2-11ec-9d1e-002248057bde
logger_0.log
root@k8s-agentpool8-a14krc0l-5:/# 
```

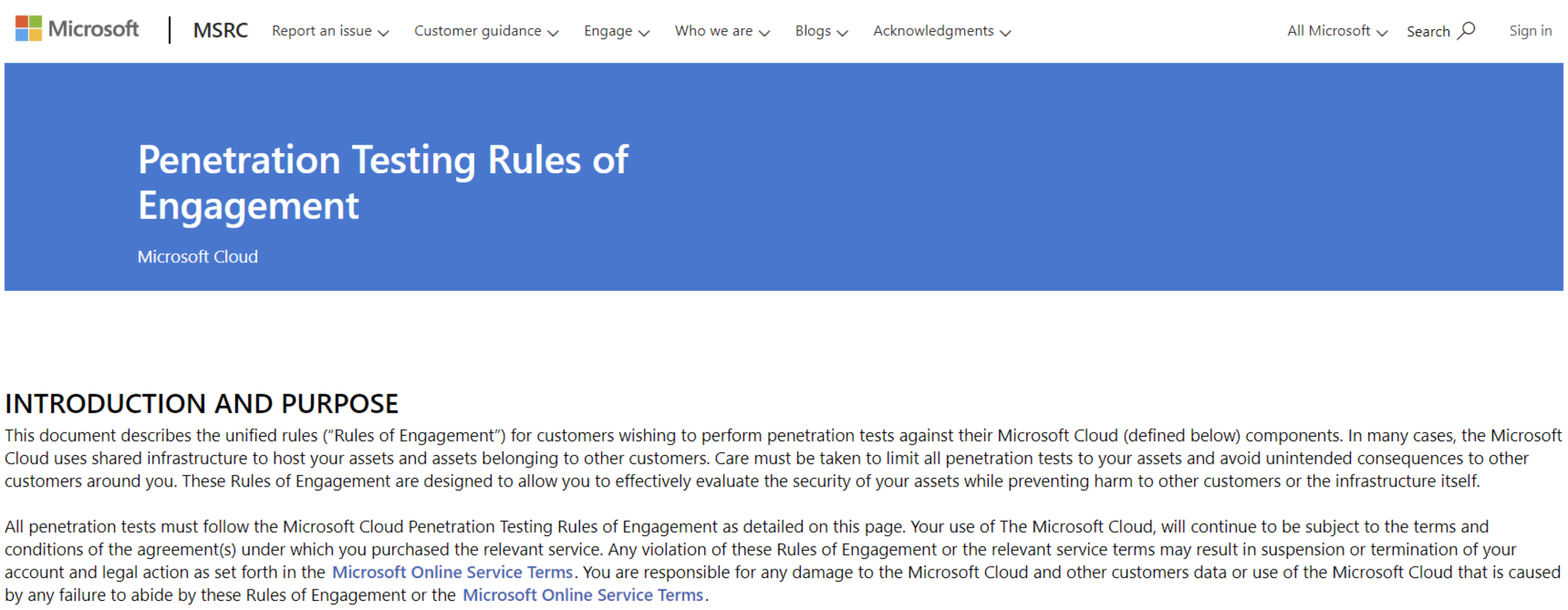
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Ⅲ 프로젝트 결론

- 버그바운티 결과

버그바운티 결과

▶ MS 버그바운티 안내



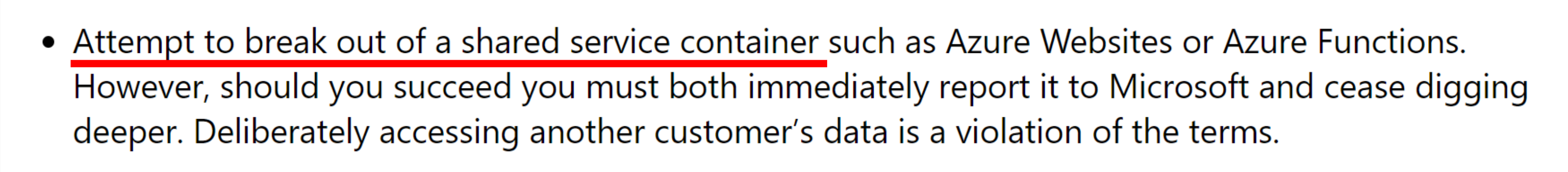
Penetration Testing Rules of Engagement

Microsoft Cloud

INTRODUCTION AND PURPOSE

This document describes the unified rules ("Rules of Engagement") for customers wishing to perform penetration tests against their Microsoft Cloud (defined below) components. In many cases, the Microsoft Cloud uses shared infrastructure to host your assets and assets belonging to other customers. Care must be taken to limit all penetration tests to your assets and avoid unintended consequences to other customers around you. These Rules of Engagement are designed to allow you to effectively evaluate the security of your assets while preventing harm to other customers or the infrastructure itself.

All penetration tests must follow the Microsoft Cloud Penetration Testing Rules of Engagement as detailed on this page. Your use of The Microsoft Cloud, will continue to be subject to the terms and conditions of the agreement(s) under which you purchased the relevant service. Any violation of these Rules of Engagement or the relevant service terms may result in suspension or termination of your account and legal action as set forth in the Microsoft Online Service Terms. You are responsible for any damage to the Microsoft Cloud and other customers data or use of the Microsoft Cloud that is caused by any failure to abide by these Rules of Engagement or the Microsoft Online Service Terms.

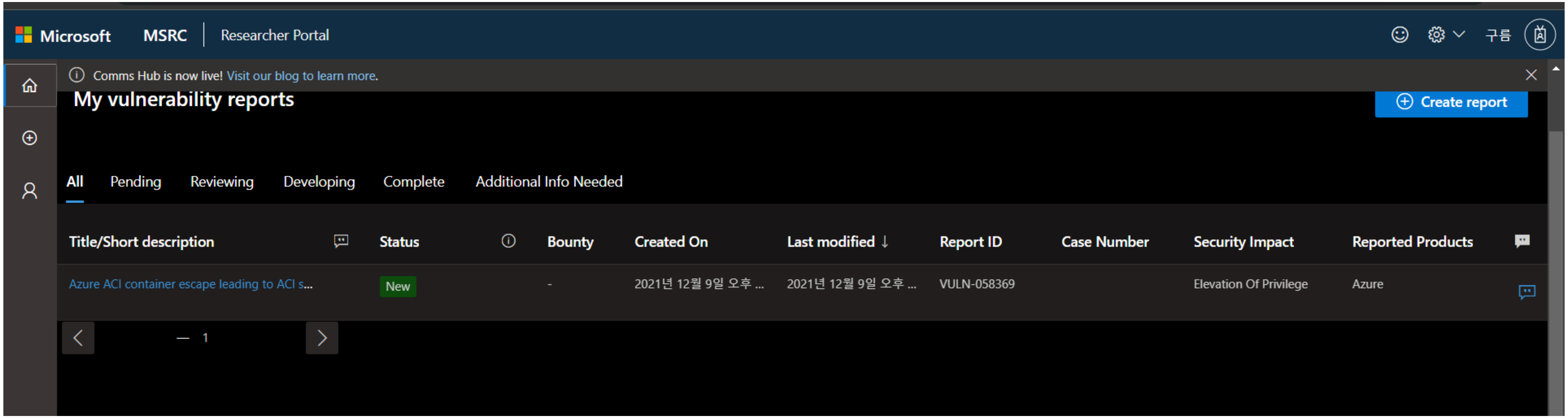


- Attempt to break out of a shared service container such as Azure Websites or Azure Functions. However, should you succeed you must both immediately report it to Microsoft and cease digging deeper. Deliberately accessing another customer's data is a violation of the terms.

1) "https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement", MSRC

버그바운티 결과

Microsoft Azure Container Instances Escape 취약점 제보: MSRC



Comms Hub is now live! Visit our blog to learn more.

My vulnerability reports

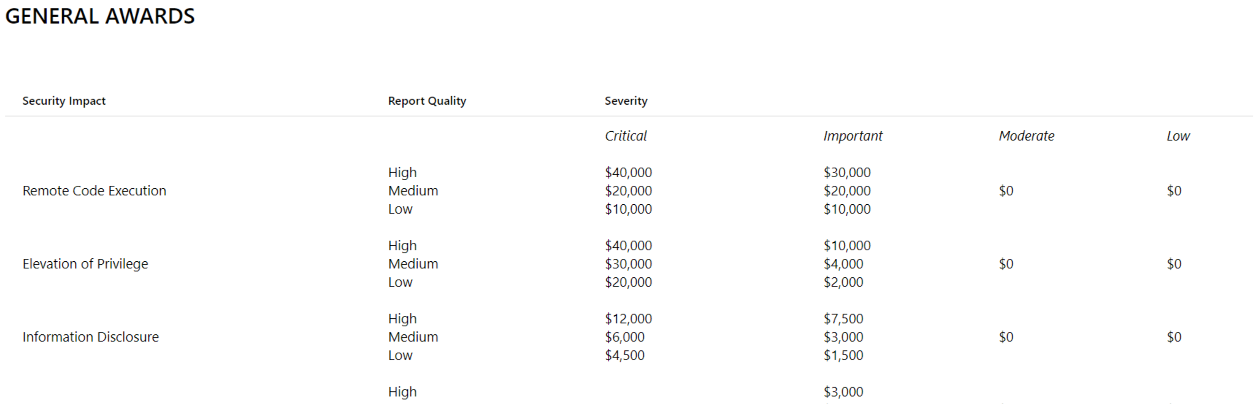
All | Pending | Reviewing | Developing | Complete | Additional Info Needed

Title/Short description	Status	Bounty	Created On	Last modified ↓	Report ID	Case Number	Security Impact	Reported Products
Azure ACI container escape leading to ACI s...	New	-	2021년 12월 9일 오후 ...	2021년 12월 9일 오후 ...	VULN-058369		Elevation Of Privilege	Azure

— 1

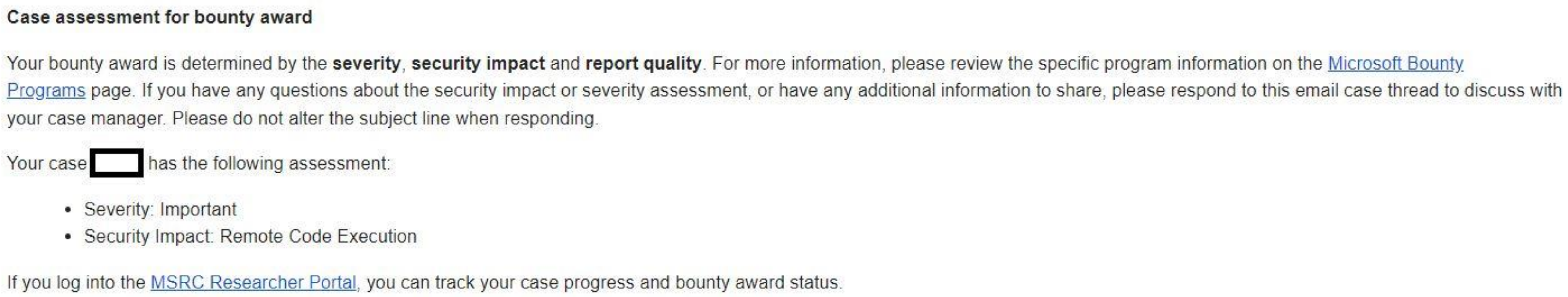
버그바운티 결과

Microsoft Azure Container Instances Escape 취약점 제보: MSRC



GENERAL AWARDS

Security Impact	Report Quality	Severity			
		Critical	*Important*	*Moderate*	*Low*
Remote Code Execution	High	$40,000	$30,000	$0	$0
	Medium	$20,000	$20,000		
	Low	$10,000	$10,000		
Elevation of Privilege	High	$40,000	$10,000	$0	$0
	Medium	$30,000	$4,000		
	Low	$20,000	$2,000		
Information Disclosure	High	$12,000	$7,500	$0	$0
	Medium	$6,000	$3,000		
	Low	$4,500	$1,500		
	High		$3,000		



Case assessment for bounty award

Your bounty award is determined by the **severity**, **security impact** and **report quality**. For more information, please review the specific program information on the Microsoft Bounty Programs page. If you have any questions about the security impact or severity assessment, or have any additional information to share, please respond to this email case thread to discuss with your case manager. Please do not alter the subject line when responding.

Your case [redacted] has the following assessment:

- Severity: Important
- Security Impact: Remote Code Execution

If you log into the MSRC Researcher Portal, you can track your case progress and bounty award status.

1) “https://www.microsoft.com/en-us/msrc/bounty-microsoft-azure”, MSRC

감사합니다!

2023. 07. 03.

Team. 구름빵