



# 500

## Oracle 퍼블릭 클라우드 워크샵

### Oracle Container Cloud

2017 년 04 월 12 일

## 소개

이 튜토리얼에서는 Oracle Container Cloud Service 를 사용하여 간단한 애플리케이션을 배치하는 방법을 설명합니다

배경:

- GitHub 의 프로젝트 소스 코드 호스트
- 소스 파일이 GitHub 에 있을 때마다 Docker 허브 링크를 GitHub 에 설정하고 새로운 Docker 이미지를 자동으로 빌드합니다.
- Oracle Container Cloud Service 를 사용하여 Docker Hub 의 이미지를 기반으로 새로운 서비스를 생성하십시오.
- 새 서비스를 사용하여 컨테이너 인스턴스를 배포합니다.

## 목표

- ☐ GitHub 의 포크 응용 프로그램
- ☐ 도커 허브 자동 빌드 만들기
- ☐ 수동으로 빌드 트리거
- ☐ Oracle Container Cloud Service 로 새로운 서비스 생성
- ☐ Oracle Container Cloud Service 에 새 서비스로 컨테이너 인스턴스 배포
- ☐ 배포 확인

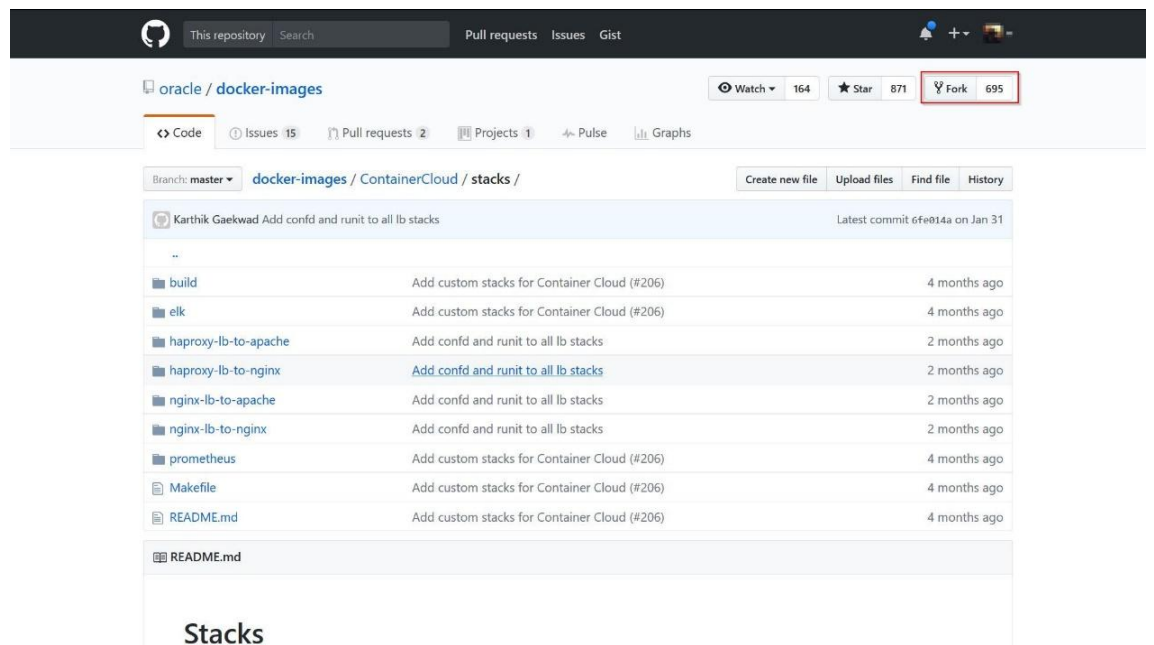
## 필수 아티팩트

- ☐ Oracle Container Cloud Service 를 사용하려면 먼저 조직의 다른 누군가가 Oracle Container Cloud Service 구독을 주문하고 활성화해야 합니다. 자세한 내용은 Oracle 도움말 센터에서 Oracle Container Cloud Service 사용을 참조하십시오. GitHub 계정 : 아직 GitHub 계정이 없다면 <https://github.com/>에 가입하여 가입하십시오. Docker 허브 계정 : Docker 허브 계정이 없다면 <https://hub.docker.com/>을 방문하여 가입하십시오.

## OCCS 에 첫 번째 앱 배포

### GitHub 에서 Application Fork 하기

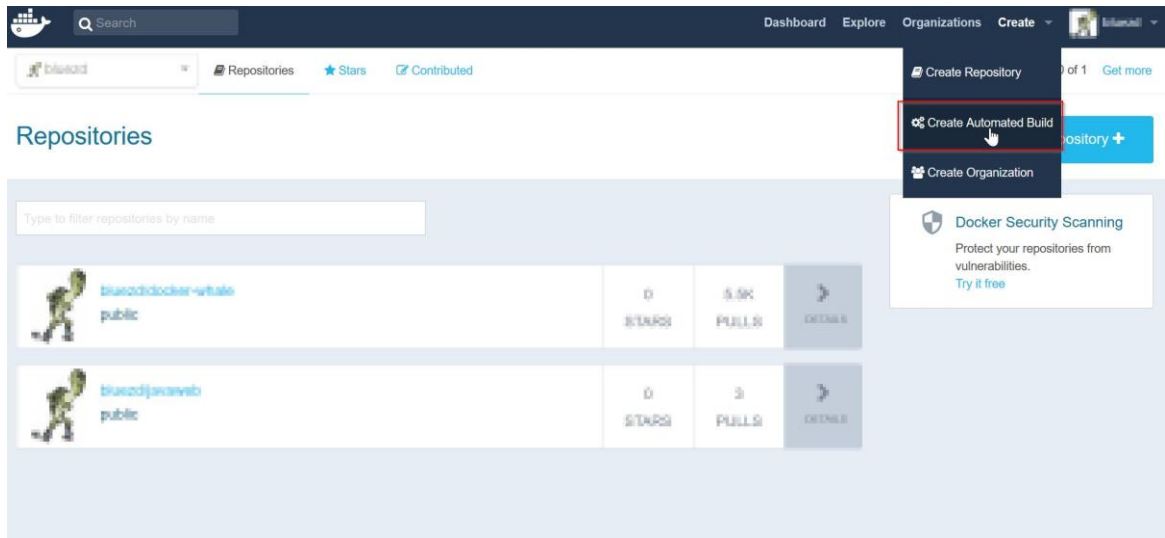
1. 브라우저에서 <https://github.com/login> 에서 GitHub 계정에 로그인하십시오. 계정이 없는 경우 새로 생성하시기 바랍니다.
2. <https://github.com/oracle/docker-images/tree/master/ContainerCloud/images/docker-hello-world> 로 이동하십시오.
3. 저장소 페이지의 상단에있는 메뉴에서 Fork 버튼을 클릭하십시오.



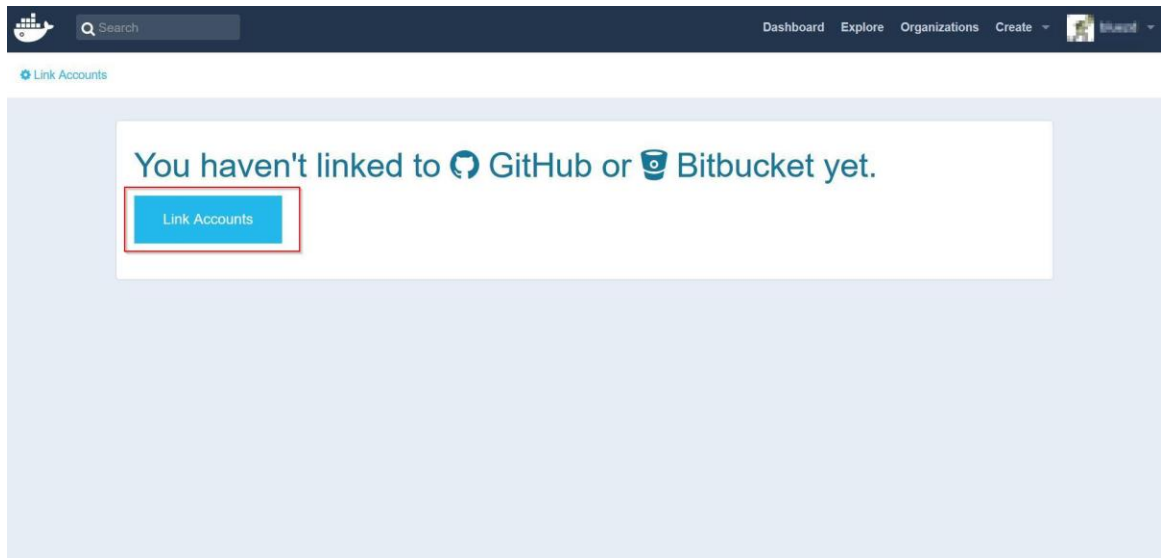
이제 docker-images 저장소의 복사본을 만들었습니다.

### 도커 허브 자동 빌드 만들기

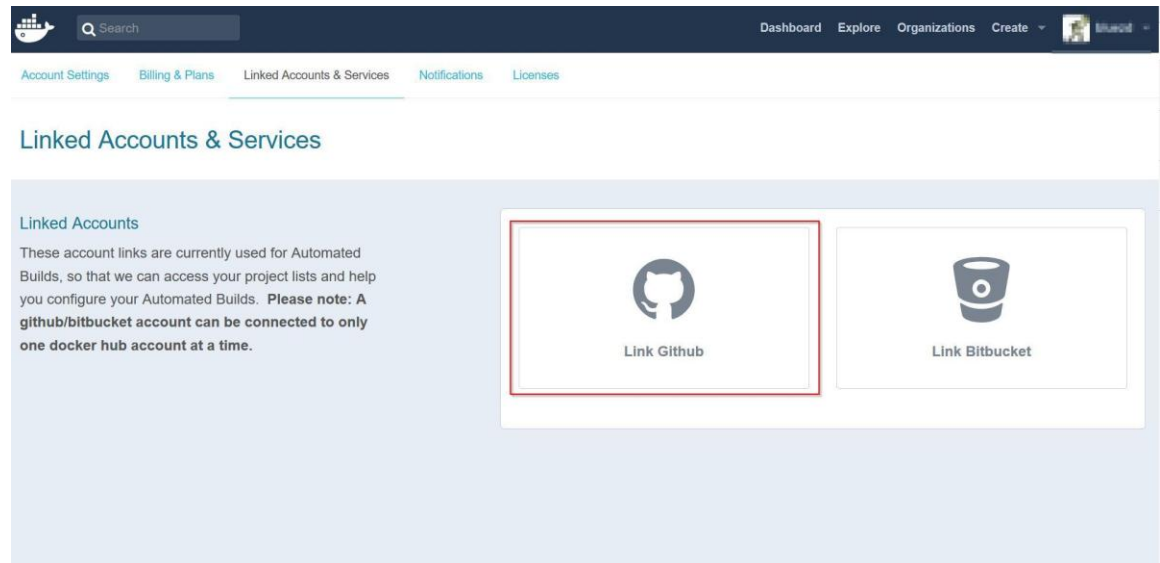
1. 저장소 페이지 맨 위에있는 메뉴에서 Fork 버튼을 클릭하십시오.
2. Docker Hub Create 메뉴에서 "Create Automated Build"를 선택하십시오.



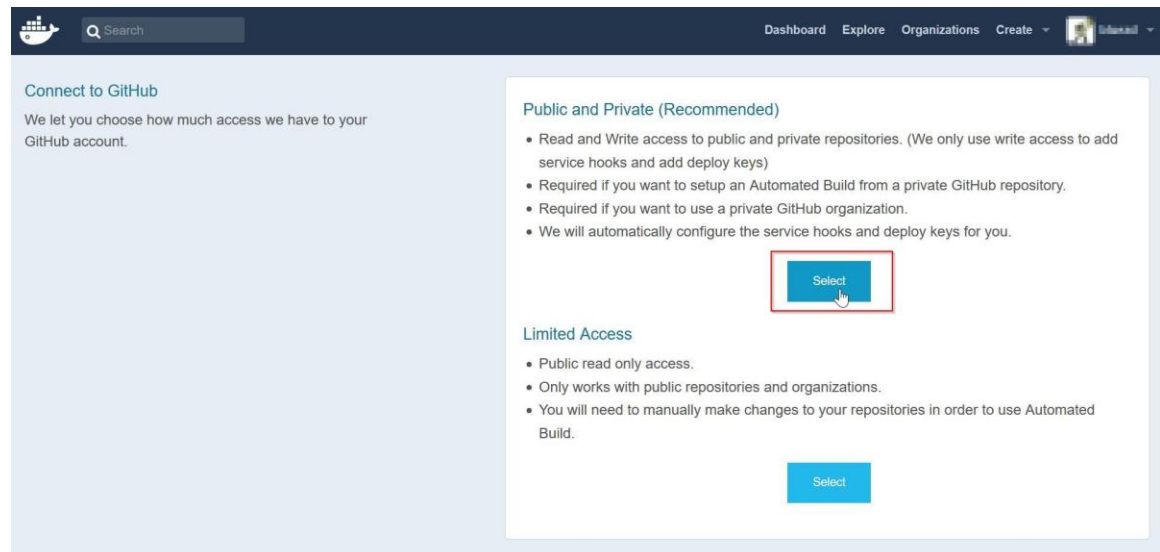
3. "계정 연결"을 클릭하십시오.



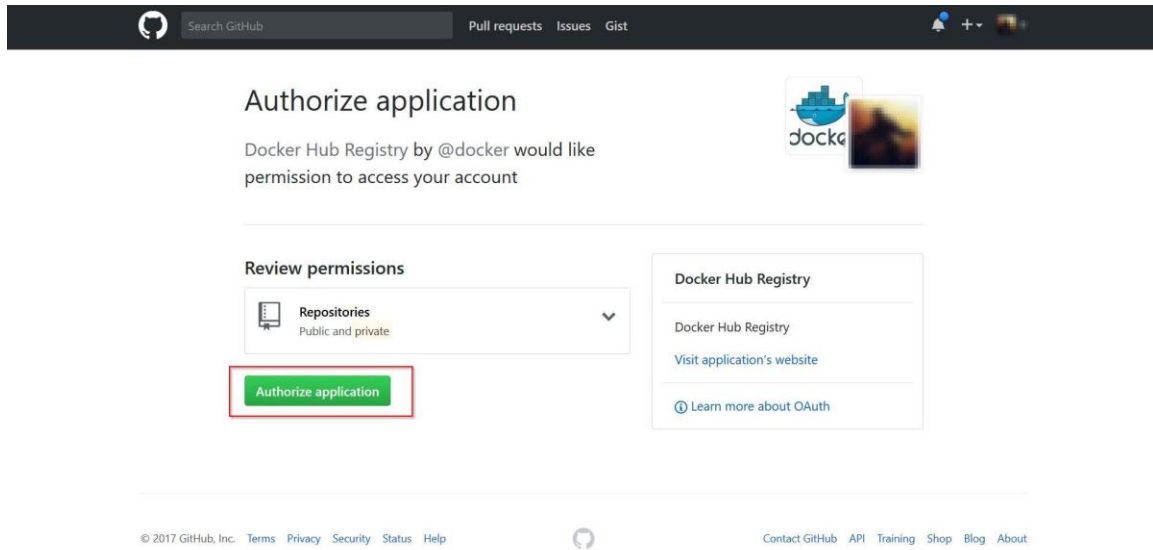
4. "Link GitHub"를 클릭하십시오.



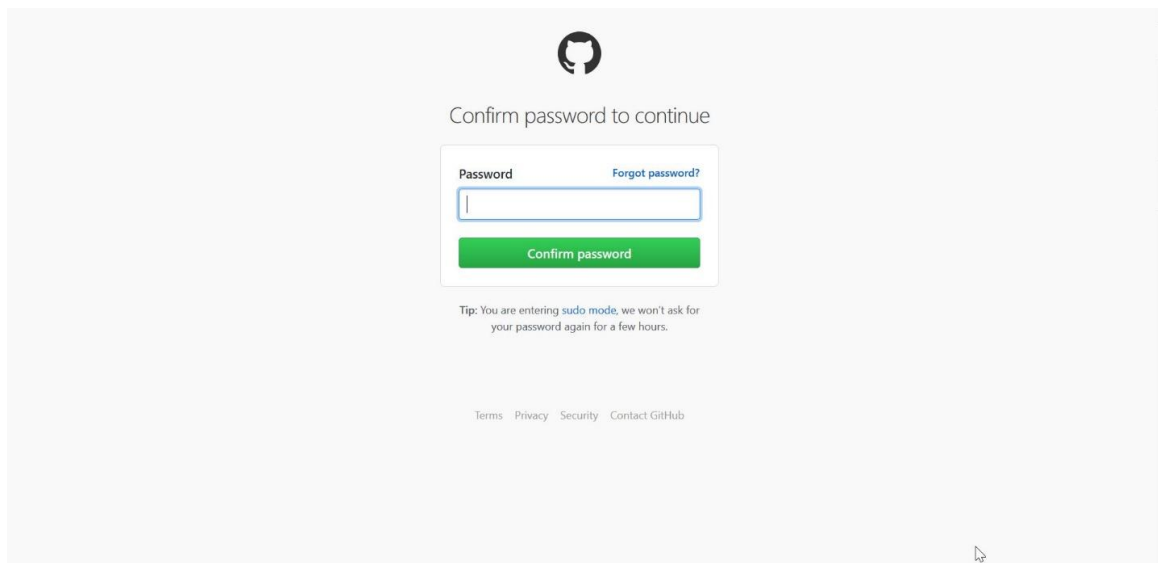
5. “공개 및 비공개”를 선택하고 선택을 클릭하십시오.



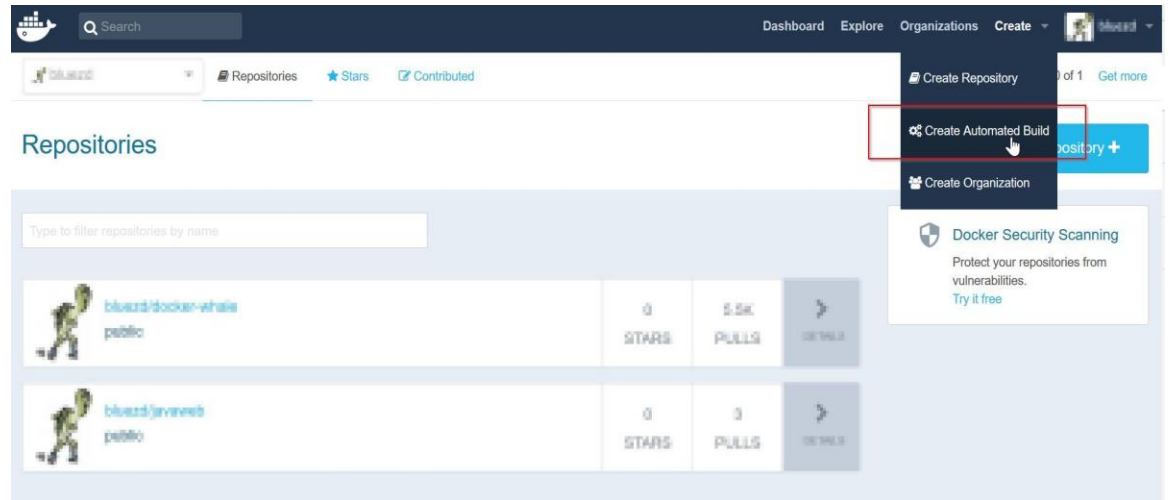
6. “Authorize Application” 클릭하십시오.



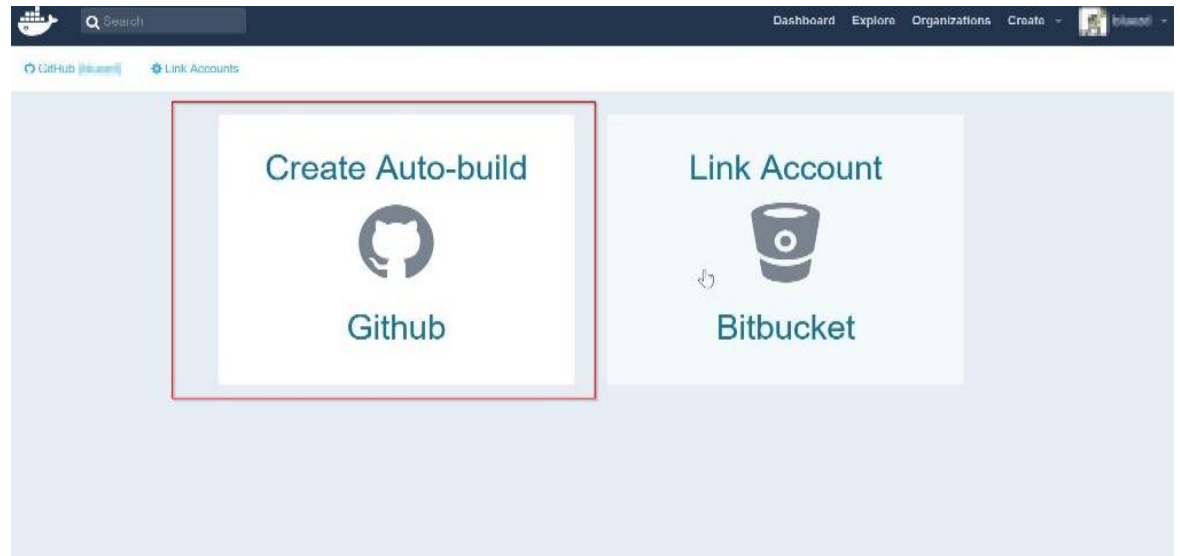
7. 확인 암호를 입력하십시오.



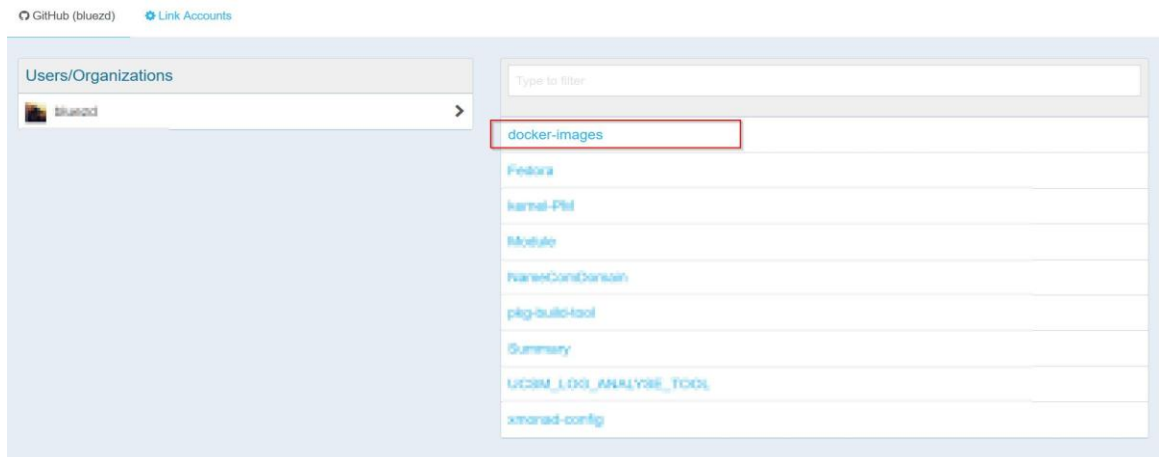
8. Docker Hub 의 "Create"메뉴에서 "Create Automated Build"를 선택하십시오.



9. “자동 빌드 GitHub 만들기”를 클릭하여 GitHub 계정에 자동 빌드를 만들 저장소가 있음을 나타냅니다.



10. 자신의 GitHub 계정으로 분기 한 “docker-images”저장소를 선택하십시오.



11. “자동 빌드 작성”페이지에서 다음을 수행하십시오.

- a. 다음 값을 입력하십시오.
  - i. 이름 : docker-hello-world
  - ii. 간단한 설명 : 간단한 Hello World 예제
- b. 만들기를 클릭하십시오.

자동 빌드가 생성되면 자동 빌드 화면의 “Repo Info”탭이 표시됩니다.

12. “Build Settings”탭으로 이동하십시오.



“Dockerfile Location” 필드의 값을 `/ContainerCloud/images/docker-hello-world/`로 변경하십시오.

Build Settings

☒ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourceref}, which refers to the source branch/tag name. [Show more](#)

Source Repository: [github/docker-images](#)

Type	Name	Dockerfile Location	Docker Tag Name
Branch	master	/ContainerCloud/images/docker	latest
Branch	All branches except master	/	Same as branch

Trigger

Save Changes

“빌드 설정” 탭에서 정의한 빌드 규칙은 이미지 작성 방법 및 시기를 지정합니다. 기본적으로 이미지는 다음과 같습니다.

- 업데이트가 GitHub 에 푸시 될 때마다 자동으로 생성됩니다.
- 마스터 GitHub 브랜치를 사용하여 빌드
- 저장소의 루트 디렉토리에있는 Dockerfile 의 마스터 빌드 규칙을 사용하여 빌드
- Docker Hub 에서 최신 태그를 받음

## 빌드를 수동으로 트리거 만들기

docker-hello-world 앱의 변경 사항이 GitHub 의 docker-images 저장소로 푸시 될 때마다 새로운 Docker 이미지가 빌드되도록 Docker 허브에 자동화 된 빌드를 설정했습니다. 또는 Docker Hub 에서 수동으로 빌드를 트리거 할 수도 있습니다.

1. “Build Settings” 탭에서 마스터 브랜치 옆의 “Trigger”를 클릭하여 기본 빌드 설정으로 이미지를 빌드하십시오.

Repo Info Tags Dockerfile Build Details **Build Settings** Collaborators Webhooks Settings

**Build Settings**

☒ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourcerefs}, which refers to the source branch/tag name. [Show more](#)

Source Repository [blue2/docker-images](#)

Type	Name	Dockerfile Location	Docker Tag Name		
Branch	master	/ContainerCloud/images/docker	latest	+	<b>Trigger</b>
Branch	All branches except master	/	Same as branch	-	

Save Changes

## 2. Docker 허브 저장소의 "Build Details" 페이지로 이동하십시오.

방금 방아쇠 한 방아쇠가 "Build Details" 페이지에 나타납니다. 지정한 빌드 설정으로 인해 이미지는 마스터 GitHub 분기를 사용하여 빌드되며 Docker 허브에 최신 태그가 제공됩니다.

이미지가 성공적으로 빌드되면 빌드 세부 정보 페이지에 latest 라는 태그가 나타납니다.

Repo Info Tags Dockerfile **Build Details** Build Settings Collaborators Webhooks Settings

Status	Actions	Tag	Created	Last Updated
<b>Queued</b>	Cancel	latest	a few seconds ago	a few seconds ago

Source Repository [blue2/docker-images](#)

---

Repo Info Tags Dockerfile **Build Details** Build Settings Collaborators Webhooks Settings

Status	Actions	Tag	Created	Last Updated
<b>Building</b>	Cancel	latest	2 minutes ago	2 minutes ago

Source Repository [blue2/docker-images](#)

---

Repo Info Tags Dockerfile **Build Details** Build Settings Collaborators Webhooks Settings

Status	Actions	Tag	Created	Last Updated
<b>Success</b>		latest	3 minutes ago	2 minutes ago

Source Repository [blue2/docker-images](#)

## 3. "Repo Info" 탭으로 이동하십시오.

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings

Short Description

Simple Hello World Example

Full Description

### NGINX Hello World Image

#### Purpose

This image is used to demonstrate a simple Hello World Docker image using NGINX. It serves up a single HTML page that shows the hostname of the container.

#### Usage

Start the container and publish port 80 to some port on the host.

```
REGISTRY=YOUR_DOCKER_HUB_USERNAME
docker build --rm -t ${REGISTRY}/docker-hello-world .
docker run -d -p 80 ${REGISTRY}/docker-hello-world
docker push ${REGISTRY}/docker-hello-world
```

NOTE: Replace YOUR\_DOCKER\_HUB\_USERNAME above with your own Docker Hub username.

Docker Pull Command

```
docker pull bluebird/docker-hello-wor
```

Owner

Source Repository

bluebird/docker-images

GitHub 저장소의 README.md 파일이 이제 Docker 저장소의 "전체 설명" 필드에 포함되어 있음을 볼 수 있습니다

#### 4. "Dockerfile" 탭으로 이동하십시오.

Repo Info Tags Dockerfile Build Details Build Settings Collaborators Webhooks Settings

Dockerfile

```
FROM nginx:1.10.1-alpine

Add index.html /usr/share/nginx/html/index.html

# Override the nginx start from the base container
COPY start.sh /start.sh
RUN chmod +x /start.sh

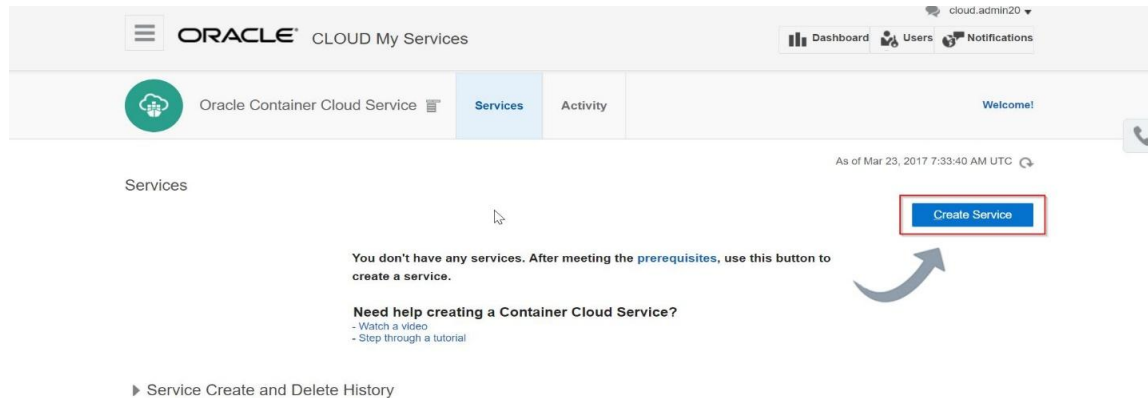
ENTRYPOINT ["/start.sh"]
```

Source Repository

bluebird/docker-images

## Oracle Container Cloud Service 로 새로운 서비스 생성

1. Oracle Container Cloud Service 를 생성하십시오.



ORACLE<sup>®</sup> CLOUD My Services

cloud.admin20 ▾

Dashboard Users Notifications

Oracle Container Cloud Service Services Activity Welcome!

As of Mar 23, 2017 7:33:40 AM UTC ↻

Services

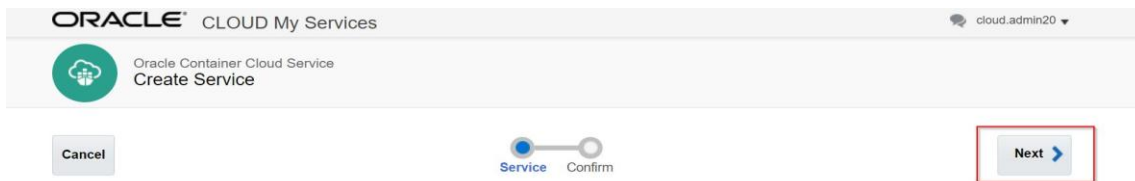
You don't have any services. After meeting the prerequisites, use this button to create a service.

**Create Service**

**Need help creating a Container Cloud Service?**

- Watch a video
- Step through a tutorial

► Service Create and Delete History



ORACLE<sup>®</sup> CLOUD My Services

cloud.admin20 ▾

Oracle Container Cloud Service Create Service

Cancel

Service Confirm

**Next** ➤

### Service

Provide basic service instance information.

#### Details

\* Service Name: OCCS-LAB ⓘ

Service Description: ⓘ

SSH Public Key: publicKey Edit ⓘ

Metering Frequency: Hourly Subscription ⓘ

#### OCCS Configuration

\* Admin Username: admin ⓘ

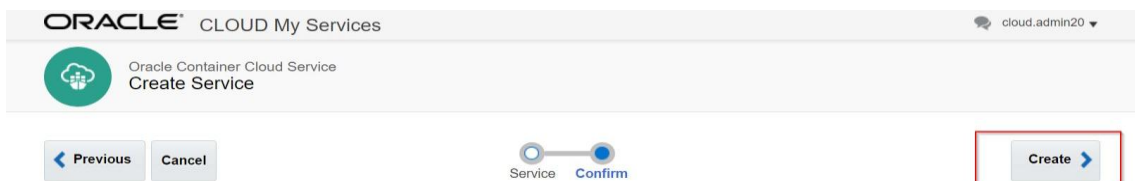
\* Admin Password: ..... ⓘ

\* Confirm Admin Password: ..... ⓘ

\* Worker node Compute Shape: OC3 - 1.0 OCPU, 7.5GB RAM ⓘ

\* Number of worker nodes: 1 ⓘ

\* Worker node data volume size (GB): 30 ⓘ



ORACLE<sup>®</sup> CLOUD My Services

cloud.admin20 ▾

Oracle Container Cloud Service Create Service

◀ Previous Cancel

Service Confirm

**Create** ➤

### Confirmation

Confirm your responses and create this OCCS instance.

#### Service

Service Name: OCCS-LAB

SSH Public Key: publicKey

Service Level: Service with tooling support

Metering Frequency: Hourly Subscription

Software Release: v17.1.5 Oracle Container Cl...

#### OCCS Configuration

Service Edition: Standard Edition

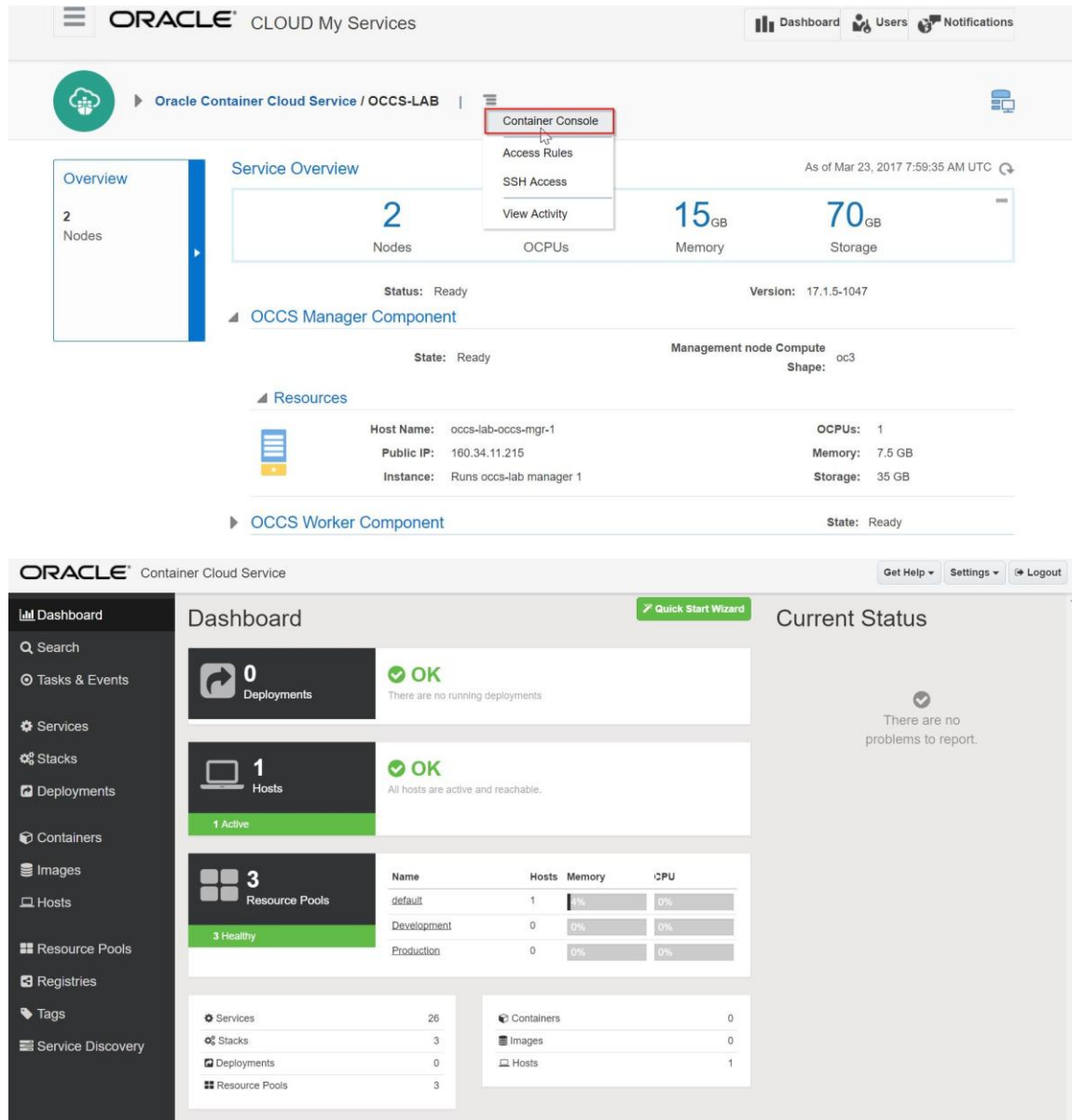
Worker node Compute Shape: OC3 - 1.0 OCPU, 7.5GB RAM

Number of included management nodes: 1

Number of worker nodes: 1

Worker node data volume size (GB): 30

2. Oracle Container Cloud Service 컨테이너 콘솔에 로그인하십시오.



The screenshot shows the Oracle Cloud My Services dashboard. The 'Oracle Container Cloud Service / OCCS-LAB' section is highlighted. A dropdown menu is open, showing options: 'Container Console' (highlighted with a red box), 'Access Rules', 'SSH Access', and 'View Activity'. The dashboard also displays service overview metrics: 2 Nodes, 15 GB Memory, and 70 GB Storage. Below this, the 'OCCS Manager Component' and 'OCCS Worker Component' are listed with their respective states and resources.

**Service Overview**

- 2 Nodes
- 15 GB Memory
- 70 GB Storage
- Status: Ready
- Version: 17.1.5-1047

**OCCS Manager Component**

- State: Ready
- Management node Compute Shape: oc3

**Resources**

Host Name	Public IP	Instance	OCPUs	Memory	Storage
occs-lab-occs-mgr-1	160.34.11.215	Runs occs-lab manager 1	1	7.5 GB	35 GB

**OCCS Worker Component**

- State: Ready

**Oracle Container Cloud Service Dashboard**

- Dashboard
- Quick Start Wizard
- Current Status

**Dashboard Metrics:**

- 0 Deployments (OK)
- 1 Hosts (OK)
- 3 Resource Pools (Healthy)

Name	Hosts	Memory	CPU
default	1	8%	0%
Development	0	0%	0%
Production	0	0%	0%

**Summary:**

- Services: 26
- Stacks: 3
- Deployments: 0
- Resource Pools: 3
- Containers: 0
- Images: 0
- Hosts: 1

3. 서비스 페이지에서 "새 서비스"를 클릭하여 서비스 편집기를 표시하고 다음을 수행하십시오.

a. 다음 값을 입력하십시오.

i. 서비스 이름 : Hello World Nginx 데모

ii. 이미지 : YourAccount / docker-hello-world : 최신 (Docker 허브 계정의 이미지)

iii. 포트 : TCP 프로토콜을 사용하여 호스트 포트 8080 에서 컨테이너 포트 80 으로의 매핑을 설정합니다 (나중에 포트 8080 을 사용하여 배포 된 응용 프로그램이 실행 중인지 확인합니다)

b. 새 서비스를 만들고 서비스 편집기를 닫으려면 "저장"을 클릭하십시오.

ORACLE<sup>®</sup> Container Cloud Service Get Help ▾ Settings ▾ Logout

**Services** Quick Start Wizard

< First < Prev Page 1 ▾ Next > Last > Per Page: 50 ▾

**New Service**

Actions	ID	Name	Description
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	alpine-linux	Alpine Linux	Runs an example alpine linux container that executes for 10 minutes and prints out a hello world message to the terminal. This example is provided as-is for educational purposes and should not be used in production.
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	apache	Apache	A Docker image with Apache and PHP. After the container deploys successfully, visit port 9001 on the host to see the phpinfo page. This example is provided as-is for educational purposes and should not be used in production.
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	busybox	BusyBox	Runs a BusyBox image that executes for 10 minutes and prints out a hello world to the terminal. This example is provided as-is for educational purposes and should not be used in production.
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	cadvisor	cAdvisor	A cAdvisor container that runs on port 8080 on the host. By default, this image will be deployed to all hosts. This example is provided as-is for educational purposes and should not be used in production.
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	cassandra	Cassandra	The officially supported Apache Cassandra image. This example is provided as-is for educational purposes and should not be used in production.
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	datadog	Datadog	The officially supported Datadog image. You will need to add your API key to publish data to your datadog account. This example is provided as-is for educational purposes and should not be used in production.
<a href="#">Deploy</a> <a href="#">Edit</a> <a href="#">Remove</a>	golang	Go Example Application	An example golang container that runs for 10 minutes and prints out a hello world to the terminal. This example is provided as-is for educational purposes and should not be used in production.

---

**Service Name**  **Service ID**  **Health Check**  No health checks defined.

**Service Description** Optional

**Scheduler**  **Availability**

**Builder** **Docker Run** **YAML**

**Image**  ⓘ

**Command**

**Environment Variables**

Name	Value
occs:description	It is used to demonstrate a simple Hello World Docker image using NGINX. It serves up a single HTML page that shows the hostname of the container.

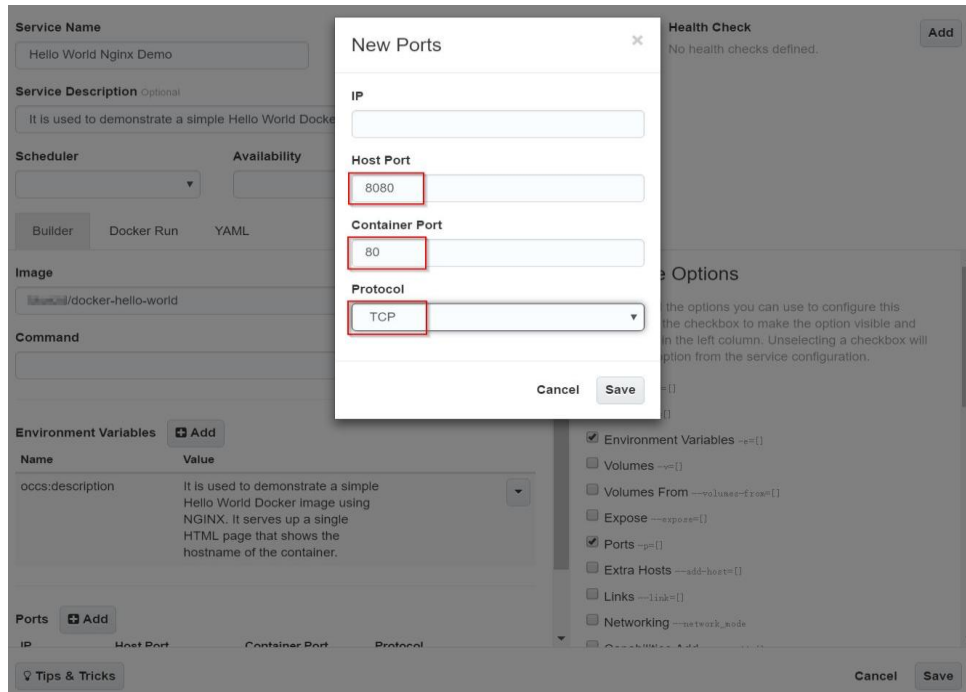
**Ports**

IP	Host Port	Container Port	Protocol
----	-----------	----------------	----------

**Available Options**

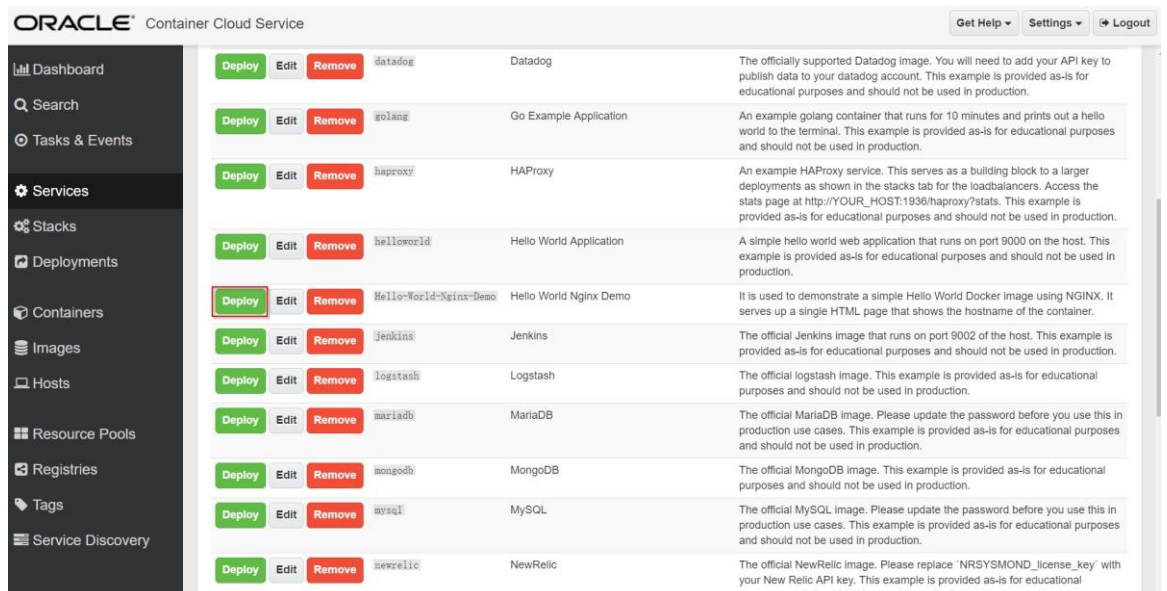
These are all the options you can use to configure this service. Use the checkbox to make the option visible and configurable in the left column. Unselecting a checkbox will remove the option from the service configuration.

- ☐ DNS `--dns=`
- ☐ Labels `--l=`
- ☒ Environment Variables `--e=`
- ☐ Volumes `--v=`
- ☐ Volumes From `--volumes-from=`
- ☐ Expose `--expose=`
- ☒ Ports `--p=`
- ☐ Extra Hosts `--add-host=`
- ☐ Links `--link=`
- ☐ Networking `--network_mode`



## Oracle Container Cloud Service 에 새 서비스로 컨테이너 인스턴스 배포

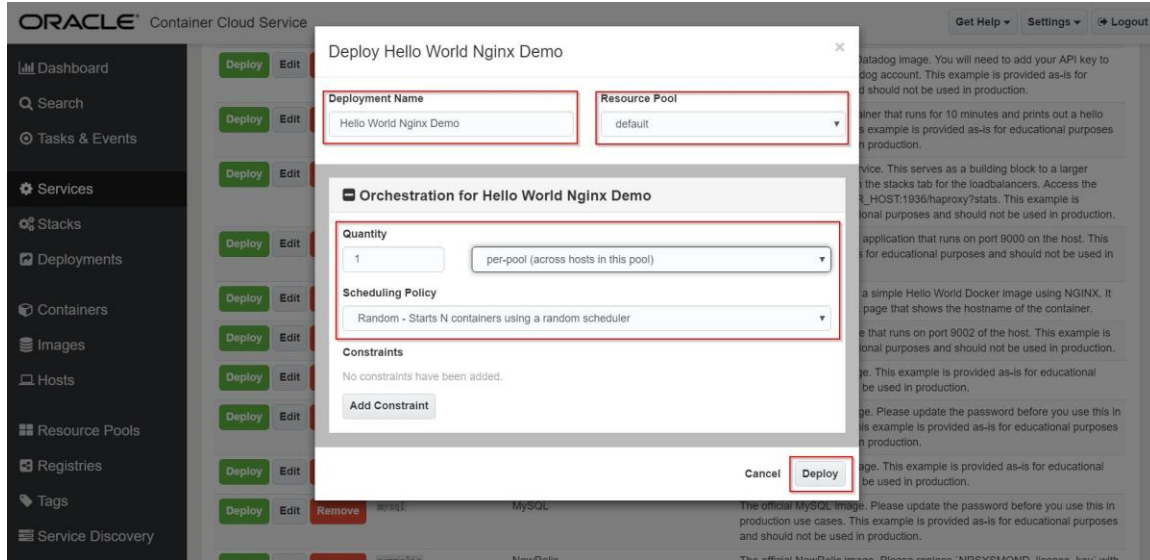
1. 컨테이너 콘솔의 서비스 페이지에서 방금 생성 한 Hello World Nginx Demo 서비스 옆에있는 "Deploy" 버튼을 클릭하십시오.





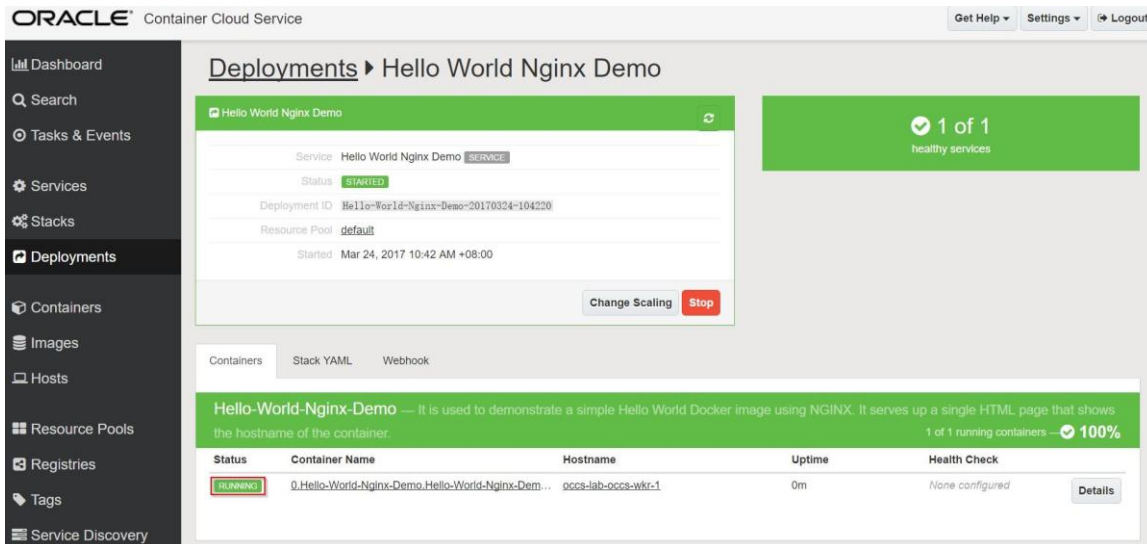
다음에 대한 기본값을 포함하는 배포 대화 상자가 나타납니다.

- 배포 이름
- 배포를 실행할 호스트 풀
- 배포 할 컨테이너 수 및 배포 할 컨테이너를 선택하는 방법



2. 배포 페이지가 나타나 방금 만든 배포의 진행 세부 정보가 표시됩니다.

우선 Docker Hub 에서 "docker-hello-world" 이미지를 가져옵니다. 이미지를 다운로드하면 컨테이너로 시작됩니다. 배치 페이지는 컨테이너의 상태를 "실행 중"으로 표시하고 컨테이너가 실행 중인 호스트의 이름을 표시합니다.



3. 컨테이너를 실행 중인 호스트의 세부 정보를 보려면 "호스트 이름" 열에서 호스트 이름을 클릭하십시오.



The screenshot shows the 'Deployments' page for 'Hello World Nginx Demo'. The service is in a 'STARTED' state. A summary box indicates '1 of 1 healthy services'. Below this, a table lists the running containers. The first container is '0.Hello-World-Nginx-Demo.Hello-World-Nginx-Demo...' with a hostname of 'occs-lab-occs-wkr-1' (highlighted with a red box). The status is 'RUNNING' and the health check is 'None configured'.

Status	Container Name	Hostname	Uptime	Health Check
RUNNING	0.Hello-World-Nginx-Demo.Hello-World-Nginx-Demo...	occs-lab-occs-wkr-1	0m	None configured

호스트 페이지가 나타나 컨테이너를 실행중인 호스트의 세부 사항을 표시합니다.

4. 호스트 페이지에서 "public\_ip" 필드에 값을 복사하십시오.

The screenshot shows the 'Hosts' page for the host 'occs-lab-occs-wkr-1'. The 'Interfaces' section lists 'eth0', 'docker0', and 'public\_ip'. The 'public\_ip' value is highlighted with a red box. The 'About' section shows 'OCSS Version: v17.1.5-10x7', 'Docker Version: 1.10.3', and 'Docker API Version: 1.22'. Below, the 'Containers on this host' section shows 1 running container. A table lists the container details:

Actions	Hostname	Name	Deployment	State	Container ID	Command
Start Stop	occs-lab-occs-wkr-1	0.Hello-World-Nginx-Demo...	Hello World Nginx Demo	Running	889f30144603	

## 배포 확인

1. 새 브라우저 탭을 엽니다.
2. http:// 형태로 "Hello World Nginx Demo" 컨테이너에서 실행되는 "docker-hello-world" 앱에 액세스 할 수 있는 URL 을 입력하십시오. <public\_ip> : 8080 여기서,

- <public\_ip> 호스트 페이지의 public\_ip 필드에서 복사 한 값입니다.
- 8080 은 이전에 컨테이너 포트 80 에 매핑 한 호스트 포트입니다.



이제 Docker Hub 에서 docker-hello-world 애플리케이션을 성공적으로 구축하고 Oracle Container Cloud Service 를 사용하여 실행중인 Docker 컨테이너에 배포했습니다.