

Docker JumpStart

Building and Managing Docker Images

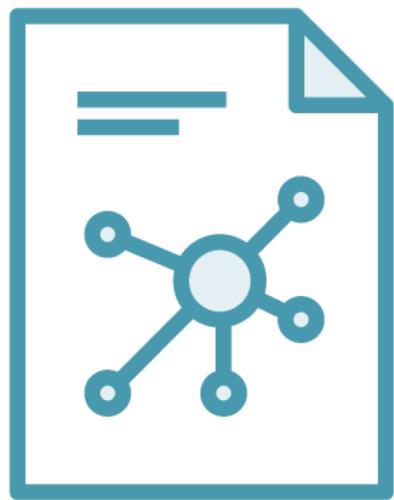
Agenda

- Container Registries
- Working with Images
- Getting Started with a Custom Dockerfile
- Building an Image
- Multi-Stage Dockerfiles
- Updating an Image
- Pushing an Image to a Registry



Container Registries

Images and Container Registries



Docker Image

Docker images need to be stored in an accessible area



Container Registry

A container registry stores Docker images

Docker Registries, Images and Containers

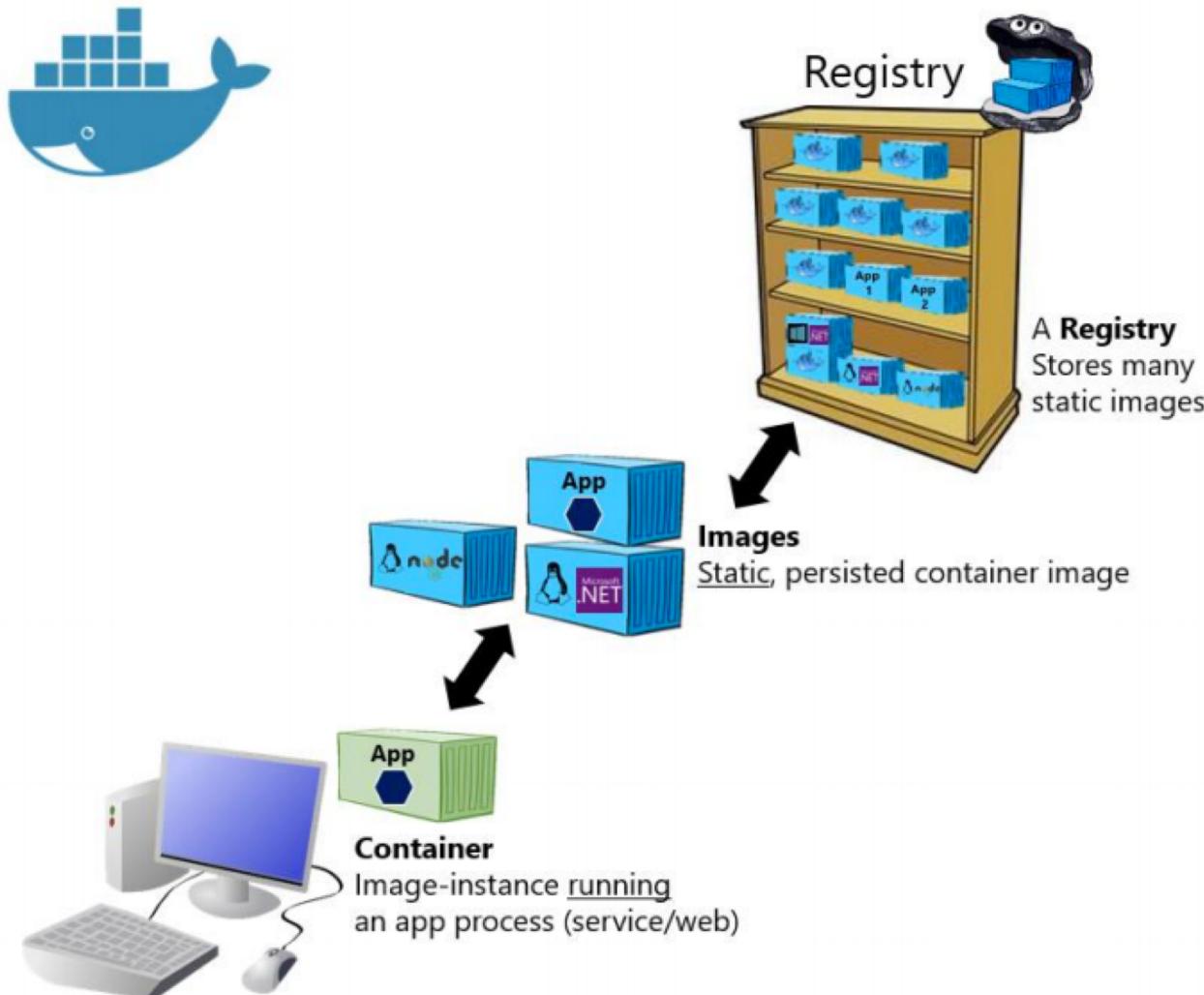


Image from <http://aka.ms/MicroservicesEbook>

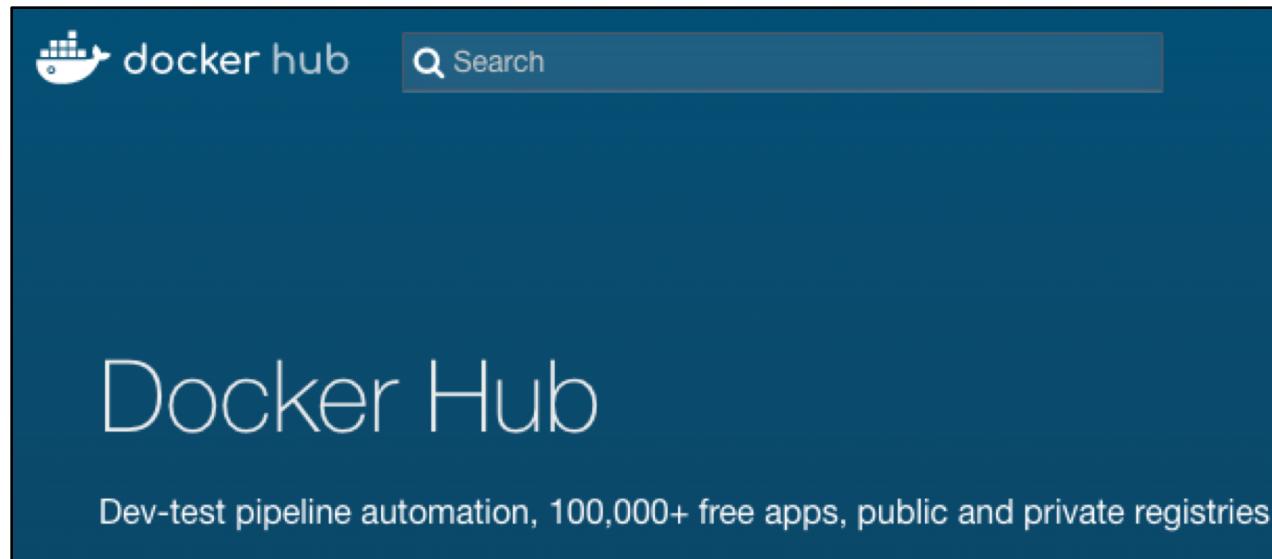
Container Registries

- Docker images can be stored in the cloud or onsite using "container registries"
- Container registry options:
 - Docker Hub
 - Azure Container Registry
 - Amazon EC2 Container Registry
 - Private registry
 - Others...



Docker Hub

- Docker Hub is a cloud-based service for hosting public or private images
- <https://hub.docker.com>



Azure Container Registry

- Azure Container Registry is a cloud-based service from Microsoft for storing images
- <https://azure.microsoft.com/en-us/services/container-registry>

Container Registry

Manage a Docker private registry as a first-class Azure resource

- ✓ Store and manage container images across all types of Azure deployments
- ✓ Keep container images near deployments to reduce latency and costs
- ✓ Maintain Windows and Linux container images in a single Docker registry
- ✓ Use familiar, open-source Docker command line interface (CLI) tools
- ✓ Simplify registry access management with Azure Active Directory



Amazon EC2 Container Registry

- EC2 Container Registry is a cloud-based service from Amazon for storing images
- <https://aws.amazon.com/ecr>

Amazon EC2 Container Registry

Amazon EC2 Container Registry (ECR) is a fully-managed [Docker](#) container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with [Amazon EC2 Container Service \(ECS\)](#), simplifying your development to production workflow. Amazon ECR eliminates the need to operate your own container repositories or worry about scaling the underlying infrastructure. Amazon ECR hosts your images in a highly available and scalable architecture, allowing you to reliably deploy containers for your applications. Integration with AWS Identity and Access Management (IAM) provides resource-level control of each repository. With Amazon ECR, there are no upfront fees or commitments. You pay only for the amount of data you store in your repositories and data transferred to the Internet.

Private/Onsite Container Registry

- Docker Hub is a cloud-based service for hosting public or private images
- <https://docs.docker.com/registry>

Docker Registry

Estimated reading time: 1 minute

What it is

The Registry is a stateless, highly scalable server side application that stores and lets you distribute Docker images. The Registry is open-source, under the permissive [Apache license](#).

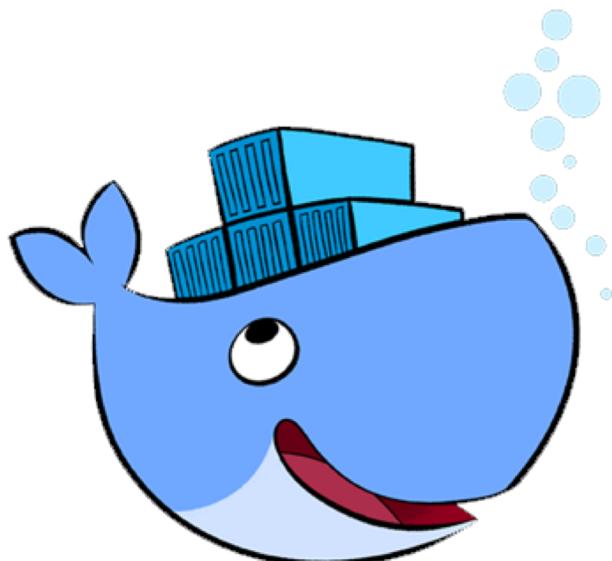
Why use it

You should use the Registry if you want to:

- tightly control where your images are being stored
- fully own your images distribution pipeline
- integrate image storage and distribution tightly into your in-house development workflow

Working with Images

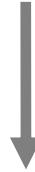
Key Docker Image Commands



`docker images`
`docker pull [image name]`
`docker rmi [imageId]`

Listing Local Images

docker images

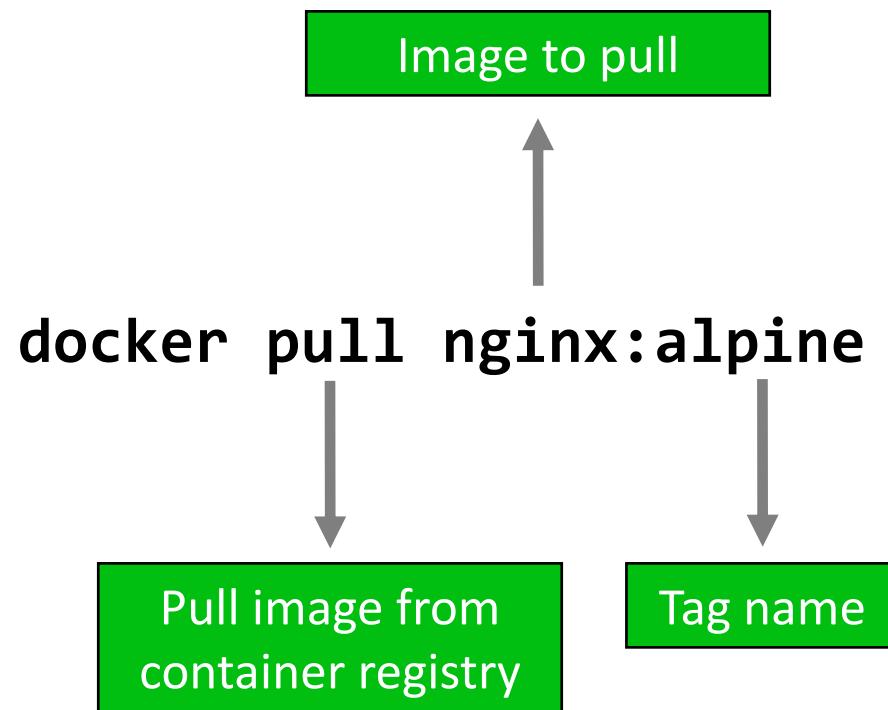


List images in
local registry

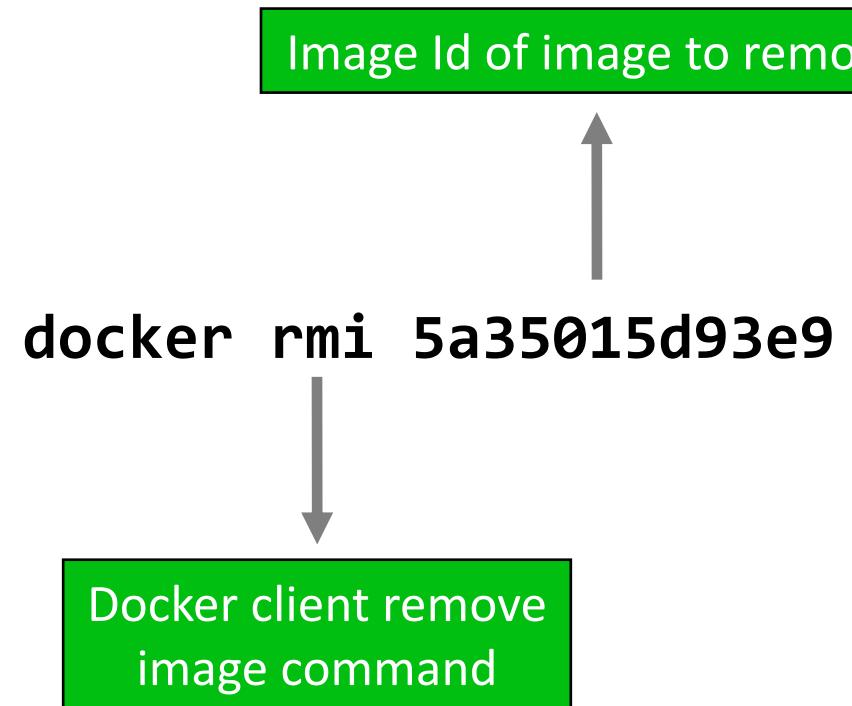
Key Image Information

Image "repository" (name)					Image size		
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE			
nginx	alpine	5a35015d93e9	2 weeks ago	15.5MB			
aspnetcoreapp	latest	c1ecdaa24d81	6 weeks ago	878MB			
danwahlin/wordpress	latest	14c6da09b637	7 weeks ago	787MB			
danwahlin/node-codewithdan	latest	73c435391935	8 weeks ago	400MB			
danwahlin/mariadb	latest	ba53e558f318	8 weeks ago	407MB			
microsoft/dotnet		af1675435d14	2 months ago	878MB			
nodeexpressmongodbdockerapp_node	latest	4efa30943640	2 months ago	689MB			
node	latest	14b343820550	3 months ago	663MB			
danwahlin/nginx	latest	83ccdbbd88ab	3 months ago	59.6MB			
danwahlin/mongo	latest	4b38a839bbfa	3 months ago	460MB			
mongo	latest	ad974e767ec4	3 months ago	402MB			
node	alpine	0895ecd79009	3 months ago	54.6MB			
wordpress	latest	109633df95f5	3 months ago	400MB			
nginx	<none>	c24ab147adf9	3 months ago	54.3MB			
postgres	latest	4023a747a01a	3 months ago	265MB			
mariadb	latest	7eca0e0b51c9	3 months ago	393MB			
google/cadvisor	latest	cc8254dd08c6	7 months ago	47.8MB			

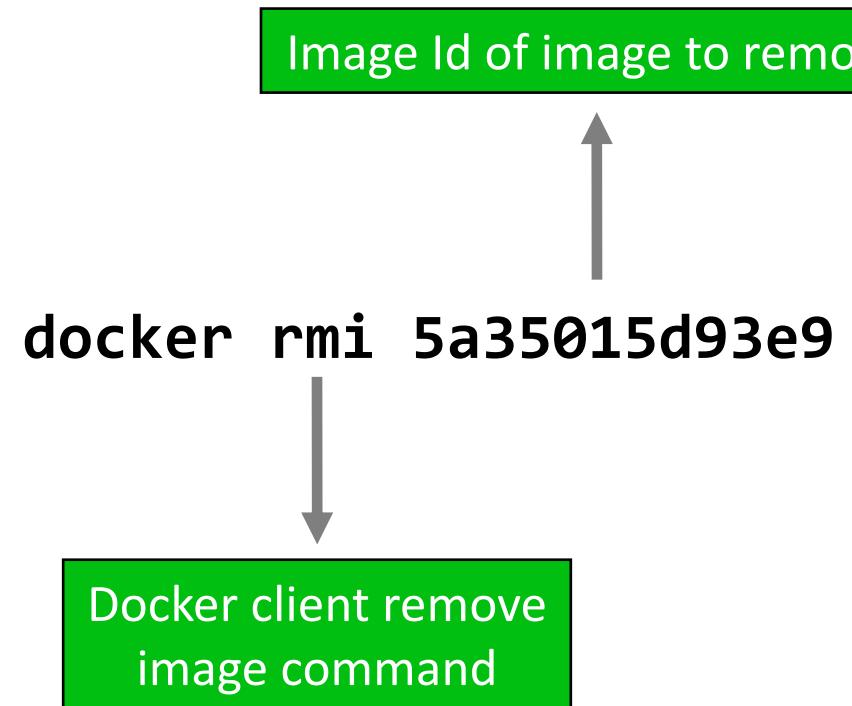
Pulling an Image from a Registry



Removing an Image

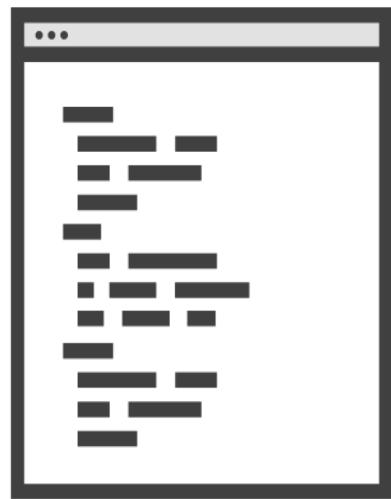


Removing an Image

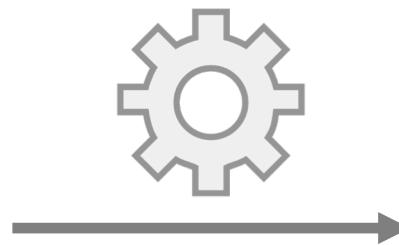


Getting Started with a Custom Dockerfile

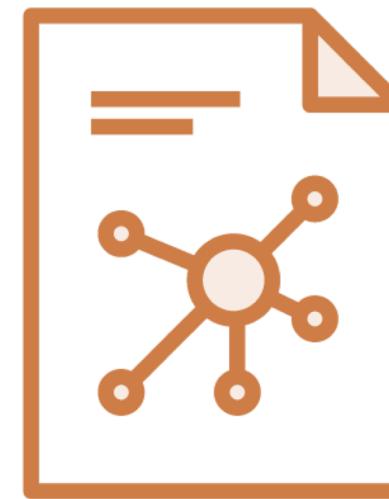
Dockerfile and Images



Dockerfile

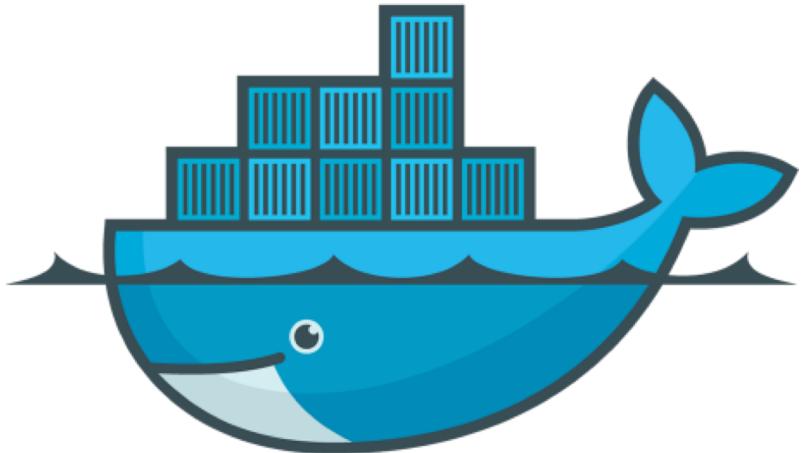


docker build



Docker Image

Dockerfile Overview



Text file used to build Docker images
Contains build instructions
Instructions create intermediate image
that can be cached to speed up future
builds
Used with "docker build" command

Key Dockerfile Instructions

FROM

LABEL

RUN

COPY

ENTRYPOINT

WORKDIR

EXPOSE

ENV

CMD

Dockerfile Example: nginx

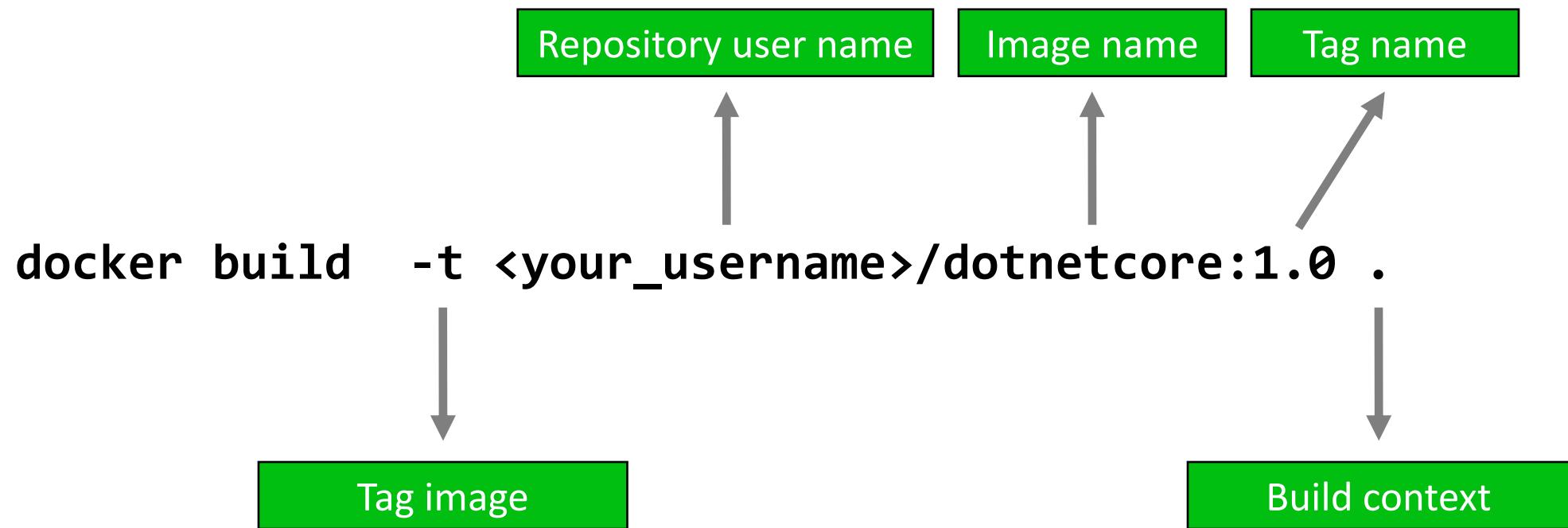
```
FROM nginx:alpine
LABEL author="Docker"
COPY ./dist /usr/share/nginx/html
COPY ./config/nginx.conf /etc/nginx/conf.d/default.conf
```

Dockerfile Example: .NET Core

```
FROM      microsoft/dotnet:x.x.x-sdk
LABEL    author="Jimmy Docker"
ENV      ASPNETCORE_URLS=http://*:5000
WORKDIR /var/www/aspnetcoreapp
COPY      .
CMD      ["/bin/bash", "-c","dotnet restore && dotnet run"]
```

Building an Image

Building a Custom Image



Building with a Custom Dockerfile

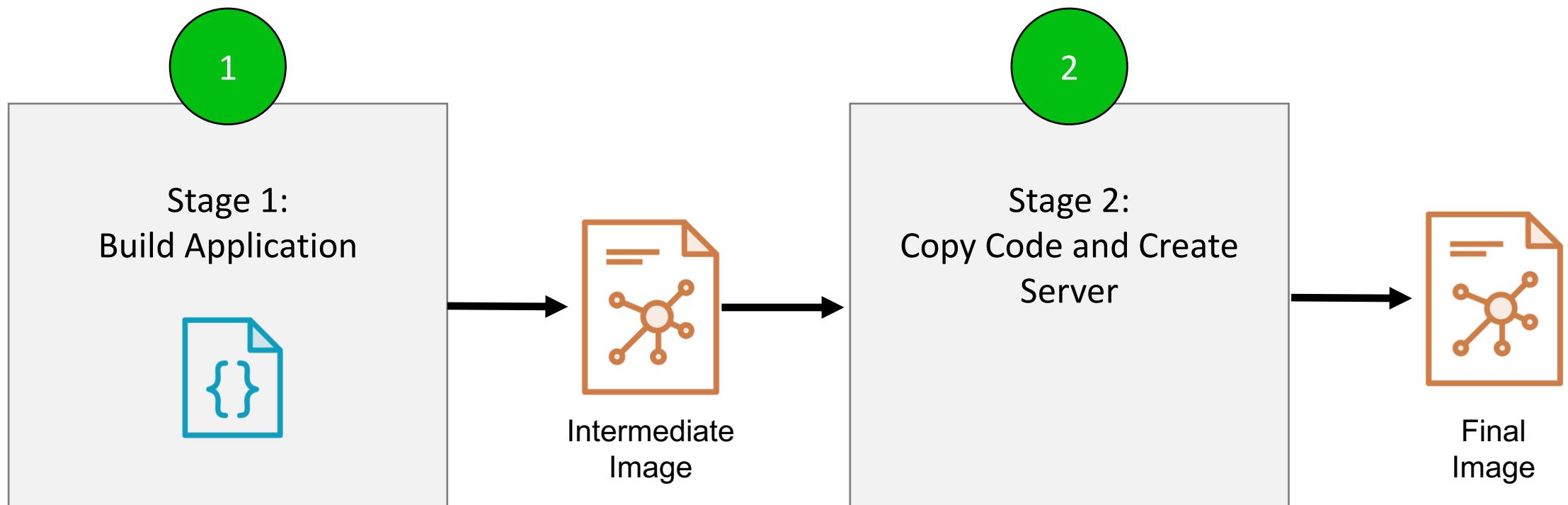
```
docker build -t dotnetcore:1.0 --file my.dockerfile .
```



Define dockerfile to use

Multi-Stage Dockerfiles

Multi-Stage Dockerfiles



Multi-stage Build Benefits



- Avoid manual creation of intermediate images
- Reduce complexity
- Selectively copy artifacts from one stage to another
- Smaller final image size

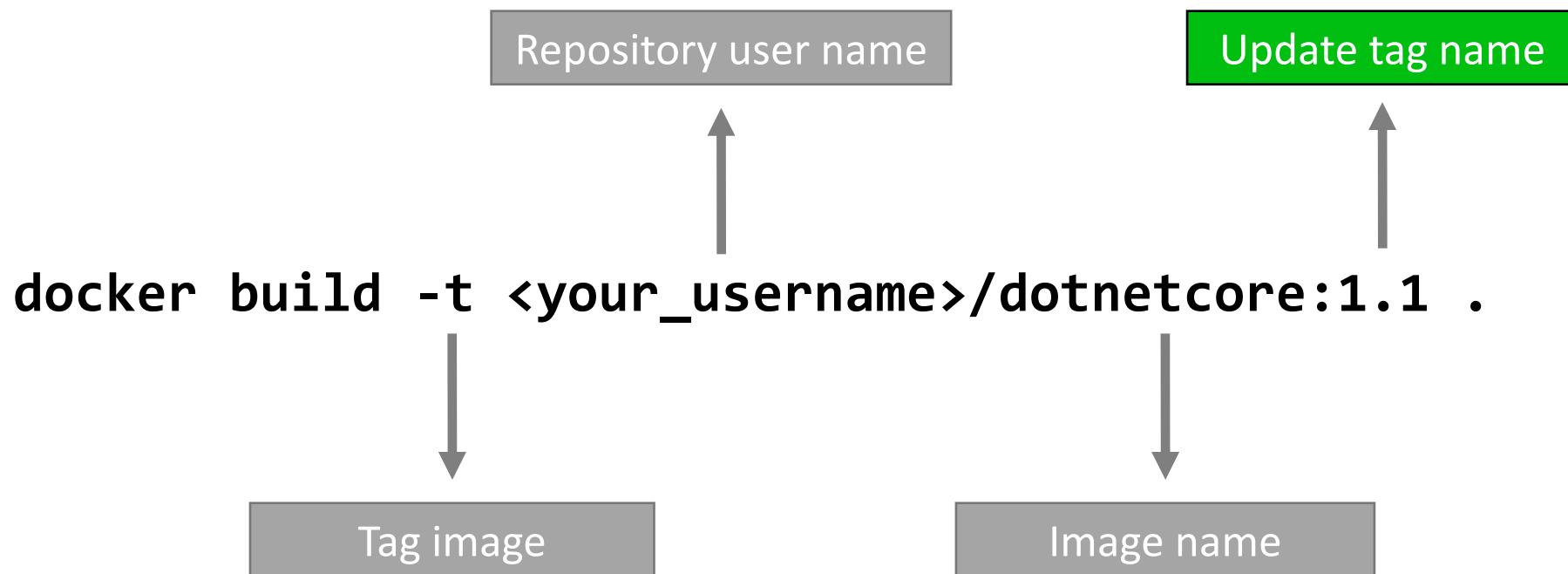
Multi-Stage Dockerfile Example

```
##### Stage 1
FROM node:latest as node
LABEL author="Dan Wahlin"
WORKDIR /app
COPY package.json package.json
RUN npm install
COPY . .
RUN npm run build -- --prod

##### Stage 2
FROM nginx:alpine
VOLUME /var/cache/nginx
COPY --from=node /app/dist /usr/share/nginx/html
COPY ./config/nginx.conf /etc/nginx/conf.d/default.conf
```

Updating an Image

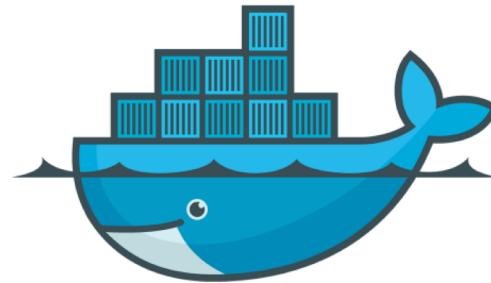
Building an Image with a Tag



Pushing an Image to a Repository

Publishing an Image to Docker Hub

```
docker push <your username>/dotnetcore
```



Container Registry

Summary

- Docker client CLI commands can be used to:
 - Interact with images
 - Build images
 - Push images to container repositories
- A custom Dockerfile can be created to build a custom image
- Images can be "tagged" to control the version
- Images can be pushed to a container repository



Lab

Building an Image Using a Dockerfile

<https://labs.codewithdan.com/docker-jumpstart?code=docker-jumpstart>

