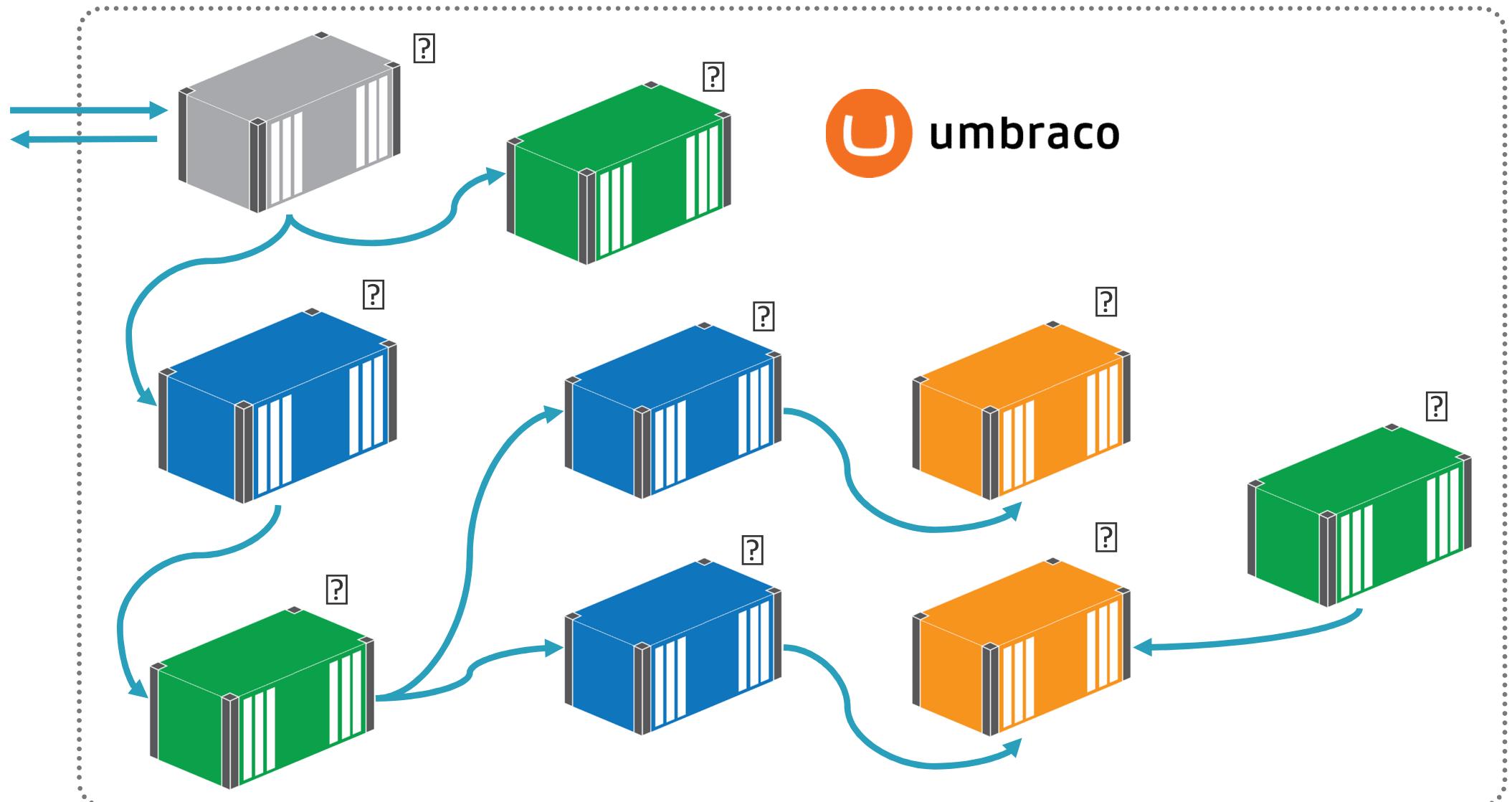


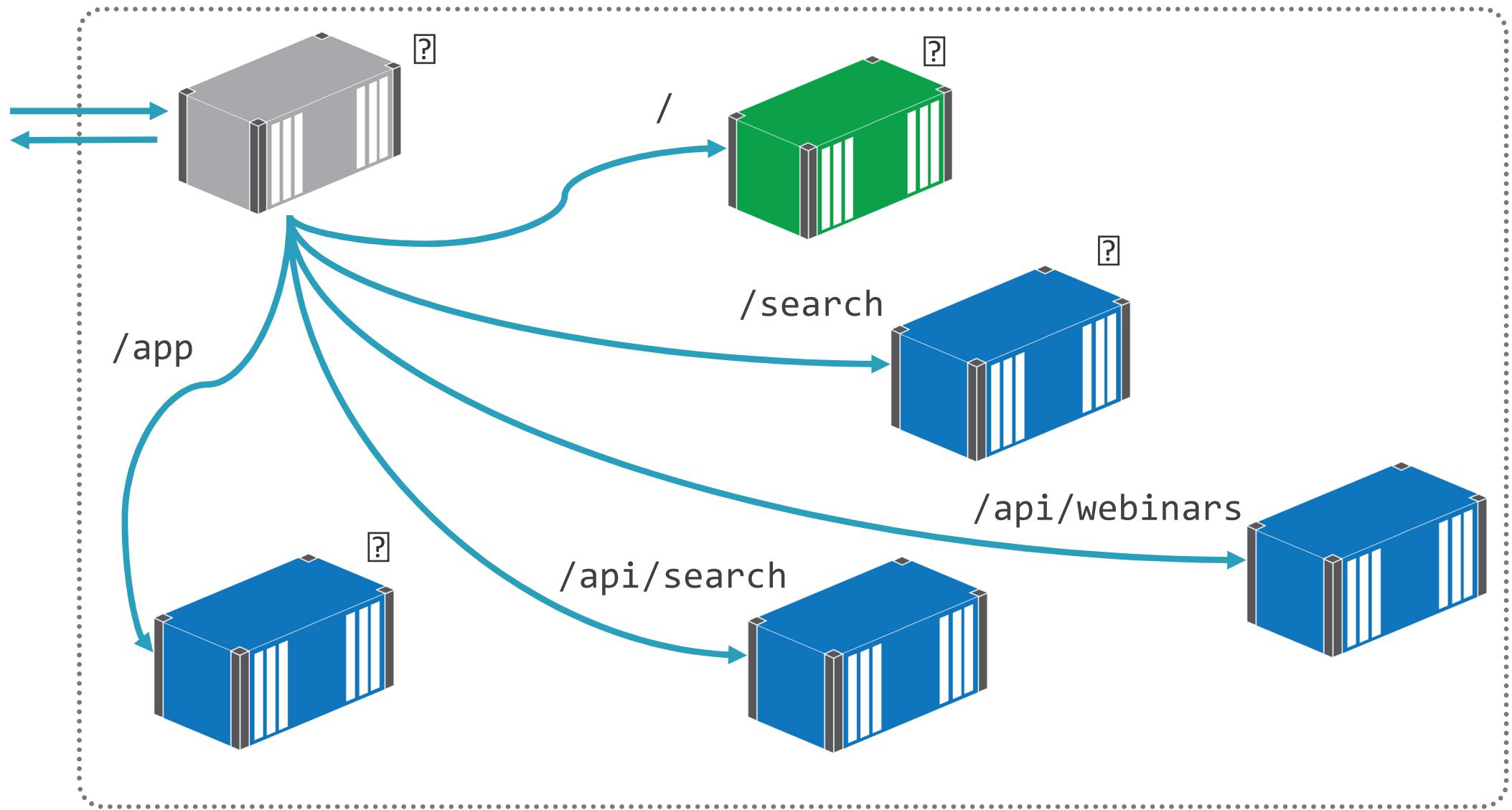
Providing Self-service Content Management with Umbraco

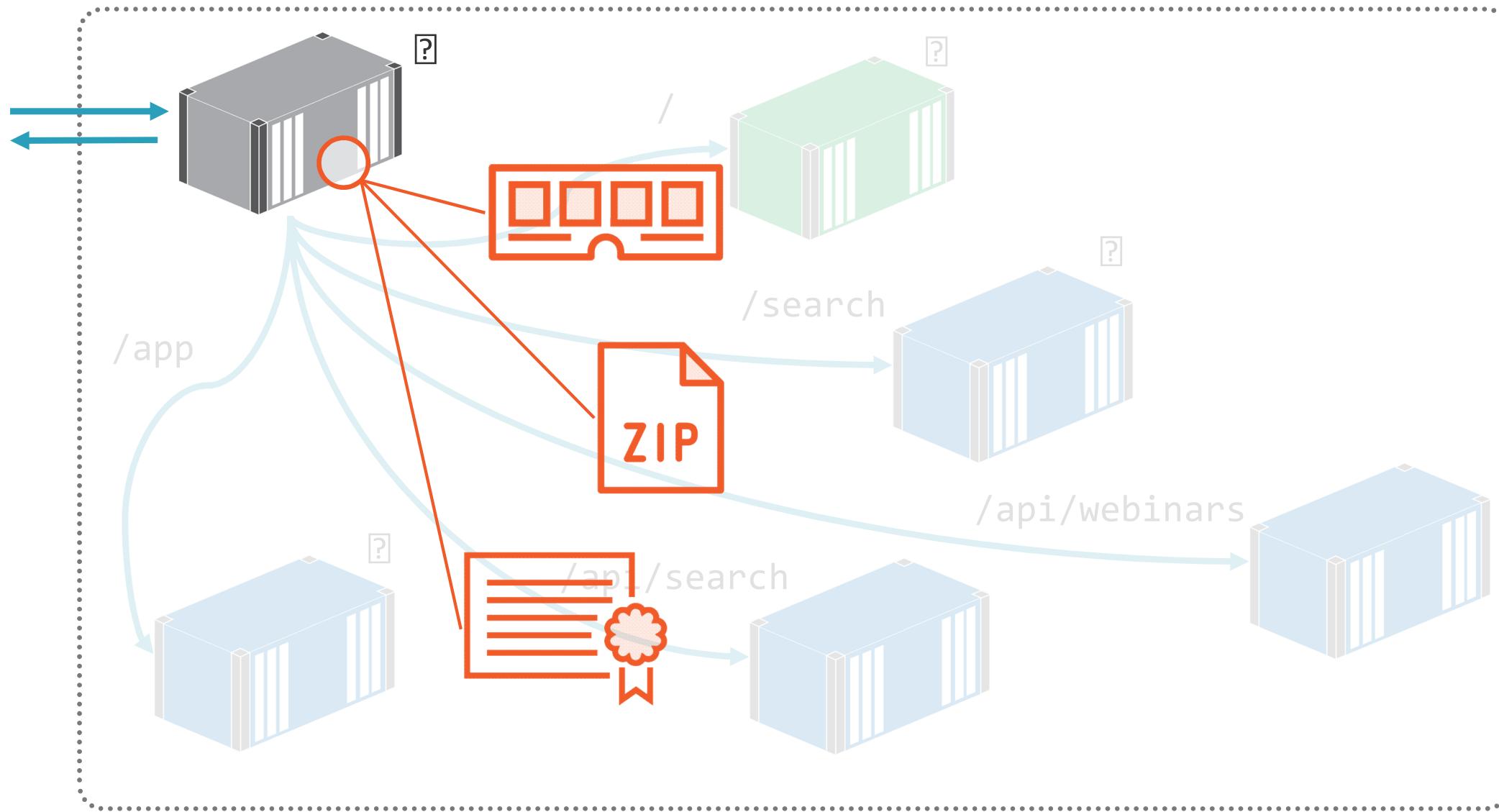


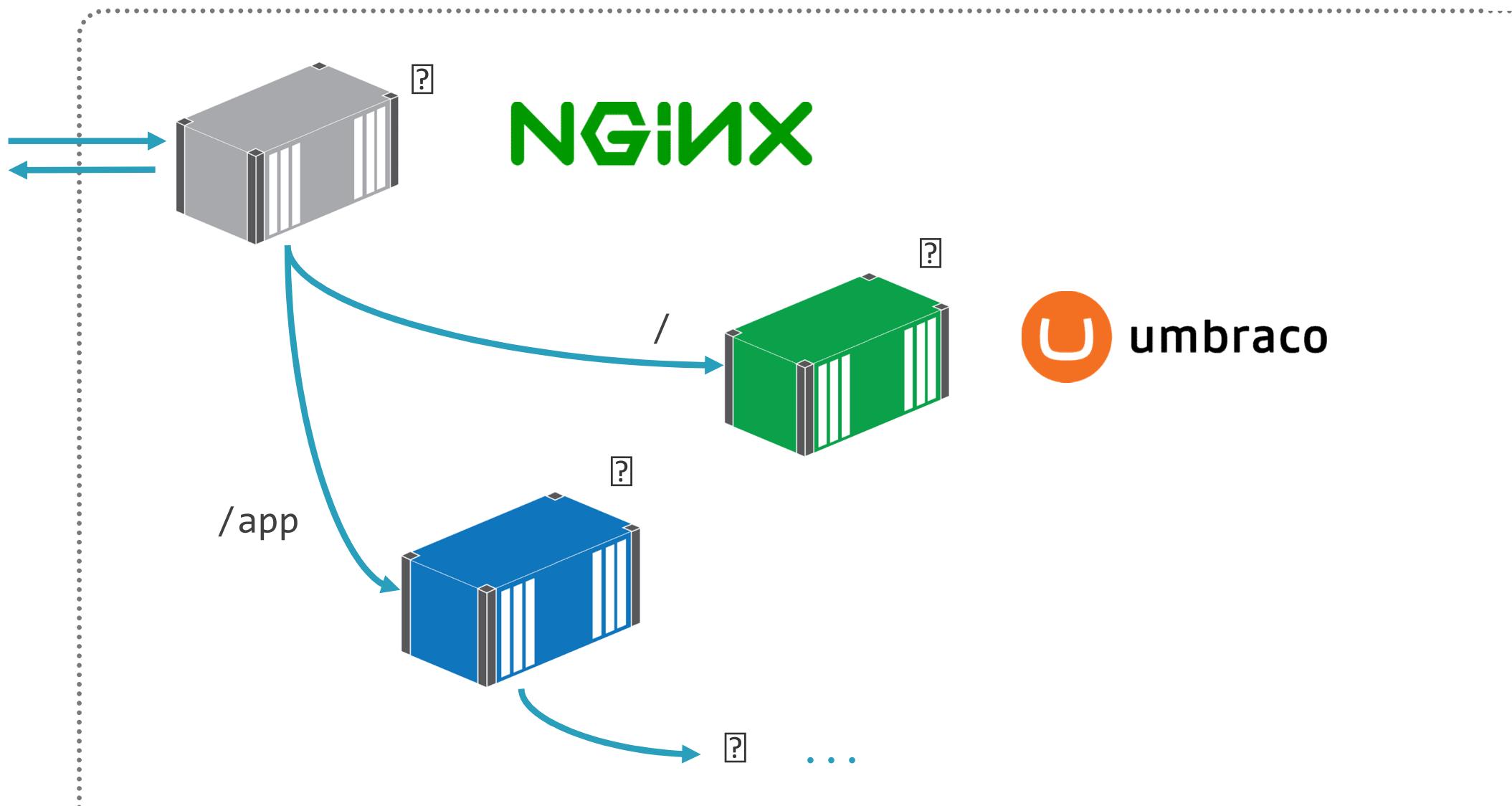
Elton Stoneman
DEVELOPER ADVOCATE

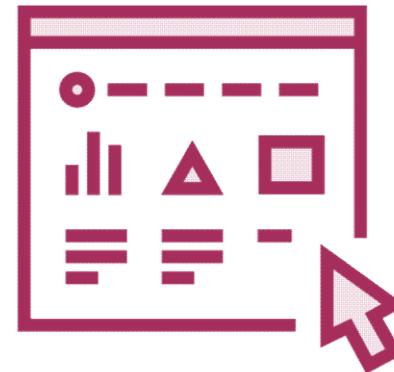
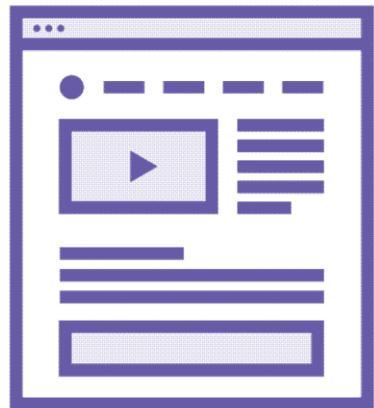
@EltonStoneman <https://blog.sixeyed.com>



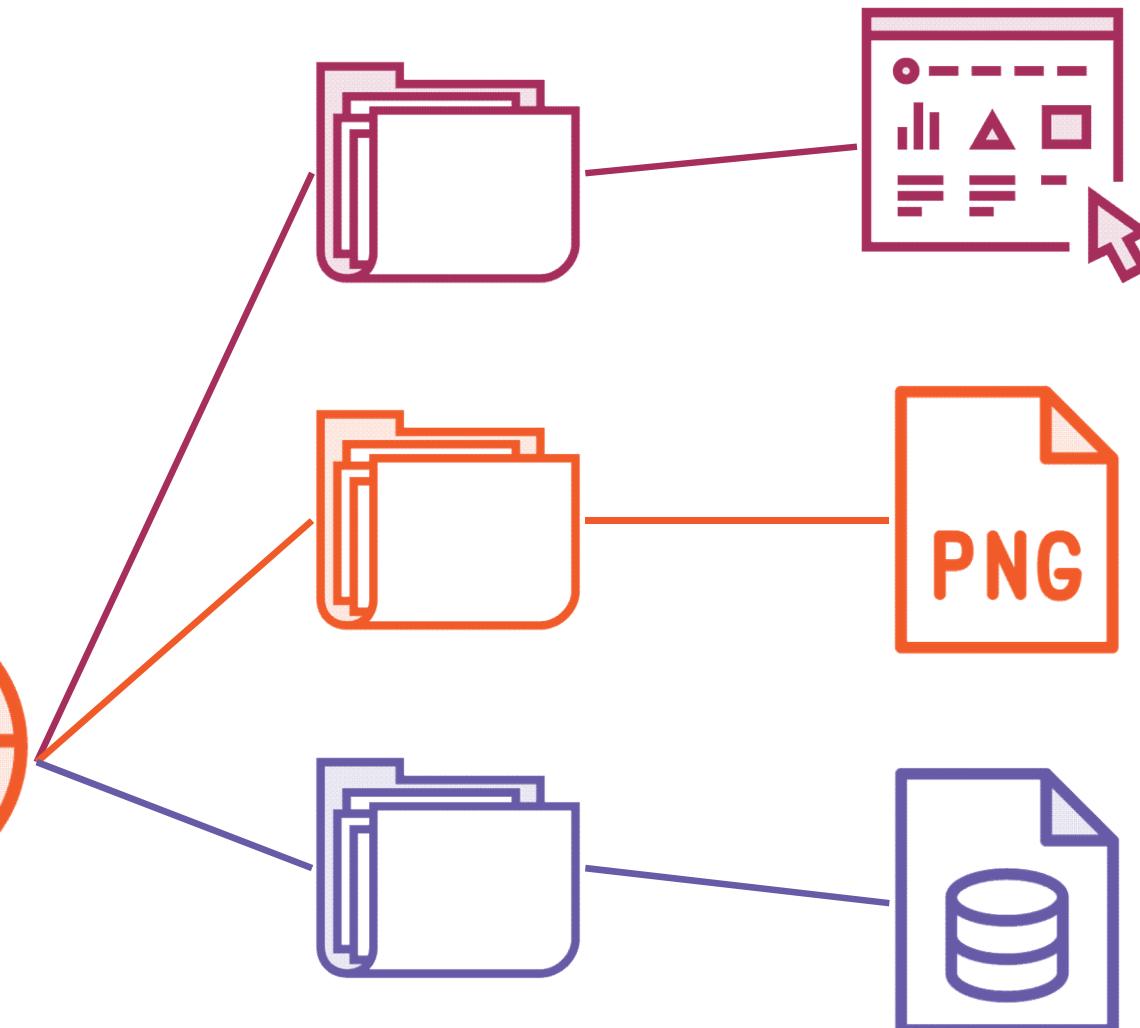




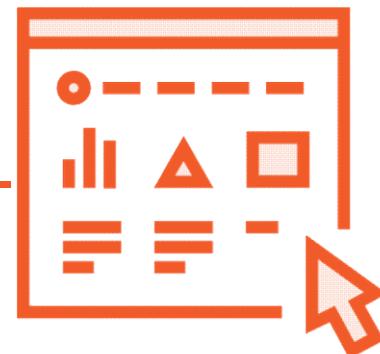




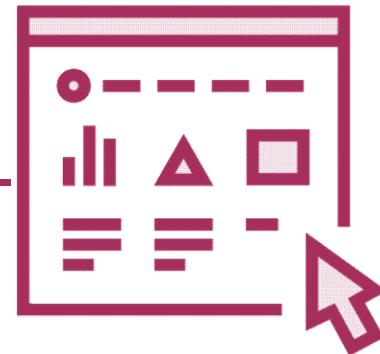
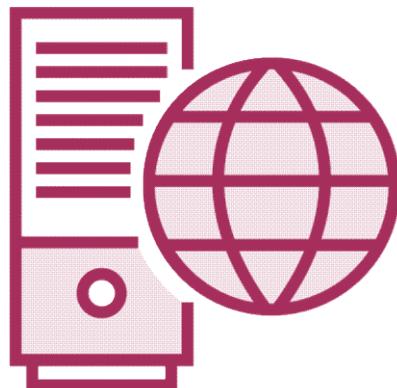
 umbraco

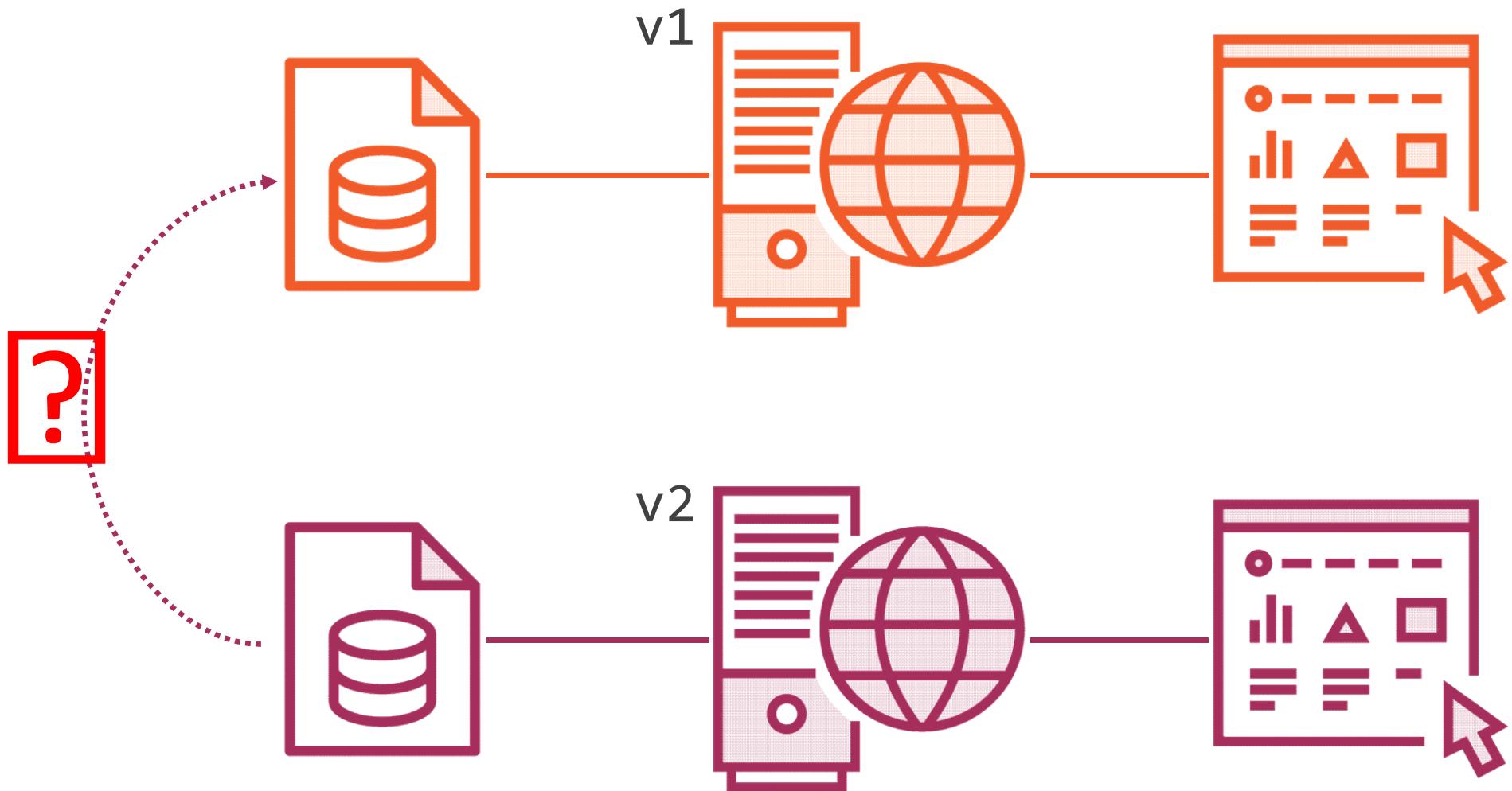


v_1

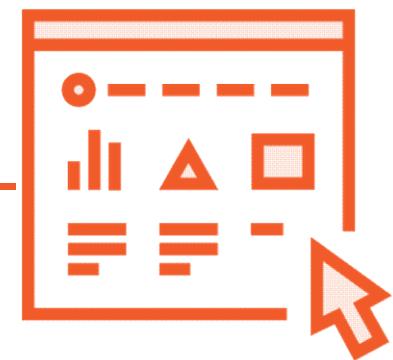


v_2





v1

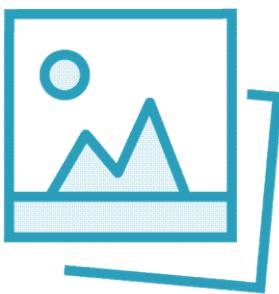


v2

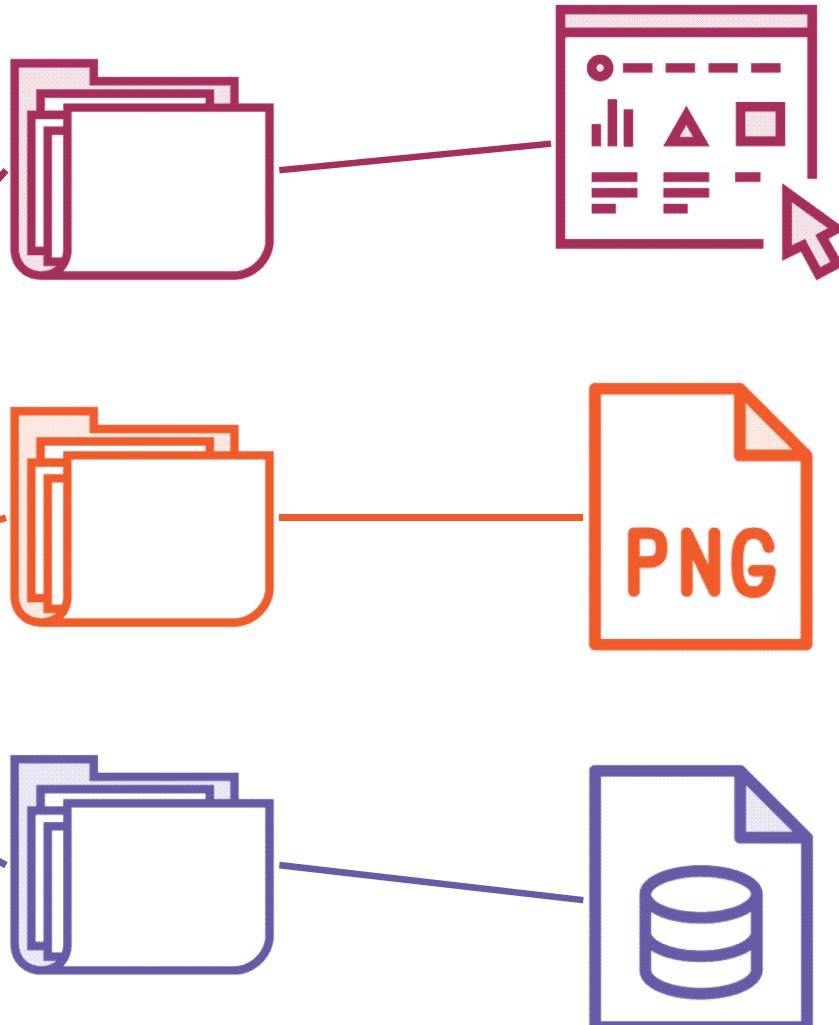


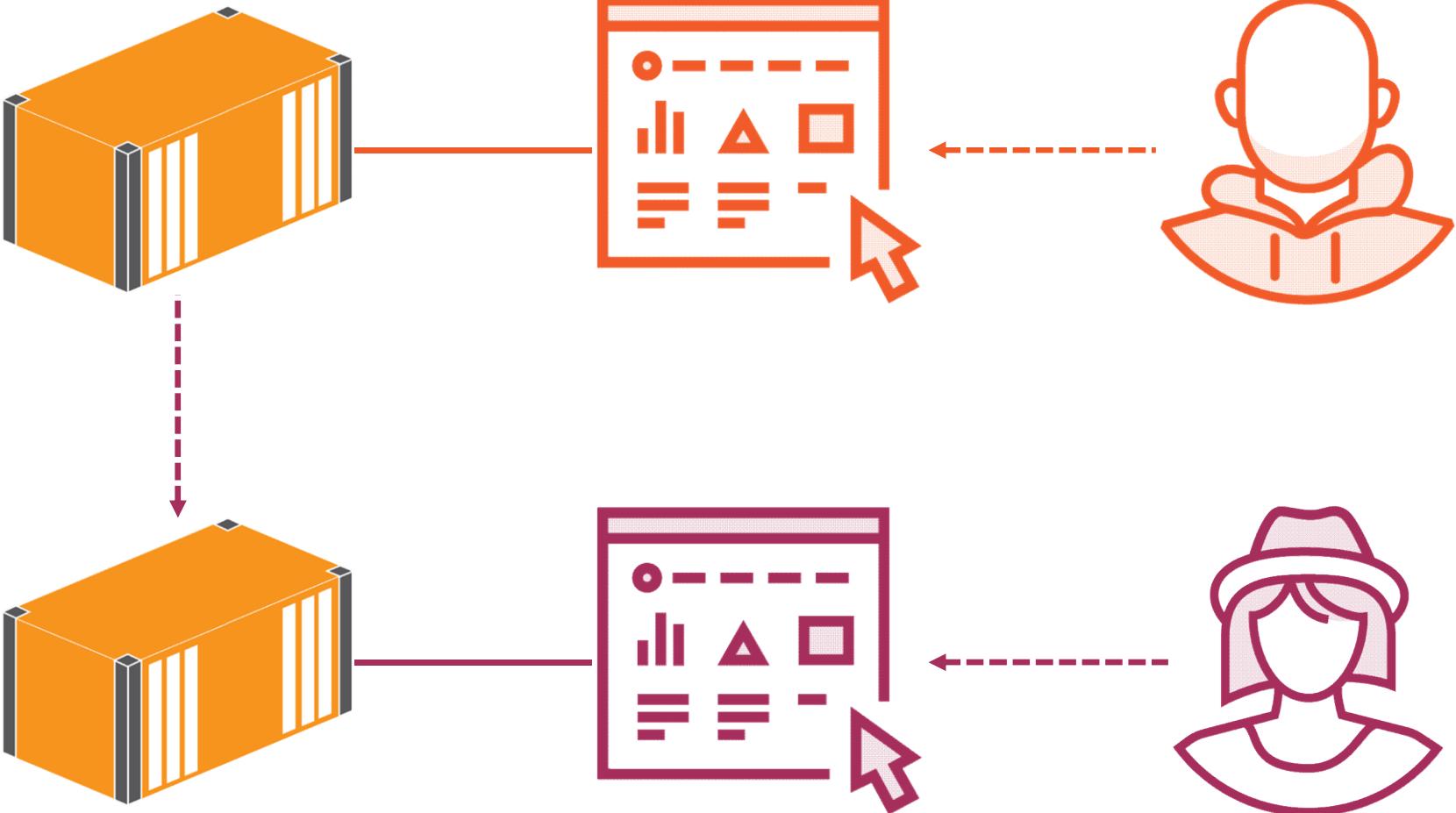


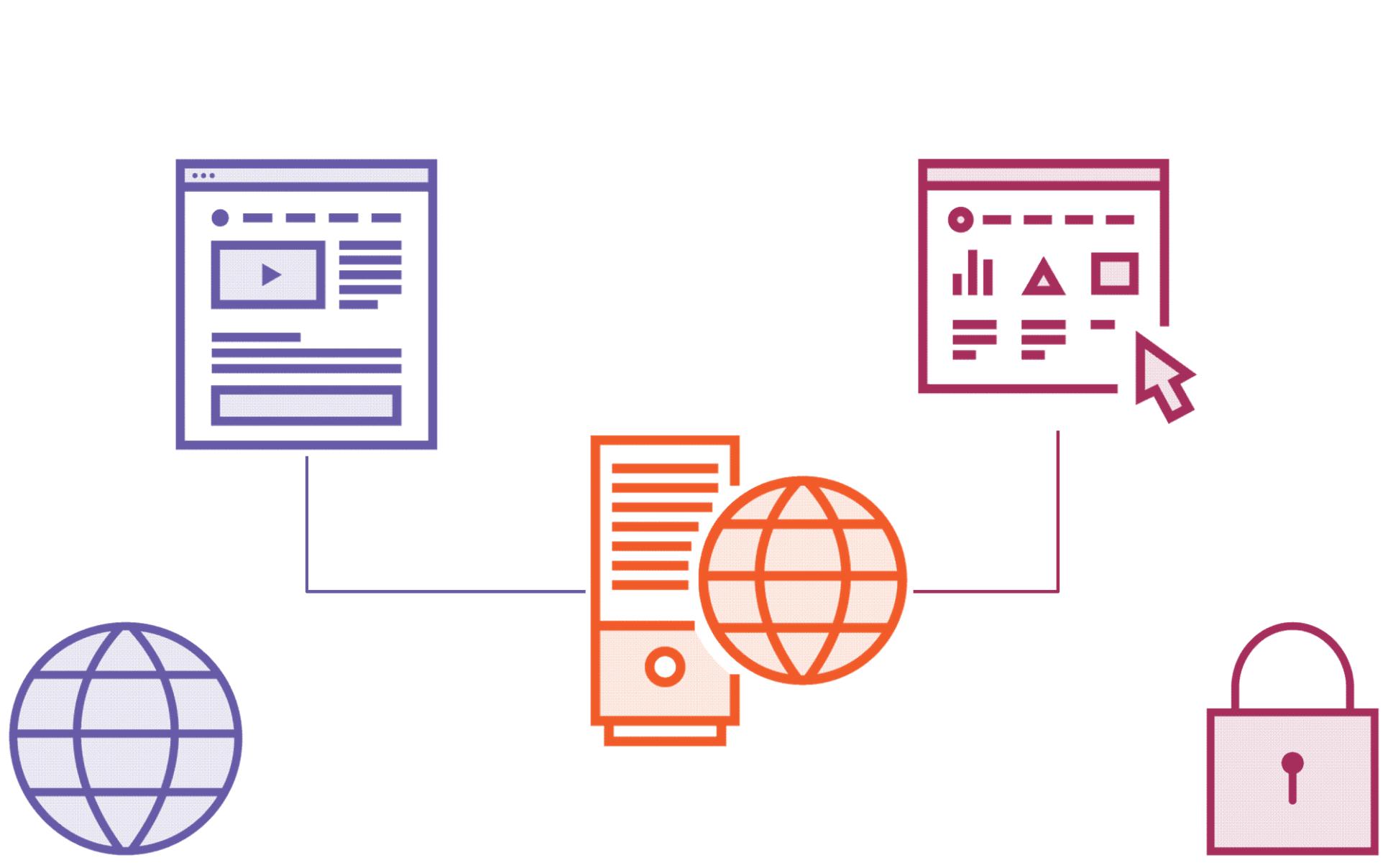
umbraco



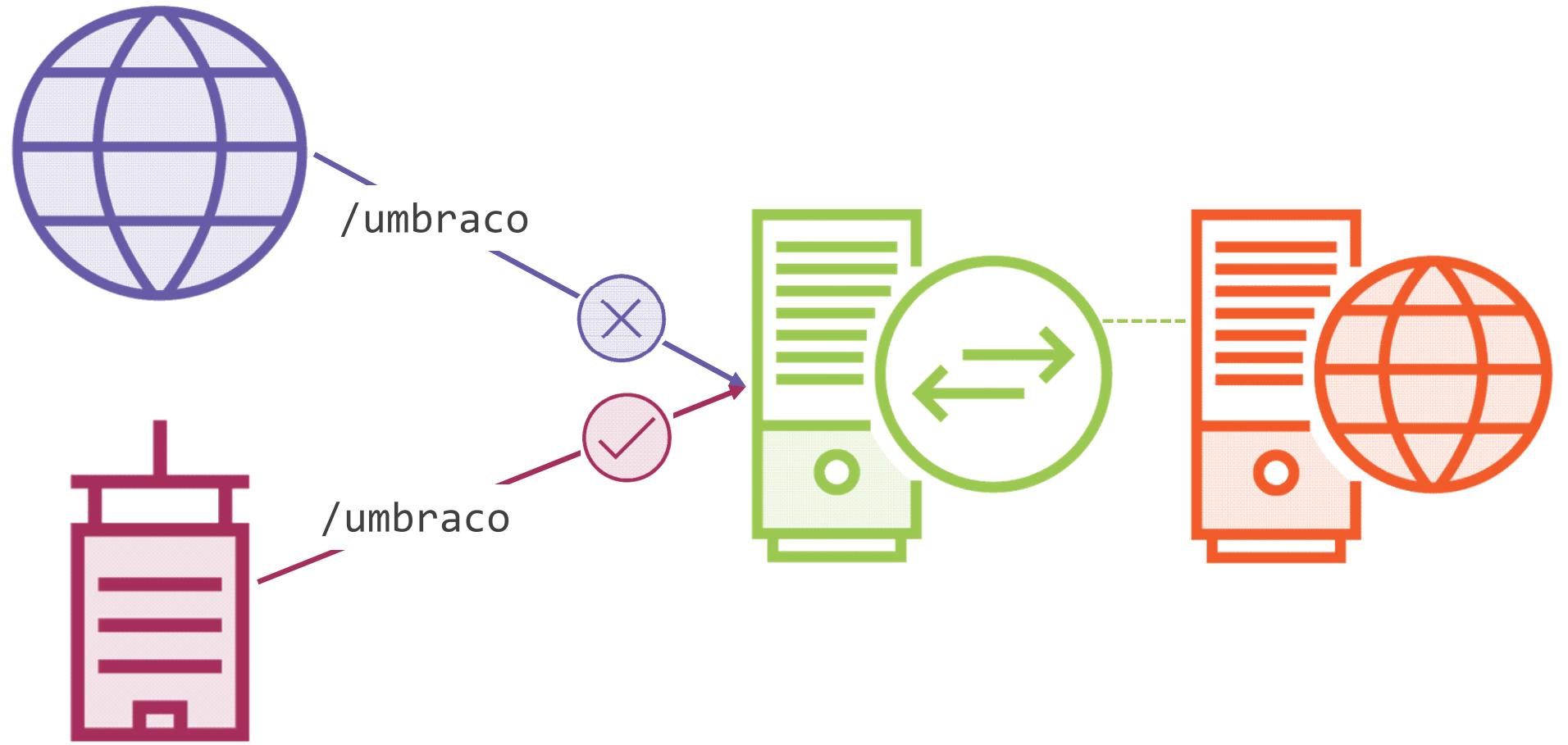
—

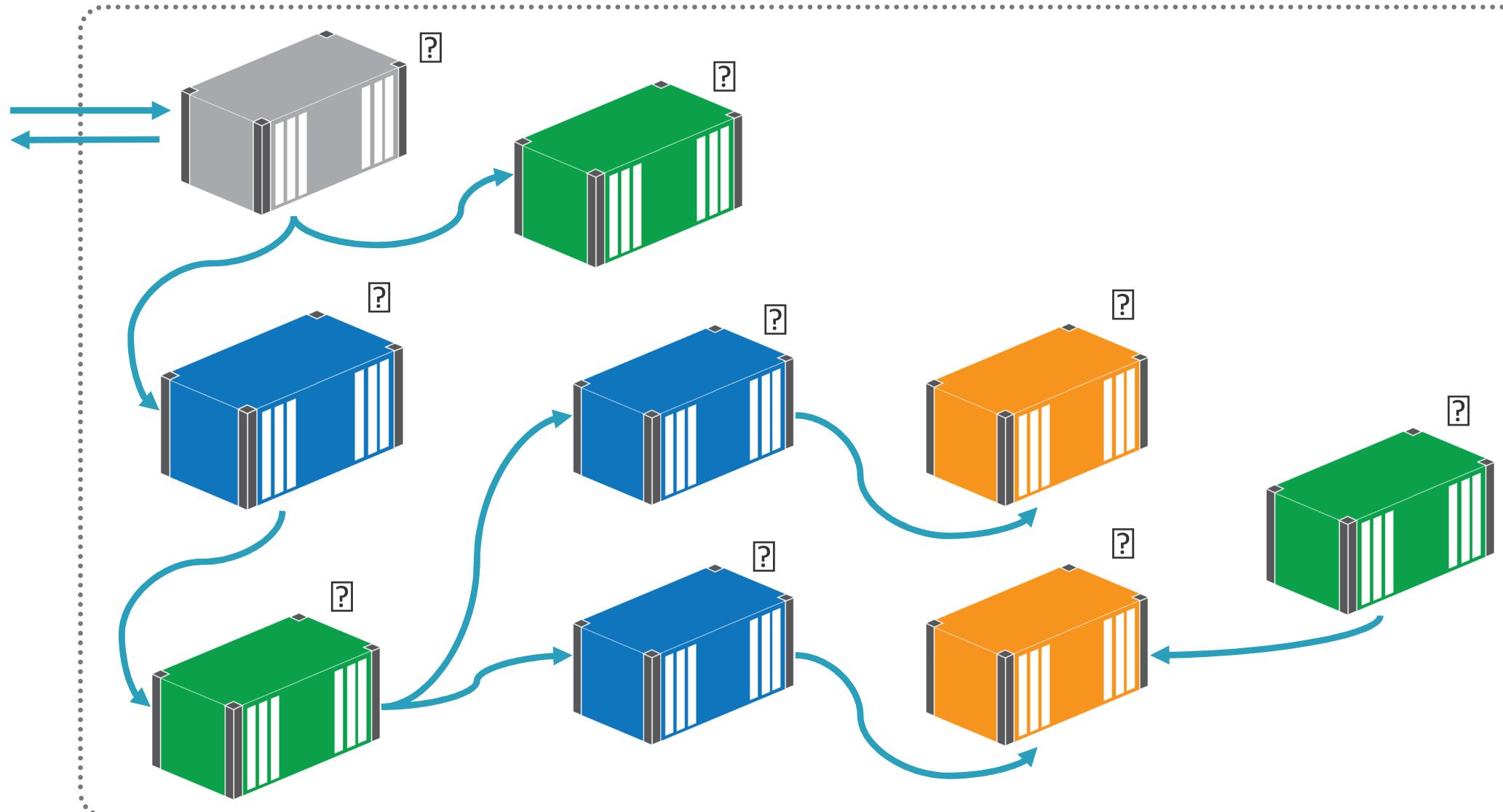




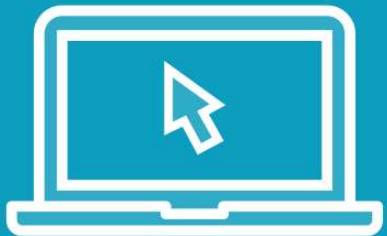








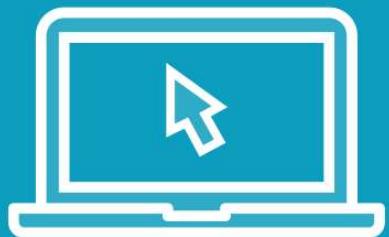
Demo



Umbraco in Docker on Windows

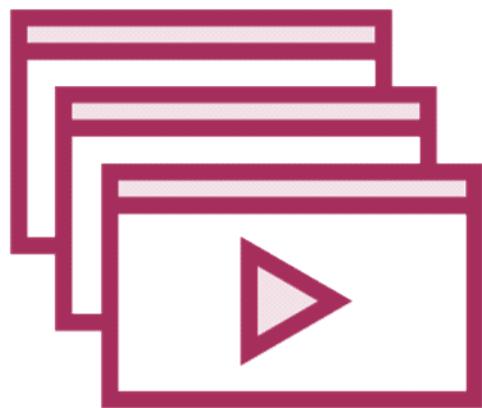
- Installation with Dockerfile
- Configuration with a container
- Extracting a configured image

Demo



Setting up the CMS Homepage

- Running Umbraco in Docker
- Configuring homepage content
- Saving the homepage image



Umbraco Jumpstart

- Aaron Powell

```
ENV UMBRACO_VERSION="7.7.4" `  
UMBRAKO_DOWNLOAD_ID="774"  
  
RUN Invoke-WebRequest -OutFile umbraco.zip ... ;`  
Expand-Archive umbraco.zip -DestinationPath C:\umbraco
```

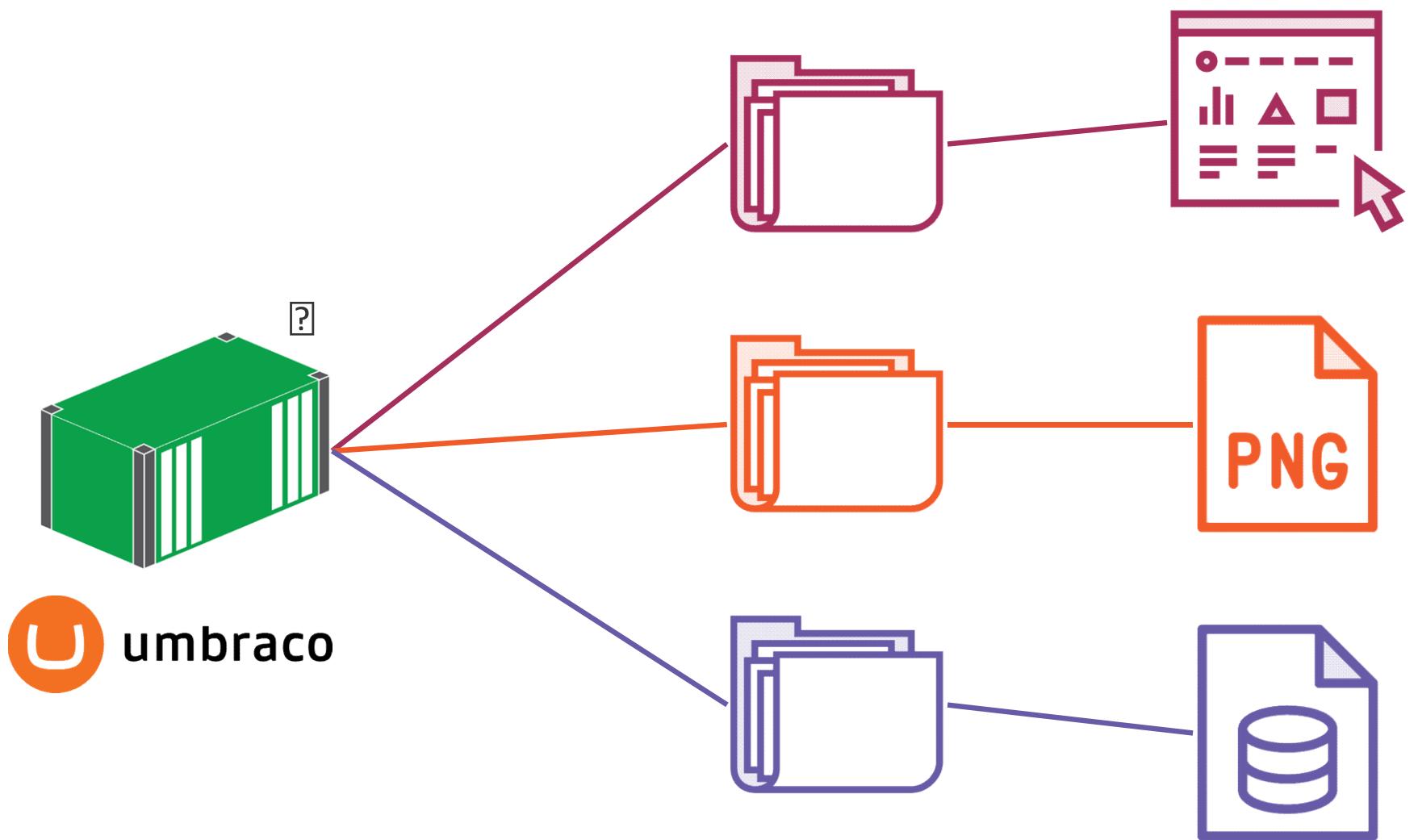
Umbraco Installer Stage
Download and unzip installation archive

```
RUN Remove-Website 'Default Web Site'; `  
New-Item -Path $env:UMBRACO_ROOT -Type Directory; `  
New-Website -PhysicalPath $env:UMBRACO_ROOT ...;  
  
COPY --from=installer C:\umbraco ${UMBRACO_ROOT}
```

Umbraco Application Stage
Configure IIS and copy application content

```
docker container run -d -P `  
  --name umbraco-setup `  
umbraco-setup
```

Run Umbraco in Docker
Complete installation interactively



```
docker container stop umbraco-setup
```

```
docker container commit umbraco-setup umbraco:v0
```

Commit Container to Image

A configured Umbraco instance - umbraco:v0

```
docker container run -d -P `  
--name umbraco:v0 `  
umbraco
```

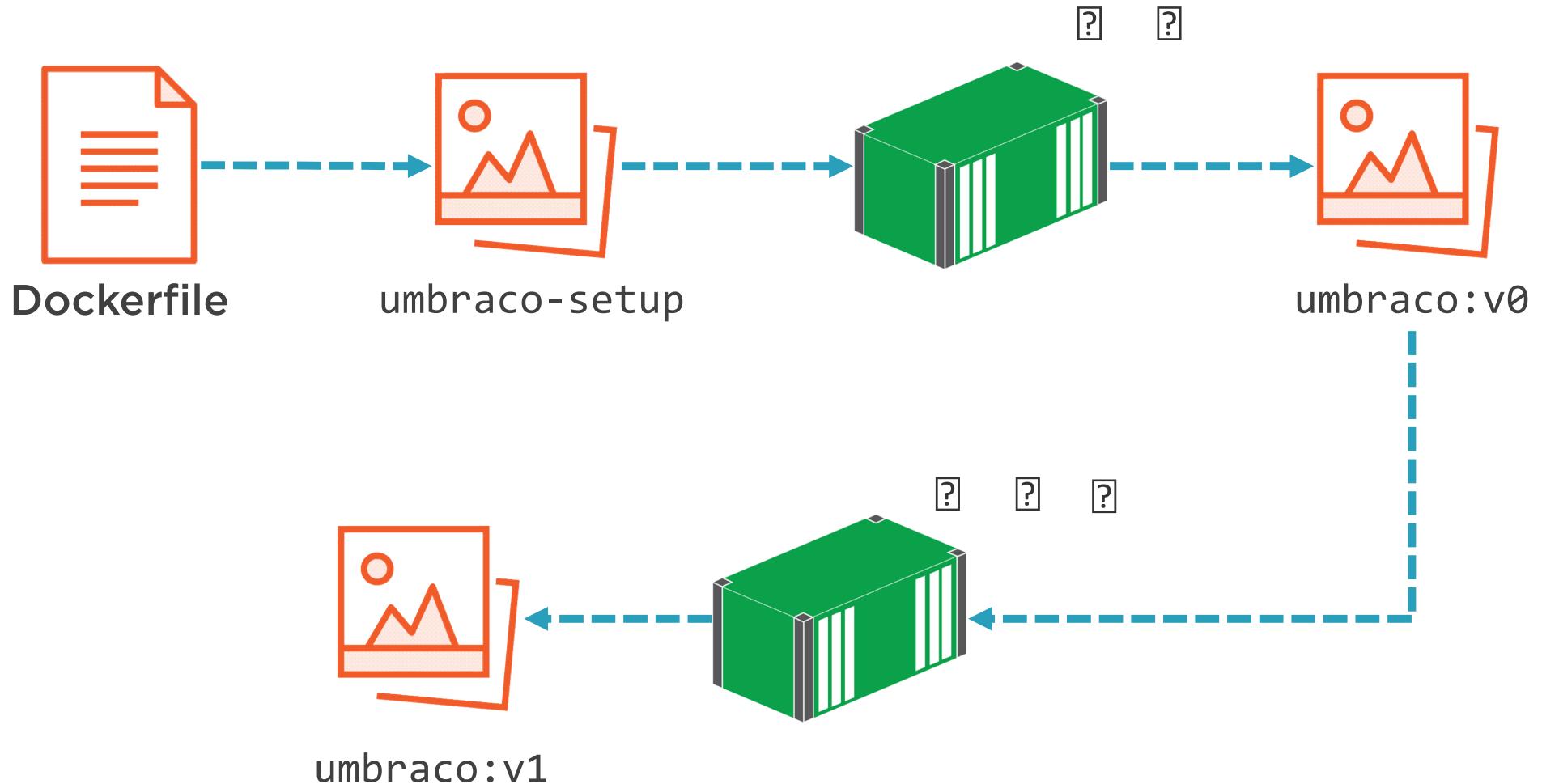
Run Umbraco in Docker
Create the homepage content for the app

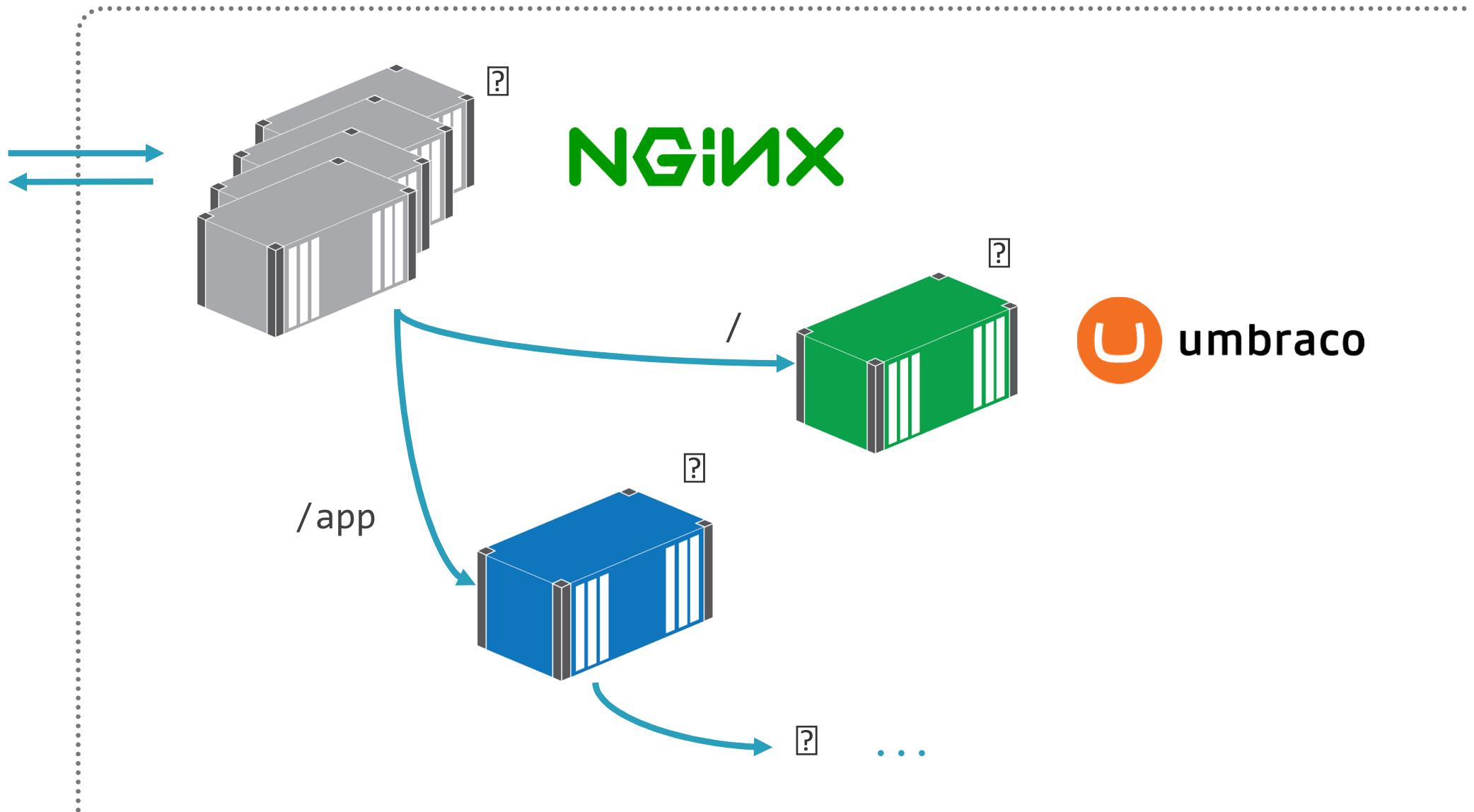
```
docker container stop umbraco
```

```
docker container commit umbraco umbraco:v1
```

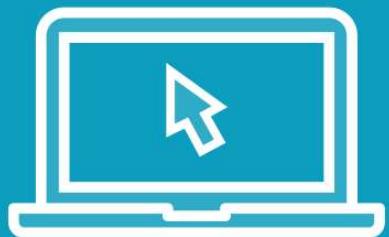
Commit Container to Image

Umbraco with the homepage content - umbraco:v1





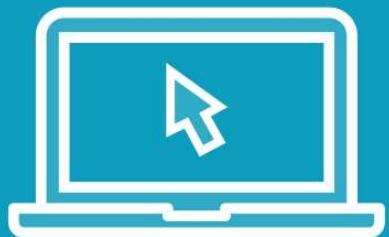
Demo



Configuring the Web Application

- Changing IIS deployment
- Set a custom path for the website
- Listening on `/app` instead of `/`

Demo



Running the Reverse Proxy

- Running Nginx in Docker
- Configuring routing rules
- Running the solution with the proxy

```
ENV NGINX_VERSION="1.12.2"

RUN Invoke-WebRequest -OutFile nginx.zip ...; `  
    Expand-Archive nginx.zip -DestinationPath C:\ ; `  
    Rename-Item "C:\nginx-$($env:NGINX_VERSION)" C:\nginx;
```

Nginx Installer Stage

Download and unzip installation archive

```
EXPOSE 80 443
```

```
WORKDIR C:\nginx
```

```
CMD ".\nginx"
```

```
COPY --from=installer C:\nginx\ .
```

```
COPY docker\nginx\conf .\conf
```

Nginx Application Stage

Configure ports and startup, copy binaries from installer

```
proxy_set_header Host $host;  
proxy_pass_request_headers on;  
server {  
    server_name _;  
    listen 80 default_server;
```

Nginx Configuration File

Default server listens on port 80

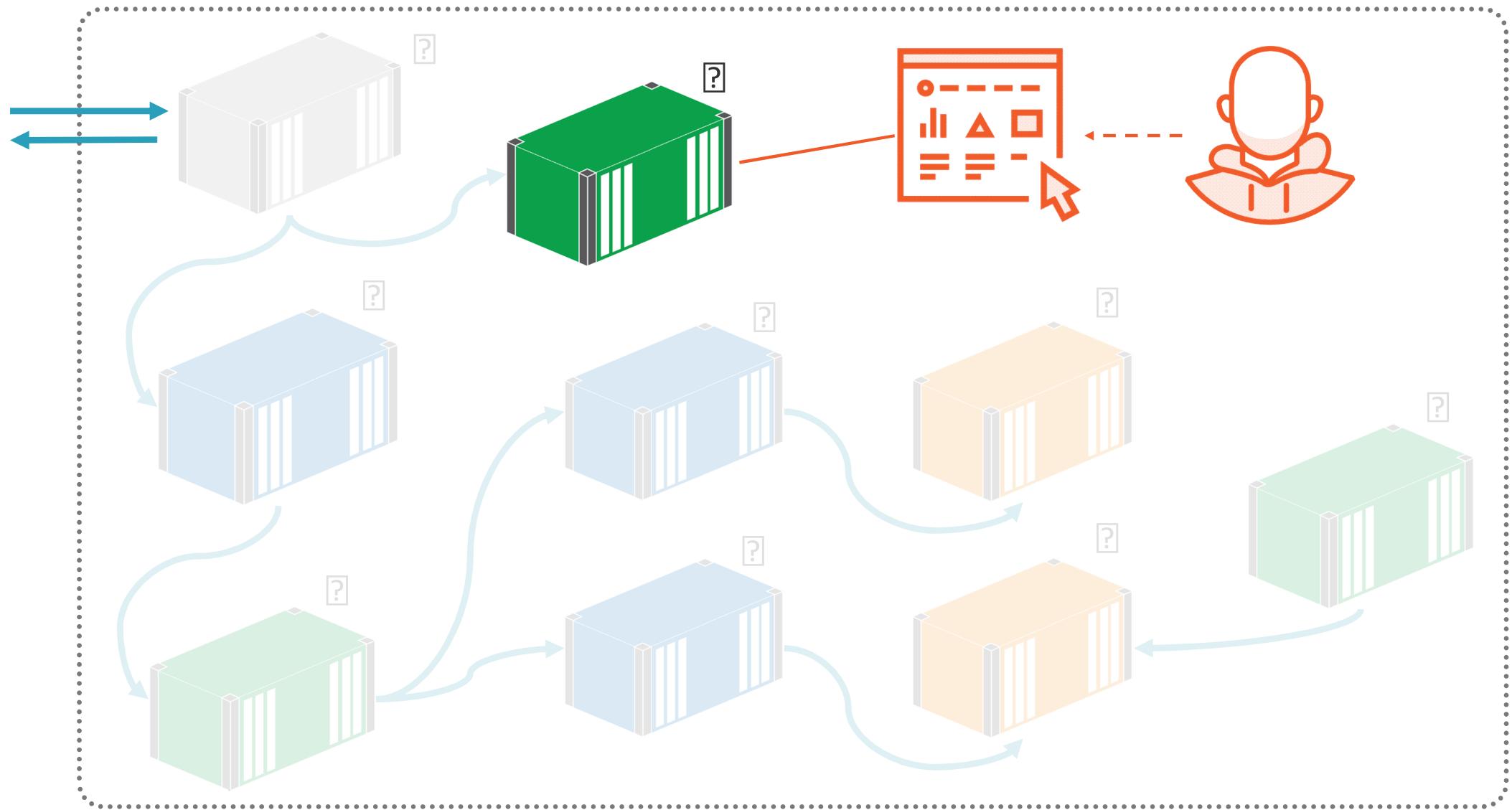
```
location / {  
    proxy_pass http://homepage/; }  
  
location /app/ {  
    proxy_pass http://web-app; }
```

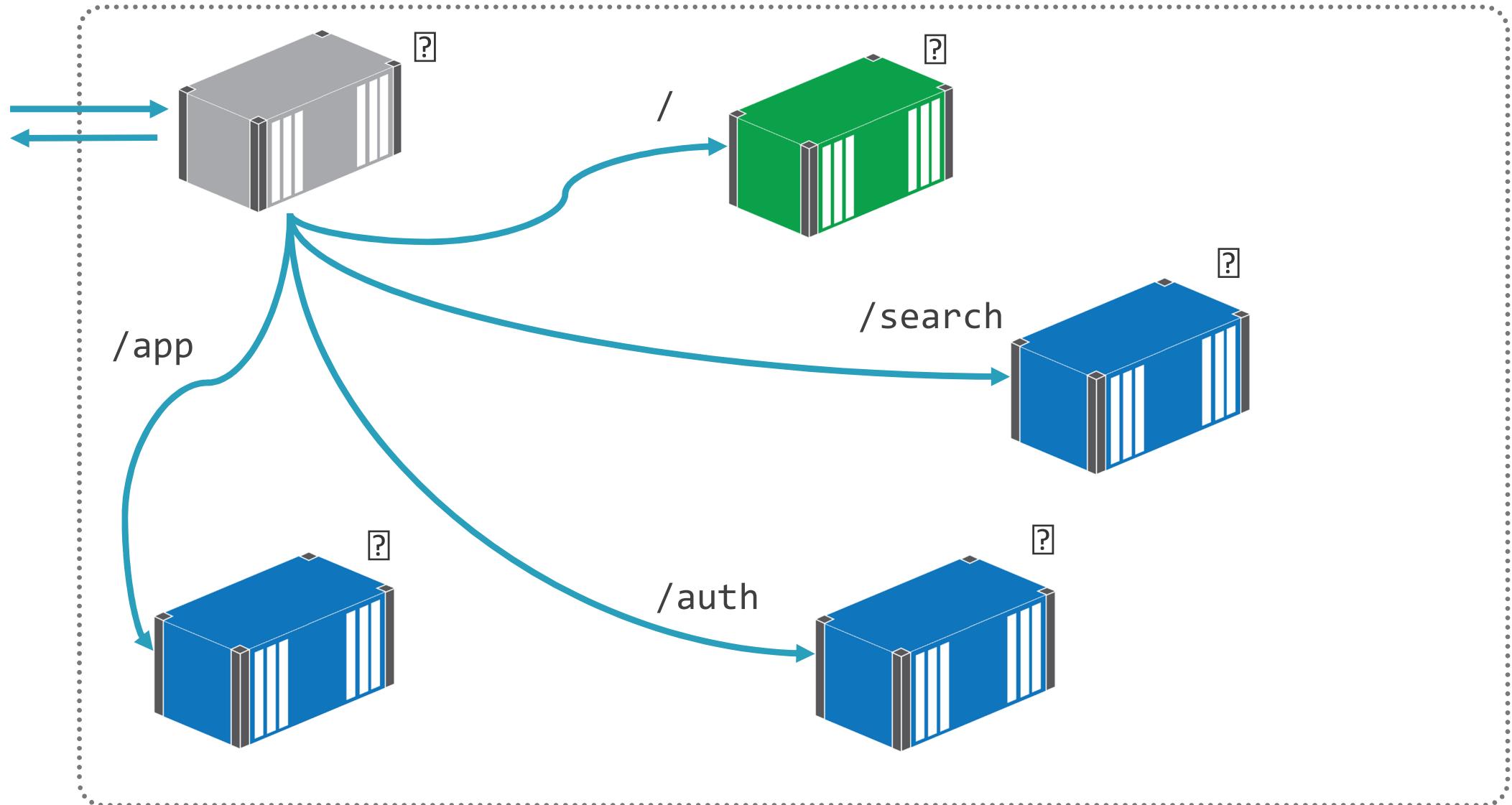
Nginx Configuration File
Location blocks route traffic to containers

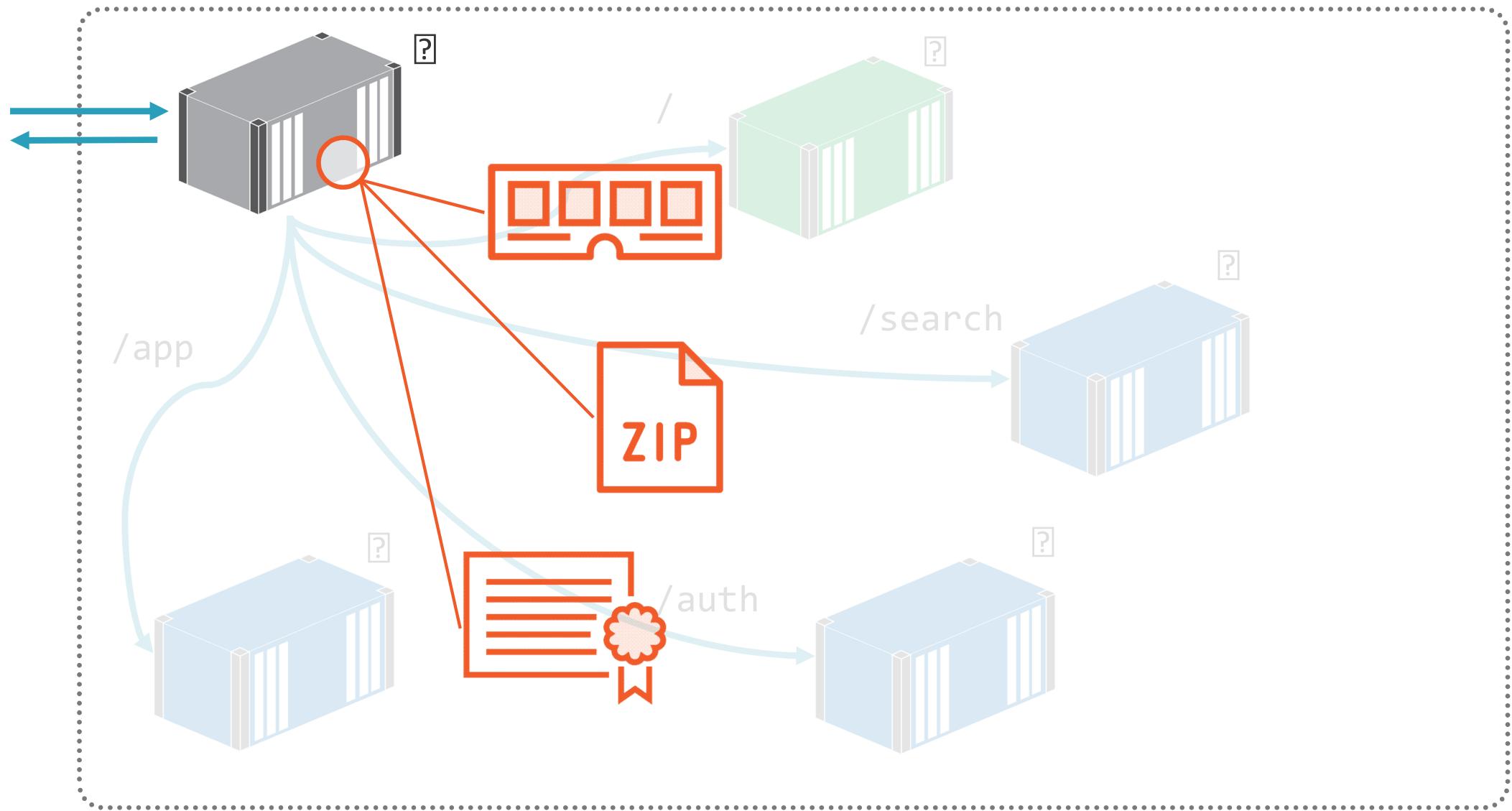
```
RUN Remove-Website -Name 'Default Web Site';`  
New-Website -Name 'web-app' -Port 80 `  
New-WebApplication -Name 'app' -Site 'web-app' ...
```

Web Application Dockerfile

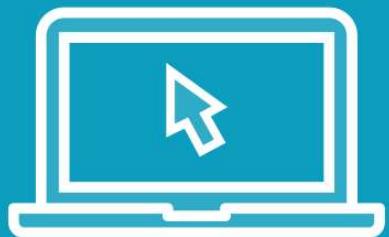
Configure website at /app path







Demo



Extending the Proxy Container

- Adding GZip compression
- Configuring client-side caching
- Enabling server-side caching

```
gzip on;  
gzip_proxied any;
```

Extended Nginx Configuration **Response compression with GZip**

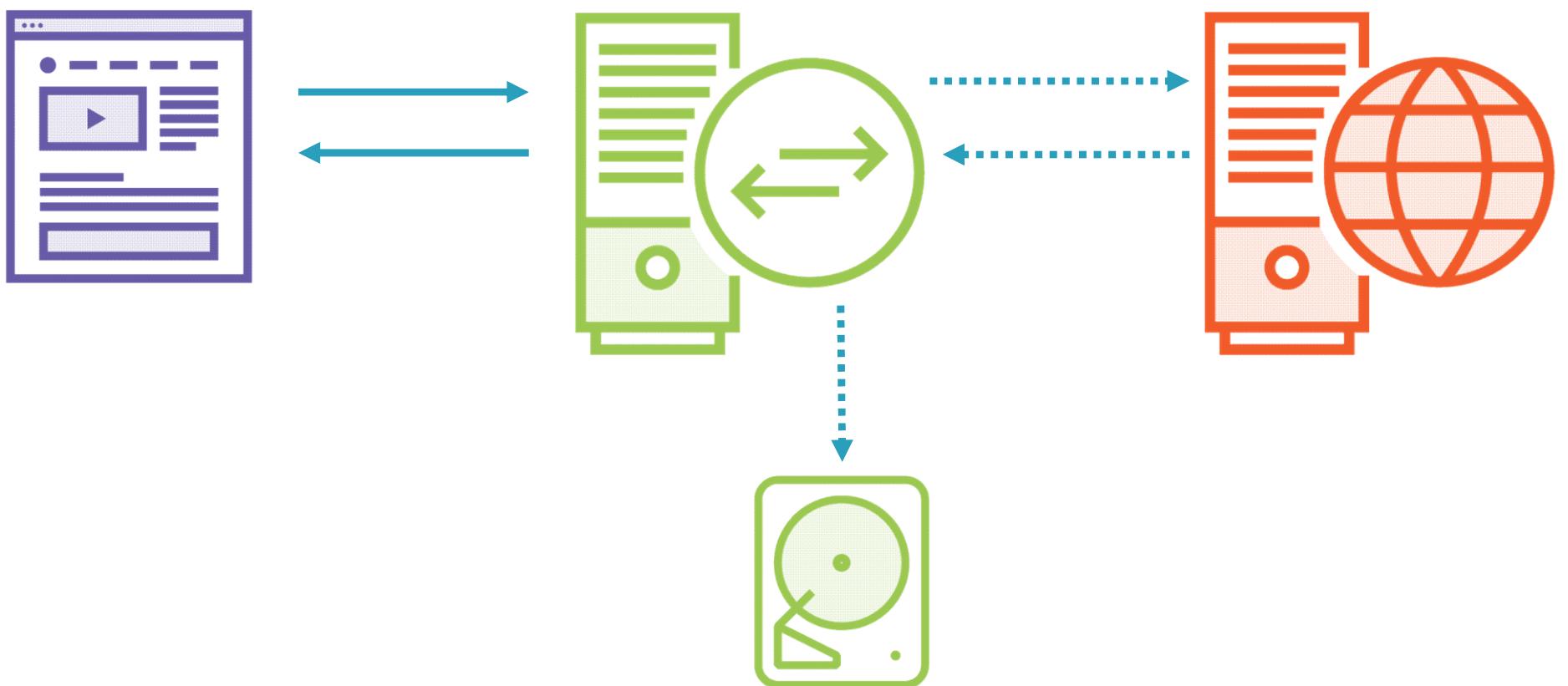
```
map $sent_http_content_type $expires {  
    default off;  
    ~image/ 1M;  
}  
  
expires $expires;
```

Extended Nginx Configuration **Expiration caching for images**

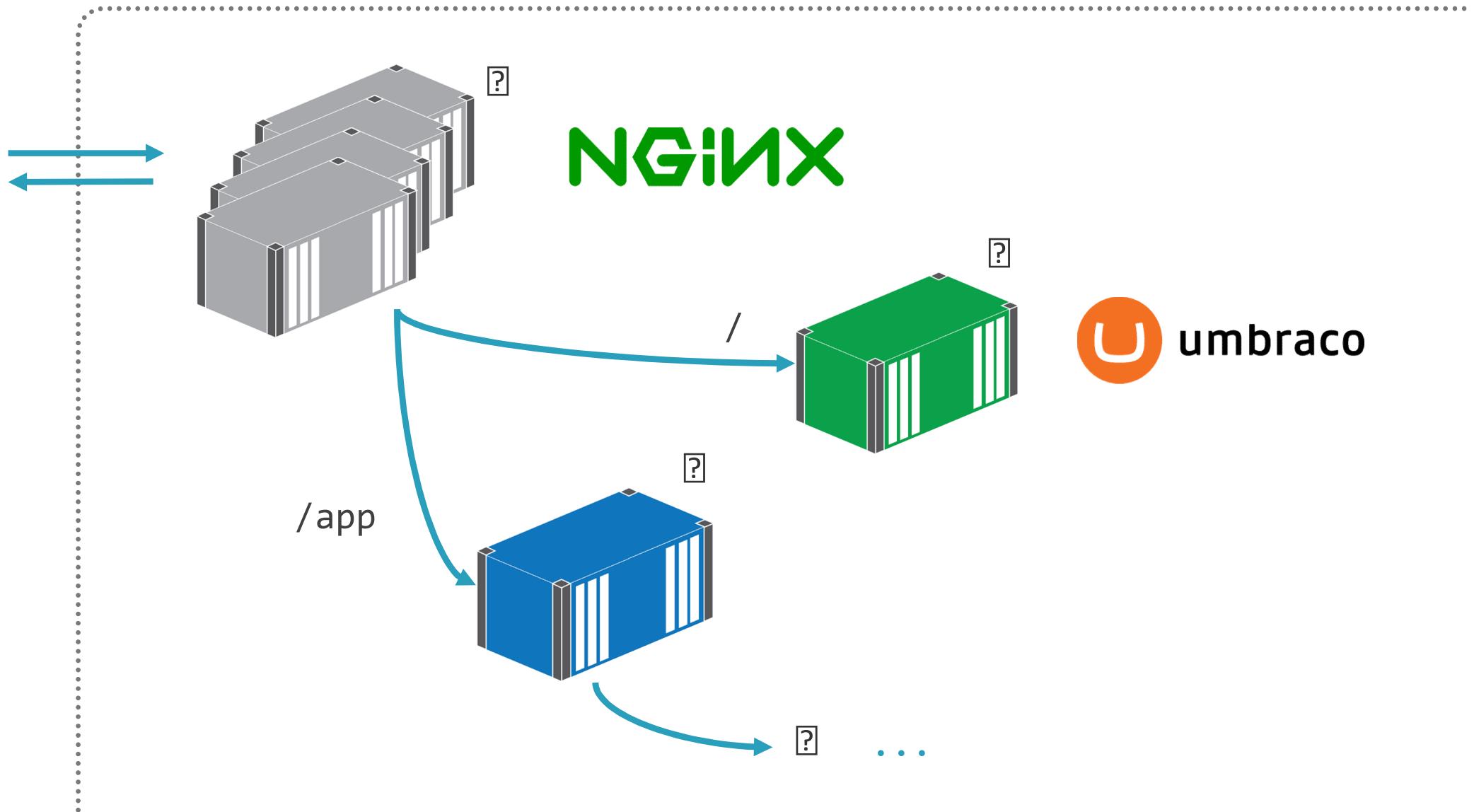
```
proxy_cache_path /nginx/cache inactive=24h max_size=100m;  
...  
proxy_cache_valid 200 1m;  
proxy_ignore_headers Expires Cache-Control;
```

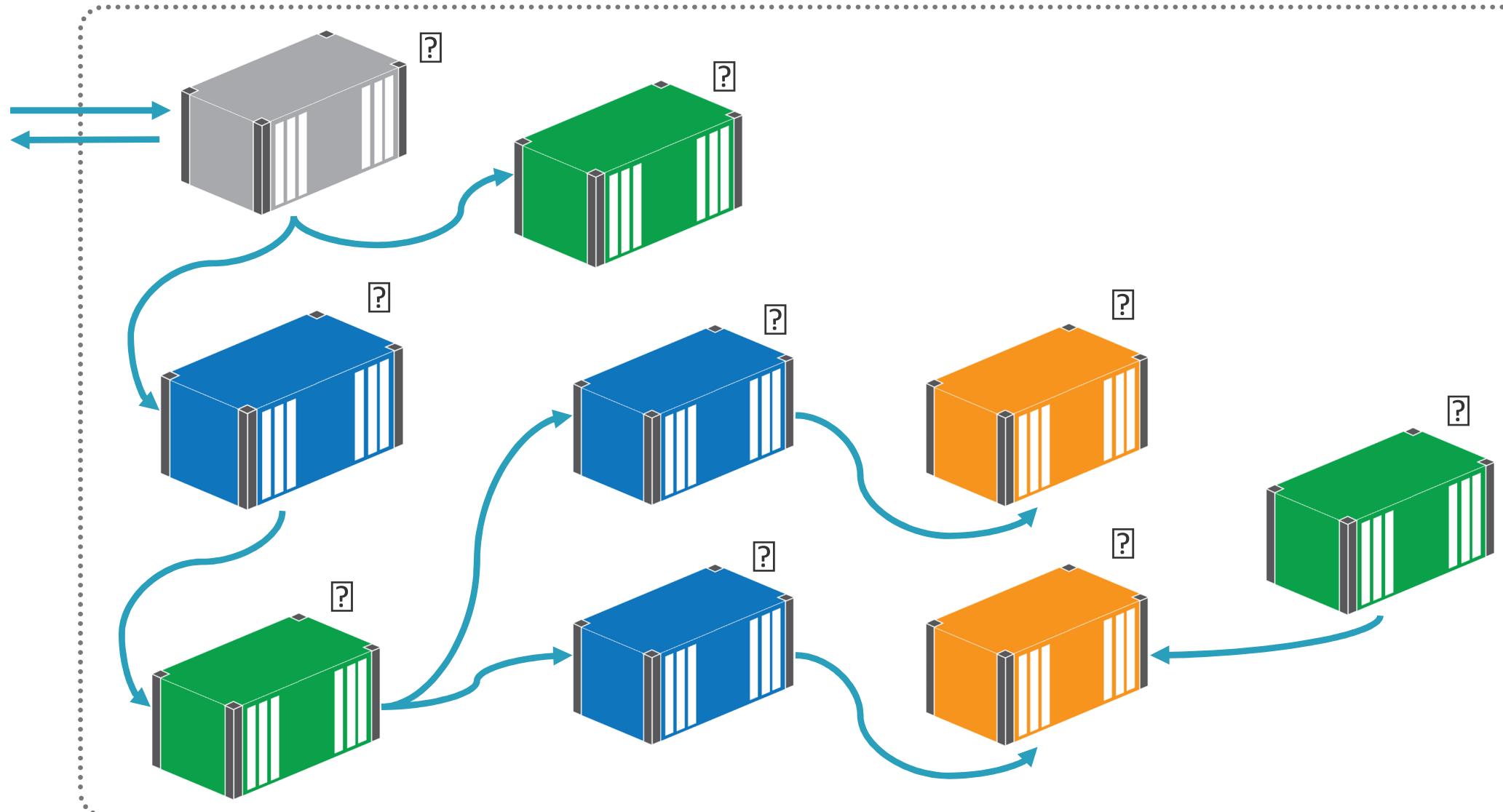
Extended Nginx Configuration

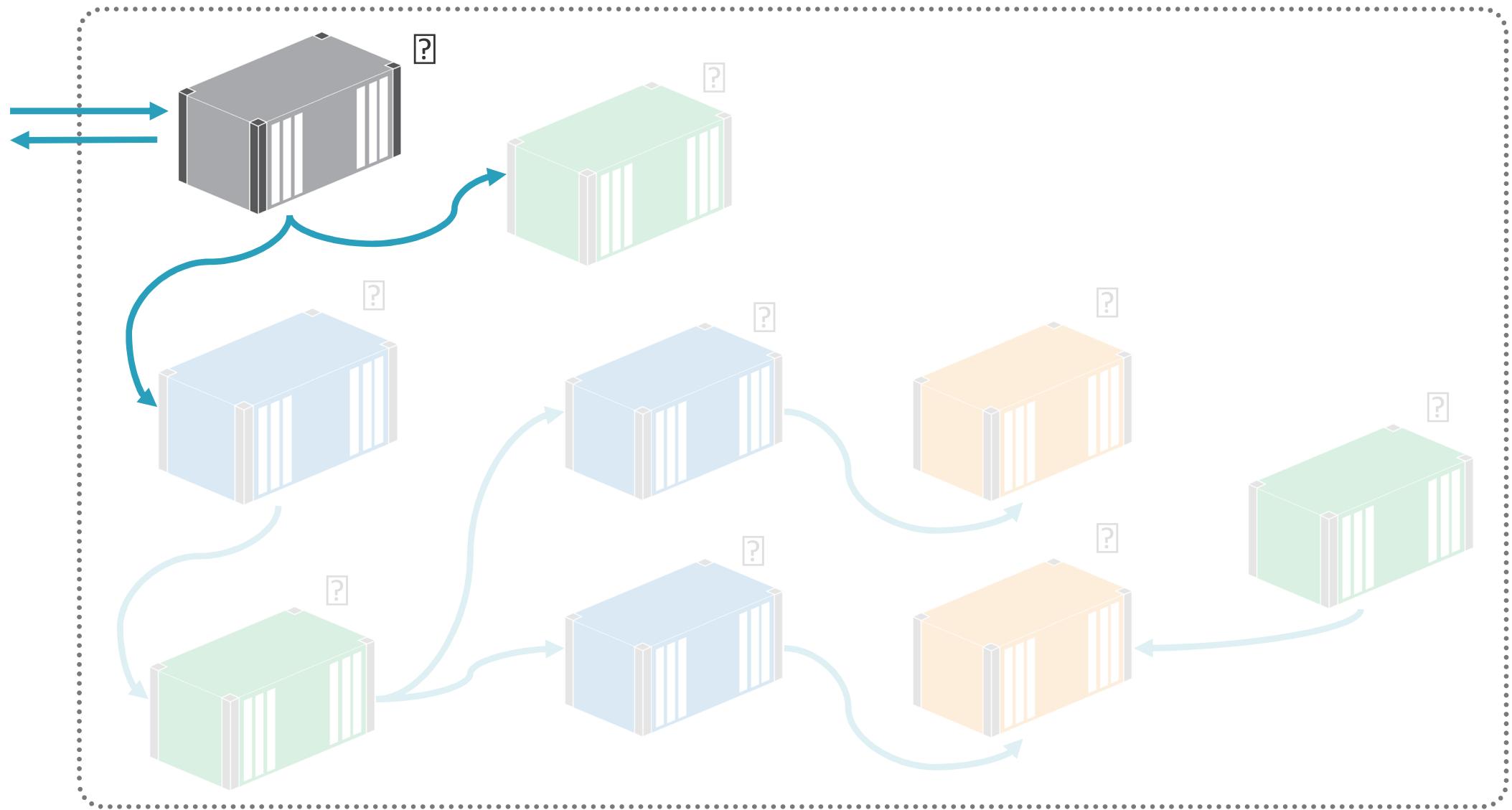
Caching upstream responses in the proxy

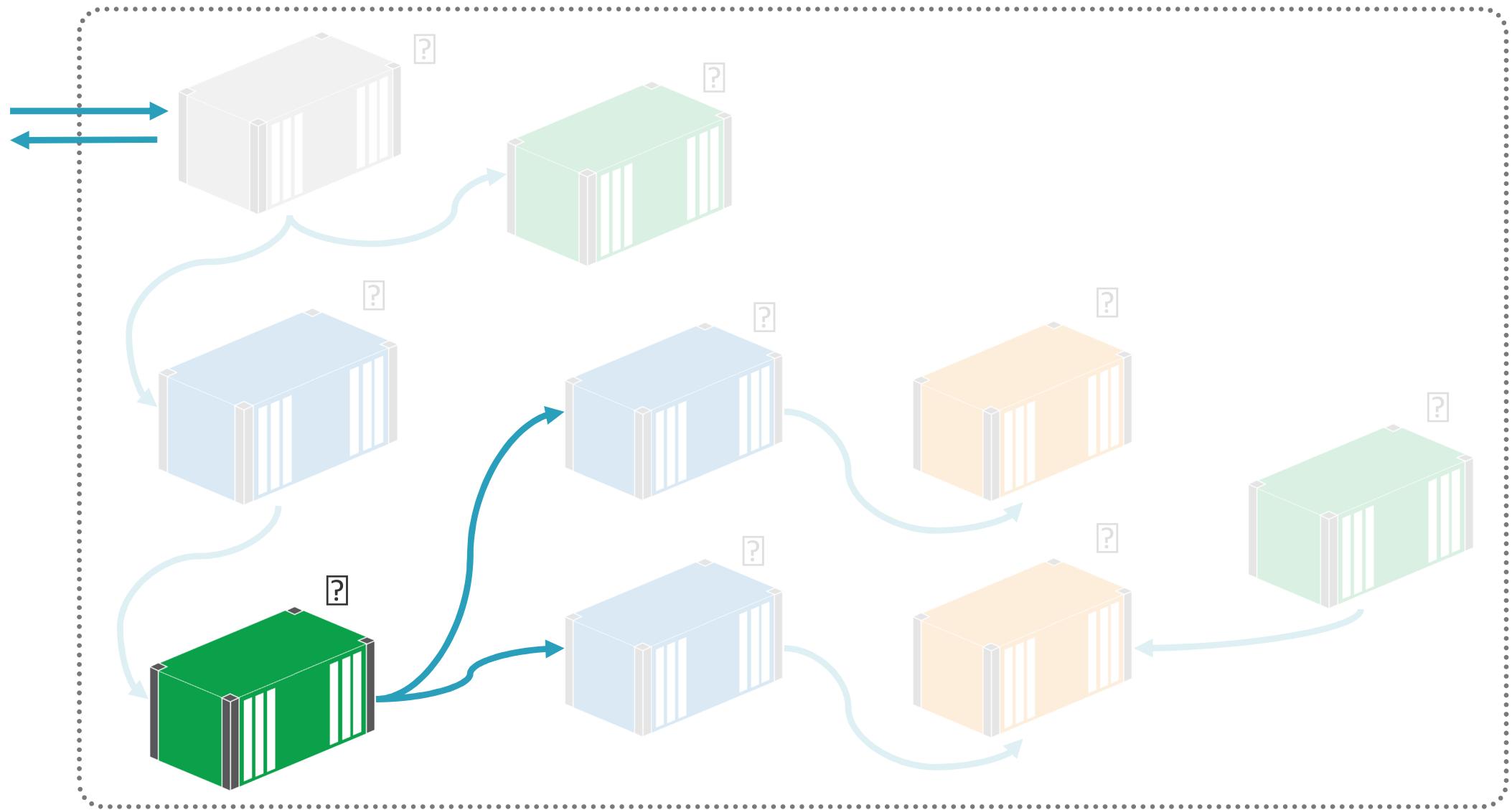


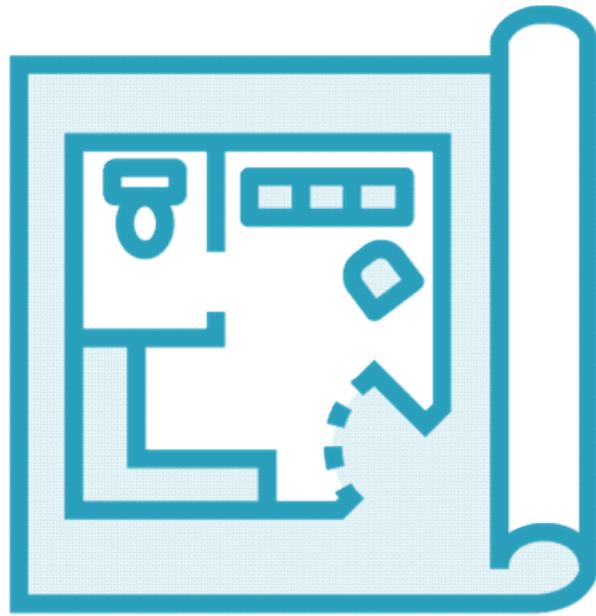












Container-first Design

- Solving business problems
- Evolving the architecture
- Enabling further evolution

Coming Next



Monitoring and Management

- Management with Docker Compose
- Adding metrics with Prometheus
- Monitoring with Grafana