

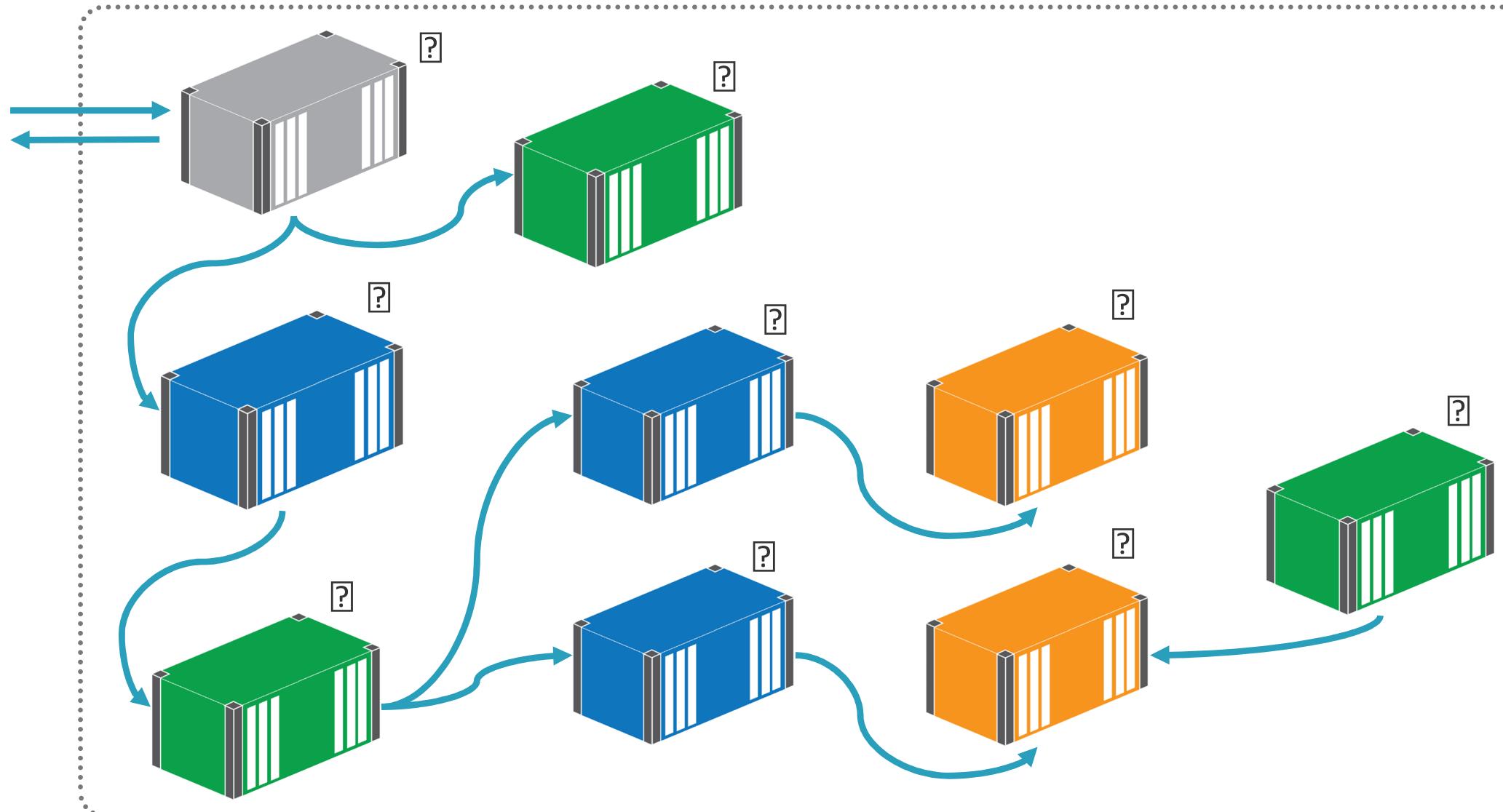
# Managing and Monitoring Multi-container Solutions

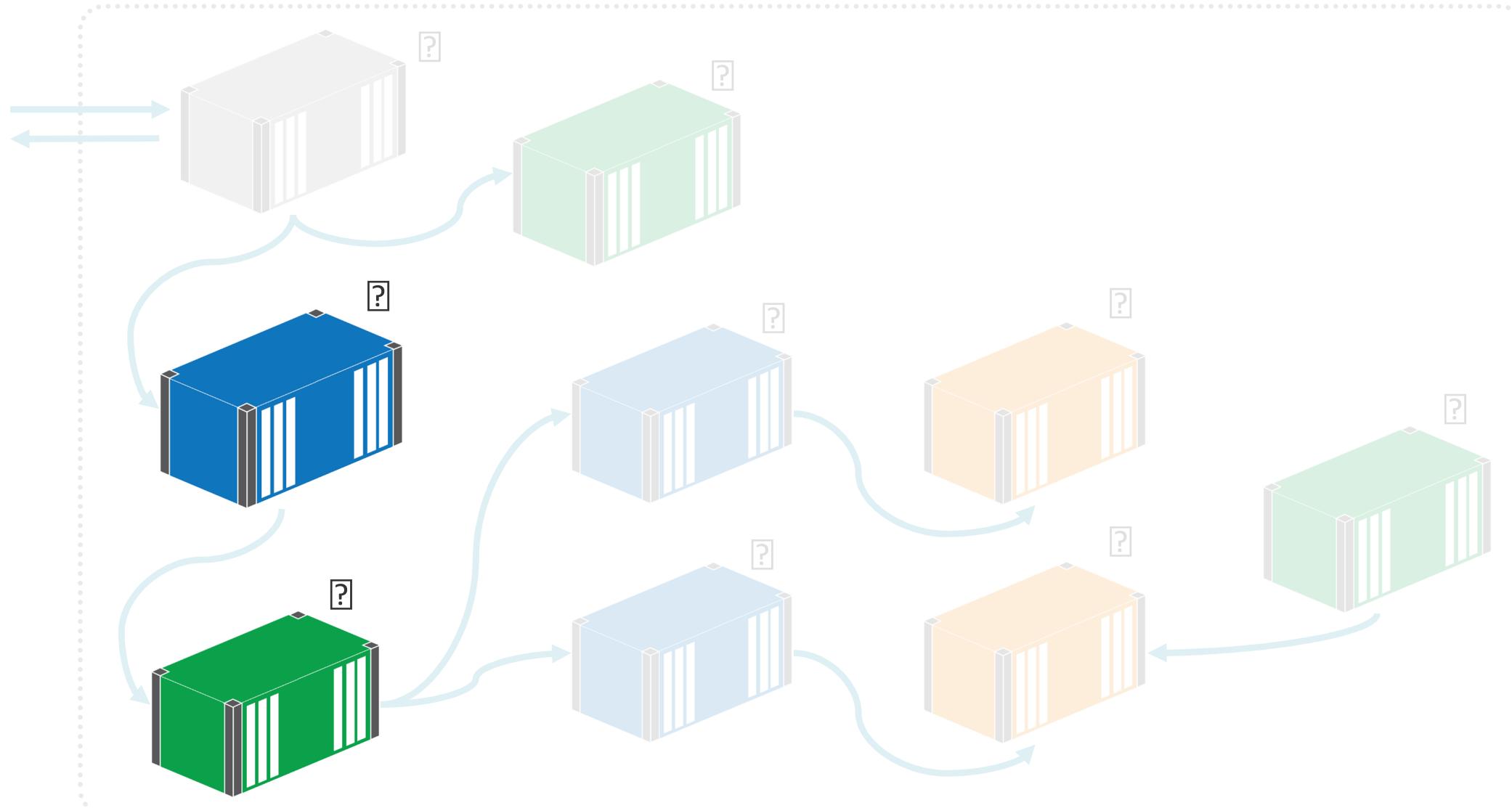
---

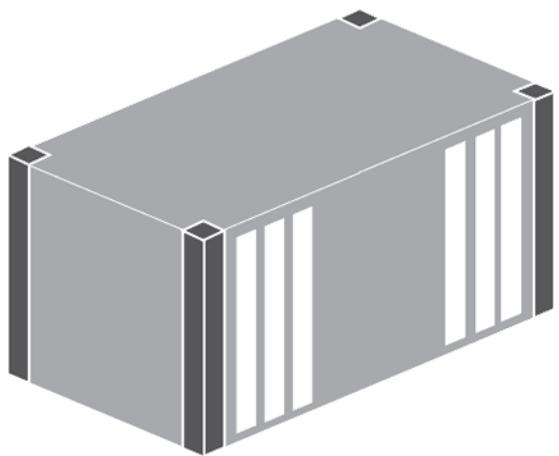


**Elton Stoneman**  
DEVELOPER ADVOCATE

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

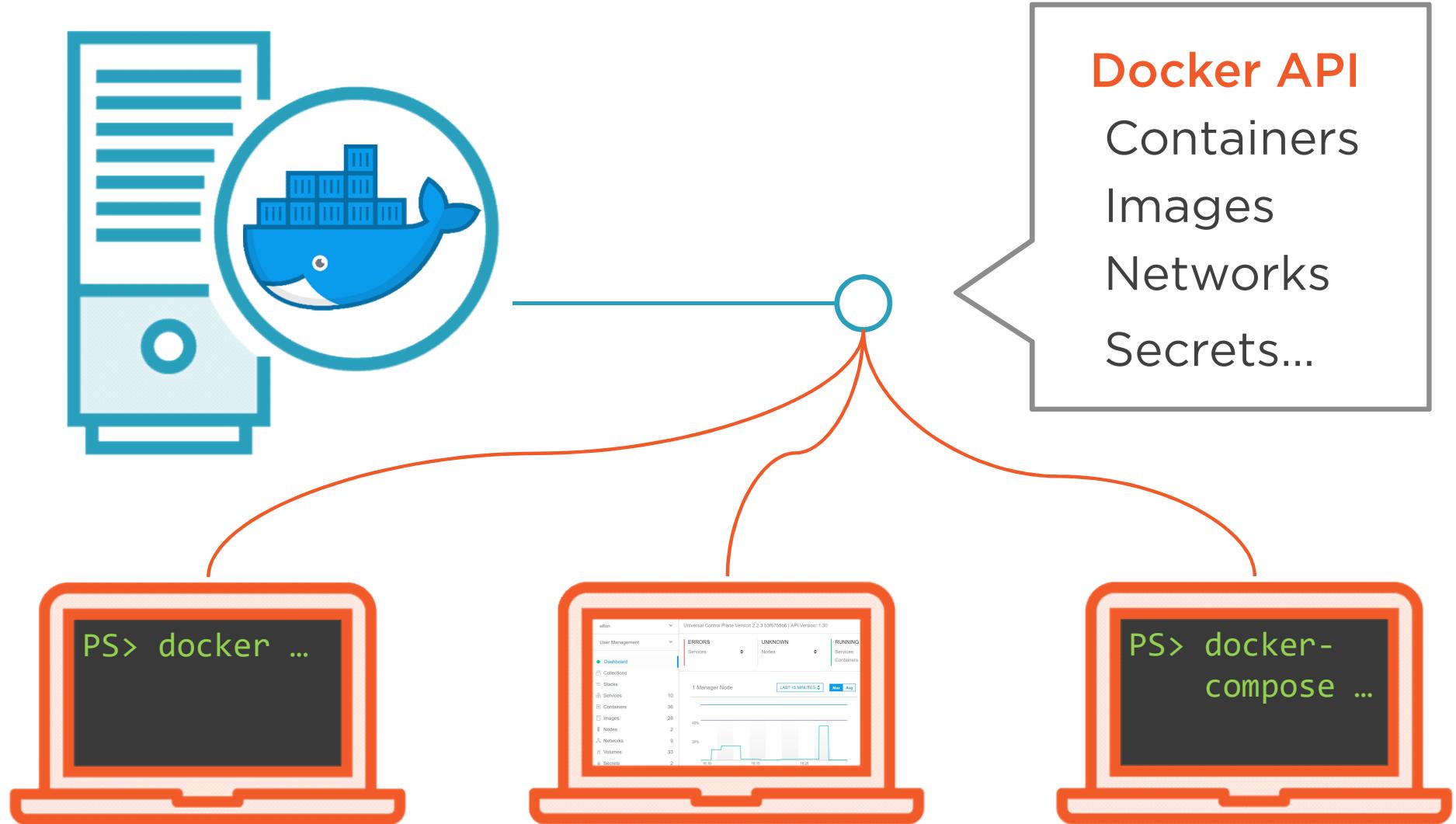


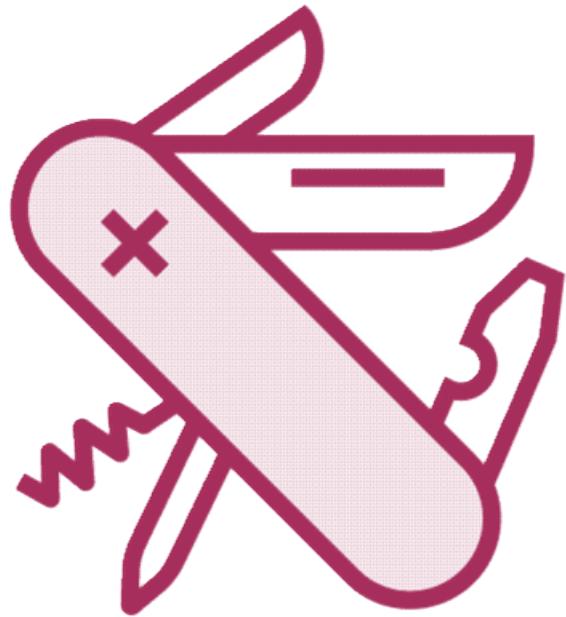




### **Container API**

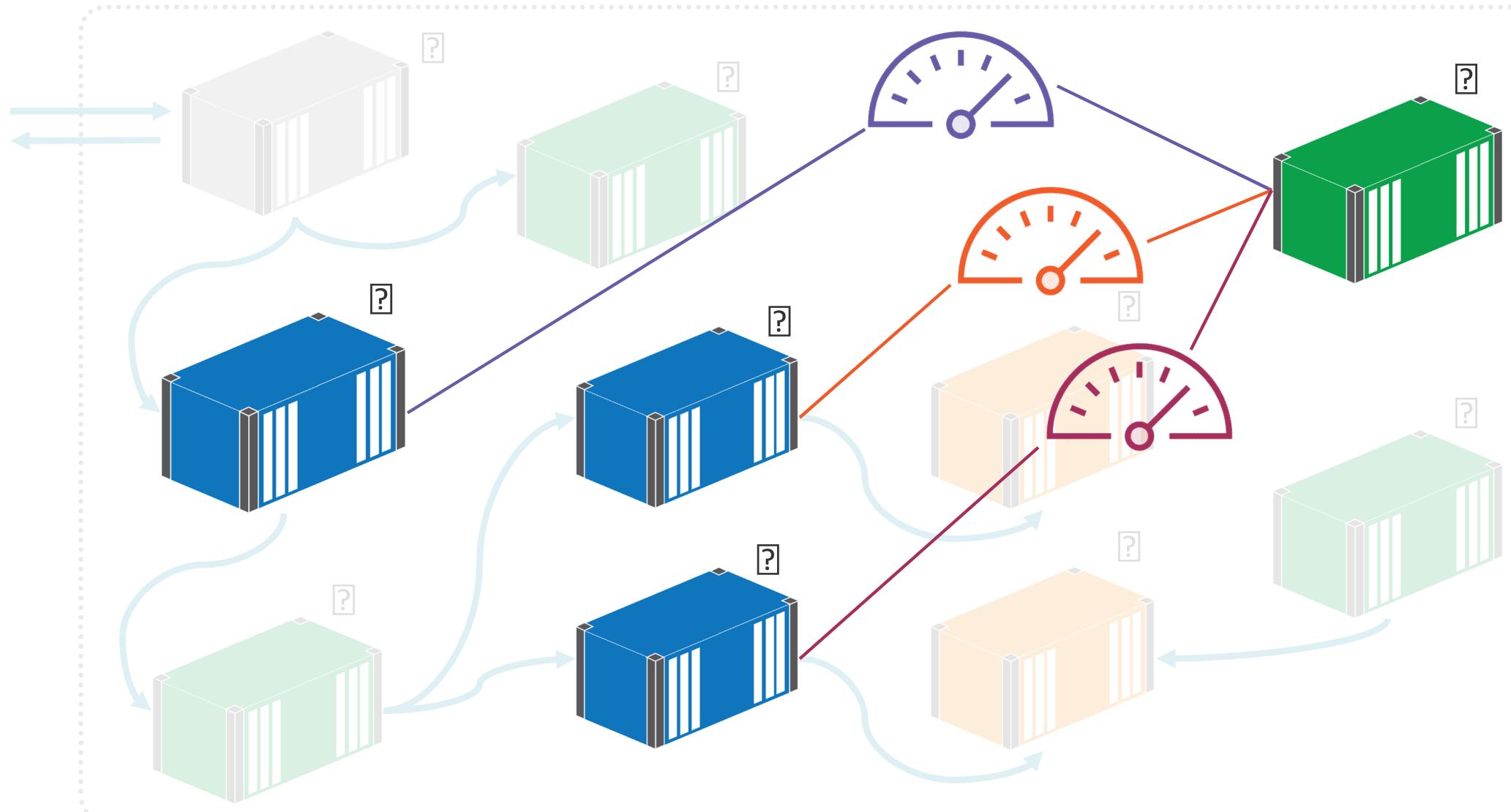
- Log entries
- Process list
- Start / stop
- Inspect

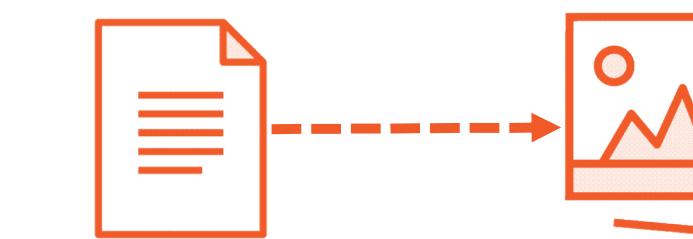




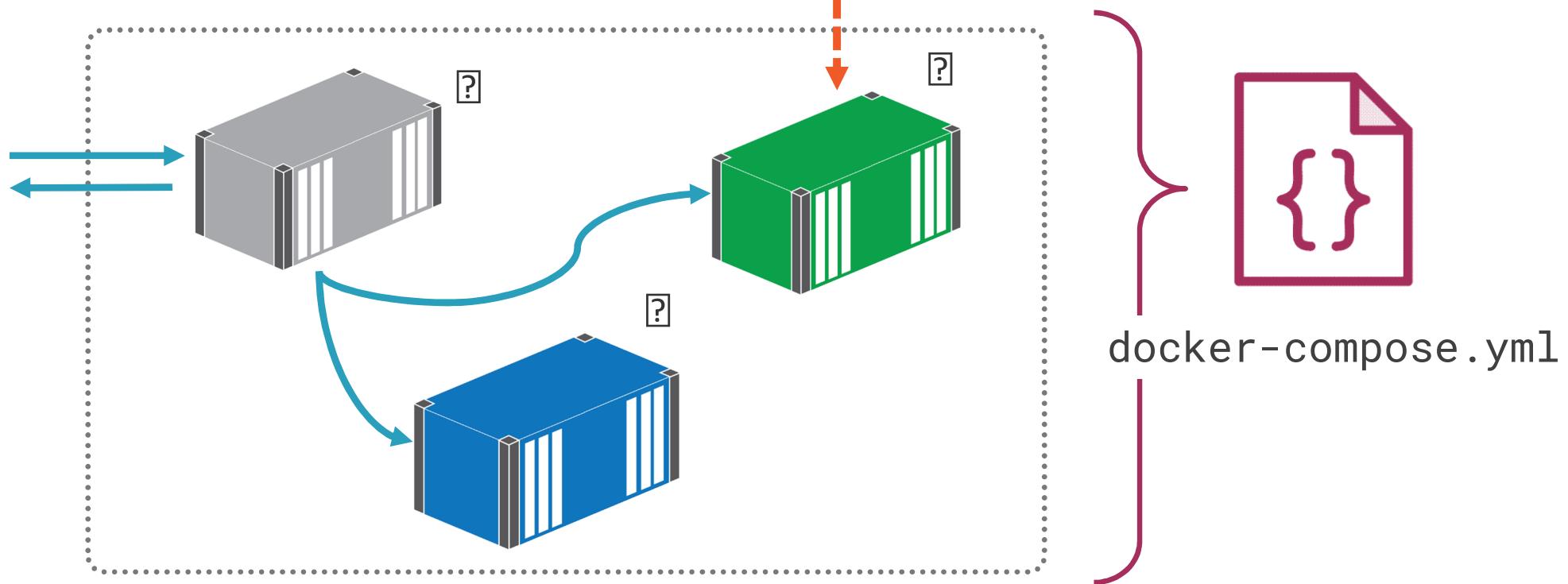
## Docker Compose

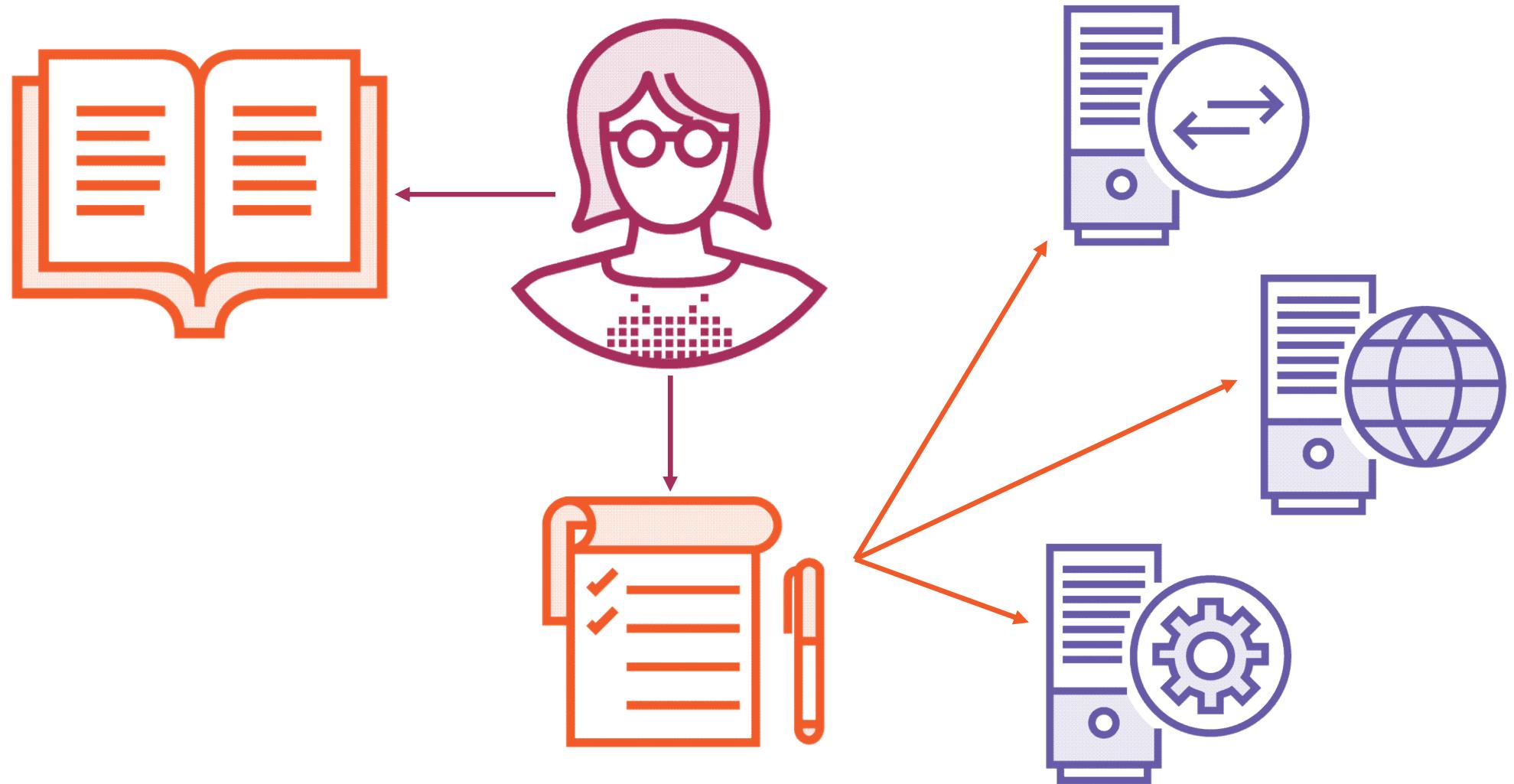
- Define multi-container solutions
- Manage as a single unit
- Stop / start / scale / build
- Single servers & clusters

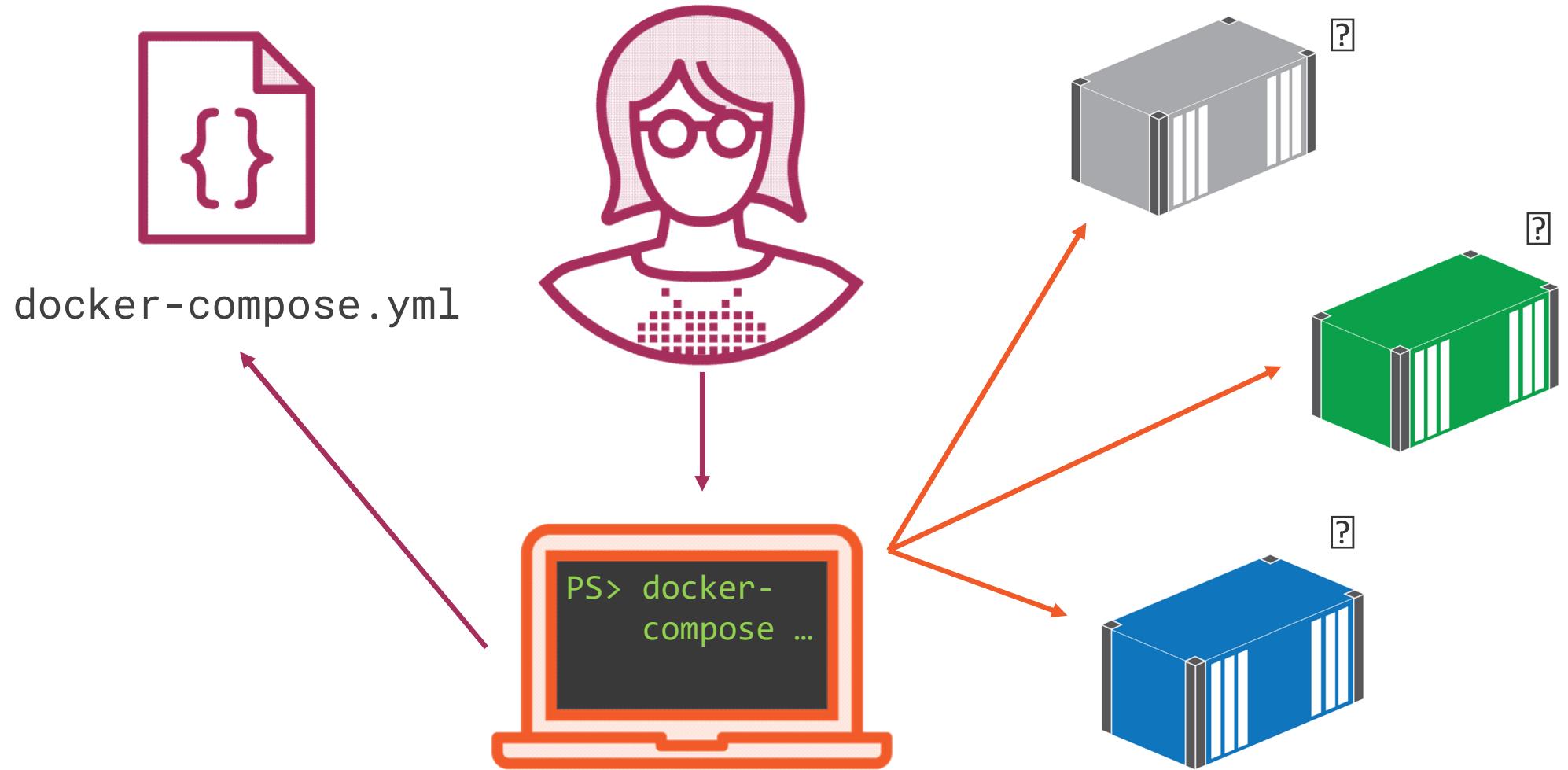


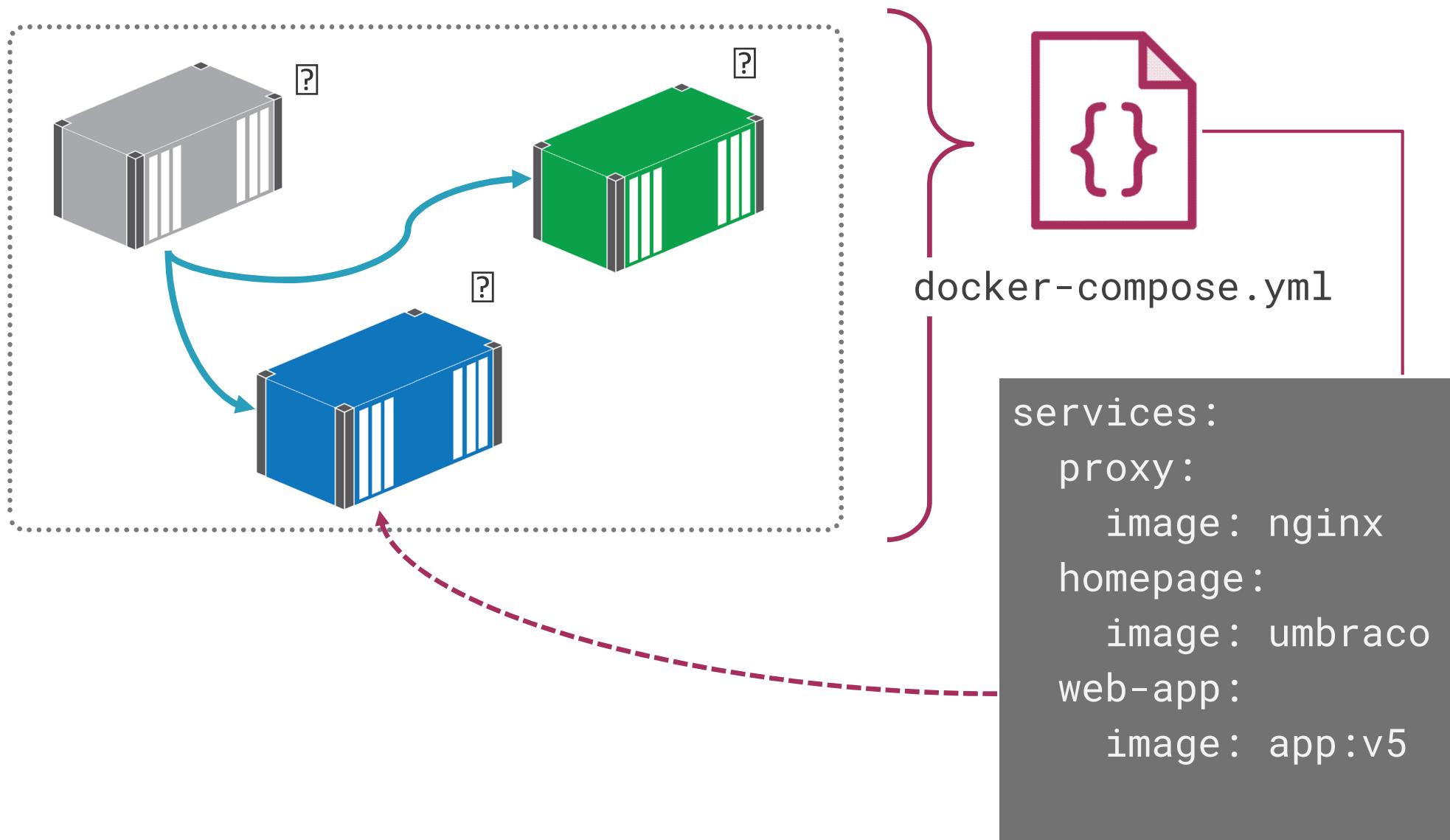


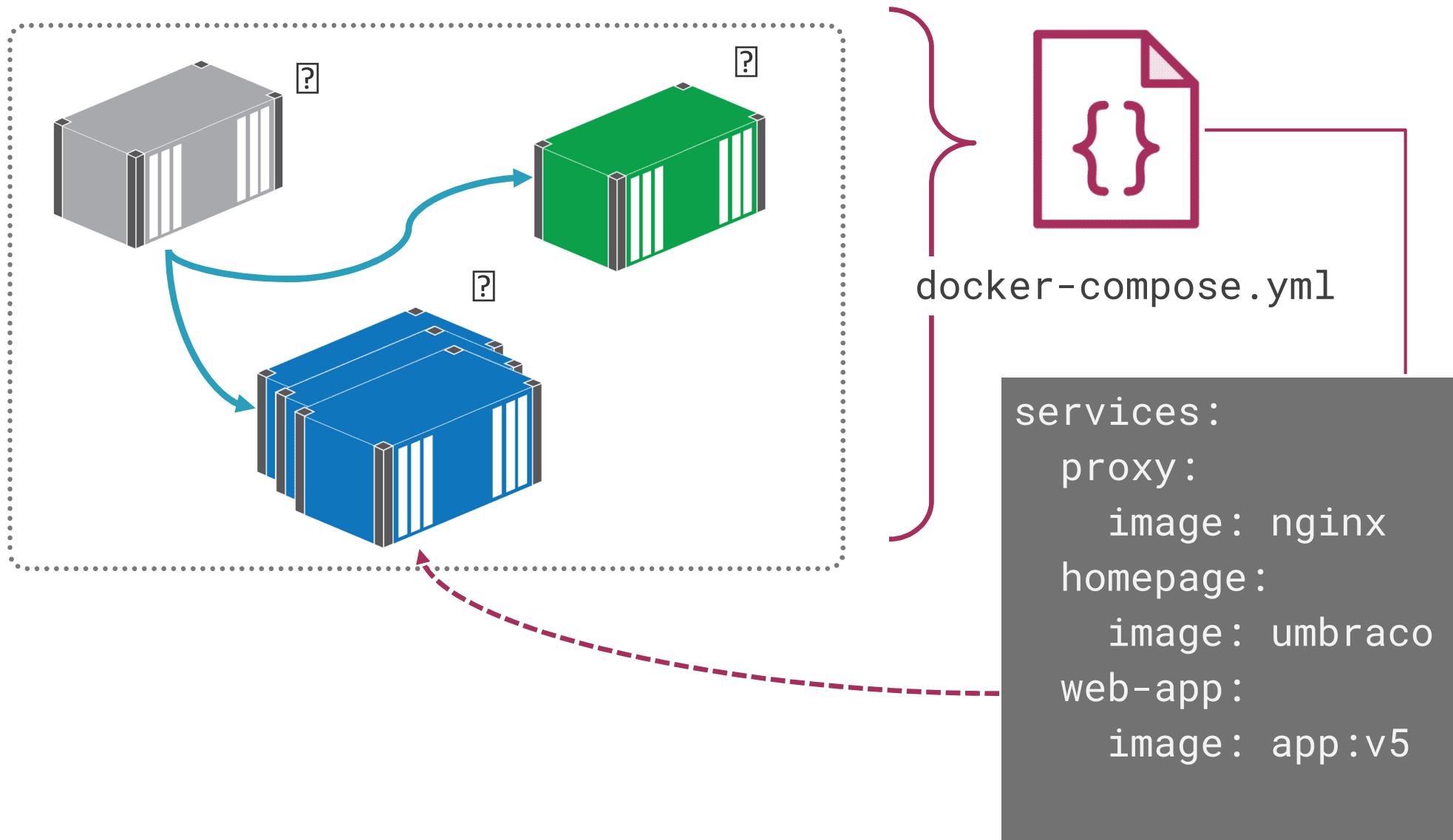
Dockerfile

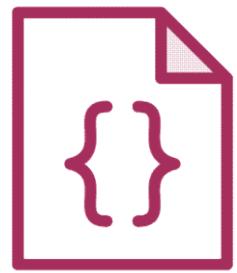




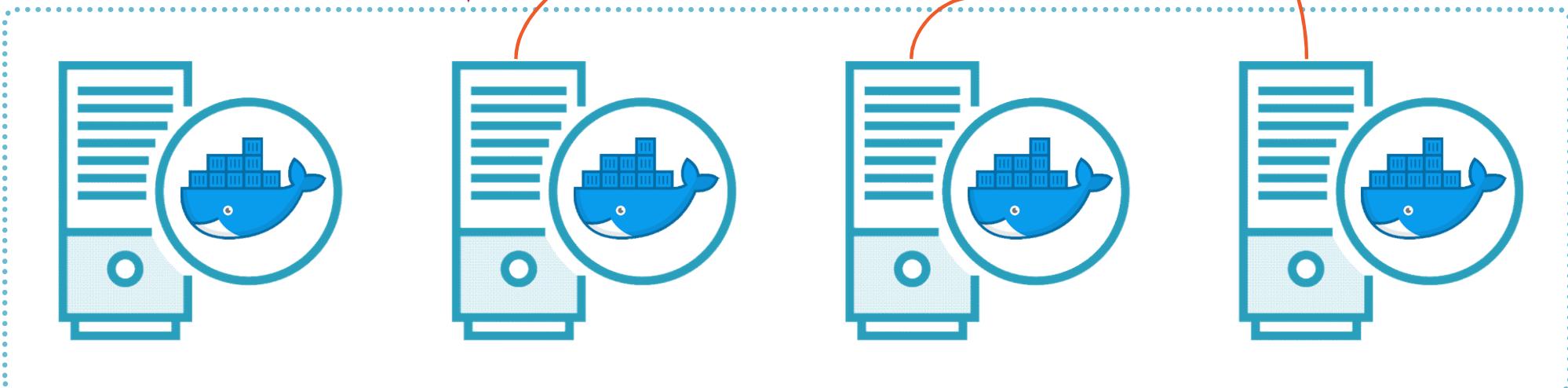
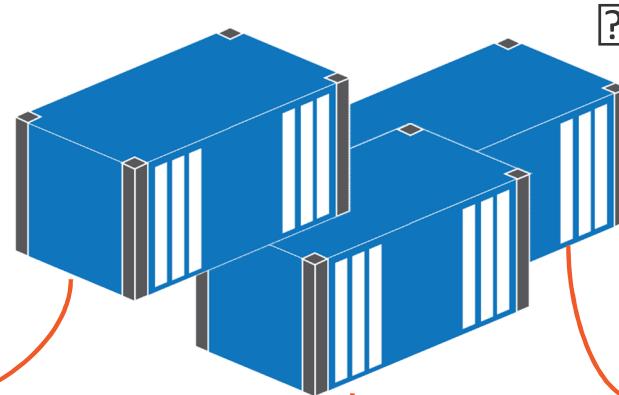


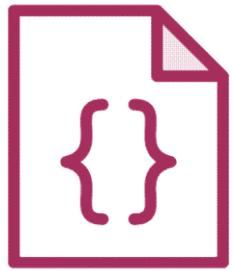




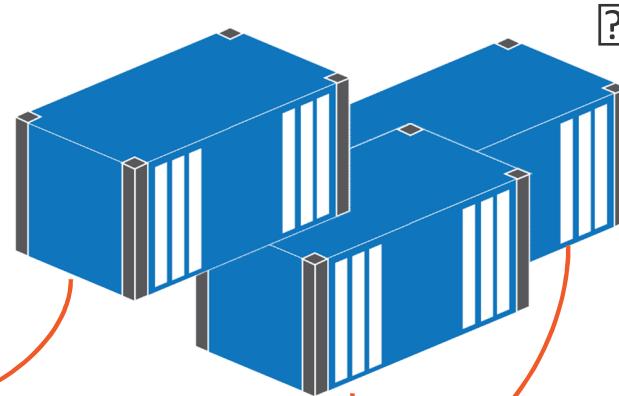


docker-compose.yml

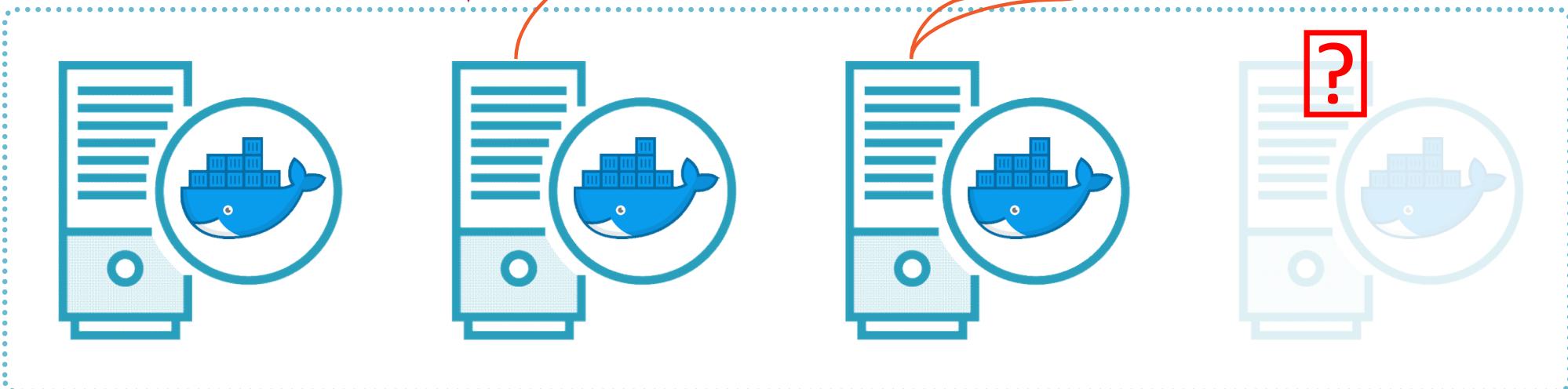


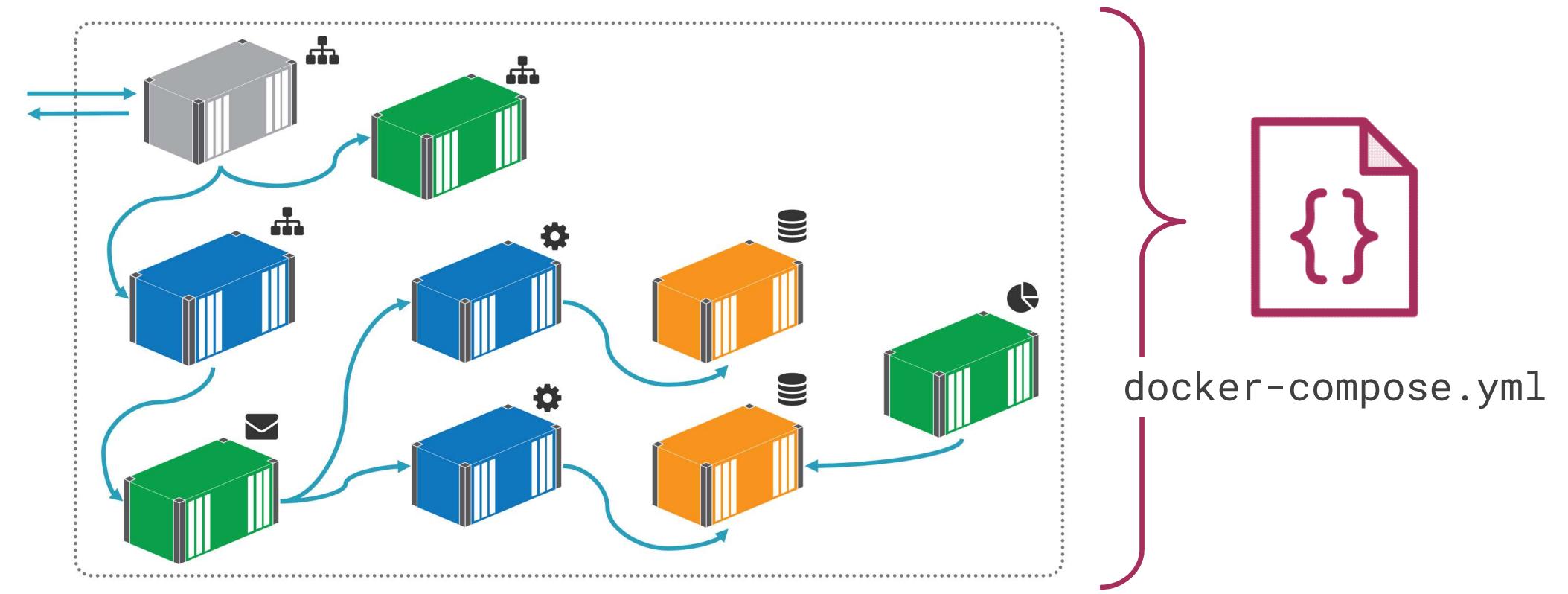


docker-compose.yml

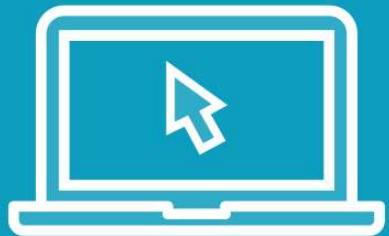


?





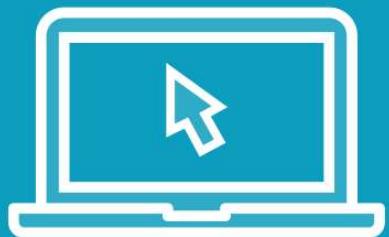
# Demo



## Managing Apps with Docker Compose

- Docker Compose YAML file
- `docker-compose` command line
- Running the solution

# Demo



## Going Further with Docker Compose

- Building application images
- Scaling services
- Managing containers

```
services:
```

```
  webinar-db:
```

```
    image: webinar-db:v2
```

```
networks:
```

```
  - app-net
```

Docker Compose YAML

**Basic service definition - image and networks**

```
elasticsearch:
```

```
  image: elasticsearch
```

```
  environment:
```

- ES\_JAVA\_OPTS=-Xms512m -Xmx512m

Docker Compose YAML

**Extended service definition – environment variables**

```
nginx:  
  image: nginx  
  ports:  
    - "80:80"  
depends_on:  
  - web-app
```

Docker Compose YAML  
**Extended service definition – ports and dependencies**

```
docker-compose up -d
```

```
docker-compose up --scale save-handler=3
```

Docker Compose Command Line

**Start and scale services**

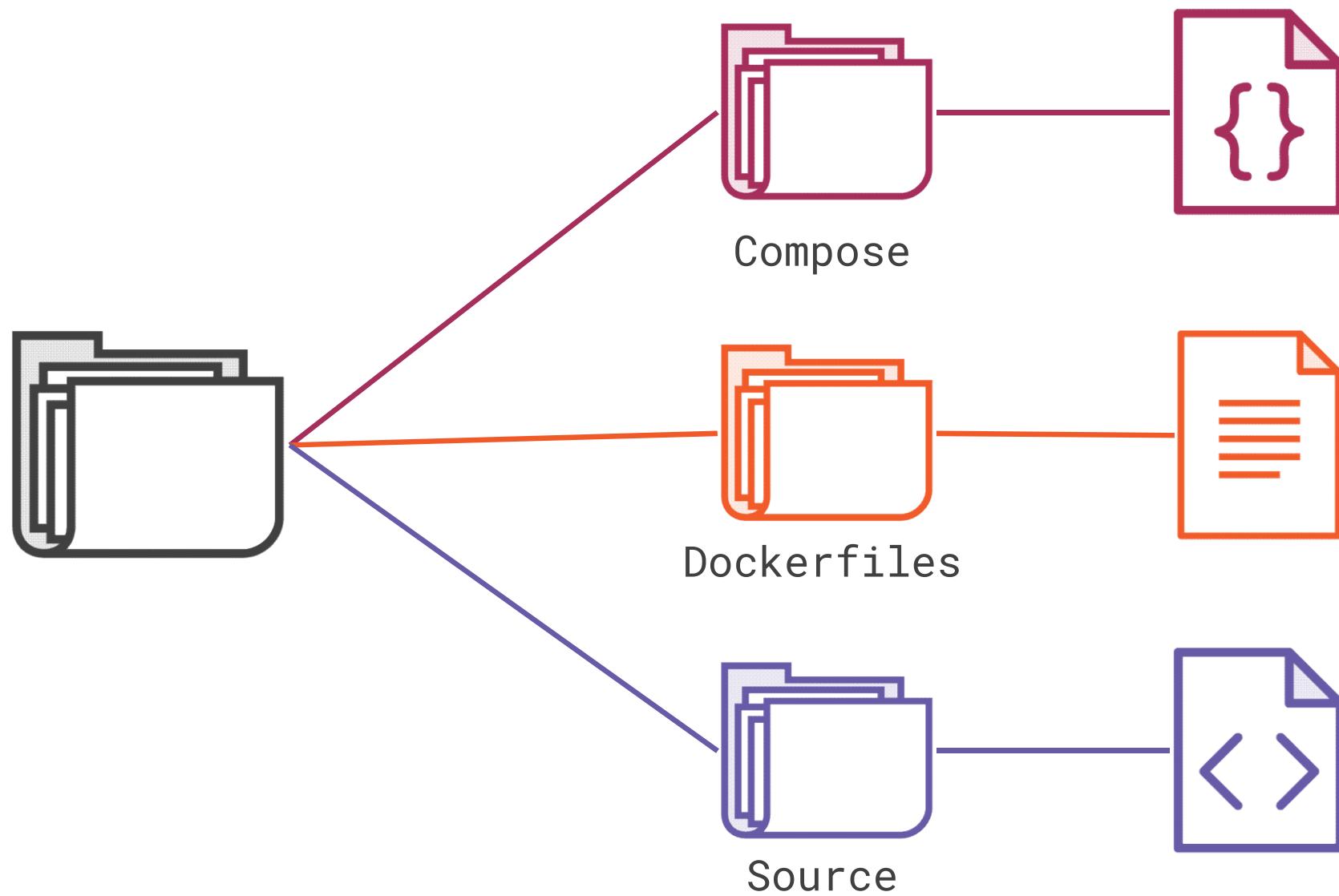
`docker-compose stop`

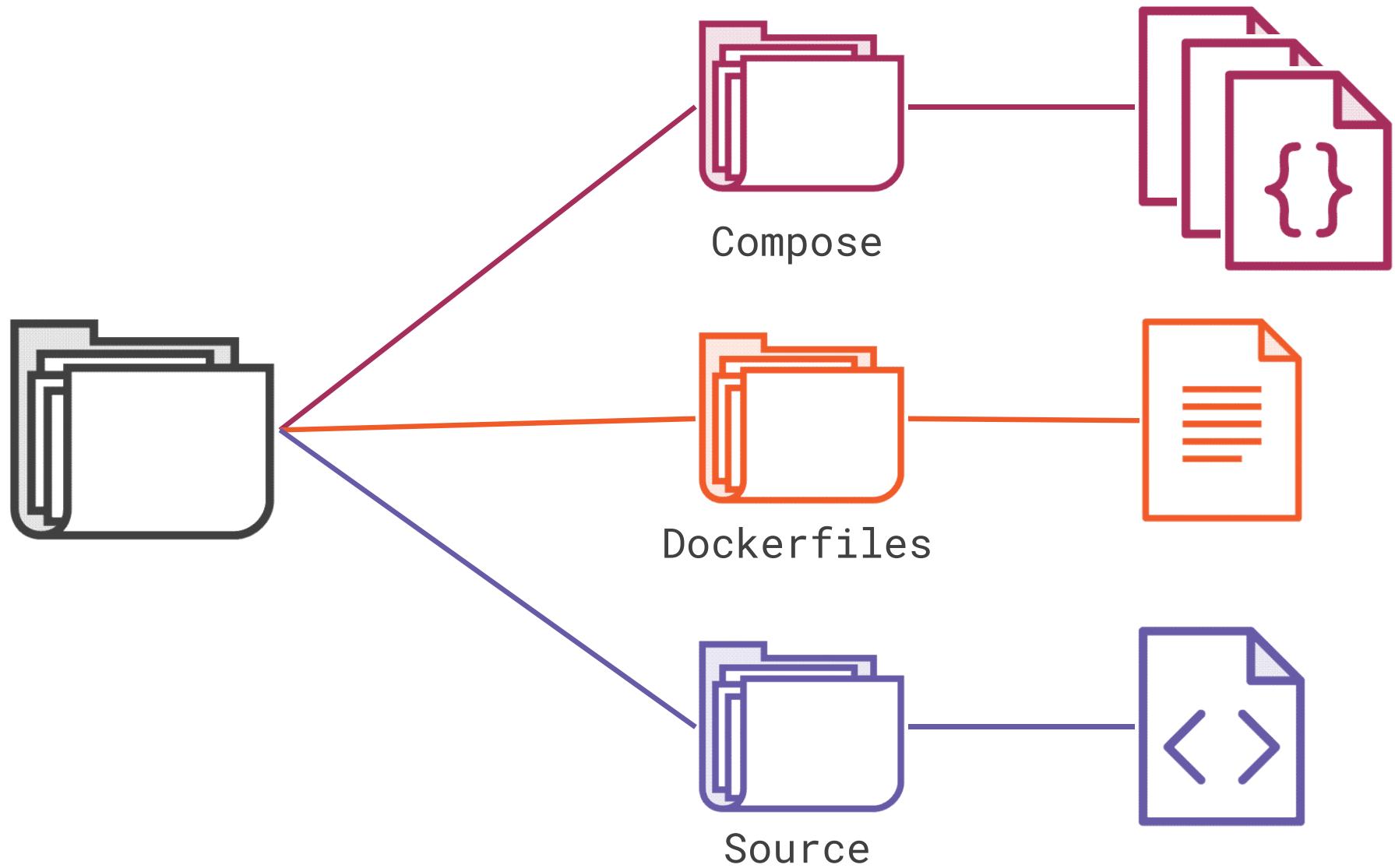
`docker-compose start`

`docker-compose down`

Docker Compose Command Line

**Manage application lifecycle**

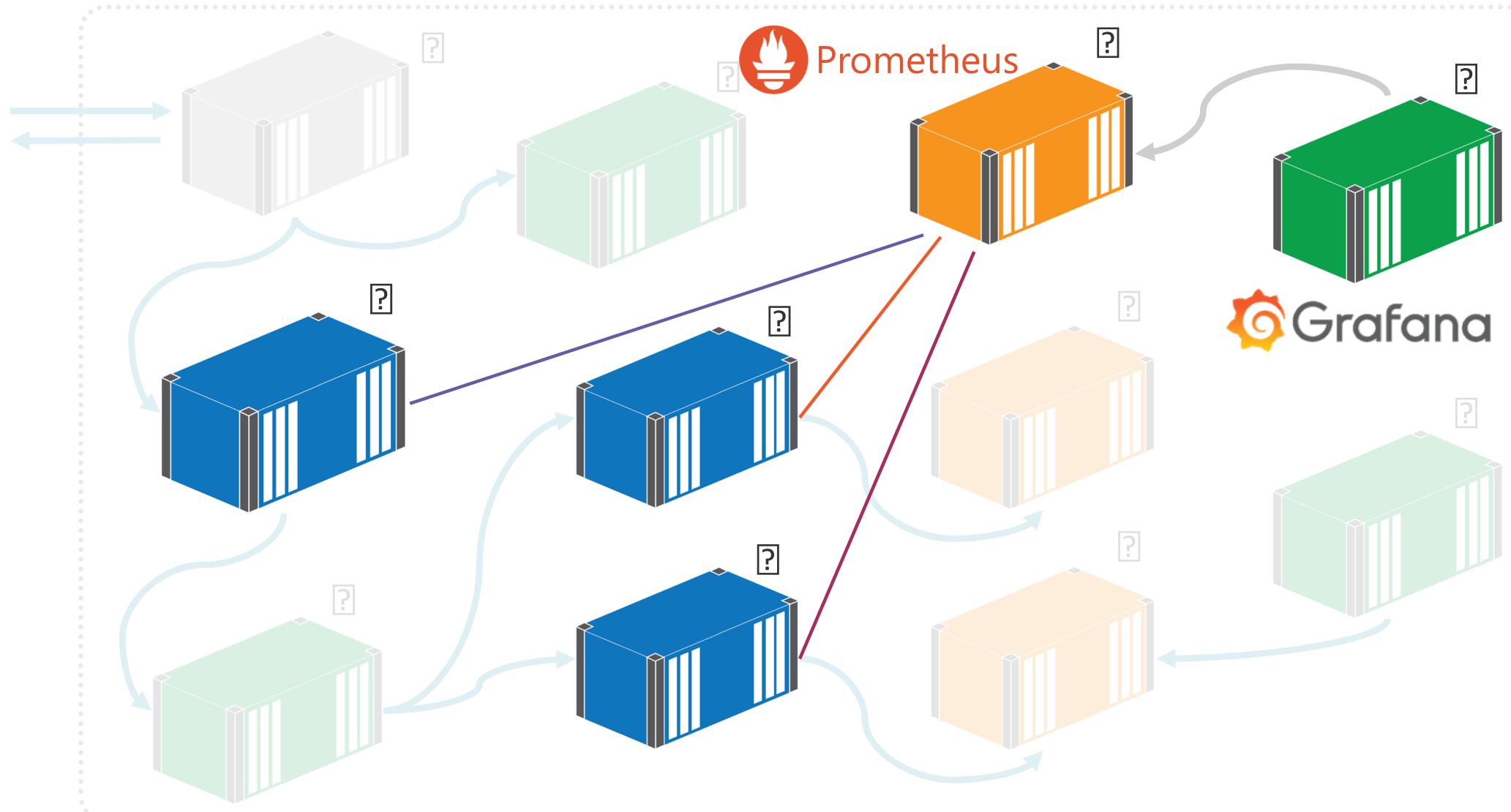




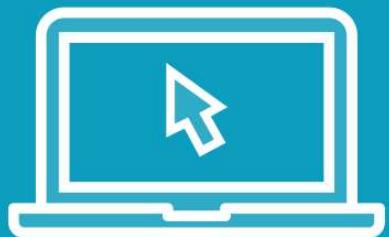
```
docker-compose logs save-handler
```

```
docker-compose top save-handler
```

Monitoring with Docker Compose  
**Application logs and process list**



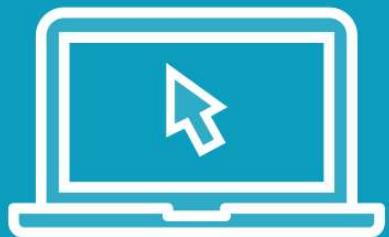
## Demo



### Collecting Metrics with Prometheus

- NuGet Prometheus package
- Custom message handler metrics
- Standard IIS performance counters

## Demo



### Running Prometheus in Docker

- Packaging a Windows image
- Adding to Docker Compose
- Check metrics from app containers

```
private static Counter _EventCounter =  
Metrics.CreateCounter("SaveHandler_Events",  
                      "Event count", "host", "status");  
//...  
_EventCounter.Labels(_Host, "received").Inc();
```

Exposing Custom Metrics  
Collecting event handler counters

```
var server = new MetricServer(50505,  
    new IOnDemandCollector[] { new DotNetStatsCollector() });  
  
server.Start();
```

## Exposing Custom Metrics Hosting a metric server endpoint

```
FROM dockersamples/aspnet-monitoring-exporter AS exporter
#...
WORKDIR C:\aspnet-exporter
COPY --from=exporter C:\aspnet-exporter .
```

Exposing IIS Performance Counters

**Adding an exporter app to the web image**

```
Write-Output 'Starting metrics exporter'  
Start-Process -NoNewWindow C:\...\aspnet-exporter.exe  
  
Write-Output 'Running ServiceMonitor'  
& C:\ServiceMonitor.exe w3svc
```

Exposing IIS Performance Counters  
Running the exporter app at startup

```
prometheus:
```

```
  image: prometheus
```

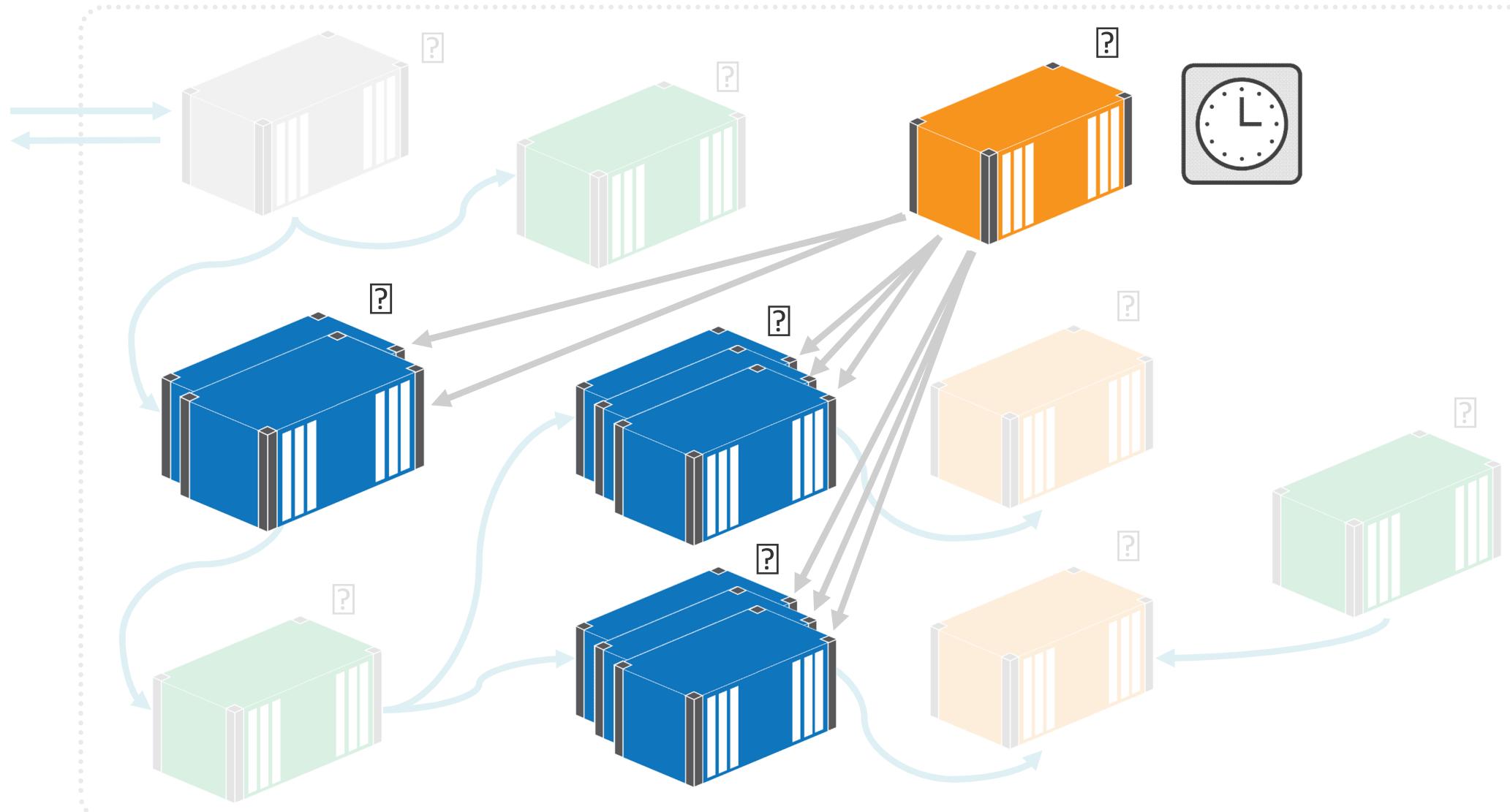
```
  ports: "9090:9090"
```

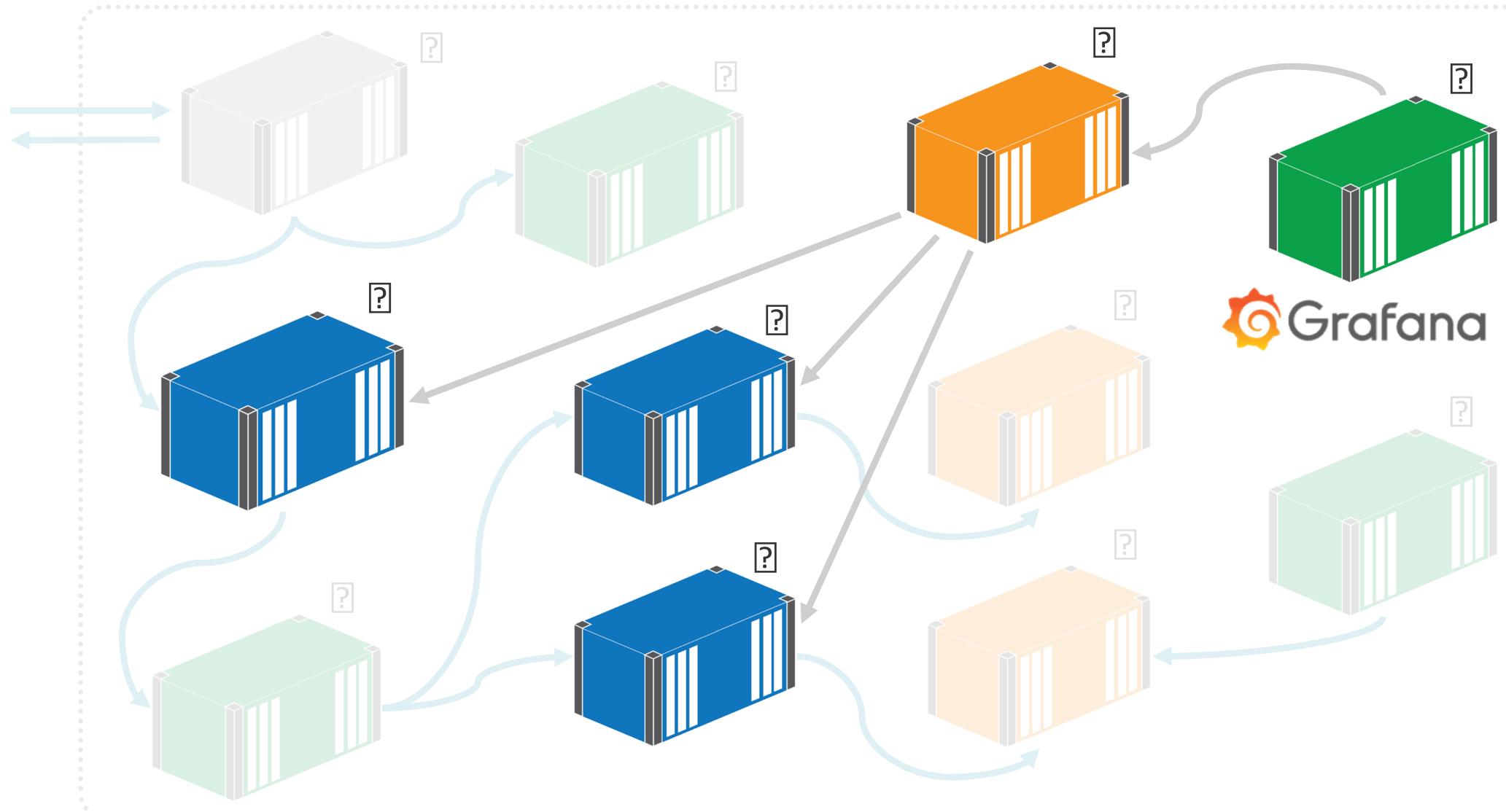
```
  networks:
```

```
    - app-net
```

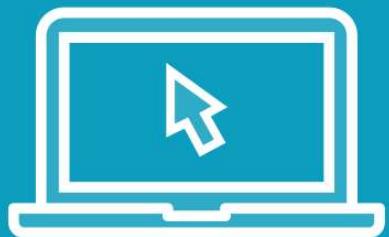
Running the Prometheus Server

**Additional service in Docker Compose file**





# Demo

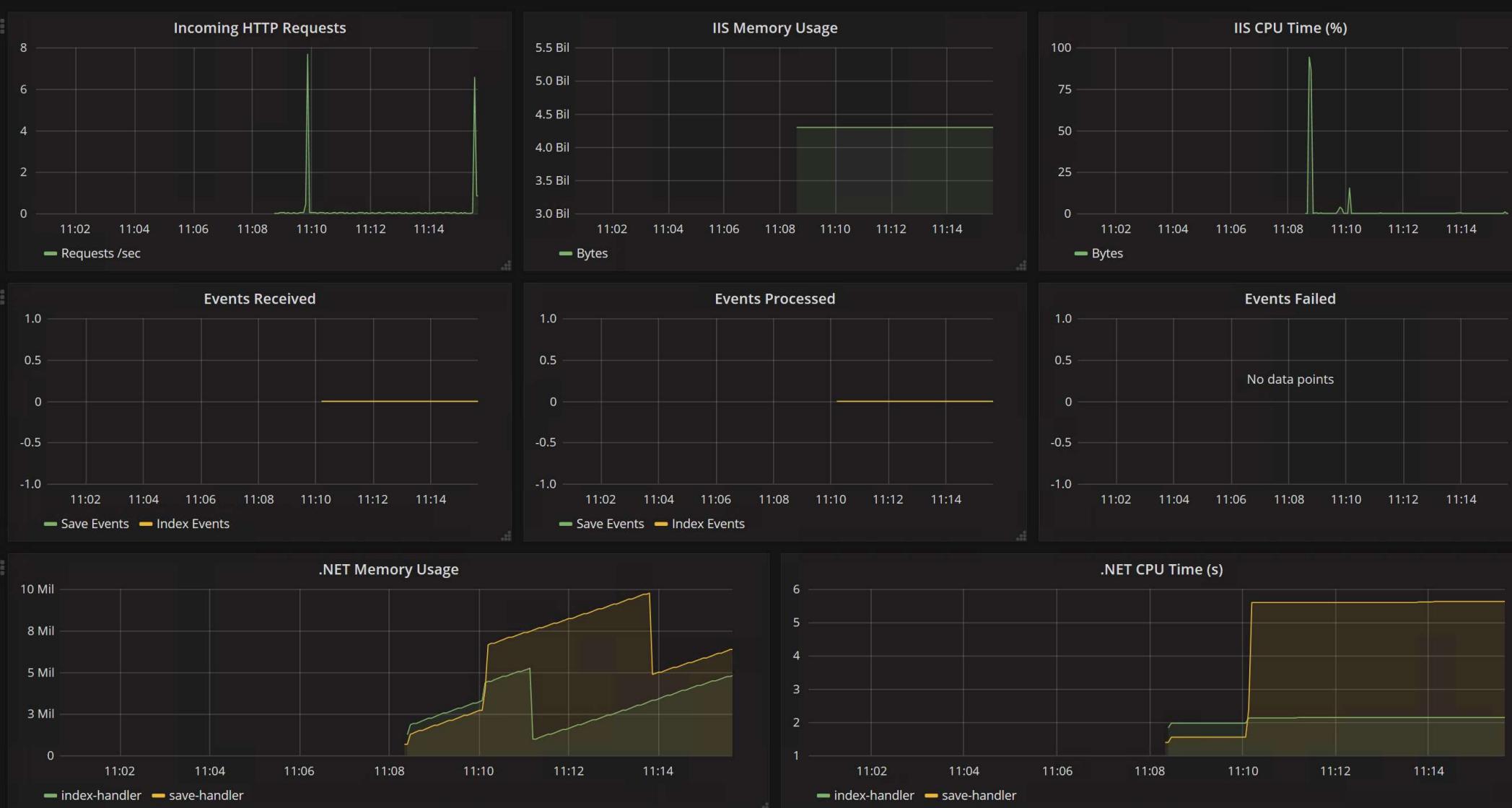


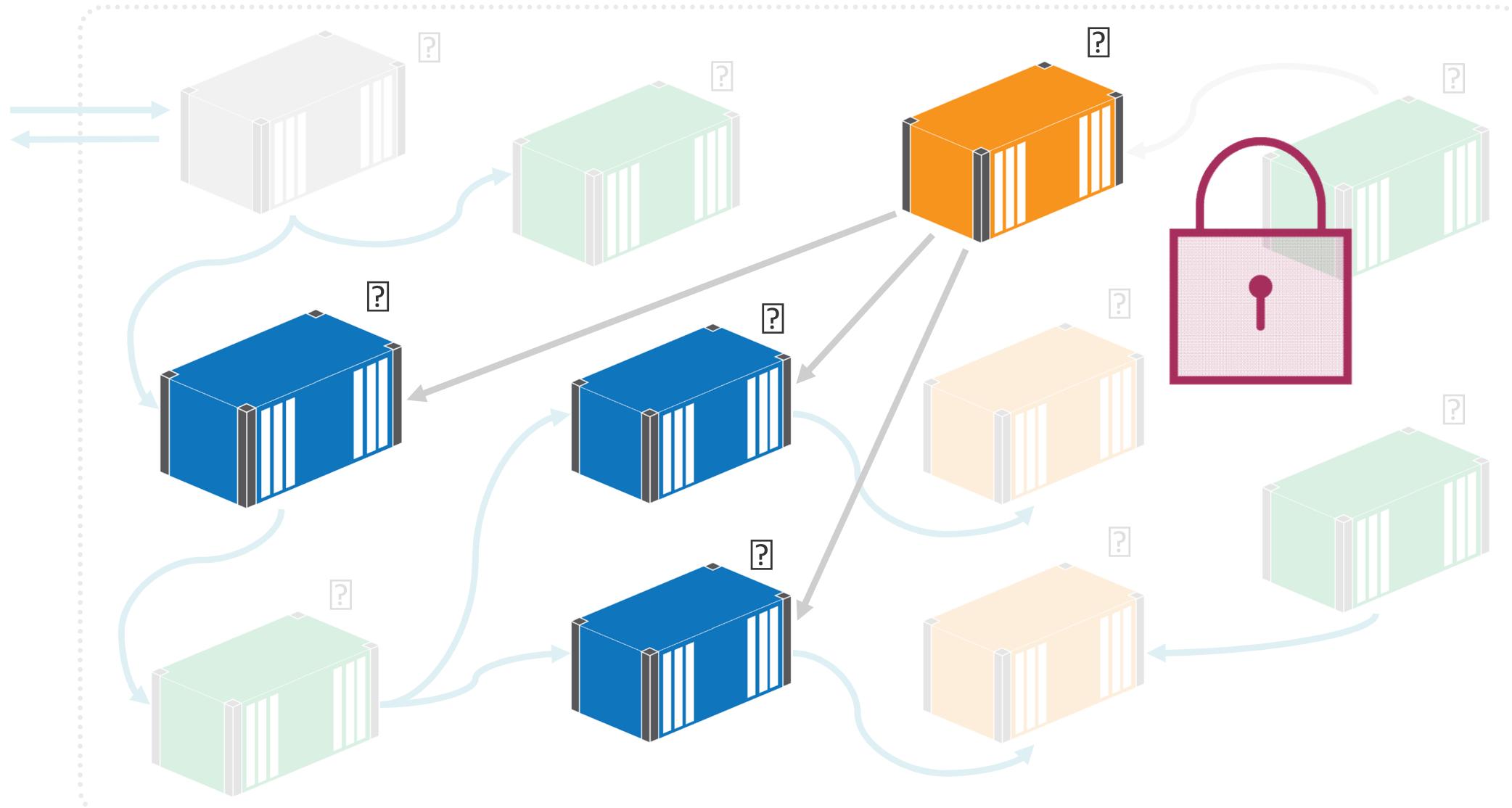
## Adding a Grafana Dashboard

- Packaging Grafana as a Docker image
- Running Grafana in a Windows container
- Adding a custom application dashboard

```
FROM microsoft/nanoserver  
EXPOSE 3000  
WORKDIR C:\grafana\bin  
CMD ["grafana-server.exe"]  
COPY --from=installer C:\grafana C:\grafana
```

Packaging Grafana in Docker  
**Multi-stage Dockerfile with web installation**

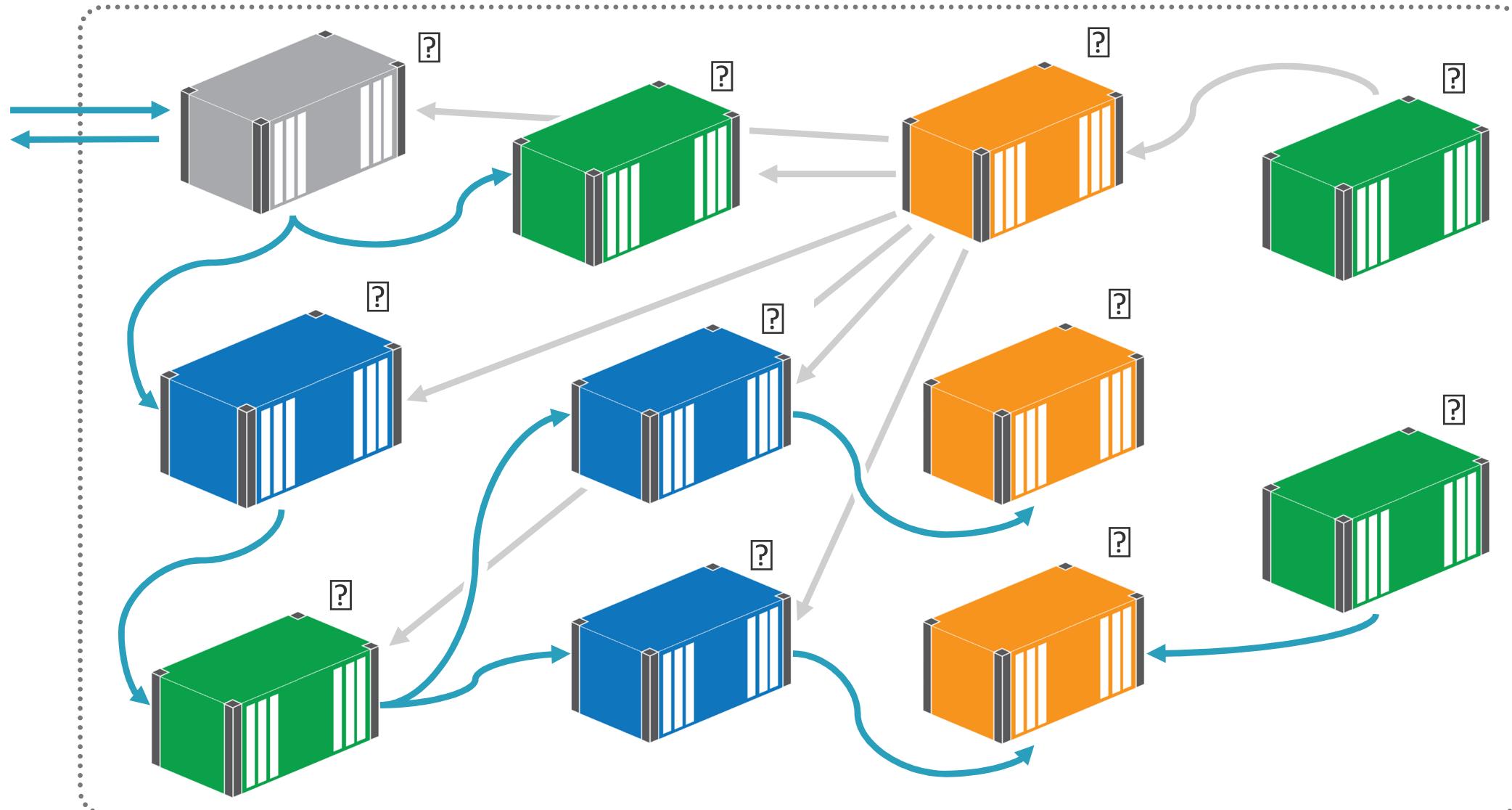






## Environmental Consistency

- Production monitoring in dev
- Developers add key metrics
- New metrics released to prod



## Coming Next



### The Path to Production

- Consistent deployment process
- Managing secrets securely
- Running with high availability