COSMO CONSULT

Business-Software for People

DYNAMICALLY SCALING BC CONTAINERS ON DOCKER SWARM USING AZURE SQL AND TRAEFIK

AREOPA WEBINAR BY TOBIAS FENSTER, DEC 10, 2019

Tobias Fenster

CTO at COSMO CONSULT Group

Dual Microsoft MVP for Business Applications and Azure

@tobiasfenster

tobiasfenster.io

tobias.fenster@cosmoconsult.com

tobiasfenster

COSMO CONSULT

Business-Software for People

WHAT ARE WE TALKING ABOUT:
SCOPE AND CHALLENGE

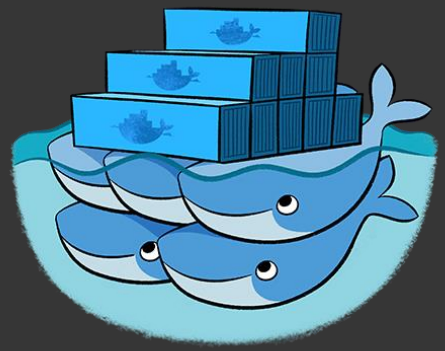
Starting point

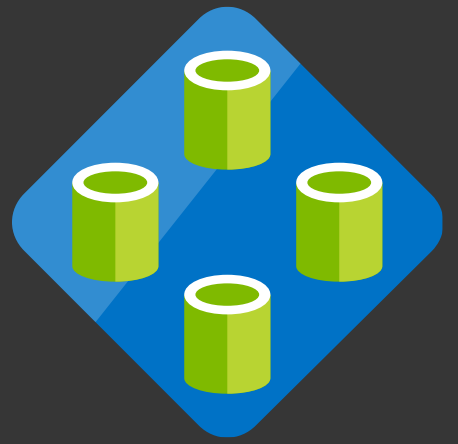
- You are running or want to run your dev or test NAV / BC environments in Docker containers
- You want to run multiple containers on the same VM
- You have solved the networking challenges (if not, see the Areopa webinar of Oct 1st 2019)

Challenges / limitations

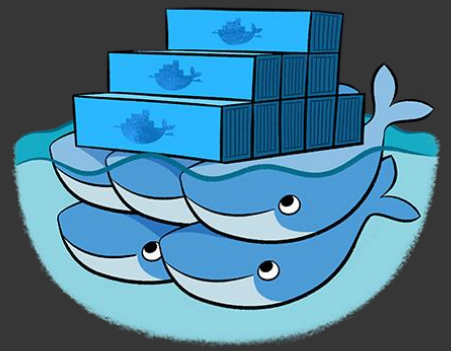
- You need more resources than one VM can provide
- You want to dynamically scale up and down
- You don't want to run SQL server in the same containers as NAV / BC (not very efficient, SQL Express, same scaling limits)

WHAT ARE WE TALKING ABOUT:
SOLUTION



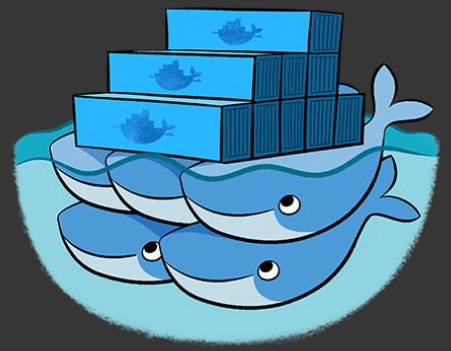


PART 1 OF THE SOLUTION:
DOCKER SWARM



- Built-in container orchestrator from Docker
- Main benefits / features:
  - Brings resources of multiple container hosts together
  - Central management and control
  - Share configuration and secrets across the swarm
  - Declarative service model
  - Automatic "self-healing" concepts
  - Advanced networking for resiliency
- What about Kubernetes and the recent Mirantis deal?

PART 1 OF THE SOLUTION:
DOCKER SWARM



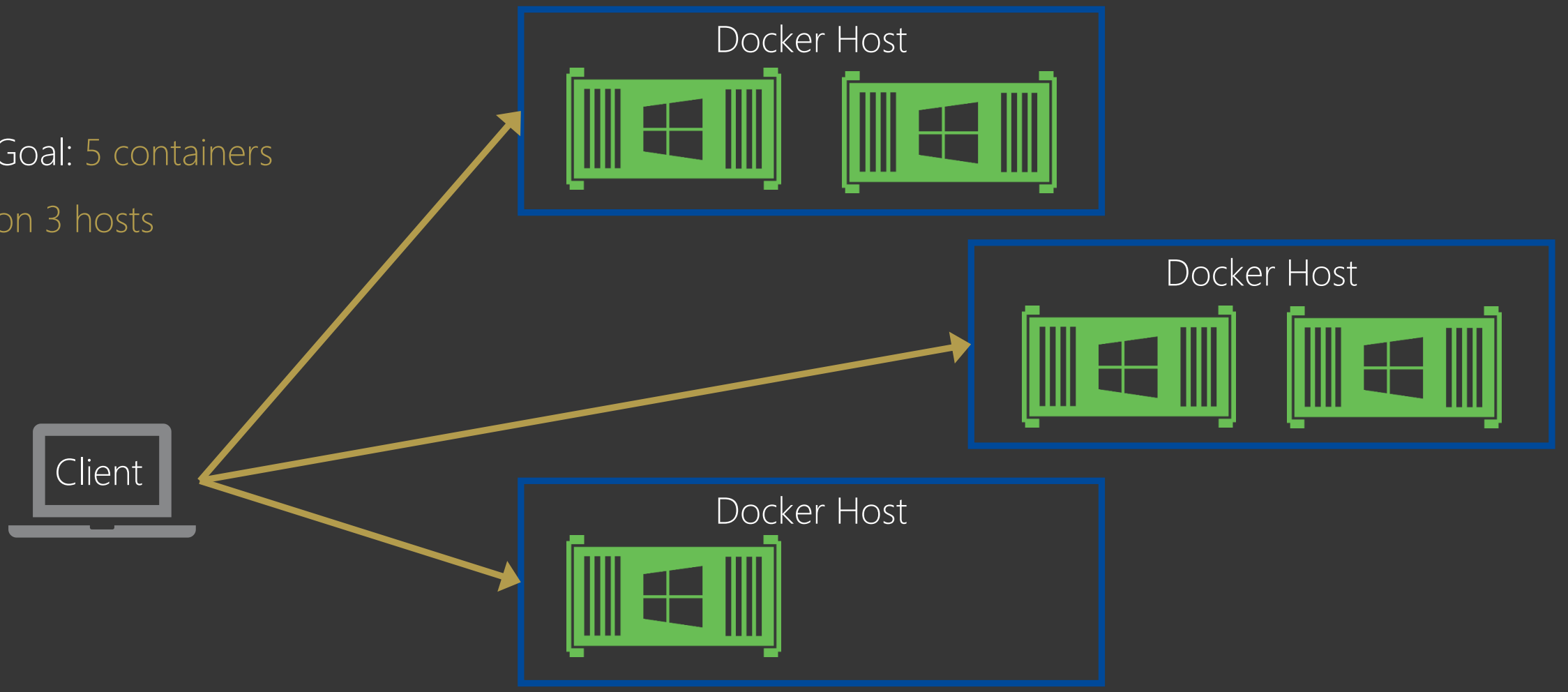
- Some basics in the Docker Swarm world:
  - Service = declaration of the images, number of containers (tasks) and configurations you want to run; can be replicated or global
  - Node = container host / engine that joined the swarm
  - Manager = node with control rights
  - Worker = node that executes tasks

- How to "run" something:
  1. Declare your service
  2. Submit that to a manager node
  3. Swarm creates necessary tasks on nodes and keeps desired state

PART 1 OF THE SOLUTION:
DOCKER SWARM

Goal: 5 containers

on 3 hosts

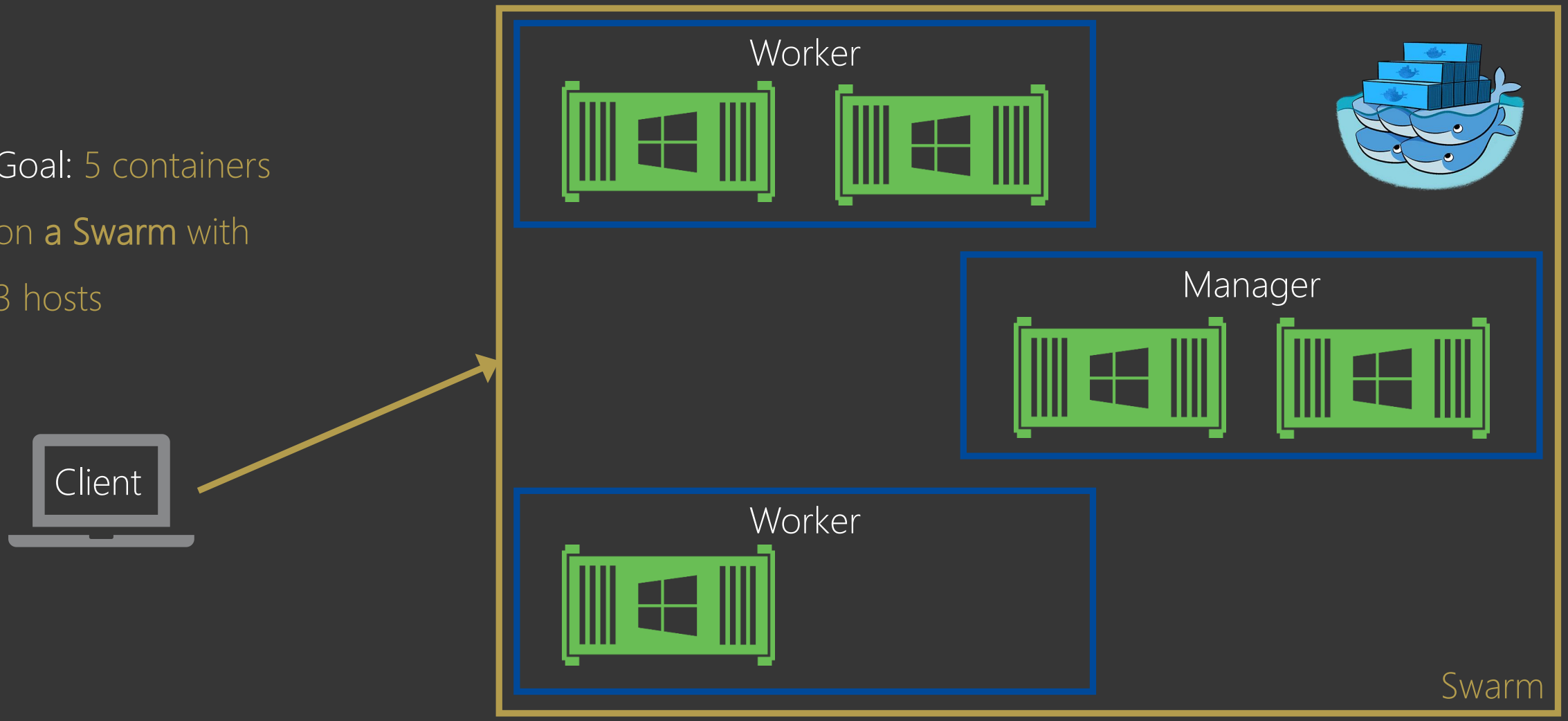


PART 1 OF THE SOLUTION:
DOCKER SWARM

Goal: 5 containers

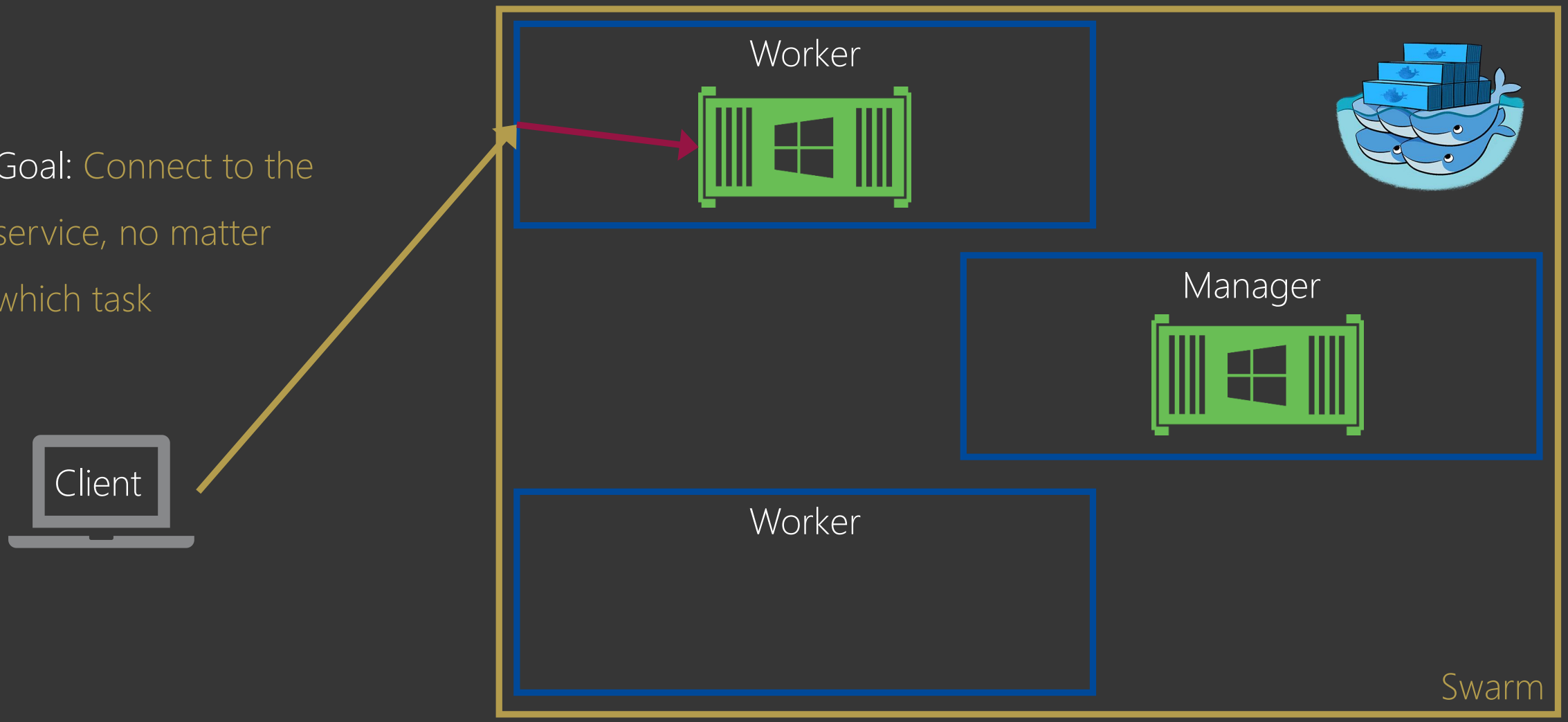
on **a Swarm** with

3 hosts



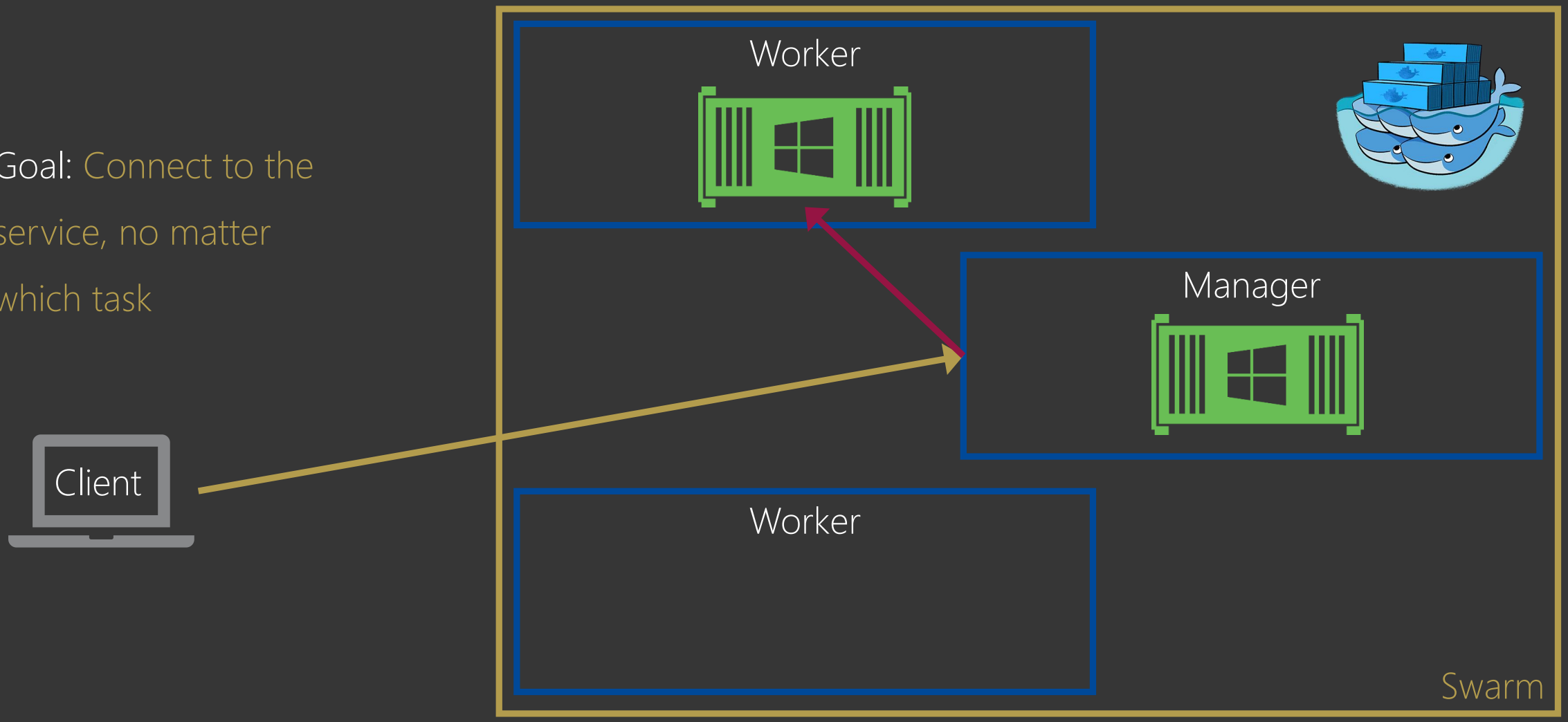
PART 1 OF THE SOLUTION:
DOCKER SWARM

Goal: Connect to the service, no matter which task



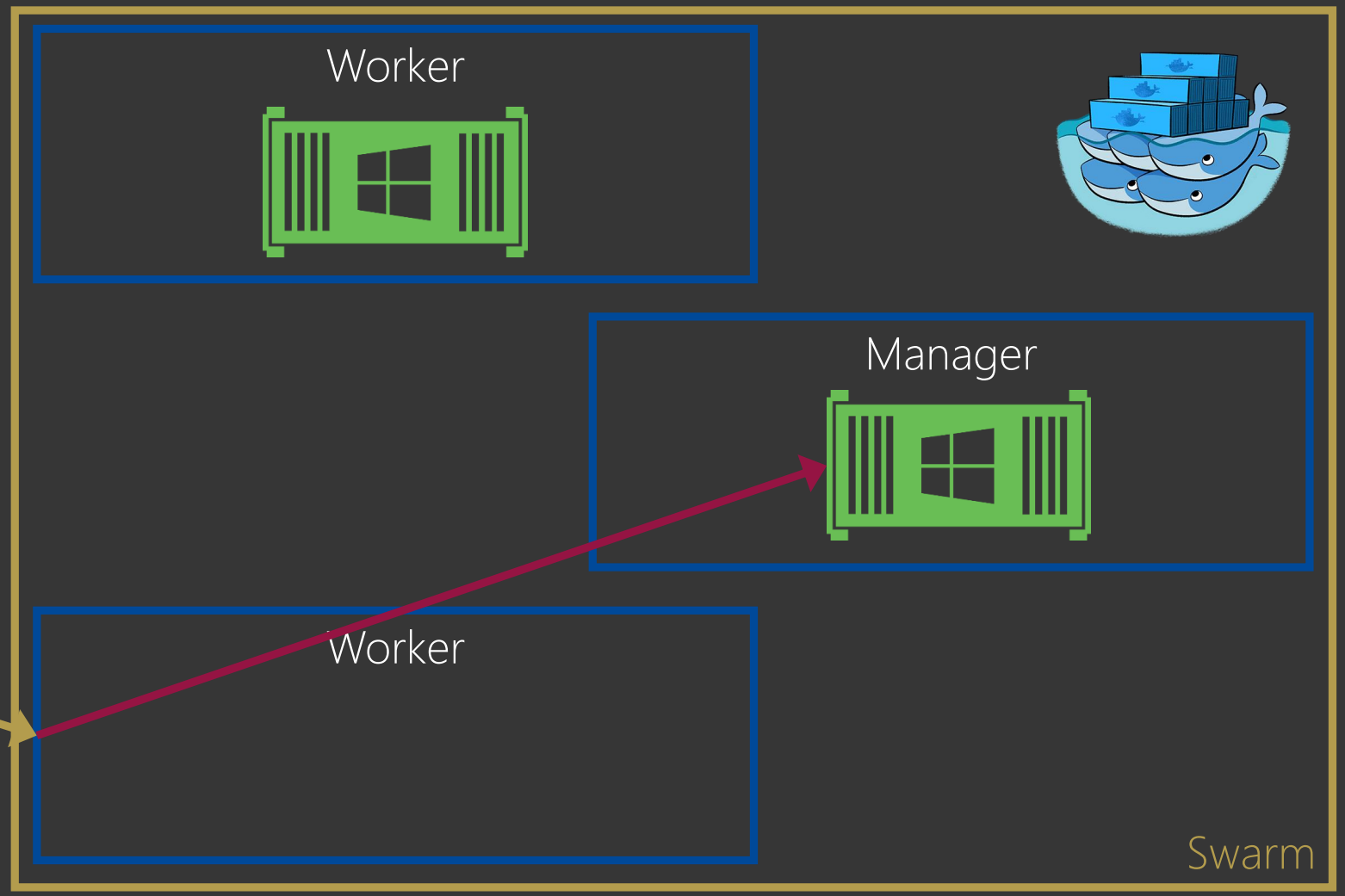
PART 1 OF THE SOLUTION:
DOCKER SWARM

Goal: Connect to the service, no matter which task



PART 1 OF THE SOLUTION:
DOCKER SWARM

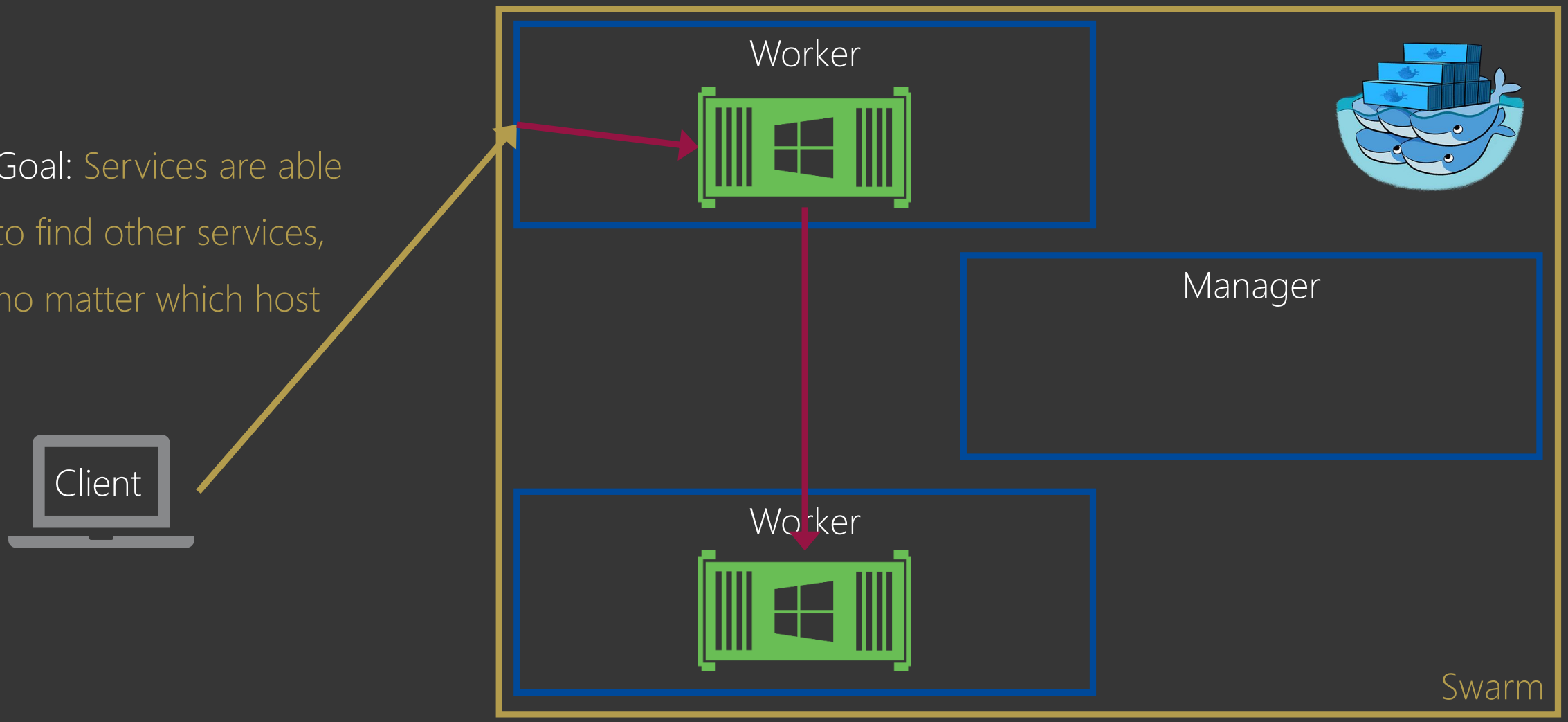
Goal: Connect to the service, no matter which task



Doesn't directly work with BC as it requires stateless service

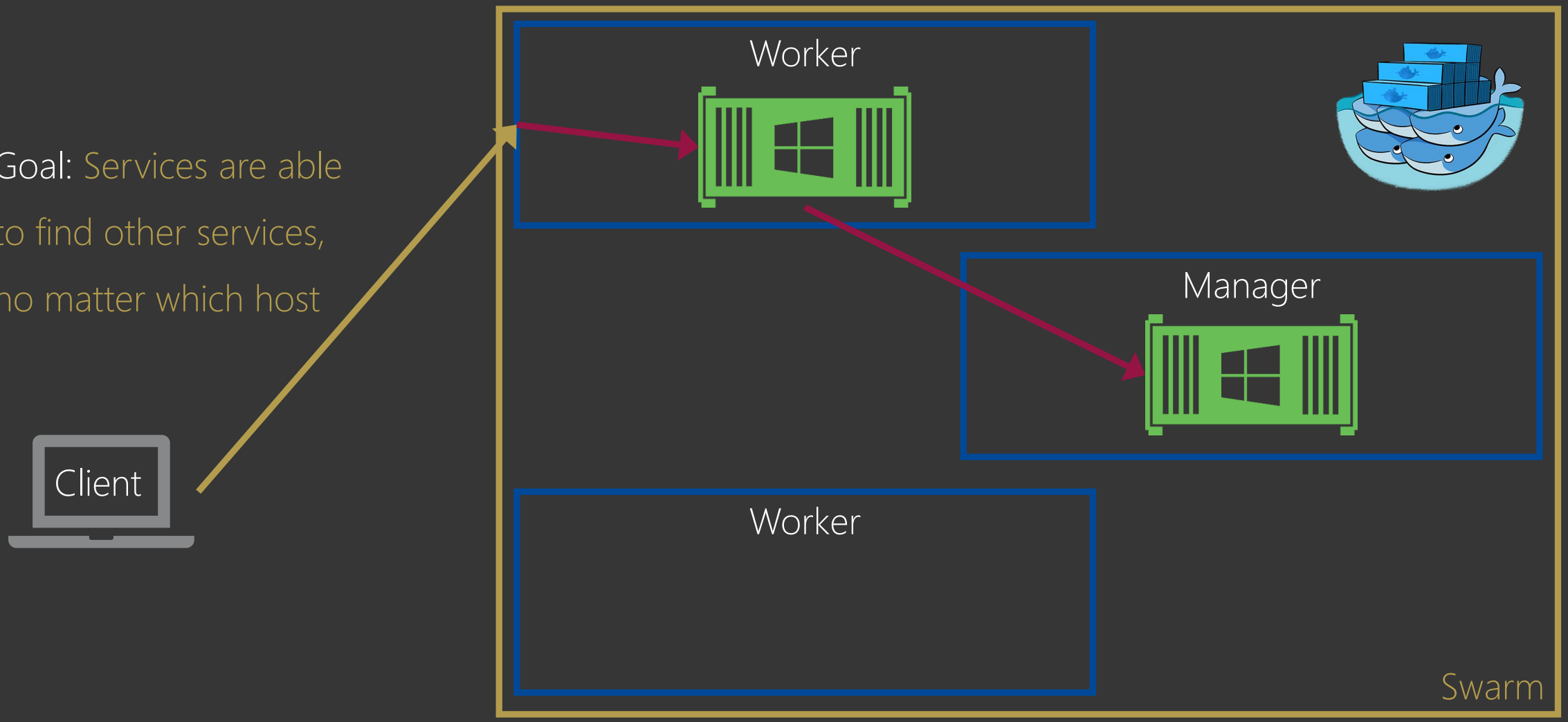
PART 1 OF THE SOLUTION:
DOCKER SWARM

Goal: Services are able to find other services, no matter which host

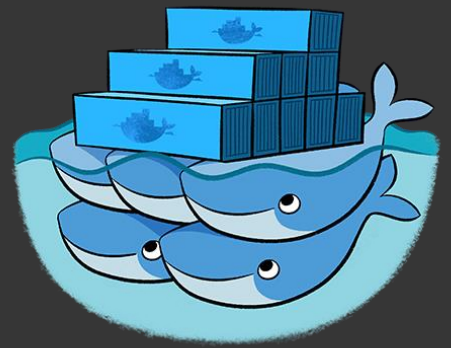


PART 1 OF THE SOLUTION:
DOCKER SWARM

Goal: Services are able to find other services, no matter which host



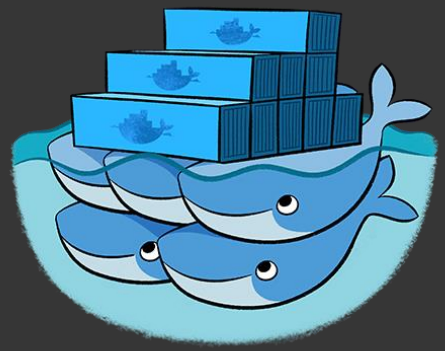
PART 1 OF THE SOLUTION:
DOCKER SWARM

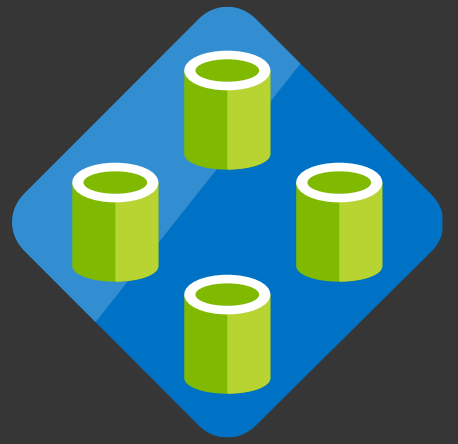


- Demo scenario:
  - Create 1 service consisting of 2 containers
  - → Check connections
  - Scale service up
  - → See placement on nodes
  - Remove task
  - → See recovery

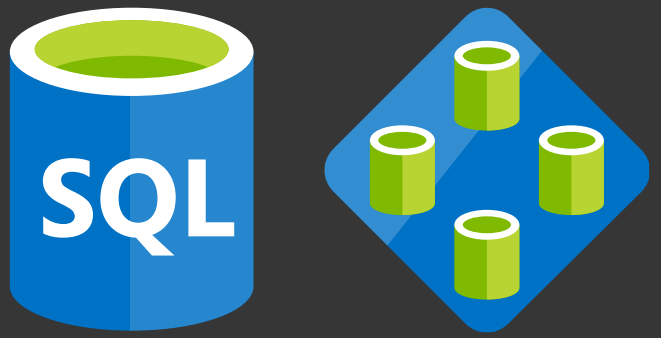
→ Let's see it!

WHAT ARE WE TALKING ABOUT:
SOLUTION





PART 2 OF THE SOLUTION:
AZURE SQL WITH AN ELASTIC POOL



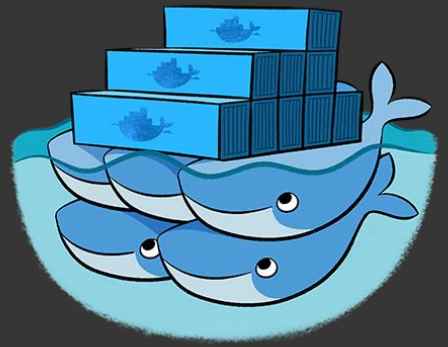
- Platform as a Service (PaaS) offering
  - Always kept current by Microsoft
  - Scale up and down dynamically
  - (Almost) no resource limits
  - Elastic pools allow resource sharing across all databases (with configurable limits)

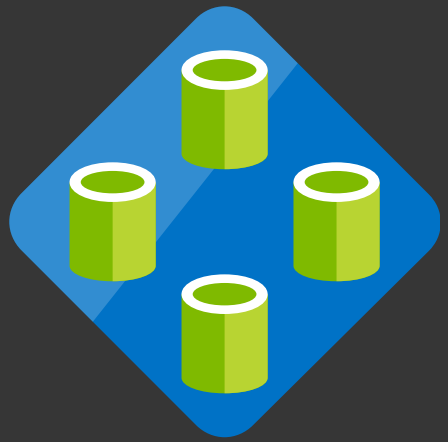
→ No server maintenance, no limits; just cost…

WHAT ARE WE TALKING ABOUT:
REMEMBER THE START?

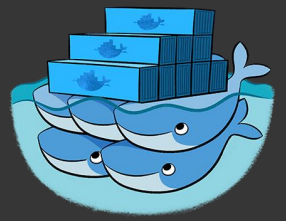
Challenges / limitations

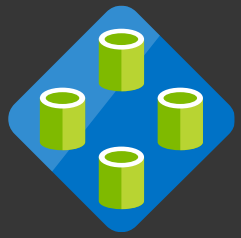
- You need more resources than one VM can provide
- You want to dynamically scale up and down
- You don't want to run SQL server in the same containers as NAV / BC (not very efficient, SQL Express, same scaling limits)





BRINGING IT TOGETHER:
SWARM (AND TRAEFIK) & AZURE SQL





- Prereq: Azure SQL database with a "template" database
- 1 ARM template to deploy
  - 1 manager (SPOF!) and x workers
  - 1 Azure SQL server with 1 elastic pool
  - Setup script to initialize the Swarm on the manager, join it on the workers and set up Traefik
  - Script to prepare access credentials for template DB and target pool and share them as secrets
- Specific scenario: Start a BC Swarm service and connect to a DB created on demand as copy of the template

→ Let's see it

COSMO CONSULT

Business-Software for People

Get production support for Docker containers with D365 BC: http://bit.ly/DockerProd

Get rid of the 120kB size limit for dev licenses: http://bit.ly/UnlimitedFlf

THANK YOU

FOR YOUR TIME

WHICH QUESTIONS CAN I TRY TO ANSWER?