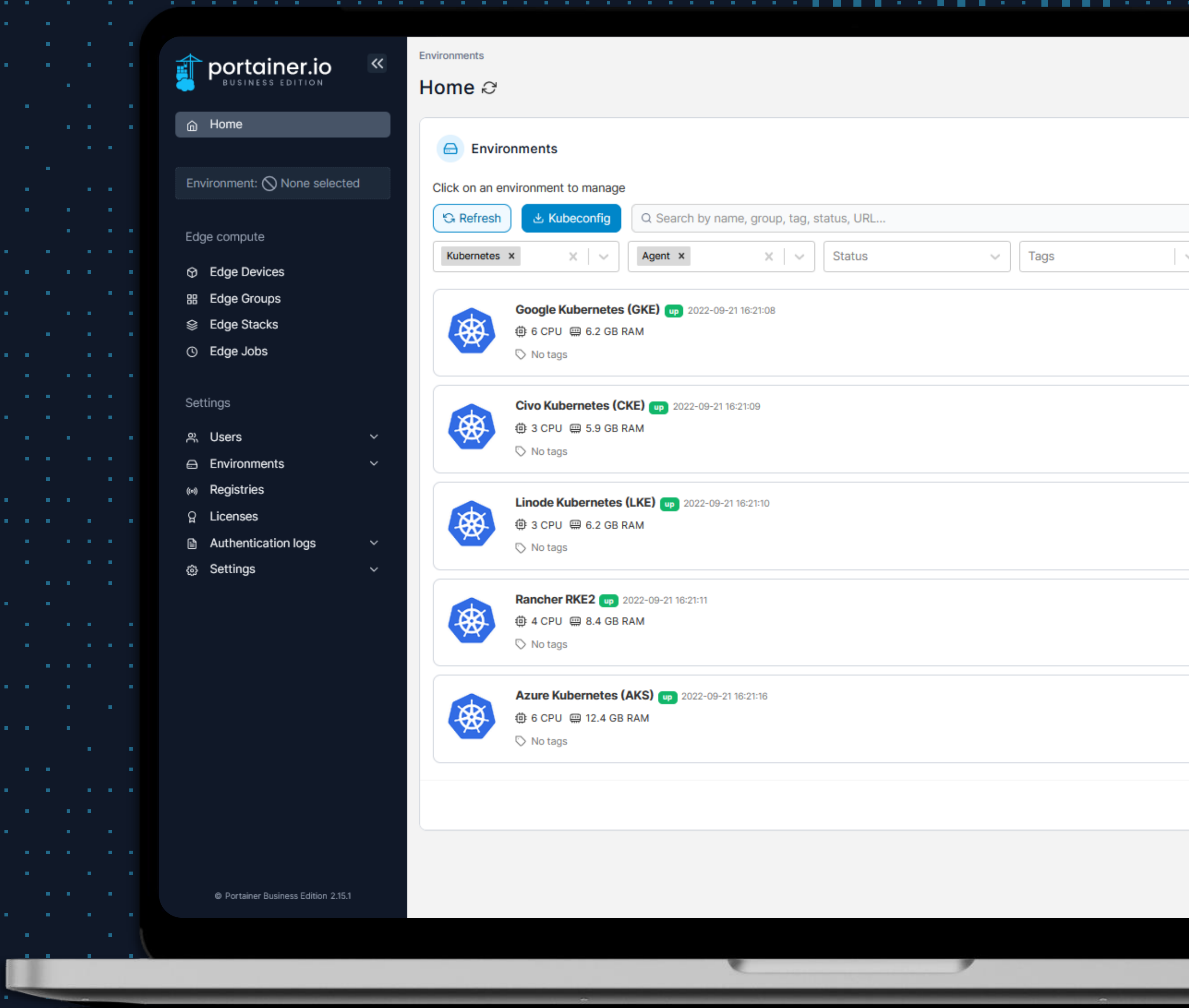
portainer.io

Introducing Portainer

Because managing containers shouldn't be hard!



Environments

Click on an environment to manage

Google Kubernetes (GKE) up 2022-09-21 16:21:08
6 CPU 6.2 GB RAM
No tags

Civo Kubernetes (CKE) up 2022-09-21 16:21:09
3 CPU 5.9 GB RAM
No tags

Linode Kubernetes (LKE) up 2022-09-21 16:21:10
3 CPU 6.2 GB RAM
No tags

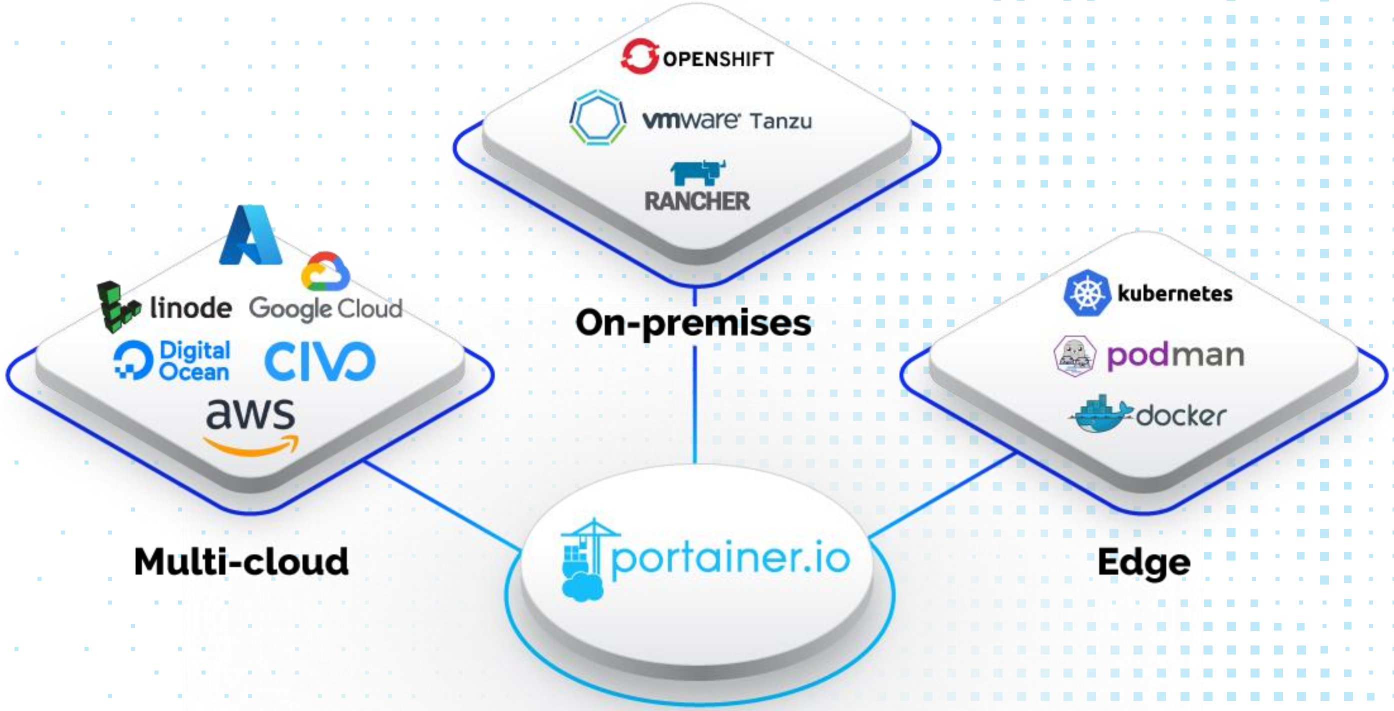
Rancher RKE2 up 2022-09-21 16:21:11
4 CPU 8.4 GB RAM
No tags

Azure Kubernetes (AKS) up 2022-09-21 16:21:16
6 CPU 12.4 GB RAM
No tags

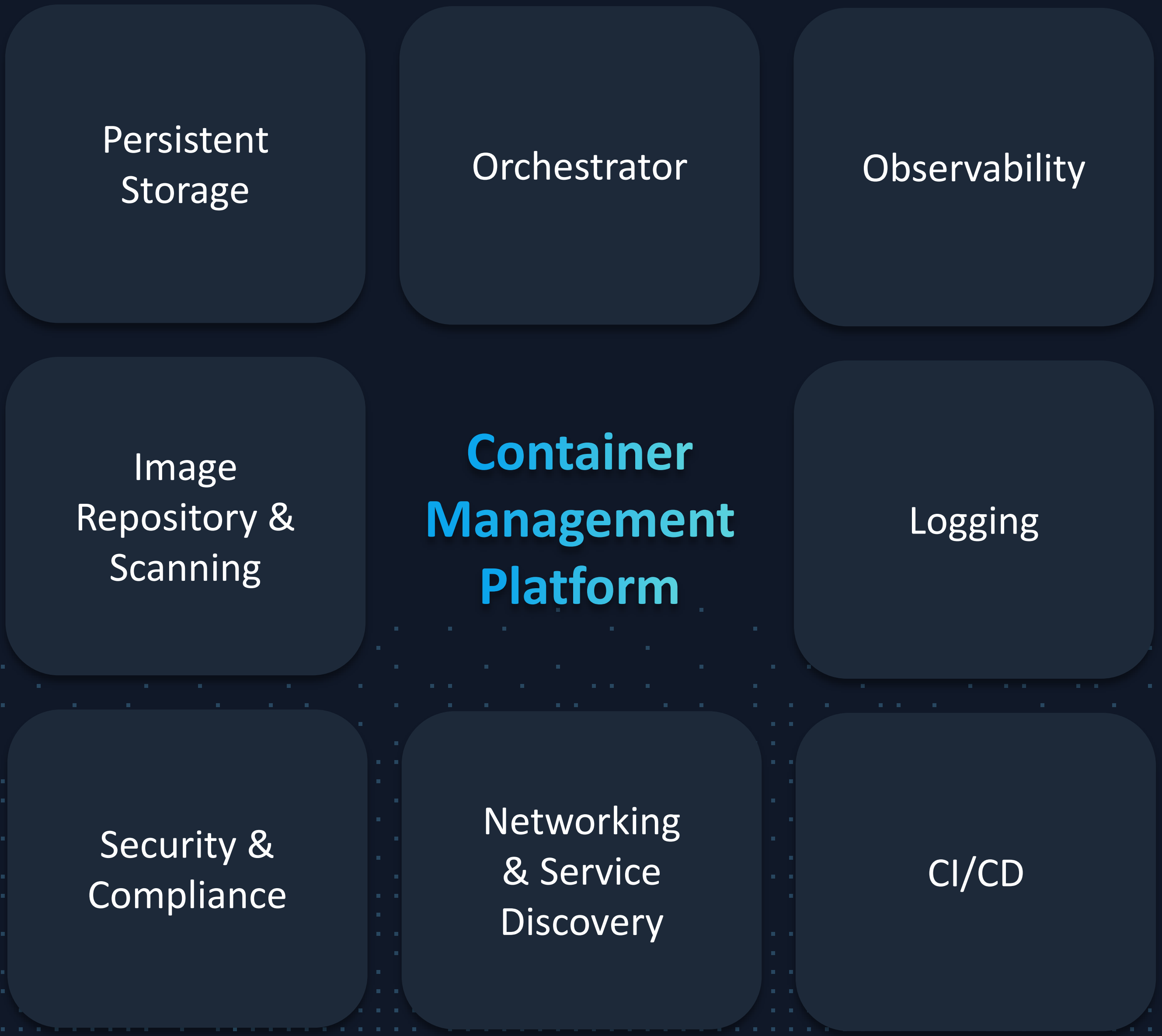
What is Portainer?

Portainer is a **centralized container management platform.**

Portainer accelerates container adoption. It reduces operational complexity and addresses the challenges of running containers in Docker, Podman, Nomad and Kubernetes.



What actually is a Container Management Platform?



Persistent Storage

Orchestrator

Observability

Image Repository & Scanning

Container Management Platform

Logging

Security & Compliance

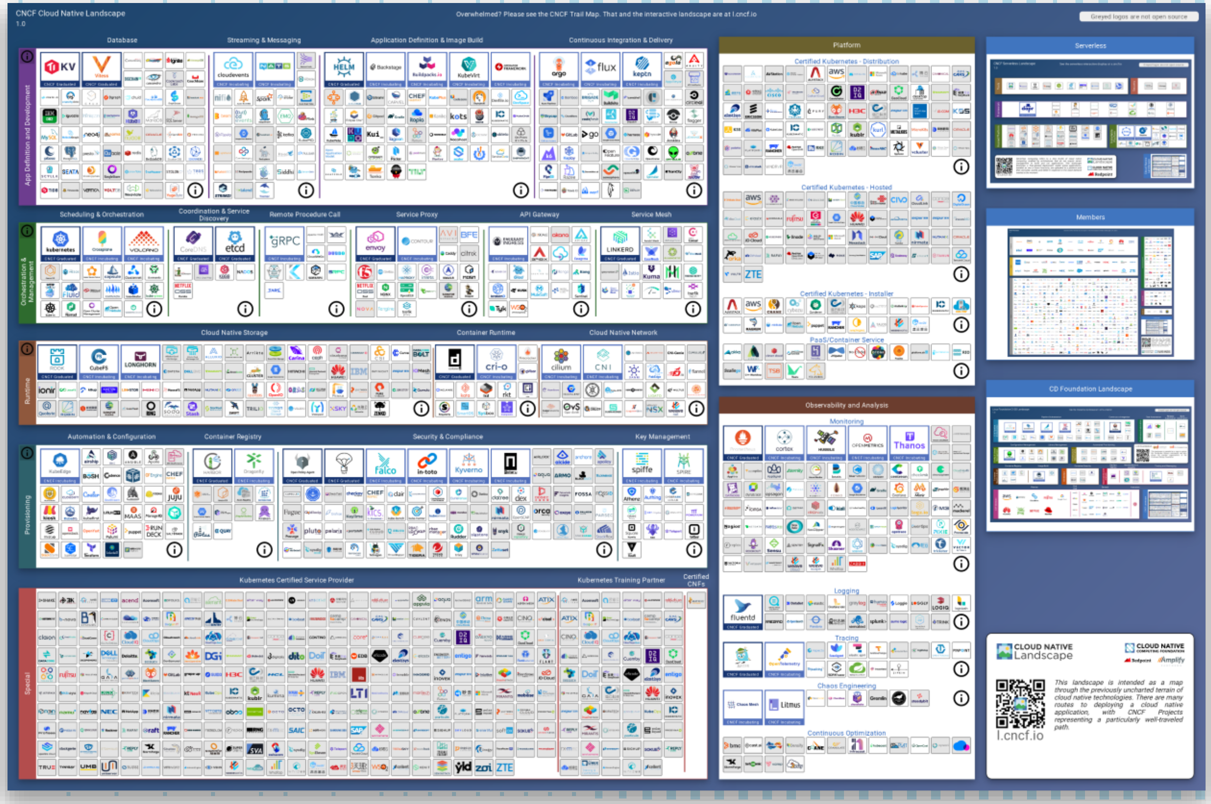
Networking & Service Discovery

CI/CD

Why should I care?

To go “into production” with containers, **regardless of scale or use case**, you need a Container Management Platform.

Either you take the time to self-assemble one from a plethora of open-source tools in the CNCF landscape, or you use Portainer; regardless, **you must choose one**.


CNCF Cloud Native Landscape
Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Serverless
Members
CD Foundation Landscape
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Runtime
Cloud Native Storage
Container Runtime
Cloud Native Network
Provisioning
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Continuous Optimization
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Certified CNFs
CLOUD NATIVE Landscape
l.cncf.io

SIX reasons why you need Portainer

1

Skilled labour shortages are hampering the widespread adoption of containerization; you need to enable your existing staff through tooling.

2

Operator skills ramp-up time is too long, impacting time for projects to go live. Tooling can shorten this.

3

Securing container platforms correctly is a minefield, only with tooling can you be sure its correct.

4

Multi-Cloud and Multi-Cluster deployments complicate the adoption of Containerization, centralized tooling eases this burden.

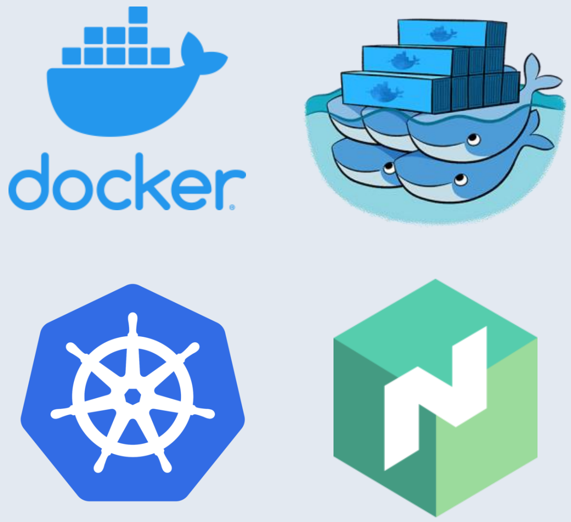
5

Day 2 Operations is confusing; triage, monitoring, alerting are all critical elements to success. These are all provided by tooling.

6

Secure Developer and Operations access to environments is hard to manage without tooling.

Portainer Product Variants



Portainer Universal Container Management Platform

Portainer Kubernetes Management Platform

Portainer Edge Management Platform

Portainer Universal Container Management Platform - Benefits

We facilitate your transition

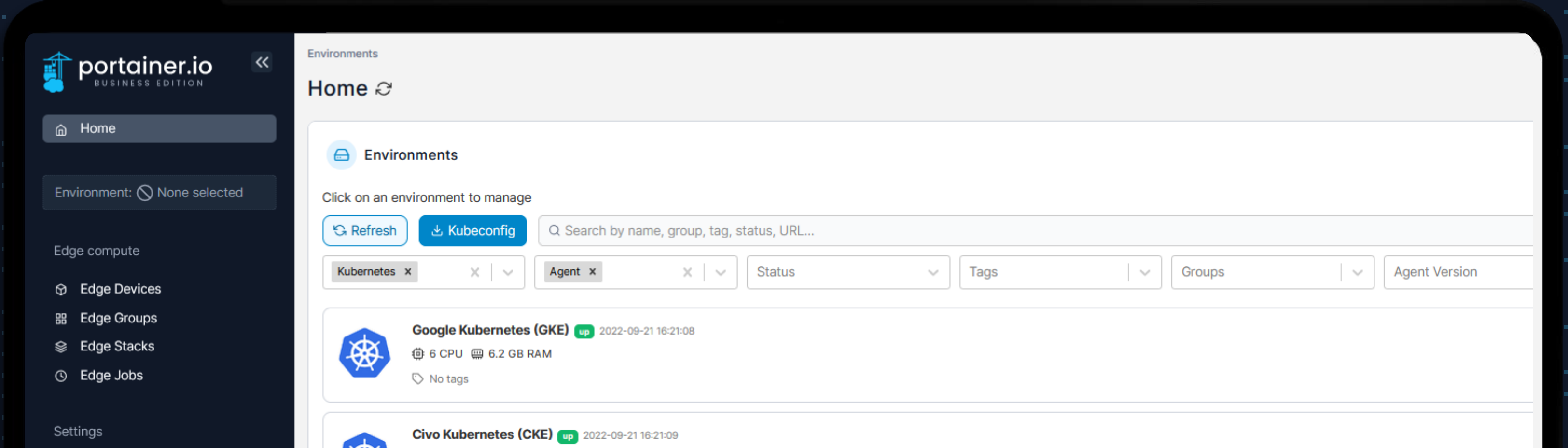
From VM’s to Containers with Portainer; we help you get started with Docker, aide in the learning of distributed micro-service fundamentals, and then when the time is right, help you transition to Kubernetes

We show you what’s possible

With containers, helping you to discover the power of this game changing technology, without you having to go learn all the commands, and what the commands do; we make all features immediately obvious, allowing for each exploration and experimentation

We help you get going fast

With your container based micro-service in the shortest amount of time possible, by removing the platform complexity commonly associated with the leap into a fully orchestrated, declarative, * as code world. Our No-Code UI lets you click to deploy a range of applications, of near limitless complexity



Environment: None selected

Click on an environment to manage

Search by name, group, tag, status, URL...

Kubernetes × | Agent ×

Google Kubernetes (GKE) up 2022-09-21 16:21:08
6 CPU 6.2 GB RAM
No tags

Civo Kubernetes (CKE) up 2022-09-21 16:21:09

Portainer Kubernetes Management Platform - Benefits

We make multi-cluster management a breeze

Whether you have 2 clusters or 2000+, we help you configure centralised user authentication, assign RBAC roles to users, and provide your users with a single Kubernetes API endpoint for their native tooling.

We remove any vendor lock-in

Doesn't matter which Kubernetes distribution you want to use, which cloud provider you deploy upon, or even if you run on-premises, we can help you centrally manage them all, in a consistent way, without having to master an array of vendor tooling.

We help you to be secure

By default, Kubernetes is not secured, and unless you know how to, you risk security vulnerabilities. We help you to set policies, assign quotas, and let platform teams define golden paths and templates for devs to self-serve within

We simplify triage and monitoring

Deploying apps is one thing, monitoring and triaging them is another. We provide you native access to Kubernetes Metrics, let you easily view logs of your Pods, and visualise cluster events in a really simple way

No expert skills needed to drive Portainer

If you think the barrier to adopting Kubernetes is access to skilled staff, you would be correct. We lower the skill level needed to use and manage Kubernetes, allowing most IT Admins/Ops/Devs to use Kubernetes without needing formal certification first.

Portainer Edge Management Platform - Benefits

We support massive scale

With edge, scale is a given. You will likely need to manage hundreds, thousands, or even hundreds of thousands of devices. We help you to centrally manage these devices, and help you to deploy your applications in a simple, highly scalable way.

Designed for OT

In the edge world, the most common use case is Industrial IOT, and in that world, the operators are often OT engineers. Portainer’s UI is highly intuitive and helps guide OT engineers so they can exploit IT technologies safely.

Devices Management

Managing a software deployment is one thing, but the devices the software runs upon also needs managing. We help you to manage the physical devices, through Zero-Touch Device Onboarding and Lights-Out management, ensuring you have the ability to deliver reliable SLAs to your business

Self Hosted or Air Gapped.

In edge, its often the case that the devices are running critical machinery or systems, and as such are in isolated or highly controlled networks. We are self-hosted, so can run safe and sound within the confines of your network

Super Lightweight

Edge hardware is often resource constrained, due to the nature of the use case. Its not uncommon to see devices with 1GB or RAM or less. We operate happily with just 20MB of RAM. Even better, we support all container runtimes, Docker, PodMan, along with Lightweight Kubernetes distributions/

Reliable

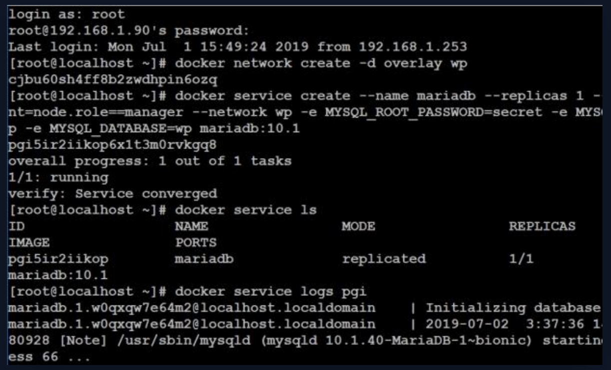
At the edge, you may need to manage devices connected to unreliable, sporadic, or high cost networks. We appreciate this and operate over high latency and sporadically connected networks, and we do this without compromising capability.

Portainer’s goal “in a picture”

Bringing expert simplicity to very complex technology

Containers are complex to use

- Available tools are “heavy”, low level, error prone and slow time to value
- High risk of failure and syntax type errors
- Requires specialist training
- Requires people with skill sets that are hard to find and expensive to retain

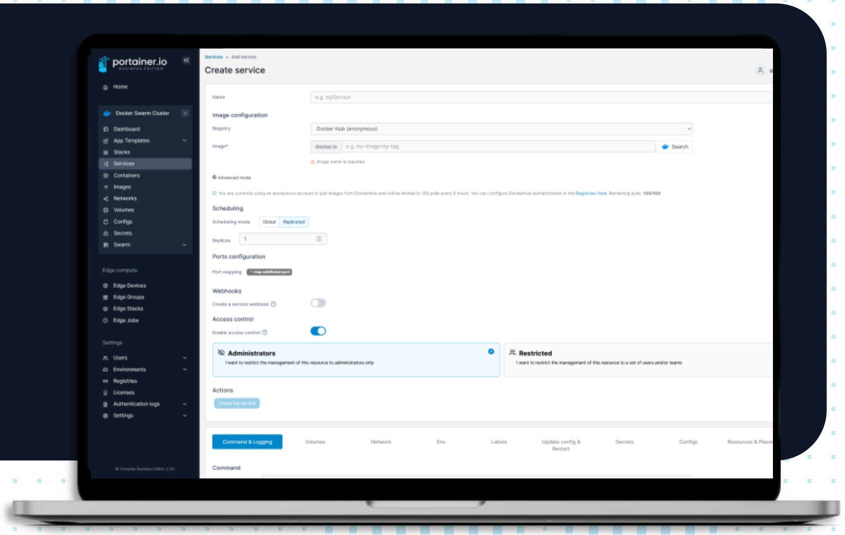


```
login as: root
root@192.168.1.90's password:
Last login: Mon Jul  1 15:49:24 2019 from 192.168.1.253
[root@localhost ~]# docker network create -d overlay wp
cjbu60sh4ff8b2zwdhpin6ozq
[root@localhost ~]# docker service create --name mariadb --replicas 1 -
nt=node.role==manager --network wp -e MYSQL_ROOT_PASSWORD=secret -e MYS
p -e MYSQL_DATABASE=wp mariadb:10.1
pgi5ir2iikop6x1t3m0rvkgq8
overall progress: 1 out of 1 tasks
1/1: running
verify: Service converged
[root@localhost ~]# docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
pgi5ir2iikop        mariadb             replicated          1/1
mariadb:10.1
[root@localhost ~]# docker service logs pgi
mariadb.1.w0qxqw7e64m2@localhost.localdomain    | Initializing database
mariadb.1.w0qxqw7e64m2@localhost.localdomain    | 2019-07-02  3:37:36 1
80928 [Note] /usr/sbin/mysqld (mysqld 10.1.40-MariaDB-1~bionic) startin
ess 66 ...
```

Portainer solves this problem

- Elegant, expert, rules-based management and operations toolset
- Substantial productivity improvements and a much better user experience
- Significantly reduces the need for operators with high skill levels

“I installed Portainer, 1 minutes (sic) later it was up and working” (Customer comment)



Services > Add service

Create service

Name: e.g. myService

Image configuration

Registry: Docker Hub (anonymous)

Image*: docker.io e.g. my-image:my-tag

Scheduling

Scheduling mode: Global Replicated

Replicas: 1

Ports configuration

Webhooks

Access control

Administrators
I want to restrict the management of this resource to administrators only

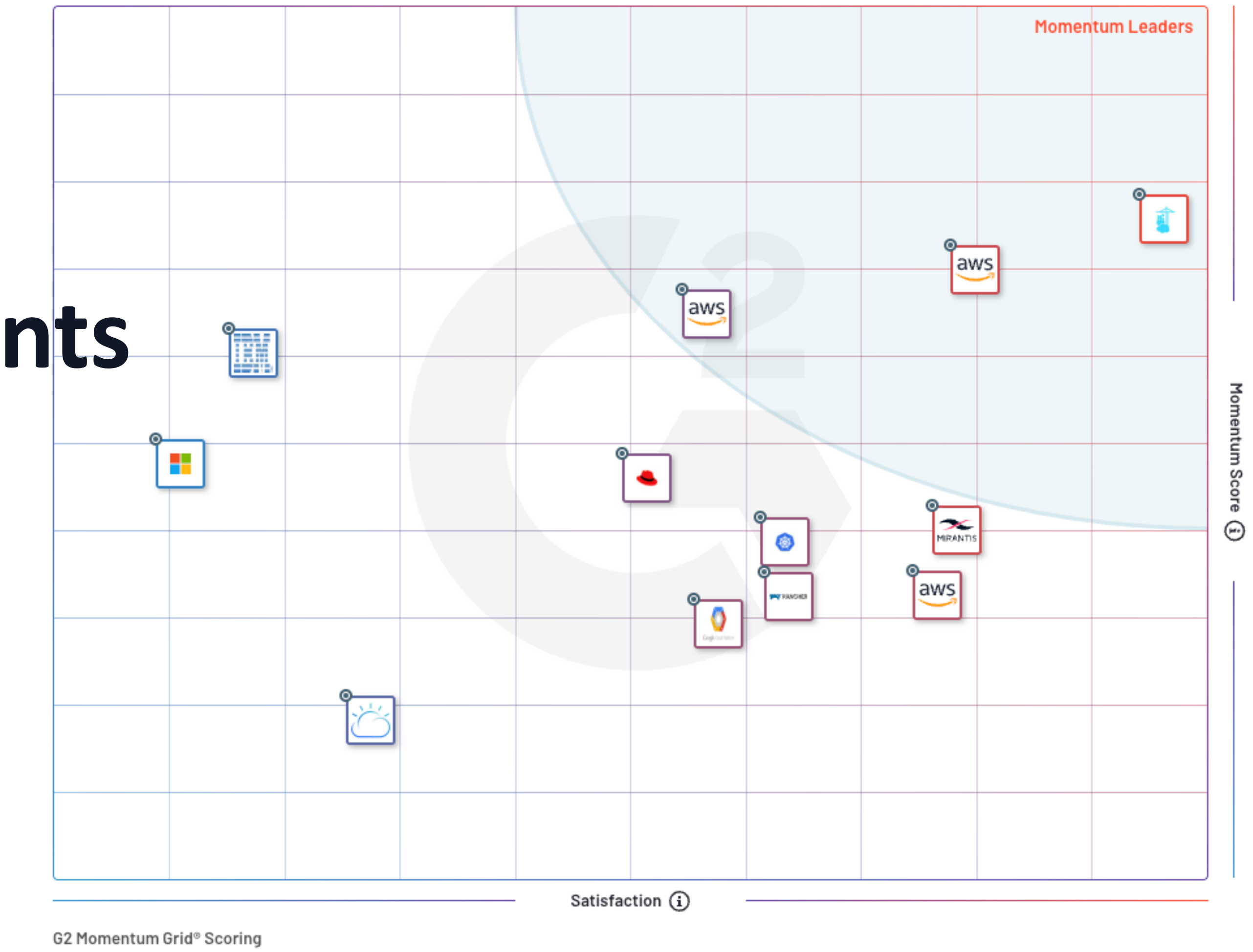
Restricted
I want to restrict the management of this resource to a set of users and/or teams

Actions

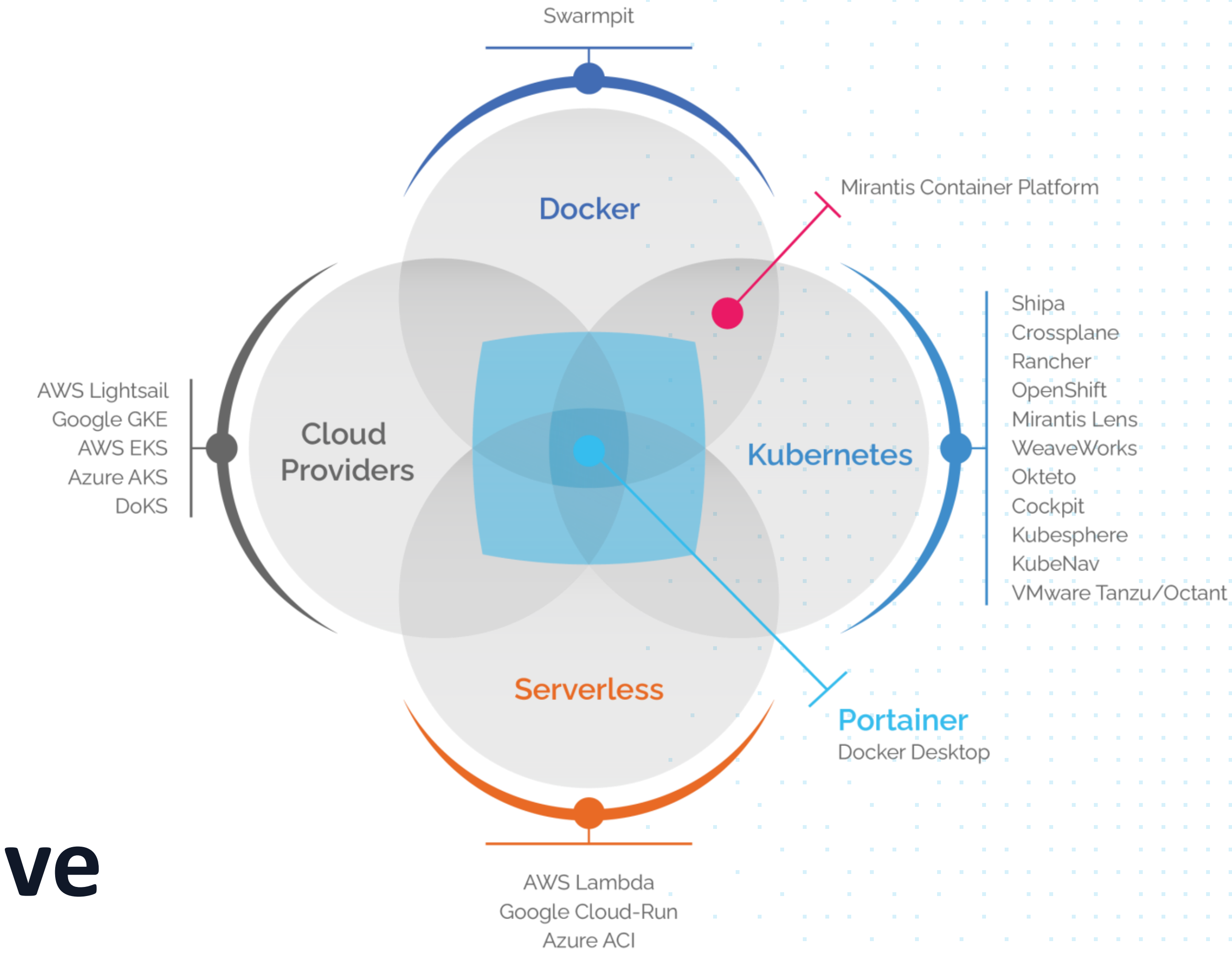
Command

Market participants

G2 Grid for
**Container
Management**

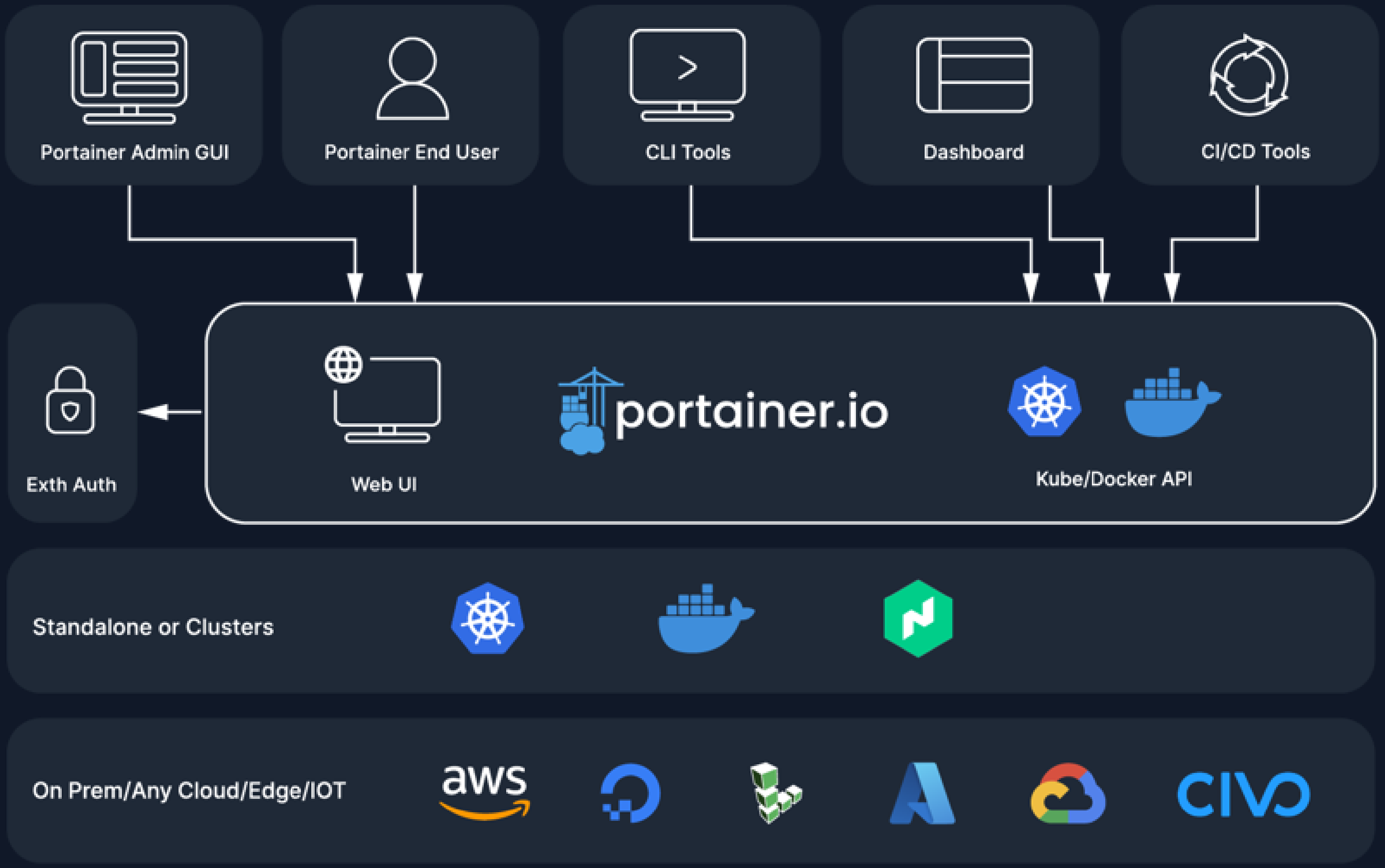


Outstanding recognition by the user community.
Recognized for vision and execution.

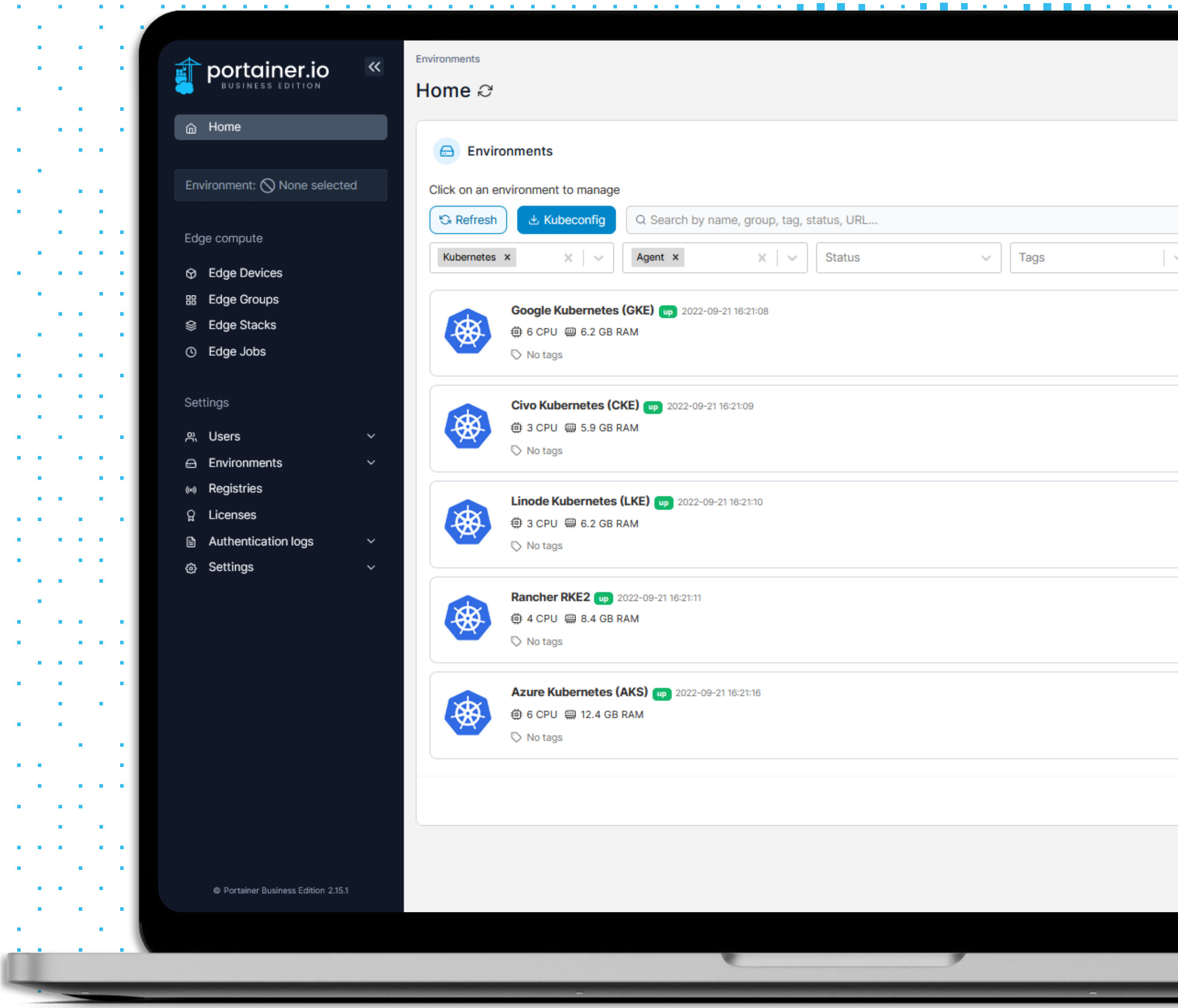


Competitive Landscape

The Portainer Platform



Architecture



Environments

Click on an environment to manage

Google Kubernetes (GKE) up 2022-09-21 16:21:08
6 CPU 6.2 GB RAM
No tags

Civo Kubernetes (CKE) up 2022-09-21 16:21:09
3 CPU 5.9 GB RAM
No tags

Linode Kubernetes (LKE) up 2022-09-21 16:21:10
3 CPU 6.2 GB RAM
No tags

Rancher RKE2 up 2022-09-21 16:21:11
4 CPU 8.4 GB RAM
No tags

Azure Kubernetes (AKS) up 2022-09-21 16:21:16
6 CPU 12.4 GB RAM
No tags

How is Portainer deployed & configured?

Self-hosted. Runs as a lightweight container inside your network

Manage anywhere from 1 – 45,000 environments

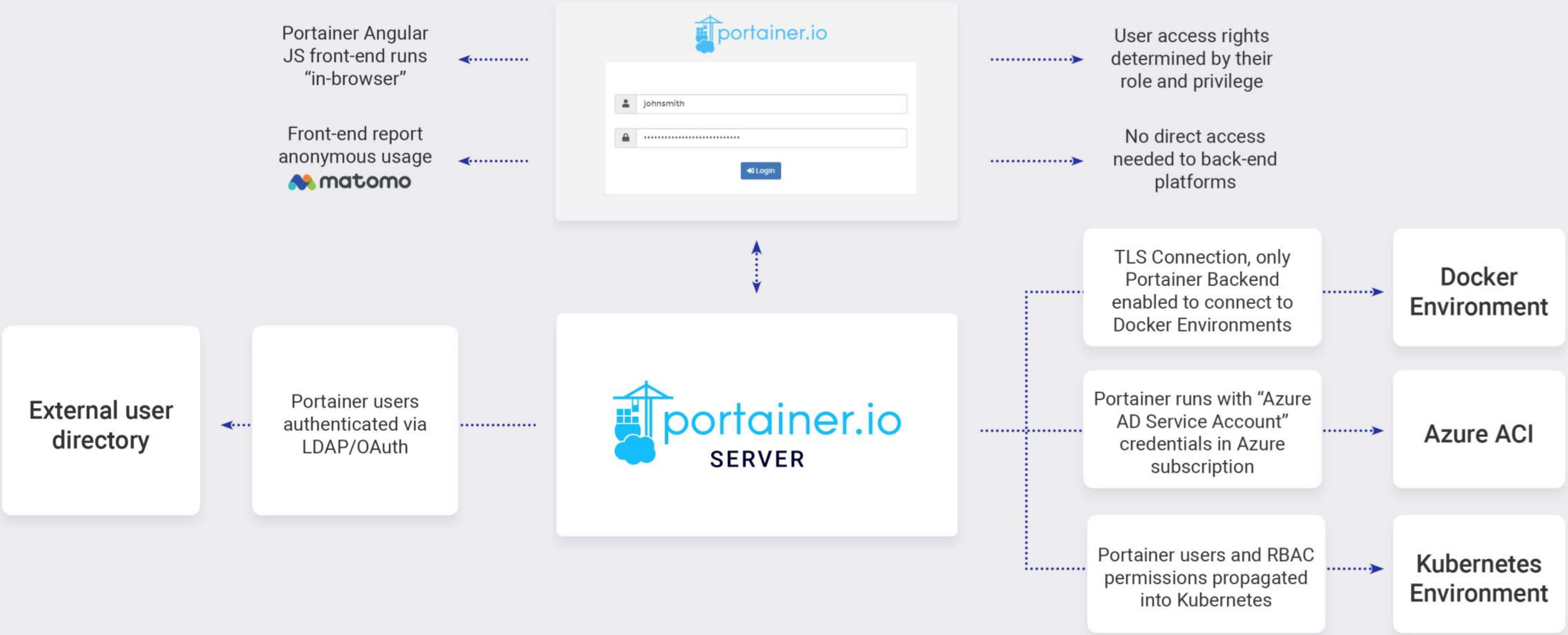
Import any container environment into Portainer

Connect Portainer to your Auth provider

Define teams, and role based access control rules centrally

Ensure users have no direct access to environments

Use Portainer as a proxy if 3rd party “Kubernetes Native” tools need to be connected



Portainer Angular JS front-end runs “in-browser”

Front-end report anonymous usage matomo

User access rights determined by their role and privilege

No direct access needed to back-end platforms

External user directory

Portainer users authenticated via LDAP/OAuth

portainer.io SERVER

TLS Connection, only Portainer Backend enabled to connect to Docker Environments

Docker Environment

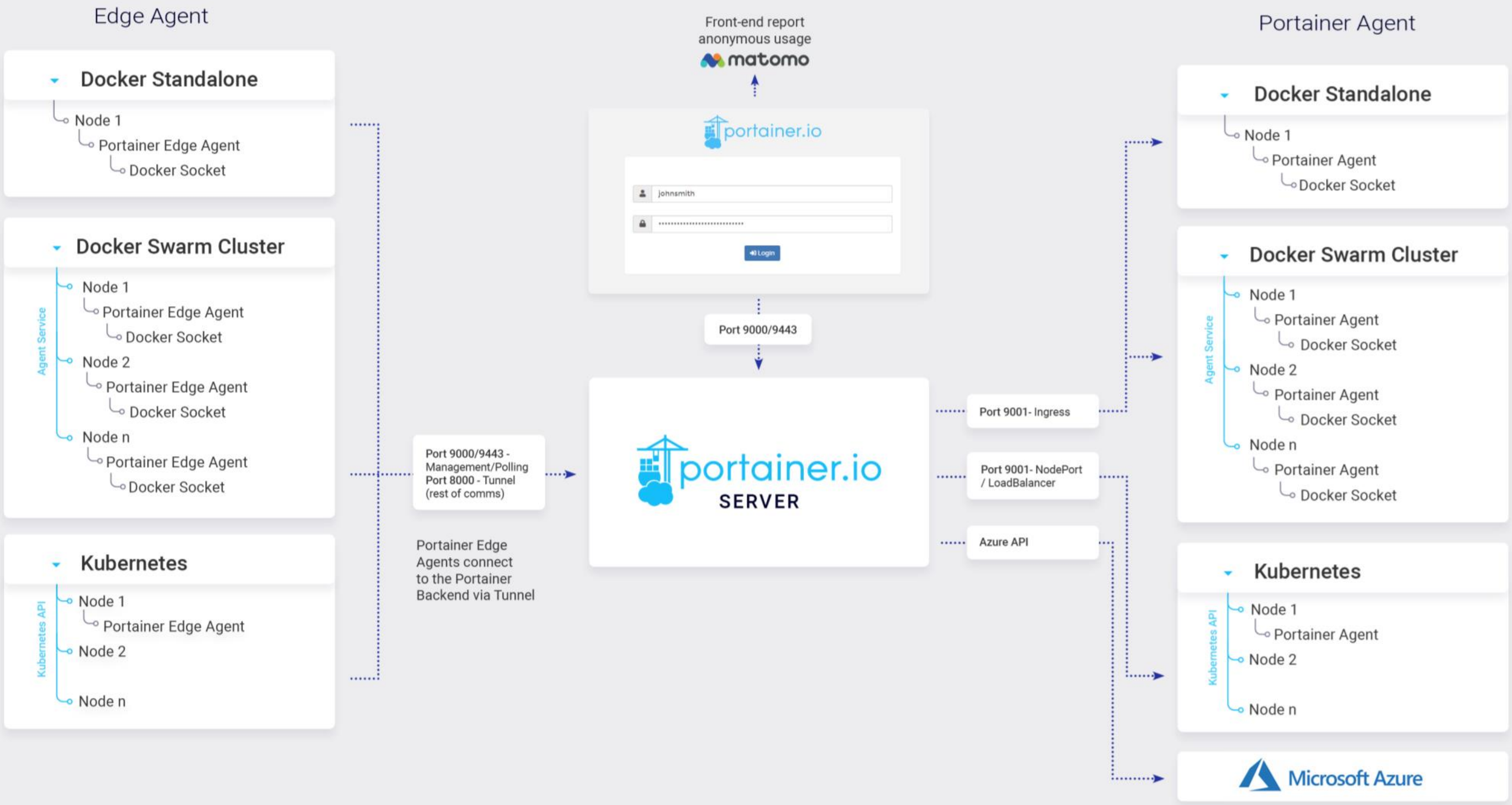
Portainer runs with “Azure AD Service Account” credentials in Azure subscription

Azure ACI

Portainer users and RBAC permissions propagated into Kubernetes

Kubernetes Environment

Portainer architecture - simplified



Edge Agent

Docker Standalone
- Node 1
  - Portainer Edge Agent
    - Docker Socket

Docker Swarm Cluster

Agent Service
- Node 1
  - Portainer Edge Agent
    - Docker Socket
- Node 2
  - Portainer Edge Agent
    - Docker Socket
- Node n
  - Portainer Edge Agent
    - Docker Socket

Kubernetes

Kubernetes API
- Node 1
  - Portainer Edge Agent
- Node 2
- Node n

Port 9000/9443 - Management/Polling
Port 8000 - Tunnel
(rest of comms)

Portainer Edge Agents connect to the Portainer Backend via Tunnel

Front-end report anonymous usage
matomo

portainer.io

johnsmith

Login

Port 9000/9443

portainer.io
SERVER

Port 9001- Ingress

Port 9001- NodePort / LoadBalancer

Azure API

Portainer Agent

Docker Standalone
- Node 1
  - Portainer Agent
    - Docker Socket

Docker Swarm Cluster

Agent Service
- Node 1
  - Portainer Agent
    - Docker Socket
- Node 2
  - Portainer Agent
    - Docker Socket
- Node n
  - Portainer Agent
    - Docker Socket

Kubernetes

Kubernetes API
- Node 1
  - Portainer Agent
- Node 2
- Node n

Microsoft Azure

Portainer architecture – in detail

portainer.io