Introduction

This questionnaire is the companion to the Portainer Container Operational Maturity Assessment and is to be used by the delivery consultant in a Q & A format with the customer.

Each question needs to be answered with a positive affirmation (YES), or negative (NO) based on the ability of the customer to answer the question. Partial competence should be answered as a NO.

Introduction

A container platform imparts a significant shift for any organization, bringing about changes in various aspects of operations, technology, and culture. To ensure a smooth transition and alignment with the operational maturity assessment questionnaire, companies need to prepare for these transformations:

1. Team Skills and Experience

a. Embracing Advanced and Basic Knowledge of Container Technologies
The assessment includes questions about Kubernetes and Docker proficiency. Companies should expect a learning curve in these areas:

- **Advanced Knowledge**: Skills in Kubernetes orchestration, including control planes, API servers, and autoscaling, as well as Docker Swarm and data persistence across nodes.
- **Basic Knowledge**: Understanding container orchestration basics, managing lifecycle, and constructing YAML manifests for Kubernetes, as well as Docker architecture and differences between VMs and containers.

Expectations: Teams will need training in these areas, potentially hiring new talent or upskilling existing staff to meet the operational demands of a containerized environment.

2. IT Operational Model Adjustments

a. Adopting Site Reliability Engineering (SRE) and Platform Engineering Practices
The questionnaire probes the adoption of SRE and platform engineering. Transitioning to these models involves:

- **SRE**: Focusing on reliability, uptime, and automation. Teams should be prepared to shift towards proactive monitoring and continuous improvement.
- **Platform Engineering**: Centralizing policy, control, and operations for container platforms, moving away from traditional IT silos.

Expectations: Organizational roles may change, with new responsibilities centered on automation and platform maintenance, requiring a culture shift towards preventive measures and centralized management.

3. Development and Deployment Practices

a. Moving Towards DevOps and Continuous Deployment
Questions in the assessment highlight the importance of DevOps adoption and centralized management of containerization:

- **DevOps**: Emphasizing the "You Build It, You Support It" approach, where development teams take ownership of the lifecycle of their applications.
- **Continuous Deployment**: Establishing pipelines for automated deployment and integrating containerization into everyday development practices.

Expectations: Teams must integrate new tools and practices for continuous integration and deployment, and foster a collaborative environment between development and operations.

4. Application Readiness and Modernization

a. Assessing Application Suitability for Containers

The questionnaire addresses different types of applications, from container-native to traditional software:

- **Container Native Applications**: Designed for horizontal scalability and microservice architecture.
- **Three-Tier and Monolith Applications**: Understanding the adaptability of existing applications to container environments.
- **Traditional Software**: Evaluating the feasibility of running legacy applications in containers.

Expectations: Companies need to evaluate their application portfolio to determine what can be containerized, requiring potential refactoring or re-architecting of applications to fit into a containerized model.

5. Platform Management and Tooling

a. Implementing Appropriate Management Tools

The assessment includes questions about platform management tooling, such as service meshes, observability tools, and centralized operations:

- **Advanced Tools**: Utilizing service meshes for microservice management, geo-distributed applications, and advanced security tools.
- **Centralized Operations**: Moving from discrete deployments to centralized management, standardizing on GitOps and ClickOps for operational control.

Expectations: Organizations should invest in and deploy robust management tools tailored to their container platform, ensuring proper visibility, security, and control over containerized environments.

6. Monitoring and Alerting Adjustments

a. Transitioning from Traditional to Container-Specific Monitoring

The questionnaire evaluates the use of traditional IT monitoring versus container-specific tools:

- **Container-Specific Monitoring**: Implementing real-time observability and historical monitoring tailored to container environments.
- **Traditional Monitoring**: Understanding the limitations of current tools and transitioning to solutions that support the dynamic nature of containerized applications.

Expectations: Teams need to adapt or replace existing monitoring and alerting systems with solutions that cater specifically to container metrics and performance.

Conclusion

Operating a container platform is a multifaceted transformation involving changes in team skills, operational models, development practices, application architectures, and tooling. Understanding the expectations outlined in the operational maturity assessment helps organizations prepare effectively for these changes, ensuring a successful transition to a containerized infrastructure. This readiness not only aligns with the assessment criteria but also sets the foundation for leveraging the full potential of container platforms.

Personal Skills and Experience Questionnaire

Each member of the IT Operations team, who are/will be responsible for supporting the container platform should answer this section. The goal is to determine the average skillset in the team, and to understand how many are at the higher levels, so as to ensure risk mitigation.

This section starts with the most complex technology first, so if the team are openly new to the technology, you can skip to lower levels of questions.

Advanced Kubernetes Knowledge	Yes/No
Do you understand the control-plane, API server, kubelet, networking, storage, deployments, service discovery, clustering, security, and monitoring in Kubernetes?	
Do you have a deep understanding of Kubernetes container orchestration?	
Have you worked with Kubernetes extensibility options such as CNI, CSI, ServiceMesh, Operators, Admission Controllers, and Policy Engines?	
Can you write comprehensive Kubernetes RBAC Roles and Cluster Roles?	
Do you manage secrets and configurations in Kubernetes?	
Have you worked with Kubernetes autoscaling (Pod and Cluster) and set per-namespace resource allocations (Quotas)?	
Do you extensively use the full array of Kubernetes CLI commands for deployment, troubleshooting, and lifecycle management?	
Can you write advanced Kubernetes manifests and HELM charts?	
Do you have a deep understanding of GitOps and policy management in Kubernetes?	

Basic Kubernetes Knowledge	Yes/No
Do you understand the basic concept of Kubernetes orchestration?	
Are you familiar with deployments, pods, containers, and services in Kubernetes?	
Do you understand how registry authentication works in Kubernetes?	
Do you understand the resource overhead of running Kubernetes?	
Have you built and managed the lifecycle of a Kubernetes cluster using bootstrap tooling?	
Do you know how to construct a basic Kubernetes YAML manifest?	
Do you have a basic understanding of Kubernetes CLI commands?	

Advanced Docker Knowledge	Yes/No
Have you managed performance and troubleshooting in distributed micro-services?	
Have you worked with Docker Swarm, and configured highly available clusters?	
Do you know how to maintain data persistence across nodes?	
Have you worked with overlay networking and tunnelling in Docker?	
Are you familiar with reverse proxies and ingress controllers?	
Do you know how to handle DNS and service discovery in Docker?	

Basic Docker Knowledge	Yes/No
Do you understand the architecture of a container and how a container runtime works?	
Do you know the differences between stateless and stateful containers?	
Do you understand how to manage data persistence in containers?	
Do you know the differences between VMs and containers?	
Do you understand how Docker volumes work and their differences from bind mounts?	
Can you write Docker Compose files and use Compose to manage groups of containers?	
Do you know how to build and distribute Docker images?	

Advanced Linux and Infrastructure Knowledge	Yes/No
Do you understand NFS/CIFS, block storage, and the difference between RWO and RWX file systems?	
Do you understand Network Bridging, NAT, VLAN, and overlay networking (VXLAN, encapsulation/tunnelling)?	

Basic Linux Knowledge	Yes/No
Can you configure an SSH server with SSH keys?	
Do you know how to install components or applications using Apt?	
Do you understand SSL/TLS?	
Can you configure iptables or IPFW?	
Do you frequently use standard Linux commands?	

IT Operational Model Questionnaire

Here we are trying to understand how IT is supported today, and if containers are already adopted, how mature the organization is in regards to modern IT service delivery models.

SRE (Site Reliability Engineering)	Yes/No
Do you understand the premise behind Site Reliability Engineering?	
Does your organization have a dedicated full-stack engineering team focused on SRE?	
Is your SRE team exclusively focused on preventative engineering, ensuring uptime/resilience, automation, and incremental improvements?	
Does your SRE team engage in proactive monitoring and remediation?	

Platform Engineering	Yes/No
Do you understand the core roles and responsibilities of a Platform Engineering team?	
Has your organization transitioned from a DevOps model to a Platform Engineering model?	
Does your organization recognize container platforms as business critical?	
Is there a centralized approach to policy, control, security, and operations for your platforms?	

DevOps Adoption	Yes/No
Does your development team follow the “You Build it, You Support it” approach to containerization?	
Is there centralized management of containerization within your organization?	
Are there discrete pockets of skills related to containerization in your organization?	
Does your development team have any exposure to infrastructure fundamentals?	
Does containerization underpin certain discrete production services?	

“Champion” Based Initial Adoption of Containerization	Yes/No
Was containerization initially adopted by a single developer or lead?	
Did the adoption of containerization “accidentally” transition to a production service?	
Are there operational controls and visibility for containerization?	
Was containerization adopted due to central IT’s reluctance to officially support it, historically known as “Shadow IT”?	

Traditional Platform Support Team	Yes/No
Is there a central platform team responsible for common platform components like cloud, virtualization, databases, web services, and API gateways?	
Does your organization offer IT as a service from a service catalogue?	
Do you have simple self-service offerings such as auto-provisioning?	
Does this team have any exposure to containerization?	

Traditional IT Operations	Yes/No
Does your organization have an infrastructure support team?	
Is there an application support team in your organization?	
Do you have a development team for internally developed software?	
Is service delivery in your organization primarily ticket-based?	

Application Readiness Questionnaire

Here we are trying to understand the types of applications that the business intends to run in Containers. There are likely to be many applications, so this section should be answered for each application.

Container Native 12 Factor Applications	Yes/No
Does your organization run, or plan to run, horizontally scalable applications in containers?	
Are your applications based on a 12-factor micro-service architecture?	
Are your applications stateless?	

Container Native Three-Tier Applications	Yes/No
Does your organization use ISV (Independent Software Vendor) provided container images and deployment manifests?	
Are container images and deployment manifests provided by your internal developers?	

Containerizable, Modern Three-Tier Applications	Yes/No
Does your organization develop three-tier (Web/Middleware/DB) applications that can be separated into an application stack running distributed across disparate servers without impacting performance?	
Do you use commercial off-the-shelf (COTS) software that can be repackaged to run within discrete containers distributed across a cluster of physical servers?	

Containerizable Monolith Applications	Yes/No
Does your organization develop monolith applications based on common frameworks/components with readily available container image bases?	
Do you use COTS software that can be repackaged to run within a single container?	

Traditional Installable Software	Yes/No
Does your organization use purchased software provided by a software vendor?	
Do you have self-developed software that is compiled and installed using an installer?	
Does your organization use Windows UI-centric applications?	

Platform Management Tooling Questionnaire

Here we are trying to understand the types of tooling already deployed within the landscape, and the appropriateness of this tooling to run in a container-centric environment.

Advanced Container Service Platform	Yes/No
Does your organization use a service mesh for managing microservices network access?	
Are your applications geo-distributed across multiple locations?	
Do you utilize advanced observability, security, and compliance tools?	

“Containers as a Service” Platform	Yes/No
Do you provide a 100% self-service portal for developers?	
Are there predefined “Golden Paths” for developers to follow?	
Do you manage applications across multi-cluster, multi-cloud, or hybrid-cloud environments?	
Is there a capability for rapid lifecycle management of clusters?	
Is infrastructure as code (IaC) a standard practice in your organization?	
Do you use GitOps as the preferred deployment model?	
Do you have a full observability suite providing real-time and historical monitoring for a wide array of metrics?	

Centralized Container Operations Platform	Yes/No
Has your organization migrated from discrete deployments to a central deployment model?	
Have operations been migrated from development to a platform engineering team?	
Is there centralized access, authorization, policy, and security management?	
Do you use a single deployment model (cloud or on-prem)?	
Do you operate with a very small number of shared environments (one or a few)?	
Are clusters treated as “cattle” rather than “pets”?	
Do you combine GitOps and ClickOps for operations?	
Do you have basic observability with real-time monitoring for a limited range of metrics?	

Discrete Container Operations	Yes/No
Are containers deployed in production primarily led by developers?	
Do you have discrete, per-project deployments rather than a common/shared platform?	
Do you use bootstrap tools like k3s, RKE, or AKS/GKE/EKS for environment creation?	
Is there no central management and control of container environments?	
Are environments treated as “pets” rather than “cattle”?	
Is management primarily done through ClickOps or a UI-centric approach?	

Container Experimentation	Yes/No
Are containers used primarily in dev environments or on developer laptops?	
Do you use tools like Docker Desktop, Podman Desktop, or MiniKube?	
Are containers used in non-critical deployments?	
Is container management distributed with no central control?	
Are discrete monitoring tools used?	
Is management primarily done via CLI or UI-centric tools?	

Traditional IT Monitoring and Alerting Tooling	Yes/No
Does your organization use VM monitoring tools?	
Do you use application performance monitoring tools?	
Is service availability monitoring implemented?	
Do you use threshold-based alerting?	
Are SYSLOG servers used for log management?	

Determining the Level of Operational Maturity Needed

Here we want to try and ascertain the level of Operational maturity needed, based on the types and criticality of applications to be run in the container platforms.

Advanced	Yes/No
Are the vast majority of your applications running or planned to run in containers?	
Is there an absolute requirement for “five 9’s” (99.999%) of availability?	

Capable	Yes/No
Are multiple business-critical applications running or planned to run in containers?	
Do you need to deliver high levels of SLA for these applications?	

Emerging	Yes/No
Are multiple business-important applications running or planned to run in containers?	
Are these applications customer-facing or internal business services?	

Opportunistic	Yes/No
Are one or more non-critical applications running or planned to run in containers to validate value and operational impact?	

Ad-Hoc / Experimentation	Yes/No
Is your organization in the early stages of experimenting with the business value of containers?	
Are there no customer-facing or internal-facing applications in this experimentation phase?	
Is the use of containers currently on a trial basis?	

Traditional	Yes/No
Is your organization just beginning to think about containerization?	
Is there no current adoption of containerization?	

Next Steps

We will now use this information collected to assess your current and required maturity level, this will be mapped to the Portainer Container Operational Maturity model.