Architecture Decision Record: Docker Swarm Container Orchestration

Decision Number: 001

Decision Maker: [Your name or position]

Date: [Date of decision]

Context

We are considering different container orchestration tools to manage our microservices-based architecture. We have evaluated different solutions like Kubernetes, Docker Swarm, and Mesosphere DC/OS. However, we have decided to focus on Docker Swarm due to its simplicity, integration with Docker, and built-in load balancing.

Decision

We have decided to use Docker Swarm as our container orchestration tool. Docker Swarm provides a simple and intuitive way to manage containerized applications across a cluster of nodes. It also allows us to leverage our existing Docker-based workflows and infrastructure. With Docker Swarm, we can easily deploy, scale, and manage our applications, all while taking advantage of built-in load balancing.

Benefits

- **Simplicity:** Docker Swarm follows the same principles as Docker, so there's no need to learn a new technology. The learning curve is relatively shallow for developers familiar with Docker.

- **Integration:** Docker Swarm integrates seamlessly with Docker tools, like Docker Compose, making it easier to manage all our containers and services from one place.

- **Load balancing:** Docker Swarm provides built-in load balancing, ensuring that our applications are always available and evenly distributed across the cluster.

- **Scalability:** Docker Swarm makes it easy to scale our applications horizontally by adding or removing nodes from the cluster.

- **High availability:** Docker Swarm automatically distributes our services across nodes, providing high availability in case of node failure.

Risks

- **Limited functionality:** Docker Swarm may lack some of the advanced features found in Kubernetes or Mesosphere DC/OS, like automatic scaling or self-healing.

- **Docker-centric:** Docker Swarm is tightly coupled with Docker, which may limit our flexibility if we ever need to move away from Docker-based solutions.

- **Immaturity:** Docker Swarm is still a relatively new technology, and there may be some stability issues or gaps in documentation.

Alternatives

- **Kubernetes:** Kubernetes is the most widely-used container orchestration platform and provides advanced features and a more mature ecosystem. However, it has a steeper learning curve and may be overkill for our needs.
- **Mesosphere DC/OS:** Mesosphere DC/OS is a powerful tool that provides advanced features like multi-cloud support and native big data and AI platform capabilities. However, it requires significant expertise to implement and may be too complex for our requirements.

Conclusion

After careful consideration, we have decided to use Docker Swarm as our container orchestration tool. Docker Swarm provides the simplicity, integration, and built-in load balancing we need to manage our containerized applications. Although it may lack some advanced features, we believe Docker Swarm's benefits outweigh its risks for our current requirements.

Credit: this page is generated by ChatGPT, then edited for clarity and format.