K8S 01: Kubernetes. Basics. Введение в Kubernetes

Для начала попробуем ответить на вопрос - зачем нужны оркестраторы контейнеров, если есть ssh + docker-compose up? Нужны всего две команды, это просто.
Чем больше растет количество разворачиваемых приложений, тем труднее становится управлять ими. Конечно, можно обойтись bash-скриптами или Ansible, но полное обновление или оптимизацию ресурсов выполнить с помощью указанных инструментов крайне сложно или невозможно.

Развитие инфраструктурных решений и интересы бизнеса, который задает тенденции этого развития, выдвигают требования к ПО, которое должно обладать следующими функциями:

- отказоустойчивость,
- масштабирование, автоматическое и по запросу,
- оптимальное использование имеющихся ресурсов,
- автоматическое обнаружение новых установленных приложений и предоставление к ним доступа,
- предоставление доступа внешним пользователям,
- обновление и откат изменений для установленных приложений.

Раньше все это делали кластерные системы, такие, как Pacemaker/Corosync. Эти системы сложны и не имеют унифицированного подхода к адаптации приложения. С появлением контейнеризации данные системы и их аналоги оказались не готовы работать с контейнеризированными приложениями, в связи с чем утратили востребованность. Ведь появилось несколько систем оркестрации, ориентированных именно на работу с контейнеризированными приложениями:

- docker swarm,
- kubernetes,
- mesos,
- nomad.

Такие системы существенно упрощают работу с большим числом контейнеров.
21 июля 2015 года выпущена версия Kubernetes 1.0, после чего компания Google, в партнерстве с Linux Foundation, организовала специальный фонд Cloud Native Computing Foundation (CNCF), которому корпорация передала Kubernetes в качестве начального технологического вклада.

История системы растет с проприетарных Borg и Omega, информацию о которых без труда можно найти в интернете.

Kubernetes позволяет запускать приложения на тысячах узлов. Из-за того, что данные приложения работают в контейнерах, они не влияют на работу друг друга. Приложение становится отделенным от базовой инфраструктуры: сервер, сеть, хранилище. Это позволяет упростить и ускорить развертывание новых приложений. Процедура разворачивания всегда одинакова и не зависит от количества узлов в кластере.

Полезные ссылки:

- Wikipedia - Kubernetes
- Kubernetes: что это?
- The Illustrated Children's Guide to Kubernetes
- Kubernetes vs. Docker: It's Not an Either/Or Question
- Container Orchestration: Dealing with Many Containers
- TOP 3 Container Orchestration Engines in Kubernetes

Задание:

1. Какие из перечисленных сервисов являются системами оркестрации контейнеров?
    1. docker swarm
    2. kubernetes
    3. docker-compose
    4. mesos
    5. Все перечисленные
    6. Ни один из перечисленных
2. Какая компания является создателем Kubernetes?
    1. Google
    2. Yahoo
    3. AWS
    4. Linux Foundation
    5. Cloud Native Computing Foundation
3. Есть ли, на Ваш взгляд, нечто такое, после чего система оркестрации обязательна, а docker-compose уже не выполняет заложенные в проект требования?