ВАРИАНТЫ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ КОНТЕЙНЕРИЗАЦИИ
OPTIONS FOR CONTAINERIZATION TECHNOLOGY

М.С. НАГОВИЦИН – магистрант, Институт информационных технологий и радиоэлектроники, кафедра ВТиСУ, группа ВТм-119, E-mail: NMS.T@yandex.ru

К.В. КУЛИКОВ – научный руководитель, к.т.н., доцент, Институт информационных технологий и радиоэлектроники, кафедра ВТиСУ, E-mail:kulikov@vlsu.ru

M.A. NAGOVITSIN – undergraduate, Vladimir State University, E-mail: NMS.T@yandex.ru

K.V. KULIKOV – candidate of technical sciences, docent, Vladimir State University, E-mail:kulikov@vlsu.ru

Аннотация: В статье представлен обзор современных технологий контейнеризации и их использования. Произведено сравнение двух существующих технологий виртуализации систем: Контейнеризация и Виртуальные машины, рассмотрены их основные характеристики, преимущества и недостатки контейнеризации. Проведен анализ проблем и перспектив применения данной технологии.

Abstracts: The article provides an overview of modern containerization technologies and their use. Comparison of two existing systems virtualization technologies: Containerization and Virtual Machines, their main characteristics, advantages and disadvantages of containers are considered. The analysis of the problems and prospects of the use of this technology.

Ключевые слова: контейнеризация, контейнер, виртуализация, виртуальная машина.

Keywords: containerization, container, virtualization, virtual machine.

На данный момент множество компаний, работающие с виртуализацией, заинтересованы в использование контейнерных технологий. К таким компания относится Google, IBM, Microsoft и другие. Выбор основан на том, что данный вид виртуализации, при помощи контейнеризации, выгоднее в плане расходов ресурсов. Так же технология контейнеризации довольно гибкая, что дает ей вариативность применения.

Одна из самых основных возможностей технологии – это упаковка образа со всем не обходимым для ее запуска, библиотеки, среда и другое обеспечивающие самодостаточную работу в изолированности. Именно данный вариант широко распространен среди специалистов, так как этот вариант решает проблему зависимостей в разных окружениях. Например, разработчик работает на свой собственной машине используя нужное окружения, но в дальнейшем окружение измениться, что рано или поздно приведет к сбоям и ошибкам.

Еще технология сокращает время разработки, а после упрощает управление производством. Это возможно благодаря легкости в настройках и изменение конфигурации. Плюсом так же является возможность хранения истории изменение объектов вместе с кодом приложения и удобными инструментами автоматическое размещение, координация и управление сложными компьютерными системами и службами, позволяющим быстро масштабировать инфраструктуру.

Кроме того, фактическое отсутствие привязки контейнеров к хостинговой платформе даёт огромную гибкость при выборе или смене провайдера — вы можете запускать их без принципиальных отличий в конечном результате на личном компьютере, bare metal серверах и в облачных сервисах.

Для понимания важно понимать отличия от подобных технологий, чтобы увидеть, насколько обширны варианты использования контейнер-

ных технологий. На данный момент существует только один аналог, схожий по концепции с технологией контейнеризации – это виртуальные машины. Первое существенное отличие контейнеров от виртуальных машин является то, что контейнер ограничен в пространстве хостовой операционной системы, а виртуальная машина представляет собой полноценную гостевую операционную систему со всеми нужными компонентами для ее работы. Из этого образуются отличия, имеющие практическое значение:

Контейнеры требуют значительно меньше ресурсов для своей работы, что положительно сказывается на производительности и бюджете.

Контейнеры можно запускать только в той же операционной системе, что стоит на host-системе — то есть запустить Windows-контейнер на host-системе с Linux не получится (на персональных устройствах это ограничение обходится с помощью технологии виртуализации). Однако это не относится к разным дистрибутивам одной и той же ОС, например Ubuntu и Alpine Linux.

Контейнеры предоставляют меньшую степень изоляции, поскольку используют ядро host-системы, что потенциально создаёт бóльшие риски в эксплуатации при небрежном отношении к безопасности.

Так же, еще один основной момент при работе с контейнерами – это введение в производство контейнерного приложения, а именно как в конечном итоге будет оно запущенно. Существует два основных варианта на сегодняшний день. Первым является «Свое железо» (bare metal). Данный вариант является консервативным. При этом выборе нужно подготовиться и заняться закупкой оборудования или его арендой, нанять специалистов. После этого можно начать запуск контейнеров. Очевидные плюсы этого варианта являются полный контроль над инфраструктурой, тонкая настройка и, как следствие, превосходящая производительность с более высокой отдачей от бюджета в долгой перспективе. Из минусов можно

подчеркнуть большие финансовые и временные затраты при старте. Этот вариант подходит больше для тех, кто обладает выверенными долгосрочными планами. Второй вариант – это «облачные решения» (SaaS). На сегодняшний день рынок предоставления подобных услуг огромен. Данный метод распространения контейнерных приложений выгоден и удобен в коротко срочной перспективе. Так же плюсом является индивидуальная настройка «облачных серверов» под себя.

По итогам можно подвести, что Контейнеры – это абстракция, они реализуются благодаря возможностям ядра Linux: namespaces и cgroups.

Существует большое количество реализаций контейнеров: Docker, Rocket, LXC, OpenVZ и другие.

Реализация концепции контейнеров возможна за счет того, что разработчики ядра предоставляют дополнительный уровень абстракции над глобальными ресурсами системы, такими как ID процесса, сетевой стек, имя пользователя и другими.

У контейнерных технологий есть большой выбор использования, есть преимущества и недостатки, так что и использования такой технологии зависит от поставленных задач.