

Практычнае вывучэнне сродкаў кантэйнернай віртуалізацыі і платформы Kubernetes

Касцюк Д.А., Луцюк П.М.

Брэсцкі дзяржаўны тэхнічны ўніверсітэт
dmitriykostiuk@gmail.com

The experience of developing a training course dedicated to the basics of the Kubernetes container orchestrator is presented, which includes the principles of container virtualization, architectural features and key components of the Kubernetes platform, deploying and scaling applications based on it. The structure of the practical part of the course, measures to ensure the operability of the studied software in the conditions of an isolated network segment and limited Internet access are given, and ways to overcome the high entrance threshold of Kubernetes are discussed.

Кантэйнерызацыя - істотная частка актуальнага падыходу да сістэмнай інжынерыі, што прадугледжвае інтэграцыю распрацоўкі і эксплуатацыі праграмных сістэм пры пабудове складаных прыкладанняў. У аснове ляжыць выкарыстанне комплексу такіх сучасных праграмных тэхналогій, як віртуалізацыя, аўтаматычны аўдыт і кантроль прадукцыйнасці, рашэнне задач з ужываннем гетэрагенных сістэм, якія ўключаюць некалькі розных платформаў.

Адным з характэрных сродкаў рэалізацыі дадзенага падыходу з'яўляецца Kubernetes - складанакампанентная сістэма аркестрацыі кантэйнераў з адкрытым зыходным кодам, якая развіваецца Google.

Аб'екты Kubernetes разгортваюць і маштабуюць прыкладання на аснове патрабаванняў да памяці, ЦП і інш. Пры гэтым падтрымліваецца любая сістэма кантэйнернай віртуалізацыі, якая задавальняе Open Container Initiative (OCI). Kubernetes забяспечвае маштабаванне і балансаванне нагрузкі, аўтаматычнае выяўленне сэрвісаў і кіраванне сакрэтамі. Пры гэтым кантэйнеры, якія з'яўляюцца кампанентамі аднаго або некалькіх прыкладанняў, ізаляваныя адзін ад аднаго, калі толькі распрацоўніку/devops-інжынеру не патрабуецца адваротнага.

Платформа бярэ на сябе забеспячэнне празрыстага ўзаемадзеяння кантэйнераў і аўтаматычнага выяўлення сэрвісаў, а таксама прадастаўляе адпаведныя магчымасці праз API.

Праблема Kubernetes - высокі парог асваення. З аднаго боку прычына ў дастаткова вялікім наборе сутнасцяў гэтай платформы, дзякуючы якім кластар на базе Kubernetes дзейнічае як адзінае цэлае. З іншага боку, колькасць кампанентаў таксама вялікая, і некаторыя з іх з'яўляюцца заменнымі. Універсальны кластар, які аўтаматычна выконвае абнаўленні, маштабаванне і самаўстаноўку, дэманструе дастаткова высокую канцэ-

птуальную складанасць.

Платформа вельмі папулярная, але яе выкладанне студэнтам - гэта часта праблема.

Мы распрацавалі курс па вывучэнні Kubernetes студэнтамі старэйшых курсаў, ужо знаёмымі з патрэбнымі тэхналогіямі, у тым ліку кампутарнымі сеткамі, элементамі сістэмнага адміністравання GNU/Linux, архітэктурай кліент-серверных прыкладанняў і інш. Гэта курс для першай ступені вышэйшай адукацыі, і ён мяркуе далейшае развіццё ў рамках другой ступені. База тэарэтычнага матэрыялу - афіцыйная дакументацыя Kubernetes [1], але лабараторны практыкум ахоўвае толькі частку тэарэтычнага матэрыялу, у выніку чаго студэнтам патрабуецца мець справу з меншай колькасцю аб'ектаў платформы.

Практычная частка курса пабудаваная на аснове Minikube - спецыялізаванага дыстрыбутыва Kubernetes, прызначанага для разгортвання на лакальнай машыне з сумяшчэннем апаратнай і кантэйнернай віртуалізацыі [2]. Minikube мае шэраг абмежаванняў, звязаных у першую чаргу з яго лакальнай прыродай. Гэта не з'яўляецца праблемай, паколькі практычная частка курса не закранае шэраг несумяшчальных аспектаў платформы: такія тэмы як, напрыклад, рэалізацыя афінітэту і антыафінітэту, выкарыстанне аператараў, разглядаюцца толькі ў тэарэтычнай частцы.

З праграмных залежнасцяў патрабуецца толькі VirtualBox з стандартных рэпазітараў Linux; астатняе выкачваецца з сеткі пры ўсталёўцы:

- Minikube выкачвае гатовы ўстаноўачны вобраз з Docker і кампанентамі Kubernetes, а таксама актуальныя версіі kubelet і kubeadm;
- далей падчас разгортвання прыкладання ў кластары Kubernetes кантэйнеры прыкладання выкачвае ўжо сам Docker.

У сувязі з гэтым генеруецца вялікі сеткавы трафік (бінарныя файлы minikube-linux-amd64 і kubect1, вобраз віртуальнай машыны minikube-v * .iso, а таксама вобразы кантэйнераў маюць агульны аб'ём больш за 600 Мб), і таму адной з задач для арганізацыі практыкума быў запуск Minikube ў ізаляваным сегменце лакальнай сеткі, ва ўмовах часткова абмежаванага доступу да знешніх Інтэрнэт-рэсурсаў.

Мы разгледзелі і апрабавалі два варыянты лакальнага выкарыстання Kubernetes:

- стварэнне прыватнага рээстра Docker і перанастройка сістэмы на яго эксклюзіўнае выкарыстанне, а таксама разгортванне прыватнага рэпазітара вобразаў для Minikube;
- запуск Minikube і пробнае разгортванне навучальных прыкладанняў праз Інтэрнэт з наступным кланаваннем атрыманага профілю на працоўныя станцыі навучальнага класа (вырашаюцца пытанні як разгортвання вобразаў для Minikube, так і імпорт вобразаў Docker праз файлавую сістэму замест іх атрымання з сеткі).

Пасля експерыментаў з абодвума варыянтамі мы спыніліся на другім, так як ён апынуўся лягчэй, з улікам ужо дзеючай сістэмы тыражавання вобразаў для працоўных станцый [3].

Пры гэтым варта адзначыць, што нягледзячы на стандартную палітыку абнаўлення кантэйнераў Kubernetes IfNotPresent (спампоўваць вобраз кантэйнера ў выпадку яго адсутнасці ў лакальным кэшы), на практыцы спатрэбілася ўсталёўваць яе значэнне Never, каб цалкам забяспечыць выкарыстанне менавіта лакальных вобразаў без абнаўлення.

Структура распрацаванага практыкума для студэнтаў першай ступені ўключае 4 лабараторныя работы. Першая носіць азнаямленчы характар: у ёй распавядаецца аб Minikube і сістэмах віртуалізацыі, якія ён можа выкарыстоўваць на розных апаратных платформах, а таксама пра асаблівасці ўстаноўкі ў выпадку выкарыстання на ўласных прыладах. У другой лабараторнай рабоце разглядаюцца асаблівасці доступу да кластару Kubernetes з дапамогай каманднага радка (kubectl), уключаючы мінімальнае канфігураванне доступу (kubectl proxy), і вэб-інтэрфейсу (Kubernetes Dashboard). У трэцяй выконваецца разгортванне мінімальнага вэб-прыкладання ў кластары, уключаючы праверку створаных Deployment, ReplicaSet і Подаў, а таксама Service з доступам праз NodePort. Чацвёртая дазваляе вывучыць разгортванне шматкампанентнага прыкладання з гатовай кліенцкай часткай на базе вэб-тэхналогій і з сервернай падсістэмай на аснове СКБД MongoDB.

Спіс выкарыстаных крыніц:

1. Kubernetes Documentation. <https://kubernetes.io/docs/home/>.
2. Hello Minikube. <https://kubernetes.io/docs/tutorials/hello-minikube/>.
3. Пойта П.С., Костюк Д.А., Дереченник С.С., Луцок П.Н. Повышение сетевой безопасности в компьютерном парке вуза за счет буферизации и изоляции ресурсов // Электроника инфо. – 2013. – №.6 (96). – С. 111–113.