

VAGRANT

Konfiguracja maszyny wirtualnej



ZANIM ZACZNIESZ!

Zweryfikuj czy masz włączoną sprzętową wirtualizację wchodząc w menadżer zadań, wydajność, klikając na CPU i weryfikując pole wirtualizacja.





ZANIM ZACZNIESZ!

Jeżeli wirtualizacja jest wyłączona musisz znaleźć stosowne pole w swoim BIOS/UEFI (Zazwyczaj pod nazwą virtualization – różne dla każdego dostawcy) i ustawić w trybie enabled.

Advanced Processor Configurat	ion	Item Specific Help
CPU Mismatch Detection: Core Multi-Processing: Processor Power Management:	[Enabled] [Enabled] [Disabled]	When enabled, a UMM (Virtual Machine Monitor) can utilize
Execute Disable Bit:	[Enabled]	capabilities provided by Vanderpool
Adjacent Cache Line Prefetch: Hardware Prefetch:	[Disabled]	Technology.
Direct Cache Access	[Disabled]	If this option is changed, a Power Off-On sequence will be
Set Max Ext CPUID = 3	[Disabled]	applied on the next boot.

My Favorites	Main	Ai Tweaker	Advanced	Monitor	Boot Tool	Exit	🔄 Hardw	are Monito
L3 Cache					6 MB		CPU	
L4 Cache					Not Supported			
Intel Adaptive The	ermal Monito				Enabled	- 1	3500 MHz	27°C
Active Processor (Cores				All	-	100.0 MHz	1.024 V
Limit CPUID Maxi	mum				Disabled	-		
							35x	
Execute Disable B	it				Enabled	-		
Intel Virtualization	n Technology		- 611	-	Enabled	-	Memory	
			190	IMALOT			Frequency	
Hardware Prefetc	her(L2 Cache)			Enabled		1600 MHz	1.500 V
Adjacent Cache Li	ne Prefetche	r			Enabled	-	Capacity	
					-		8192 MB	
Boot performance	e mode				Max Non-Turbo Pe	erformance 👻		
Dynamic Storage	Accelerator				Disabled	-	Voltage	
							+12V	
 CPU Power Manag 	ement Confi	guration					12.000 V	5.080 V
Intel Virtualization T	achnologyma	akos a single susten	appear as multip	la independent	curtame to Structor	This allows for multiple	+3.3V	
independent opera	ting systems to	o be running <u>simu</u>	taneously on <u>a sin</u>	gle system.	systems to software.	This allows for multiple	3.328 V	

ZACZYNAMY

Na początek wejdź na stronę VirtualBox i pobierz klienta odpowiedniego dla Twojego systemu operacyjnego:

https://www.virtualbox.org/wiki/Downloads

VirtualBox 6.1.8 platform packages

- Grader Brite Books
 Grader Books
 <l
- G⇒OS X hosts
- Linux distributions
- G⇒Solaris hosts





VIRTUALBOX

Zainstaluj VirtualBox. Na panelu po lewej możesz wybrać folder instalacji. Pamiętaj również, że VirtualBox może chwilowo zrestartować połączenie sieciowe, więc zapisz wszystkie aktywne zadania. Korzystaj z przycisków next, yes i install, aby kontynuować.





VAGRANT



Następny krok to instalacja Vagranta. W tym celu wejdź na stronę poniżej. Pamiętaj, aby wybrać odpowiednią platformę oraz wersję (32/64 bit).

https://www.vagrantup.com/downloads

MAC OS X WINDOWS	B LINUX DEBIAN	CENTOS
	Windows 64-bit ~	Download



VAGRANT



Podobnie jak z VirtualBox wykorzystaj przyciski next i install, aby zainstalować Vagranta. Po tym kroku musisz zrestartować komputer.



GIT

Trzecie i ostatnie narzędzie do zainstalowania to Git. Możesz skorzystać też z wersji przenośnej (Portable), która nie wymaga instalacji, a tylko pobrania.

https://git-scm.com/download/win

Other Git for Windows downloads

Git for Windows Setup 32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition") 32-bit Git for Windows Portable.

64-bit Git for Windows Portable.





LOKALIZACJA DEMO

Przejdź do folderu z Gitem (niezależnie czy zainstalowanym, czy przenośnym) i uruchom konsolę git-cmd. Konsola, domyślnie, otworzy się w Twoim katalogu użytkownika.

Aby uruchomić nasze demo, potrzebujesz przynajmniej 20 GB wolnego miejsca na dysku. Jeżeli posiadasz tyle na dysku C, możesz pozostać w tym katalogu, jeżeli nie to przejdź do wybranego folderu na innym dysku: <litera dysku>:, a następnie cd <ścieżka\do\folderu>.

📙 bin	20.04.2020 18:10	Folder plików	
🔤 📊 cmd	01.06.2020 13:41	Folder plików	
dev 🔒	20.04.2020 18:10	Folder plików	
etc	20.04.2020 18:10	Folder plików	
mingw64	20.04.2020 18:10	Folder plików	
📕 tmp	20.04.2020 18:10	Folder plików	
usr	20.04.2020 18:10	Folder plików	
🚸 git-bash	20.04.2020 17:34	Aplikacja	150 KB
🔷 git-cmd	20.04.2020 17:34	Aplikacja	149 KB
	20.04.2020 18:10	Dokument tekstowy	19 KB
🗋 magic.py	20.04.2020 05:11	Plik PY	9 KB
💿 post-install	20.04.2020 18:10	Plik wsadowy Win	2 KB
README.portable	20.04.2020 18:10	Plik PORTABLE	4 KB

C:\Users\XXX\Vagrant>cd d:\Panda

C:\Users\XXX\Vagrant>d:

d:\Panda>



POBRANIE REPOZYTORIUM

Teraz musisz pobrać 2 repozytorium z github na swój komputer. Zrobisz to za pomocą komend:

git clone https://github.com/Panda-Academy-Workshops/Vagrant

git clone https://github.com/Panda-Academy-Workshops/Minikube

Wejdź do katalogu z repozytorium:

cd Vagrant

W konsoli, będąc nadal w katalogu Vagrant, wpisz komendę vagrant up.

Otwórz drugie okno konsoli git-cmd, a następnie wejdź do katalogu Minikube używając komendy cd i uruchom drugą maszynę używając vagrant up.

Jeżeli z jakiegoś powodu, maszyna budując się zwróci błąd – wykonaj w katalogu komendę vagrant provision. Uruchomisz ponownie Ansible (wykonane już kroki zostaną pominięte). Jeżeli to nie pomoże – skontaktuj się z nami na Discordzie ©.



URUCHOMIENIE MASZYNY



Teraz musisz dłuuuuuuugo poczekać, aż komenda skończy pracę (może to potrwać nawet godzinę, w zależności od łącza internetowego i dostępności serwerów!).

Jeżeli będzie ci się wydawać, że budowa zawiesiła się na jakimś kroku, a w szczególności poniższym:

Prawdopodobnie tak nie jest ⁽²⁾.

Czekaj cierpliwie na zakończenie i dostępność linii komend.



URUCHOMIENIE MASZYNY

Po uruchomieniu vagranta Minikube, w ostatnim kroku wykonywania się Ansible, pojawi się informacja o haśle admina dla ArgoCD:

Zapisz to hasło – będzie potrzebne, aby dostać się do GUI ArgoCD.

Jeżeli utracisz to hasło możesz je odzyskać poprzez połączenie się po ssh z vagrantem (vagrant ssh w katalogu Minikube) oraz wykonaniu następującej komendy:

sudo kubectl -n argocd get secret argocd-initial-admin-secret -o
jsonpath="{.data.password}" | base64 -d; echo



DOSTĘPNE ADRESY GUI

- Jenkins: 192.168.44.44:8880 (login panda:panda)
- Artifactory: 192.168.44.44:8081 (login admin:password)
- Sonarqube: 192.168.44.44:9000 (login admin:panda)
- ArgoCD: 192.168.44.45:8080 (login admin:<poprzedni slajd>)
- Frontend aplikacji: 192.168.44.45:30080
 (dostępny po uruchomieniu jobów Jenkins)



INTERFEJSY GRAFICZNE

W kolejnym kroku wejdź do przeglądarki internetowej i otwórz karty z następującymi adresami:

- http://192.168.44.44:8880 Użytkownik: panda, Hasło: panda
- http://192.168.44.45:8080

Użytkownik: admin, Hasło: zapisane podczas uruchomienia maszyny Minikube

Znajdziesz w nich kolejno interfejsy graficzne Jenkinsa (budowanie aplikacji) i ArgoCD (wdrożenie aplikacji).





URUCHOMIENIE JOBA

Zaloguj sie do Jenkinsa i kliknij w odnośnik backend lub frontend (możesz otworzyć oba odnośniki w osobnych zakładkach przeglądarki)

🙀 Jenkins



URUCHOMIENIE JOBA



Uruchom oba Joby za pomocą Uruchom. Następnie poczekaj na zakończenie jobów:

Declarative: Checkout SCM	Get Code	Unit tests	Sonarqube analysis	Build application image	Pushing image to Artifactory	Declarative: Post Actions
13s	1s	21s	41s	58s	17s	1s
13s	1s	21s	41s (paused for 5s)	58s	17s	1s





Po otwarciu ArgoCD w przeglądarce otrzymasz informację o certyfikacie SSL, który nie jest zaufany dla Twojego komputera.

Zaakceptuj ryzyko, aby przejść do ArgoCD (przykład po prawej dla przeglądarki Firefox)



Firefox wykrył potencjalne zagrożenie bezpieczeństwa i nie wczytał "192.168.44.45". Jeśli otworzysz tę stronę, atakujący będą mogli przechwycić informacje, takie jak hasła, adresy e-mail czy dane kart płatniczych.

Więcej informacji..

 Wróć do poprzedniej strony (zalecane)
 Zaawansowane...

 Witryna "192.168.44.45:8080" używa nieprawidłowego certyfikatu bezpieczeństwa.

 Certyfikat nie jest zaufany, ponieważ jest samopodpisany.

 Kod błędu: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT

 Wyświetl certyfikat

 Wróć do poprzedniej strony (zalecane)

 Akceptuję ryzyko, kontynuuj



Zaloguj się do ArgoCD. Po maksymalnie 2 minutach powinnieneś zobaczyć obie aplikacje w zielonym kolorze.





Wejdź do środka każdej z nich i zobacz czy wdrożenie przeszło poprawnie:







Wejdź na adres aplikacji (<u>http://192.168.44.45:30080/</u>). Aplikacja powinna być dostępna:

Name:
Animal:
Add Animals Get Animals
Animals:
Action log:





URUCHOMIENIE JOBA 2

Ponownie zaloguj sie do Jenkinsa i kliknij w odnośnik frontend.

Następnie naciśnij:

鬱 Konfiguruj

Przewiń stronę i znajdź:

Branch Specifier (blank for 'any') final

Zmień wartość final na workshop. Ponownie uruchom job frontend.

6.72	In	
(CO).	6191	
10		





Ponownie zaloguj się do ArgoCD, wejdź do aplikacji frontend. Po maksymalnie 2 minutach aplikacja frontend powinna zostać automatycznie zaktualizowana – przez jakiś czas powinno być widać 2 pody :





APLIKACJA 2

Wejdź na adres aplikacji (<u>http://192.168.44.45:30080/</u>) lub odśwież stronę. Aplikacja powinna być zaktualizowana:

A change has been made !
Name:
Animal:
Add Animals Get Animals
Animals:
Action log:





SUKCES !

Gratulacje! Udało Ci się poprawnie uruchomić demo warsztatu z laboratorium DevOps. Zachęcamy do przejrzenia i eksperymentowania z komponentami. W przypadku jakichkolwiek błędów, wystarczy ponownie stworzyć maszyny wirtualne i zacząć jeszcze raz.

Kolejne kroki pozwolą Ci posprzątać Komputer z plików warsztatowych.





VAGRANT - USUNIĘCIE



Aby całkowicie usunąć maszynę wirtualną należy w terminalu wewnątrz katalogów Minikube oraz Vagrant wpisać komendę vagrant destroy -f.

C:\Users\

\Downloads\Vagrant\devops-workshop>vagrant destroy -f ==> default: Forcing shutdown of VM...

==> default: Destroying VM and associated drives...



DZIĘKUJEMY !

Gotowe! Na koniec odinstaluj (jeżeli nie chcesz ich już używać) programy z pierwszych slajdów!

Dziękujemy za uczestnictwo w warsztatach!

