
Infrastruktura jako kod

Dynamiczne systemy w epoce chmury

Kief Morris

przekład: Janusz Machowski

Spis treści

Przedmowaxiii

Podziękowania.....xxi

Część I. Podstawy

1. Co to znaczy infrastruktura jako kod?.....	3
Od epoki żelaza do epoki chmury.....	4
Infrastruktura jako kod.....	5
Korzyści z infrastruktury jako kodu.....	6
Używanie infrastruktury jako kodu do optymalizacji pod kątem zmian.....	6
Zarzut: nie dokonujemy zmian tak często, aby była uzasadniona ich automatyzacja.....	7
Zarzut: najpierw trzeba utworzyć, a potem automatyzować.....	7
Zarzut: musimy wybierać między szybkością i jakością.....	8
Cztery kluczowe wskaźniki.....	10
Trzy podstawowe praktyki dotyczące infrastruktury jako kodu.....	11
Podstawowa praktyka: definiowanie wszystkiego jako kodu.....	11
Podstawowa praktyka: stałe testowanie i dostarczanie wszystkiego na bieżąco.....	12
Podstawowa praktyka: tworzenie małych, prostych elementów, które można zmieniać niezależnie.....	12
Podsumowanie.....	12
2. Zasady infrastruktury w epoce chmury.....	13
Zasada: zakładaj, że systemy są zawodne.....	13
Zasada: rób tak, aby wszystko było odtwarzalne.....	14
Pułapka: systemy śnieżynki.....	15
Zasada: twórz rzeczy zastępowalne.....	15
Zasada: minimalizuj zróżnicowanie.....	16
Dryf konfiguracji.....	17
Zasada: pilnuj, abyś mógł powtórzyć każdy proces.....	18
Podsumowanie.....	20

3. Platformy infrastruktury	21
Części systemu infrastruktury	21
Platformy infrastruktury	22
Zasoby infrastruktury	25
Zasoby obliczeniowe	26
Zasoby pamięci masowej	27
Zasoby sieciowe	28
Podsumowanie	30
4. Podstawowa praktyka: definiuj wszystko jako kod	31
Dlaczego należy definiować infrastrukturę jako kod	31
Co można zdefiniować jako kod	32
Wybieraj narzędzia z eksternalizacją konfiguracji	32
Zarządzaj kodem w systemie kontroli wersji	33
Języki kodowania infrastruktury	34
Skrypty infrastruktury	35
Deklaratywne języki infrastruktury	37
Programowalne, imperatywne języki infrastruktury	39
Języki deklaratywne czy imperatywne do infrastruktury	40
Języki dziedziczne infrastruktury	40
Języki ogólnego przeznaczenia czy DSL infrastruktury	42
Zasady implementacji w przypadku definiowania infrastruktury jako kodu	42
Oddzielaj kod deklaratywny od imperatywnego	43
Traktuj kod infrastruktury jak prawdziwy kod	43
Podsumowanie	44

Część II. Praca ze stosami infrastruktury

5. Tworzenie stosów infrastruktury jako kodu	47
Co to jest stos infrastruktury?	47
Kod stosu	49
Instancja stosu	49
Konfigurowanie serwerów w stosie	49
Języki infrastruktury niskiego poziomu	50
Języki infrastruktury wysokiego poziomu	51
Wzorce i antywzorce konstruowania stosów	52
Antywzorzec: stos monolityczny	52
Wzorzec: stos grupy aplikacji	54
Wzorzec: stos usług	56
Wzorzec: mikrostos	57
Podsumowanie	58

6. Tworzenie środowisk przy użyciu stosów	59
O co chodzi w tych środowiskach	59
Środowiska dostarczania	59
Wiele środowisk produkcyjnych	60
Środowiska, spójność i konfiguracja	61
Wzorce budowania środowisk	62
Antywzorzec: stos wielu środowisk	62
Antywzorzec: środowiska kopiuj-wklej	63
Wzorzec: stos wielokrotnego użytku	65
Tworzenie środowisk z wieloma stosami	67
Podsumowanie	69
7. Konfigurowanie instancji stosu	71
Używanie parametrów stosu do tworzenia unikatowych identyfikatorów	72
Przykładowe parametry stosu	73
Wzorce konfigurowania stosów	74
Antywzorzec: ręczne parametry stosu	74
Wzorzec: zmienne środowiskowe stosu	76
Wzorzec: parametry skryptowe	78
Wzorzec: pliki konfiguracyjne stosu	81
Wzorzec: stos opakowujący	84
Wzorzec: parametry stosu potokowego	87
Wzorzec: rejestr parametrów stosu	90
Rejestr konfiguracji	93
Implementowanie rejestru konfiguracji	93
Jeden czy wiele rejestrów konfiguracji	95
Obsługa wpisów tajnych jako parametrów	96
Szyfrowanie wpisów tajnych	96
Autoryzacja bez wpisu tajnego	97
Wstrzykiwanie wpisów tajnych podczas wykonywania	97
Wpisy tajne jednorazowe	98
Podsumowanie	98
8. Podstawowa praktyka: ciągle testuj i dostarczaj	99
Po co ciągle testować kod infrastruktury?	100
Co oznacza ciągle testowanie	100
Co należy testować w przypadku infrastruktury?	102
Wyzwania związane z testowaniem kodu infrastruktury	104
Wyzwanie: testy kodu deklaratywnego mają często małą wartość	105
Wyzwanie: testowanie kodu infrastruktury jest powolne	107
Wyzwanie: zależności komplikują testowanie infrastruktury	109
Testowanie progresywne	109

Piramida testów	110
Model testowania „ser szwajcarski”	112
Potoki dostarczania infrastruktury	113
Etapy potoku	114
Zakres komponentów testowanych w ramach etapu	115
Zakres zależności używanych w etapie	115
Elementy platformy wymagane przez etap	116
Usługi i oprogramowanie potoków dostarczania	117
Testowanie w środowisku produkcyjnym	119
Czego nie można powielić poza środowiskiem produkcyjnym	120
Zarządzanie ryzykiem testowania w środowisku produkcyjnym	121
Podsumowanie	122
9. Testowanie stosów infrastruktury	123
Przykładowa infrastruktura	123
Przykładowy stos	124
Potok dla przykładowego stosu	125
Etapy testowania offline dla stosów	125
Sprawdzanie składni	126
Statyczna analiza kodu w trybie offline	126
Statyczna analiza kodu z użyciem API	127
Testowanie z użyciem atrapy API	127
Etapy testowania online dla stosów	128
Podgląd: patrzenie, jakie zmiany zostaną dokonane	128
Weryfikacja: stosowanie asercji dotyczących zasobów infrastruktury	129
Wyniki: potwierdzanie prawidłowego działania infrastruktury	131
Używanie warunków początkowych testu do obsługi zależności	132
Atrapy dla zależności nadrzędnych	133
Warunki początkowe testu dla zależności podrzędnych	134
Refaktoryzacja komponentów w celu umożliwienia ich izolowania	135
Wzorce cyklu życia dla testowych instancji stosów	136
Wzorzec: trwały stos testowy	136
Wzorzec: efemeryczny stos testowy	137
Antywzorzec: podwójny etap stosu – trwały i efemeryczny	138
Wzorzec: okresowa odbudowa stosu	140
Wzorzec: ciągle resetowanie stosu	141
Orkiestracja testów	143
Wspomaganie lokalnego testowania	143
Unikanie silnego sprzężenia z narzędziami potoku	144
Narzędzia do orkiestracji testów	144
Podsumowanie	145

Część III. Praca z serwerami i innymi platformami wykonawczymi aplikacji

10. Środowiska wykonawcze aplikacji	149
Infrastruktura natywna dla chmury i oparta na aplikacjach	150
Cele środowiska wykonawczego aplikacji.	151
Wdrażalne części aplikacji.	151
Pakiety wdrożenia	152
Wdrażanie aplikacji na serwerach	153
Pakowanie aplikacji do kontenerów	153
Wdrażanie aplikacji w klastrach serwerów	154
Wdrażanie aplikacji w klastrach aplikacji.	155
Pakiety do wdrażania aplikacji w klastrach	156
Wdrażanie aplikacji bezserwerowych na platformach FaaS.	157
Dane aplikacji.	158
Schematy i struktury danych	158
Infrastruktura magazynu aplikacji natywna dla chmury	159
Łączność aplikacji	159
Odnajdywanie usług	160
Podsumowanie	162
11. Budowanie serwerów jako kodu	163
Co jest trzymane na serwerze.	164
Skąd pochodzą rzeczy trzymane na serwerze	165
Kod konfiguracji serwera	166
Moduły kodu konfiguracji serwera	167
Projektowanie modułów kodu konfiguracji serwera	168
Wersjonowanie i promowanie kodu serwera	169
Role serwerów.	169
Testowanie kodu serwera	171
Testowanie progresywne kodu serwera.	171
Co testować w przypadku kodu serwera.	172
Jak testować kod serwera.	172
Tworzenie nowej instancji serwera	173
Ręczne tworzenie nowej instancji serwera	174
Tworzenie serwera przy użyciu skryptu	175
Tworzenie serwera przy użyciu narzędzia zarządzania stosem	175
Konfigurowanie automatycznego tworzenia serwerów przez platformę	176
Tworzenie serwera przy użyciu sieciowego narzędzia wyposażania.	177
Wstępne budowanie serwerów.	178
Klonowanie serwera „na gorąco”	178
Używanie migawek serwera	179

Tworzenie czystego obrazu serwera	179
Konfigurowanie nowej instancji serwera	180
Smażenie instancji serwera	181
Pieczenie obrazów serwera	182
Łączenie pieczenia i smażenia	182
Stosowanie konfiguracji serwera podczas tworzenia serwera	183
Podsumowanie	184
12. Zarządzanie zmianami w serwerach.	185
Wzorce zarządzania zmianami: kiedy stosować zmiany	186
Antywzorzec: stosowanie każdej zmiany oddzielnie	186
Wzorzec: ciągła synchronizacja konfiguracji	187
Wzorzec: serwer niezmienny	189
Jak stosować kod konfiguracji serwera	191
Wzorzec: wypychanie konfiguracji serwera	192
Wzorzec: pobieranie konfiguracji serwera	193
Pozostałe zdarzenia w cyklu życia serwera	196
Zatrzymywanie i restartowanie instancji serwera	196
Zastępowanie instancji serwera	197
Odzyskiwanie serwera po awarii	198
Podsumowanie	199
13. Obrazy serwerów jako kod	201
Budowanie obrazu serwera	201
Po co budować obraz serwera?	202
Jak zbudować obraz serwera	203
Narzędzia do budowania obrazów serwerów	203
Proces budowania obrazu online	204
Proces budowania obrazu offline	207
Zawartość źródłowa obrazu serwera	208
Budowanie przy użyciu stockowego obrazu serwera	208
Budowanie obrazu serwera od podstaw	209
Pochodzenie obrazu serwera i jego zawartości	209
Zmianie obrazu serwera	210
Ogrzewanie czy pieczenie świeżego obrazu	210
Wersjonowanie obrazu serwera	211
Aktualizowanie instancji serwera po zmianie obrazu	212
Udostępnianie obrazu serwera do wykorzystania przez wiele zespołów	213
Obsługa istotnych zmian w obrazie	214
Używanie potoku do testowania i dostarczania obrazu serwera	215
Etap budowy obrazu serwera	215
Etap testowania obrazu serwera	217

Etap dostarczania obrazu serwera	218
Używanie wielu obrazów serwera	218
Obrazy serwera dla różnych platform infrastruktury	219
Obrazy serwera dla różnych systemów operacyjnych	219
Obrazy serwera dla różnych architektur sprzętowych	219
Obrazy serwera dla różnych ról	220
Warstwowanie obrazów serwera	220
Współdzielenie kodu przez obrazy serwerów	221
Podsumowanie	222
14. Budowanie klastrów jako kodu	223
Rozwiązania dla klastrów aplikacji	224
Klaster jako usługa	224
Spakowana dystrybucja klastra	225
Topologie stosu dla klastrów aplikacji	226
Stos monolityczny wykorzystujący klaster jako usługę	227
Stos monolityczny dla spakowanego rozwiązania klastrowego	228
Potok dla monolitycznego stosu klastra aplikacji	229
Przykład wielu stosów dla klastra	232
Strategie współdzielenia dla klastrów aplikacji	234
Jeden duży klaster do wszystkiego	235
Oddzielne klastry dla etapów dostarczania	236
Klastry dla nadzoru	237
Klastry dla zespołów	238
Siatka usług	238
Infrastruktura dla bezserwerowej usługi FaaS	240
Podsumowanie	242

Część IV. Projektowanie infrastruktury

15. Podstawowa praktyka: małe, proste elementy	245
Projektowanie pod kątem modularności	246
Cechy dobrze zaprojektowanych komponentów	246
Zasady projektowania komponentów	247
Używanie testowania do podejmowania decyzji projektowych	250
Modularyzacja infrastruktury	250
Komponenty stosów a stosy jako komponenty	250
Używanie serwera w stosie	252
Wytaczanie granic między komponentami	255
Dopasowywanie granic do naturalnych wzorców zmian	256
Dopasowywanie granic do cykli życia komponentów	256

Dopasowywanie granic do struktur organizacyjnych	258
Wytyczanie granic wspierających odporność	258
Tworzenie granic wspierających skalowanie	259
Dopasowywanie granic do zasad bezpieczeństwa i nadzoru	262
Podsumowanie	263
16. Budowanie stosów z komponentów	265
Języki infrastruktury dla komponentów stosu	266
Ponowne używanie kodu deklaratywnego za pomocą modułów	266
Dynamiczne tworzenie elementów stosu za pomocą bibliotek	267
Wzorce dla komponentów stosu	268
Wzorzec: moduł fasady	268
Antywzorzec: moduł zaciemniający	270
Antywzorzec: moduł niewspółdzielony	272
Wzorzec: moduł pakietowy	273
Antywzorzec: moduł spaghetti	274
Wzorzec: jednostka domeny infrastruktury	277
Budowanie warstwy abstrakcji	279
Podsumowanie	280
17. Używanie stosów jako komponentów	281
Wykrywanie zależności między stosami	281
Wzorzec: dopasowywanie zasobów	282
Wzorzec: wyszukiwanie danych w stosie	285
Wzorzec: wyszukiwanie w rejestrze integracji	288
Wstrzykiwanie zależności	290
Podsumowanie	293

Część V. Dostarczanie infrastruktury

18. Organizowanie kodu infrastruktury	297
Organizowanie projektów i repozytoriów	297
Jedno repozytorium czy wiele?	298
Jedno repozytorium na wszystko	298
Oddzielne repozytorium dla każdego projektu (mikrorepo)	300
Wiele repozytoriów z wieloma projektami	301
Organizowanie różnych rodzajów kodu	302
Pliki pomocnicze projektu	302
Testy wielu projektów	303
Dedykowane projekty testów integracji	305
Organizowanie kodu według koncepcji domeny	305

Organizowanie plików z wartościami konfiguracyjnymi	306
Zarządzanie kodem infrastruktury i aplikacji	307
Dostarczanie infrastruktury i aplikacji	308
Testowanie aplikacji razem z infrastrukturą	309
Testowanie infrastruktury przed integracją	310
Używanie kodu infrastruktury do wdrażania aplikacji	310
Podsumowanie	312
19. Dostarczanie kodu infrastruktury	313
Dostarczanie kodu infrastruktury	313
Budowanie projektu infrastruktury	314
Pakowanie kodu infrastruktury jako artefaktu	315
Używanie repozytorium do dostarczania kodu infrastruktury	315
Integrowanie projektów	317
Wzorzec: integracja projektów podczas budowania	319
Wzorzec: integracja projektów podczas dostarczania	322
Wzorzec: integracja projektów podczas stosowania	324
Używanie skryptów do opakowywania narzędzi infrastruktury	327
Zbieranie wartości konfiguracyjnych	328
Upraszczenie skryptów opakowujących	329
Podsumowanie	330
20. Przepływy pracy zespołowej	331
Ludzie	332
Kto pisze kod infrastruktury?	334
Stosowanie kodu do infrastruktury	336
Stosowanie kodu z poziomu lokalnej stacji roboczej	336
Stosowanie kodu z poziomu scentralizowanej usługi	337
Prywatne instancje infrastruktury	338
Gałęzie kodu źródłowego w przepływach pracy	340
Zapobieganie dryfowi konfiguracji	341
Minimalizacja opóźnienia automatyzacji	341
Unikanie stosowania ad hoc	342
Ciągłe stosowanie kodu	342
Infrastruktura niezmiennalna	342
Nadzór w przepływie pracy opartym na potoku	343
Reorganizowanie obowiązków	344
Przesunięcie w lewo	345
Przykładowy proces w przypadku infrastruktury jako kodu podlegającej nadzorowi	345
Podsumowanie	346

21. Bezpieczne zmienianie infrastruktury	347
Ograniczanie zasięgu zmiany	347
Małe zmiany	349
Przykład refaktoryzacji	351
Wypychanie niekompletnych zmian do produkcji	352
Instancje równoległe	353
Transformacje kompatybilne wstecz	356
Przełączniki funkcji	357
Zmiana działającej infrastruktury	360
Chirurgia infrastruktury	361
Powiększanie i zmniejszanie	363
Zmiany bez przestojów	366
Ciągłość	367
Ciągłość poprzez zapobieganie błędom	368
Ciągłość poprzez szybkie odzyskiwanie	369
Ciągłe odzyskiwanie po awarii	370
Inżynieria chaosu	371
Planowanie na wypadek awarii	371
Ciągłość danych w zmieniającym się systemie	373
Blokowanie	373
Segregowanie	374
Replikacja	374
Ponowne ładowanie	374
Mieszane podejścia do ciągłości danych	375
Podsumowanie	375
Indeks	377
O autorze	395