Azure i urządzenia

Podstawy operacyjne Azure oraz modele własności urządzeń.

- Azure: podstawy operacyjne
- Urządzenia służbowe i prywatne: BYOD, COPE, COBO

Azure: podstawy operacyjne

Poziom: podstawy **Dotyczy: Azure CLI** **Czas: ~25 min**

Zanim wydamy pierwszą komendę `az`, warto ustalić, w jakiej hierarchii żyje każdy zasób w Azure — inaczej niż w AWS, ta hierarchia ma jeden dodatkowy, obowiązkowy poziom.

Komendy wykonujemy z CLI bash (lub PowerShell na Windows) z zainstalowanym Azure CLI (`az`) dostępnym w zmiennej `PATH`.

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
AWS	Amazon Web Services: chmura Amazonu
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
RBAC	Role-Based Access Control: uprawnienia nadawane rolom, a role ludziom
S3	Simple Storage Service: magazyn plików w chmurze AWS

Fiszka

TYLKO ODCZYT

Zalogowanie się do Azure: az login

```
az login
```

Otwiera przeglądarkę do interaktywnego logowania kontem Microsoft Entra ID — po pomyślnym zalogowaniu CLI zapisuje token sesji lokalnie i wypisuje listę subskrypcji, do których zalogowane konto ma dostęp.

Przykład:

```
[
  {
    "cloudName": "AzureCloud",
    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "isDefault": true,
    "name": "Subskrypcja-Produkcja",
    "state": "Enabled",
    "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
    "user": {
      "name": "nazwa_usera@zielona.example",
      "type": "user"
    }
  }
]
```


CLI = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Sprawdzenie aktywnej subskrypcji: az account show

```
az account show
```

Odpowiednik `aws sts get-caller-identity` — pierwsza komenda przy „nie wiem, na której subskrypcji aktualnie pracuję", zwłaszcza gdy konto ma dostęp do wielu subskrypcji naraz.

Przykład:


```
{
  "environmentName": "AzureCloud",
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "isDefault": true,
  "name": "Subskrypcja-Produkcja",
  "state": "Enabled",
  "user": {
    "name": "nazwa_usera@zielona.example",
    "type": "user"
  }
}
```

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Przełączenie się na inną subskrypcję: az account set

```
az account set --subscription NazwaSubskrypcji
```

Zmienia subskrypcję domyślną dla wszystkich kolejnych komend w bieżącej sesji CLI — bez tego każde polecenie działa na subskrypcji oznaczonej jako `isDefault: true` w wyniku poprzedniej fiszki, co po zalogowaniu na konto z wieloma subskrypcjami bywa nie tą subskrypcją, na której faktycznie chcemy pracować.

□□ **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Wylistowanie grup zasobów: az group list

```
az group list -o table
```

Zwraca wszystkie grupy zasobów widoczne w bieżącej subskrypcji, z regionem, w którym każda z nich została utworzona.

Przykład:

```
Name                  Location       Status
--------------------  -------------  ---------
rg-produkcja          polandcentral  Succeeded
rg-testowe            westeurope     Succeeded
```

□□ **az** = Azure CLI: narzędzie chmury Microsoftu

□□ TYLKO ODCZYT

Wylistowanie maszyn wirtualnych: az vm list

```
az vm list -o table
```

Flaga `-o table` (skrót `--output table`) formatuje wynik jako czytelną tabelę zamiast domyślnego JSON-a — działa identycznie jak `--output table` w AWS CLI opisanym na stronie AWS CLI: instalacja, profile, poświadczenia.

Przykład:

```
Name             ResourceGroup    Location       Zones
---------------  ---------------  -------------  -------
app01            rg-produkcja     polandcentral  1
app02            rg-produkcja     polandcentral  2
```

AWS = Amazon Web Services: chmura Amazonu · **CLI** = Command Line Interface: praca przez wpisywanie komend w terminalu · **az** = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Wylistowanie kont magazynu: az storage account

```
az storage account list -o table
```

> **Pojęcie: konto magazynu (storage account).** Kontener najwyższego poziomu dla usług magazynowych Azure — dopiero wewnątrz konta magazynu tworzy się kontenery blobów, udziały plikowe czy kolejki. Odpowiada mniej więcej kubełkowi S3, ale jest o jeden poziom wyżej: jedno konto magazynu może zawierać wiele kontenerów blobów, podczas gdy w S3 kubełek jest już najniższym poziomem organizacji obiektów.

Przykład:

```
Name              ResourceGroup    Location       Kind
----------------  ---------------  -------------  -------------
przykladstorage01 rg-produkcja     polandcentral  StorageV2
```

S3 = Simple Storage Service: magazyn plików w chmurze AWS · **az** = Azure CLI: narzędzie chmury Microsoftu

TYLKO ODCZYT

Wylistowanie blobów w kontenerze: az storage blob

```
az storage blob list --account-name NazwaKonta --container-name NazwaKontenera -o table
```

Odpowiednik `aws s3 ls s3://NazwaKubelka/` — zwraca listę obiektów (blobów) wewnątrz wskazanego kontenera danego konta magazynu.

Przykład:

```
Name               Blob Type    Length    Last Modified
-----------------  -----------  --------  -------------------------
faktury/f-001.pdf  BlockBlob    102934    2026-01-15T08:12:33+00:00
```

az = Azure CLI: narzędzie chmury Microsoftu

Z życia wzięte

Z ŻYCIA WZIĘTE

O 11:50 nowa maszyna wirtualna Tomka odmawia utworzenia z błędem o brakującym argumencie, chociaż dokładnie taka sama komenda działała u kolegi z innego zespołu bez problemu. Sprawdza polecenie kolegi i widzi, że różni się tylko jednym parametrem: `--resource-group`, którego Tomek nie podał, licząc na to, że Azure jakoś sam dobierze domyślne miejsce, tak jak przy tworzeniu zasobu w AWS. Dodaje istniejącą grupę zasobów do komendy i maszyna powstaje bez dalszych przeszkód.

Subskrypcja, grupa zasobów, zasób — hierarchia

Pojęcie: subskrypcja (subscription). Jednostka rozliczeniowa i graniczna w Azure, odpowiednik konta AWS — każdy zasób istnieje wewnątrz dokładnie jednej subskrypcji, a limity ilościowe (liczba maszyn wirtualnych, liczba adresów IP) są liczone w jej obrębie. Organizacja może mieć wiele subskrypcji, np. osobną dla produkcji i osobną dla środowisk testowych, pogrupowanych w grupę zarządzania (Management Group).

Pojęcie: grupa zasobów (Resource Group). Kontener logiczny wewnątrz subskrypcji, do którego przypisujemy powiązane ze sobą zasoby — w odróżnieniu od AWS, gdzie rolę

grupowania pełnią wyłącznie tagi, w Azure niemal każdy zasób *musi* należeć do jakiejś grupy zasobów w momencie utworzenia. Grupa zasobów ma własne uprawnienia RBAC, własny region (co nie ogranicza regionu zasobów wewnątrz niej) i usuwa się jednym poleceniem razem z całą zawartością — najczęstszy sposób sprzątania środowiska testowego w Azure to usunięcie jego grupy zasobów, nie pojedynczych zasobów po kolei.

Pojęcie: zasób (resource). Pojedynczy obiekt zarządzany przez Azure — maszyna wirtualna, konto magazynu, sieć wirtualna. Każdy zasób ma unikatowy identyfikator zasobu (Resource ID) w postaci ścieżki zawierającej subskrypcję, grupę zasobów, dostawcę zasobu i nazwę: `/subscriptions/NazwaSubskrypcji/resourceGroups/NazwaGrupyZasobow/providers/Microsoft.Compute/virtualMachines/NazwaMaszyny`.

Hierarchia czyta się więc tak: subskrypcja zawiera grupy zasobów, grupa zasobów zawiera zasoby. Ta sama nazwa zasobu (np. maszyny wirtualnej) może wystąpić w dwóch różnych grupach zasobów tej samej subskrypcji bez konfliktu, bo pełny identyfikator zawiera też nazwę grupy.



Subskrypcja zawiera grupy zasobów, a grupa zasobów zawiera poszczególne zasoby, takie jak maszyny wirtualne czy konta magazynu (kliknięcie otwiera pełny rozmiar)

Nazewnictwo: AWS kontra Azure

Pojęcie	AWS	Azure
Jednostka rozliczeniowa najwyższego poziomu	konto (Account)	subskrypcja (Subscription)
Magazyn obiektowy	S3	Blob Storage
Maszyna wirtualna	EC2 (Elastic Compute Cloud)	Virtual Machine
Tożsamość i uprawnienia	IAM	Microsoft Entra ID
Sieć wirtualna	VPC (Virtual Private Cloud)	VNet (Virtual Network)
Geograficzny obszar infrastruktury	region	region

Tabela ta jest pomocna głównie przy pierwszym kontakcie z drugim dostawcą, gdy szukamy odpowiednika usługi, którą już znamy — dokładne mapowanie funkcjonalności nie zawsze jest jeden do jednego (np. Blob Storage ma inną strukturę niż S3, opisaną wyżej przy koncie magazynu), ale nazewnictwo z tej tabeli wystarcza, żeby zorientować się, w którym miejscu dokumentacji szukać.

Częste błędy

Próba utworzenia zasobu bez wcześniej istniejącej grupy zasobów. W odróżnieniu od AWS, gdzie zasób po prostu powstaje w koncie i regionie, w Azure niemal każda komenda tworząca zasób wymaga `--resource-group` wskazującego już istniejącą grupę — próba pominięcia tego parametru kończy się błędem o brakującym argumencie, nie utworzeniem zasobu w jakimś domyślnym miejscu.

Praca na złej subskrypcji po zalogowaniu na konto z wieloma subskrypcjami. `az login` ustawia jako domyślną subskrypcję oznaczoną w Azure jako podstawowa dla danego konta, co nie musi być tą subskrypcją, na której akurat pracujemy — `az account show` przed rozpoczęciem pracy jest tu tym samym nawykiem, co `aws sts get-caller-identity` w AWS.

Mylenie usunięcia zasobu z usunięciem grupy zasobów. Usunięcie pojedynczej maszyny wirtualnej pozostawia po niej dysk, kartę sieciową i adres IP jako osobne zasoby w tej samej grupie, jeśli nie zostały utworzone jako część jednej operacji `az vm delete` z odpowiednimi flagami — pełne posprzątanie środowiska testowego jest prostsze przez usunięcie całej grupy zasobów niż przez ręczne usuwanie każdego zasobu z osobna.

□□ Ćwiczenia

> Zadania wykonujemy z zainstalowanym Azure CLI i dostępem do subskrypcji testowej — bez dostępu do subskrypcji rozwiązujemy punkty 1 i 3 opisowo.

1. Wyjaśniamy, czym różni się hierarchia zasobów w Azure (subskrypcja → grupa zasobów → zasób) od hierarchii w AWS (konto → zasób, z tagami jako jedynym grupowaniem).
2. Logujemy się komendą `az login`, sprawdzamy aktywną subskrypcję i — jeśli konto ma dostęp do więcej niż jednej — przełączamy się na inną komendą `az account set`.
3. Opisujemy różnicę między kontem magazynu a kontenerem blobów i podajemy, który z tych dwóch poziomów jest bliższy pojęciowo kubełkowi S3.
4. Tworzymy testową grupę zasobów, wewnątrz niej dowolny drobny zasób, a następnie sprawdzamy, że jedno polecenie usuwające grupę zasobów usuwa też ten zasób.

Powiązane: Chmura: pojęcia, które trzeba znać zanim cokolwiek uruchomimy · AWS CLI: instalacja, profile, poświadczenia · S3: kubełki, obiekty, uprawnienia

Źródła: dokumentacja Azure CLI · dokumentacja Azure Resource Manager, sekcja „Resource groups"

Urządzenia służbowe i prywatne: BYOD, COPE, COBO

Poziom: podstawy

Dotyczy: zarządzanie urządzeniami mobilnymi (MDM), niezależnie od platformy

Czas: ~15 min

Zanim wybierzemy narzędzie MDM i zaczniemy pisać polityki, musimy rozstrzygnąć jedno pytanie leżące poziom wyżej — czyje jest urządzenie i kto w związku z tym ponosi jakie ryzyko.

Skróty użyte na tej stronie (7)

Skrót	Co znaczy
BYOD	Bring Your Own Device: pracownik używa prywatnego urządzenia do pracy
COBO	Corporate Owned, Business Only: urządzenie firmowe wyłącznie do celów służbowych, bez użytku prywatnego
COPE	Corporately Owned, Personally Enabled: firma kupuje urządzenie, ale pozwala go używać też prywatnie
CYOD	Choose Your Own Device: pracownik wybiera model urządzenia z zamkniętej listy firmowej
IAM	Identity and Access Management: zarządzanie tożsamościami i uprawnieniami (w AWS: użytkownicy, role, polityki)
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
MDM	Mobile Device Management: system zdalnego zarządzania telefonami i tabletami firmowymi

Z życia wzięte

Z ŻYCIA WZIĘTE

> O 12:40 dział kadr pyta, czy może zdalnie wyczyścić prywatny telefon zwolnionego pracownika, na którym zostawiona jest jeszcze poczta firmowa. Tomek sprawdza w konsoli MDM, że urządzenie jest zarejestrowane w modelu BYOD, więc pełne wyczyszczenie naruszyłoby prywatne dane pracownika, do których firma nie ma żadnego prawa. Zamiast tego cofa dostęp aplikacji pocztowej do konta w Microsoft Entra ID, co natychmiast odcina firmową skrzynkę od telefonu bez dotykania niczego prywatnego na urządzeniu.

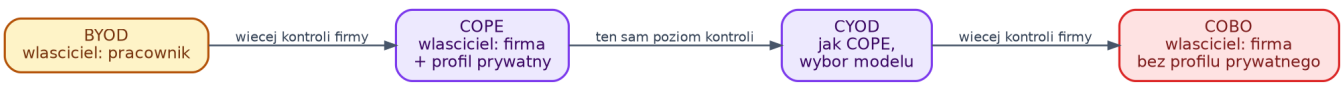
Model własności urządzenia

> **Pojęcie: model własności urządzenia.** Ustalenie, kto jest formalnym właścicielem sprzętu, na którym pracownik wykonuje obowiązki służbowe — firma czy sam pracownik — i wynikający z tego podział ryzyka oraz zakresu kontroli, jaki firma może narzucić na to urządzenie. To rozstrzygnięcie poprzedza każdą decyzję techniczną: zanim skonfigurujemy politykę MDM, musimy wiedzieć, czy w ogóle mamy prawo np. wymusić zdalne wyczyszczenie całego urządzenia, czy tylko profilu służbowego na nim.

Ryzyko rozkłada się w zależności od modelu w dwóch wymiarach naraz: kto ponosi koszt zakupu i serwisu sprzętu, oraz jak daleko firma może się posunąć w kontrolowaniu urządzenia bez naruszenia prywatności pracownika. Im więcej kontroli firma chce zachować, tym więcej musi wziąć na siebie odpowiedzialności za samo urządzenie — i odwrotnie.

Cztery modele

> **Pojęcie: BYOD (Bring Your Own Device).** Pracownik używa własnego, prywatnego urządzenia do celów służbowych. Firma nie jest właścicielem sprzętu i nie ma pełnej kontroli nad całym urządzeniem — może zarządzać wyłącznie wydzielonym profilem służbowym albo aplikacjami służbowymi przez rozwiązania klasy MAM (Mobile Application Management), a nie całym systemem operacyjnym.



Od BYOD przez COPE i CYOD po COBO rośnie zakres kontroli firmy nad urządzeniem (kliknięcie otwiera pełny rozmiar)

Pojęcie: COPE (Corporate-Owned, Personally Enabled). Urządzenie jest własnością firmy, ale pracownik ma prawo używać go też do celów prywatnych, z zachowaniem separacji między danymi służbowymi a prywatnymi w osobnym profilu. Administrator kontroluje ustawienia całego urządzenia (np. wymuszenie hasła, blokadę Bluetootha czy roamingu danych), ale profil prywatny pozostaje poza zasięgiem tej kontroli — dane i aplikacje w nim mają pozostać prywatne.

Pojęcie: CYOD (Choose Your Own Device). Wariant pośredni — firma jest właścicielem urządzenia (jak w COPE), ale pracownik wybiera model z zatwierdzonej przez firmę listy, zamiast dostawać jeden narzucony model odgórnie. Poziom kontroli i separacji danych jest taki sam jak w COPE — różnica dotyczy wyłącznie procesu zakupu, nie zarządzania urządzeniem po jego wydaniu.

Pojęcie: COBO (Corporate-Owned, Business Only). Urządzenie jest własnością firmy i służy wyłącznie do celów służbowych — bez profilu prywatnego, bez założenia, że pracownik będzie na nim trzymał cokolwiek osobistego. Daje firmie pełną kontrolę nad całym urządzeniem, bo nie ma na nim obszaru zastrzeżonego dla prywatności.

Co wolno narzucić w każdym modelu

Model	Właściciel sprzętu	Kontrola firmy	Dane prywatne na urządzeniu
BYOD	pracownik	tylko profil/aplikacje służbowe (MAM)	tak, poza zasięgiem firmy
COPE	firma	całe urządzenie, poza profilem prywatnym	tak, w wydzielonym, chronionym profilu
CYOD	firma	całe urządzenie, poza profilem prywatnym	tak, w wydzielonym, chronionym profilu
COBO	firma	całe urządzenie, bez wyjątku	nie, urządzenie wyłącznie służbowe

W modelu COPE administrator może typowo wymusić m.in.:

- ustawienie minimalnej złożoności hasła urządzenia,
- kontrolę Bluetootha i roamingu danych,

- konfigurację ochrony przed przywróceniem ustawień fabrycznych (factory reset protection), tak żeby zgubione albo skradzione urządzenie nie dało się po prostu wyzerować i użyć dalej.

Te same ustawienia w profilu prywatnym pozostają nienaruszone — użytkownik zachowuje pewność, że jego prywatne dane i aplikacje nie są ani widoczne, ani zarządzane przez administratora firmowego, mimo że fizycznie leżą na tym samym urządzeniu.

Analogia

Samochód służbowy dobrze oddaje różnicę między modelami. Auto prywatne, którym pracownik czasem jedzie w delegację (odpowiednik BYOD) — firma może zwrócić koszt paliwa i przebiegu, ale nie ma prawa zainstalować w nim śledzenia GPS ani zabronić przewożenia w nim czegokolwiek prywatnego, bo auto nie jest jej własnością. Samochód służbowy, którym pracownik jeździ też prywatnie po godzinach (odpowiednik COPE) — firma jest właścicielem i może wymagać np. regularnych przeglądów czy ograniczyć maksymalny przebieg prywatny, ale bagażnik z prywatnymi zakupami zostaje bagażnikiem pracownika, nie przedmiotem kontroli. Samochód służbowy używany wyłącznie do celów firmowych, wracający na parking firmowy po godzinach (odpowiednik COBO) — pełna kontrola firmy nad każdym aspektem jego użycia, bo nikt nie zakłada prywatnego użytku.

Rozdzielenie danych służbowych i prywatnych

> **Pojęcie: konteneryzacja profilu.** Techniczny mechanizm (np. profil służbowy w Android Enterprise, kontener MAM w iOS) tworzący na jednym urządzeniu dwie logicznie odseparowane przestrzenie — służbową i prywatną — z osobnym szyfrowaniem, osobną listą aplikacji i brakiem możliwości kopiowania danych między nimi bez jawnej zgody administratora (np. przez politykę zabraniającą kopiowania schowka między profilami). To właśnie ten mechanizm sprawia, że COPE różni się praktycznie od COBO nie tylko deklaracją w polityce firmowej, tylko realną, techniczną granicą na urządzeniu.

W BYOD separacja jest węższa — obejmuje zwykle tylko aplikacje służbowe i ich dane (np. pocztę firmową w wydzielonej aplikacji z własnym szyfrowaniem), bo firma nie ma prawa zarządzać całym prywatnym urządzeniem pracownika.

Rozstanie z pracownikiem

To, co dzieje się z urządzeniem i danymi w momencie zakończenia współpracy, jest bezpośrednią konsekwencją wybranego modelu:

- **COBO i CYOD/COPE bez separacji danych prywatnych** — urządzenie fizycznie wraca do firmy, więc pełne wyczyszczenie albo ponowna konfiguracja nie budzi wątpliwości prawnych.
- **COPE z profilem prywatnym** — firma może zdalnie usunąć wyłącznie profil służbowy (selective wipe), zostawiając profil prywatny nienaruszony, zanim urządzenie fizycznie wróci do działu IT.
- **BYOD** — firma nigdy nie ma prawa wyczyścić całego urządzenia, bo nie jest jego właścicielem; jedyną dopuszczalną operacją jest usunięcie profilu służbowego albo cofnięcie dostępu aplikacji służbowej do danych firmowych, np. przez unieważnienie tokenu logowania w Microsoft Entra ID czy odpowiedniku innego dostawcy tożsamości.

Uważaj: próba pełnego, zdalnego wyczyszczenia prywatnego urządzenia pracownika w modelu BYOD, nawet w dobrej wierze (np. po zgubieniu telefonu), może naruszać jego prawo własności do danych prywatnych na tym urządzeniu — polityka MDM dla BYOD powinna być od początku skonfigurowana tak, żeby taka operacja nie była technicznie możliwa, a nie tylko zabroniona proceduralnie.

Wymagania prawne przy danych prywatnych na urządzeniu służbowym

Sytuacja odwrotna do BYOD — dane prywatne na urządzeniu formalnie należącym do firmy (COPE, CYOD) — rodzi swoje własne zobowiązania. Skoro pracownik ma prawo trzymać na firmowym urządzeniu prywatne dane (zdjęcia, prywatną korespondencję, kontakty), firma jako administrator tego urządzenia staje się też stroną odpowiedzialną za to, żeby nie uzyskać do nich dostępu bez podstawy prawnej. W praktyce oznacza to co najmniej:

- jasną politykę informującą pracownika, do czego dokładnie firma ma wgląd (ustawienia urządzenia, inwentarz zainstalowanych aplikacji), a do czego nie (treść prywatnych wiadomości, zawartość prywatnych zdjęć),
- techniczne wymuszenie tej granicy przez konteneryzację profilu, a nie samą deklarację w regulaminie,
- procedurę na wypadek utraty albo kradzieży urządzenia, która nie prowadzi do przypadkowego ujawnienia albo usunięcia danych prywatnych pracownika przy okazji zabezpieczania danych firmowych.

Częste błędy

Wybór COBO tam, gdzie realna potrzeba to COPE. Odmówienie pracownikowi możliwości używania firmowego telefonu do celów prywatnych bez wyraźnego powodu biznesowego (np. wymogów bezpieczeństwa w środowisku regulowanym) zwykle prowadzi do tego, że pracownik i tak nosi przy sobie drugie, prywatne urządzenie — podwaja to koszt sprzętu bez realnej korzyści w bezpieczeństwie.

Brak jasnej polityki wygaszania dostępu w modelu BYOD. Poleganie wyłącznie na tym, że odchodzący pracownik „sam usunie" aplikację służbową z prywatnego telefonu, zamiast na wymuszonym cofnięciu dostępu po stronie tożsamości (Entra ID, IAM), zostawia lukę czasową, w której dane firmowe pozostają dostępne na urządzeniu poza kontrolą firmy.

Traktowanie deklaracji w regulaminie jako wystarczającego zabezpieczenia prywatności. Zapis „firma nie ma dostępu do danych prywatnych" bez technicznej konteneryzacji profilu nie jest żadną gwarancją — administrator z uprawnieniami MDM do całego urządzenia w modelu bez separacji profili może fizycznie zobaczyć wszystko, niezależnie od tego, co obiecuje polityka.

□□ Ćwiczenia

> Zadania sprawdzające, bez dostępu do konsoli MDM.

1. Firma rozważa wyposażenie zespołu handlowego w telefony służbowe z możliwością użytku prywatnego. Uzasadniamy, dlaczego COPE jest tu zwykle lepszym wyborem niż BYOD i niż COBO.
2. Opisujemy techniczną różnicę między „usunięciem urządzenia" a „selektywnym usunięciem profilu służbowego" i wskazujemy, w którym modelu własności druga opcja jest jedyną dopuszczalną.
3. Wyjaśniamy, dlaczego CYOD nie zmienia niczego w zakresie kontroli i separacji danych względem COPE, mimo że brzmi jak osobny model.
4. Pracownik na urządzeniu w modelu COPE zgłasza zgubienie telefonu. Opisujemy, jakie kroki może wykonać dział IT bez naruszenia prywatności danych osobistych pracownika, a jakich wykonać nie powinien.

Powiązane: Chmura: pojęcia, które trzeba znać zanim cokolwiek uruchomimy · Azure: podstawy operacyjne

Źródła: dokumentacja Microsoft Intune, sekcja „Android Enterprise corporate-owned devices with a work profile" · ogólne wzorce klasyfikacji modeli własności urządzeń (BYOD/COPE/CYOD/COBO) stosowane w branży MDM