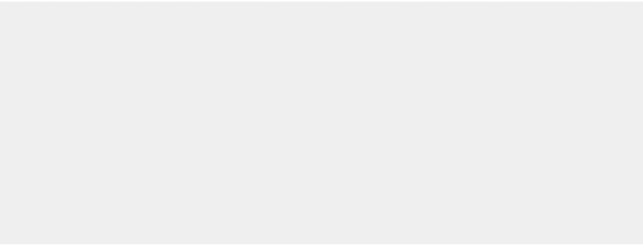


# Projektowanie systemów obiektowych i rozproszonych

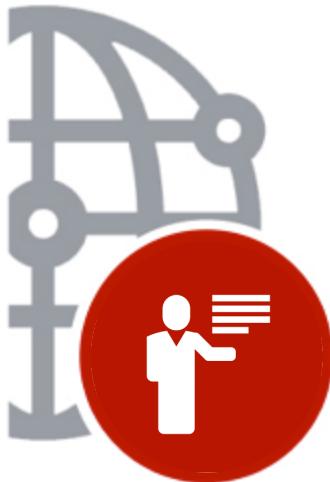
## Usługi w chmurach obliczeniowych



Tomasz Adamczewski  
AMG.net

Radosław Adamus  
PŁ





- 1. Przegląd usług platform cloudowych**
- 2. Wirtualizacja zasobów**
- 3. Cloud compute**
- 4. Cloud storage**
- 5. Cloud networking**

# ” Przegląd usług platform cloudowych



**Service (usługa)**: element oprogramowania, mogący działać niezależnie od innych oraz posiadający zdefiniowany interfejs, za pomocą którego udostępnia realizowane funkcje. Sposób działania każdej usługi jest w całości zdefiniowany przez interfejs ukrywający szczegóły implementacyjne.

**SOA, Service Oriented Architecture (architektura zorientowana na usługi)**: koncepcja tworzenia systemów informatycznych, w której główny nacisk stawia się na definiowanie usług spełniających wymagania użytkownika oraz komunikację między nimi. Komunikacja pomiędzy usługami odbywa się poprzez ustalony protokół komunikacji. SOA jest przeciwnieństwem tzw. architektury monolitycznej

## Abstrakcja i enkapsulacja

usługi komunikują się wyłącznie poprzez zestaw ściśle ustalonych, publicznych kontraktów (interfejsów)

## Luźne powiązanie i rozproszenie

każda usługa może być implementowana w innej technologii i uruchomiona w innej fizycznej lokalizacji

## Wielokrotne użycie i niezależność

każda usługa to zamknięty zestaw operacji, które opcjonalnie komunikują się z innymi usługami (poprzez interfejsy); implementacje poszczególnych usług można niezależnie wymieniać

## Ziarnistość i modularność

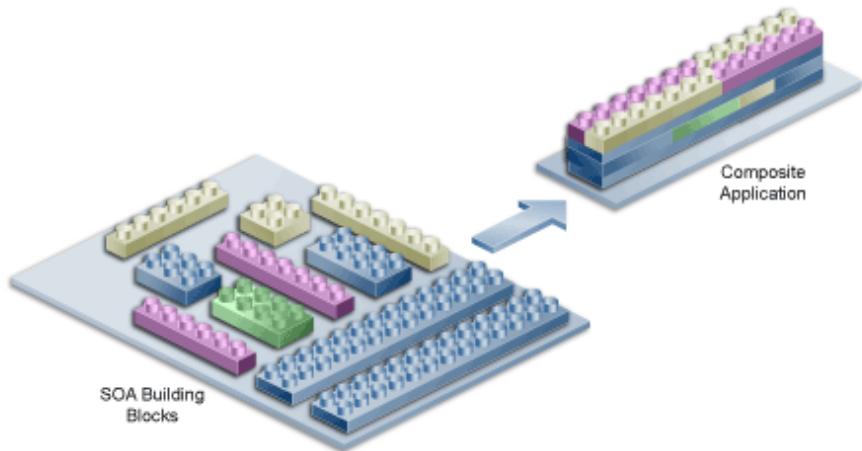
każda usługa jest w pełni odpowiedzialna za określony zestaw operacji i encji biznesowych

## Bezstanowość

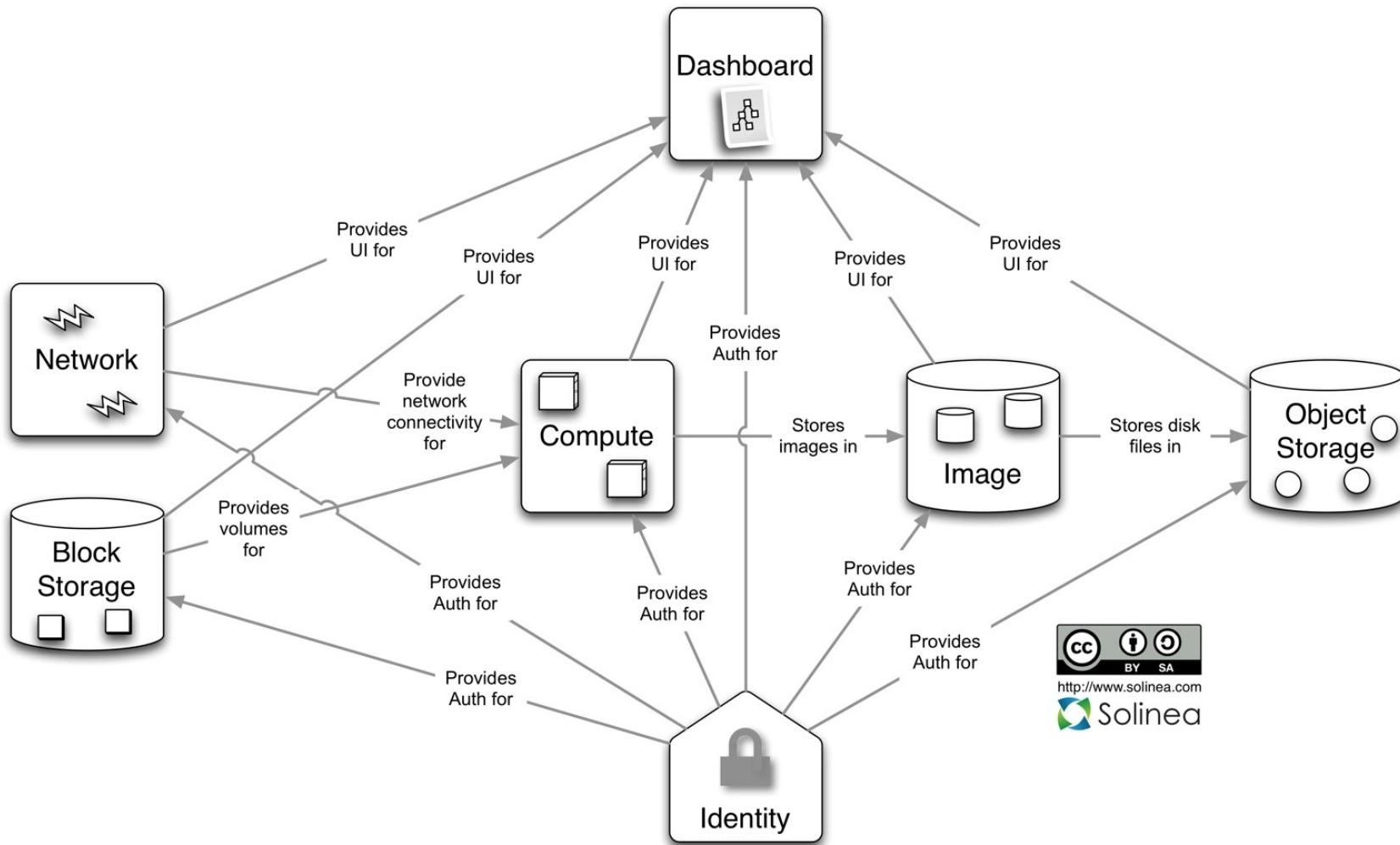
usługa nie utrzymuje sesji z klientem; każde wywołanie może być przetworzone niezależnie (np. przez innąinstancję usługi)

## Standaryzacja interfejsów

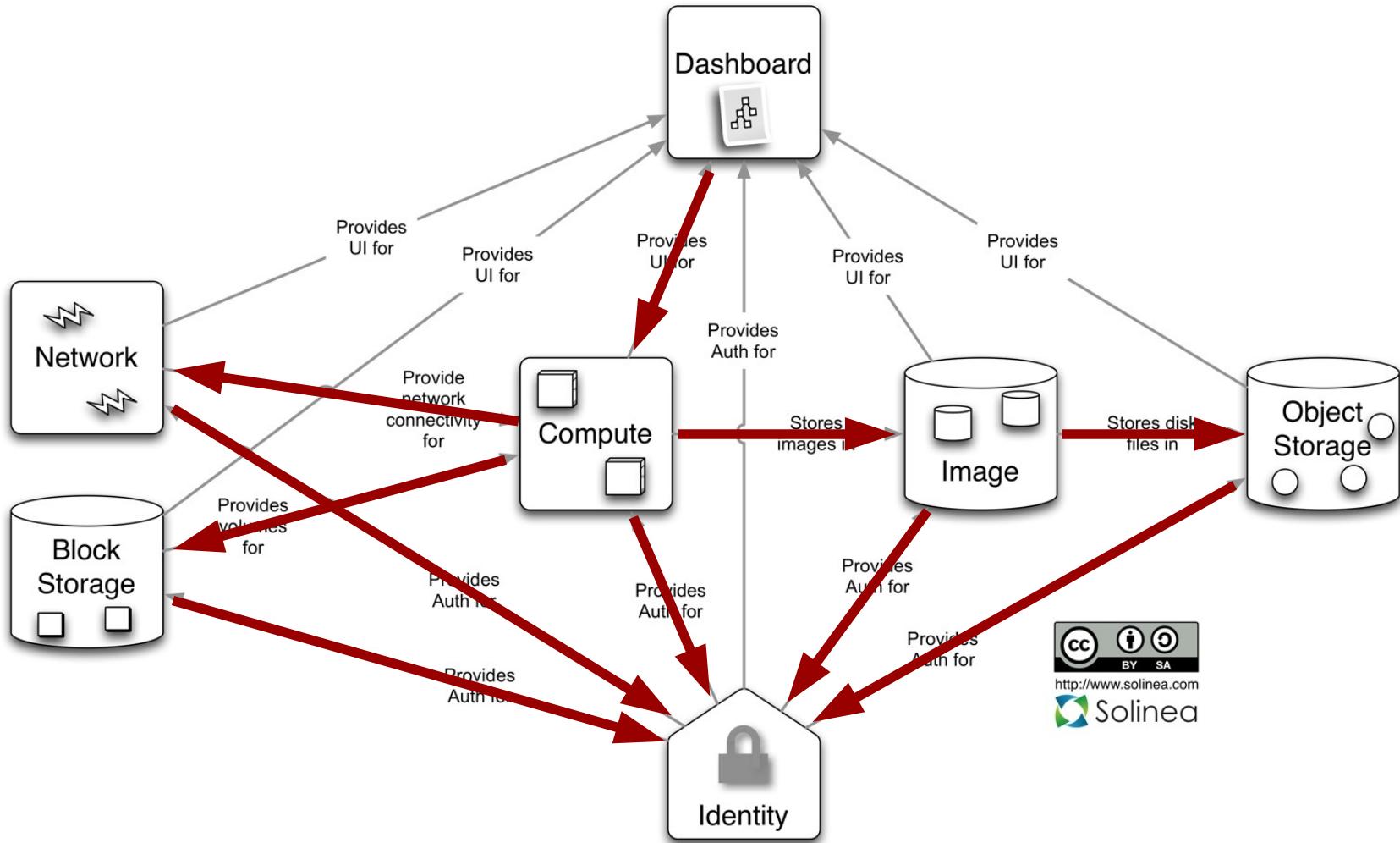
dla zachowania spójności, interfejsy są projektowane wokół wspólnego zestawu wytycznych (np. protokół komunikacji, metoda autoryzacji, wspierany format komunikatów)

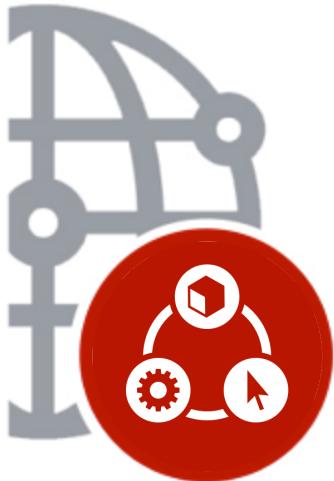


# Komunikacja pomiędzy usługami



# Uruchamiamyinstancję





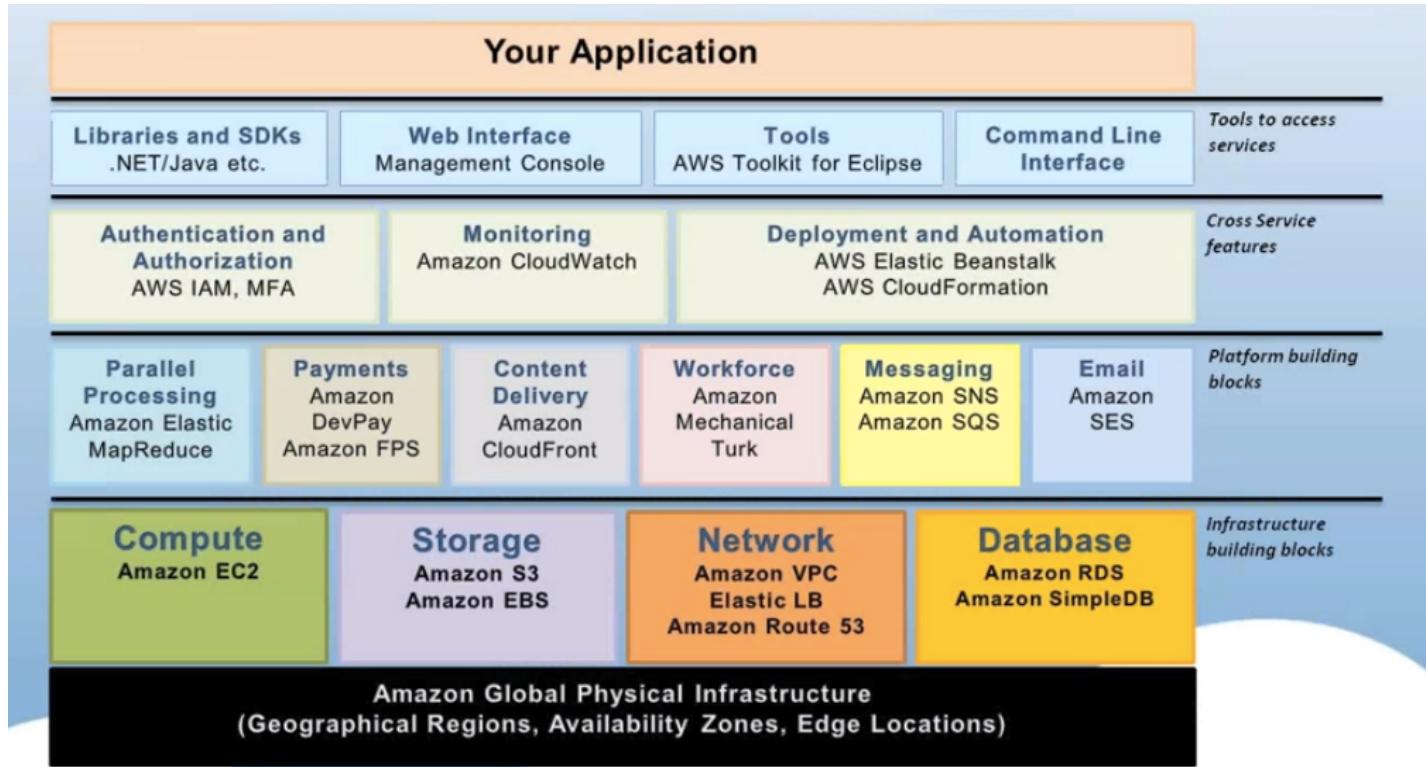
**Konsola Administracyjna:** aplikacja webowa dostarczająca interfejs graficzny dla poszczególnych usług.

Wykorzystywana podczas wykonywania okazjonalnych zmian w infrastrukturze oraz przeglądania raportów

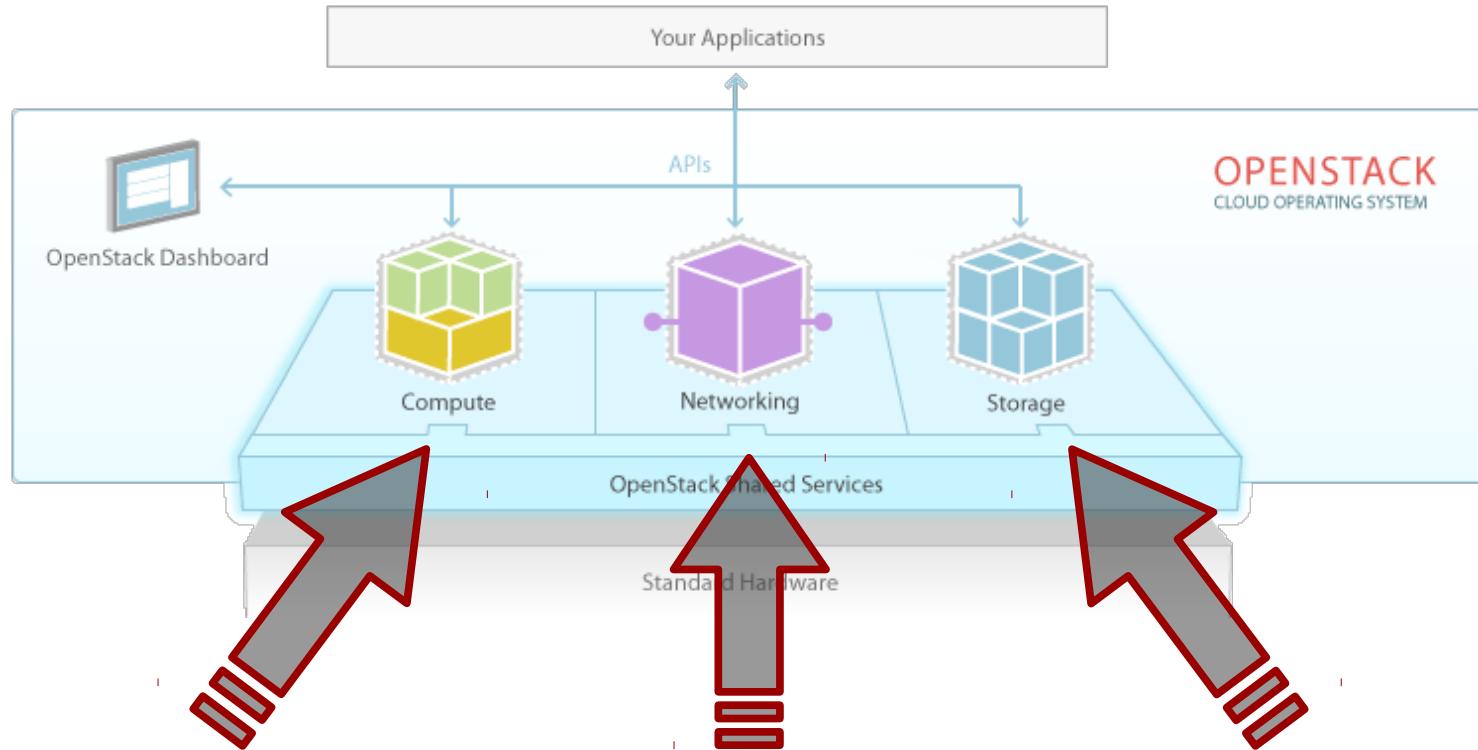
**Command Line Interface (CLI):** gotowe biblioteki umożliwiające komunikację z usługami chmury poprzez konsolę. Wykorzystywana podczas automatyzacji zadań (np. komunikacja z wirtualnej instancji podczas kontekstualizacji)

**Software Development Kit (SDK):** zestaw gotowych bibliotek klienckich, zaimplementowanych w popularnych językach programowania. Wykorzystywana w aplikacjach klienckich (np. w modelu SaaS, w celu komunikacji z infrastrukturą chmury)

**Klient REST API:** jeśli nie istnieje gotowe SDK dla naszego języka, możemy takie utworzyć samodzielnie (dzięki popularności i łatwości korzystania z API RESTowych)



# Podstawowe usługi w chmurach obliczeniowych



Zarządzainstancjami, na których uruchamiane są aplikacje klientów

Odpowiada za bezpieczeństwo oraz dostęp do instancji klienta

Umożliwia przechowywanie danych klienta



## Regions

W pełni niezależne i fizycznie odizolowane lokalizacje (komunikacja pomiędzy regionami odbywa się poprzez internet). Każdy region składa się z co najmniej paru availability zone'ów.

## Availability Zones

Lokacje odseparowane od siebie w takim stopniu, by wykluczyć awarię kilku availability zone'ów naraz (niezależne zasilanie, chłodzenie, sieć, okna serwisowe).

Część usług w platformach cloudowych ma zasięg globalny (np. DNS, IAM).

Inne, np. bazodanowe, mogą mieć zasięg obejmujący region - dostawca platformy zapewnia replikację oraz fail-over pomiędzy availability zonami.

Część usług operuje na encjach w konkretnej availability zone (np. instancja).

Taki podział ma duży wpływ na architekturę aplikacji cloudowej i wymaga znajomości konkretnej platformy podczas projektowania rozwiązania.

# ” Wirtualizacja zasobów

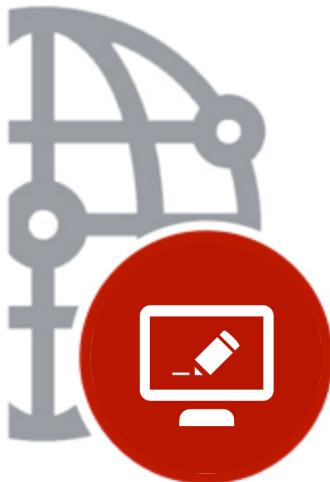


Wirtualizacja wprowadza separację pomiędzy systemem operacyjnym a fizycznymi zasobami. Taka separacja umożliwia większą elastyczność i łatwość zarządzania systemem operacyjnym przy niewielkim narzucie wydajnościowym (przyjmuje się ok. 2% na samą wirtualizację).

Tradycyjnie instalowany system operacyjny jest silnie uzależniony od hardware'u na którym działa (np. przeniesienie skonfigurowanego systemu na inny serwer jest praktycznie niemożliwe). Awaria sprzętowa często całkowicie uniemożliwia uruchomienie systemu.

Dzięki wirtualizacji, system zarządzany jest dedykowanym środowiskiem uruchomieniowym (tzw. **hypervisor**), które wprowadza dodatkową warstwę abstrakcji.

Wirtualizacja zasobów jest podstawowym narzędziem wykorzystywany w chmurach obliczeniowych.



## Migracja

przenoszenie instancji pomiędzy fizycznymi hostami (z tym samym hypervisorom)

## Konsolidacja

Uruchamianie kilku instancji na jednym hoście; przenoszenie wirtualnych instancji na wspólnego hosta w celu oszczędzenia energii, wykonania prac administracyjnych itp...

## Over-allocation

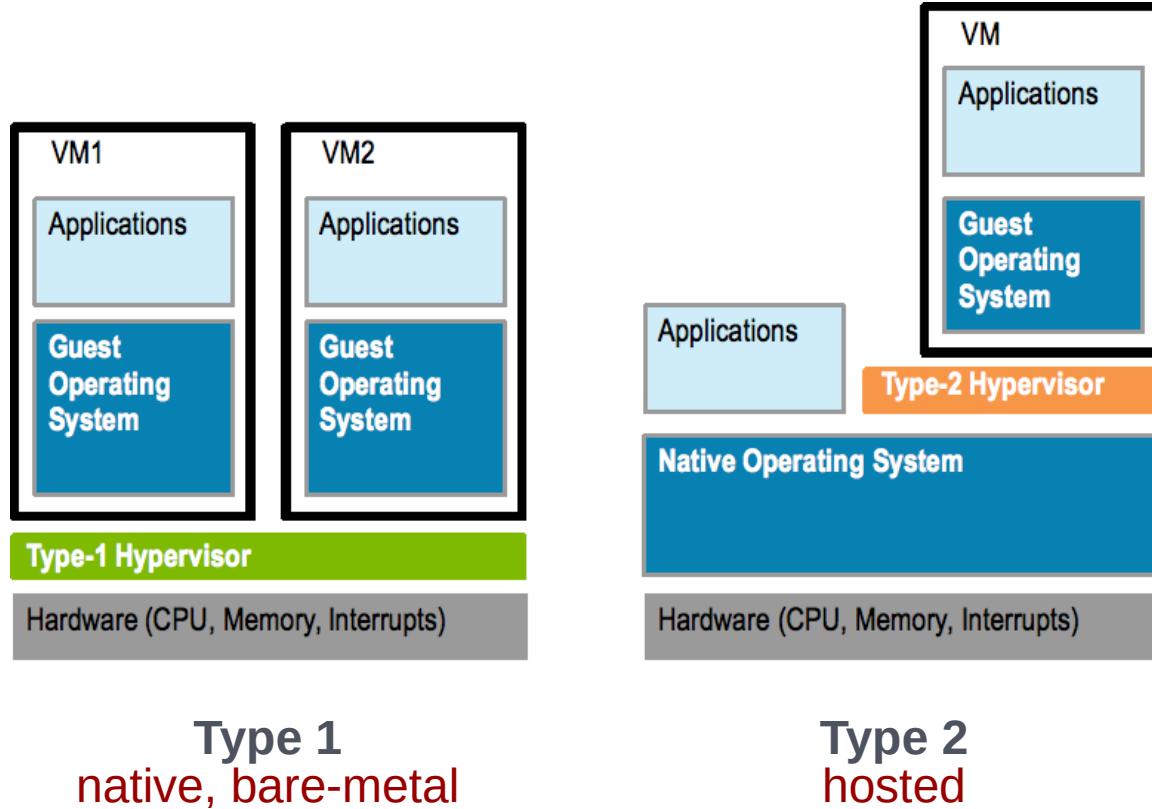
Dynamiczne przydzielanie fizycznych zasobów do instancji. Pozwala na lepsze wykorzystanie zasobów. Przeciwieństwem jest static allocation (gdzie hypervisor przydziela zasoby do instancji przy ich uruchamianiu).

## Snapshot

Zrzut stanu uruchomionej instancji, który można niezależnie przechowywać i uruchamiać.

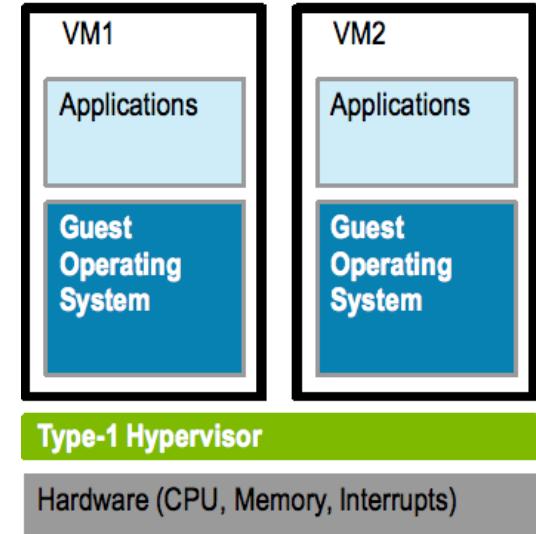
Mechanizm przyrostowy umożliwia tworzenie snapshotów działających instancji bez ograniczenia ich funkcjonalności (oraz z zapewnieniem spójności danych).

# Klasyfikacja Hypervisorów



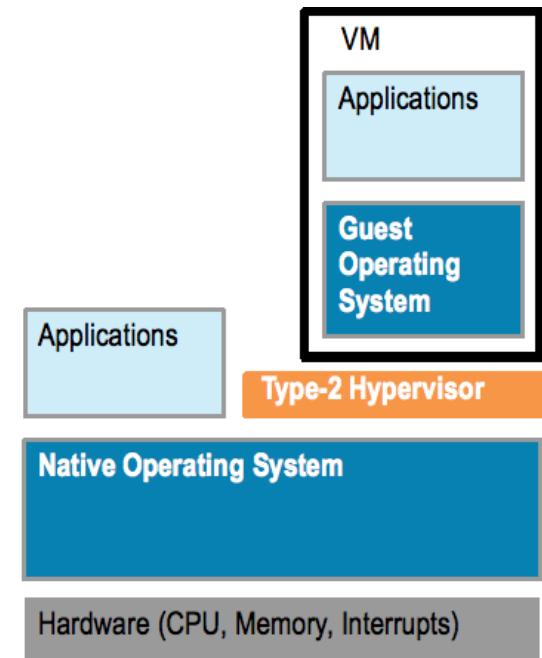


- Zaawansowany typ wirtualizacji, wykorzystywany produkcyjnie (w serwerowniach)
- Ponieważ fizyczne hosty nie posiadają swojego systemu operacyjnego, type 1 hypervisor **wymaga zdalnej konsoli administracyjnej** (aplikacji klienckiej umożliwiającej komunikację z hypervisorami), zarządzającej wszystkimi przypisanymi hostami i uruchomionymi na nichinstancjami
- Pozwala **automatyzować zarządzanie instancjami** (np. migracja instancji na podstawie profilu wykorzystania zasobów, konsolidacja w celu oszczędności energii, automatyzacja snapshotów jako backup, fail over)
- Umożliwia **over-allocation**





- Hypervisor uruchamiany jako **jedna z aplikacji hostującego systemu operacyjnego**
- Dużo **łatwiejszy w użyciu** (przy uruchamianiu pojedynczych instancji)
- **Statyczna alokacja zasobów**  
– istnieje możliwość przydzielenia zbyt dużej puli zasobów wirtualnyminstancjom (i zablokowanie hostującego systemu operacyjnego)



# Kontenery czyli „lekka wirtualizacja”



- Wykorzystanie wbudowanych mechanizmów jądra Linux (cgroups\*, namespaces\*\*)
- Tworzenie pseudo-izolowanych środowisk uruchomieniowych (na fizycznym sprzęcie, na wirtualnej maszynie)
- Kontener zawiera aplikacje i wszystkie jej wymagane zależności. Uruchamiany jest jako izolowany proces w tzw. “user-space”, współdzieli



\* control groups – mechanizm jądra Linux pozwalający na izolowanie zasobów dostępnych dla kolekcji procesów.

\*\* namespaces – mechanizm jądra Linux pozwalający na wirtualizację zasobów (np. system plików, dostęp do sieci) dostępnych dla kolekcji procesów.

# ” Cloud compute



Usługa cloud compute dostarcza skalowaną moc obliczeniową w chmurach obliczeniowych. Umożliwia konfigurację oraz uruchamianie wirtualnych instancji (**instance**, virtual machine).

Cloud compute jest najważniejszą usługą chmur obliczeniowych w modelu IaaS.

Przykładowe usługi cloud compute w platformach cloudowych:

- Amazon AWS EC2
- Google Compute Engine
- Windows Azure VM Roles
- OpenStack Nova

## Katalog obrazów

Zestaw prekonfigurowanych snapshotów, których można użyć do uruchomienia instancji

## Typ instancji

Definiuje profil zasobów wykorzystywanych przezinstancję (liczba rdzeni, pamięć RAM, architektura, wirtualizacja GPU itp)

## Klucze SSH

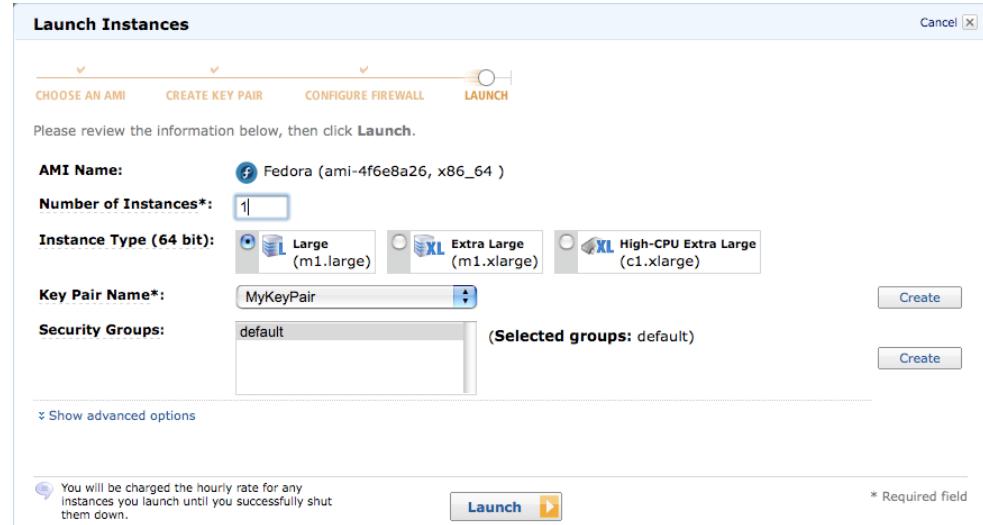
Usługa compute pozwala na wskazanie publicznego klucza uwierzytelniającego, który zostanie wgrany nainstancję podczas jej uruchamiania

## Security group

Definiuje reguły dostępu na poszczególne porty instancji (jak w firewallu)

## Kontekstualizacja

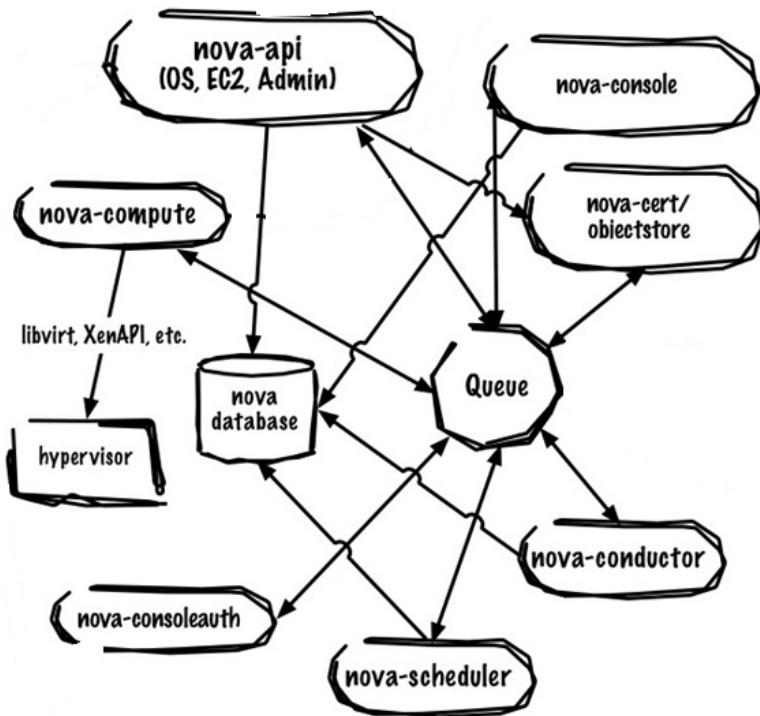
Definiowane przez użytkownika skrypty, które są uruchamiane podczas startu instancji (np. w celu jej skonfigurowania)





- Uruchamianie wirtualnych instancji w infrastrukturze chmury
- Kontekstualizacja wirtualnych instancji
- Komunikacja z hypervisorami
- Zarządzanie dostępnymi typami instancji
- Podział na regiony i availability zony
- Targetowanie instancji na fizyczne zasoby (w danej serwerowni)
- Skalowanie poziome i pionowe instancji

# Architektura usługi compute (OpenStack Nova)



**nova-api** komponent odpowiedzialny za uwierzytelnienie i translację wywołań HTTP (REST)

**queue** wewnętrzna szyna odpowiedzialna za komunikację między modułami (AMQP)

**nova-database** współdzielona baza MySQL w której poszczególne moduły przechowują swój stan

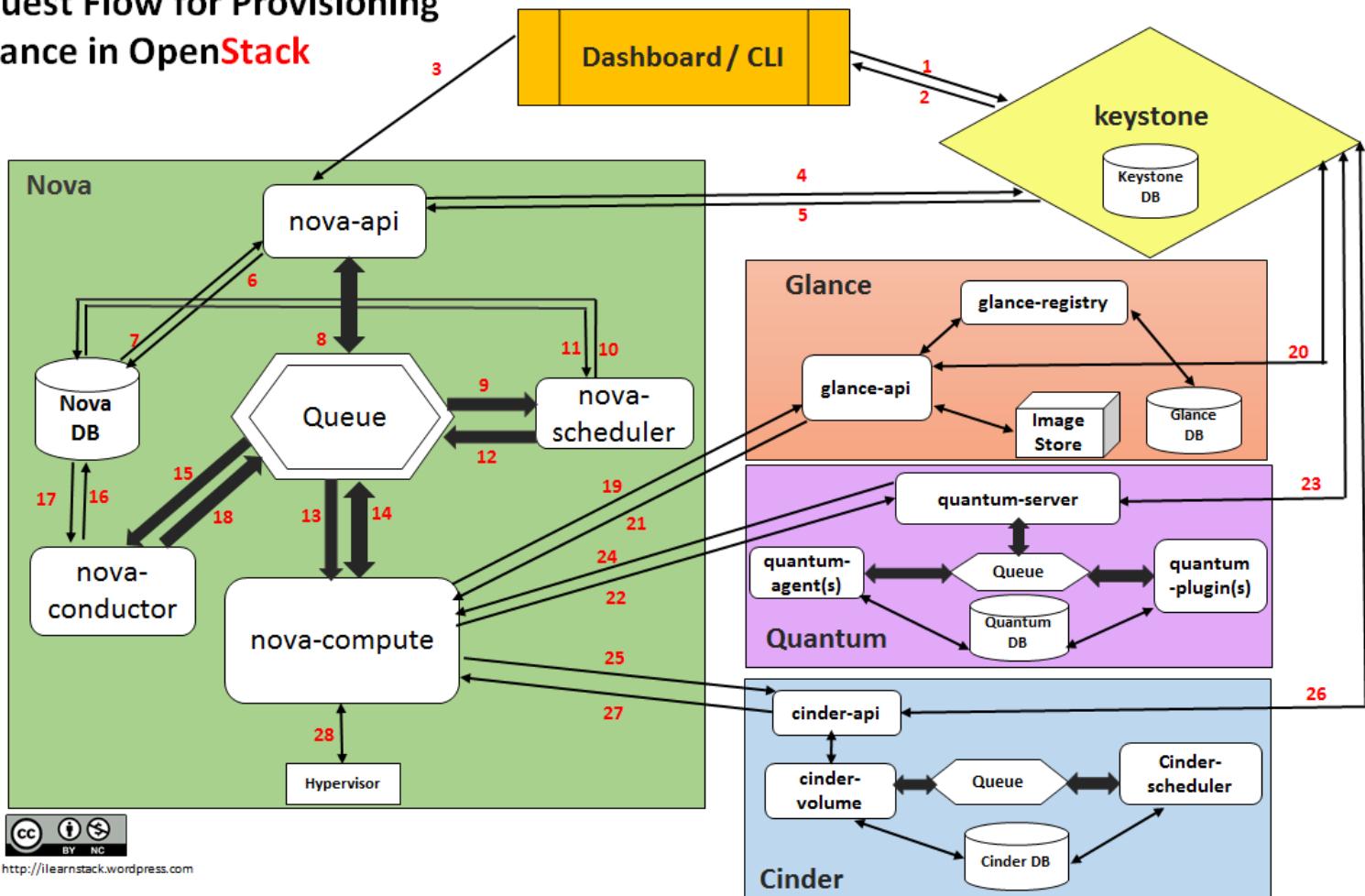
**nova-conductor** zarządza i orchiestruje komunikację z nova-compute

**nova-scheduler** implementuje logikę związaną z przydzielaniem instancji do fizycznych hostów

**nova-compute** adapter realizujący komunikację z hypervisorami iinstancjami (fizycznie, uruchomiony na każdym compute nodzie)

# Przepływ komunikacji dla OpenStack Nova

## Request Flow for Provisioning Instance in OpenStack



<http://ilearnstack.wordpress.com>



# Wymagania stawiane przed usługami compute

## Automatyzacja

Możliwość automatycznej konfiguracji instancji w celu zredukowania „czynnika ludzkiego”

## Bezpieczeństwo

Możliwość pełnej konfiguracji dostępu do instancji (security group, klucze)

## Dostępność

Uruchamianie instancji w odseparowanych fizycznie serwerowniach, fail-over

## Efektywność

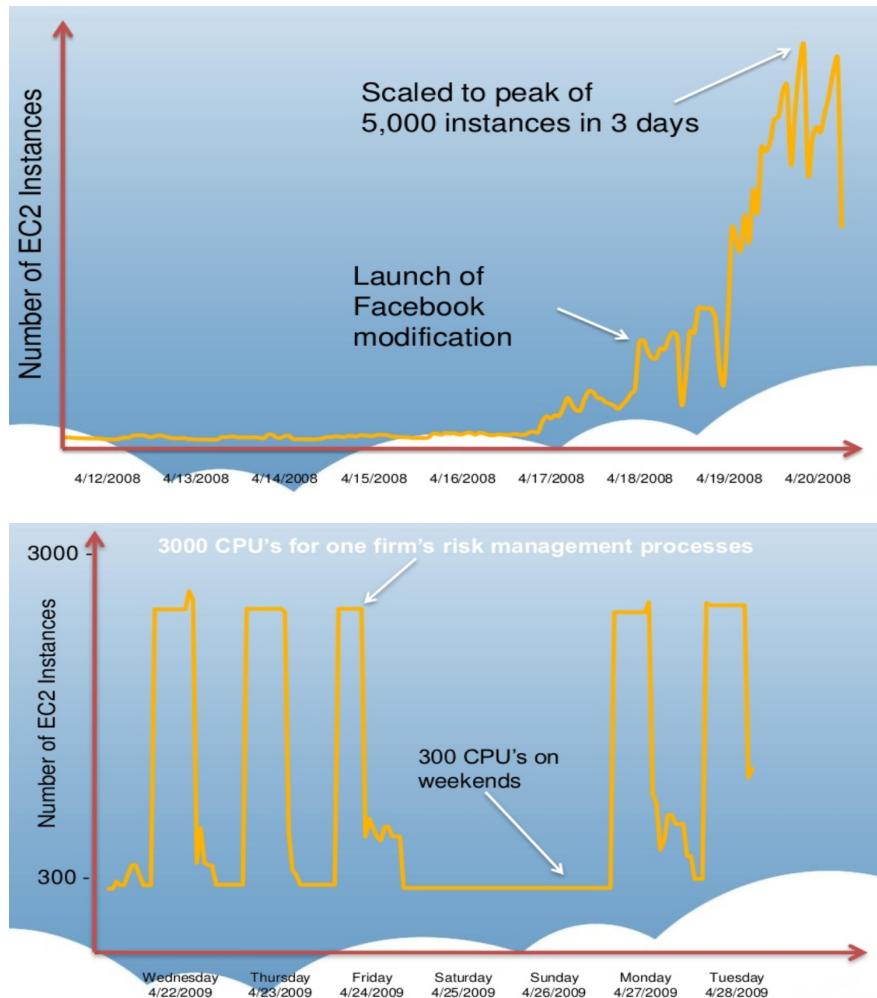
...w zarządzaniu i monitorowaniu instancji

## Skalowalność

Dopasowanie używanych zasobów do wymagań uruchamianych aplikacji

## Elastyczność

Wiele sposobów konfigurowania i konfigurowania instancji



# ”Cloud storage



Politechnika Łódzka  
Wydział Elektrotechniki, Elektroniki,  
Informatyki i Automatyki



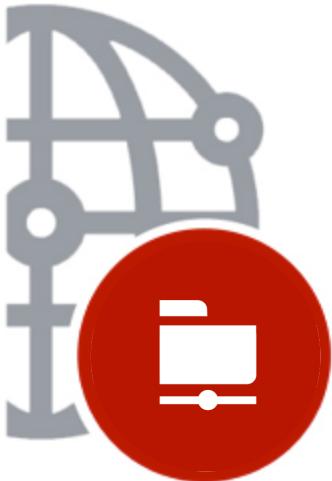


Usługi cloud storage zapewniają skalowalne i trwałe rozwiązania umożliwiające przechowywanie danych w chmurach obliczeniowych.

W zależności od przeznaczenia, usługi cloud storage dzielimy na **obiektowe** oraz **blokowe**.

Przykładowe usługi cloud storage w platformach cloudowych:

- Amazon AWS S3, EBS, Glacier
- OpenStack Swift, Cinder
- Google Cloud Storage
- Windows Azure BLOB, Drive



Standardowa przestrzeń dyskowa instancji jest ulotna – jeśli instancja zostanie zatrzymana (ręcznie lub na skutek awarii), dane zostaną utracone.

Usługi block store zarządzają tzw. **volumami**, czyli persistentnymi plikami urządzenia reprezentującymi wirtualne partycje, które mogą być montowane do wirtualnych instancji .

Zamontowane partycje mogą być następnie wykorzystywane przez system operacyjny instancji jak normalne dyski (po sformatowaniu z użyciem wskazanego systemu plików).

Usługi block storage zapewniają dodatkowe funkcjonalności, jak tworzenie snapshotów, klonowanie, przepinanie pomiędzy instancjami (volume może być w danym momencie zamontowany tylko w jednej instancji).

Fizycznie, usługi block storage wykorzystują zreplikowaną (z reguły software'owo, ze względu na znacznie niższe koszty) przestrzeń dyskową na wypadek awarii sprzętowej.



Usługi object storage to wysoce skalowalne i redundantne systemy przechowywania danych.

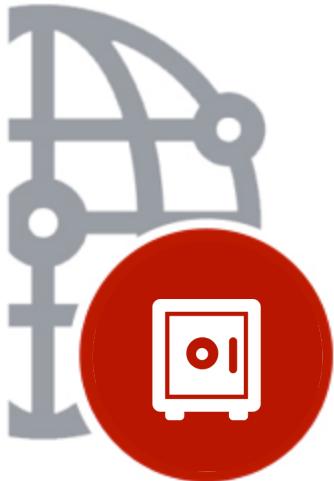
Logicznie, dane użytkownika przechowywane są jako obiekty, grupowane w kontenery (buckets, containers).

Kontenery i obiekty są wielokrotnie replikowane pomiędzy różnymi fizycznymi dyskami w różnych (izolowanych fizycznie) serwerowniach, w celu zapewnienia bardzo wysokiego SLA (np. Amazon AWS S3 zapewnia SLA dla utraty danych na poziomie 99.999 999 999%).

Usługi object storage same (software'owo) zapewniają replikację i zachowanie integralności danych, dzięki czemu nie wymagają zaawansowanych (czytaj – drogich) serwerów i dysków.



- Binaria aplikacji uruchamianych „w compute”
- Statyczny kontent aplikacji
- Snapshoty (wykorzystywane obrazy instancji)
- Dane wejściowe/wyjściowe przetwarzania
- Backup krytycznych danych (np. do disaster recovery)
- Aktualne pliki logów (np. z ostatniego tygodnia, do których potrzebujemy stosunkowo szybki dostęp)



Niektóre platformy cloudowe oferują usługi dedykowane do archiwizacji danych i backupu (np. AWS Glacier).

Organizacja danych jest analogiczna do usług object storage, jednak dostęp do zarchiwizowanych danych nie jest natychmiastowy (request dostępu do obiektu przetwarzany jest asynchronicznie i z reguły trwa kilka godzin).

Cena korzystania z usług archiwizacji i backupu jest wielokrotnie niższa niż w object storze.

Usługi object storage są zintegrowane z usługami archiwizacji, dzięki czemu możliwe jest definiowanie cyklu życia obiektów (automatyczna archiwizacja, usuwanie przedawnionych obiektów).



- Archiwalne pliki logów
- Archiwalne dokumenty
- Archiwa multimedialne
- Dane finansowe lub medyczne
- Dane wejściowe do przetwarzania wsadowego (po przetworzeniu, ale mogą być kiedyś ponownie potrzebne)
- Archiwalne backupy baz danych
- Dane, które muszą być archiwizowane ze względu na prawnych

# ” Cloud networking



Usługi cloud networking pozwalają na zarządzanie różnymi aspektami związanymi z siecią. Ich głównymi rolami jest zapewnienie **bezpieczeństwa** oraz **dostępu** do instancji uruchamianych w chmurach obliczeniowych.

Przykładowe usługi cloud networking w platformach cloudowych:

- Amazon AWS VPC, Route 53, ELB
- OpenStack Quantum, Moniker
- Windows Azure CDN, Traffic Manager



Software-defined networking to koncepcja, w której konfiguracja infrastruktury sieciowej (tzw. control plane) jest zwirtualizowana i odseparowana od fizycznych zasobów (data plane).

SDN umożliwia deklaratywną konfigurację infrastruktury sieciowej oraz separację zasobów użytkowników w chmurach obliczeniowych.

Dzięki wirtualizacji topologii sieciowej, użytkownicy mogą dynamicznie modyfikować dostęp do swoich instancji:

- Definiować podsieci i przypisywać do nich instancje
- Konfigurować routing
- Przypisywać publiczne adresy IP
- Określać reguły dostępu dla poszczególnych podsieci
- Tworzyć wirtualne firewalle definiujące reguły ingress/egress dla instancji

## **Virtual Network (Virtual Private Cloud)**

Wirtualna sieć dedykowana dla konta (użytkownika), logicznie odseparowana od innych wirtualnych sieci.

## **Subnet**

Zakres adresów IP w wirtualnej sieci. Podsieci mogą być prywatne lub publiczne – te drugie dostępne są z internetu. Instancje przypisywane są do podsieci poprzez interfejsy siecowe (automatycznie wybierany jest adres z zakresu podsieci, lub użytkownik jawnie wskazuje adres, który chce nadać instancji).

## **Floating IP (Elastic IP)**

„Publiczny” adres IP, który możemy przypisać instancjom dostępnym z internetu.

## **Router**

Zapewnia komunikację pomiędzy komponentami wewnętrz wirtualnej sieci

## **Internet Gateway**

Umożliwia komunikację instancji w publicznych podsieciach z internetem.

## **Access Control List, ACL**

Definiują reguły dostępu ingress/egress do podsieci; Odpowiednik security group dla podsieci.

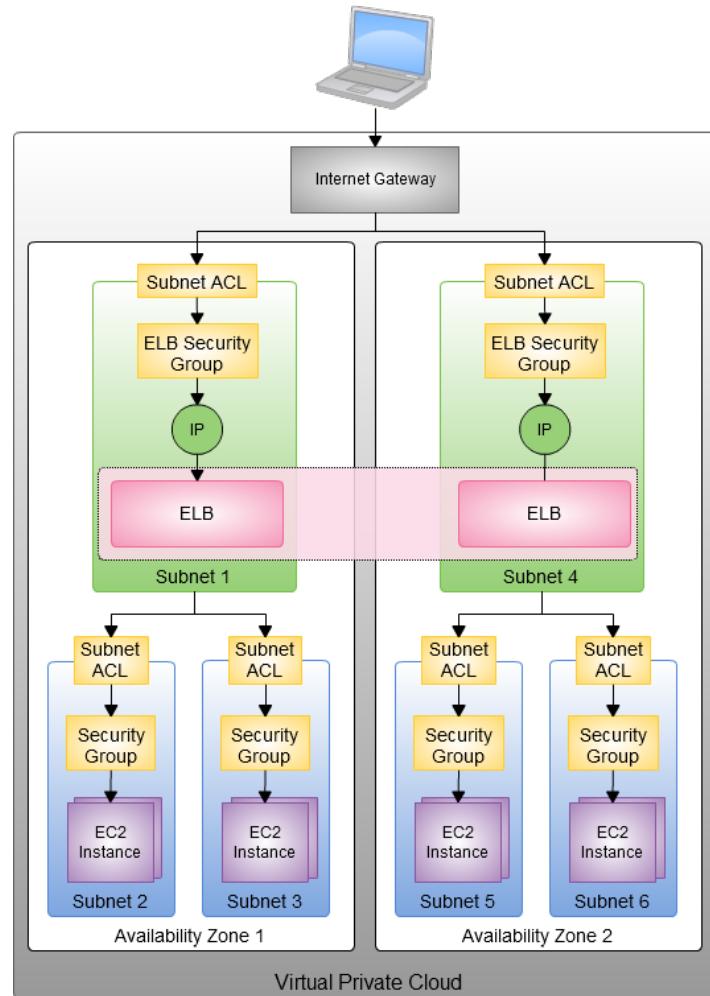
## **Network Address Translation, NAT**

Umożliwia komunikację outbound (inicjowaną z instancji) z internetem dla instancji bez publicznego adresu IP.

## **Virtual Private Gateway**

Umożliwia zestawienie połączenia VPN do wirtualnej sieci.

# Topologia sieciowa w chmurze





Load balancery odpowiedzialne są za kierowanie ruchu do poszczególnych instancji w celu **wyrównania ich obciążenia**.

Load balancery muszą umieć rozpoznawać instancje przeciążone lub zgaszone (i odpowiednio modyfikować swoją konfigurację). Wykorzystują w tym celu **mechanizm health check** – cyklicznie wysyłają komunikaty do wszystkich instancji i określają ich status na podstawie czasu odpowiedzi.

Przykładowy algorytm load balancingu w chmurach (np. AWS ELB) to **leastconns**, w którym load balancer wysyła nowe requesty do instancji, która obsługuje aktualnie najmniej requestów. (w ten sposób reagując na losowe problemy wydajnościowe instancji).

Usługi load balancera mogą działać w obrębie availability zone (tzw. **instance availability**), lub Cross-Zone (**zonal availability**).

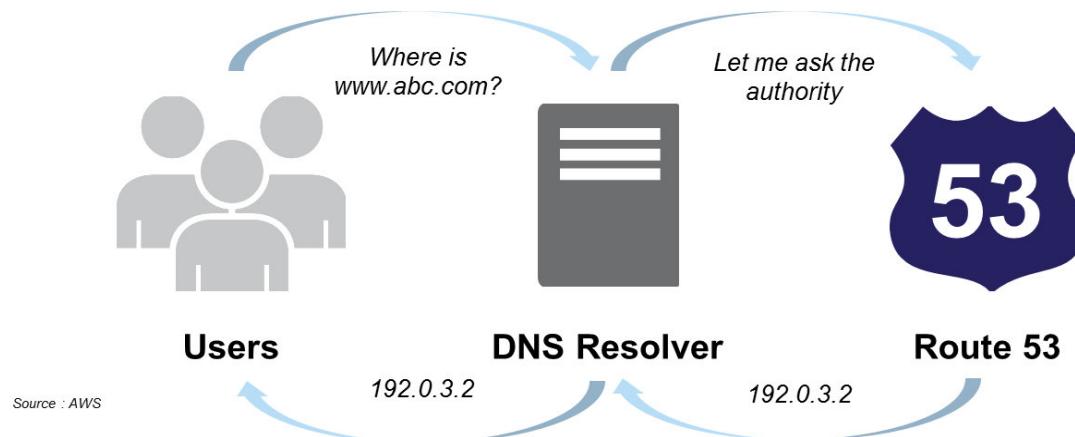
Load balancery również muszą się skalować, żeby obsłużyć przychodzący ruch (jednak dzieje się to w sposób przezroczysty dla klienta).



## Domain Name System (DNS, system nazw domenowych)

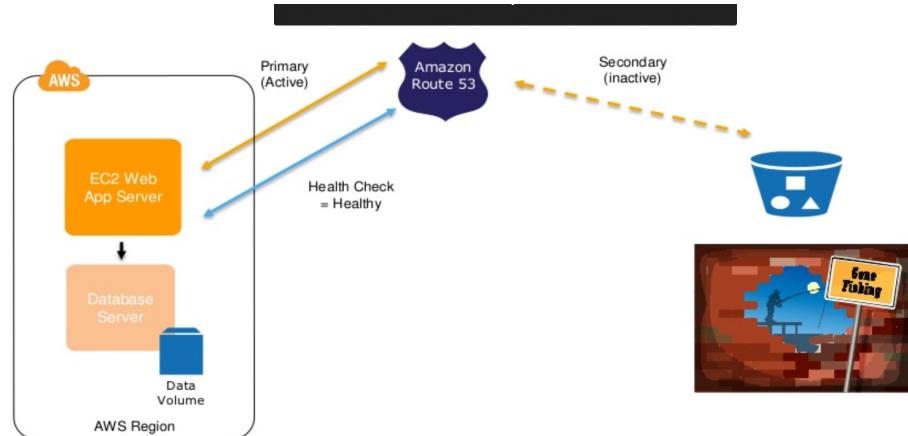
Usługa pozwalająca na translację adresów znanych użytkownikom Internetu (tzw. nazw mnemonicznych, np. `www.example.org`) na adresy zrozumiałe dla urządzeń tworzących sieć komputerową (adresy IP, np. `192.0.3.6`).

Technicznie, DNS to **rozproszona baza danych** adresów sieciowych, analogiczna do książki telefonicznej.



DNS as a Service to skalowalna, wysokowydajna usługa platform cloudowych, umożliwiająca:

- Zarządzanie rekordami DNS poprzez **web API**
- **Dynamiczny routing** użytkowników do najbliższego regionu
- Integrację z innymi usługami platformy cloudowej (np. konfiguracja DNS dla instancji, floating IP, load balancerów).
- Mechanizm **DNS failover** (w obrębie jednego lub pomiędzy wieloma regionami)



Latency Based Routing bazuje na **czasach odpowiedzi** z poszczególnych regionów. Zapewnia tym samym dynamiczny routing użytkowników do optymalnego regionu.



Usługa DNS próbuje poszczególne regiony z różnych lokalizacji i dynamicznie modyfikuje tabele routingu w zależności od statusu i czasu odpowiedzi.

W przypadku awarii w jednym z regionów (lub, po prostu, zgaszenia w nim aplikacji), LBR wraz z mechanizmem health checków zapewni **fail-over** – przekieruje użytkownika do innych regionów, w których aplikacja jest dostępna.





wydział  
elektrotechniki  
elektroniki  
informatyki  
i automatyki

**AMG.net**  
A Bull Group Company



Dziękujemy za uwagę