Duże zbiory danych
big data

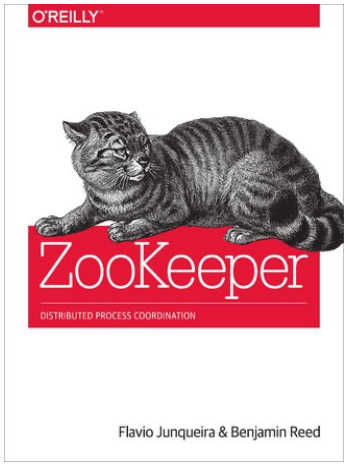
dr hab. inż. Przemysław Korytkowski, prof. ZUT

Wykład 6

1

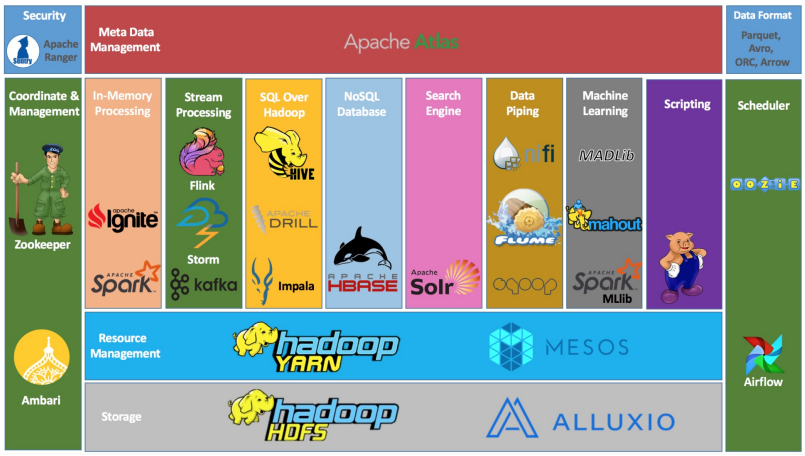
Wykład powstał na podstawie

- Flavio Junqueira, Benjamin Reed (2014) ZooKeeper. Distributed Process, O'Reilly



2

2

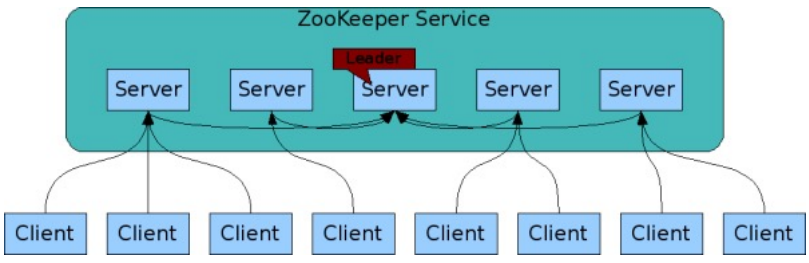


3

3

Apache ZooKeeper

- Rozproszona usługa koordynacyjna dla aplikacji rozproszonych, początkowo opracowany w Yahoo!
- W przeszłości każda aplikacja była pojedynczym programem działającym na pojedynczym komputerze z pojedynczym procesorem.
- W świecie Big Data i Cloud Computing, aplikacje składają się z wielu niezależnych programów działających na ciągle zmieniającym się zestawach komputerów.
- Koordynowanie działań tych niezależnych programów jest znacznie trudniejsze niż napisanie pojedynczego programu, który ma działać na jednym komputerze.



4

4

Wyzwania systemów rozproszonych

Opóźnienia komunikatów
- Komunikaty mogą być losowo opóźnione, na przykład z powodu przeciążenia sieci. Takie losowe opóźnienia mogą wprowadzać niepożądane sytuacje. Na przykład, proces P może wysłać wiadomość zanim inny proces Q wyśle swoją wiadomość, zgodnie z zegarem referencyjnym ale komunikat Q może zostać dostarczony jako pierwszy.

Szybkość procesora
- Harmonogramowanie i przeciążenie systemu operacyjnego może powodować losowe opóźnienia w przetwarzaniu komunikatów. Kiedy jeden proces wysyła komunikat do drugiego, całkowite opóźnienie tego komunikatu jest w przybliżeniu sumą czasu przetwarzania u nadawcy, czasu transmisji i czasu przetwarzania u odbiorcy. Jeśli proces wysyłający lub odbierający wymaga czasu na zaplanowanie przetwarzania, wówczas opóźnienie komunikatu jest wyższe.

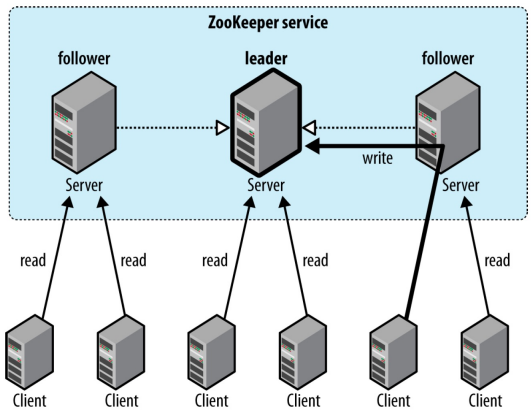
Dryft zegara
- Nierzadko spotyka się systemy, które wykorzystują pewne pojęcie czasu, np. przy określaniu czasu, w którym zdarzenia występują w systemie. Zegary procesorów nie są niezawodne i mogą oddalać się od siebie. W związku z tym, poleganie na zegarach procesora może prowadzić do błędnych decyzji.

5

5

Zarządzanie klastrem

- Udostępnia on API, inspirowane API systemu plików, które pozwala programistom na implementację typowych zadań koordynacyjnych, takich jak:
  - wybór serwera głównego,
  - zarządzanie członkostwem w grupach
  - zarządzanie metadanymi.
- ZooKeeper jest biblioteką aplikacji (API Java i C) oraz komponentem usługowym zaimplementowanym w Javie, który działa na zestawie dedykowanych serwerów.



6

6

Zookeeper

API Zookeeper zapewnia:

- Silną gwarancję spójności, uporządkowania i trwałości.
- Możliwość implementacji typowych prymitywów synchronizacji.
- Prostszy sposób radzenia sobie z wieloma aspektami współbieżności, które często prowadzą do niepoprawnego zachowania w rzeczywistych systemach rozproszonych

Zookeeper nie służy do:

- Przechowywania danych aplikacji.
- Projektując aplikację z ZooKeeperem, najlepiej oddzielić dane aplikacji od danych kontrolnych lub koordynacyjnych.

7

7

Cechy Zookeeep'era

- **Spójność sekwencyjna** - aktualizacje od poszczególnych klientów są stosowane w kolejności ich wysyłania. Oznacza to, że jeśli klient aktualizuje węzeł znode z do wartości a, a następnie aktualizuje znode z do wartości b, to żaden klient nie zobaczy znode z wartością a po tym, jak zobaczy znode z wartością b (jeśli żadne inne aktualizacje nie zostaną wykonane na znode z).
- **Atomowość** - aktualizacje albo się udają albo nie. Oznacza to, że jeśli aktualizacja się nie powiedzie, żaden klient nigdy go zobaczyć.
- **Pojedynczy obraz systemu** - klient zobaczy ten sam widok systemu, niezależnie od serwera, z którym się łączy. Oznacza to, że jeśli klient połączy się z nowym serwerem podczas tej samej sesji, nie zobaczy starszego stanu systemu niż ten, który widział na poprzednim serwerze. Gdy serwer ulegnie awarii i klient próbuje połączyć się z innym w zespole, serwer którego stan jest wcześniejszy niż ten, który uległ awarii, nie będzie akceptował połączeń od klienta, dopóki dopóki nie dogoni serwera, który uległ awarii.
- **Trwałość** - gdy aktualizacja się powiedzie, będzie trwała i nie będzie można jej cofnąć. Oznacza to, że aktualizacje przetrwają awarie serwerów.
- **Aktualność** - opóźnienie w widoku systemu dla każdego klienta jest ograniczone, więc nie będzie on nieaktualny o więcej niż wielokrotność kilkudziesięciu sekund. Oznacza to, że zamiast pozwalać klientowi klientowi zobaczyć dane, które są bardzo nieświeże, serwer zostanie zamknięty, zmuszając klienta do przełączenia się na bardziej aktualny serwer.

8

8

Zookeeper używany jest przez:

- Hadoop
  - Wykrywanie awarii - każda z maszyn NameNode w klastrze utrzymuje stałą sesję w ZooKeeper. Jeśli maszyna ulegnie awarii, sesja ZooKeeper wygaśnie, informując inne NameNode o konieczności uruchomienia procedury przełączenia.
  - Wybór aktywnego węzła NameNode - ZooKeeper zapewnia mechanizm wyboru węzła jako aktywnego. Jeśli aktualny aktywny węzeł NameNode ulegnie awarii, inny węzeł może przejąć specjalną blokadę wyłączności w ZooKeeper, wskazując, że powinien stać się kolejnym aktywnym węzłem.
- Hbase
  - jest używany do wybierania mastera klastra, śledzenia dostępności serwerów i przechowywania metadanych klastra.
- Hive
  - rozproszony menedżer blokad do obsługi współbieżności w HiveServer2
- Kafka
  - Jeż używany do wykrywania awarii, do implementacji wyszukiwania tematów oraz do utrzymywania stanu produkcji i konsumpcji dla tematów.
- Neo4j
  - Jest używany w komponentach wysokiej dostępności Neo4j do wyborów write-master, koordynacji read slave.

9

9

Architektura master-slave – wyzwania

- Awaria mastera
  - Jeśli master jest uszkodzony i staje się niedostępny, system nie może przydzielać nowych zadań lub ponownie przydzielać zadań od robotników (slave'ów), które również uległy awarii.
- Awaria robotnika (slave)
  - Jeśli robotnik ulegnie awarii, przydzielone mu zadania nie zostaną ukończone.
- Awarie komunikacji
  - Jeżeli master i robotnik nie mogą wymieniać wiadomości, robotnik może nie dowiedzieć się o nowych zadaniach, które zostały mu przydzielone.

10

10

Zadania realizowane przez Zookeepera

1. Wybór mastera
   Krytyczne dla postępu jest posiadanie dostępnego mastera, który przydziela zadania robotnikom.
2. Wykrywanie awarii
   Master musi być w stanie wykryć, kiedy robotnicy ulegają awarii lub rozłączają się.
3. Zarządzanie członkostwem w grupie
   Master musi być w stanie dowiedzieć się, którzy robotnicy są dostępni do wykonywania zadań.
4. Zarządzanie metadanymi
   Master i robotnicy muszą być w stanie przechowywać zadania i statusy ich wykonania w niezawodny sposób.

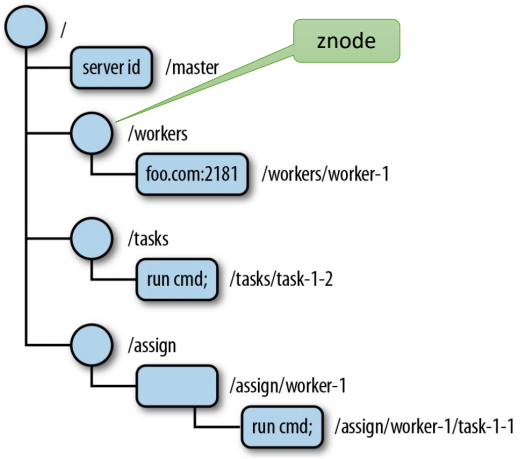
11

11

Drzewo danych

Zookeeper API:

- create /path data
- delete /path
- exists /path
- setData /path data
- getData /path
- getChildren /path



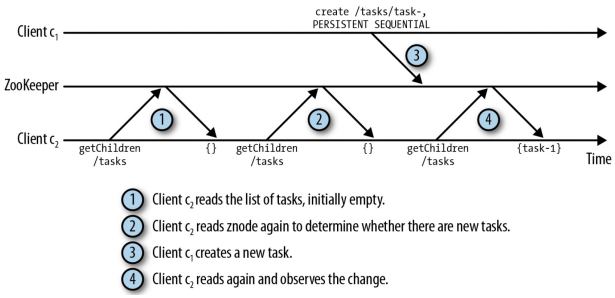
12

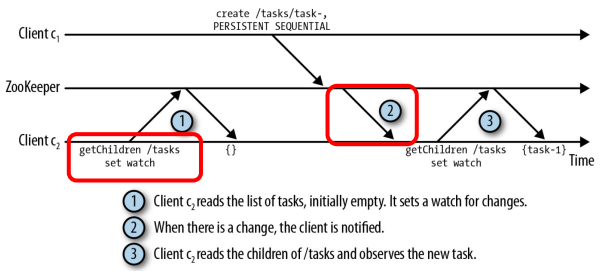
12

znode

- Trwały (persistent)
  Przechowuje informacje nawet gdy serwer, który go utworzył nie jest już aktywny (nie jest częścią systemu).
- Efemeryczny (ephemeral)
  Jego istnienie oznacza, że serwer, który go utworzył jest aktywny. Nie mogą mieć potomków.
- Sekwencyjny (sequential)
  Zarówno trwały jak i efemeryczny znode może być sekwencyjny
  Nazwa znode'a uzupełniana jest o kolejny numer

Czujki (watches) i powiadomienia (notifications)



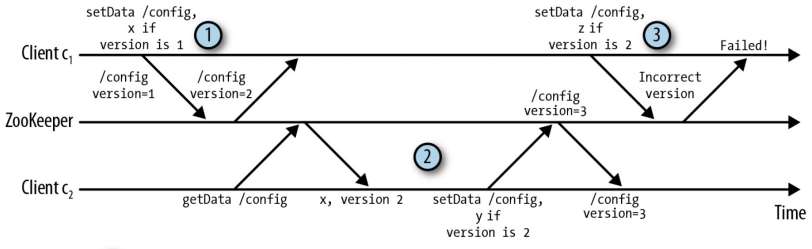


- Założenie czujki oznacza otrzymanie jednego powiadomienia.
- Od wersji 3.6.0 (marzec 2020) dodano stałe czujki.
- Aby otrzymać kolejna powiadomienia należy założyć nową czujkę.
- Istnieje ryzyko nieotrzymania powiadomień o wszystkich modyfikacjach.

Typy czujek:
1. NodeCreated
2. NodeDeleted
3. NodeDataChanged
4. NodeChildrenChanged

Wersje znode

- Każdy znode ma przypisaną wersję, która jest inkrementowana po każdym zapisie.

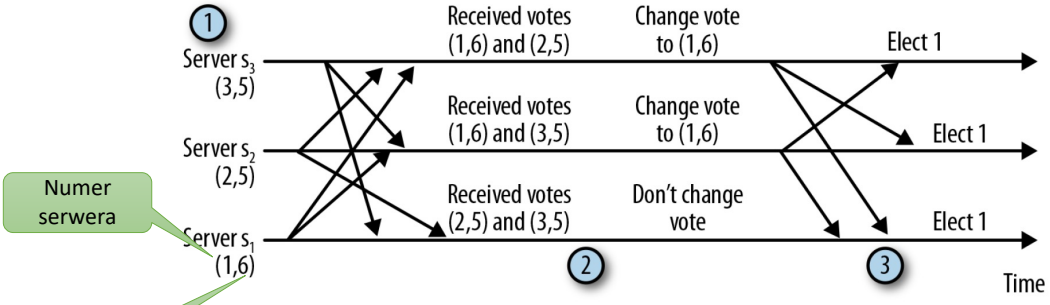


1. Client c_1 writes the first version of /config.
2. Client c_2 reads /config and writes the second version.
3. Client c_1 tries to write a change to /config, but the request fails because the version does not match.

15

15

Wybór lidera (mastera)

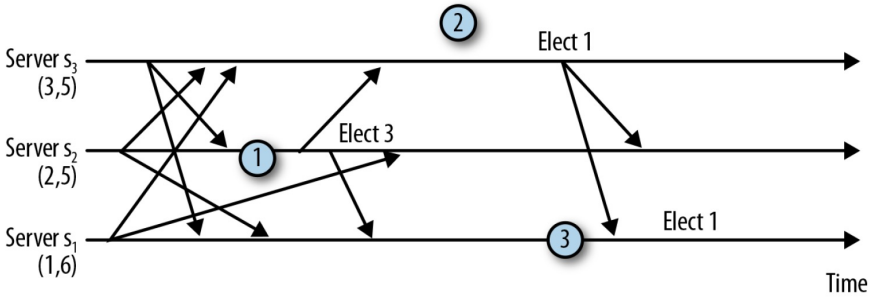


1. Server s_1 starts with vote (1,6), server s_2 with (2,5) and server s_3 with (3,5).
2. Servers s_2 and s_3 change their vote to (1,6) and send a new batch of notifications.
3. All three servers receive the same vote from a quorum and elect server s_1.

16

16

Wybór lidera (mastera)

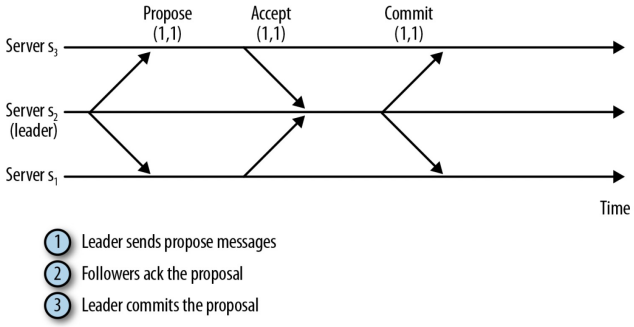


17

17

ZAB – Zookeeper Atomic Broadcast Protocol

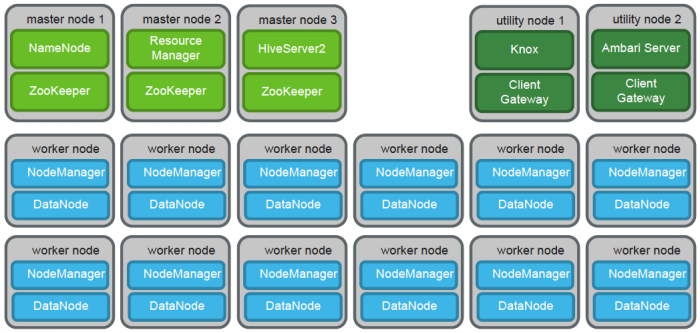
1. Lider wysyła wiadomość PROPOSAL p do wszystkich serwerów.
2. Po otrzymaniu wiadomości p, serwer odpowiada liderowi komunikatem ACK, informując go, że przyjął propozycję.
3. Po otrzymaniu potwierdzenia od kworum (kworum obejmuje samego lidera), lider wysyła wiadomość informującą zwolenników, aby ją COMMIT.



18

18

Klaster Hadoop



19

19

Chmury publiczne

Chmury prywatne

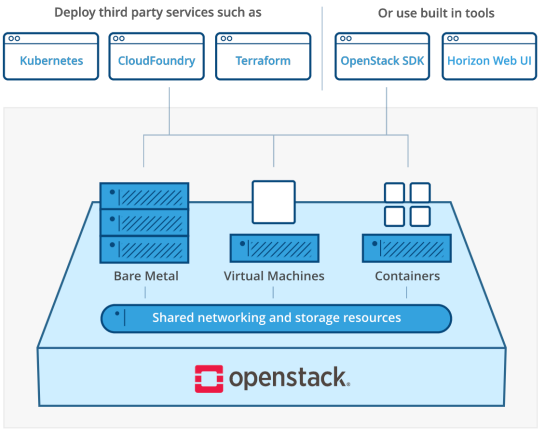
Hadoop i jego usługi można uruchomić wszędzie!

20

20

OpenStack

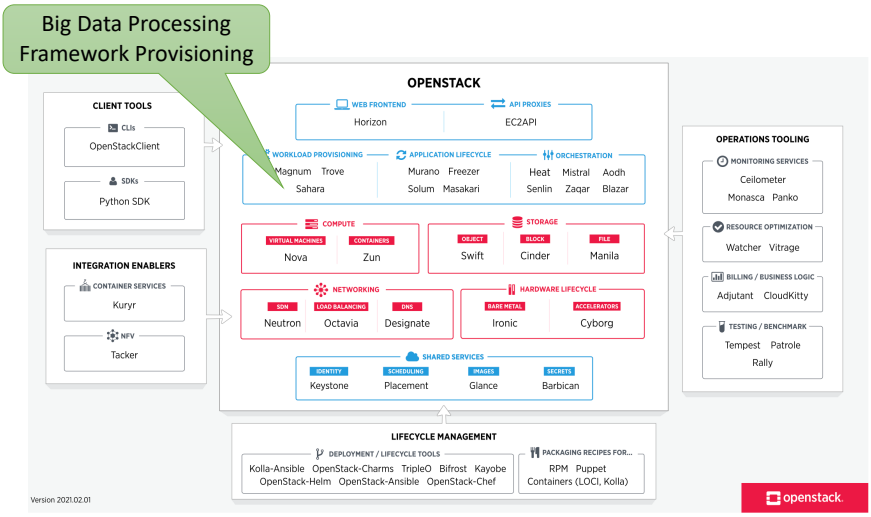
- **Free Cloud Infrastructure for Virtual Machines, Bare Metal, and Containers**
- Openstack controls large pools of compute, storage, and networking resources, all managed through APIs or a dashboard.
- It is mostly deployed as infrastructure-as-a-service (IaaS) in both public and private clouds where virtual servers and other resources are made available to users.
- Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.



21

21

OpenStack



22

22

OpenStack Sahara

Sahara aims to provide users with a simple means to provision Hadoop, Spark, and Storm clusters by specifying several parameters such as the framework version, cluster topology, hardware node details and more. After a user fills in all the parameters, sahara deploys the cluster in a few minutes. Also sahara provides means to scale an already provisioned cluster by adding or removing worker nodes on demand.

The solution will address the following use cases:

- fast provisioning of data processing clusters on OpenStack for development and quality assurance(QA).
- utilization of unused compute power from a general purpose OpenStack IaaS cloud.
- “Analytics as a Service” for ad-hoc or bursty analytic workloads (similar to AWS EMR).

23

23

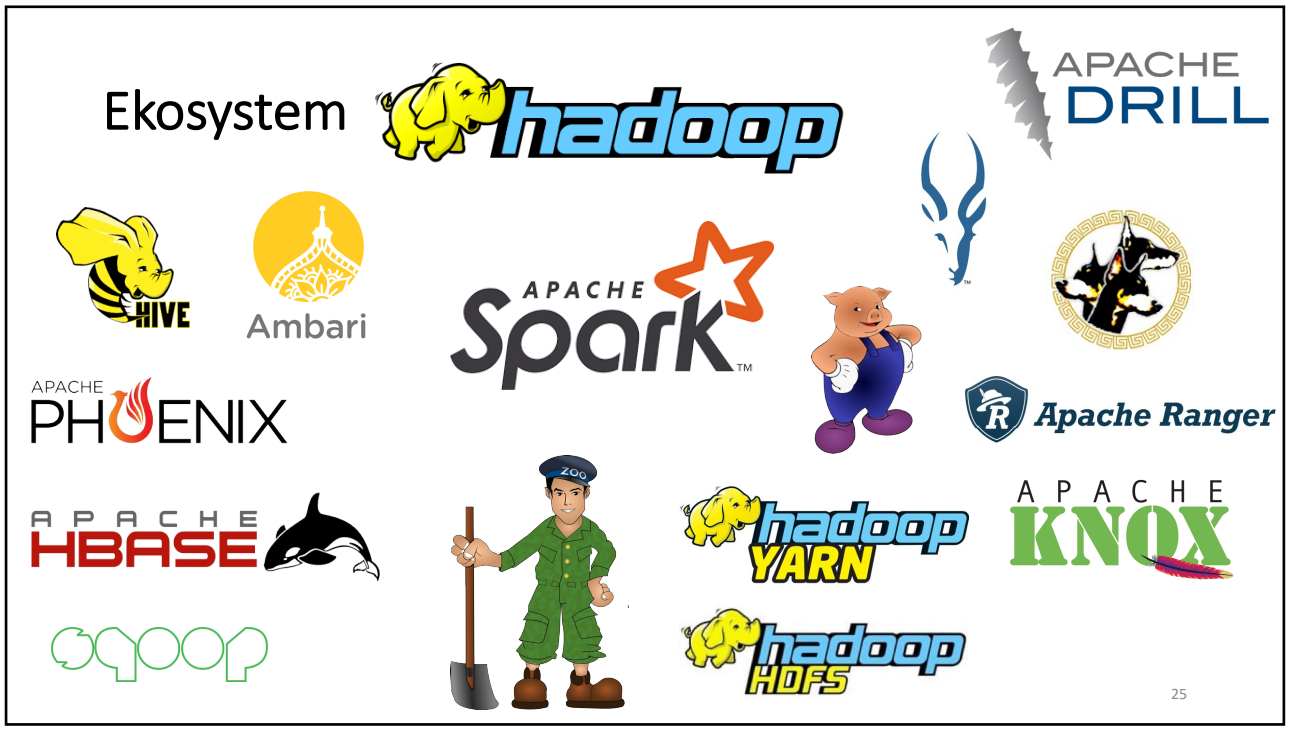
OpenStack Sahara

Key features are:

- managed through a REST API with a user interface(UI) available as part of OpenStack Dashboard.
- support for a variety of data processing frameworks:
  - multiple Hadoop vendor distributions.
  - Apache Spark and Storm.
  - pluggable system of Hadoop installation engines.
  - integration with vendor specific management tools, such as Apache Ambari and Cloudera Management Console.
- predefined configuration templates with the ability to modify parameters.

24

24



25

Dziękuję za uwagę!

26