Jakarta EE Meets NoSQL in the Cloud Age

Otavio Santana

@otaviojava

platform.sh

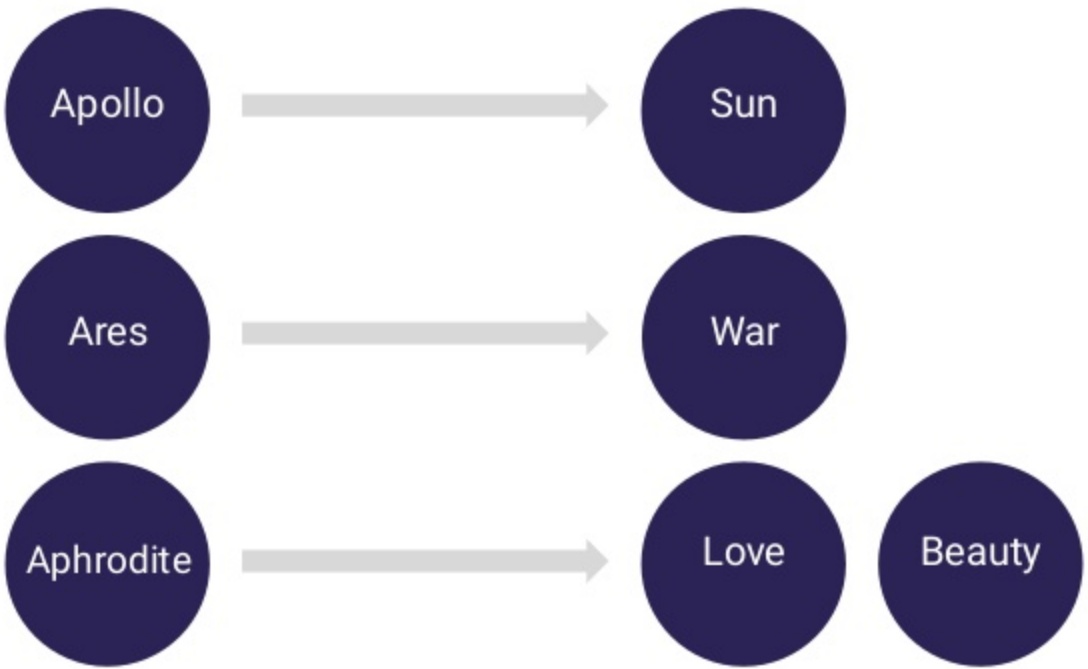
NoSQL

NoSQL

- Database
- Doesn't use structure
- Not Transaction
- Base
- Five different types

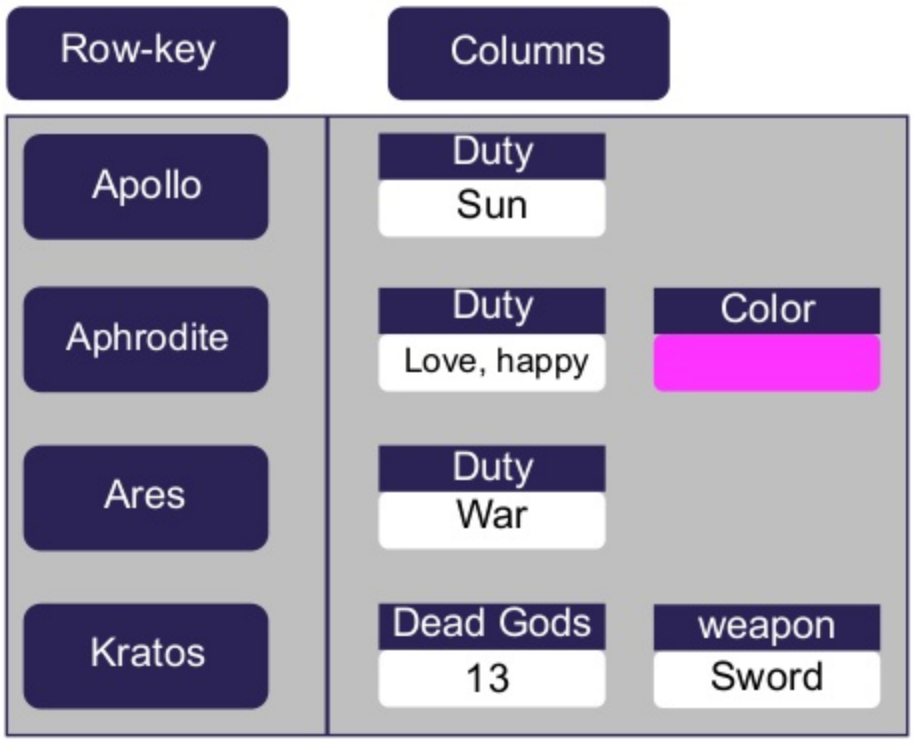
Key Value

- AmazonDynamo
- AmazonS3
- Redis
- Hazelcast



Column Family

- HBase
- Cassandra
- Clouddata
- DynamoDB



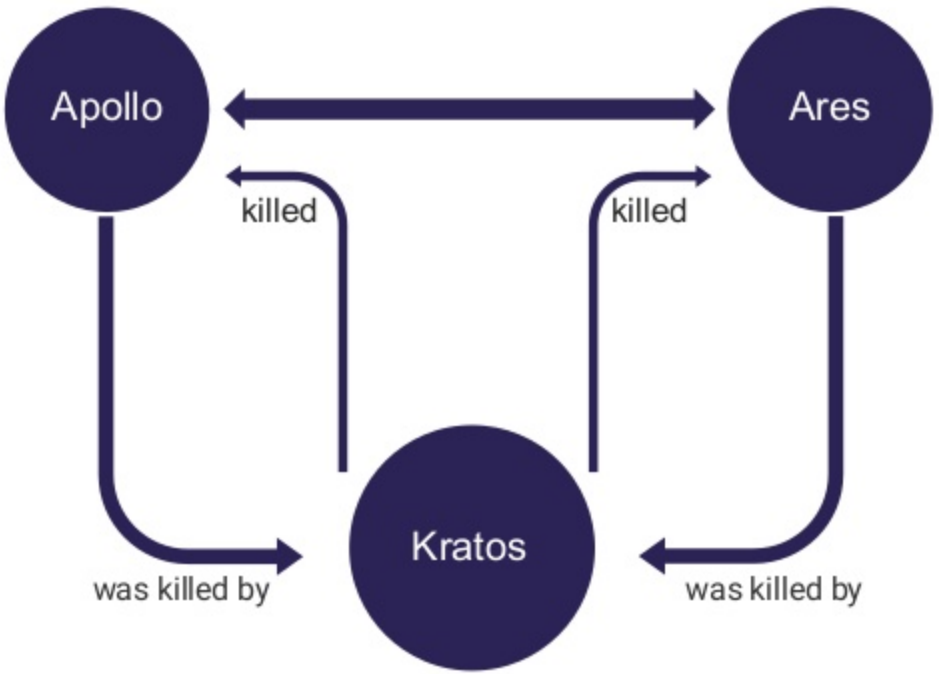
Document

- ApacheCouchDB
- MongoDB
- Couchbase

```
{
    "name":"Diana",
    "duty":[
        "Hunt",
        "Moon",
        "Nature"
    ],
    "siblings":{
        "Apollo":"brother"
    }
}
```

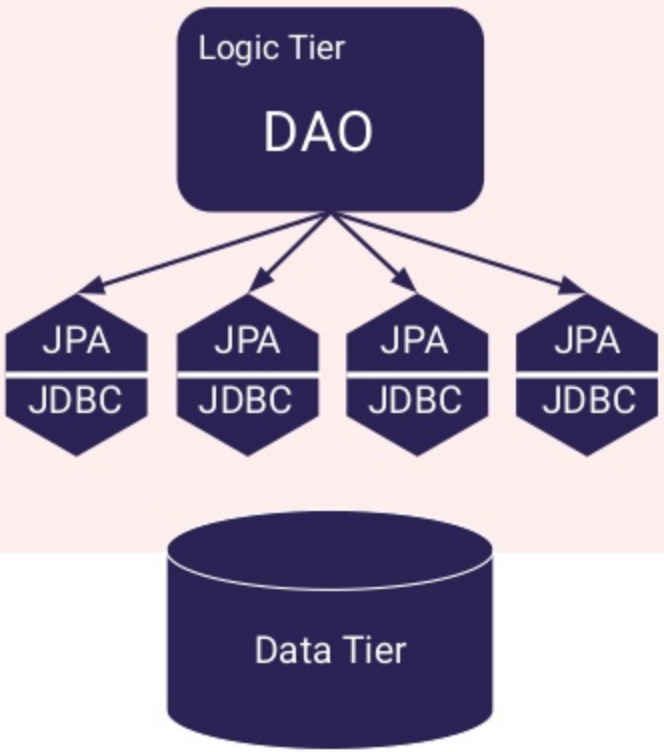
Graph

- Neo4J
- InfoGrid
- Sones
- HyperGraphDB

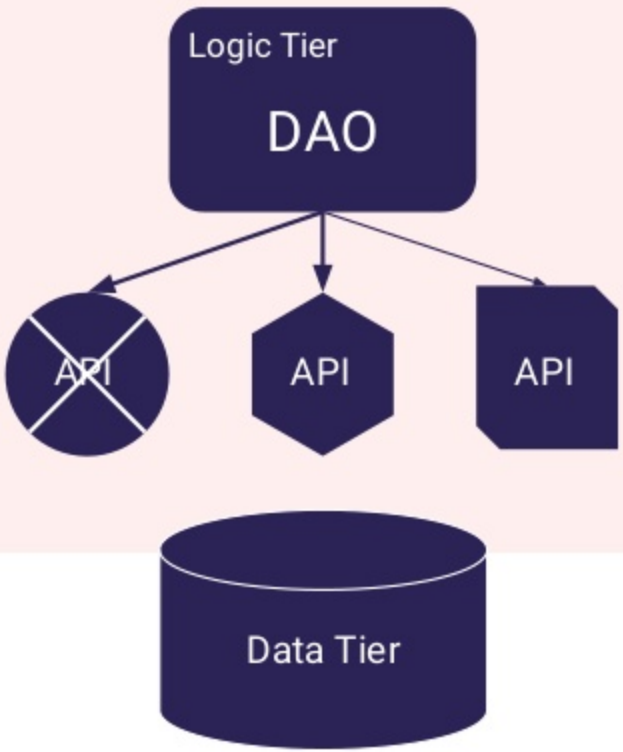


Applications

Relational Application



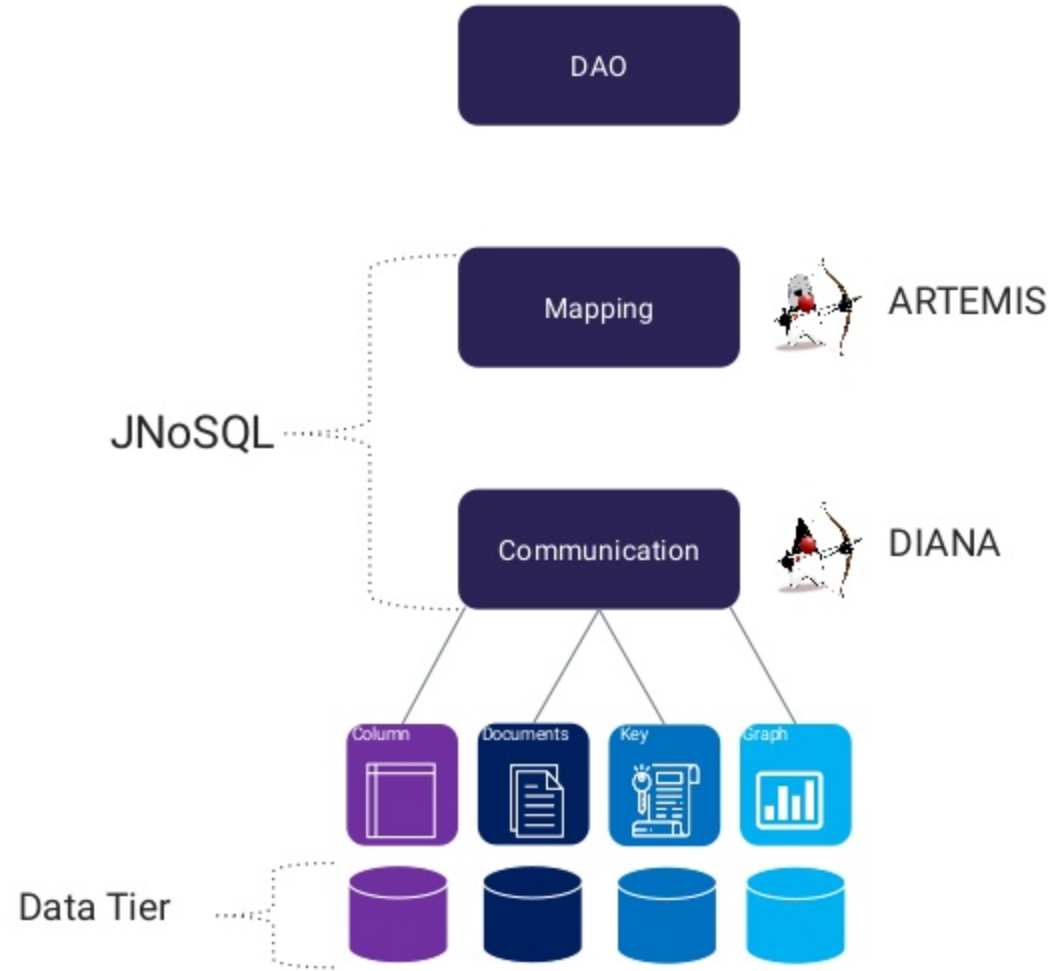
NoSQL Application



JPA problem for NoSQL

- Async Callback
- Time to Live (TTL)
- Consistency Level
- SQL based
- Diversity in NoSQL

Jakarta NoSQL



Communication Issue


```
BaseDocument baseDocument = new BaseDocument();
baseDocument.addAttribute(name, value);
```

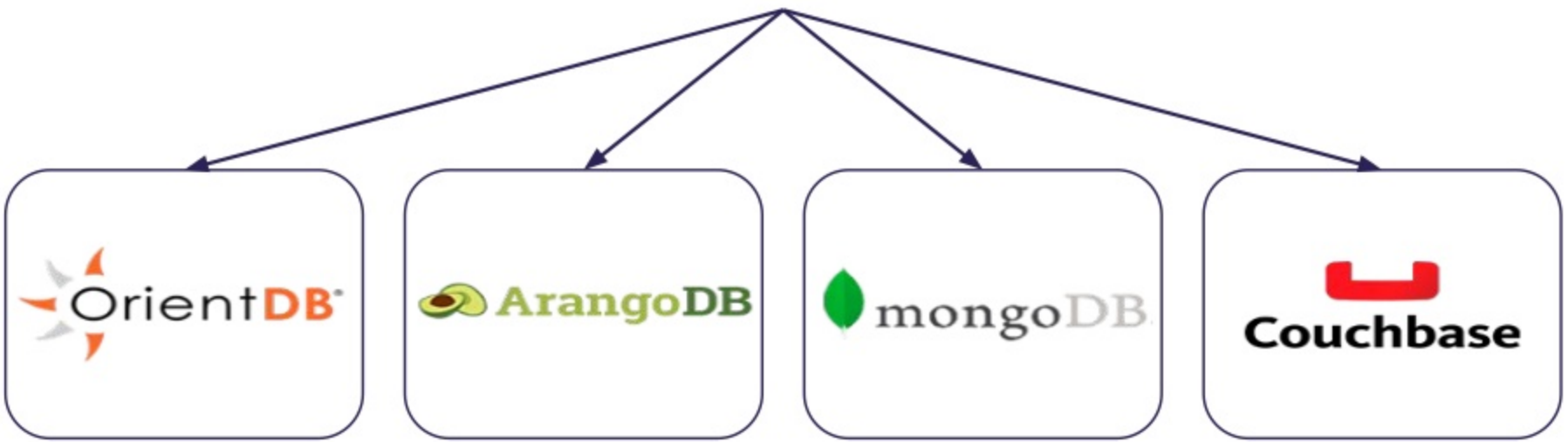

```
Document document = new Document();
document.append(name, value);
```


```
JsonObject jsonObject = JsonObject.create();
jsonObject.put(name, value);
```


```
ODocument document = new ODocument("collection");
document.field(name, value);
```

Communication

```
DocumentEntity entity = DocumentEntity.of("collection");
entity.add(name, value);
```



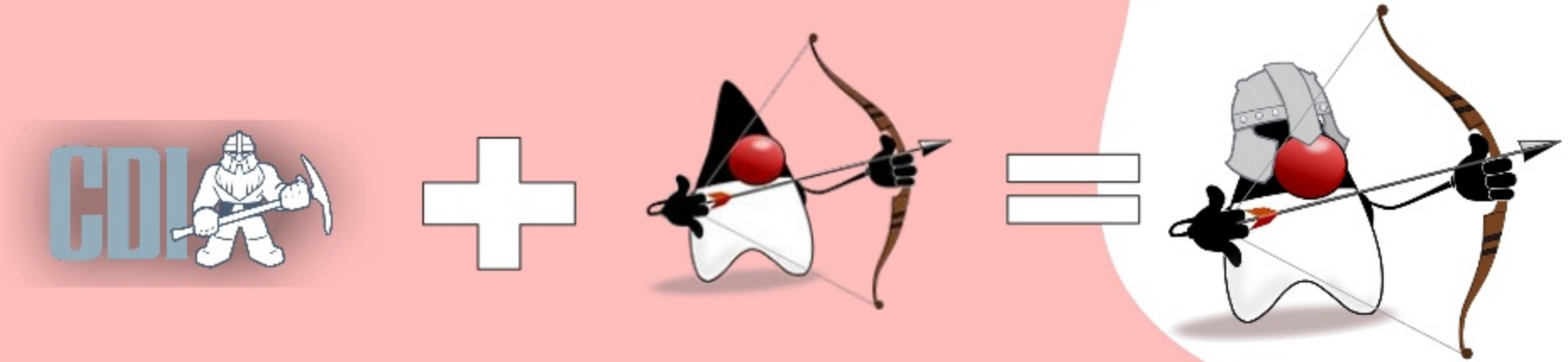
Communication

```
DocumentEntity entity = ...;
manager.insert(entity);
manager.update(entity);
List<DocumentEntity> entities = select().from("god")
        .where("power").eq("hunt").execute(manager);
delete().from("god").where("power").eq("hunt").execute(manager);
```

Diversity

```
List<ColumnEntity> entities = managerCassandra
.cql("select * from greece.good where id=10;");
managerCassandra.insert(entity, ConsistencyLevel.ALL);
```

Mapping



Annotations

- Mapped Superclass
- Id
- Entity
- Column

```
@Entity("god")
public class God {

    @Id
    private String id;

    @Column
    private String name;

    @Column
    private long age;

    @Column
    private Set<String> powers;
}
```

Templates

```
God artemis = ...;
DocumentTemplate template = …;
template.insert(artemis);
template.update(artemis);
DocumentQuery query = ...
List<God> gods = template.select(query);
```

Repository

```
interface GodRepository extends Repository<God, String> {

    Optional<God> findByName(String name);

    Stream<God> findAll();

}
```

Diversity

```
@Entity("god")
public class God {

@Column
private String name;

@UDT("weapon")
@Column
private Weapon weapon;
}
```

Diversity

```
interface GodRepository extends CassandraRepository<God, String> {

    @CQL("select * from God where name = ?")
    List<God> findByName(String name);

}
```

Queries

```
documentTemplate.query("select * from God where name = 'Diana'");
columnTemplate.query("select * from God where age = 25");
keyValueTemplate.query("get 'Diana'");
graphTemplate.query("g.V().hasLabel('God')");
```

Queries

```
PreparedStatement statement = template.prepare("select * from God
where name = @name");
preparedStatement.bind("name", "Diana");
List<God> adas = statement.getResultList();
```

Queries

```
interface GodRepository extends Repository<God, Long> {

    @Query("select * from God")
    List<God> findAll();

   @Query("select * from God where id = @id")
   Optional<God> findById(@Param("id") String id);

}
```

Providers



cassandra

SCYLLA.

elastic

ArangoDB

OrientDB

riak

ORACLE
NOSQL DATABASE

Couchbase

mongoDB

neo4j

redis

APACHE
HBASE

hazelcast

basho

Azure Cosmos DB

blazegraph

CouchDB
relax

GRAKN.AI

IBM Graph

Solr

JanusGraph

KeyLines

LINKURIOUS

RAVENDB

Stardog

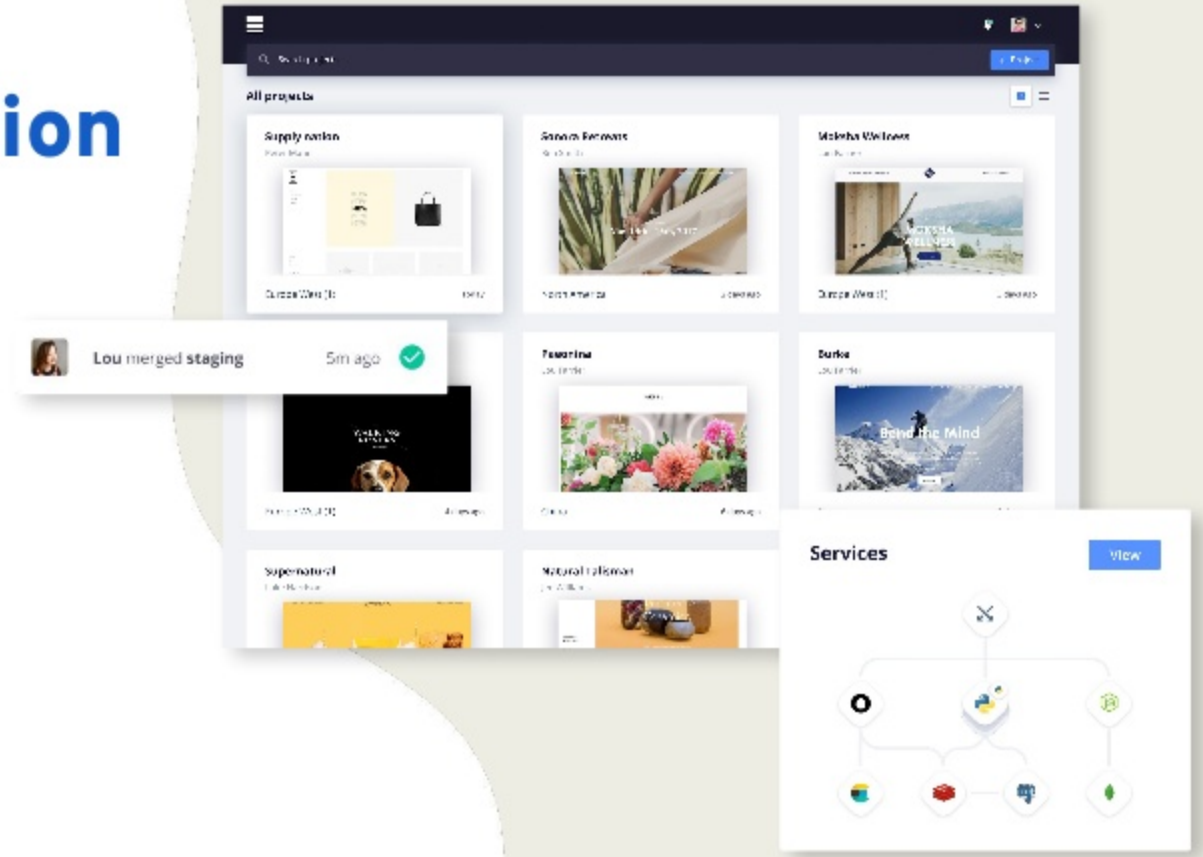
TITAN

Tom Sawyer
SOFTWARE

Infinispan


Demo Time

Platform.sh was built on the idea that your application comes first



It's what your customers care about.

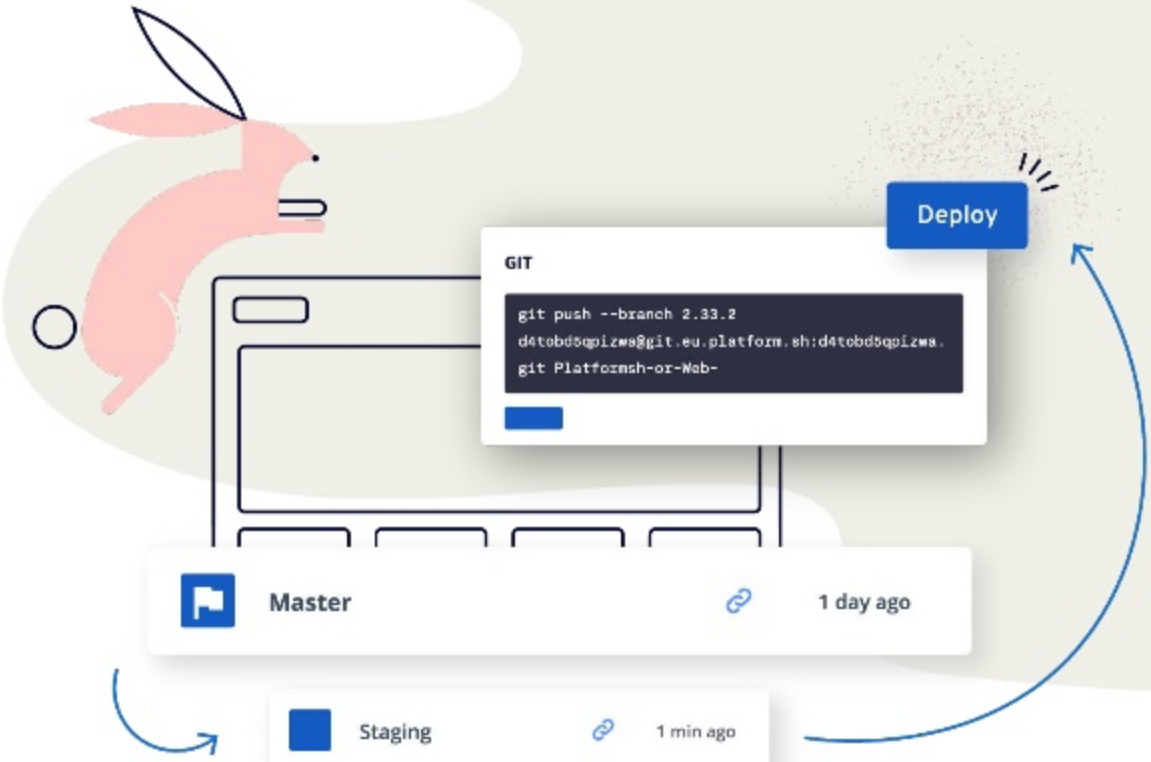
It's what drives you and your team.

We handle the rest.

Deploy anytime (even Fridays)

`git merge` packages your fully tested build, and our continuous deployment process sends those containers to production.

Because every change is being tested on a byte-for-byte clone of your production application, deployments become non-events.



Platform.sh at a glance

1,000s

of E-commerce, Life Sciences, Government, Education, Media and Entertainment, and High Tech customers

$47MM

invested by top-tier international partners

Johnson & Johnson

WITH PLATFORM.SH

Johnson & Johnson launches its brands into new markets worldwide with confidence and consistency.

WITH PLATFORM.SH

Arsenal doesn’t worry about innovating for their fans - even on game day.

INVIQA

WITH PLATFORM.SH

Inviqa focuses on client work that engages and impresses, with unprecedented speed.

unity

WITH PLATFORM.SH

Unity serves its community of innovators with the performance they expect.

Languages

Golang

Services

elastic

influxdb

mongoDB

MySQL

PostgreSQL

redis

Solr

RabbitMQ

VARNISH

CACHE

Java


spring


JAKARTA EE


MICROPROFILE
OPTIMIZING ENTERPRISE JAVA


Gradle


payara


THORNTAIL


maven


WildFly

Thank you

Jakarta EE Meets NoSQL in the Cloud Age

Otavio Santana
@otaviojava
@platformsh