# Client request: Java Application based on Kubernetes

January 29th, 2023

Emiliano Sturniolo
IriusRisk

# Customer needs

- HA and Stateful application
- Must have:
  - Load Balancer
  - Java backend
  - Relational Database
- Cloud architecture

# Customer concerns

What if:

- There is traffic spikes
- The app needs to be updated
- An update fails
- There is accidental data loss

# Proposal

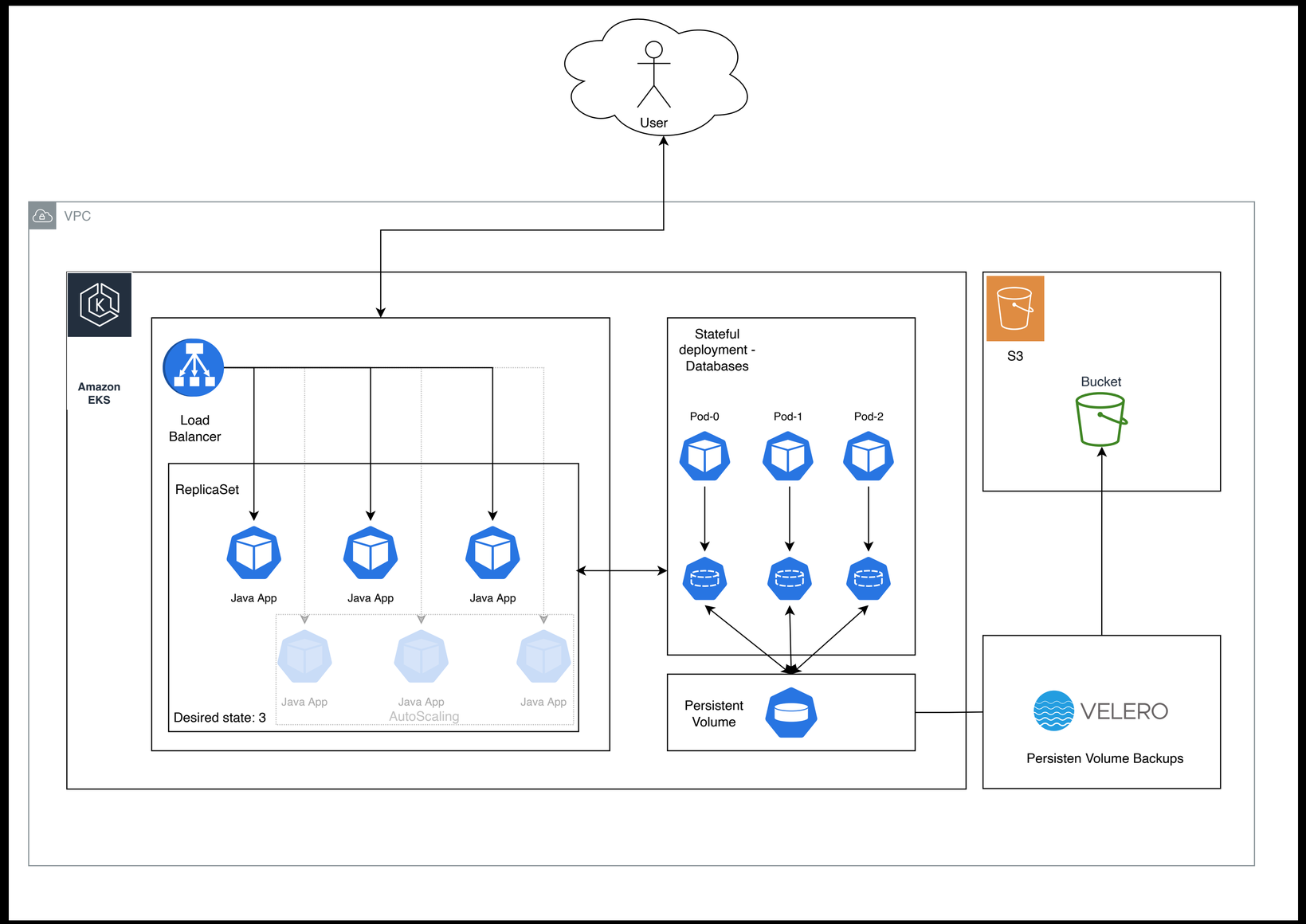
A cloud architecture based on AWS will be maybe the best solution.
Their all managed services make that this kind of situation be easy to implement and maintain.

For this scenario would be to use a EKS (AWS Kubernetes service) cluster with a Load Balancer, a set of Replica Sets for the Java app and a StatefulSet for the Database.

To add an extra layer of data security we can add the Open Source tool called “Velero” to create backups in a S3 bucket of the Persistent Volume that the database will use. This is a solution that AWS mentioned in their own blog.[1]

[1] Link to the AWS Blog https://aws.amazon.com/es/blogs/containers/backup-and-restore-your-amazon-eks-cluster-resources-using-velero/

# Diagram


User
VPC
Amazon EKS
Load Balancer
ReplicaSet
Java App
Java App
Java App
Java App
Java App
Java App
AutoScaling
Desired state: 3
Stateful deployment - Databases
Pod-0
Pod-1
Pod-2
Persistent Volume
S3
Bucket
VELERO
Persisten Volume Backups

Using this infrastructure the StatefulSet will be used to install a database as it is designed to provide ordered and consistent state programs and ensure that the data is not lost if the node fails.

In case of traffic spikes, the ReplicaSet will recreate as many new Replicas with the Java App as you defined in your “kubeconfig” file. Once the traffic get back to normal again, the extra Replicas will be destroyed until reach the desired state.

To update the application we recommend to use the “Rolling Updates” feature that allow to update the Replicas one at a time until update all of them. Doing this we avoid downtime in your app.

In case of update failure you can use the Kubernetes' “Rollback” feature. This will revert any changes made to it previous state in a quick and easy way.

Finally in a “Disaster Scenario” with the PV (Persistent Volumes) feature, you will be able to restore the data to a previous state. And using Velero you could be able to migrate the data from one cluster to another if you need it.

"IriusRisk" trademark and IriusRisk logo are owned by IriusRisk.