Zanibal
Cloud Business Solutions

ZANIBAL 8.0 RELEASE

Design considerations & architecture

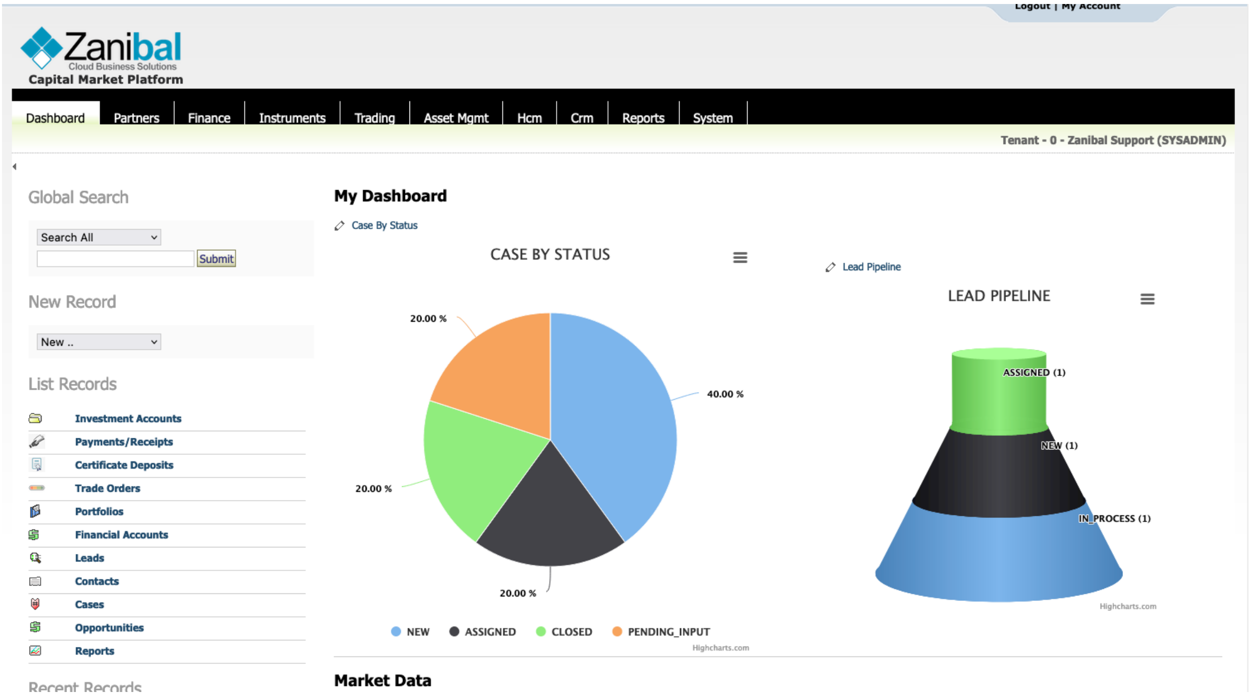
Abstract

This document provides a high-level overview of the 8.0 conceptual, logical and deployment design. It also discusses how Kubernetes is deployed for scheduling and managing the application workload.

Zanibal Engineering
engineering@zanibal.com

INTRODUCTION

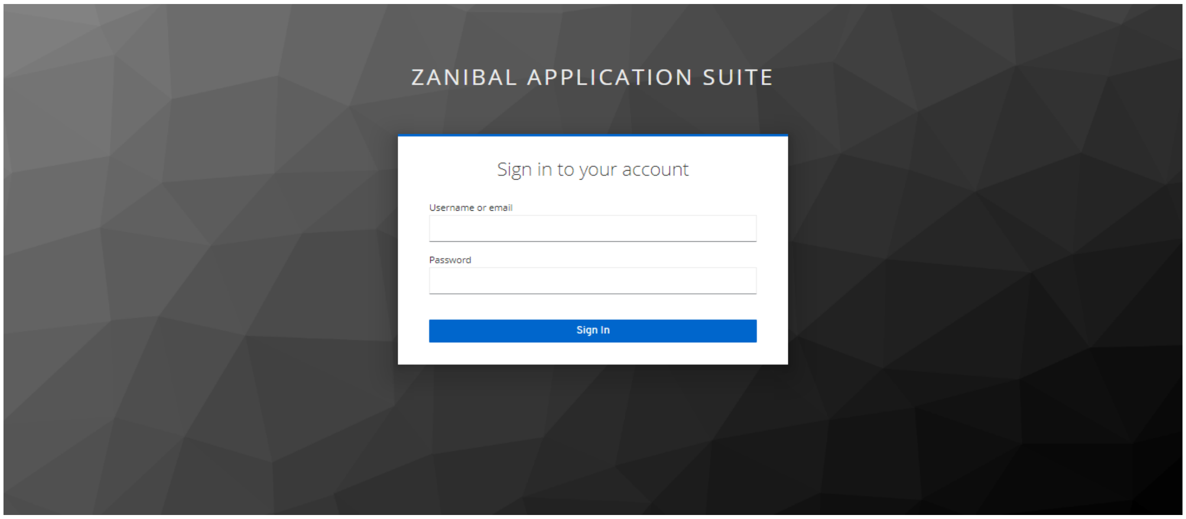
Zanibal Capital Market's Platform forms the backbone of investment firms and asset managers. Our clients depend on this platform and technology for Straight-Through Processing of transactions, accurate accounting, effective portfolio management and responsive client service. Back & middle office functions such as firm and client accounting, settlement and processing of cash transactions are all seamlessly integrated and automated. The solution ensures that there is a single source of truth for managing relationships, processing client transactions, analyzing and managing risk and understanding the financial position of the firm in real time.



The Zanibal 8.0 Platform was completely redesigned and built to deliver the following design objectives:

a. A microservices architecture where every component of the platform now runs in a separately managed runtime.
b. Real time analytics using application and traffic data to drive business, security and performance management.
c. Improved tracking and management of application security across all layers of the stack.
d. Significantly improved tools for third party integration and support for event driven / serverless end points.
e. Revamped back-office, client portal and mobile applications, rebuilt for better performance and improved security.
f. Support for intelligent customer engagement across all channels including Voice, Sms, WhatsApp, Email & Others from within the application

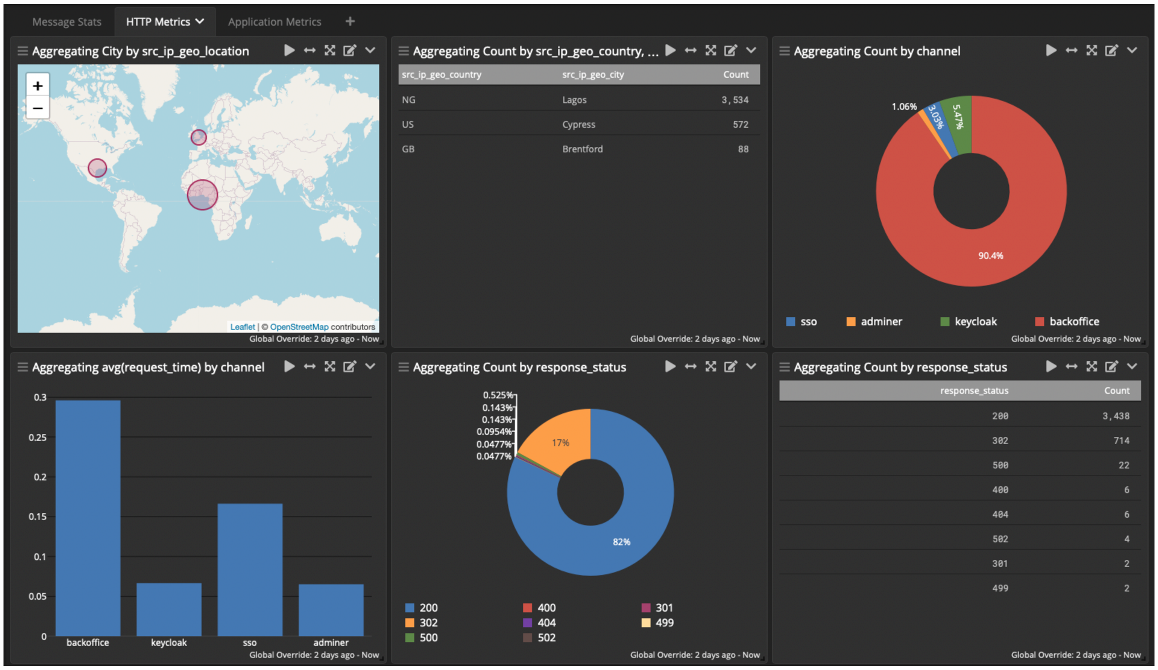
IDENDITY & ACCESS MANAGEMENT



The new identity and access management module handles authentication and authorization for all modules within the suite. This module provides a single-sign on/off feature for the suite as well as two-factor authentication (2FA) with support for most 2FA applications such as Google Authenticator. It can also be easily integrated with third party authentication modules such as Active Directory or Social Login providers such as Facebook, Google, etc.

OBSERVABILITY – LOG MANAGEMENT & METRICS

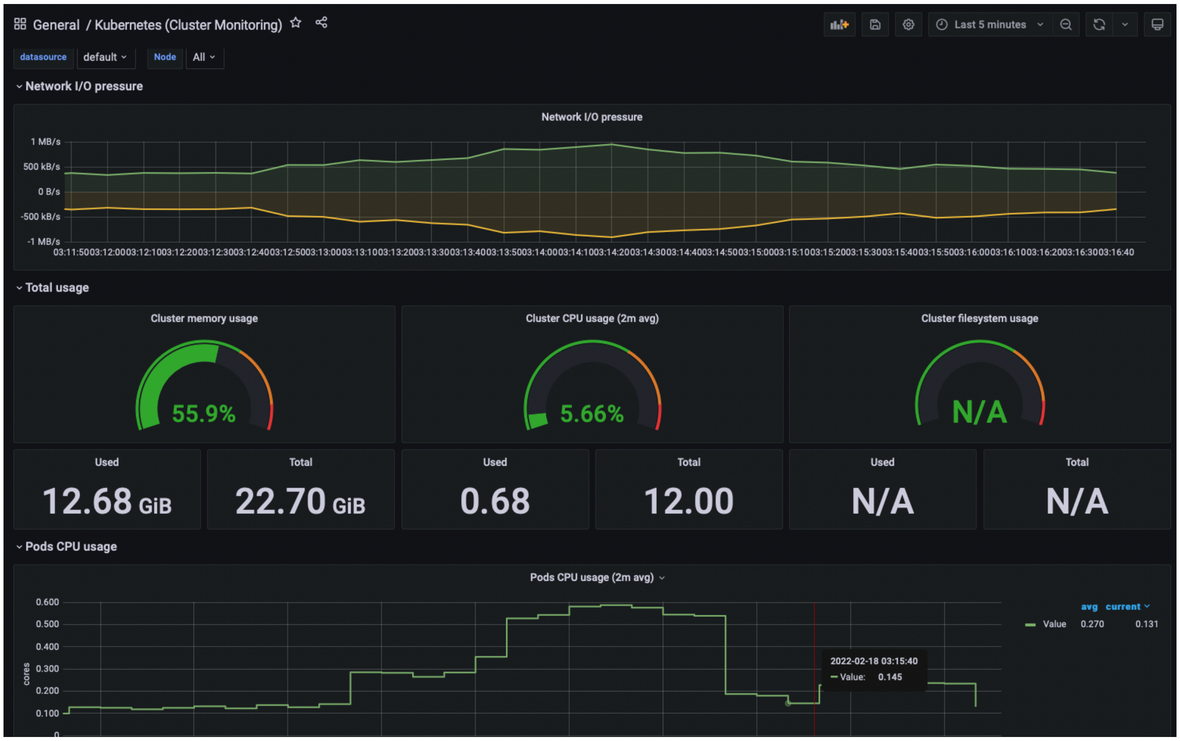
We provide tools for real time log analysis, management of threat intelligence and metrics. These tools enable real-time tracking of application activity such as access patterns, failed login attempts, response times, etc. The dashboards also provide real time visibility into the performance of the resources used to run our workload and send out real time alerts when defined thresholds are breached.


Message Stats
HTTP Metrics
Application Metrics
+
Aggregating City by src_ip_geo_location
+
−
Leaflet | © OpenStreetMap contributors
Global Override: 2 days ago - Now
Aggregating Count by src_ip_geo_country, ...
Global Override: 2 days ago - Now
Aggregating Count by channel
1.06%
3.03%
5.47%
90.4%
sso
adminer
keycloak
backoffice
Global Override: 2 days ago - Now
Aggregating avg(request_time) by channel
0.3
0.25
0.2
0.15
0.1
0.05
0
backoffice
keycloak
sso
adminer
Global Override: 2 days ago - Now
Aggregating Count by response_status
0.525%
0.143%
0.143%
0.0954%
0.0477%
0.0477%
17%
82%
200
302
500
400
404
502
301
499
Global Override: 2 days ago - Now
Aggregating Count by response_status
Global Override: 2 days ago - Now

src_ip_geo_country	src_ip_geo_city	Count
NG	Lagos	3,534
US	Cypress	572
GB	Brentford	88

response_status	Count
200	3,438
302	714
500	22
400	6
404	6
502	4
301	2
499	2

Combining these tools with a container orchestration platform ensures that our clients can provide a very reliable, secure and resilient application platform to all their stakeholders.

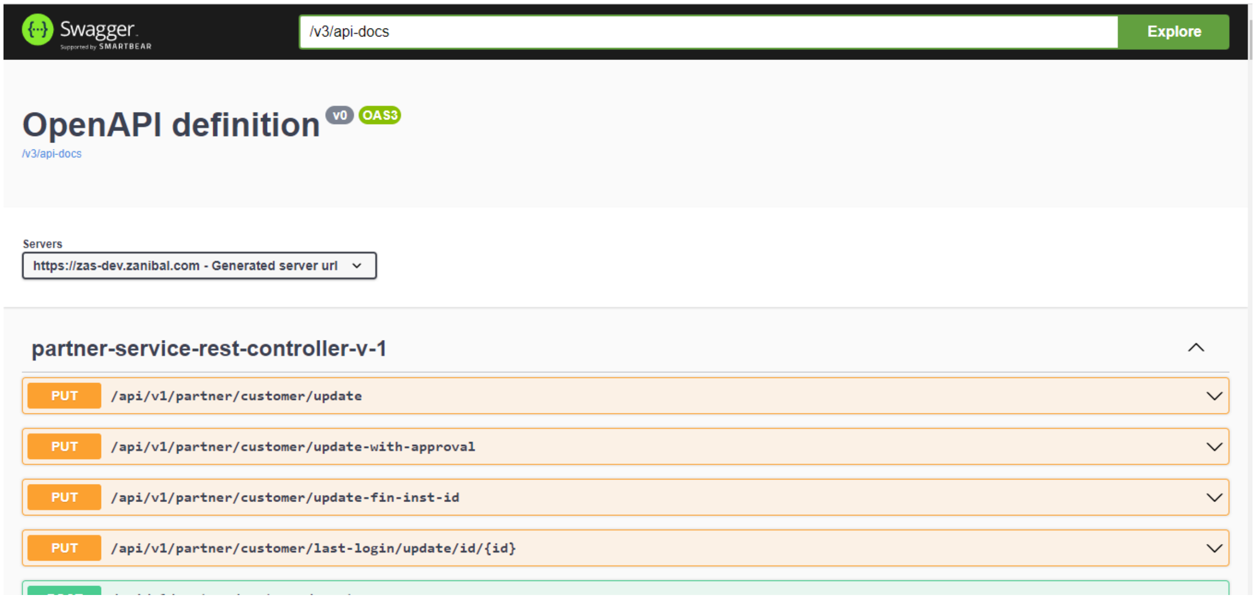


Used	Total	Used	Total	Used	Total
12.68 GiB	22.70 GiB	0.68	12.00	N/A	N/A

	avg	current
Value	0.270	0.131

THIRD PARTY INTEGRATION & WEB SERVICES

Zanibal has a very extensive set of APIs that exposes almost every component of the platform to third party channels. These APIs can be used for integration with various platforms such as web portals, trading venues, mobile applications amongst others and supports a range of protocols such as REST, FIX, SOAP, WEB Sockets and GPRC. The API management console is now bundled with the application and speeds up your integration process by providing you with an interactive console that can be used to test all available end points using your application data in context.



/v3/api-docs

OpenAPI definition

/v3/api-docs

Servers: https://zas-dev.zanibal.com - Generated server url

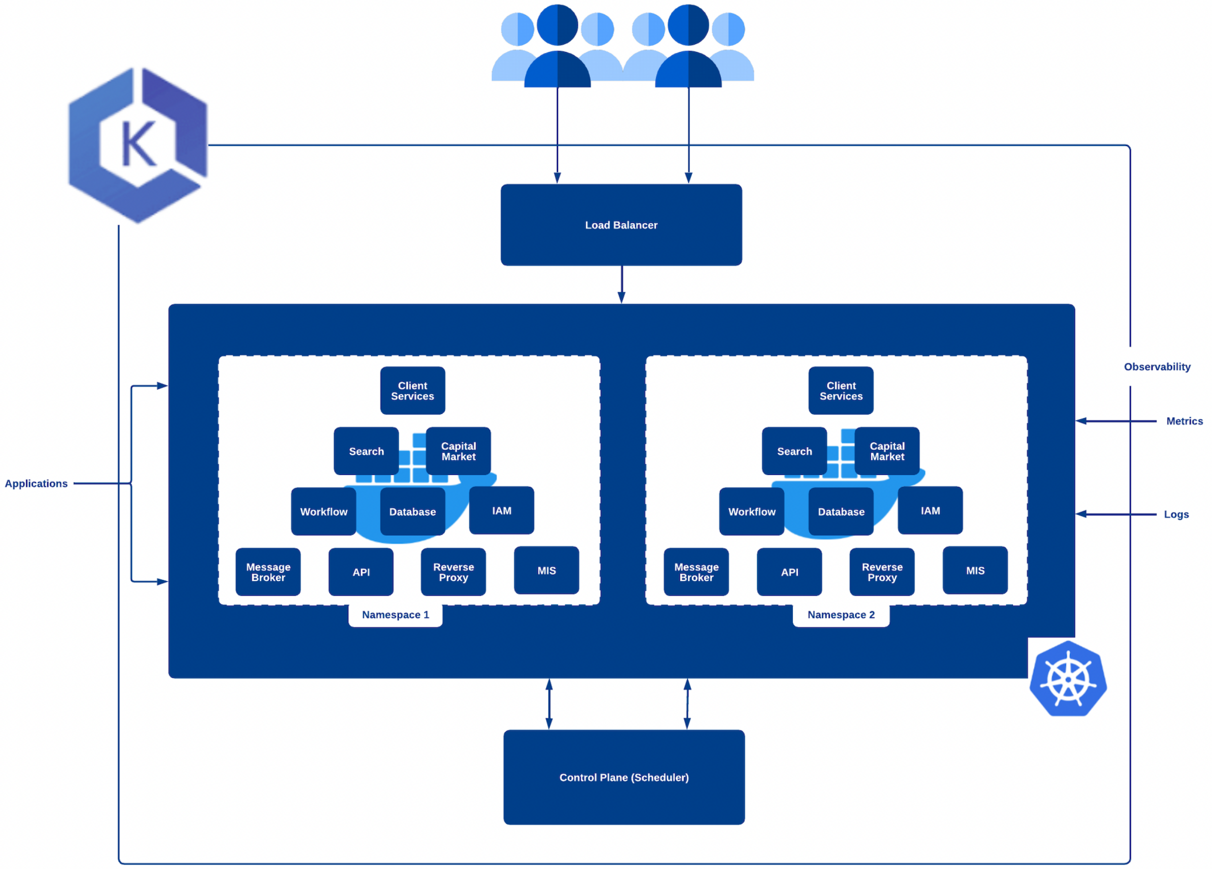
partner-service-rest-controller-v-1

```
PUT /api/v1/partner/customer/update
PUT /api/v1/partner/customer/update-with-approval
PUT /api/v1/partner/customer/update-fin-inst-id
PUT /api/v1/partner/customer/last-login/update/id/{id}
```

We have published a lot of complete and working code samples in python, node, java and curl for very common user stories such as account opening, cash processing, trading and client reporting to make it easier for people new to the platform to quickly build working applications.

ZANIBAL ON KUBERNETES

As a financial service provider, we know that your clients, their data, secure and timely access to market information, execution venues and accurate transaction processing is at the heart of your business. The ability to provide a platform that consistently delivers your business services and seamlessly scales with your business was a primary engineering objective for the Zanibal 8.0 release.



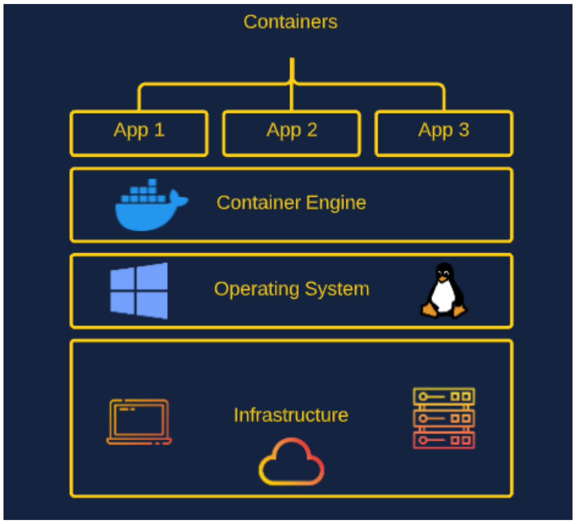
The Zanibal core applications, along with other required components are now deployed in docker containers which are orchestrated with Kubernetes. Kubernetes manages the workload, and its auto scaling and load balancing capabilities ensures that you can provide very accurate, scalable and resilient electronic services to all stakeholders.

Kubernetes now manages the underlying infrastructure resources for the Zanibal application such as the amount of compute, network, and storage resources required at different times of the day. Using Amazon’s Elastic Kubernetes Service (EKS), Zanibal in this new version of the application provides the power of Kubernetes to all our deployments.

CONTAINERS

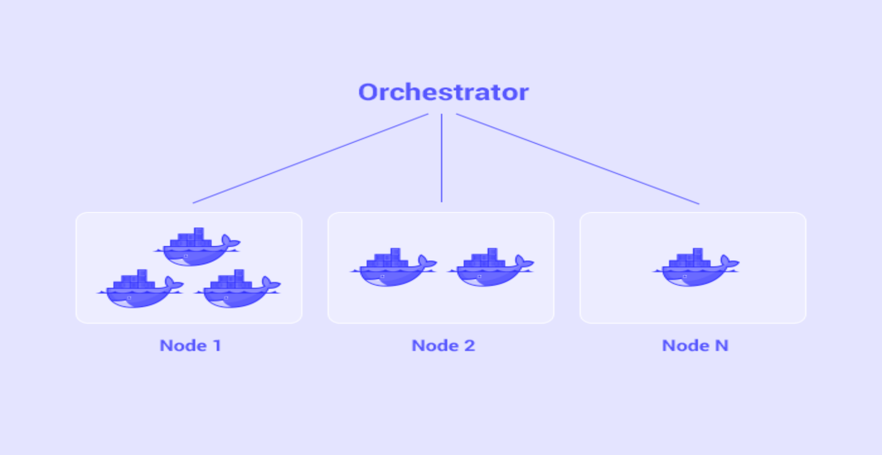
Containers are used for deploying and managing software on premise or in the cloud. Unlike virtual machines, containers virtualize the operating system instead of hardware and are more portable and efficient. Containers are used to abstract applications from the physical environment in which they are running and package all dependencies related to a software component and run them in an isolated environment.

Container platforms, Docker being the most common, are used to package applications so that they can access a specific set of resources on a physical or virtual host's operating system. This isolation and security allow many containers to run simultaneously on a given host. With application images, commonly executed using the Docker container runtime, applications deploy consistently in any environment, whether a public cloud, a private cloud, or a bare metal machine. Container images become containers at runtime and in the case of Docker containers - images become containers running within the Docker Engine.



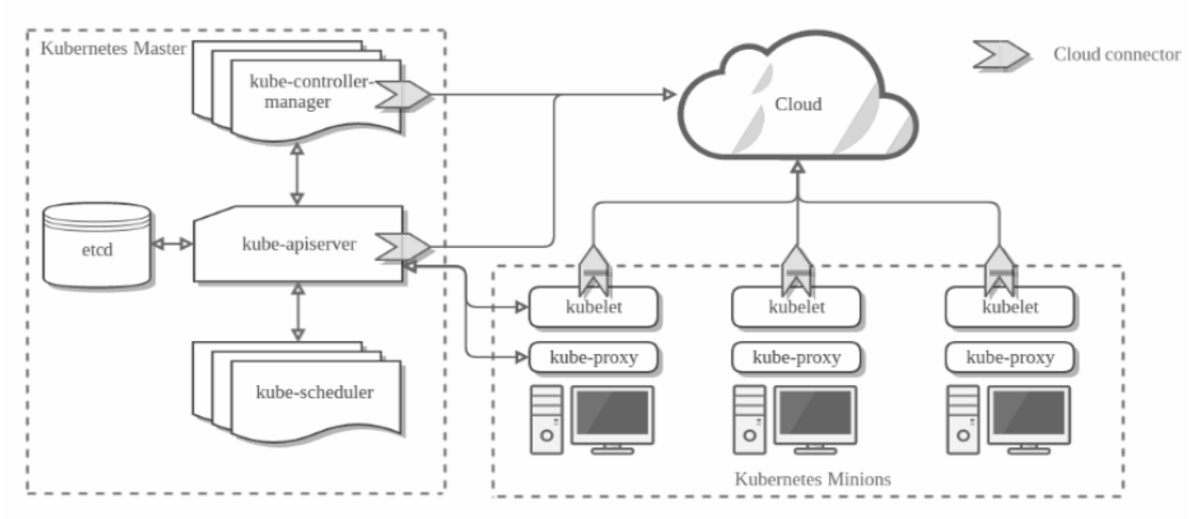
CONTAINER ORCHESTRATION

When operating at scale, container orchestration—automating the deployment, management, scaling, networking, and availability of your containers—becomes essential. Container orchestration is all about managing the life cycle of containers, especially in large, dynamic environments.



THE KUBERNETES PLATFORM

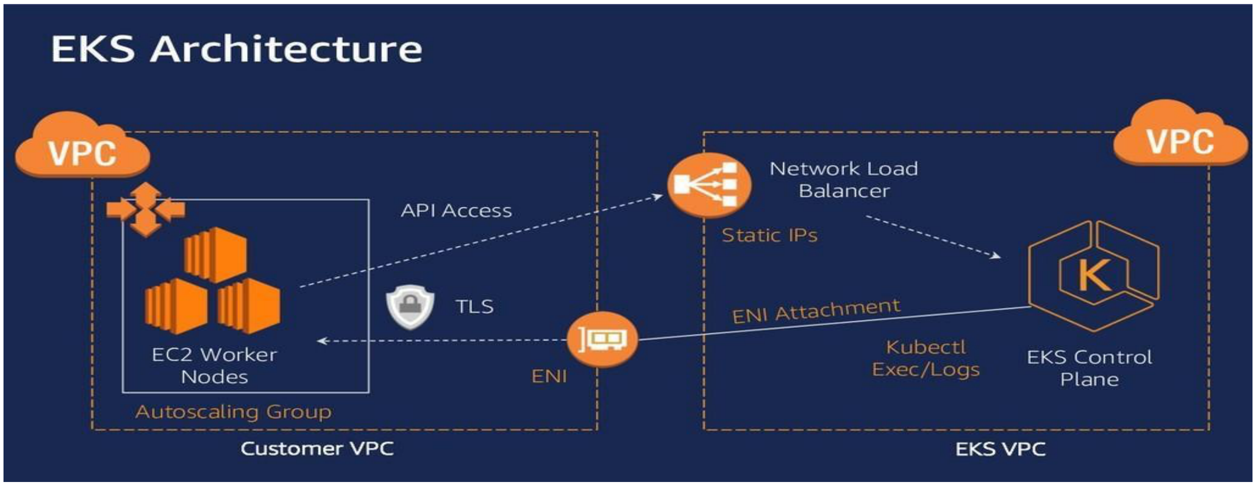
Kubernetes orchestrates the operation of multiple containers and manages the pods (containers) that run the applications to deliver the availability and performance objectives that are configured. It manages the operation of underlying the infrastructure resources for containerized applications such as the amount of compute, network, and storage resources required.



Orchestration tools like Kubernetes make it easier to automate and scale container-based deployments for large scale production workloads.

AMAZON ELASTIC KUBERNETES SERVICE (EKS)

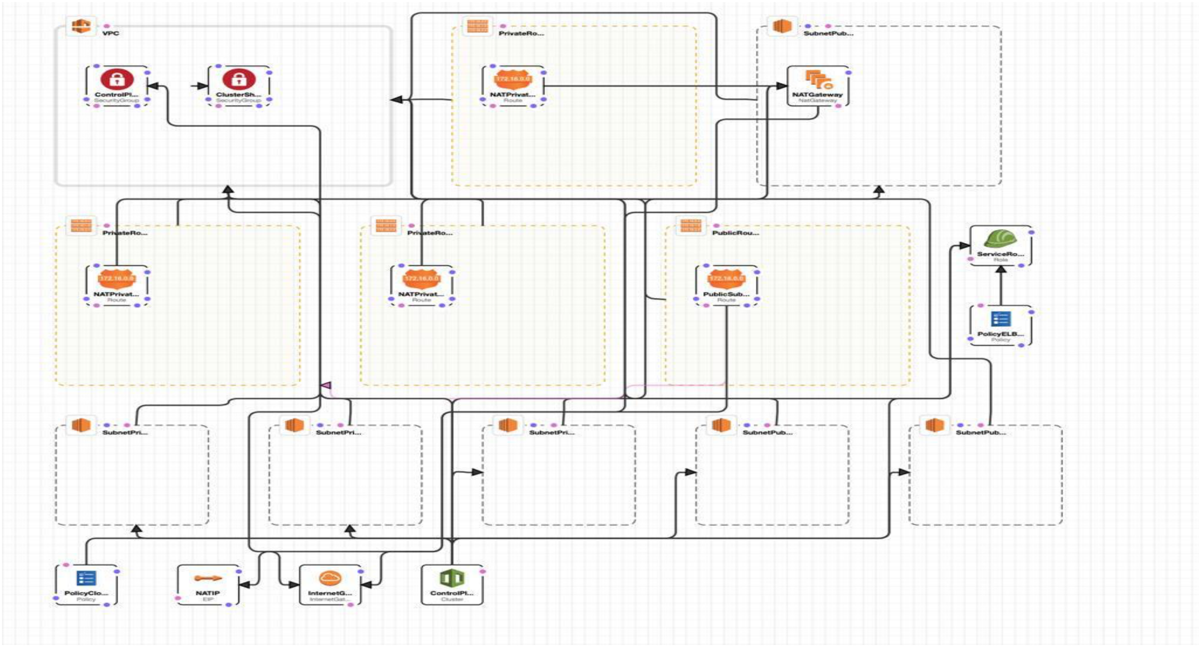
Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy to run Kubernetes on AWS and on-premises.



This platform provides a secure Kubernetes environment with security patches automatically applied to your cluster's control plane and integrates seamlessly with storage, network, compute and CI/CD services on the AWS platform.

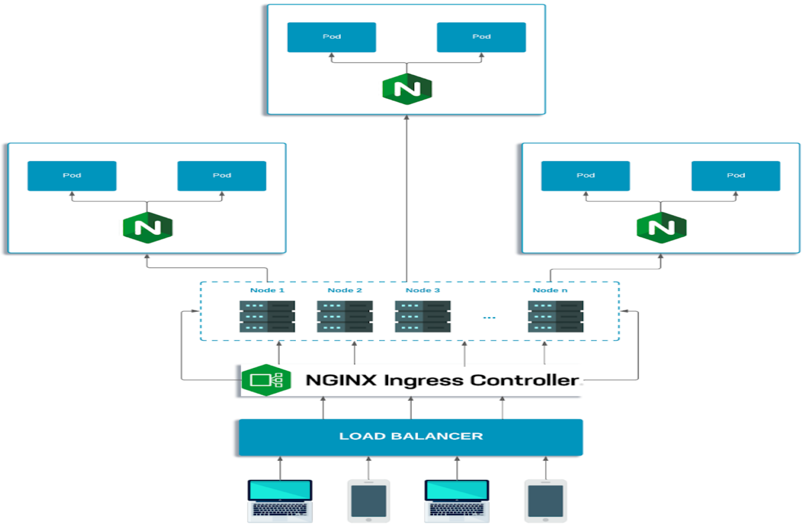
ZANIBAL'S EKS CLUSTER NETWORK DESIGN

The design of the cluster's network ensures that all ingress and egress traffic is securely routed and monitored. The network traffic logs are analyzed in real-time and archived for 30 days.



WORKLOAD ISOLATION & INGRESS CONTROL

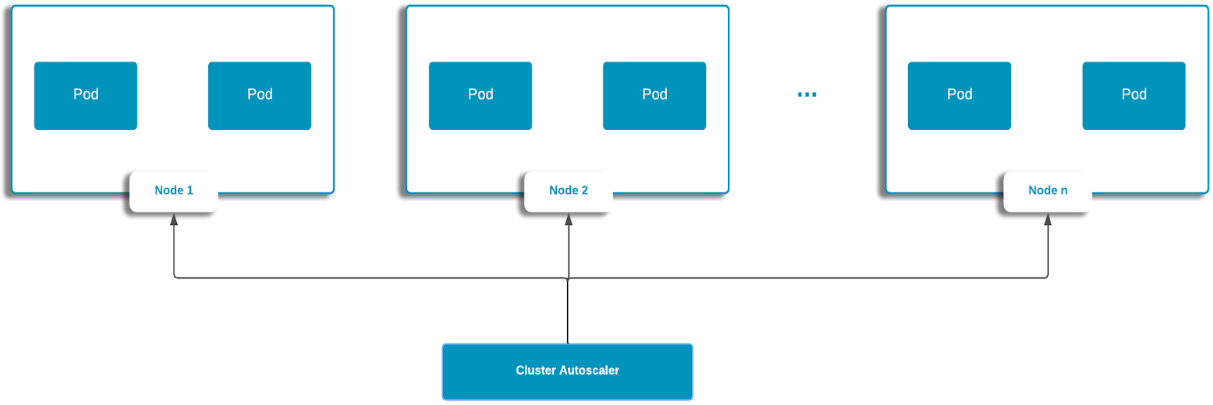
Given the complexity of the application architecture and the need to ensure consistent isolation of all the workloads running in different namespaces, an ingress controller is used to route traffic to the appropriate namespace resources, while also automatically managing all external DNS records for the cluster.



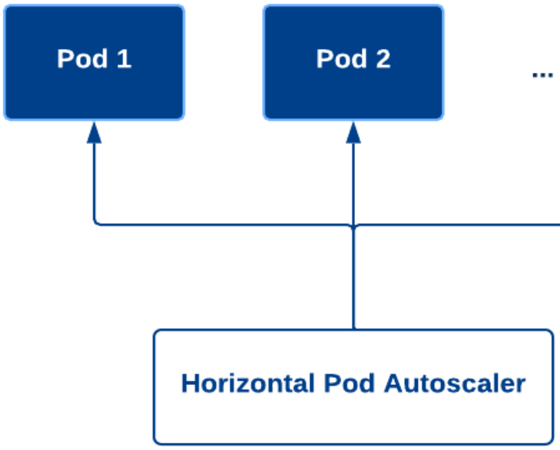
AUTO SCALING

Demand, efficiency and several other metrics can lead to a need to scale the application. Kubernetes provides several tools for this purpose that allows for automated scaling of the Zanibal application and infrastructure. The Cluster auto scaler (CAS), which works on the infrastructure layer, and the Horizontal pod auto scaler (HPA) which scales pods based on traffic patterns are used in the design for capacity management.

The cluster auto scaler increases or decreases the size of a Kubernetes cluster (by adding or removing nodes), based on the presence of pending pods and node utilization metrics. It adds nodes to the cluster whenever it detects pending pods that could not be scheduled due to resource shortages and removes nodes from a cluster, whenever the utilization of the pods falls below a certain threshold defined by our administrators. It ensures that the underlying cluster infrastructure is elastic and scalable and can meet the changing demands of the workloads of our clients.

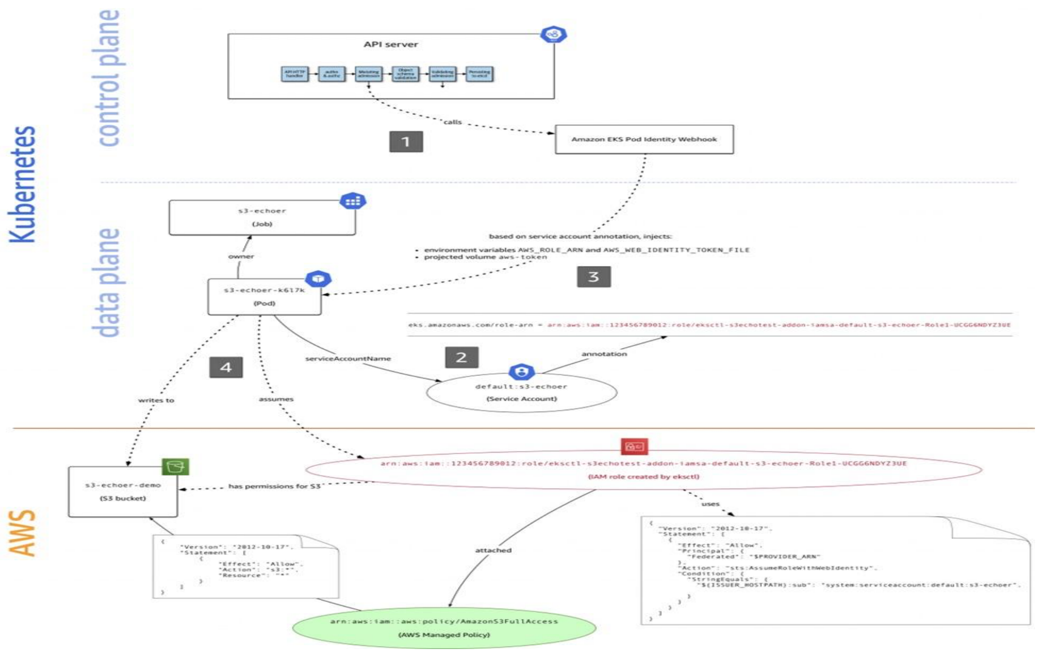


HPA scales the number of pods in a replica set based on CPU utilization, custom or external metrics. It ensures that critical applications are elastic and can scale out to meet increasing demand as well as scale down to ensure optimal resource usage.



SECURITY

We leverage EKS & EC2 resources to provide security for our Kubernetes clusters. For example, IAM provides fine-grained policy-based access control that is used to manage the service accounts used by the containers and the VPC network design and security policies provides the required network security for application data and other resources.


Kubernetes
control plane
API server
API HTTP handler
authn & authz
Mutating admission
Object schema validation
Validating admission
Persisting to etcd
calls
1
Amazon EKS Pod Identity Webhook
data plane
s3-echoer
(Job)
owner
s3-echoer-k617k
(Pod)
based on service account annotation, injects:
• environment variables AWS_ROLE_ARN and AWS_WEB_IDENTITY_TOKEN_FILE
• projected volume aws-token
3
eks.amazonaws.com/role-arn = arn:aws:iam::123456789012:role/eksctl-s3echotest-addon-iamsa-default-s3-echoer-Role1-UCGG6NDYZ3UE
serviceAccountName
2
annotation
default:s3-echoer
(Service Account)
4
writes to
assumes
AWS
arn:aws:iam::123456789012:role/eksctl-s3echotest-addon-iamsa-default-s3-echoer-Role1-UCGG6NDYZ3UE
(IAM role created by eksctl)
s3-echoer-demo
(S3 bucket)
has permissions for S3
uses
attached
arn:aws:iam::aws:policy/AmazonS3FullAccess
(AWS Managed Policy)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        }
    ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "$PROVIDER_ARN"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${ISSUER_HOSTPATH}:sub": "system:serviceaccount:default:s3-echoer",
        }
      }
    }
  ]
}
```

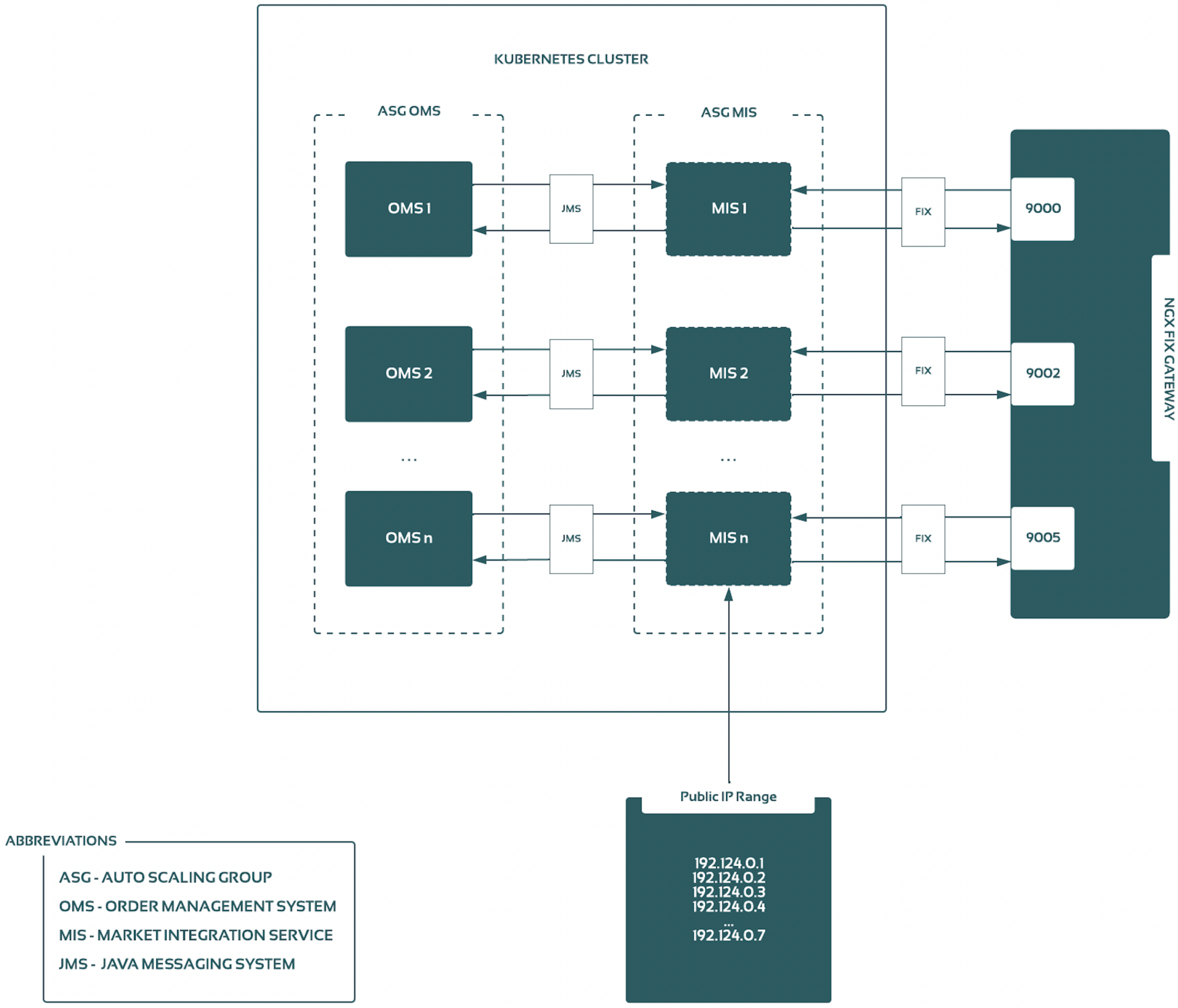
AWS Security Groups for pods is also another measure that has been deployed to control inbound and outbound network traffic within the cluster.

In addition, the Kubernetes cluster health is monitored in order to have real-time and accurate data on the environment's health and detect any form of unauthorized activity within the pods. Some of the metrics tracked are:

- Resource Metrics – This is used to monitor the utilization of resources in comparison to the workload. It checks the usage and capacity of pods, underlying EC2 instances, and containers to ensure that different layers of the cluster, including nodes and pods running on them, can run the workload or accommodate additional workload.
- The state of Kubernetes objects – This checks the health status and availability of the current objects such as nodes and pods within the cluster's control plane. (The control plane allows the admin to have an overall picture of the performance and throughput of requests made within the clusters.) This metric helps in identifying cluster-related problems that might require human intervention.
- Pod network activity and traffic patterns measured against baseline metrics.

INTEGRATION WITH EXECUTION VENUES

Zanibal supports integration with execution venues (exchanges) for the routing and processing of orders. Some of these venues include NGX (Nigeria), NSE (Kenya), USE (Uganda), and others. At the network layer, these markets use a range of network security policies ranging from CIDR filters to VPNs to control access to their trading engines and the trading platform subsequently verifies a firm or trader credentials using FIX, JWT Tokens, etc. To support venues that filter traffic based on whitelisted IP addresses, we deploy all the Market Integration Services (MIS) on nodes (servers) from a dedicated Auto Scaling Group (ASG) that uses a pool of reserved public IP addresses. This will make it easier for the venues to configure and grant access to a known list of IP addresses while also supporting the ability for the cluster to optimize the scheduling of pods for the MIS based on metrics such as Memory or CPU pressure as well a complete node or failure zone outages.



ABBREVIATIONS

ASG - AUTO SCALING GROUP

OMS - ORDER MANAGEMENT SYSTEM

MIS - MARKET INTEGRATION SERVICE

JMS - JAVA MESSAGING SYSTEM