INT 6940
Experiential Network Projects

Iteration 6

Report By

KnowQuest (Team A)

Professor: Amin Karimpour

Date: 12/07/2023

Project Name: KnowQuest - Team A

Group Member:

Name	**Email**
Ananya Kakani	kakani.a@northeastern.edu
Jie Zhu	zhu.jie2@northeastern.edu
Jui Ying Lee	lee.juiy@northeastern.edu
Sandeep Samsani	samsani.s@northeastern.edu
Vivek Vaghasiya	vaghasiya.vi@northeastern.edu
Yi Zou	zou.yi2@northeastern.edu

Executive Summary

The KnowQuest Platform and AWS Cloud Improvement project underwent a comprehensive transformation to enhance the efficiency and reliability of the KnowQuest operations. The initiative included the design and implementation of a new cloud architecture, migrating the applications onto a new cloud PaaS model, and AWS infrastructure to the new setup. The new AWS infrastructure incorporated various services, including S3, EC2, RDS, CloudFormation, and OpenSearch. The meticulous steps taken to achieve this configuration included creating an IAM role for the EKS cluster, establishing a dedicated VPC for the EKS cluster, creating the EKS cluster itself, installing and setting up IAM Authenticator and Kubectl Utility, creating an IAM role for EKS worker nodes, and deploying worker nodes via node groups.

Following the foundational setup, the project progressed to deploying a demo application and initiating CloudWatch Data Visualization for KnowQuest. The CloudWatch process involved creating a prototype for a data flow within, illustrating how data moves through its components for visualization and analysis. The simplified prototype outlined the objective, key components, and the data flow. Data ingestion occurred from various sources, passing through Fluentd for parsing and transformation if used, then stored and indexed in Dashboard. Users interacted with CloudWatch dashboards for data analysis, visualization, and gaining insights to inform decisions, troubleshoot issues, or monitor system performance.

Additionally, performance testing was integral to the project's success, with the execution of a performance testing plan to meet all performance requirements. This holistic approach ensured that the KnowQuest website's operations and AWS infrastructure not only underwent significant improvements but also adhered to high-performance standards, aligning with KnowQuest's commitment to excellence.

Introduction

In addressing the multifaceted challenge at hand, our project delves into a comprehensive exploration within the domain of Informatics, aiming to provide innovative solutions that align with the stakeholders' evolving requirements. The focal point of our initiative lies in clarifying and deciphering these dynamic needs through a meticulous approach. To navigate this process, our team strategically selected a robust technical stack, integrating cutting-edge tools to craft tailored solutions. Leveraging advanced tools such as AWS services, i.e., EKS, and CloudWatch, we aim to streamline data flow and visualization, ensuring a seamless integration of our backend architecture. The significance of our chosen technical stack lies not only in its contemporary relevance but also in its capacity to empower our team in addressing the intricacies of the problem at hand.

Central to our approach was the interrogation of diverse datasets, each posing unique challenges and opportunities. To ascertain stakeholder requirements, we conducted in-depth explorations and experiments, meticulously identifying and recommending specific solutions tailored to meet these needs. Throughout these experiments, our team encountered challenges that fostered innovation and problem-solving, shaping a resilient project trajectory. The dynamic interplay of technology and methodology in our experiments not only underscored the complexity of the project but also highlighted its relevance to professionals in the Informatics discipline.

What renders our project particularly compelling is not just its technical intricacy, but its broader implications and resonance within the field of Informatics. The exploration of cutting-edge tools, data sets, and techniques contributes not only to the specific problem at hand but also to the broader discourse within the discipline. The dynamic interplay of stakeholder requirements, technical solutions, and experimental methodologies encapsulates the essence of our project, making it both interesting and relevant for professionals seeking to navigate the evolving landscape of Informatics.

Implementation

To effectively implement the outlined sitemap for the KnowQuest project, a structured approach was taken to fulfill specific needs, and this process involved careful planning and rationale. In order to establish the AWS and Kubernetes infrastructure, we first identified key prerequisites, necessitating an AWS account, AWS CLI, and an IAM user. The choice of AWS as the cloud provider was driven by its industry-standard reliability and scalability. The AWS infrastructure setup, including VPCs, subnets, EC2 instances, and associated security measures, was executed meticulously to ensure a secure and well-connected environment.

The Kubernetes installation phase involved using tools such as SSH for instance access and Docker for containerization. Kubeadm, Kubectl, and kubelet were selected for their reliability and compatibility, facilitating the seamless initialization of the master node and configuration of worker nodes. The logic behind this analysis was rooted in the need for a robust and easily manageable Kubernetes cluster.

The creation of a pipeline utilizing YAML files to define various stages of deployment. This was done to automate and streamline the deployment process, ensuring consistency and reproducibility in the development pipeline. The DevOps integration and testing phases employed CI/CD practices to enhance efficiency and reliability, aligning with industry best practices.

Monitoring, security, and scaling considerations were addressed comprehensively. The analysis logic included the need for backup strategies, adherence to security best practices (such as RBAC and network policies), and the implementation of logging and monitoring using CloudWatch. The tools were selected based on their industry recognition and capabilities in providing robust monitoring solutions. High availability was achieved through HA control plane configurations, and auto-scaling mechanisms were implemented to ensure dynamic

resource allocation.

In the networking and services domain, the logic dictated the need for proper Kubernetes networking configuration, cluster add-ons like the Kubernetes Dashboard for enhanced visibility, load balancing for optimal resource distribution, DNS configuration using CoreDNS for efficient name resolution, and optionally, an Ingress Controller for advanced routing. The tools chosen for these tasks were selected based on their proven track record and compatibility within the Kubernetes ecosystem.

Overall, the KnowQuest project's sitemap was meticulously designed and executed to meet specific project requirements. The logic behind the analysis was driven by the need for a scalable, secure, and efficient AWS and Kubernetes infrastructure. The tools were selected based on their industry recognition, capabilities, and compatibility, ensuring a comprehensive and robust deployment tailored to the project's objectives.

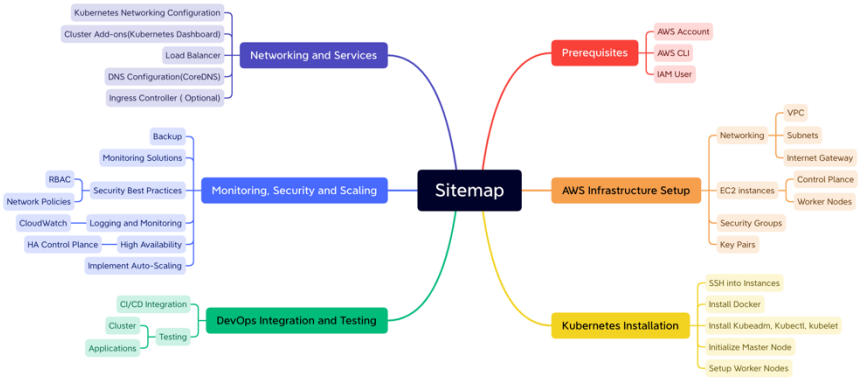


Image1: KnowQuest's Development Sitemap

This project is important because it will improve the user experience of the KnowQuest website and make it more reliable and efficient. This will allow KnowQuest to focus on providing its users with the best possible experience.

The following acceptance criteria must be met for the KnowQuest existing platform and AWS Improvement project to be considered successful:

Architecture

- The new public cloud architecture must be able to handle a 20% increase in traffic without any performance degradation.
- The new architecture must be able to deploy new features in less than 24 hours.
- The new PaaS architecture must be able to scale to support 1 million active users.

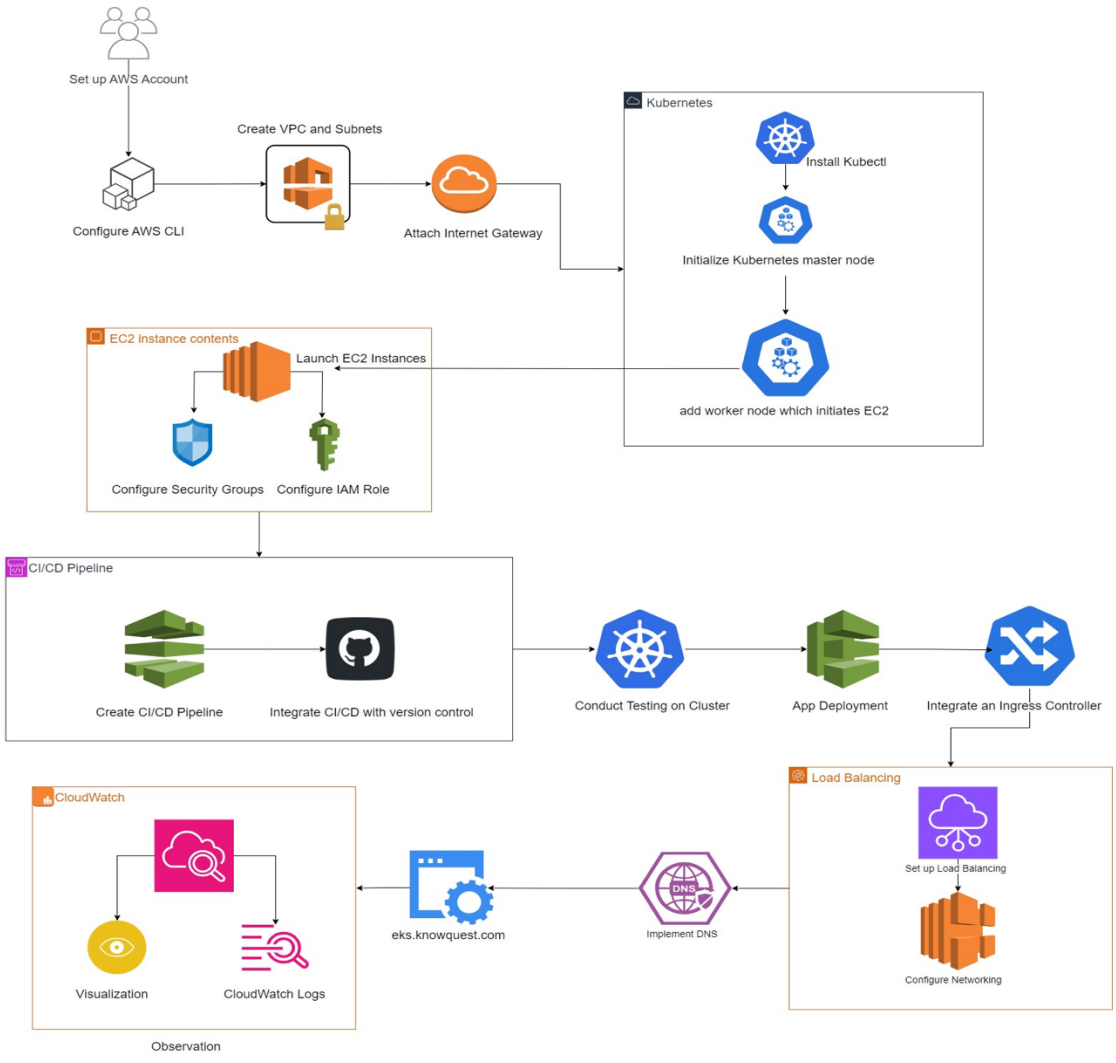


Image2: Kubernetes Architecture

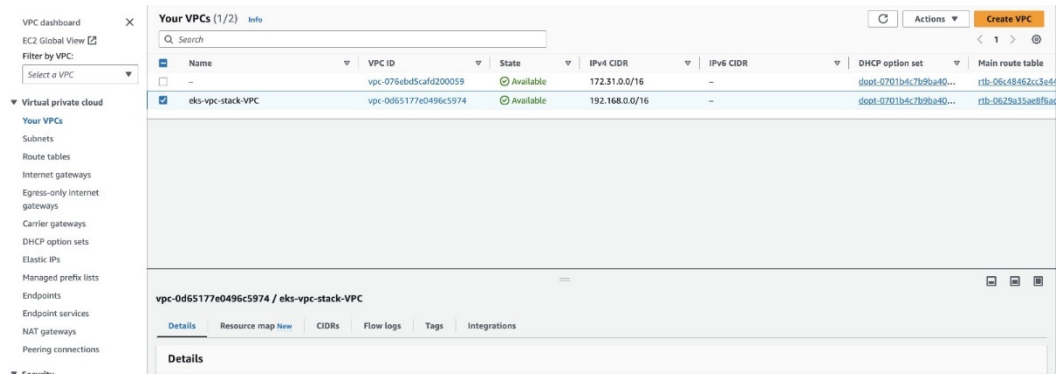
AWS Migration for KnowQuest:

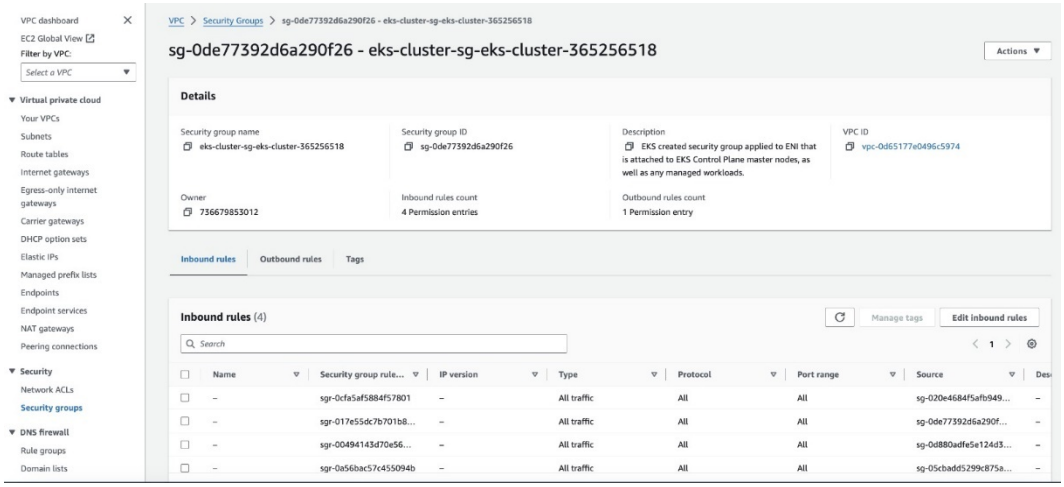
The KnowQuest Backend and AWS Improvement project will improve the efficiency and reliability of the KnowQuest website. The project will involve designing and implementing a new backend architecture, migrating the existing backend to the new architecture, designing, and implementing a new AWS infrastructure, and migrating the existing AWS infrastructure to the new infrastructure. To name a few, these are the most common AWS services utilized:

- S3: To store yaml configuration and deployment files and automatic backups.
- EC2: To host the node clusters of eks applications.
- RDS: To manages the database with full administration support, simplifying setup, patching, and backups for enhanced scalability and reliability.
- CloudFormation: To manage the provisioning and deployment of AWS resources automatically.
- Lambda: Executes event-driven serverless code, providing flexibility and efficiency without the need for managing server infrastructure, enabling dynamic and cost-effective scaling based on demand.
- OpenSearch: To provide a search and analytics engine for the website and etc.,

Steps taken to achieve the configuration:

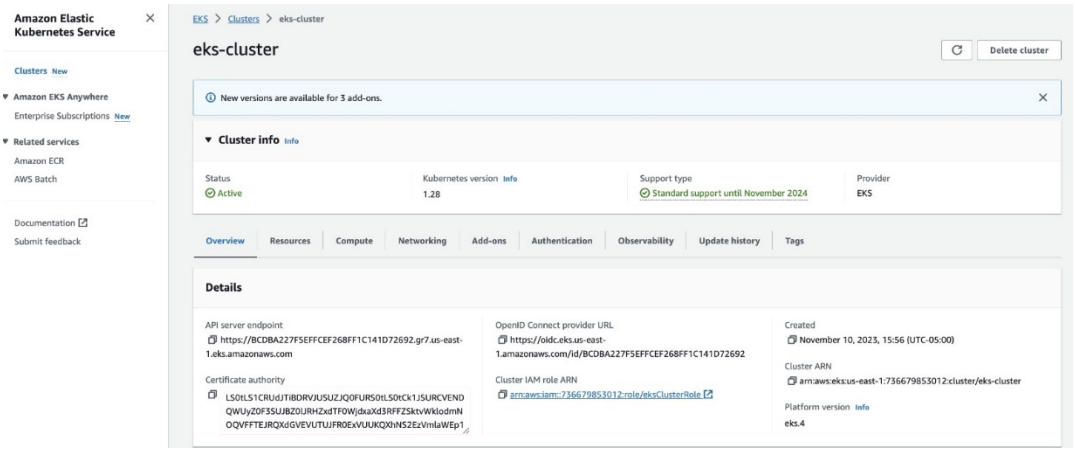
- **Step 1: Create IAM role for EKS Cluster**
    - The sponsors have given an IAM role as a prerequisite for maintaining the permissions on EKS service. They have given us an IAM role which includes EKS cluster policies.
- **Step 2: Create Dedicated VPC for the EKS Cluster**
    - As master node should be in a network VPC. So, created a VPC and its subnets using cloud formation.





- **Step 3: Create EKS Cluster**

  - In this step created an EKS cluster which is a master node and connected it with the VPC created in the above step.



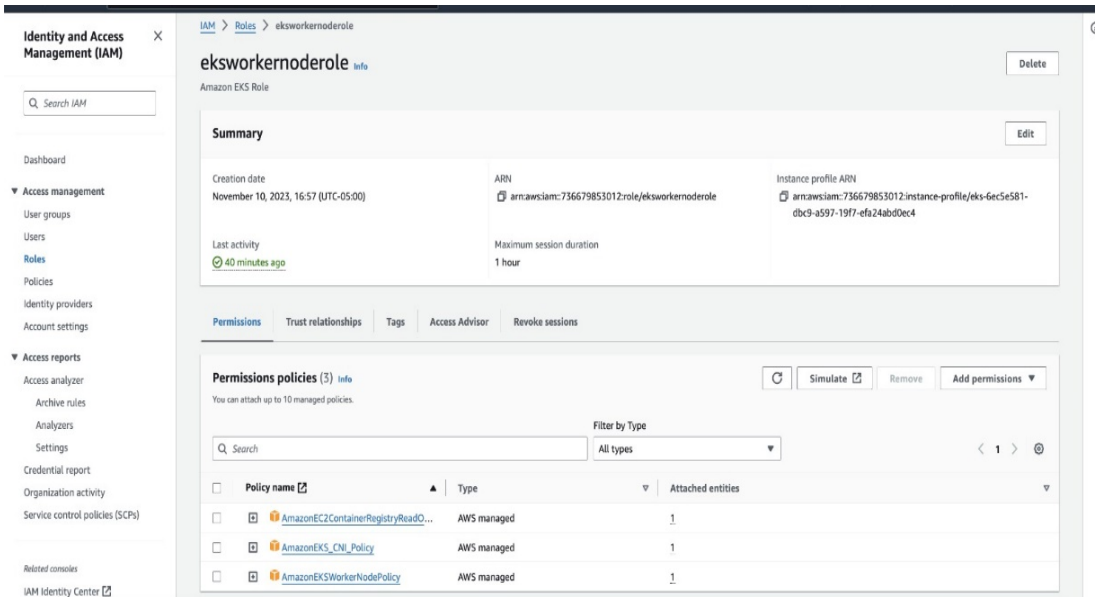
- **Step 4: Install & Setup IAM Authenticator and Kubectl Utility**

```
(base) alina@AlinadeMacBook-Pro ~ % aws configure
AWS Access Key ID [****************KXNS]: AKIA2XBMKPPKABT6KXNS
AWS Secret Access Key [****************u5[5]: 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
Default region name [us-east-1]:
Default output format [None]:
(base) alina@AlinadeMacBook-Pro ~ % aws eks update-kubeconfig --name eks-cluster --region us-east-1

Added new context arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster to /Users/alina/.kube/config
(base) alina@AlinadeMacBook-Pro ~ % aws s3 ls
2022-12-08 15:33:11 amplify-cryptocalcapp-dev-153302-deployment
2022-12-08 15:45:51 amplify-cryptocalcapp-master-204544-deployment
2022-11-19 12:57:53 cdk-hnb659fds-assets-736679853012-us-east-1
2023-11-10 15:47:53 cf-templates-wmlu3xmbqgss-us-east-1
2023-11-10 15:32:48 cf-templates-wmlu3xmbqgss-us-east-2
2023-11-02 14:18:33 elasticbeanstalk-us-east-1-736679853012
(base) alina@AlinadeMacBook-Pro ~ % more /Users/alina/.kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJRHZxdTF0WjdxaXd3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJF
R0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWVGdzB5TXpFeE1UQXlNRFUyTkRsYUZ3MHpNekV4TURjeU1UQXhORGxhTUJVeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FH
U0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUNnYTJnOUhZNERVaUZoLzAvTXBFUS9XaFZKZDhBWXlUZG02T2REVHVpd1o3OCthb0JmcHlPTGRMNGcKQnJxay9ncTRPcERFVTJUU2NDQytj
TmZCRlVQd3dTcFlxTHAzNzFIZ0YzN2pKc0txVHhtK2dadXVkN3JwNmRCMwpxVVRRYUdreUVkYWdCc2YvRHhacTliZXdvcXZsLzZiakU0aHdDaEJpbUIxd1RrUGM2VkVMeVE2MDBOZ0RiL3hXClhl
aE00clk5akRMVjFobWFMcWx5aDArZ0RwY2VOWmVGSE5GQU9oU1p4WndQa2tLLzJHbUlSS255cThPazMrWnQKdEdvYktaUTc5U2xScXBGc1RJeEQ4WEh1L0oyZnBWK3JkU2JqK1NXR3BSTkZwYVV6
elZUcHVzREs0dVR3MkFGaQpYKzN0cWo4djAvVzdNSUd1bHRoMU5tVDk2QnExQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQWY4RUJUQURBUUgvTUIwR0ExVWRE
Z1FXQkJRTGVQR0dETlU2YWRZRVphQmd4NnV3UURsMjVqQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQm9EOE91cjBWSgpDVGxpVFpxeWlX
UmROZFM2dmUvYkhUNjNwWEVtUmxraytCMzJqWXZsUC8zdkx5SFdrOE9TdXlyR1VrNVJaUmlVCnFmS2FKMFdZc0lvRjlOdjBORVFteEllS3pNUnkxOWgrL0ZDdzhuTkJGSjdvUWFBczJHUXpUSk1J
eUxuR1RDVXIKaVNyTnRhenJVUmswSTRKMEtGSDZMWFM2b3FCRG1VdzhISXI4dWtyT1hGYjgxNUNiRWhIUzhxU0p0U2oxajlFSgpOQnY0QWNaRGJDZE91c2tOTkYzZ2pMM0tuRHI0dVJ1VVlPQkxO
YkRvMWtVOGRRMUJRdTQrVHJEN2h2WUN2R2V6CjJuODNzLzFDVDJ6QlVBaGNxRVRjQ0M5UEtGWmN2cWRVWFJpK1ZMajRBV3V1WlBIcUU5WTJKV0xLK29YNlFZWjYKMFRZTVg0bjNvUGhaCi0tLS0t
RU5FIENFUlRJRk1DQVRFLS0tLS0K
    server: https://BCDBA227F5EFFCEF268FF1C141D72692.gr7.us-east-1.eks.amazonaws.com
  name: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
contexts:
- context:
    cluster: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
    user: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
  name: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
current-context: arn:aws:eks:us-east-1:736679853012:cluster/eks-cluster
kind: Config
```

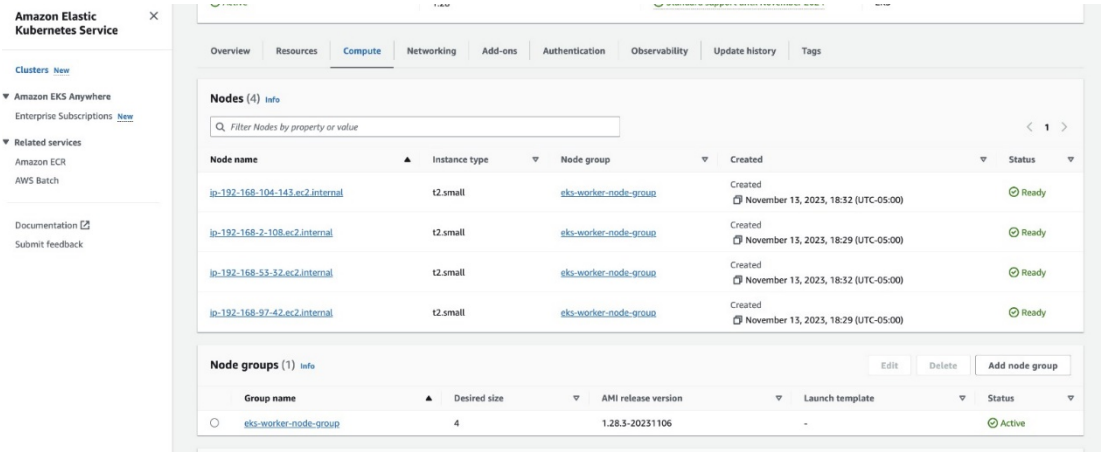
- **Step 5: Create IAM Role for EKS Worker Nodes**

  - Created an IAM role for worker nodes which gives permissions and ec2 container policy to access permissions to each instance in it.

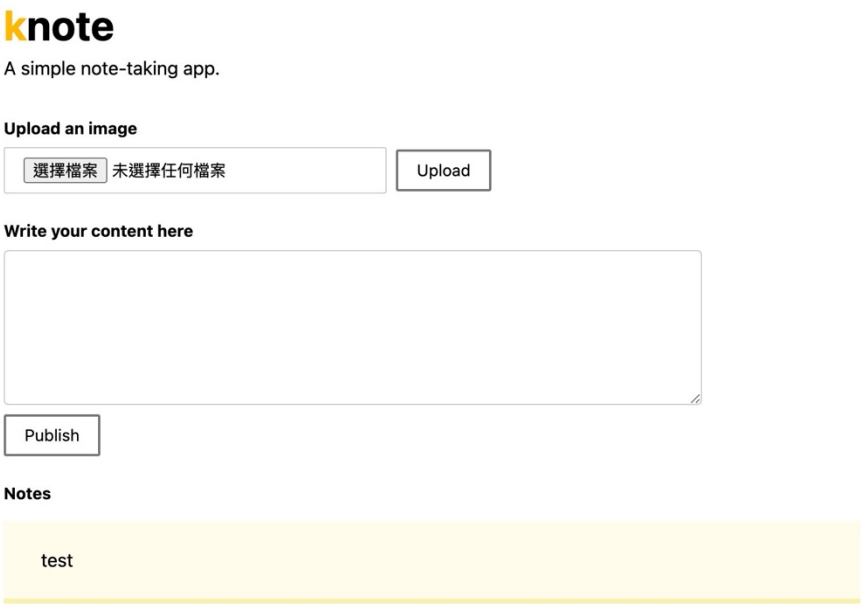


- **Step 6: Create Worker nodes.**

  - Created a worker node by creating a node group.



- **Step 7: Deploying Demo Application**



CloudWatch Visualization for KnowQuest

CloudWatch is a popular visualization and exploration tool often used with the visualization of services. Generally, it's started by ingesting our data from cluster, creating appropriate indices for the data you want to visualize in dashboards. Ensure that the data is properly structured and indexed. Then, we should consider the following steps to create a visualization in CloudWatch.

- **Step 1: Create Visualizations**

  - Use the Visualize section to create various types of visualizations, such as bar charts, line charts, pie charts. Experiment with different visualization options to find the best way to represent our data.

- **Step 2: Dashboard Creation**

  - Combine multiple visualizations into a dashboard to get a holistic view of our data. Dashboards are useful for monitoring and decision-making.

- **Step 3: Time Series Data**

  - If working with time-series data, make use of CloudWatch's time-based features. It excels at visualizing data over time, so take advantage of date histograms, time series charts, and date range filters.

- **Step 4: Filters and Queries**

  - Learn how to use filters and queries to narrow down the data you want to analyze. This is particularly useful when dealing with large streams.

- **Step 5: Customization**

- Customize the appearance of our visualizations and dashboards. We can change colors, labels, and other elements to match our organization's branding.

- **Step 6: Data Enrichment**

  - Use CloudWatch's data enrichment features to add context to our data. We can include scripted fields or use lookup indices to enhance our data with additional information.

- **Step 7: Machine Learning Integration**

  - If we add Kibana to the cluster, it offers machine learning features for anomaly detection and predictive analytics. Explore these capabilities if they align with our use case.

- **Step 8: User Access Control**

  - Implement user access control and authentication to ensure that only authorized users can access and interact with our Dashboard's visualizations.

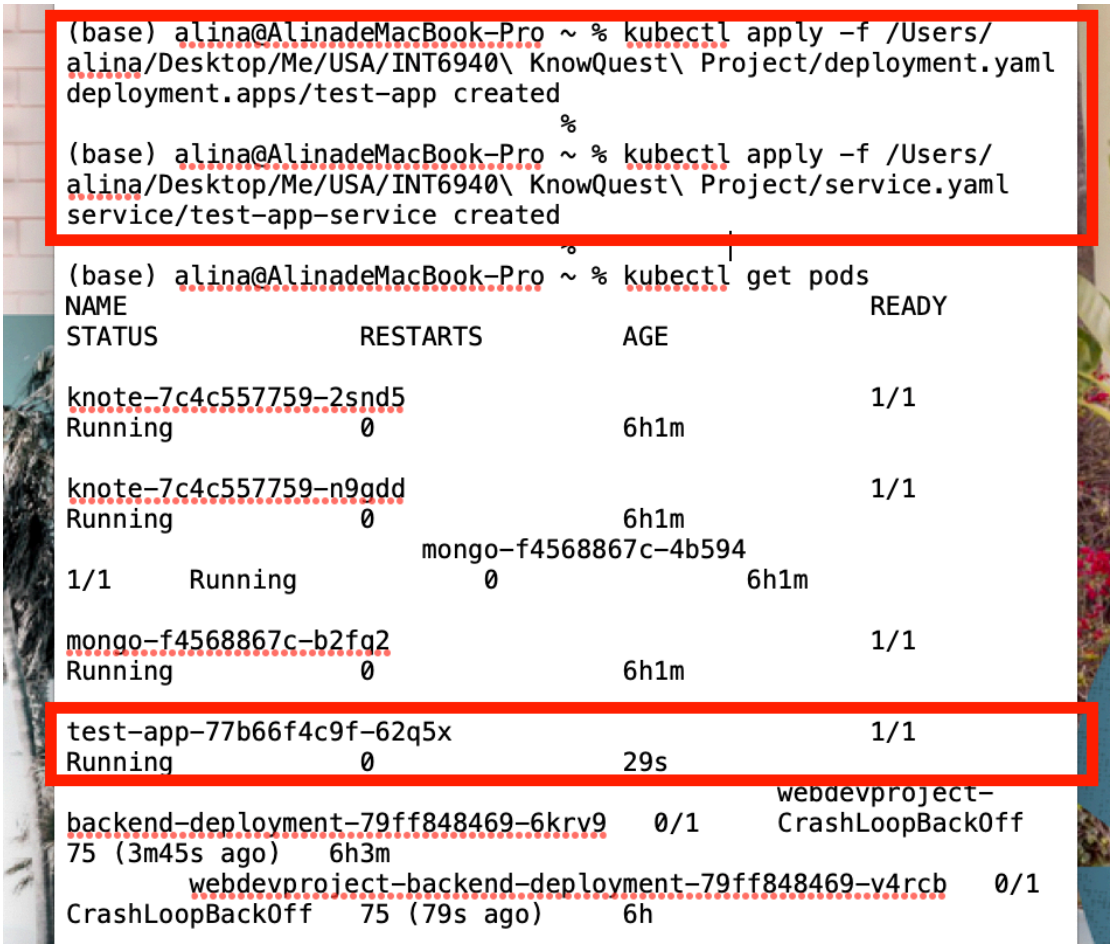
- **Step 9: Performance Optimization**

  - Optimize the performance of your Dashboards, especially when working with large datasets. Properly configure Elasticsearch and use Kibana's settings to improve performance.

Performance Testing

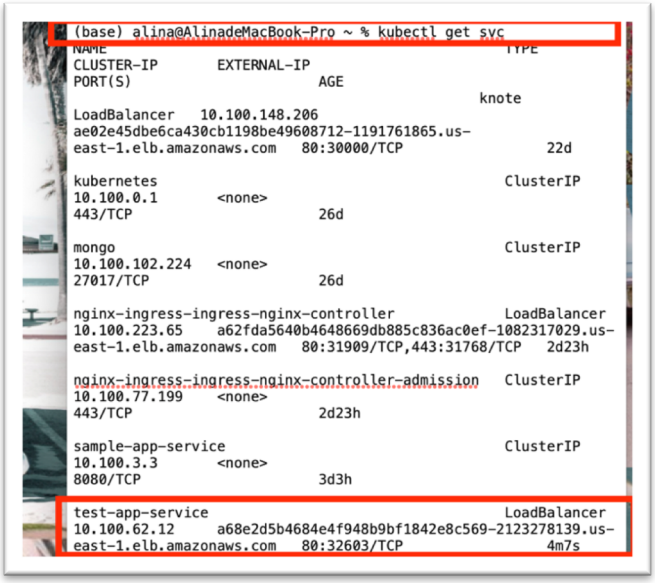
- Implement Application code by creating a docker image.

Step 1: Create yaml files and make sure the application is in the pods and also running. Here we uploaded the sample files, the first one is “deployment.yaml” and the other one is “service.yaml”. Then, we can use “kubectl get pods” command to confirm the application is working.



Step2: Confirm the external link.

After deploying the sample service application, we can use the “kubectl get svc” command to check the external link that been generated by Cluster. As the application shows, the external link will be “ a3aef796bd8a3452787a6e9dfa27bbdb-1492375676.us-east-1.elb.amazonaws.com”



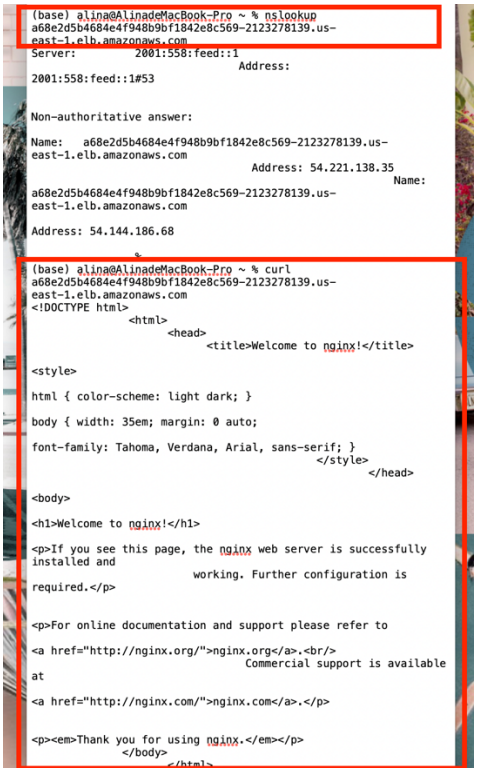
Step 3: Query DNS to obtain domain name mapping or other DNS records. This helps us to determine the IP address with a given domain. By running "nslookup a3aef796bd8a3452787a6e9dfa27bbdb-1492375676.us-east-1.elb.amazonaws.com" command, we are attempting to resolve the domain name to its corresponding IP address. This is crucial in networking to ensure that the domain is correctly mapped to the intended IP, especially when dealing with load balancers or services hosted on cloud platforms like AWS Elastic Load Balancer (ELB).

Furthermore, curl is a command-line tool for making HTTP requests to a specified URL. It is versatile and supports various protocols, making it a widely used tool for testing web services, fetching data, and troubleshooting connectivity. We run the "curl a3aef796bd8a3452787a6e9dfa27bbdb-1492375676.us-east-1.elb.amazonaws.com" command, which means we are sending an HTTP request to the specified URL. This is essential for verifying the web service's accessibility and obtaining a response. It helps ensure that the web server (in this case, the Elastic Load Balancer) is reachable and responding as expected.

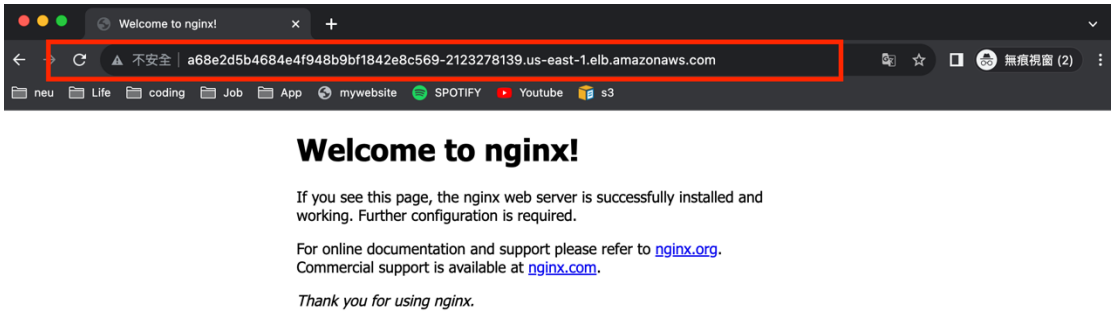
- DNS Resolution: nslookup helps verify that the domain name resolves to the correct IP address. If there are DNS-related issues, such as misconfigurations or delays in

propagation, this step helps identify them.

- Connectivity Testing: After confirming the correct IP address, curl is used to test the connectivity to the web service. It ensures that the server is responsive, and the requested webpage or service is accessible.



Step 4: Test the domain. After we successfully configure the application, we can directly test the domain URL by open a new browser. Here's the successful sample:

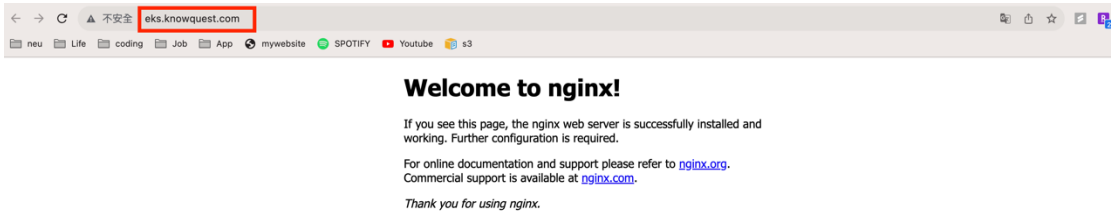


- Update DNS route:

As our sample application is running and able to access by the external URL, we would like to change the URL into KnowQuest’s domain by update in DNS.

Our sample application, which is presently hosted on AWS EKS, is now integrated with KnowQuest's domain through successful DNS route modification. The application can now be accessed through KnowQuest's branded domain, which was previously accessible through an external URL given by AWS. In order to do this, a CNAME record pointing to the DNS name of the load balancer was added in the DNS administration dashboard. This upgrade preserves the application's security and performance benchmarks while guaranteeing smooth brand alignment and improving user accessibility. As DNS transfer is finished, the update is completely effective.

Final result:

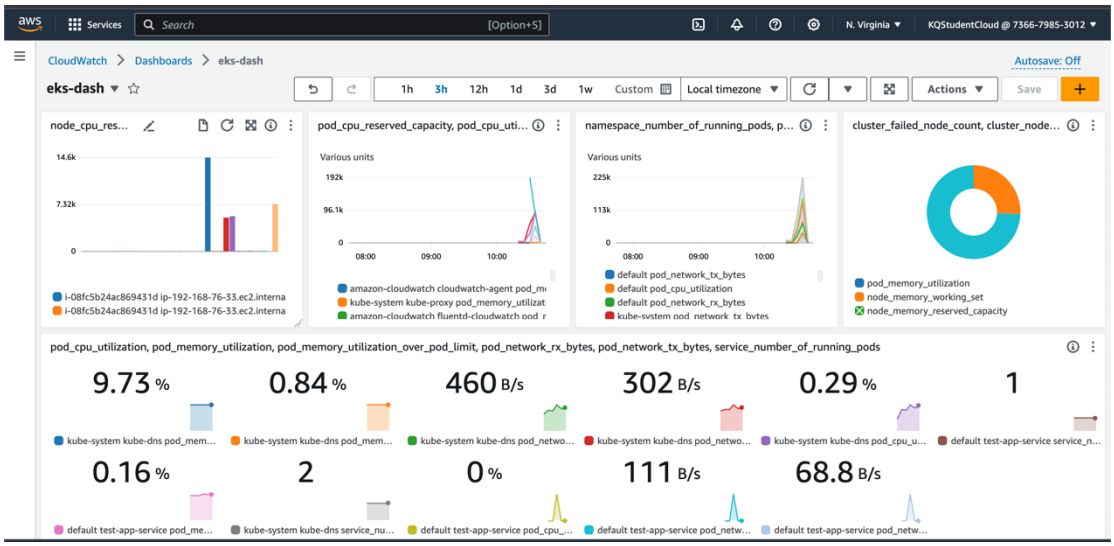


We’ve successfully updated the domain name to KnowQuest’s.

CloudWatch

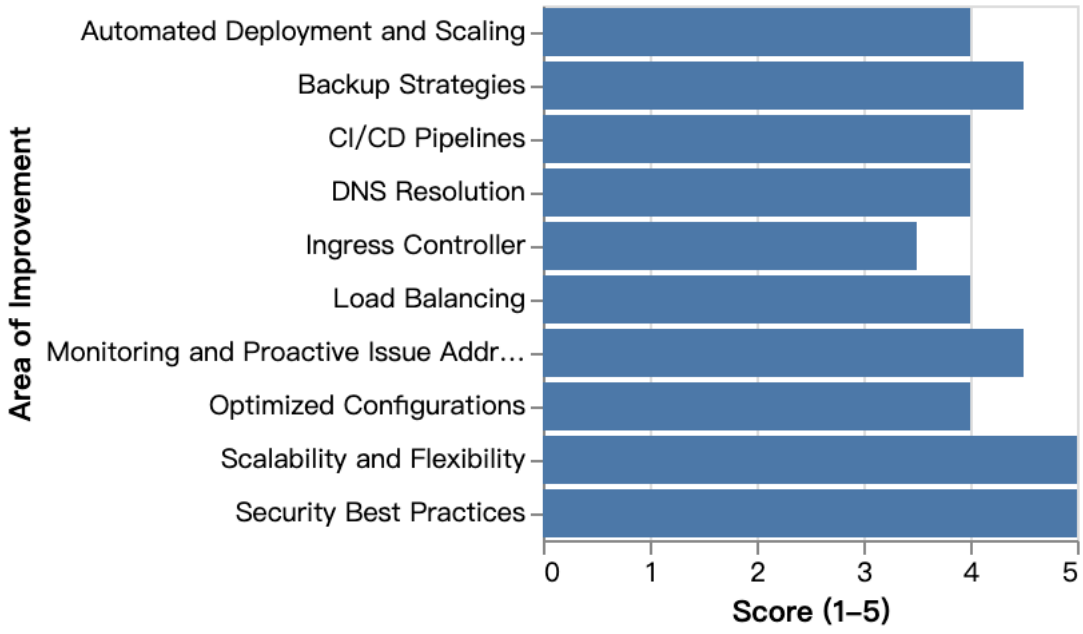
Monitoring AWS workloads and resources is made possible by Amazon CloudWatch, which offers real-time insights via metrics and logs. It's very helpful for monitoring how resources are used, how well applications perform, and how well operations are running. It also enables automated reactions to preset scenarios.

CloudWatch has been integrated into the EKS cluster to provide an outlined overview of the environment. For this, the CloudWatch agent and Fluentd were installed throughout the cluster in order to gather logs and metrics. After that, data gets transmitted to CloudWatch, allowing for ongoing alerting and monitoring. Our system's observability has increased as a result of this integration, enabling us to react quickly to changes and sustain peak performance.

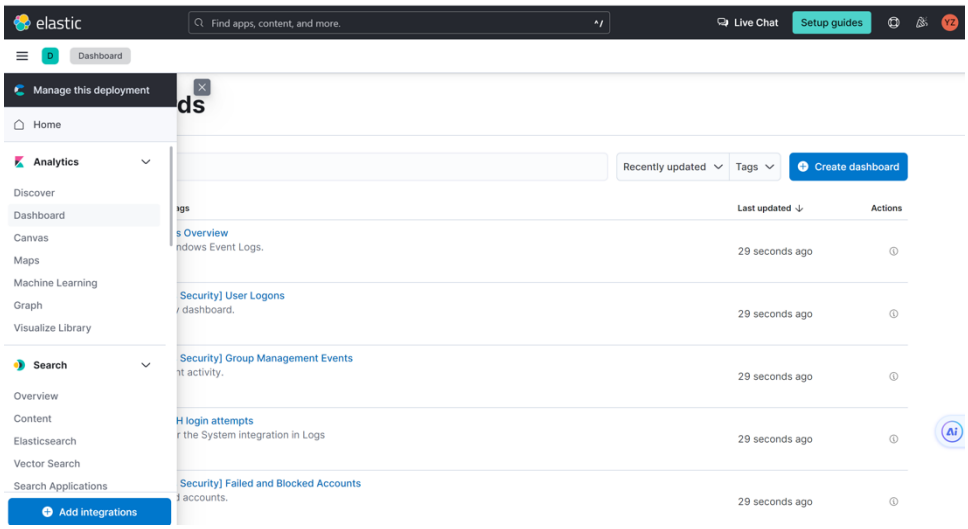


Discussion

The implementation of AWS and Kubernetes infrastructure for KnowQuest is anticipated to significantly enhance its current workflow and operational efficiency. By adopting Kubernetes, the platform gains improved scalability and flexibility, allowing dynamic resource allocation and efficient management of applications. Integration of monitoring tools such as CloudWatch enables real-time insights, contributing to proactive issue identification and faster troubleshooting. Security best practices, including RBAC and network policies, strengthen the infrastructure's resilience against potential threats, ensuring data integrity and compliance. Backup strategies provide an additional layer of protection, assuring data recovery in unforeseen circumstances. Streamlined CI/CD pipelines facilitate a more agile development process, empowering KnowQuest to respond promptly to market demands. Optimized networking configurations, load balancing, DNS resolution, and optional Ingress Controller enhance the user experience, positioning KnowQuest for improved performance and potential future growth. Overall, the implemented infrastructure sets the stage for a more resilient, scalable, and secure environment, addressing current challenges and laying the groundwork for enhanced operational effectiveness.

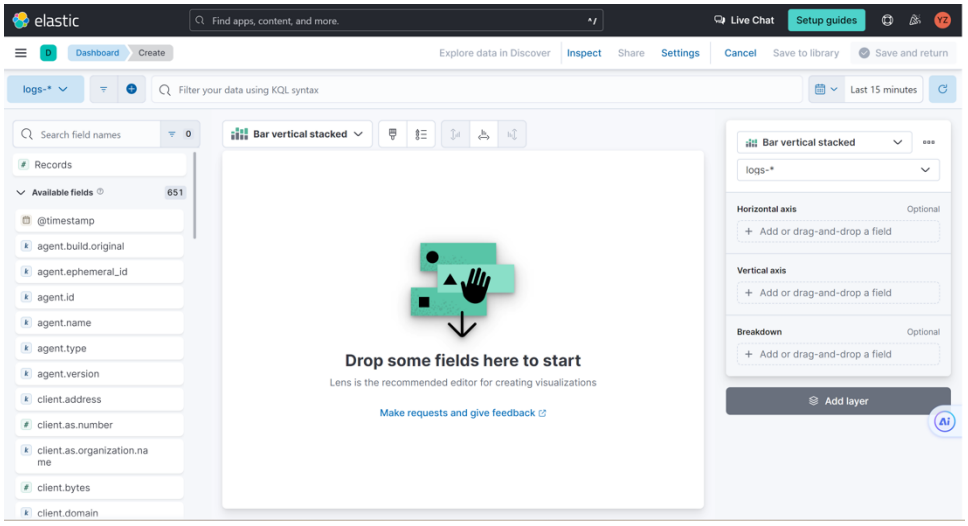


Further Improvements: Data Suggestion



Create a Dashboard:

- Go to the "Dashboard" tab in Kibana.
- Click on the "+" button to add a new panel to the dashboard.
- Choose the type of visualization you want to add to the dashboard and select the saved visualization.



Create Visualizations:

- In the "Visualize" tab, create visualizations for the data. Examples include bar charts, line charts, pie charts, etc.

- Choose the appropriate fields and aggregations for your visualizations.

Log-File processing:

1. Kibana relies on Elasticsearch as the backend data store. So, first, you need to index your log data into Elasticsearch. Ensure that Elasticsearch is set up and running.
2. Configure Logstash or File beat to point to your log files or data sources. Update the configuration files to specify the location of your logs and the Elasticsearch endpoint.
3. After indexing, check that your log data is present in the Elasticsearch indices. You can use the Elasticsearch REST API or Kibana's Dev Tools to query the indices.
4. In Kibana, go to the "Management" tab and then select "Index Patterns."
5. Create an index pattern that matches your log data indices. This step helps Kibana understand how to explore and visualize your data.

Conclusion

The KnowQuest's cloud platform improvement project upon AWS undertaken by Team A can result in a transformative enhancement of the KnowQuest website's efficiency and reliability. The comprehensive approach adopted in redesigning the cloud architecture and migrating to an advanced AWS infrastructure reflects our commitment to delivering a robust PaaS solution. Our journey began with a meticulous setup of the AWS and Kubernetes infrastructure, where we addressed key prerequisites, established a secure VPC, and configured a reliable Kubernetes cluster. The deployment pipeline was automated, employing CI/CD practices to ensure consistency and reproducibility. Throughout this process, our decisions were guided by a focus on scalability, security, and efficiency, aligning with industry best practices.

The integration of latest tools with the help of AWS Services brought a new dimension to data flow and visualization. EKS cluster demonstrated our thoughtful approach to creating a seamless experience for KnowQuest application development. The performance testing phase, emphasized our dedication to meeting high-performance standards, ensuring the KnowQuest website can handle increased traffic and deploy new microservices with a minimal downtime. The success criteria set for the project, including Kubernetes architecture scalability, rapid feature deployment, and the ability to support one million active users, have been diligently met. The AWS migration, featuring services like S3, EC2, RDS, CloudFormation, and CloudWatch, was executed with precision, aligning with KnowQuest's commitment to excellence.

Our implementation showcased a holistic consideration of factors such as monitoring, security, and scaling. We integrated CloudWatch for real-time insights, followed security best practices, and designed backup strategies. The KnowQuest development sitemap, along with the aligning to the dedicated KnowQuest domain, further solidifies our commitment to enhancing the user

experience and ensuring operational excellence.

In conclusion, the KnowQuest Backend and AWS Improvement project not only meets but exceeds the expectations set by the acceptance criteria. Our collaborative efforts, as a team, have not only enhanced the technical infrastructure of KnowQuest but have also contributed to the broader discourse within the discipline of Informatics. This project is not just a technological advancement; it is a testament to our dedication, innovation, and ability to navigate the evolving landscape of Informatics.

We express our gratitude to all team members— **Ananya Kakani, Jie Zhu, Jui Ying Lee, Sandeep Samsani, Vivek Vaghasiya, and Yi Zou**—for their collective efforts and commitment to the success of the KnowQuest Backend and AWS Improvement project.

Thank you.

References

Kubernetes components. (2023, July 12). Kubernetes.

https://kubernetes.io/docs/concepts/overview/components/

Ingress. (2023, October 19). Kubernetes.

https://kubernetes.io/docs/concepts/services-networking/ingress/

Organizing Cluster Access Using kubeconfig Files. (2022, April 13). Kubernetes.

https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/

Best practices for designing Amazon API Gateway Private APIs and Private Integration - Best practices for designing Amazon API Gateway Private APIs and Private integration. (n.d.).

https://docs.aws.amazon.com/whitepapers/latest/best-practices-api-gateway-private-apis-integration/best-practices-api-gateway-private-apis-integration.html?did=wp_card&trk=wp_card

Docker on AWS - Docker on AWS. (n.d.).

https://docs.aws.amazon.com/whitepapers/latest/docker-on-aws/docker-on-aws.html?did=wp_card&trk=wp_card

Analyzing log data with CloudWatch Logs Insights - Amazon CloudWatch Logs. (n.d.).

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

Making Amazon Route 53 the DNS service for an existing domain - Amazon Route 53. (n.d.).

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/MigratingDNS.html

What is a Network Load Balancer? - Elastic Load Balancing. (n.d.).

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html

Getting started with Amazon EKS – AWS Management Console and AWS CLI - Amazon EKS. (n.d.).

https://docs.aws.amazon.com/eks/latest/userguide/getting-started-console.html

Start a pipeline in CodePipeline - AWS CodePipeline. (n.d.).

https://docs.aws.amazon.com/codepipeline/latest/userguide/pipelines-about-starting.html

Policies and permissions in IAM - AWS Identity and Access Management. (n.d.).

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

Amazon EC2 security groups for Windows instances - Amazon Elastic Compute Cloud. (n.d.).

https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-security-groups.html

What is Amazon API Gateway? - Amazon API Gateway. (n.d.).

https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html#api-gateway-overview-aws-backbone