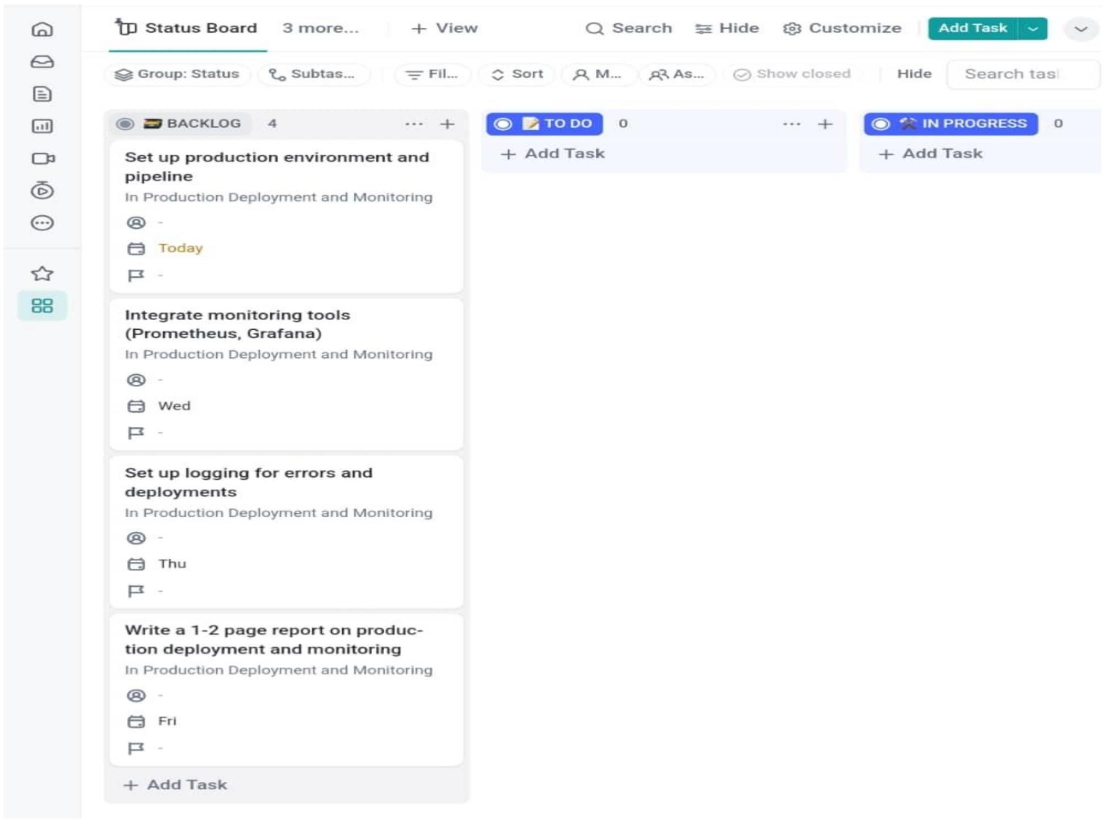
WEEK FOUR REPORT: PRODUCTION, DEPLOYMENT, AND MONITORING

OBJECTIVES:

- Implement production deployment with rollback mechanisms using Azure services.
- Integrate monitoring and logging tools for both staging and production environments within Azure and external services like Prometheus and Grafana.

TASKS



Set up production environment and pipeline
In Production Deployment and Monitoring
-
Today
-

Integrate monitoring tools (Prometheus, Grafana)
In Production Deployment and Monitoring
-
Wed
-

Set up logging for errors and deployments
In Production Deployment and Monitoring
-
Thu
-

Write a 1-2 page report on production deployment and monitoring
In Production Deployment and Monitoring
-
Fri
-

ACTIVITIES OVERVIEW

Each team member was responsible for specific tasks related to production deployment, monitoring, and logging. Below are the steps they took to complete their assignments.

---

1. **Production Deployment and Pipeline Setup**

Assigned to: Ochieng Devote Boniface

Devote was tasked with setting up the production environment using Azure services and configuring the CI/CD pipeline to support automated deployment and rollback strategies.

Steps Taken:

Step 1: Production Environment Setup:

- Set up the production environment on Azure, configuring virtual machines and ensuring the required software (web server, application dependencies) were installed.
- Enforced security protocols such as firewalls, network security groups, and role-based access control (RBAC) in Azure.
- Verified production readiness by performing health checks on the Azure environment, confirming deployment prerequisites.

Step 2: CI/CD Pipeline Setup:

- Configured Azure DevOps to manage the CI/CD pipeline, integrated with GitHub for version control.
- Integrated Azure Kubernetes Service (AKS) with Helm for deploying and rolling back applications.
- Tested rollback mechanisms by triggering a failed deployment and validating Helm's rollback capabilities.
- Set up notifications for deployment status using Azure Application Insights and Azure notifications.

Deliverables:

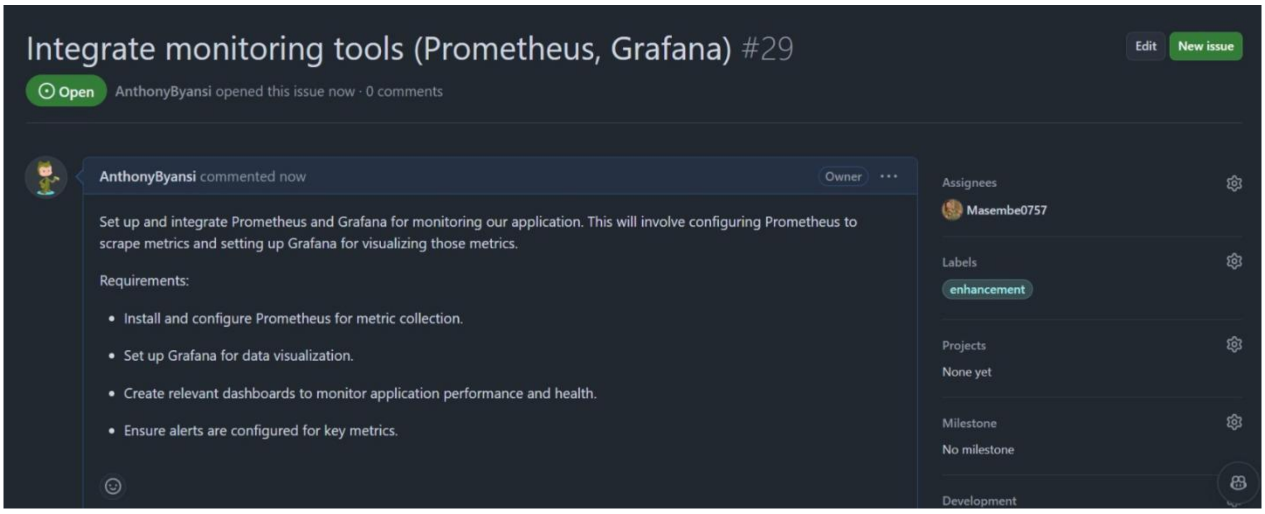
- Fully configured production environment on Azure.
- CI/CD pipeline with rollback capabilities.
- Azure Application Insights and Azure notifications for deployment events.

---

2. **Monitoring Tool Integration (Prometheus and Grafana)**

Assigned to: Sendi Joseph

Joseph was responsible for integrating monitoring tools to track system performance and ensure the application's health on Azure.



Integrate monitoring tools (Prometheus, Grafana) #29

Open AnthonyByansi opened this issue now · 0 comments

AnthonyByansi commented now

Set up and integrate Prometheus and Grafana for monitoring our application. This will involve configuring Prometheus to scrape metrics and setting up Grafana for visualizing those metrics.

Requirements:

- Install and configure Prometheus for metric collection.
- Set up Grafana for data visualization.
- Create relevant dashboards to monitor application performance and health.
- Ensure alerts are configured for key metrics.

Assignees: Masembe0757

Labels: enhancement

Projects: None yet

Milestone: No milestone

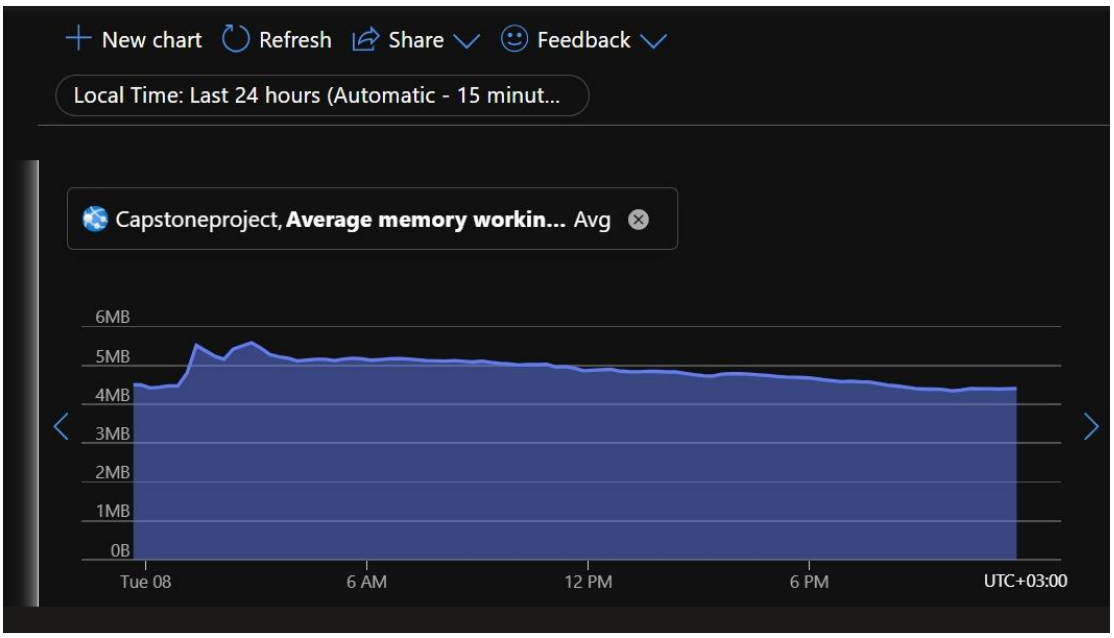
Steps Taken:

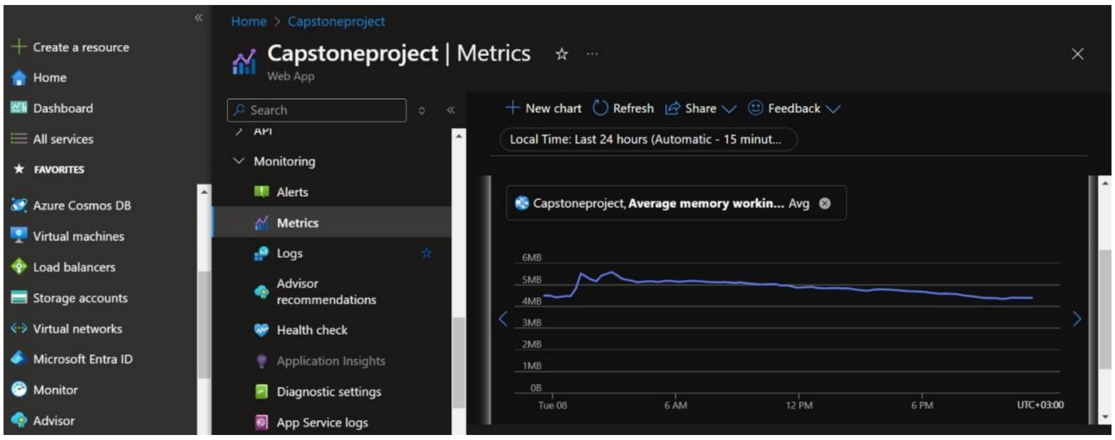
Step 1: Prometheus Setup:

- Set up Prometheus on both the staging and production environments hosted on Azure Kubernetes Service (AKS).
- Configured Prometheus to collect and scrape metrics such as CPU usage, memory utilization, page load times, and response times from Azure.
- Monitored AKS pods and services in real-time, providing insights into resource consumption and system performance.

Step 2: Grafana Dashboard Setup:

- Configured Grafana to visualize metrics gathered from Prometheus, focusing on key performance indicators like average working memory, request rates, and CPU usage.
- Created dashboards to track HTTP 4xx and 5xx errors, page load times, and system health.
- Set up alert rules to notify the team via Azure Application Insights when metrics cross defined thresholds, such as high memory usage or failed requests.


New chart
Refresh
Share
Feedback
Local Time: Last 24 hours (Automatic - 15 minut...
Capstoneproject, Average memory workin... Avg
6MB
5MB
4MB
3MB
2MB
1MB
0B
Tue 08
6 AM
12 PM
6 PM
UTC+03:00



Deliverables:

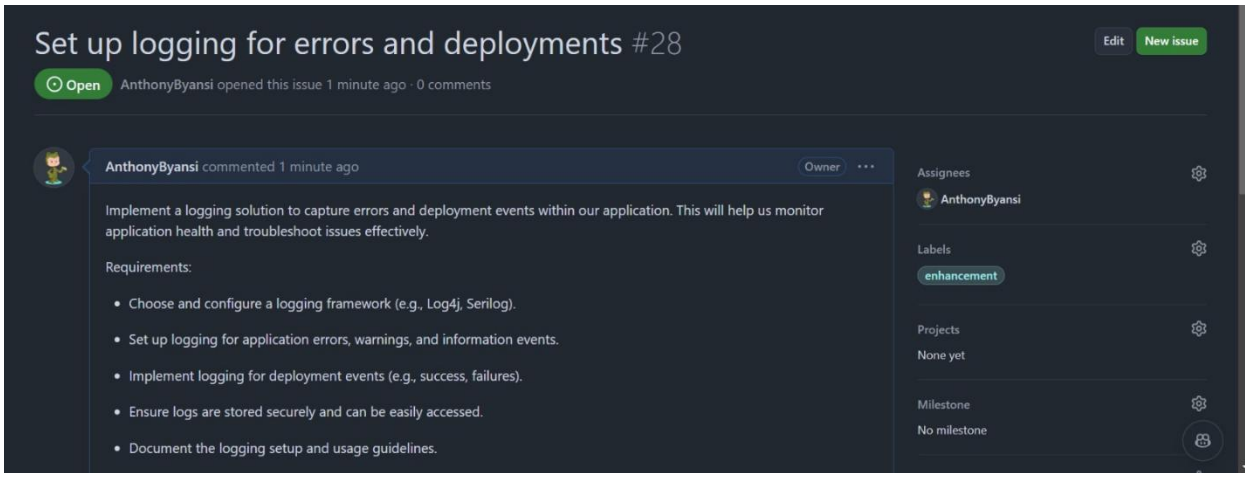
- Prometheus monitoring integrated into Azure AKS for both staging and production environments.
- Grafana dashboards visualizing real-time system performance metrics.
- Alerts configured for key metrics using Azure Application Insights.

---

3. **Logging Mechanism Setup**

Assigned to: Byansi Anthony

Anthony was responsible for setting up a logging mechanism to capture deployment events and application errors on Azure.



Set up logging for errors and deployments #28

Edit New issue

Open AnthonyByansi opened this issue 1 minute ago · 0 comments

AnthonyByansi commented 1 minute ago Owner

Implement a logging solution to capture errors and deployment events within our application. This will help us monitor application health and troubleshoot issues effectively.

Requirements:

- Choose and configure a logging framework (e.g., Log4j, Serilog).
- Set up logging for application errors, warnings, and information events.
- Implement logging for deployment events (e.g., success, failures).
- Ensure logs are stored securely and can be easily accessed.
- Document the logging setup and usage guidelines.

Assignees
AnthonyByansi

Labels
enhancement

Projects
None yet

Milestone
No milestone

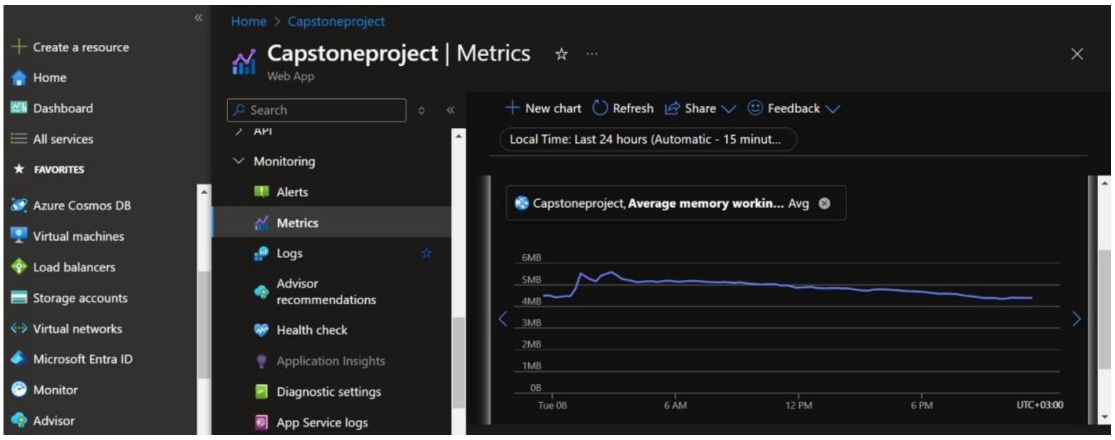
Steps Taken:

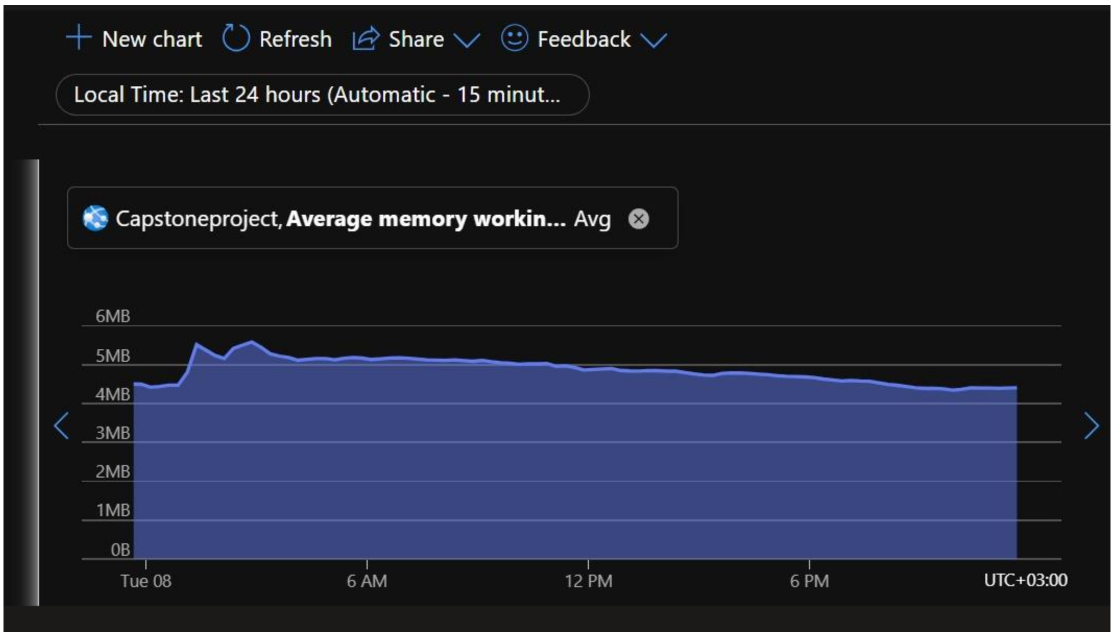
Step 1: Azure Monitor and ELK Stack Installation

- Set up Azure Monitor and integrated it with the Elastic Stack (ELK: Elasticsearch, Logstash, Grafana) for centralized logging.
- Configured Azure Monitor to collect logs from the application and Kubernetes clusters in Azure.

Step 2: Log Collection and Filtering:

- Configured Logstash to aggregate logs from application services and deployment events within Azure.
- Set up filters to capture critical logs such as system errors, HTTP 4xx and 5xx errors, and security events.





Step 3: Azure Application Insights Dashboards:

- Used Azure Application Insights to visualize deployment logs, error tracking, and security incidents.
- Azure Application Insights with ELK were used to track logs related to key deployment cycles and failures.

Deliverables:

- ELK Stack fully set up and integrated with Azure.
- Azure Application Insights dashboards displaying filtered error and deployment logs.
- Continuous logging for error and deployment tracking.

4. Report Writing on Production, Deployment, and Monitoring

Assigned to: Aparo Cecilia

Cecilia was responsible for compiling the Week 4 report, documenting the tasks related to production deployment and monitoring setup using Azure services.

Steps Taken:

Step 1: Compilation of Production Setup Details:

- Gathered details from Devote regarding the production environment setup in Azure and the CI/CD pipeline configuration.
- Documented the usage of Helm for deployment rollbacks within Azure Kubernetes Service.

Step 2: Documentation of Monitoring Tools:

- Collected information from Joseph on the integration of Prometheus and Grafana, including the KPIs being tracked such as average memory usage, request rates, and error rates.
- Highlighted alert configurations set for CPU and memory consumption, page load times, and HTTP error rates using Azure Application Insights.

Step 3: Logging Setup Documentation:

- Included Anthony's work on logging, detailing how Azure Monitor and ELK Stack were used to capture and visualize logs.
- Provided screenshots of the Azure Application Insights dashboard showing average working memory, deployment logs etc.