Welcome

Yusuf Choudhari
DevOps Project

INTRODUCTION TO CLIENT PROJECT

We have UK based client where we are managing their ecommerce based application.

That need to process more than ١٥٥٥٥ transaction & order request daily and parallelly

Manage and store the application data.

To give seamless experience to our customer we came across AWS EKS one stop solution

With ideal Operational Expense & Capital Expense.

PROBLEMS TO OVERCOME

- Earlier we don't have Micro-container service architecture and our ecommerce Application is based on monolithic architecture. Its replicated on multiple ec2 vm.
- it doesn't have proper request routing and scheduling of Api request to ec2 vm.
    - No optimize resources some nodes get down and some has unused resources

By which we have 2 hours of downtime at peak hours.

- Reliability – If there's an error in any module, it could affect the entire application's availability.

AWS ELASTIC KUBERNETES SOLUTION

- **Solution**: Use a container network interface [CNI] plugin like Calico or Flannel to simplify networking configuration. Implement network policies to control traffic between pods.
- **Solution**: Use EBS persistent volume to claims [PVCs] and manage storage volumes. for seamless integration with different pods.
    - **Solution**: Implement cluster autoscaling to automatically add or remove nodes based on resource utilization. Use horizontal pod autoscaling [HPA] to scale individual deployments based on CPU or custom metrics.
    - **Solution**: Utilize resource quotas and limits to control resource allocation per namespace or pod. Implement resource monitoring and analysis tools to identify and optimize resource usage.
    - **Solution**: Implement RBAC [Role-Based Access Control] to control user and service account permissions. Utilize Kubernetes Secrets and ConfigMaps for secure storage of sensitive information.

CHALLENGES THAT ACHIEVED.

The most challenging phase for us is to create Docker images for different API requests.

By making the correct version of libraries, packages and migration of that files to Container and making this Images available at docker registry.

Thanks for our team we achieved it successfully.

Tools and technology

Version Control System: Git

Continuous Integration Server: Jenkins

Build Tool: Maven

Code Quality Tool: SonarQube

Artifact Repository: Nexus

Containerization Tool: Docker

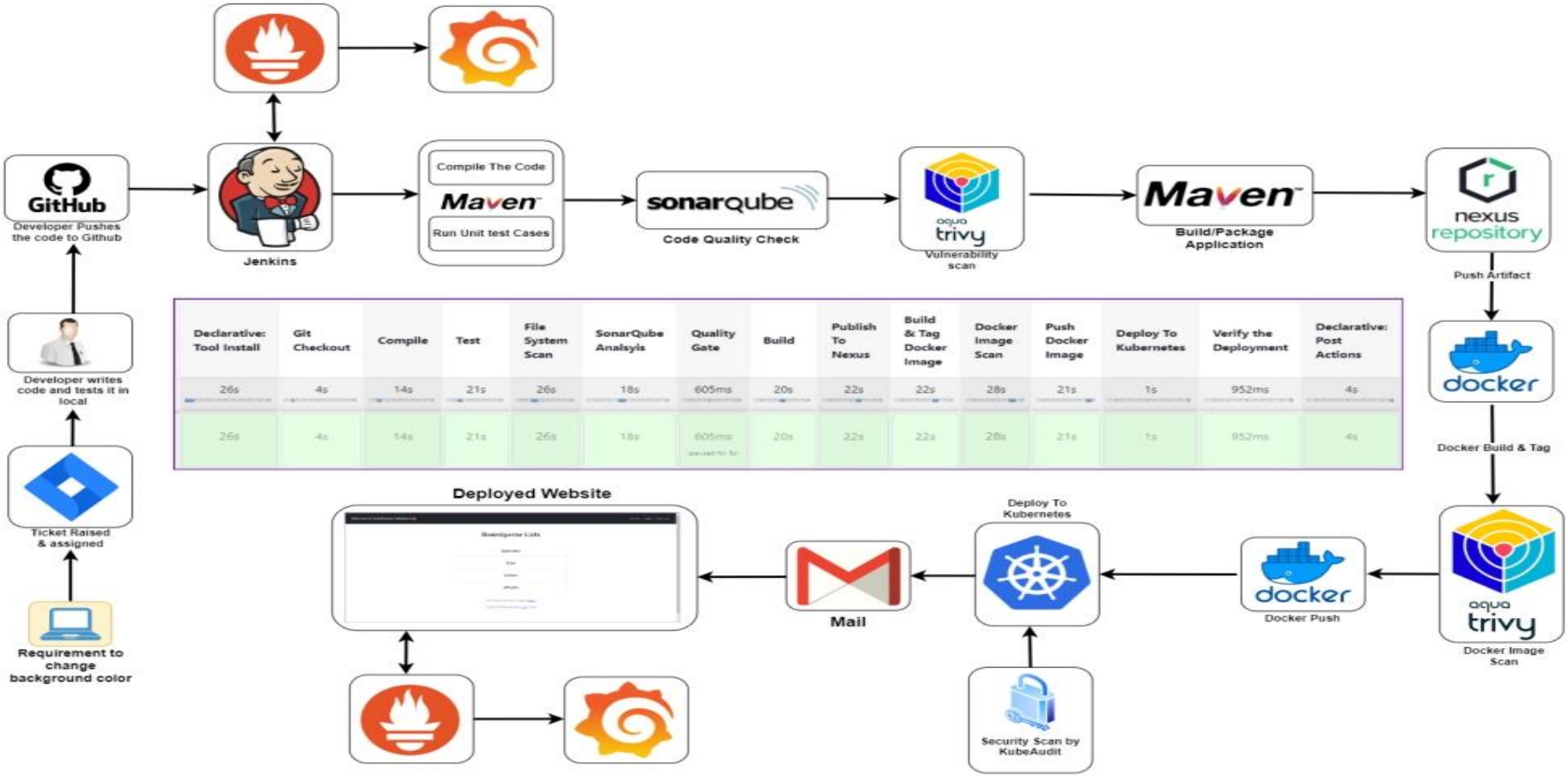
Container Orchestration Platform: Kubernetes

Security Scanners: Trivy

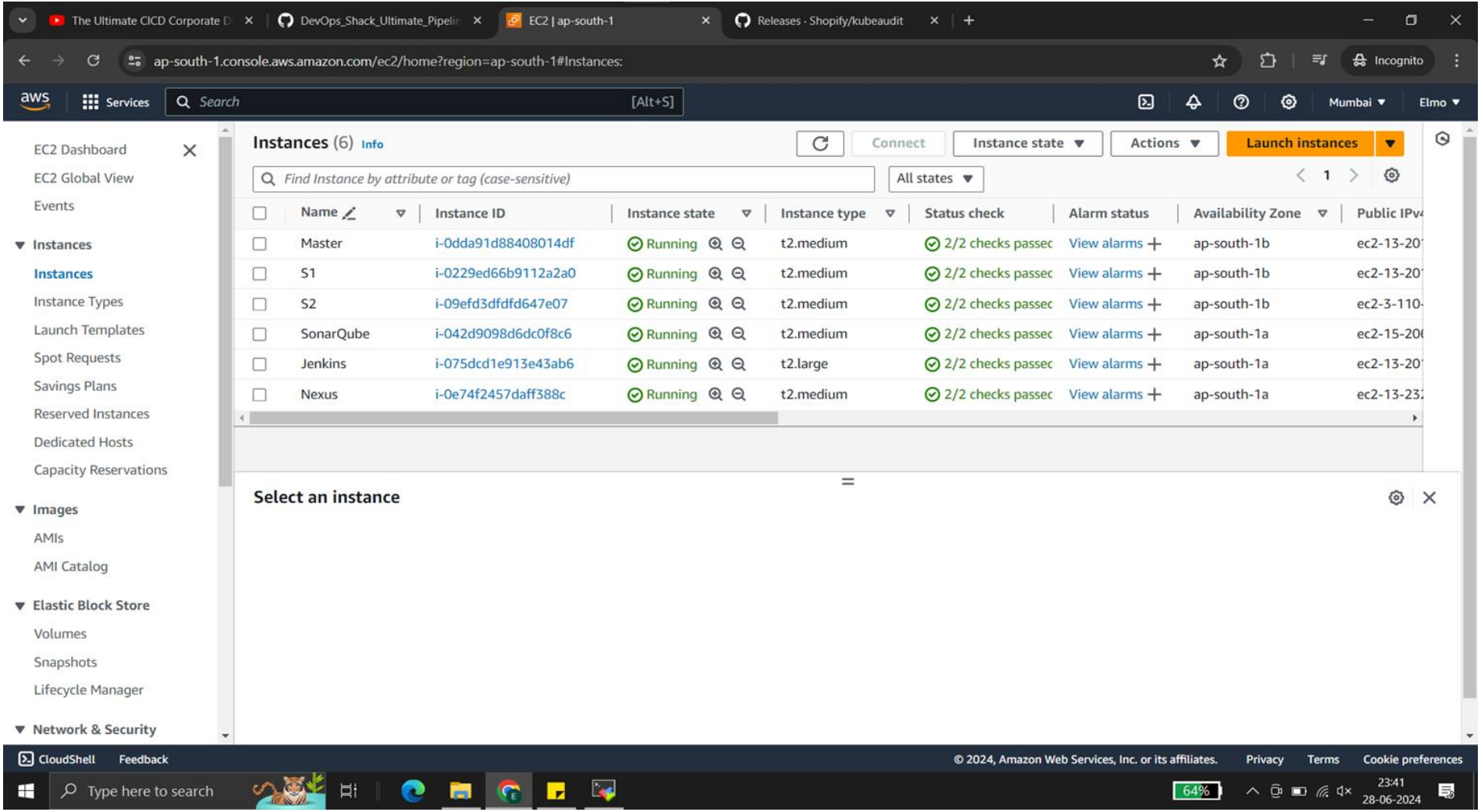
Monitoring tools : Prometheus and Grafana

AWS Services

- Amazon CloudTrail
- Amazon CloudWatch
- Amazon EC2 (Elastic Compute Cloud)
- Amazon ECR (Elastic Container Registry)
- Amazon EKS (Elastic Kubernetes Service)
- Amazon RDS (Relational Database Service)
- Amazon Route 53
- Amazon S3 (Simple Storage Service)
- AWS CloudFormation
- AWS IAM (Identity and Access Management)
- AWS VPC (Virtual Private Cloud)



Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analsyis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	Verify the Deployment	Declarative: Post Actions
26s	4s	14s	21s	26s	18s	605ms	20s	22s	22s	28s	21s	1s	952ms	4s
26s	4s	14s	21s	26s	18s	605ms	20s	22s	22s	28s	21s	1s	952ms	4s


The Ultimate CICD Corporate D
DevOps_Shack_Ultimate_Pipelin
EC2 | ap-south-1
Releases · Shopify/kubeaudit
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:
Incognito
aws
Services
Search
[Alt+S]
Mumbai
Elmo
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Instances (6) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4
Master i-0dda91d88408014df Running t2.medium 2/2 checks passed View alarms ap-south-1b
S1 i-0229ed66b9112a2a0 Running t2.medium 2/2 checks passed View alarms ap-south-1b
S2 i-09efd3dfdfd647e07 Running t2.medium 2/2 checks passed View alarms ap-south-1b
SonarQube i-042d9098d6dc0f8c6 Running t2.medium 2/2 checks passed View alarms ap-south-1a
Jenkins i-075dcd1e913e43ab6 Running t2.large 2/2 checks passed View alarms ap-south-1a
Nexus i-0e74f2457daff388c Running t2.medium 2/2 checks passed View alarms ap-south-1a
Select an instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
64%
23:41
28-06-2024

Inbound rules (8)

Manage tags | Edit inbound rules

Search

1

Type	Protocol	Port range	Source
SMTP	TCP	25	0.0.0.0/0
Custom TCP	TCP	3000 - 10000	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
Custom TCP	TCP	6443	0.0.0.0/0
SMTPS	TCP	465	0.0.0.0/0
Custom TCP	TCP	30000 - 32767	0.0.0.0/0

Thank You for your Support
And Time.