DEPLOYING WHITEBOARD APPLICATION IN CLOUD

CLOUD SERVICE: AMAZON WEB SERVICES

Documentation

Introduction

To launch a scalable whiteboard application, this study details the design and construction of a private cloud architecture utilizing AWS (Amazon Web Services). The main objective was to develop a system that could scale automatically in response to user demand while maintaining data consistency and real-time communication amongst dispersed instances. To provide a durable and elastic cloud environment, the system incorporates capabilities like load balancing, multi-availability zone deployment, and secure networking by utilizing AWS's powerful services.

With a synced state across instances, the distributed whiteboard application was designed to enable smooth user communication. Performance and dependability were maximized by the smart use of AWS services, such as CloudWatch for monitoring, S3 for scalable storage, and EC2 for computation. The VPC offered a safe and segregated network environment that was necessary for operational integrity, and auto-scaling guaranteed resource availability under fluctuating traffic demands.

The project's achievements and potential areas for development are assessed in this report. Scalability and fault tolerance are two of the system's main advantages, although issues like latency optimization and sophisticated security measures are noted for future development. The project provides insightful information for related projects and demonstrates how AWS services may be successfully integrated to create scalable and reliable cloud-based solutions.

Design of Private Cloud

AWS services are used in the private cloud architecture to provide a stable, adaptable, and secure environment. AWS was chosen because of its extensive service portfolio, worldwide reach, and cutting-edge security and scalability characteristics. The infrastructure was built with performance optimization, high availability, and fault tolerance in mind.

Amazon Simple Storage Service (S3) for scalable storage, Elastic Compute Cloud (EC2) instances for computational resources, and Virtual Private Cloud (VPC) for isolated networking are the main components of the concept. To increase dependability and reduce downtime, the VPC was designed with many subnets spread around Availability Zones (AZs). NAT gateways and Internet gateways were used to set up network routing, guaranteeing safe and effective connectivity.

A distributed architecture is incorporated into the design to guarantee fault tolerance and scalability. Application Load Balancers (ALBs), which are used in load balancing systems, uniformly distribute user traffic among EC2 instances. To ensure responsiveness and cost-effectiveness, auto-scaling groups were set up to dynamically modify resources in response to demand.

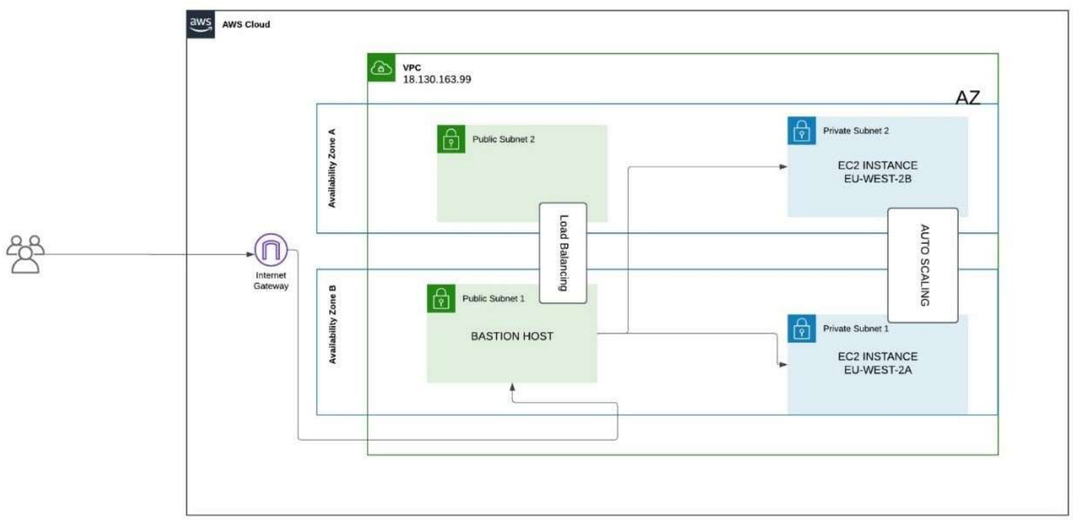
VPCs, subnets, EC2 instances, ALBs, and S3 storage are among the components whose interconnections are depicted in the architectural diagram. After weighing several options, each component was selected to satisfy the project's scalability, consistency, and fault tolerance criteria.

Architecture Diagram

VPCs, subnets, EC2 instances, ALBs, and S3 storage are among the components whose interconnections are depicted in the architectural diagram. After weighing several options, each component was selected to satisfy the project's scalability, consistency, and fault tolerance criteria.

The figure, which displays the allocation of resources among Availability Zones (AZs), offers a clear visual representation of the architectural configuration. The Bastion Host and Internet Gateway are located on public subnets, which provide external connectivity and safe administrative access. In contrast, EC2 instances are hosted on private subnets to guarantee the security and isolation of backend operations.

The architecture's use of Application Load Balancers (ALBs) guarantees that traffic is distributed fairly among EC2 instances situated in various Availability Zones. Because auto-scaling groups automatically provide or decommission resources in response to real-time demand, they further increase the flexibility of the infrastructure. By avoiding over-provisioning and minimizing latency, this design preserves peak performance across a range of workloads.



Implementation of Private Cloud

The establishment of a Virtual Private Cloud (VPC) to create a secure and isolated network environment marked the start of the implementation phase. To offer fault tolerance and high availability, the VPC was set up with subnets dispersed among many Availability Zones (AZs). To ensure accessibility for authorized services and users while guaranteeing strong data privacy and integrity, security groups were meticulously created to control both inbound and outgoing traffic.

The private cloud's computational backbone was then formed by allocating Elastic Compute Cloud (EC2) instances within the VPC. These instances were set up to function as a component of auto-scaling groups, which dynamically modified resources in response to real-time data tracked by CloudWatch, including CPU use and network traffic. Cost effectiveness and reliable performance were made possible by the smooth adaptability to changing workloads that this automatic resource allocation guaranteed.

Amazon Simple Storage Service (S3) was used as the storage layer to control data backup and durability. S3 buckets were set up to manage both recurring backups and real-time application state storage. Amazon CloudWatch made it easier to monitor and log, which allowed for quick problem detection and crucial insights into system performance. In order to sustain performance levels even under high loads and avoid service outages brought on by unequal resource use, Application Load Balancers (ALBs) were added to distribute incoming traffic evenly among EC2 instances.

Development and Deployment of Distributed Whiteboard Application

The distributed whiteboard application was adapted from the Paint Web App open-source repository on GitHub to meet project needs. The development of an application designed to satisfy real-time multi-user collaboration requirements was based on this open-source project. The backend infrastructure was significantly improved to support distributed architecture, real-time updates, and distant synchronization while maintaining data integrity.

As part of the deployment process, the application was hosted on EC2 instances dispersed among many Availability Zones (AZs) in the Virtual Private Cloud (VPC). High availability and fault tolerance were ensured by this multi-AZ strategy. Real-time propagation of state changes between instances was made possible by the implementation of a distributed event synchronization system. In addition, conflict resolution techniques were created to guarantee the integrity of the application state if many users made changes at the same time.

The whiteboard program supports the following important interactive features:
Drawing tools: Users can draw lines, shapes, and freehand. The capability to add and modify text on the canvas is known as text annotations.
State retrieval: To guarantee consistency, the application retrieves and shows the whiteboard's current state when the user logs in.
The web application's clear and simple interface was created with the user experience in mind. The application stays responsive even when there is a lot of traffic since the backend effectively manages several requests at once. The application is adaptable for team talks, brainstorming sessions, and teaching because of features like colour customization, undo/redo, and a transparent board.

A quick and reliable system was guaranteed by integrating AWS services like CloudWatch for real-time performance monitoring and S3 for permanent state storage. By reducing service disruptions and offering insights into system health, these AWS technologies improved fault tolerance. The whiteboard application's strong deployment approach establishes it as a scalable, fault-tolerant, and intuitive real-time collaboration tool.

Demonstration of Private Cloud

To prove its functioning and confirm that it could fulfill project objectives, the system underwent extensive testing in a controlled setting. The private cloud's dynamic scalability was demonstrated throughout the demonstration by adapting in real-time to changing user loads. Amazon CloudWatch was used to track metrics including response times, resource use, and consistency, which gave useful information about how well the system performed in various situations.

Verifying the distributed nature of the whiteboard application was one of the demonstration's most important components. Updates done on one instance instantly synced with all instances, and users were linked to many EC2 instances spread across several Availability Zones. This real-time synchronization ensured a smooth and reliable user experience by demonstrating the efficacy of the event propagation and conflict resolution methods.

An important factor in improving the system's fault tolerance and dependability was the multi-AZ deployment. The application maintained high availability across simulated instance failures, with the Application Load Balancer (ALB) smoothly redirecting traffic to live instances. This demonstrated how the system can reduce downtime and guarantee continuous operation. All things considered, the demonstration successfully illustrated the system's scalability, durability, and capacity to manage collaborative real-world situations.

Critical Review of the System

Advantages

• **Scalability and Auto-scaling:** The system's scalability was outstanding. Dynamically managed resources are auto scaled in response to user traffic, guaranteeing peak performance without resource waste or overprovisioning.

• High availability and fault tolerance were made possible by the **multi-AZ implementation**. The system operated without any noticeable downtime, even during simulated failures.

• **Consistency Across Instances:** Data consistency across several nodes was guaranteed by the distributed design. A smooth user experience was made possible by changes that were synchronized in real time.

• **Performance Monitoring:** By integrating with CloudWatch, real-time system performance tracking was made possible, which facilitated the efficient identification and removal of bottlenecks.

Weaknesses:

• **Latency in Real-Time Collaboration:** Despite the distributed architecture, real-time collaboration was hampered by latency problems that were noticed in situations with significant traffic.

• **Security Improvements Required**: Although sufficient, security measures might be reinforced with features including more stringent access controls, end-to-end encryption, and sophisticated intrusion detection.

• **Cost Efficiency:** During unforeseen traffic surges, auto-scaling occasionally resulted in over-provisioning, raising operating expenses.

Upcoming Enhancements:

- **Latency Optimization:** To shorten the time, it takes for users to reach servers, implement edge computing.

- Put **content delivery networks (CDNs)** into place to store frequently requested information near users. Improve response times by optimizing network routing.

- **Improved Security Measures:** Encrypt all communications from beginning to end. Multi-factor authentication (MFA) should be implemented for user access.
  To find and eliminate any risks, use AI-powered intrusion detection systems.

- **Cost management:** Use predictive scaling to modify resources in accordance with past consumption trends. To cut expenses for predictable workloads, use reserved EC2 instances. Optimize storage use by putting S3 data lifecycle regulations into place.

CONCLUSION

In Conclusion, this project effectively illustrates how to use AWS services to design, build, and deploy a scalable private cloud architecture. High availability, fault tolerance, and dynamic resource management are provided by the system by utilizing features like load balancing, auto scaling, and multi-AZ deployment. Adapted from an open-source repository, the distributed whiteboard application demonstrates the capacity to manage real-time multi-user collaboration while guaranteeing data integrity and dependability.

The project's major accomplishments are the smooth integration of AWS services, improved system performance in a range of traffic scenarios, and an intuitive interface for teamwork. Although there is always need for development in areas like latency optimization and sophisticated security measures, the strong architecture and deployment plan offer a strong basis for comparable cloud-based solutions.

All things considered, this project demonstrates how AWS cloud architecture can satisfy the needs of contemporary, scalable, and robust applications while providing insightful information about actual implementation situations.

References

1. “Private Cloud - Amazon Virtual Private Cloud (VPC) - AWS,” *Amazon Web Services, Inc.*, 2024. https://aws.amazon.com/vpc (accessed Dec. 12, 2024).

2. “Amazon EC2,” *Amazon Web Services, Inc.* https://aws.amazon.com/ec2 (accessed Nov. 07, 2021).

3. “Cloud Object Storage – Amazon S3 – Amazon Web Services,” *Amazon Web Services, Inc.* https://aws.amazon.com/s3 (accessed Dec. 08, 2022).

4. M. van Steen and A. S. Tanenbaum, “A brief introduction to distributed systems,” *Computing*, vol. 98, no. 10, pp. 967–1009, Aug. 2016, doi: https://doi.org/10.1007/s00607-016-0508-7.

5. A. Alalawi and Alauddin Al-Omary, “A Survey On Cloud-Based Distributed Computing System Frameworks,” *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, Oct. 2020, doi: https://doi.org/10.1109/icdabi51230.2020.9325662.

6. Prometheus, “Overview | Prometheus,” *Prometheus.io*, 2016. https://prometheus.io/docs (accessed Dec. 12, 2024).

7. “Technical documentation | Grafana Labs,” *Grafana Labs*, 2024. https://grafana.com/docs (accessed Dec. 12, 2024).

8. R. Buyya, J. Broberg, and A. Goscinski, “CLOUD COMPUTING Principles and Paradigms Edited by.” Available: http://dhoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/CLOUD%20COMPUTING%20Principles%20and%20Paradigms.pdf

9. K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, “Parallel data processing with MapReduce,” *ACM SIGMOD Record*, vol. 40, no. 4, pp. 11–20, Jan. 2012, doi: https://doi.org/10.1145/2094114.2094118.

10. asudbring, “Azure networking documentation,” *learn.microsoft.com*. https://learn.microsoft.com/en-us/azure/networking/

11. R. Ramakrishnan *et al.*, “Azure Data Lake Store,” *Proceedings of the 2017 ACM International Conference on Management of Data*, May 2017, doi: https://doi.org/10.1145/3035918.3056100.

12. J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008, doi: https://doi.org/10.1145/1327452.1327492.

13. “Nagios Documentation,” *Nagios*. https://www.nagios.org/documentation/

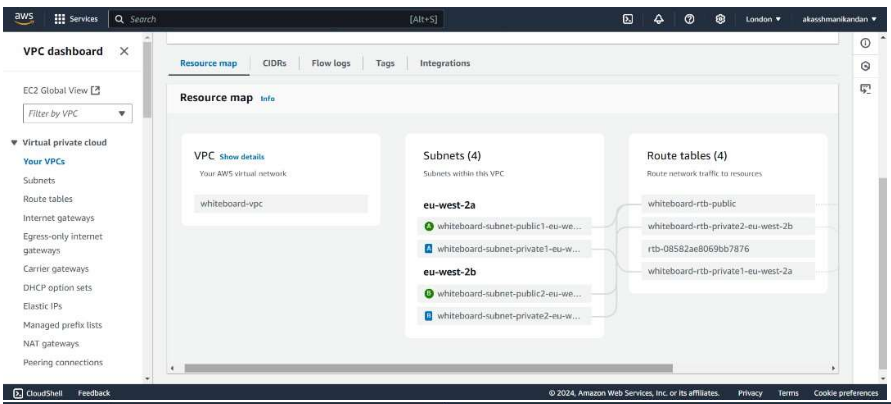
14. P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” NIST, Sep. 2011. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

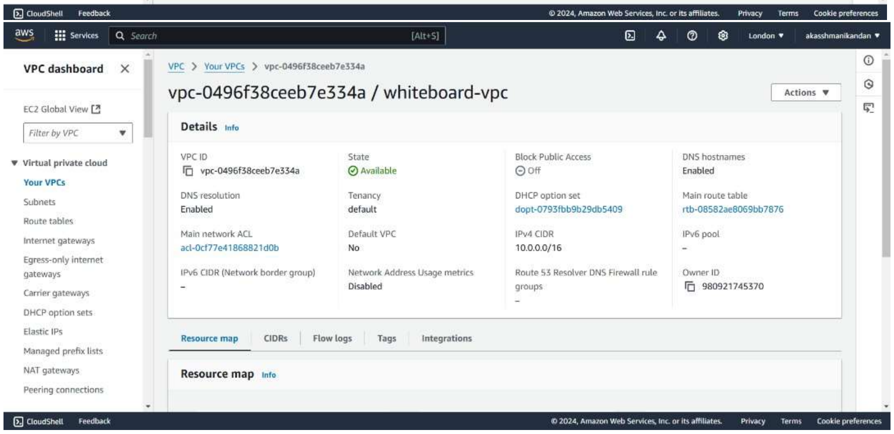
15. “Ganglia - scalable distributed monitoring system - LinuxLinks,” *LinuxLinks*, Sep. 15, 2023. https://www.linuxlinks.com/ganglia/ (accessed Dec. 12, 2024).

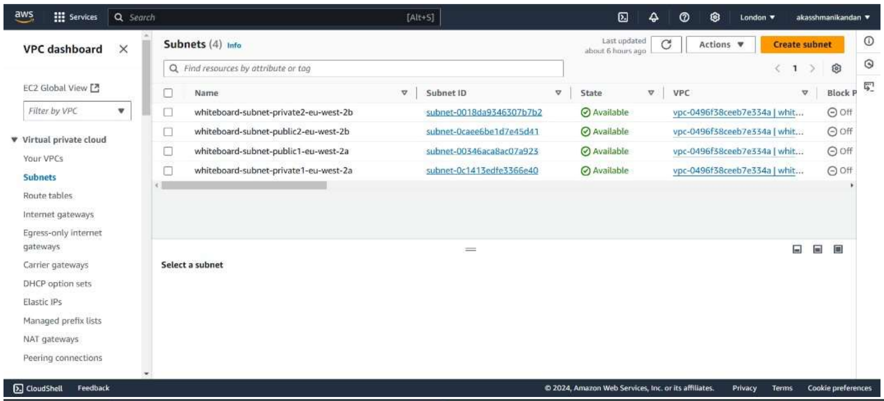
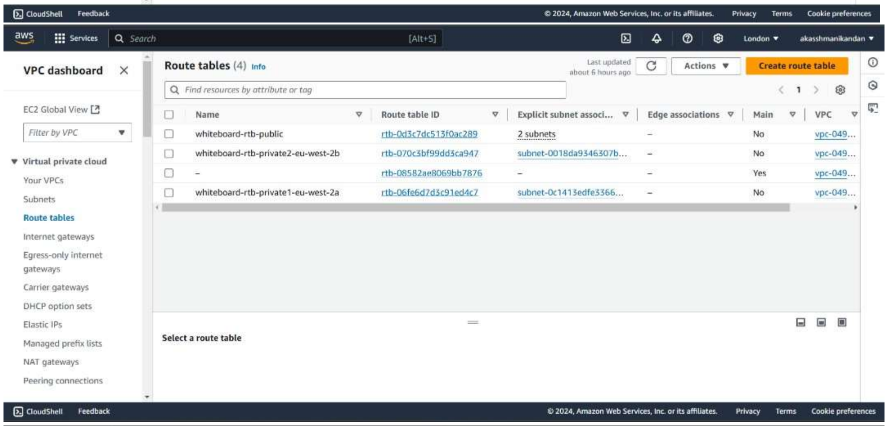
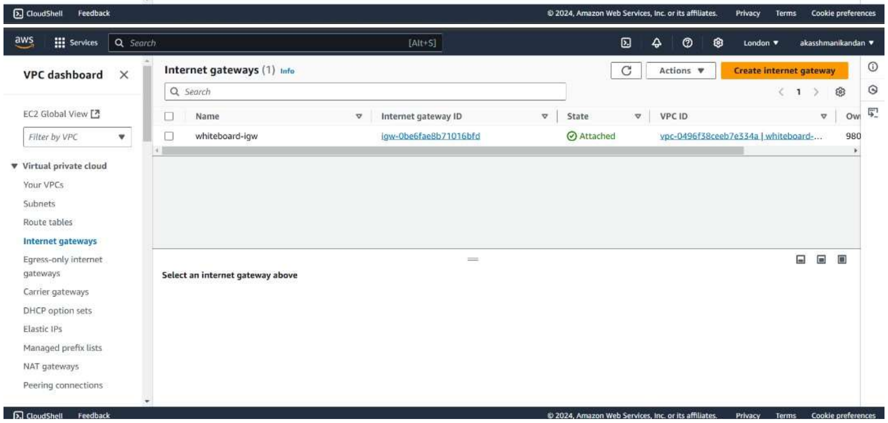
APPENDIX

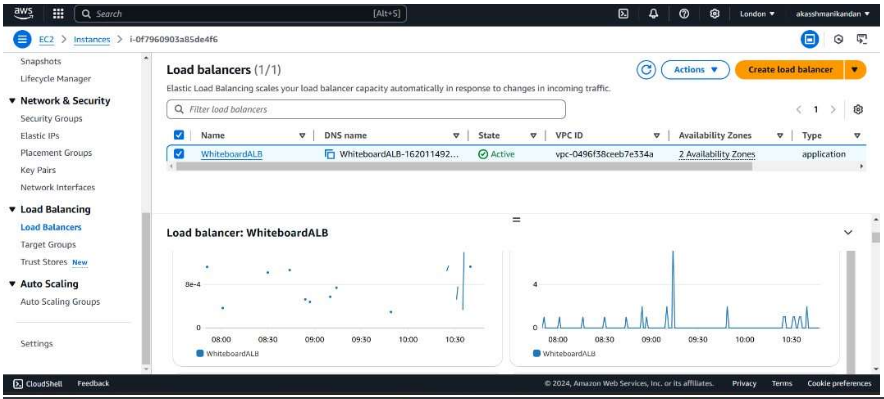
Github Source Code of the whiteboard application :- https://github.com/codewarnab/paint-web-app

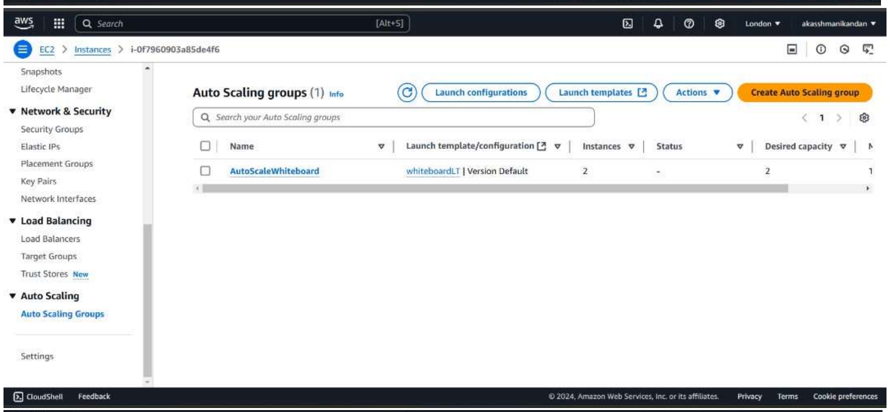
Screenshots of the Cloud Service:

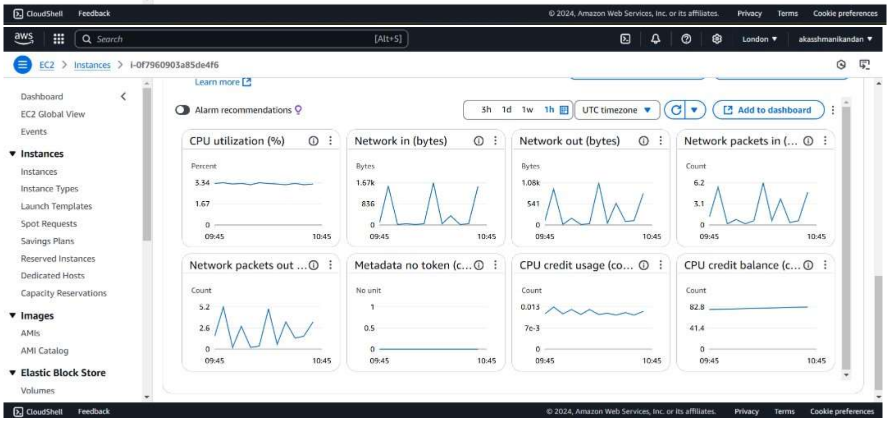


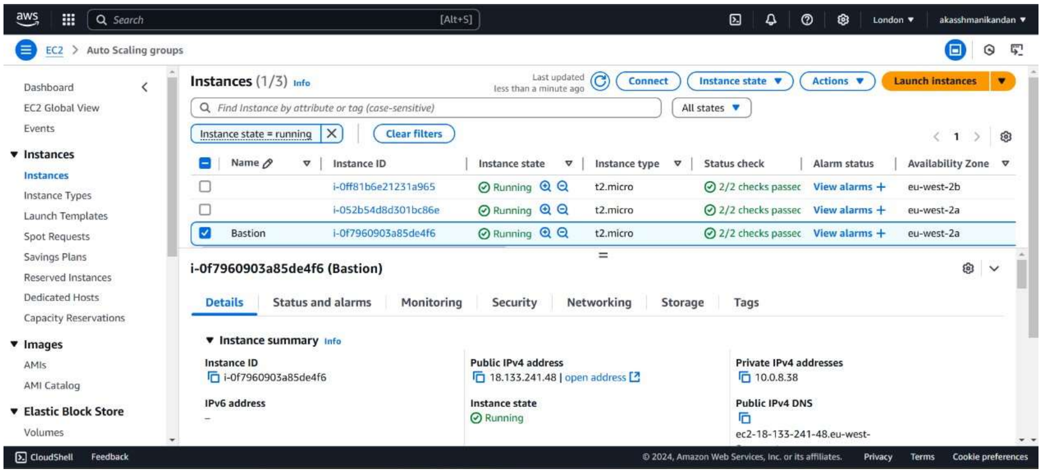


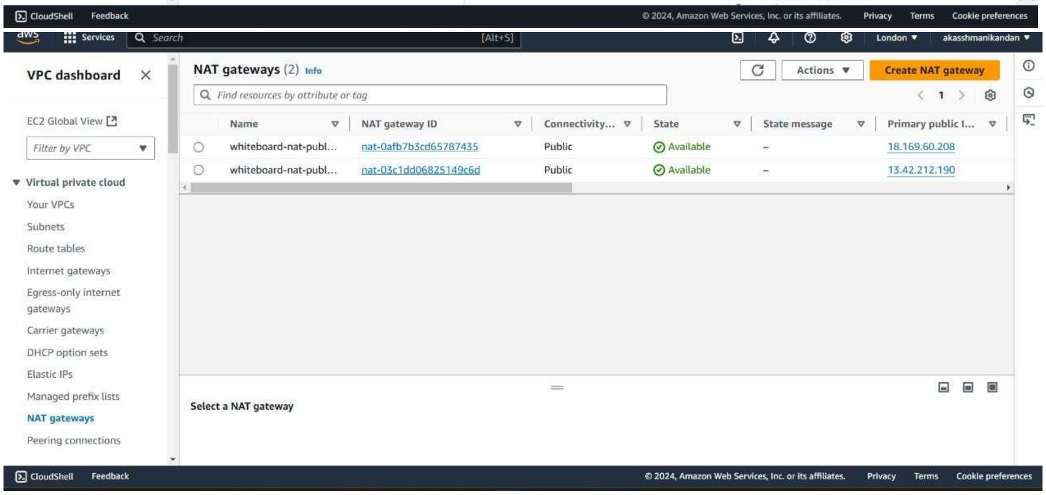

Services
Search [Alt+S]
London
akasshmanikandan
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Subnets (4) Info
Last updated about 6 hours ago
Actions
Create subnet
Find resources by attribute or tag
Name
Subnet ID
State
VPC
Block P
whiteboard-subnet-private2-eu-west-2b
subnet-0018da9346307b7b2
Available
vpc-0496f38ceeb7e334a | whit...
Off
whiteboard-subnet-public2-eu-west-2b
subnet-0caee6be1d7e45d41
Available
vpc-0496f38ceeb7e334a | whit...
Off
whiteboard-subnet-public1-eu-west-2a
subnet-00346aca8ac07a923
Available
vpc-0496f38ceeb7e334a | whit...
Off
whiteboard-subnet-private1-eu-west-2a
subnet-0c1413edfe3366e40
Available
vpc-0496f38ceeb7e334a | whit...
Off
Select a subnet
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Services
Search [Alt+S]
London
akasshmanikandan
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route tables (4) Info
Last updated about 6 hours ago
Actions
Create route table
Find resources by attribute or tag
Name
Route table ID
Explicit subnet associ...
Edge associations
Main
VPC
whiteboard-rtb-public
rtb-0d3c7dc513f0ac289
2 subnets
–
No
vpc-049...
whiteboard-rtb-private2-eu-west-2b
rtb-070c3bf99dd3ca947
subnet-0018da9346307b...
–
No
vpc-049...
–
rtb-08582ae8069bb7876
–
–
Yes
vpc-049...
whiteboard-rtb-private1-eu-west-2a
rtb-06fe6d7d3c91ed4c7
subnet-0c1413edfe3366...
–
No
vpc-049...
Select a route table
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Services
Search [Alt+S]
London
akasshmanikandan
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Internet gateways (1) Info
Actions
Create internet gateway
Search
Name
Internet gateway ID
State
VPC ID
Ow
whiteboard-igw
igw-0be6fae8b71016bfd
Attached
vpc-0496f38ceeb7e334a | whiteboard-...
980
Select an internet gateway above
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

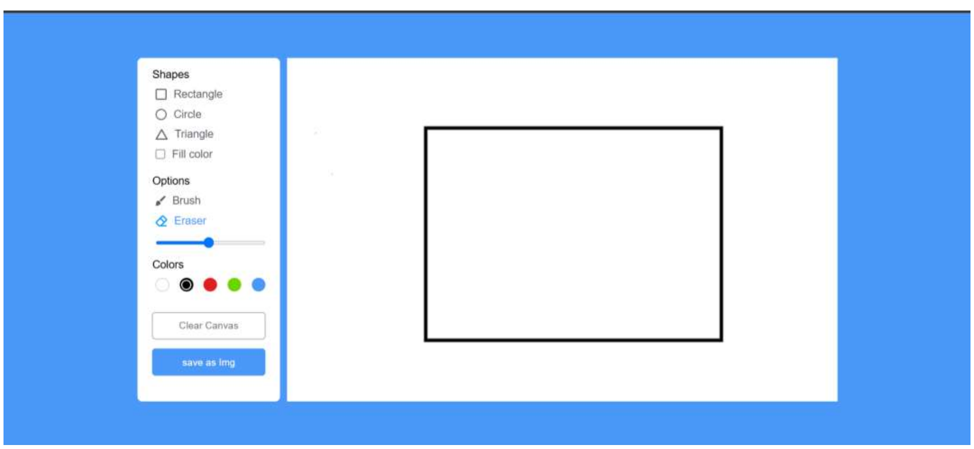

Load balancers (1/1)
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Name | DNS name | State | VPC ID | Availability Zones | Type
WhiteboardALB | WhiteboardALB-162011492... | Active | vpc-0496f38ceeb7e334a | 2 Availability Zones | application
Load balancer: WhiteboardALB
WhiteboardALB


Auto Scaling groups (1)
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Name | Launch template/configuration | Instances | Status | Desired capacity
AutoScaleWhiteboard | whiteboardLT | Version Default | 2 | - | 2

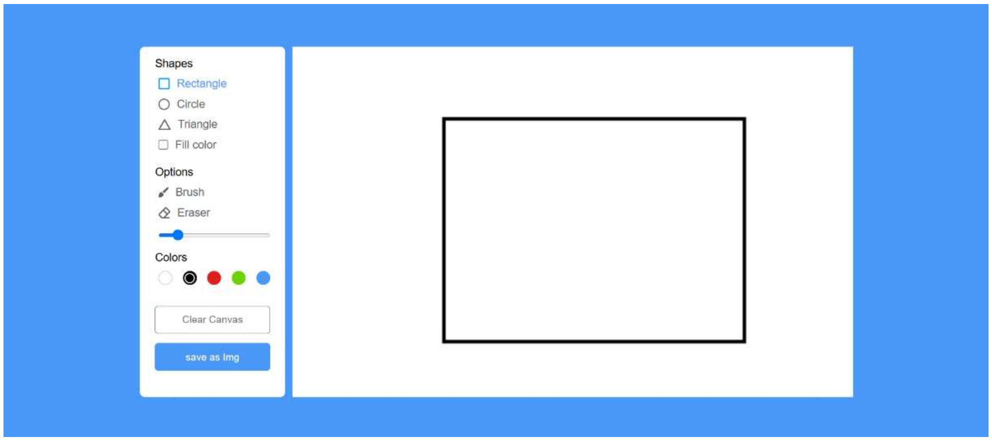

Alarm recommendations
Add to dashboard
CPU utilization (%)
Network in (bytes)
Network out (bytes)
Network packets in (...
Network packets out ...
Metadata no token (c...
CPU credit usage (co...
CPU credit balance (c...


EC2 > Auto Scaling groups
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
Instances (1/3) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone
i-0ff81b6e21231a965 | Running | t2.micro | 2/2 checks passed | View alarms | eu-west-2b
i-052b54d8d301bc86e | Running | t2.micro | 2/2 checks passed | View alarms | eu-west-2a
Bastion | i-0f7960903a85de4f6 | Running | t2.micro | 2/2 checks passed | View alarms | eu-west-2a
i-0f7960903a85de4f6 (Bastion)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0f7960903a85de4f6
Public IPv4 address
18.133.241.48 | open address
Private IPv4 addresses
10.0.8.38
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-18-133-241-48.eu-west-
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
NAT gateways (2) Info
Actions
Create NAT gateway
Find resources by attribute or tag
Name | NAT gateway ID | Connectivity... | State | State message | Primary public I...
whiteboard-nat-publ... | nat-0afb7b3cd65787435 | Public | Available | – | 18.169.60.208
whiteboard-nat-publ... | nat-03c1dd06825149c6d | Public | Available | – | 13.42.212.190
Select a NAT gateway
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Shapes
Rectangle
Circle
Triangle
Fill color
Options
Brush
Eraser
Colors
Clear Canvas
save as Img

SERVER 1


Shapes
Rectangle
Circle
Triangle
Fill color
Options
Brush
Eraser
Colors
Clear Canvas
save as Img

SERVER 2