DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

Assignment 2

In
The Class of

CSCI 5411: ADVANCED CLOUD ARCHITECTING

by

Shreya Kapoor (B00957587)

Table of Contents

Ques 1. Evaluate the requirements of the application and determine the type and size of EC2 instances that would be most appropriate for your needs. Consider factors like the application's CPU, memory, storage, and network performance requirements.?

Based on the requirements outlined, a combination of Compute-Optimized Instances (C-Series) and network optimization stands out as the most appropriate approach to achieve high overall performance for Spacetech Galactic's Holodeck application.

The C5 series of compute-optimized instances provides advanced networking capabilities that are crucial for maintaining low-latency connections, particularly important during multiplayer interactions within Spacetech Galactic's Holodeck application. These instances can handle up to 25 Gbps of network capacity, ensuring fast and efficient data transfer across different geographic locations. Built on the AWS Nitro system, they optimize overall performance, enhance availability, and bolster security by offloading traditional virtualization tasks to dedicated hardware and software components. This setup not only supports high-speed data processing but also maintains a robust infrastructure for seamless user experiences across the platform [1].

Factors Considerations:

- **CPU and Memory:**
  Considering starting with c5.large or c5.xlarge instances, providing adequate CPU (2 to 4 vCPUs) and memory (4 to 8 GiB) for moderate workloads. Scale up to larger sizes like c5.2xlarge or c5.4xlarge as needed, offering increased CPU power (8 to 16 vCPUs) and memory (16 to 32 GiB) for more demanding tasks.

- **Storage and Networking**:
  Utilizing SSD-backed volumes (gp3 or io2) for high-performance storage, scaling based on application data requirements and leveraging the C5 series' enhanced networking capabilities to maintain high throughput and low latency connections crucial for multiplayer interactions.

- **Auto Scaling:**
  Automatically adjust instance counts based on real-time demand and predictable traffic patterns (e.g., weekends, holidays) to optimize resource utilization.

Ques 2. Implement a Multi-AZ and Multi-Region deployment to ensure high availability and fault tolerance?

To ensure fault tolerance and high availability for Spacetech Galactic's Holodeck application, deploying across multiple Availability Zones (AZs) and regions is crucial.

Multi-AZ Deployment

Deploying across multiple Availability Zones within the same AWS region ensures redundancy and isolates failures. This setup minimizes downtime and sustains continuous operation even if one AZ encounters issues [2]. EBS volumes are automatically replicated within its Availability Zone. EFS standard on the other hand store data redundantly across multiple geographically within the region.

Multi-Region Deployment
Extending the application deployment across multiple AWS regions mitigates the impact of regional outages and enhances global user experience with minimal latency. Standardizing deployment using Amazon Machine Image (AMI) facilitates consistent setup across regions or AZs, enabling swift recovery and deployment uniformity [3]. Implementing a Load Balancer distributes traffic efficiently across instances in different AZs or regions, optimizing application availability and fault tolerance worldwide. These strategies collectively bolster the application's resilience, supporting uninterrupted user experiences for virtual space travel and immersive simulations.

Ques 3: Plan a strategy for data storage using Amazon EBS and/or EFS.?

Amazon EFS (Elastic File System) is the optimal choice for Spacetech Galactic’s Holodeck application primarily due to its ability to seamlessly scale with workload demands while offering shared access across multiple AWS services. EFS is well-suited for applications like Holodeck that require dynamic storage capacity, collaborative access, and high availability. Its automatic scaling ensures that storage capacity adjusts in real-time, accommodating fluctuating data needs without manual intervention. This capability is crucial for maintaining performance during peak usage periods when running intensive tasks such as real-time physics simulations or rendering high-definition graphics. It can also used for running shared volumes, or for big data analysis as the requirement includes generating analytical reports and predictive models [4].

Key Advantages of Amazon EFS:

- **Scalability**:
  Automatically scales storage capacity up or down based on demand, supporting fluctuating workloads and sudden spikes in data access requirements.

- **Shared Access**:
  Enables simultaneous access to shared file systems across multiple EC2 instances and AWS services via NFS, facilitating collaboration and data sharing among distributed teams.

- **Elastic Storage Capacity**:
  Provides elastic storage that grows and shrinks automatically as data is added or removed, ensuring efficient resource utilization and cost-effectiveness.

- **Integration with Serverless Architectures**:
  Compatible with AWS Lambda and other serverless computing services, allowing

applications to access shared data directly from EFS without managing infrastructure.

- **Managed Service:**
  Fully managed by AWS, handling maintenance tasks such as patching and updates automatically to ensure high availability and reliability.

- **Cost-Effectiveness**:
  Utilizes a pay-as-you-go pricing model based on actual usage, eliminating upfront costs and aligning expenses with operational needs [5].

Ques 4. Incorporate spot instances and reserved instances in your architecture.

Incorporating spot and reserved instances into Spacetech Galactic's architecture plays a crucial role in balancing cost efficiency and performance across its diverse workload needs:

Spot Instances:
Spot instances are highly cost-effective for tasks that can tolerate interruptions and have flexible scheduling needs, such as generating analytical reports and running predictive models during off-peak hours. They offer significant cost savings compared to On-Demand instances, which is crucial for managing operational expenses efficiently. Spot instances can dynamically adjust to workload fluctuations by utilizing features like hibernation or saving application states, ensuring minimal disruption and maximizing resource utilization. Automated tools like AWS Auto Scaling and Spot Fleet provide additional flexibility by intelligently managing spot instance fleets based on workload demand patterns, thereby optimizing cost-effectiveness without compromising application performance [6].

Reserved Instances:
Reserved instances are particularly advantageous for workloads with predictable usage patterns, such as the physics engine simulations within Spacetech Galactic's Holodeck application. By committing to reserved instances, the company secures cost predictability over a specified term, leading to substantial savings compared to paying full On-Demand prices. Reserved instances offer flexibility through different purchasing options—Standard, Convertible, and Scheduled—allowing Spacetech Galactic to tailor instance types to specific workload requirements. Continuous monitoring and optimization using AWS tools like Cost Explorer and Trusted Advisor ensures that reserved instances are effectively utilized, maximizing cost efficiency while maintaining the necessary performance levels for their virtual space experiences [7].

By strategically integrating both spot and reserved instances into their architecture, Spacetech Galactic not only enhances cost management but also strengthens the scalability, reliability, and performance of their Holodeck application. This approach underscores their commitment to delivering immersive and uninterrupted user experiences while effectively managing operational expenditures in a dynamic cloud environment.

Ques 5. Explain how you would use Amazon Machine Images (AMIs) to quickly deploy and replicate your application.?

To quickly deploy and replicate our application, Amazon Machine Images (AMIs) provide a powerful toolset. AMIs allow us to capture snapshots of our instances, enabling swift provisioning and duplication across different AWS regions or availability zones. Here's how we can effectively use AMIs:

1. **Create a Base AMI**:
   Start by creating a foundational AMI that includes the necessary operating system, essential software dependencies, and initial configurations tailored to our application’s requirements. Customize this base AMI by adding any additional software packages or libraries needed.

2. **Customize the AMI:**
   Once the base AMI is ready, customize it further by integrating our application code, configuration files, and specific dependencies. Adjust settings like network configurations, security measures, and environment variables as per our application’s needs.

3. **Replicate the AMI**:
   Duplicate the customized AMI across multiple AWS regions or availability zones using AWS tools like the Management Console, CLI, or SDKs. This ensures redundancy and fault tolerance for our application across different geographic locations.

4. **Launch Instances from the AMI:**
   Deploy the application by launching instances directly from the replicated AMIs in desired regions or availability zones. Utilize AWS Auto Scaling to manage instance scaling dynamically based on workload fluctuations.

5. **Update and Version Control**:
   Keep the AMI up to date by incorporating the latest application versions, configurations, and dependencies. Maintain version control to ensure consistency across deployments and facilitate easy rollbacks if necessary.

6. **Automate Deployment**:
   Streamline the deployment process using automation tools that define infrastructure parameters including AMIs, networking setups, and security configurations. Automation ensures efficient and consistent deployment while improving reliability and scalability [8].

The below diagram showcase the lifecycle of AMI

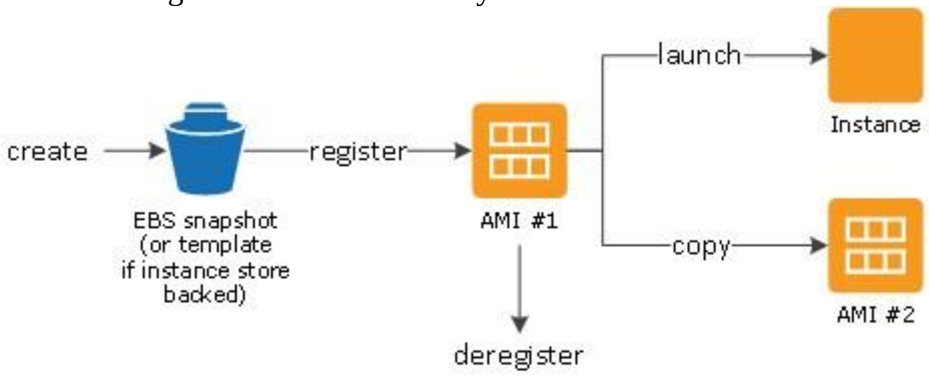

Figure 1: Lifecycle of AMI [8]

Ques 6. Provide a rough estimate of the costs of running this infrastructure and discuss the strategies you would use to manage these costs.?

Mentioned below are the rough Cost Estimate for a Single C5 Instance:

1. **EC2 Instance (C5 Series)**
   For primary compute tasks in the Spacetech Galactic architecture, including handling the physics simulation engine and running the Holodeck application we will be using C5 instance of EC2 with dedicated host to support strict licensing requirement.
   Monthly Cost: 1 instance * 24 hours/day * 30 days/month * $0.34/hour = ~$244.80.

2. **EC2 Spot instances**
   Utilized for non-critical workloads such as batch processing, generating analytical reports, and running predictive models. Spot instances help reduce costs significantly for tasks that can tolerate interruptions.
   Monthly Cost: 1 instance * 24 hours/day * 30 days/month * $0.102/hour = ~$73.44.

3. **EFS Storage**
   a. **Standard Storage:**
      Provides scalable and elastic file storage for the Holodeck application data, including 3D models, user data, and other static content. Ensures data is accessible from multiple instances.
      Monthly Cost: 10,000 GB * $0.30/GB = $3,000**.**
   b. **Replication Costs:**
      EFS cross-region replication ensures high availability and fault tolerance by replicating data across multiple AWS regions, protecting against regional failures.
      Monthly Cost: 5,000 GB * $0.08/GB = $400**.**

4. **Network Costs**

Data transfer between Availability Zones (AZs) to support low-latency, high-throughput communication required for multiplayer interactions and real-time data processing.
Monthly Cost: 10,000 GB * $0.01/GB = $100

5. **CloudWatch for Monitoring:**
   Provides monitoring and observability for the entire infrastructure, helping to track performance metrics, set alarms, and ensure the health and efficiency of the application.
   Monthly Cost: ~$100.

6. **AWS Backup:**
   Ensures that critical data stored in EFS and other resources is regularly backed up and protected against data loss.
   Monthly Cost: ~$200 [9].

Combining the costs for the entire architecture -
EC2 Instance: ~$244.80.
Spot Instance: ~$73.44.
EFS Storage: ~$3,000.
EFS Replication: ~$400.
Network Costs: ~$100.
CloudWatch and Backup: ~$300.
Total: ~$4,118.24 USD.

To efficiently manage and optimize costs in the Spacetech Galactic AWS infrastructure, here are some key strategies:

1. **Reserved Instances (RIs)**
   Reserved Instances (RIs) are a cost-effective solution for applications with consistent and predictable workloads. They offer significant savings (up to 75%) compared to On-Demand pricing. For Spacetech Galactic, purchasing RIs for the physics simulation engine, which has steady usage, will lock in lower rates and provide substantial cost savings. Analyzing workload patterns helps identify which instances would benefit most from Ris [7].

2. **Spot Instances**
   Spot Instances are perfect for non-critical, flexible workloads that can tolerate interruptions. These instances are ideal for tasks like batch processing, data analysis, and generating analytical reports. By using Spot Instances, Spacetech Galactic can reduce costs by up to 90% compared to On-Demand instances. Automation tools like AWS Auto Scaling or Spot Fleet can manage Spot Instance capacity and handle interruptions smoothly, ensuring non-critical tasks are cost-effective [6].

3. **Storage Optimization**
   Optimizing storage usage is crucial for managing costs associated with Amazon EFS, which can become expensive with large storage requirements. Implementing data lifecycle policies to move infrequently accessed data to lower-cost storage

classes can significantly reduce costs. Regularly reviewing storage utilization and cleaning up unused data will help maintain cost efficiency. Additionally, data compression techniques can further decrease storage consumption.

4. **Auto Scaling**
   Auto Scaling adjusts the number of EC2 instances based on demand, ensuring a balance between cost and performance. By configuring Auto Scaling groups, Spacetech Galactic can automatically scale instances in and out according to traffic patterns. This dynamic adjustment ensures instances are only running when needed, reducing costs during low-demand periods. This strategy is essential for maintaining an optimal balance between resource usage and cost.

5. **AWS Cost Explorer and Budgets**
   AWS Cost Explorer and AWS Budgets are powerful tools for visualizing, analyzing, and managing AWS costs. Cost Explorer helps analyze spending patterns and identify cost-saving opportunities, while AWS Budgets allow for the creation of custom cost and usage budgets. Setting up alerts for when usage exceeds predefined thresholds helps Spacetech Galactic proactively manage and control costs, ensuring they stay within budget and avoid unexpected expenses.

Ques 7. Design an architecture where the application has a need for low-latency, high throughput communication between instances.

To ensure fast and efficient communication between instances with low latency and high throughput, AWS EC2 Placement Groups are an ideal solution. Here’s a step-by-step approach to designing this architecture:

1. **Create EC2 Placement Group:**
   Begin by establishing an EC2 cluster placement group. This allows precise control over where instances are physically located to optimize their performance.

2. **Launch Instances in Placement Group:**
   Deploy instances within the placement group ensuring they are similar in configuration and have optimized network settings for effective communication.

3. **Enable Enhanced Networking:**
   Activate Enhanced Networking on the instances within the Placement Group. This feature uses advanced network interfaces and drivers to boost data transfer speeds and reduce delays.

4. **Configure Security Group Rules:**
   Set up necessary security group rules to enable seamless communication between instances within the placement group while ensuring data security.

5. **Optimize Application Design:**
   Refine the application architecture to fully utilize the low-latency and high throughput capabilities provided by the Placement Group. This includes optimizing data transfer protocols and utilizing parallel processing techniques.

6. **Monitor and Optimize Performance:**
   Continuously monitor instance performance within the Placement Group using AWS CloudWatch and other tools. Identify performance bottlenecks and implement optimizations to further enhance overall system efficiency.

By leveraging EC2 Placement Groups effectively, configuring enhanced networking, optimizing application design, and monitoring performance closely, you can create an architecture that facilitates rapid and reliable communication between instances. This setup is particularly beneficial for applications requiring real-time data processing, distributed computing, or intensive computing tasks where speed and efficiency are crucial [10].

Ques 8. Design a scenario where temporary, high-IOPS storage is required and an instance store would be used.

The architecture is designed to support a data processing application that handles extensive user data, generates analytical reports, and runs predictive models. It relies on intelligent systems for operational management and performs intensive computations on large datasets, requiring low-latency and high IOPS (Input/Output Operations Per Second) for optimal performance.

To meet these requirements, leveraging an instance store proves advantageous due to its ability to provide temporary storage with high IOPS. Here's how the architecture can be effectively structured:

1. **EC2 Instances with Instance Store:**
   Deploy EC2 instances equipped with instance stores. These local storage solutions are directly attached to the EC2 instance hardware, offering low-latency performance and high IOPS. They serve as ideal temporary storage for data processing tasks that demand high performance [11].

2. **Application Deployment Strategy:**
   Deploy the data processing application on these EC2 instances. Design the application to utilize the instance store for temporary storage during intermediate data processing stages. This approach maximizes the application's performance by capitalizing on the instance store's high IOPS capabilities.

3. **Optimized Data Processing Workflow:**
   Structure the data processing workflow to fully leverage the instance store's benefits. Store large datasets within the instance store to ensure rapid access and processing. Execute computationally intensive tasks and intermediate data processing directly within the instance store environment to harness its high performance.

4. **Data Persistence Strategies:**
   Since instance store storage is temporary, implement mechanisms for data

persistence. Establish scheduled data transfer processes from the instance store to durable storage solutions like Amazon S3 or Amazon EFS to ensure data durability and availability beyond the lifecycle of EC2 instances.

5. **Monitoring and Scalability Measures:** Utilize AWS CloudWatch or similar tools to monitor instance store utilization, IOPS metrics, and overall system performance. Based on workload demands and resource usage patterns, dynamically adjust the number or type of EC2 instances to maintain optimal performance and storage efficiency.

Ques 9. Plan for scenarios where the application has to comply with strict licensing terms (BYOL) or meet dedicated hardware requirements.

Spacetech has highlighted the strict licensing requirements for the physics simulation engine, the core of the Holodeck experience. These requirements demand the use of dedicated hardware, adding complexity to the architectural design. Amazon EC2 Dedicated Hosts provide a solution by allowing you to use your own software licenses from vendors like Microsoft and Oracle on Amazon EC2. This setup combines the flexibility and cost-effectiveness of using your own licenses with the resilience, simplicity, and elasticity of AWS. An EC2 Dedicated Host is a physical server fully dedicated to your use, helping to meet corporate compliance needs.

Additionally, AWS License Manager integrates with Dedicated Hosts to manage software licenses, including those for Microsoft Windows Server and Microsoft SQL Server. License Manager lets you define licensing terms and preferences for host allocation and capacity utilization. Once configured, AWS handles these administrative tasks, enabling you to launch virtual machines on Dedicated Hosts as easily as you would with AWS-provided licenses [12].

To address scenarios requiring strict licensing terms or dedicated hardware, the following plan can be implemented:

- **Understand Licensing Requirements:**
  Clearly understand the licensing terms and restrictions for the application, ensuring compliance with dedicated hardware needs.

- **Evaluate Dedicated Hosts:**
  Use EC2 Dedicated Hosts to ensure compliance with licensing terms that mandate dedicated resources. Choose appropriate instance types and configurations to meet specific hardware requirements.

- **Consider Dedicated Instances:**
  If Dedicated Hosts are not mandatory, use Dedicated Instances for isolated host-level compliance, leveraging the benefits of cloud infrastructure.

- **Verify Hardware Compliance:**
  Ensure selected Dedicated Hosts or Instances meet the application's licensing and

hardware compliance requirements.

- **Plan Capacity and Scaling:**
  Assess workload demands and plan capacity and scalability for dedicated resources to handle expected spikes and growth.

- **Configure Network and Security:**
  Set up network and security settings to align with the application's needs, including network access controls, security groups, and encryption measures.

- **Regular Compliance Audits:**
  Conduct periodic audits to ensure ongoing compliance with licensing terms and hardware requirements. Stay updated on licensing agreements and adjust the architecture as needed.

Ques 10. Describe how you would use EC2 metadata and user data to handle configuration tasks and pass information to instances at launch time.

Using EC2 metadata and user data offers powerful features for managing configuration tasks and passing information to instances when they launch.

1. **EC2 Metadata**

   a. **Accessing Metadata:**
   EC2 instances can get metadata by making HTTP requests to a special URL (http://169.254.169.254/latest/meta-data/). This metadata includes details like the instance ID, availability zone, network configuration, and security groups.

   b. **Dynamic Configuration:**
   Use metadata to adjust settings specific to each instance. For example, you can use the availability zone information to change settings based on the instance’s location or use the instance ID for better integration within your system.

   c. **Automation and Scripting:**
   Create scripts or use tools that use metadata to automate important tasks. For example, you can set up network configurations, mount storage volumes, or run custom actions during the instance’s startup [13].

2. **User Data**

   a. **Launch-time Customization:**
   Use user data to provide scripts or commands that run when the instance starts. This lets you automatically set up configurations or actions as the instance launches.

b. **Streamlined Software Installation:**
Use user data to install and configure software automatically. Include commands in the user data script to download and install necessary software, set up application settings, and perform initial setup tasks.

c. **Environment-specific Configurations:**
Use user data to pass specific configurations to customize instances based on their roles. This can include setting environment variables, configuring application settings, or defining runtime options to fit the instance's purpose.

c. **Instance Personalization:**
Use user data to personalize instances for their intended roles. For example, you can install monitoring tools, configure the instance as a web server or database server, or make other customizations to suit specific needs [14].

By leveraging the dynamic nature of EC2 metadata and the flexibility of user data, we can efficiently manage configuration tasks and pass crucial information to instances during their launch. This approach allows us to automate deployments, streamline software installations, customize instances, and ensure efficient post-launch activities.

Ques 11. Describe how to use tools like AWS Compute Optimizer and Trusted Advisor for identifying optimal EC2 instance types and for maintaining cost efficiency.

To achieve optimal performance and cost efficiency in an EC2 environment, tools like AWS Compute Optimizer and Trusted Advisor are invaluable. Following comprehensive approach on how to leverage these tools:

1. **AWS Compute Optimizer**

   a. Enable Compute Optimizer:
   Start by enabling AWS Compute Optimizer in your AWS account. This service analyzes your resource utilization and provides tailored recommendations for the best EC2 instance types.

   b. Analyze Recommendations:
   Regularly review the recommendations generated by Compute Optimizer. These suggestions are based on a detailed analysis of your historical usage patterns and specific performance needs.

   c. Implement Recommendations:
   Assess the recommendations carefully and make informed decisions about changing instance types. Consider factors such as CPU capacity, memory requirements, storage options, and network capabilities. Implement the changes by resizing your current EC2 instances as recommended.

   d. Monitor Performance:

After making adjustments, continuously monitor the performance of the new EC2 instances to ensure they meet your application's performance expectations [15].

2. **AWS Trusted Advisor**

   a. **Activate Trusted Advisor:**
   Enable Trusted Advisor in your AWS account to access its wide range of best practice recommendations, which cover cost optimization, performance, security, and fault tolerance.

   b. **Access Cost Optimization Recommendations:**
   Use the cost optimization section of Trusted Advisor to find potential cost-saving opportunities in your EC2 environment. Recommendations may include right-sizing instances, eliminating idle resources, and optimizing purchasing options such as Reserved Instances or Spot Instances.

   c. **Review Recommendations:**
   Carefully review the cost optimization recommendations and prioritize those that align with your cost reduction goals. Consider how each recommendation will affect performance, business needs, and budget constraints.

   d. **Implement Recommendations:**
   Implement the relevant cost optimization recommendations, which may involve resizing instances, optimizing storage options, or adjusting purchasing models to achieve a balance between performance and cost.

   e. **Regular Monitoring and Review:**
   Continuously monitor the cost optimization recommendations provided by Trusted Advisor. Regularly review your EC2 environment to identify new cost-saving opportunities and apply best practices as needed [16].

References

[1] "Amazon EC2 C5 Instances," *Amazon.com*. [Online]. Available: https://aws.amazon.com/ec2/instance-types/c5/. [Accessed: June 21, 2024].

[2] "Amazon RDS Multi-AZ," *Amazon.com*. [Online]. Available: https://aws.amazon.com/rds/features/multi-az/. [Accessed: June 21, 2024].

[3] "Copy an AMI," *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html. [Accessed: June 21, 2024].

[4] "Amazon EBS vs EFS vs S3: Picking the best AWS storage option for your business," *Missioncloud.com*. [Online]. Available: https://www.missioncloud.com/blog/resource-amazon-ebs-vs-efs-vs-s3-picking-the-best-aws-storage-option-for-your-business. [Accessed: June 21, 2024].

[5] "Amazon EFS Features," *Amazon.com*. [Online]. Available: https://aws.amazon.com/efs/features/. [Accessed: June 21, 2024].

[6] "Amazon EC2 Spot Instances," *Amazon.com*. [Online]. Available: https://aws.amazon.com/ec2/spot/. [Accessed: June 22, 2024].

[7] "Amazon EC2 Reserved Instances," *Amazon.com*. [Online]. Available: https://aws.amazon.com/ec2/pricing/reserved-instances/. [Accessed: June 22, 2024].

[8] "Amazon Machine Images (AMI)," *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html. [Accessed: June 22, 2024].

[9] "What is AWS Billing and Cost Management?," *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-what-is.html. [Accessed: June 22, 2024].

[10] "Placement Groups," *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html. [Accessed: June 22, 2024].

[11] "Amazon EC2 instance store," *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html. [Accessed: June 22, 2024].

[12] "AWS EC2 Dedicated Hosts," *Amazon.com*. [Online]. Available: https://aws.amazon.com/ec2/dedicated-hosts/. [Accessed: June 22, 2024].

[13] “Work with instance metadata,” *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html. [Accessed: June 22, 2024].

[14] “Run commands on your Amazon EC2 instance at launch,” *Amazon.com*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html. [Accessed: June 22, 2024].

[15] “AWS Cloud Optimizer,” *Amazon.com*. [Online]. Available: https://aws.amazon.com/compute-optimizer/. [Accessed: June 22, 2024].

[16] “AWS Trusted Advisor,” *Amazon.com*. [Online]. Available: https://aws.amazon.com/premiumsupport/technology/trusted-advisor/. [Accessed: June 22, 2024].