AWS Operational Guides

I. EC2 and Key Pair Management(Problem 1)

This section focuses on instance lifecycle and secure access, particularly the process of recovering a lost key.

Concept	Technical Detail / Justification	Source Reference
Key Pair Creation (CLI)	Command uses `--query "KeyMaterial" --output text` to extract only the raw private key data, which is then redirected (`>`) into the `.pem` file. This file is required for SSH connection (`ssh -i key.pem user@ip`).	
Security Group (SG) Rules	The SG (`recovery-sg`) must authorize inbound traffic for **protocol TCP** on **port 22** from the source **`0.0.0.0/0`** (meaning any IP address on the internet). *Justification:* Port 22 is the standard port for SSH, allowing remote management.	
Key Pair Recovery Mechanism	**No recovery of the original key is possible.** Recovery simulates transferring the persistent data (the disk/OS) to a new instance launched with a functional key pair.	
Recovery Steps	1. **Create an Image (AMI)** of the existing instance. 2. **Create a New Key Pair** (`recovered-key`). 3. **Launch a New Instance** from the AMI, associating it with the new key pair.	
Instance Verification	Validation uses `aws ec2 describe-instances --query "Reservations[].Instances[].State.Name"` to confirm the instance is running.	

II. VPC Networking and Isolation (Problems 2 & 3)

Concept	Technical Detail / Justification	Source Reference
VPC CIDR Block	Created using the IPv4 CIDR `10.0.0.0/16` . *Justification:* A `/16` block allows for 65,536 private IP addresses within the VPC, providing ample space for subnets.	
Subnet Allocation	Four subnets are created, using smaller `/24` blocks (256 addresses each, e.g., `10.0.1.0/24`).	
Availability Zone (AZ) Requirement	Subnets are explicitly split across two AZs (`us-east-1a` and `us-east-1b`). *Justification:* This design is mandatory for **fault tolerance** and high availability for components like Load Balancers and Auto Scaling Groups.	
Internet Gateway (IGW)	Must be created (`aws ec2 create-internet-gateway`) and then **attached** to the VPC (`aws ec2 attach-internet-gateway --vpc-id <vpc-id>`). *Justification:* The IGW provides the necessary link for public subnets to communicate with the internet.	

Concept	Technical Detail / Justification	Source Reference
Public Route Table Configuration	The public route table (`public-rt`) must contain a route where the **Destination is** `0.0.0.0/0` (all traffic outside the VPC) and the **Target is the IGW ID**.	
Private Route Table Principle	The private route table (`private-rt`) is created but **explicitly does NOT add the IGW route**. *Justification:* This ensures that resources associated with the private subnet are isolated and cannot be reached directly from the internet.	
Subnet Association (CLI)	The configuration requires using `aws ec2 associate-route-table` for each public subnet ID and the public route table ID (`<public-rt-id>`).	

III. IAM and Security Policy Management (Problems 4 & 5)

Concept	Technical Detail / Justification	Source Reference
User (`example-user` **)**	Granted highly specific or full access policies directly (e.g., `AmazonEC2FullAccess` , `AmazonS3FullAccess`).	
User Group (`example-group` **)**	Used to assign common, baseline permissions (e.g., `ReadOnlyAccess`) to multiple users efficiently. *Justification:* Supports the **Principle of Least Privilege** by providing default, limited access.	
Custom Policy (`describe-ec2-only` **)**	Defined using a JSON document. The `Effect` is `Allow` , the `Action` is restrictive (`ec2:Describe*`), and the `Resource` is global (`*`). *Justification:* Grants the ability to view EC2 resources but not modify them.	
IAM Role (`ec2-s3-role` **)**	Assigned to the **AWS service EC2**. Requires a **trust policy** (`trust.json`). *Trust Policy Detail:* Must allow the Principal (`Service: ec2.amazonaws.com`) to perform the `sts:AssumeRole` action. *Justification:* Allows EC2 instances to assume the role's permissions (e.g., `AmazonS3ReadOnlyAccess`) without needing static access keys.	
Inline Policy (`allow-start-instances` **)**	A policy embedded directly into the user (`example-user`). CLI command is `aws iam put-user-policy` . *Justification:* This policy is logically part of the user and is deleted if the user is deleted, making it ideal for unique, non-sharable permissions.	
MFA Configuration	Required security layer. Configured via the Console by associating a virtual device (Authenticator App) and submitting two consecutive OTPs (One-Time Passwords).	

IV. S3 Storage Operations (Problems 6 & 9)

Concept	Technical Detail / Justification	Source Reference
Multipart Upload	Demonstrated using both CLI (`aws s3 cp`) and Console. *Mechanism:* AWS automatically divides large files (over 5MB or	

Concept	Technical Detail / Justification	Source Reference
	100MB in the example setup) into parts, uploads them in parallel, and reassembles them.	
Versioning (Purpose)	When enabled (`Status=Enabled`), S3 keeps track of multiple versions of an object, protecting against accidental overwrites or deletions.	
Versioning (CLI Syntax)	Requires the use of `aws s3api put-bucket-versioning` command, specifying the bucket name and a `versioning-configuration` structure.	
Default Encryption (SSE-S3)	Ensures that every new object uploaded to the bucket is encrypted at rest using **Server-Side Encryption with S3-managed keys (AES256)**.	
Encryption (CLI Syntax)	Requires the `aws s3api put-bucket-encryption` command and a complex JSON structure defining the encryption rules and the **`SSEAlgorithm` as `AES256`**. *Justification:* The JSON structure ensures compliance and defines the exact method of protection.	

V. Load Balancing and Auto Scaling (Problems 7, 8, 10, 11)

This section focuses on high-availability concepts and the automated management of compute capacity.

Concept	Technical Detail / Justification	Source Reference
Load Balancer Type	**Application Load Balancer (ALB)** is used. *Justification:* ALBs operate at Layer 7 (HTTP/HTTPS) and are suited for web application traffic distribution.	
ALB Network Mapping	The ALB must be mapped to **at least two subnets** across different Availability Zones (e.g., `us-east-1a` , `us-east-1b`). *Justification:* If one AZ fails, the ALB can continue routing traffic through the other AZ.	
Target Group	Defines where traffic is sent and includes **health checks**. The targets (EC2 instances) are registered using `aws elbv2 register-targets` .	
Listener Configuration	Configured to listen on **HTTP port 80** and defines the **Default Action** as forwarding traffic to the specific Target Group (`web-tg` or `cli-tg`).	
Launch Template (LT)	Used as the blueprint for ASG. In the CLI configuration, the `UserData` (bootstrapping script) must be provided in the JSON file in **Base64 encoded format**.	
ASG Capacity Parameters	Essential definitions: `Min Size` (1), `Max Size` (3), and `Desired Capacity` (2). *Justification:* These limits define the fleet's boundary, ensuring costs are contained (`Max`) and availability is maintained (`Min`).	
Scaling Policy Type	The sources demonstrate **Simple Scaling policies**. The policies define a threshold (e.g., CPU > 70%) to trigger an adjustment.	
Scaling Adjustment	Scale-out uses `scaling-adjustment 1` . Scale-in uses `scaling-adjustment -1` . Both use `adjustment-type`	

Concept	Technical Detail / Justification	Source Reference
(CLI)	`ChangeInCapacity`. *Justification:* This tells the ASG to add or remove a fixed number of instances.	
ASG Recovery	If an instance is manually terminated or becomes unhealthy, the ASG detects the loss and automatically launches a new instance using the Launch Template to maintain the `Desired Capacity`.	

Analogy: The AWS Kitchen

- **VPC** is the **Building Structure** (defines the boundaries and space).
- **Subnets** are the **Workstations** (specific areas for prep, cooking, or storage, isolated by firewalls).
- **IAM** is the **Security System and Staff Management** (defining which chef, server, or delivery person (Role/User) can access which area or tool).
- **ALB** is the **Expediter** (Layer 7 traffic cop) who sends incoming orders (traffic) to the least busy cook.
- **ASG** is the **Automated Staffing Agency** (monitors the grill temperature (CPU utilization); if it gets too hot, it automatically hires and installs a new cook (instance) from the blueprints (Launch Template)).
- **Key Pair Recovery** is like losing the master key: you don't find the old key; you have to take the contents of the safe (AMI) and put it into a new safe (New Instance) with a new key (New Key Pair).