SECURE TRAINING PORTAL

AWS Cloud Infrastructure Design | Week 2 Practical Task

Project purpose

This document proposes a secure, scalable AWS environment for a fictional training company. The portal is intended for approximately 100 registered participants, with around 20 concurrent users. It provides a public HTTPS entry point, private participant data, protected document uploads and restricted staff administration.

Design objectives

- Publish the participant portal over HTTPS.
- Keep the database and uploaded documents private.
- Restrict administrative access to approved staff using individual identities and MFA.
- Give the application only the permissions it needs.
- Use managed AWS services and multi-AZ options where appropriate.

Proposed service selection

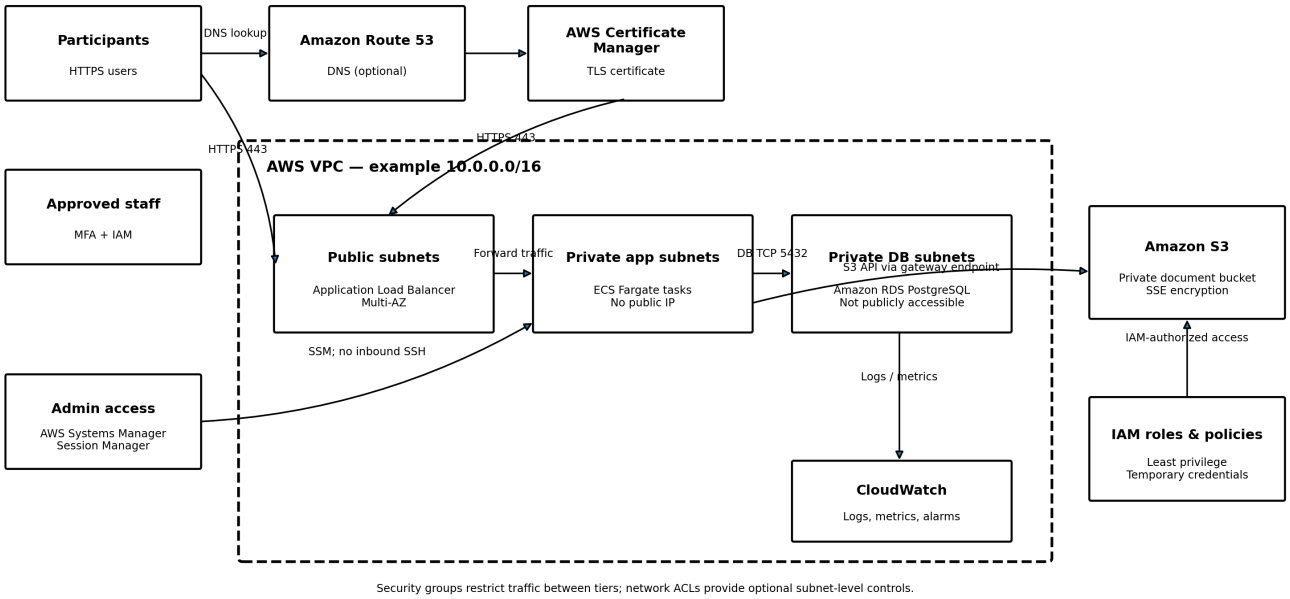
Service	Purpose	Reason and assumptions / limitations
Amazon VPC	Network boundary and subnets	Separates public entry from private application and database tiers. CIDR and subnet sizes are design assumptions.
Application Load Balancer	HTTPS entry point	Routes web requests to healthy application tasks across Availability Zones. Requires a TLS certificate.
Amazon ECS on Fargate	Run portal application	Managed container compute without managing servers. Application image and deployment pipeline are not included in this design.
Amazon RDS for PostgreSQL	Participant records	Managed relational database in private subnets. Backups and Multi-AZ depend on selected configuration and budget.
Amazon S3	Uploaded documents	Durable object storage with Block Public Access and encryption. Bucket remains private.
IAM, ACM, CloudWatch, SSM	Access, TLS, monitoring, administration	Provides least-privilege permissions, certificate management, operational visibility and managed admin sessions.

Scope: This is an architecture and security design, not a deployed application. No paid AWS resources are required to complete this assignment.

1. Proposed architecture

The diagram shows the proposed traffic flow and security boundaries. The VPC contains the public load balancer subnets and private application and database subnets. Amazon S3 is an AWS-managed service outside the VPC; the application accesses it using an S3 gateway VPC endpoint where configured.

Secure Training Portal — Proposed AWS Architecture



Request flow: Participants resolve the portal DNS name and connect to the Application Load Balancer over HTTPS (TCP 443). The load balancer forwards requests to ECS Fargate tasks in private application subnets. The application reads and writes participant records through RDS PostgreSQL on TCP 5432 and accesses approved S3 object paths using its task IAM role.

Administration and operations: Approved staff authenticate with individual IAM identities and MFA. Administrative access to supported compute resources uses Systems Manager Session Manager rather than exposing SSH/RDP to the internet. CloudWatch collects application and infrastructure logs and metrics.

Availability: Place the load balancer, application tasks and database subnets across at least two Availability Zones where the selected service configuration supports it. For a small training portal, start with modest task capacity and scale based on measured demand.

2. Network access plan

Use separate public, private application and private database subnets in the VPC. Attach an internet gateway for the public load balancer subnets. Application and database resources do not receive public IP addresses. If private application tasks need outbound internet access for updates or external APIs, use controlled egress (for example, NAT) only where required; this can add cost. Prefer VPC endpoints for supported AWS services such as S3.

Source	Destination	Port / service	Decision	Justification
Internet users	ALB security group	TCP 443 HTTPS	Allow	Public secure portal entry point.
Internet users	ALB security group	TCP 80 HTTP	Redirect only	Redirect HTTP to HTTPS; do not serve portal content over HTTP.
ALB security group	Application task security group	Application listener port (e.g. 8080)	Allow	Only load balancer traffic reaches application tasks.
Application task security group	RDS security group	TCP 5432 PostgreSQL	Allow	Database accepts connections only from the application tier.
Internet / public clients	RDS database	TCP 5432	Deny	Database is private and not publicly accessible.
Application task role	S3 document bucket	S3 API over HTTPS	Allow, scoped	IAM policy limits access to the required bucket and object prefix.
Unapproved sources	Admin interfaces	SSH/RDP inbound	Deny	No public management ports. Use approved SSM access.

Routing and boundaries: Public subnet route tables send internet-bound traffic to the internet gateway. Private application subnets have no direct inbound route from the internet; outbound access is limited to required destinations. Database subnets have no internet route. Security groups act as stateful instance-level firewalls and should reference the security group of the permitted upstream tier rather than broad public CIDR ranges.

3. Identity, application permissions and document protection

Staff access: Give each approved staff member an individual identity through IAM Identity Center (preferred for workforce access) or appropriately managed IAM identities. Require MFA, avoid shared administrator accounts, and grant permissions by job role. Use a separate, tightly controlled break-glass process only if the organization requires one.

Application access: Attach an IAM task role to ECS tasks. The application obtains short-lived credentials from the AWS task role; do not place long-lived access keys in source code, container images, environment files or Git repositories. Grant only the actions and resources needed by the application.

Example least-privilege S3 policy (replace the placeholders before use):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListOnlyTrainingDocumentsPrefix",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::REPLACE_WITH_BUCKET_NAME",
      "Condition": {
        "StringLike": {
          "s3:prefix": ["participant-documents/*"]
        }
      }
    },
    {
      "Sid": "ReadWriteOnlyTrainingDocuments",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::REPLACE_WITH_BUCKET_NAME/participant-documents/*"
    }
  ]
}
```

This sample allows listing only the specified prefix and reading or uploading objects beneath it. It does not grant public access, bucket deletion, bucket policy changes or access to other buckets. Add encryption, tagging or object-version permissions only when the application actually requires them. A separate application authorization check must confirm that the signed-in participant owns the requested document; an S3 prefix policy alone does not prove participant ownership.

Private document controls: Enable S3 Block Public Access at account and bucket level; keep Object Ownership set to bucket-owner-enforced; enable server-side encryption; deny anonymous access; and consider versioning and lifecycle rules. For downloads, the application should authenticate the participant, verify document ownership, then issue a short-lived pre-signed URL or stream the file after authorization. Never trust a participant-supplied object key without checking ownership.

Database controls: Store participant records in private RDS subnets, require encrypted connections, enable encryption at rest, automated backups and retention appropriate to the business. Keep database credentials in AWS Secrets Manager or another approved secrets manager and rotate them. Do not place credentials in application source code.

4. Verification plan, assumptions and references

The following are proposed checks. They have not been executed because this submission is a design exercise and no AWS application environment has been deployed.

Test	Expected result
1. Open portal using HTTPS	The portal responds over HTTPS with a valid certificate; HTTP redirects to HTTPS.
2. Attempt admin access as an unapproved user	Access is denied; only approved staff identities with MFA and required permissions can administer resources.
3. Attempt direct public connection to RDS	Connection fails because the database is private and its security group permits only the application tier.
4. Request an S3 document anonymously	Access is denied; bucket is private and public access is blocked.
5. Request a document as its authorized participant	Application verifies identity and ownership, then permits the authorized download.
6. Request another participant's document	Application denies access even if the requester guesses or supplies the object key.
7. Review application role permissions	Role can access only the required bucket prefix and actions; unrelated S3 resources remain inaccessible.

Assumptions and limitations

The portal serves about 100 registered participants and around 20 concurrent users. Actual CPU, memory, storage, database sizing and scaling thresholds must be confirmed through load testing. The design assumes a containerized web application can run on ECS Fargate and that the application supports authentication and participant-level authorization. Availability, backup retention, logging retention, NAT usage, data transfer and Multi-AZ choices affect cost. No cost estimate is presented because region, retention, traffic, sizing and uptime requirements have not been finalized. Use the AWS Pricing Calculator before implementation. The actual training portal has not been deployed or tested.

References

AWS VPC documentation: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html
Amazon ECS documentation: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html
Amazon RDS documentation: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html
Amazon S3 Block Public Access: https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-control-block-public-access.html
IAM best practices: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
AWS Pricing Calculator: https://calculator.aws/

Assistance acknowledgement: This design and report were prepared with AI assistance. The learner should review the architecture, verify assumptions and be prepared to explain each design decision.