2️⃣

Lecture 2 - Core services - IAM, S3 (1h)

- Q&A about previous lesson (5m)
- Couple of words about Organisations and SCPs

IAM

- know how to create a static user credentials
- users/groups/roles
- IAM policies and API calls contain
  - Allow, Deny statements
  - Conditions
  - Permissions boundaries
- IAM + AWS CLI Deep Dive
  - Creating Users/Groups
  - Policies, how they work
    - Policies eval logic
    - “Actual permissions” diagram - User-attached policies + Resource policies + Permissions boundaries = actual permissions
    - Statements

        - Allow
        - Deny
    - Resources
    - Principals
- Writing own policies
    - Policy generator
        - S3 bucket policies
        - IAM policies
- Users vs Roles
- Service Roles
- STS overview
    - federated vs regular users
    - Cross account access
    - Trusted entities
    - sts:assume role
- IAM best practices
    - never use root account for daily activities
    - use SSO, Identity providers with roles instead of leveraging users/groups
    - write explicit policies - to which resources/entities the access should be allowed
    - do not embed the AWS access keys anywhere
    - policies should make sense - better create separate policies and grant them separately, than pack all in one policy
    - follow the principle of least privilege
    - use permission boundaries
    - combine user attached policies with resource policies

        - use MFA if not enforced by the organization
    - Configuring first cli profile
        - installing aws cli v2
        - installing and enabling the autocompletion
        - aws cli api logic
    - configuring profile
    - (optional) configuring aws sso cli
        - differences between sso cli and profiles

S3 core

- what’s a bucket
- why the bucket names should be unique
- object storage vs filesystem
- uploading files
    - api
    - UI
- lifecycle rules
- bucket policies
- public/private access
- storage classes
- bucket object versioning
- s3 high level and low level api
- s3 encryption
- CORS
- object ownership
- pre-signed URLs

- Use cases
  - S3 static website serving with CloudFront
  - in a deployment process
  - as an object storage for the applications
  - backups (even of personal data)
  - data lakes
- S3 pricing
  - storage classes
  - per storage
  - per api calls
- Querying S3 text data with Athena

Q&A session

- topic discussion
- sharing useful external resources and links

Homework

Hands-on - https://aws.amazon.com/premiumsupport/knowledge-center/cloudfront-serve-static-website/