Lession 1 - What is Cloud Computing?

Characteristics

- On-Demand
- Broad Network Access
- Resource Pooling - Multi-tenant enviornment
- Rapid Elasticity
- Measured Service

Service Models

- Infrastructure as a Service
- Platform as a Service
- Software as a Service

Region and Availability Zones

- Oregon(us-west-2) us-west-2a/us-west-2b/us-west-2c

Edge Locations

- More than 100 edge locations worldwide

Scope of Services

- Global Services
    - AWS IAM
    - Amazon CloudFront
    - Amazon Route53
- Regional Serives
    - Amazon DynamoDB
    - Amazon Simple Storage Service
    - Elastic Load Balancing
    - Amazon Virtual Private Cloud
    - Amazon Elastic Block Cloud
    - Amazon Elastic Compute Cloud
    - Subnets

Lession 2 - Identity ana Access Management

Authentication

Authorization

- users
- groups
- password policy
- multifactor authentication (6 digits)

AWS API (Access ID + Secret Key)

- Operating System (User name + password/key pair)
- Application (User name + password)

CLI

SDKs - python or java

Web-basd Management Console

Users and Groups

- Are created and exist within IAM service
- Login to Mangaement Console
- Can have long-term access keys
- Can enable per-user MFA device

Types of credential types

Credential	Usage
Email address + Password	Master account (root) access
Username + Password	AWS Web Console
Access Key + Secret Key	CLI, SDK
Access/Secret Keys + Session Token	Role-based access

Create Groups/Policies

- For example, groups like admin, developers, everyone

Create Users

- For example, users with programmatic access (enable an access key and secrete access key for AWS API, CLI, SDK, and other development tools), AWS Management Console access (enable a psw that allows users to sign-in to the AWS Management Console). Set permissions.

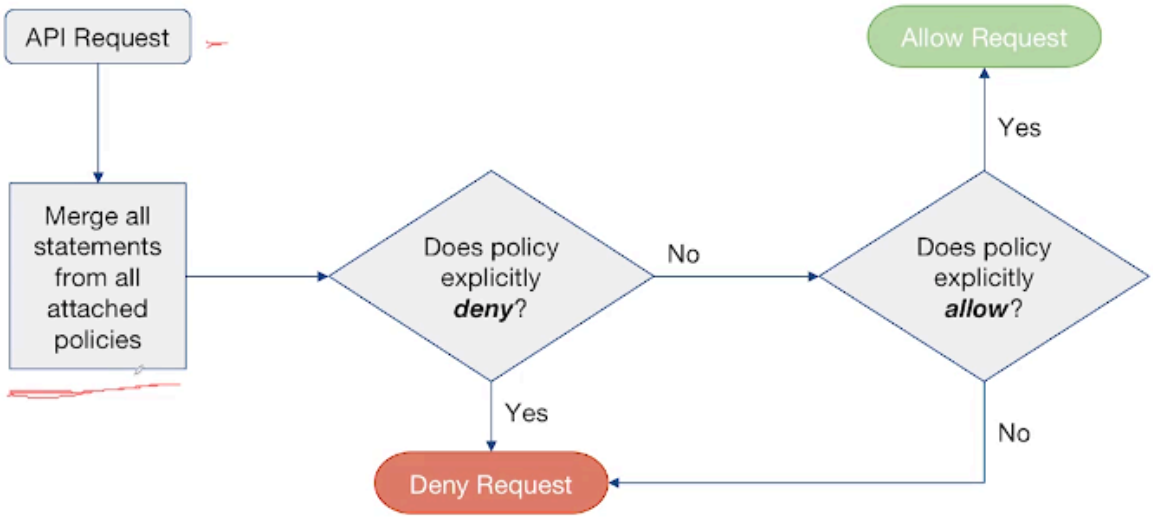
Policies

- Determine authorization (permissions)
- Written in JSON
- Policy Types
    - Mangaed Policy
        - AWS managed
        - Customer managed
    - Inline Policy
- Create policies via generator, or hand written policy

    - Version is a date-based version that represents when the policy standard was published by AWS
- Statement is a list of permissions being granted
    - Effects states whether an action is allowed or denied
- Actions are namespaced accoding to service, specifically load(put) and download(get)
    - Resource specifies a signle or list of resources on which the actions may or may not be performed
- Conditions - under the conditions
    - Polies will merged and denies always win

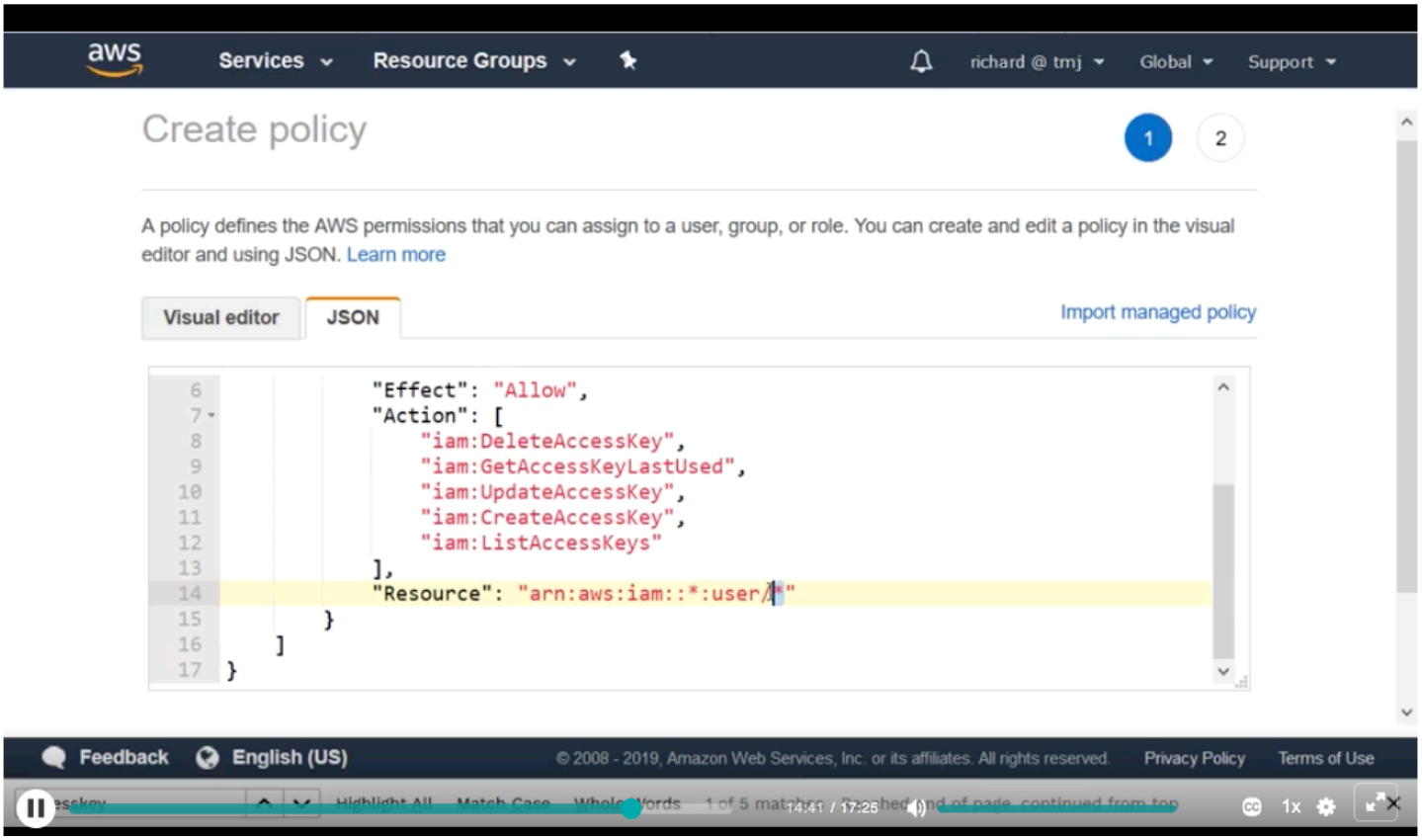
Policies

Policy Evaluation Logic



- Attach Policy to the groups (Manged Policies)
    - AWSLambdaFullAccess
    - AmazonS3FullAccess
    - AmazonDynamoDBFullAccess
    - IAMReadOnlyAccess
    - IAMSelfMangedService
    - IAMSelfUserChangePassword
- Inline Policies that are embedded in the group - Example DenyDangeous
    - Policy generator (Allow/Deny)
    - AWS Service (AmazonDynamoDB)
    - Actions(DeleteTable)
    - Amazon Resource Name(ARN) - *

Policies



Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs.

Credentials

- Should not embed access keys in code
- Should not embed in environment variable
- Shuodl not share with third parties, hundreds of enterprise users, millions of web/app users
- Use temporary credentials
    - delegate permissions to EC2 instance, AWS service, A user/elevate privileges
    - seperate account, one you own, third party account

Roles for EC2

- Allows AWS services to perform actions on your behalf. e.g., EC2, Lambda, etc
- For example, the application on EC2 (APP+SDK) needs to authenticate agianst the AWS API for accessing other services
- Apply a Role to the EC2 instance which provides a way to apply the permissions and retrieve temporary credentials
- Policy is applied to the role, which specifies the actions and resources allowed or denied.
- AWS STS, Security Token Service generates temporary credentials
- AWS SDK will automatically retrieve temporary credentials from the IMDS

Roles for Cross-Accounts Access

- Roles (Admin account; Production account) -> trust relationship tells ASW that our production account is trusting, or delegating permission to the Admin account
- Policy (Allow assume role)
- Assuming a role requires a call to STS to retrieve temporary credentials
- Use temporary credentials are then used to access resources in the other account

Existing Users

- Organizational Users -
    - LDAP
    - Microsoft Active Directory
    - Web/mobile application users: Facebook, Google

Federation Options

- SAML AWS Mangement Console, CLI, and API access authenticating with enterprise credentials.
- AWS Management Console access authenticating with Active Directory username + password.
- Amazon Cognito API access for web/mobile app users authenticating with OpenID Connect (OIDC).
- AWS SSO - Single sign-on provides access to multiple AWS accoutns and business applications.

First Steps

- Enable AWS CloudTrial
- Create an admin user in IAM (with optinal condition)
- Enable Multi-Factor Authentication on root account
- Enable Cost and Usage Report
- Log out of **root account**
- Log in with admin user
- Create additional users, groups, etc

IAM Best Practices

- Master account (root) credentials
- Email address + password
- Protect at all costs
- Delete any existing access/secret keys
- Do not use for day-to-day operations
- Follow principle of least privilege
- Rotate long-term credentails (access keys/passwords)
- Enable Multi-Factor Authentication (MFA)