# Cloud Security

# Cloud Security Overview

- Concerns
  - Privacy
  - Security assurance
  - Copyright Protection
- Control based safeguards to protect
  - Cloud infrastructure
  - Applications
  - Data from theft, leakage or loss

# Cloud Security Issues

- Data Loss
  - Accidental deletion
  - Malicious attacks
  - Software bugs
  - Synchronization errors
  - Cloud Provider Outages or Failures
  - Lack of backups
  - Data remanence – improper deletion leaving residual data.

# Data Loss Mitigation

- Regular backups
- Strong access controls and authentication
- Monitoring and alerting for suspicious activities
- Data recovery plans
- Overwrite data multiple times, cryptographic erasure
- Example
  - Code spaces security incident → led to closure

# Cloud Security Issues

- Data Privacy
  - Unauthorized access
  - Data breaches
  - Data Location and Sovereignity
    - Depending on law of country where data is stored
  - Data remanence

# Cloud Security Issues

- Data Privacy
  - Unauthorized access
  - Data breaches
  - Data Location and Sovreignity
    - Depending on law of country where data is stored
  - Data remanence
- External and Internal Threats
  - BYOD

# Cloud Security Requirements

## Physical Security

Secure data centers against physical threats

- Natural disasters
- Man made – intruders, human errors

Multi Layered Monitoring

- Monitoring centers, staff training

## Virtual Security

Identity Management

Access Management

Break Glass procedure – in case of emergency

Key Management

Auditing

Security monitoring and testing

# IAM – Identity and Access Management

## Manages

- **Identities** – Users, groups, roles and services
- **Access** – permissions and policies that define what the identities can do

## IAM Authentication – prove who user is

- Username/Password
- MFA – Multi-Factor Authentication
- Access Key/API keys for programmatic access
- Federated Identity – SSO(Single Sign On)
- Certificates

## Example

- Google, Azure - OAuth

# Class Exercise

- Consider logging into a Linux machine.
- How does the system recognize you as a valid user?

# Authentication - solution

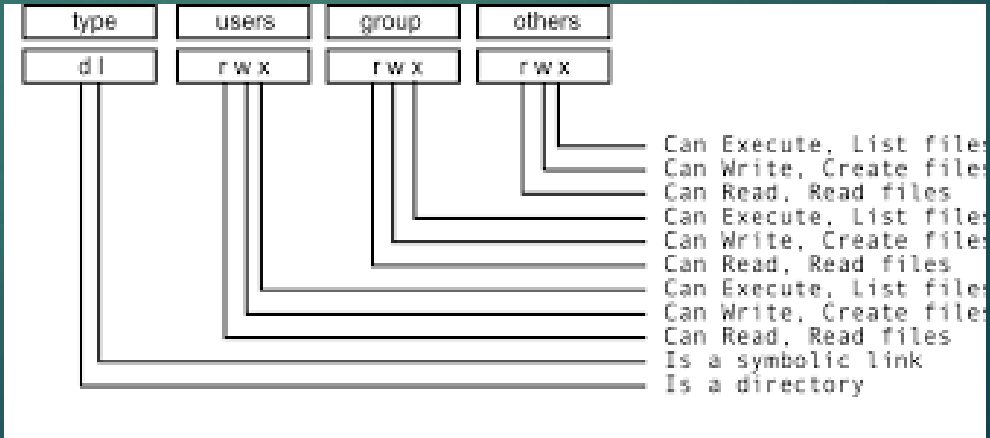
- Authentication
  - Proving who you are
- Need an identity
- Some secret sharing mechanism.

# Problem

- Ok. You proved who you are, but how does the system know you have the access rights?
- What is the Unix solution?

# Authorization – access control

- User name of a stored file - storage
- User id of the running process - compute
- Id decides what rights you have
- Who enforces access control - OS



type | users | group | others

d l | r w x | r w x | r w x

Can Execute, List files
Can Write, Create files
Can Read, Read files
Can Execute, List files
Can Write, Create files
Can Read, Read files
Can Execute, List files
Can Write, Create files
Can Read, Read files
Is a symbolic link
Is a directory

# Class Exercise: Moving to a IaaS system

- Consider that there are a group of machines and you would like to login and work on one of them.
- Will the same mechanism work?
- If not, list he problems?

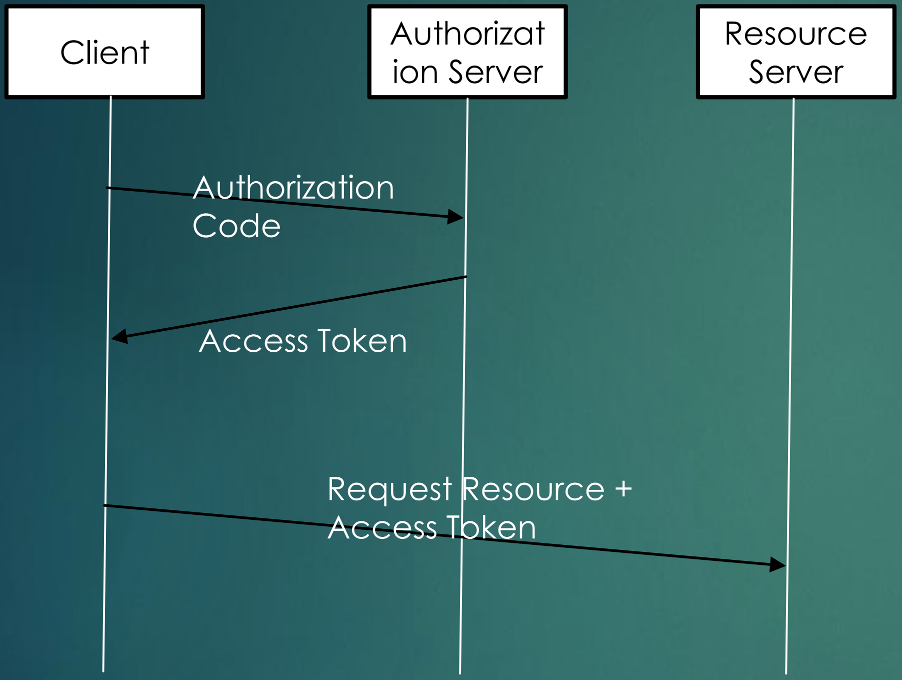
# Need a mechanism to

- Authenticate a user
- Enforce access control for the user on
  - Services
  - Projects
- List out all the services – provide a registry

# OAuth – Open Authorization

- Open Standard for Access Delegation
- Secure Authorization between Applications
- Instead of giving name
  - Give a token that allows limited access to resource
- Components
  - Resource Owner – the user who owns the data
  - Client – application requesting access
  - Authorization Server – Issues Tokens after verifying the user
  - Resource Server – Hosts the protected resources

# OAuth flow



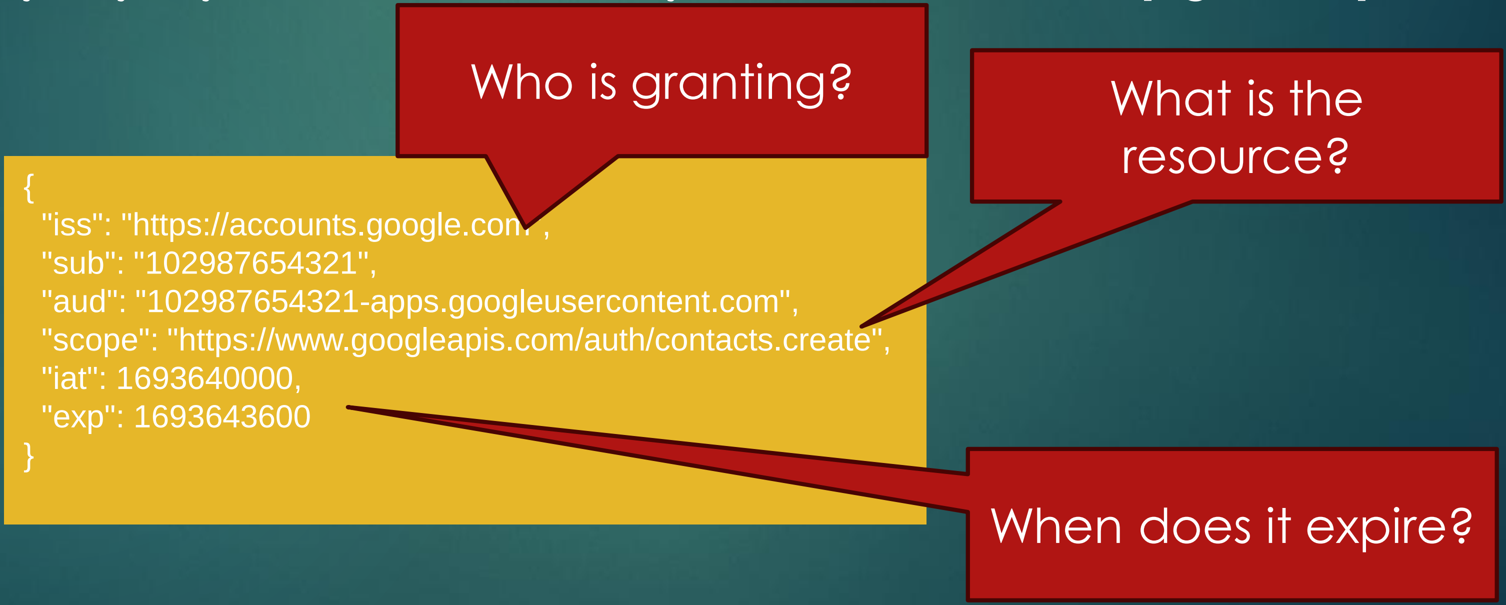
Client
Authorizat ion Server
Resource Server
Authorization Code
Access Token
Request Resource + Access Token

# What is in a token?

- JWT (JSON Web Token) format is used in base64
  - Takes binary data and encodes it into a set of 64 printable ASCII characters (A–Z, a–z, 0–9, +, /).
  - Every 3 bytes of binary data are converted into 4 Base64 characters.
  - Padding with = is used if the input data isn’t a multiple of 3 bytes.
- String of bytes encoded in Base64 format
  - Header
  - Payload
  - Signature
- Signature checked first using appropriate algorithm
  - Ensures token issued by trusted source
- Validate Claims
  - Expiration time, Not Before, Issued by, Audience, Subject
- If checks pass → token in valid
- What happens if user is deleted?

# JWT Token

- Example token to give permission to access contacts
  - eyJhbGciOiJSUzI1NiIsImtpZCI6IjEyMzQ1NiJ9.
    eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJzdWIiOiI
    xMDI5ODc2NTQzMjEiLCJhdWQiOiIxMDI5ODc2NTQzMjEtYXBwcy5nb
    29nbGV1c2VyY29udGVudC5jb20iLCJzY29wZSI6Imh0dHBzOi8vd3d3Lmdv
    b2dsZWFwaXMuY29tL2F1dGgvY29udGFjdHMuY3JlYXRlIiwi
    aWF0IjoxNjkzNjQwMDAwLCJleHAiOjE2OTM2NDM2MDB9. [signature]



```
{
  "iss": "https://accounts.google.com",
  "sub": "102987654321",
  "aud": "102987654321-apps.googleusercontent.com",
  "scope": "https://www.googleapis.com/auth/contacts.create",
  "iat": 1693640000,
  "exp": 1693643600
}
```

Who is granting?

What is the resource?

When does it expire?