

AWS Rest

• IAM Security tools

1. IAM Credential report (Account-level)



Report that lists all your account's users & the status of their various credentials.

2. IAM Access Advisor (User-level)



Shows the service permissions granted to a user & when those services were last accessed.

• Shares resp. AWS

- Infra
- Config. & Vulnerability
- Compliance val.

User

- User, group, roles - etc.
- MFA - Keys - Perm.
- Pattern, etc.

◦ EC2 Instance Types

1. General purpose

↳ great for a diversity of work loads

such as web servers or code repos.

Eg: t2.

2. Compute Optimized

↳ Batch workloads, Media, ML, Gaming

Eg C6g.

3. Memory optimized

↳ Db, web scale cache stores, BI.

Eg. R, X, Z, High memory

4. Storage opt.

↳ OLTP, DB, warehouse, DHT, file sys -

disk layout, e.g.: D, E, H, I, M1, M2, M3, M4

- Sec. groups - multiple instances

↳ good to one sys. for SSM access

↓

22 = SSH, SFTP; 21 → FTP, 80

HTTP

T

443 → HTTPS, 3389 → RDP

	SSM	path	ECC instance count
Mac	✓		
win < 10		✓	✓
Linux	✓		
win > 10		✓	✓

Pushing Opt.

↳ On-Demand

2. Reserved (min 1-3yr)

↳ Res. inv → long work

↳ Convertable " " → long, with flexible

work - " " → every ~~but~~ friday 3-6pm

3. Spot, instance

↳ Dedicated host → book entire physical server

↳ " instance → no other host. → same h/w

Share AWS EC2

Over

- intra

- 39 rules

- Isolation on phy. hosts

- OJ

- replace h/w

- S/W

- Compliance validation.

- IAM roles

- Data security of ec2

- You can launch EC2 from a public AMI, own AMI or marketplace

- EFS - Infrequent Access (EFSIA)

- Vertical scalability → +2 microservices
- Horizontal scaling → +1 EFT
- Scalability → ability to accept load
- Elasticity → auto-scaling
- Agility → add new res. in clicks
- Auto scaling groups - Scaling strategies

1. Manual Scaling

2. Dynamic Scaling

↳ Simple Load scaling.

• CPU > 70%

↳ Target tracking.

Adjust CPU at around 90%

↳ Scheduling scale

↳ Predictive : AI

• S3 Security

1. User based : IAM policies

2. Resource based.

↳ Bucket policies, Obj- & Bucket ~~ACL~~.

3. Encryption

• S3 replication

1. cross region replication (CRK)

↳ configure, low latency access,
replication across accounts.

2. SSR - log aggregation, live replication b/w prod. & test acc's.

• S3 storage class

1. General purpose → Big data analytics, mobile & gaming

app. content dist.

- low latency (high throughput)

Sun Mon Tue Wed Thu Fri Sat
○ ○ ○ ○ ○ ○ ○

Date _____
Subject _____



Std.

2. ^ Infrequent Access

- ↳ As a data stored for ~~dot~~ disaster recovery

2. O

3. Intelligent tiering

- ↳ same latency & throughput as S3

↓ divides into ↘
frequent access infrequent access

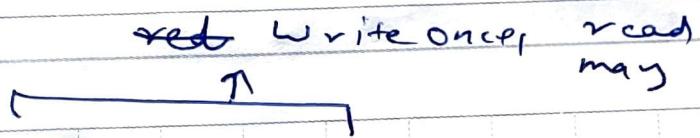
- ↳ Resilient against event that impact entire AZ.

4. S3 one zone - IA

- ↳ Same as IA but in one zone

5. Glacier

6. Glacier Deep Archive



, S3 object lock is glacier vault lock

S3 Encryption

↳ no ↳ server-side ↳ client-side

- **Share** → AWS
 - infra
 - conf. &
 - vul. ana.
 - compliance validation
- **AWS storage Cloud native opt.**

1. Block

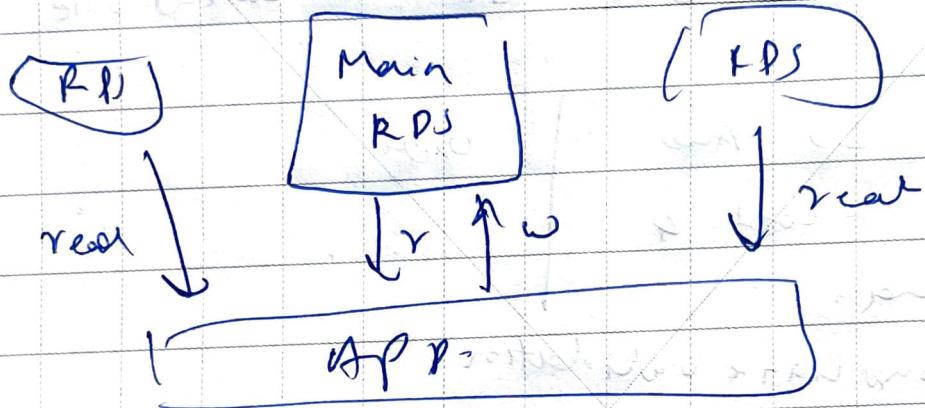
↳ eBS ↳ EC2 instance

2. file → EFS

3. object → S3, Glacier

o RDS deployment

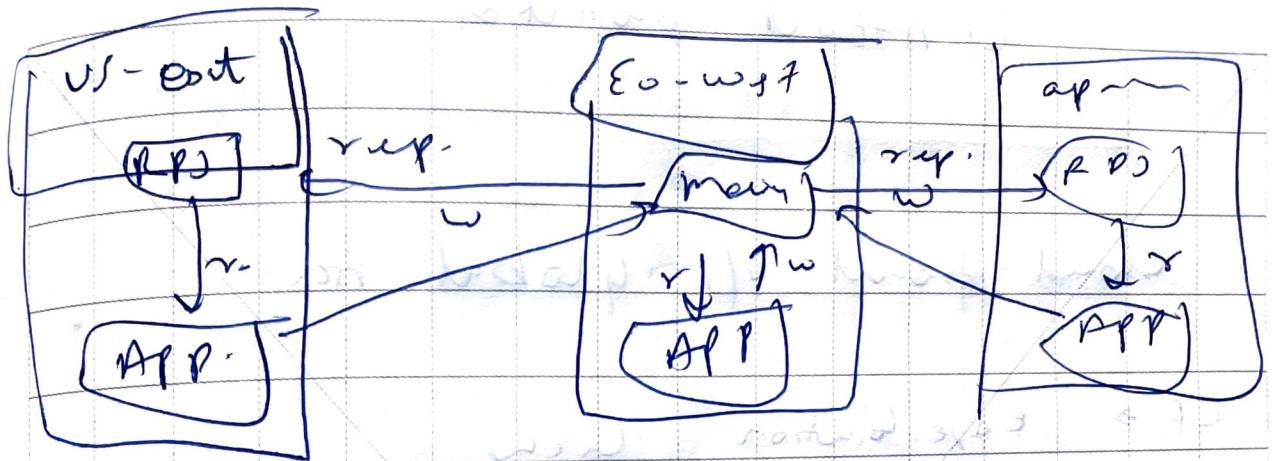
1. Read replicas:



2. Multi-AZ



3. Multiple region



Services

- S3, Dynamo DB, Lambda

- Benefits of Cloud formation

1. Intra-regions.
2. IoT
3. Prod.
4. Use temp.
5. Support all AWS reg.

Sun Mon Tue Wed Thu Fri Sat
○ ○ ○ ○ ○ ○ ○

Date _____

Subject _____

MEETING
WITH

- Global app.

↳ ↓ latency ↳ Disaster recovery.

↳ attack protection

- Cloud front v/s Global Acc.

CF → edge location, cache

GA1 no cache, moving part at edge

- Outro out benefits.

↳ low latency, local data processing, data residency, easier-migration, fully managed service

- Global App. Arch.

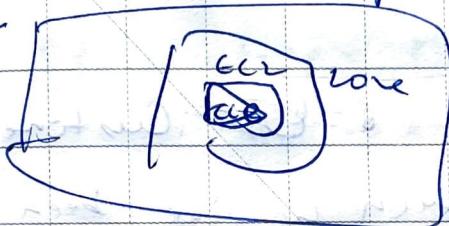
1. Single region, single A2.

- ✓ right ARA. - ✗ globular battery

no \oplus

- less octopus diff.

Ry



2. Single R, multiple A2.

- ✓ - ✗

1 wa - med.

3. MR, Active Positive

- ✓ globularized battery

- ✗ u write \rightarrow u

- Δ diff.

4. MR, AA. - ✓ - ✓ - Δ

Important Metrics

1. EC2 instances ~~per day~~ 3. S3 Bucket
 2. EBS vol ~~usage~~ 4. Billing
 3. Service limit ~~& Ext Customer~~
by how much you've been using available API
- ## DDoS protection on AWS

1. AWS Shield std.
2. CloudFront Adm.
3. WAF
4. CloudFront & Route 53
5. AWS Auto Scaling

- Penetration testing allowed :-

1. EC2
2. RDS
3. CloudFront
4. Aurora
5. API gateway
6. Lambda
7. Lightsail
8. Beanstalk env.

- ② Types of Customer Master keys (CMK) :-

1. Cost managed treated as cost by AWS
2. AWS managed on behalf of cost.
3. AWS owned

↳ Collection of CMKs AWS services own & manage them in multiple acc.

↳ CloudWatch Metrics

- 4. CloudWatch Metrics

- Root User Privileges

- ✓ Change account settings

MEETING
WITH

2. View certain ~~other~~ ten invoices

✓ 3. ~~Re~~ close your AWS acc.

4. Restore IAM user perm.

✓ 5. Change or cancel AWS support plan

✓ 6. Reg. as seller

7. S3 to enable MFA

8. Edit / del. S3 policy

9. Sign up for GovCloud

• Service Control Policies (SCP)

↳ Gov permissions

o AWS Pricing Models

1. Pay as you go.
2. reserve.
3. Pay less by using more
4. Pay less as AWS grows

o free services in AWS

- IAM - VPC - Billing
- Elastic BS
- Cloud Formation } No pay for creation
- AWS CI

free tier - EC2, t2.micro for a year

- S3, B&B, ELB, AWS Data Transfer

o Price Lambda

- Pay per call
- Pay for duration

Sun Mon Tue
○ ○ ○ ○ ○ ○

Subject

MEETING
WITH

o Price \leftrightarrow Left

↳ No. & size of obj.

o d ↳ No. & type of req.

↳ Data out

o EBS \rightarrow vol. type

↳ storage vol. in GB / month

↳ Data out

o RDS \rightarrow per hour billing

↳ on-demand

↳ reserved

↳ no. of HTTPS req.

o Cloud front \rightarrow diff. all reg.

↳ data out

o Billing & cost tools

Date
Subject

MEETING
WITH

Sun Mon Tue Wed Thu Fri Sat

○	○	○	○	○	○	○
---	---	---	---	---	---	---

total cost ownership

- 1- TCO → pay as you go
2. simple month cal. / pricing

- cost allocation tags
- tagging & resource group
- cost & usage report
- cost explorer

, anal acc. to foll
how to budget

- AWS trust advisor - support plan

- 7 core checks : S3 Bucket permission,
SG, IAM user (1 min.)
MFA on main acc., EB public snapshot,
RDS in 7 service units

- Priority: AWS support

1. Basic support free
2. Dev.
3. Enterprise

o AWS Cloud Best prac. - Design

1. Scalability.

2. Disposable res.

3. Automation

5. Services, not
service

4. Loose coupling

o Well Arch. framework for all as:-

1. Operational excellence 2. Security

3. Reliability. 4. Performance

5. Cost optimization