**Content**

[1. Questions 2](#_Toc16200179)

[1. iSCSI 2](#_Toc16200180)

[2. Intro – General - IAM 4](#_Toc16200181)

[1. Well Architected Framework 4](#_Toc16200182)

[2. Edge Locations 4](#_Toc16200183)

[3. STS and Temporary Security Credentials 6](#_Toc16200184)

[4. Identity Federation 7](#_Toc16200185)

[2.4.1. Custom Identity Provider 7](#_Toc16200186)

[2.4.2. LDAB / Active Directory 7](#_Toc16200187)

[2.4.3. Web Identity 8](#_Toc16200188)

[5. Organizations 9](#_Toc16200189)

[3. EC2 10](#_Toc16200190)

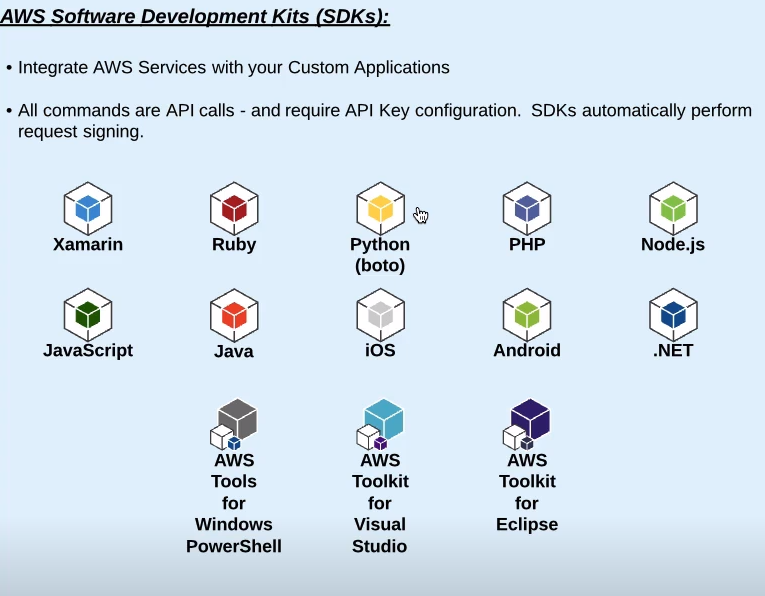
1. Questions
2. Incorrect. EBS is block storage and can only be attached to a single instance.
3. EMR is designed for processing of big data/large unstructured data sets. It's not a good fit for long term data warehousing.
4. Redshift: AWS product which will allow storage of that data to be used for long term reporting, querying, forecasting, and business intelligence.
5. Natively Highly Available services: Internet Gateway, Virtual Private Gateway, Dynamic Hardware VPC VPN, DynamoDB, S3
6. An IGW is resilient by design, and only one needs to be attached to a VPC in order to provide all subnets in all AZ's with resilient internet connectivity.
7. How to mask a failure of an instance? An Elastic IP can be remapped from a failed instance to a replacement instance. That doesn't require any application or DNS changes.
8. It is not possible to encrypt an existing EBS volume. You can take a snapshot of the unencrypted volume. Once the snapshot is taken, copy the snapshot and enable encryption on the copy so that the target snapshot is encrypted. Once the target snapshot is created, you can attach a new encrypted volume to the EC2 instance, and restore the encrypted snapshot to a new volume.
9. AWS Shield offers managed DDOS protection
10. Placement groups are a clustering of EC2 instances in one Availability Zone with fast (up to 25Gbps) connections between them. This feature is used for applications that need extremely low-latency connections between instances.
11. IOPS provisioned instances does not have an impact on the degree of latency between instances.
12. Amazon EFS now allows you to instantly provision the throughput required for your applications independent of the amount of data stored in your file system. This allows you to optimize throughput for your application’s performance needs.
13. EFS mount only for Linux instances, not for windows.

1. iSCSI

In computing, iSCSI is an acronym for Internet Small Computer Systems Interface, an Internet Protocol (IP)-based storage networking standard for linking data storage facilities. It provides block-level access to storage devices by carrying SCSI commands over a TCP/IP network. iSCSI is used to facilitate data transfers over intranets and to manage storage over long distances. It can be used to transmit data over local area networks (LANs), wide area networks (WANs), or the Internet and can enable location-independent data storage and retrieval

1. Intro – General - IAM

WE interact w services through service API. Service API calls are made to service API endpoints.



* 1. Well Architected Framework

Best practices and recomendations.

* Operational Excellence
* Reliability
* Security
* Performance Efficiency
* Cost Optimization
  1. Edge Locations

AWS datacenters (points of presence) located around the world, designed to give low latency access to 2 AWS services:

* Route 53 – DNS Lookups
* CloudFront – CDN, Cached content, streaming distributions, acceleration.

Services that run in CloudFront: security / accelerate other services.

1. Shield
2. WAF – web application framework
3. Lambda@Edge
4. S3 Transfer
5. API Gateway

**Point of Presence** is the point at which two or more different networks or communication devices build a connection with each other.

**Edge Locations vs Regional Edge Caches**

The nine new Regional Edge Cache locations are in Northern Virginia, Oregon, São Paulo, Frankfurt, Singapore, Seoul, Tokyo, Mumbai, and Sydney. These locations sit between your origin webserver and the 68 global edge locations that serve traffic directly to your viewers. As the popularity of your objects reduce, individual edge locations may evict those objects to make room for more popular content. Regional Edge Caches have larger cache-width than any individual edge location, so your objects remain in cache longer at these locations. This helps keep more of your content closer to your viewers, reducing the need for CloudFront to go back to your origin webserver, and improving overall performance for viewers. For instance, our edge locations in Europe now go to the regional edge cache in Frankfurt to fetch an object before going back to your origin webserver.

To deliver content to end users with lower latency, Amazon CloudFront uses a global network of 187 Points of Presence (176 Edge Locations and 11 Regional Edge Caches)

**Vertical scaling** – increase the capacity of a single instance or server

**Horizontal scaling** – add or terminate the number of instances

**Overprovisioning** – when deploying counting w peak loads + some buffer. Not cost efficient.

Root account is logging in with email address.

Resource Groups – a collection of resources that you want to identify to being part of a group. You can use it in conjunction with Tag Editor. Group together resources and manage the tags.

Power user access – Admin access except it does not allow user/group management.

Inline policy – when a user gets a special rights.

* 1. STS and Temporary Security Credentials

You can use the AWS Security Token Service (AWS STS) to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use, with the following differences:

* Temporary security credentials are short-term, as the name implies. They can be configured to last for anywhere from a few minutes to several hours. After the credentials expire, AWS no longer recognizes them or allows any kind of access from API requests made with them.
* Temporary security credentials are not stored with the user but are generated dynamically and provided to the user when requested. When (or even before) the temporary security credentials expire, the user can request new credentials, as long as the user requesting them still has permissions to do so.

These differences lead to the following advantages for using temporary credentials:

* You do not have to distribute or embed long-term AWS security credentials with an application.
* You can provide access to your AWS resources to users without having to define an AWS identity for them. Temporary credentials are the basis for roles and identity federation.
* The temporary security credentials have a limited lifetime, so you do not have to rotate them or explicitly revoke them when they're no longer needed. After temporary security credentials expire, they cannot be reused. You can specify how long the credentials are valid, up to a maximum limit.

STS allows you to create temporary security credentials that grant users access to your AWS resources. These temp creds are for short-term use, with configurable session duration between 15 min and 12h / 36h

When requested thorugh an STS API call, a credential object is returned containing:

* Session Token
* Access Key ID
* Secret Access Key
* Expiration Timestamp

**API keys are needed when working "programmatically" through the CLI, PowerShell, Direct HTTP calls, and SDK API access.**

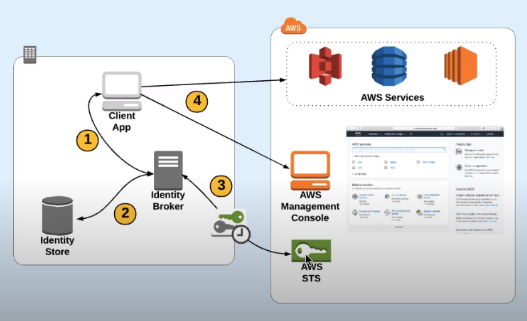
Benefit: avoid distributing or embedding long-term AWS security credentials in an application.

* 1. Identity Federation

Authenticate users using an Identity Broker Application running outside of AWS.

* + 1. Custom Identity Provider

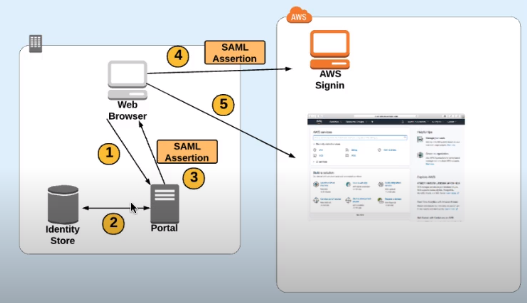
Authenticate against the identity broker app then it checks if you provided valid creds. Requests for you an STS temp cred (you are gonna be write a little bit of code at step 3 to request the credentials and forward them) and you are ready to go.



* + 1. LDAB / Active Directory

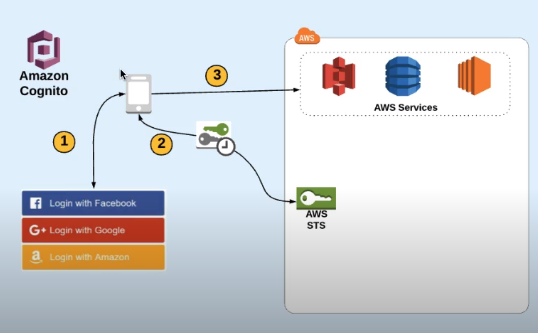
SAML – Security Assertion Markup Language – an open standard for exchanging authentication and authorization data between 2 different parties, such as your active directory and AWS.

In the console you choose identity provider. A user browses to a webpage, where he type in its active directory credentials, they got authenticated. Then the SAML compatible application, like Active Directory generates, what’s called a SAML assertion. Sends back to browser, than it post to a special sign in URL, which validates the assertion. Then automatically sends back a response which redirects the user to the console. The permission that the user will have is defined by a role, which is pre-associated with the users active directory group. That’s how to do SSO



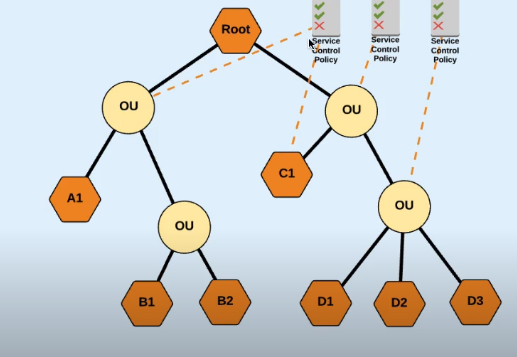
* + 1. Web Identity

You want to use them in web-based / mobile applications. When you have a mobile application, instead of having every user of you app needing a set of AWS credentials, you allow the users to log in with what’s called an **Open ID connect provider like Facebook / Google / Amazon.** You can also create your own as long as its **open id connect compatible**. Once the user is authenticated against the choosen platform the app requests credentials. Here helps Amazon Coginito – takes care of the part of exchangeing your web identity authorization token for STS credentials



* 1. Organizations

To enforce what accounts what are allowed to do. Which services? Regions? How much money they are able to spend? Allows you to create a root account then invite other accounts to join to the organization beneath that root account. You can group together accounts in OUs and then apply SCPs. You can’t create any IAM policies that are in violation with SCPs. Consolidated Billing – all of the accounts payments will roll up to the root account.



1. EC2

An AMI includes the following:

* A **template for the root volume** for the instance for e.g. an operating system, an application server, and applications
* **Launch permissions** that control which AWS accounts can use the AMI to launch instances for e.g. AWS account ids with whom the AMI is shared
* **A block device mapping** that specifies the volumes to attach to the instance when it’s launched

AMIs are specific to a region and if needed in other region must be copied over

Launch permissions define who has access to the AMI

* Public – Accessible to all AWS accounts
* Explicit – Shared with specific AWS accounts

1. Private – Owned and available for AMI creator AWS account only