Slide 4:

The shared responsibility model for infrastructure services such as EC2, specifies that AWS manages the security of the following assets:
Facilities
Physical security of hardware
Network Infrastructure
Virtualization infrastructure

Infrastructure Services: This category includes… these services, you can architect and build a cloud infrastructure using technologies similar to on-premisis solutions. You control the OS, and you configure identity management for user that interact with these services.
(**EBS** is persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.)

Container Services: Services in this category typically run on separate Amazon EC2 instances, but its possible dont manage the operating system or the platform layer. AWS provides a managed service for these application "Containers". . Examples of container services include RDS (relational database service), EMR (elasitc MapReduce), and Elastic Beanstalk

Abstracted Services: This category includes high-level storage, database, and messaging services, such as S3, Amazon Glaciaer, DynamoDB, SQS, and SES. THese services abstract the platform or management layer on which you can build and operate cloud applications. You can access the endpoints of these abstracted services using AWS APIs, while AWS manages the underlying service components or the oeprating system on which they reside. You share the underlying infrastructure, abstracted services porvide a platform which isolates your data in a secure fashion using IAM.

Amazon Relational Database Service (**Amazon RDS**) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud.

Amazon Virtual Private Cloud (Amazon **VPC**) enables you to launch (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Amazon EMR processes big data across a Hadoop cluster of virtual servers on **Amazon** Elastic Compute Cloud (EC2) and **Amazon** Simple Storage Service (S3). The elastic in **EMR's** name refers to its dynamic resizing ability, which allows it to ramp up or reduce resource use depending on the demand at any given time.

Slide 5:

This is essentially going to be the Outline of what I plan to talk about today.

Slide 6:

The customer is responsible for the security of all services they choose to use in the blue area.
You indirectly access AWS IAM
To simplify, pretend Customer IAM is the virtual machine you have full control over, and the AWS IAM is the host/hardware/software beneath the Virtual machine.

So for example:

(Amazon VPC), and **Amazon S3** are categorized as Infrastructure as a Service (IaaS) and, as such, require the customer to perform all of the necessary security configuration and management tasks.

The opacity layer can include data encryption, data integrity authentication, Software - and data-signing, secure Time-stamping, and more.

All of this sits on top of AWS global infrastructures which is for AWS to manage. They operate from the specific region that they've been launched.

Slide 7:

The customer is responsible for the security of all services they choose to use in the blue area.
So for example:
Customer Data - S3 bucket example, contains categories for where data is stored. Access to this is controlled with IAM policies or Bucket policies. For protecting the **data in Transit** you can use things such as HTTPS encapsulation
Firewall Configuration - Security groups which can control access at the instance level
for managing platform-level identity and access management separately from IAM
At the **platform level**, you can use IAM tools to apply ACL-type permissions for individual resources.
EC2 Instances would reside in the Operating system area which you are responsible for as well as updates and security patches),
Firewall (called a security group) on each instance.

Storage= EBS volumes, S3
Database= RDS (l**relational database service**)
Networking = The phyiscal networks

relational database service by Amazon Web **Services** (AWS). It is a web **service** running "in the cloud" designed to simplify the setup, operation, and scaling of a **relational database** for use in applications.

Slide 9:
IAM enables you to…. read slide

Everything in AWS can be controlled via policy in IAM
Think of roles and policies controlling the Authorization

Slide 10:
The initial account you use to when you sign up with AWS will be the account that has full permissions to the AWS Console. Remember this account should not be used for everyday activity. You'll want to create users for either the AWS console and use IAM to set their permissions, or for the EC2 instances themselves, which you will use Security groups to enforce access. Network ACLs can be done through Security Groups at a subnet level.

Slide 11:
In terms to availability of your instances and resources, this is delegated across Regions, Availability zones and endpoints.

Each Regions will comprise of at least two Availability Zones.
Data is only stored in the specific region you decide to put it in. The responsibility lies on you if that data does need to be replicated.

Slide 12:
Ext: Availability zones are connected to multiple ISPs and different power grids.

Slide 14:
Its common for an organization to have multiple AWS accounts. AMI's and EBS snapshots can be shared across AWS accounts. You can restrict the users that have access to use these AMI's or EBS snapshots within the each AWS account.

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots which are *incremental*.

Slide 15:
I put the concept of least privilege as the starting point for this slide because everything I am going to explain about permissions and access is going to come from that fundamental of thinking.

Recall that when you first sign up with AWS, you are given an initial account that has full access to everything in the console. This AWS account represents the business relationship between you and AWS

This account should never be used in day to day interactions. Instead, additional accounts with delegated access for what their purpose of being made for should be created.

Optional:
So generally you might use several AWS accounts for example, one for each department for example, and then create an IAM user within each account for the appropriate people and resources

Slide 16:
With IAM you can create multiple users, each with individual security credentials, all controlled under a single AWS account. IAM users can be defined as a person, service, or application that needs access to your AWS resources through the management console, CLI, or directly via APIs. Want to stress the idea of least privilege when you are creating IAM users.
Optional:
The best practice is to create individual IAM users for each individual that needs access services and resources in your AWS account. You can create fine-grained permissions to resources under your AWS account, apply them to groups you create, and then assign users to those groups. This best practice helps promote least privilege.

Slide 17:
So any app that runs on an EC2 instance that needs to access an AWS resource such as an S3 bucket, must have security credentials in order to make programmatic request to AWS.

Theres a couple ways we can do this. A developer might use their creds in each instance or App that needs to access these resources but distributing long-term credentials to each instance is challenging to manage and a potentail security risk.

Lets say you dont want to create accounts directly within AWS for an IAM user because its permanent. With Identity Federation, AWS allows you to delegate access to resources to identities outside of AWS, such as your corporate directory. This can be done with IAM roles which provide temporary security credentials to address this issue. These temporary security credentials have a configurable expiration and can be automatically rotated. The benefit here is that you dont have to manage long term credentials and IAM users for each entity that requries access to a resource.

Optional:
Cross-account access - To manage access to resources, you might have multiple AWS accounts. You could isolate a dev environment from a prod environment. However, users from one account might need to access resources in the other accounts such as promoting an update from the dev environment to prod environment. Although users who work in both accounts could have separate identities in each account, managing creds for multiple accounts makes identity management difficult.

Slide 18:

So to explain, in this example of a IAM role implementation, a developer is running an application (EC2 instance) that requires access to a S3 bucket named photos. An admin creates the “get-pics” role. The role includes policies that grant read permissions for the bucket, and allow developer to launch the role within an EC2 instance. When an application runs on the

instance, it can access the photos S3 bucket by using this role's temporary credentials. The admin doesnt have to grant the developer permission to access the S3 photo bucket and the developer never has to share the creds.

Slide 19:

In an example with Identity Federation, we can also use IAM roles to achieve this. We would create an "Identity Broker" that sits between our corporate users and our AWS resources, which manages the authentication and authorization without needing to recreate of our corporate users as other IAM users within AWS.

Step 1. Enterprise users access the the Identity broker App. Think of this as a web form which integrate with AD
Step 2. Identity broker application authenticates the user against our corporate identity store.
Step 3. The IB application has permissions to access the AWS security token service to request a temporary security credential.
Step 4. Enterprise users given a temporary url that gives them access to the AWS APIs or the management console within AWS.

Optional:

the Identity Broker here could be a web form, which intergrates with Active Directory (Corporate identity store) checking if you authenticated successfully. Then it will direct you to the STS to acquire a Token

Slide 21:

Recall that there is a difference between AWS Console accounts and accounts that will access our EC2 instances. Now for accessing EC2 instances...

EC2
- Each user can have multiple EC2 key pairs, and can launch new instances using different key pairs. Important to note that EC2 key pairs are not related to the AWS account or IAM user credentials.
- Whereas IAM use credentials to control access to other AWS Services; EC2 key pairs control access only to your specific instance

The first time you create an instance, you're presented with a EC2 Key pair. You are required to download and securely store the private key. Important to note: AWS does not store the private key; if it is lost you'll have to generate a new key pair

Slide 22:

Using cloud-init when creating a new instance from a standard AMI, the public key of the Amazon EC2 key pair is appended to the initial operating system user's ~/.ssh/authorized_keys file

The user can then use an SSH client to connect to the Amazon EC2 Linux instance with their EC2 instance username as their identity and providing the private key file for user authentication.

Slide 23:

Windows Instances

- Windows uses what is called ec2config, which is a service
- When a new instance from a standard AWS AMI is launched, the ec2Config service sets a new random Administrator password for the instance and encrypts it using the corresponding EC2 key pairs public key
- The user can get the windows instance password by using the AWS management console or command line tools, and providing the corresponding Amazon EC2 private key to decrypt the password.
- This password, along with the default Administrative account for the EC2 instance, can be used to authenticate to the Windows Instance.

Slide 24:

You can also use Amazon Services to do Key Management. AWS provides a set of flexible and practical tools for managing Amazon EC2 keys and providing industry-standard authentication into newly-launched Amazon EC2 Instances. If you have security requirements, you can implement alternative authentication mechanisms, including LDAP or Active Directory Authentication, and disable EC2 key pair authentication.

Slide 25:

EC2 presents a true virtual environment, in which you can use web service interfaces to launch instances with a variety of operating systems with custom preloaded applications. You can also create your own AMIs that fit your bussniess need and publish them for internal or even for public if desired. On to some best practices for securing your AMIs...

Slide 26:

Disabling Insecure Applications:

- Disable services and protocols that authenticate users in clear text over the network or otherwise insecurely

Minimize Exposure

- Disabling Non-essential network services on startup
- Only Administrative services like SSH and RDP and services required for essential applications should be started

Protecting Creds

- Securely delete all AWS credentials from Dis and configuration files
- Securely delete any 3rd party credentials from disk and configuration files
- Securely delete all additional certs or key material from the system
- Ensure that software installed does not use default internal accounts and passwords.

Slide 27:

Protecting Credentials - You can configure EC2 Config service to have new random passwords generated on Instance creation for the administrator account upon book, but you must explicitly configure this before bundling the image

Slide 28:

After the hardened AMI is deployed, you can still amend and update security controls by using bootstrapping applications

Bootstrapping actions to consider

1. Security **Software updates** install the latest patches, service packs, and critical updates beyond the patch level of the ami
2. Initial application patches install application level updates, beyond the current application level build as captured in the AMI
3. Use tools like Chef to ensure non essential services aren't accidentally installed during software installs or application updates
4. Register instances with remote security monitoring and management systems

Slide 29:

Untrusted AMIs

- Launch instances from trusted AMIs only
- Trusted AMIs include the standard Windows and Linux AMIs provided by AWS and AMIs from trusted third parties.
- If you dervive your own custom AMIs from the standard and trusted AMI, all the addtional software and strings you apply to it must be trusted as well
- Launching an untrusted third-party AMI can compromise and infect your entire cloud environment

Untrusted Software

-Only install and run trusted software from a trusted software provider
-A trusted software provider is one who is well regarded in the industry, and develops software in a secure and responsible fashoin, not allowing malicious scode into its software packages.
- Open source software is fine if trusted and you should be able to compile your own packages.

Patching: Keeping external and internal systems to the latest security level. Worms often spread through unpatched systems on the network.

Host-Based IDS Software

Many AWS customers install host-based IDS software, such as the open source product OSSEC, which includes file intergrity checking and rootkit detection.

Slide 31:

After a user or IAM role has been authenticated, they can access resources to which they are authorized. You provide resource policies or capability policies depending on whether you want the user to have control over the resources, or whether you want to override individual user control

Slide 32:

There are appropriate cases in where a user creates resources and then wants to allow other

users to access those resources. In this model, the policy is attached directly to the resource and describes who can do what with the resource. The user is in control of the resource. You can provide IAM user with explicit access to a resource. The root AWS account always has access to manage resource policies, and is the owner of all resources created in that AWS account.

Unlike a user-based policy, a resource-based policy specifies who (in the form of a list of AWS account ID numbers) can access that resource.

A few AWS services that support resource-based policies are:

Amazon S3 Buckets - The policy is attached to the bucket, but the policy controls access to both the bucket and the objects in it

Slide 33:

Capability (User-based permissions) policies are often used to enforce company wide policies. Capability policies are assigned to an IAM user either directly or indirectly using an IAM group.
They can also be assigned to a role which would be assumed at run time. Capability policies define what capabilities or actions the user is allowed or denied to perform. They can override resource-based policy permissions by explicitly denying them.

IAM Policies - can be used to restrict access to specific IPs, or during specific days and times of the day.

Both resource policies and capability policies are cumulative in nature. An individual user's effective permission is the union of resource policies and capabilities permissions granted directly or through group membership

Slide 34:

- The bucket is mostly private and requires my key/secret to add/remove/update/list files.
- There is a "directory" (i.e. key prefix) called "incoming" that will allow anonymous users to upload content to but not list.

Some things to note here:
Action: what they can do
Resource: what they can do it against
Effect: Allow - by default effect wont be there and its always deny by default.

Slide 35:

For those of us who dont write JSON freestyle at will, AWS has...

Slide 36:

Designate data as confidential and limit the number of users who can access it. Use AWS permissions to manage access to resources for services such as S3.
To ensure that data integrity is not compromised through deliberate or accidental modification, use resource permissions to limit the scope of users who can modify the data. Even with resource permissions, accidental deletion by a privileged user is still a threat. Perform data intergrity checks to detect data compromise and restore the data from backup or in the case of S3, from a previous version.

Slide 37:
Using the correct permissions and the rules of the least privilege is the best protection against accidental or malicious deletion. For services such as S3, you can use MFA delete to require multi-factor auth to delete an object, limiting access to objects. If you detect data compromise, restore the data from backup, or in the case of S3, previous Version.

So what about System, Infrastructure, Hardware or software availability? In the case of a system failure or a natural disaster, restore your data from backup, or from replicas. Some services such as S3 and DynamoDB, provide automatic data replication between multiple Availability zones within a region. Just to emphasis the strategies here, backups, and replication

Slide 41:
Read S3 stuff first…

(Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS allows for persistent storage after reboot within instances.

EBS encryption uses AWS Key Management Service (AWS KMS) master keys when creating encrypted volumes and any snapshots created from your encrypted volumes. The first time you create an encrypted EBS volume in a region, a default master key is created for you automatically.

Slide 43:
Encryption and data integrity authentication are important for protecting the communications channel, but it is equally important to authenticate the identity of the remote end of the connection. An encrypted channel is worthless if the remote end happens to be an attacker, or an imposter relaying the connection to the intended recipient.

Slide 46:
Code Spaces was built mostly on AWS, using storage and server instances to provide its services. Those server instances weren't hacked, nor was Code Spaces' database compromised or stolen. According to the message on the Code Spaces' website, an attacker

gained access to the company's AWS control panel and demanded money in exchange for releasing control back to Code Spaces.

What good are backups if you keep them accessible in the same panel? I also read reports that Codespaces did not implement MFA auth on their logins...