AWS EC2 Security

## EC2 Key Pairs

* EC2 uses public-key cryptography to encrypt & decrypt login information
* Public-key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data.
* Public and private keys are known as a key pair.
* To log in to an EC2 instance, a key pair needs to be created and specified when the instance is launched, and the private key can be used to connect to the instance.
* Linux instances have no password, and the key pair is used for ssh log in
* For Windows instances, the key pair can be used to obtain the administrator password and then log in using RDP
* EC2 stores the public key only, and the private key resides with the user. EC2 doesn’t keep a copy of your private key
* Public key content (on Linux instances) is placed in an entry within  ~/.ssh/authorized\_keys at boot time and enables the user to securely access the instance without passwords
* Public key specified for an instance when launched is also available through its instance metadata http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
* EC2 Security Best Practice: **Store the private keys in a secure place as anyone who possesses the private key can decrypt the login information**
* Also, if the private key is lost, there is no way to recover the same.
  + For instance store, you cannot access the instance
  + For EBS-backed Linux instances, access can be regained.
    - EBS-backed instance can be stopped, its root volume detached and attached to another instance as a data volume
    - Modify the authorized\_keys file, move the volume back to the original instance, and restart the instance
* Key pair associated with the instances can either be
  + Generated by EC2
    - Keys that EC2 uses are 2048-bit SSH-2 RSA keys.
  + Created separately (using third-party tools) and Imported into EC2
    - EC2 only accepts RSA keys and does not accept DSA keys
    - Supported lengths: 1024, 2048, and 4096
* supports five thousand key pairs per region
* Deleting a key pair only deletes the public key and does not impact the servers already launched with the key

Network Interface

Graphical user interface

Description automatically generated

Text

Description automatically generated

## Graphical user interface Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

## EC2 Security Groups

* An EC2 instance, when launched, can be associated with one or more security groups with the instance, which acts as a virtual firewall that controls the traffic to that instance
* Security groups helps specify rules that control the inbound traffic that’s allowed to reach the instances and the outbound traffic that’s allowed to leave the instance
* **Security groups are associated with network interfaces**. Changing an instance’s security groups changes the security groups associated with the primary network interface (eth0)
* An Network interface can be associated with 5 security groups and with  60 rules per security group
* Rules for a security group can be modified at any time; the new rules are automatically applied to all instances associated with the security group.
* All the rules from all associated security groups are evaluated to decide where to allow traffic to an instance
* Security Group features
  + For the VPC default security group, it allows all inbound traffic from other instances associated with the default security group
  + By default, VPC default security groups or newly created security groups allow all outbound traffic
  + Security group rules are always permissive; deny rules can’t be created
  + Rules can be added and removed any time.
  + Any modification to the rules are automatically applied to the instances associated with the security group after a short period, depending on the connection tracking for the traffic
  + **Security groups are stateful** — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC security groups, this also means that responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules
  + If multiple rules are defined for the same protocol and port, the **Most permissive rule** is applied for e.g. for multiple rules for tcp and port 22 for specific IP and Everyone, everyone is granted access being the most permissive rule

### Connection Tracking

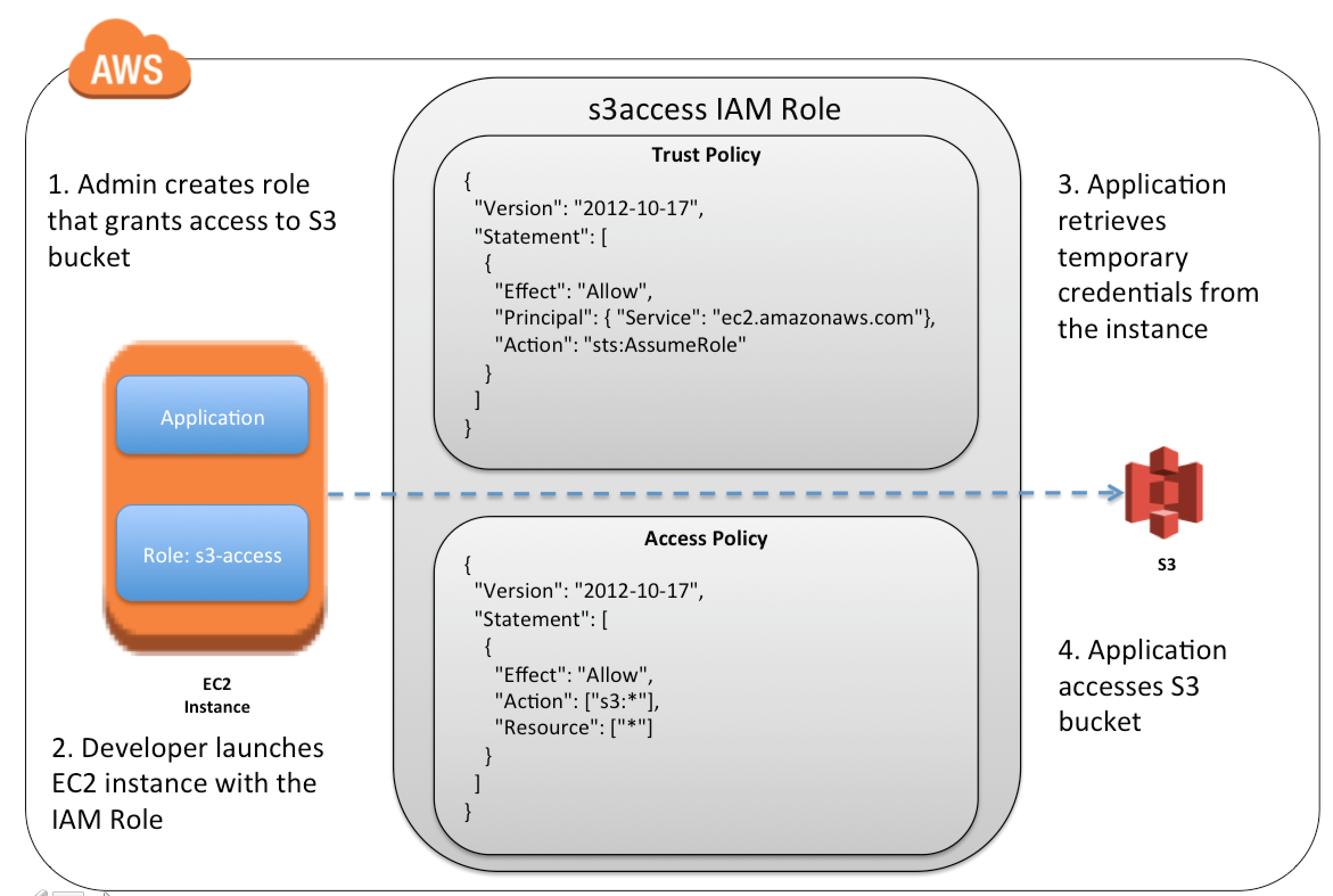
* Security groups are Stateful and they use Connection tracking to track information about traffic to and from the instance.
* This allows responses to inbound traffic to flow out of the instance regardless of outbound security group rules, and vice versa.
* Connection Tracking is maintained only if there is no explicit Outbound rule for an Inbound request (and vice versa)
* However, if there is an explicit Outbound rule for an Inbound request, the response traffic is allowed on the basis of the Outbound rule and not on the Tracking information
* Any existing flow of traffic, that is tracked, is not interrupted even if the rules for the security groups are changed. To ensure traffic is immediately interrupted, use NACL as they are stateless and therefore do not allow automatic response traffic.
* Also, If the instance (host A) initiates traffic to host B and uses a protocol other than TCP, UDP, or ICMP,  the instance’s firewall only tracks the IP address and protocol number for the purpose of allowing response traffic from host B. If host B initiates traffic to your instance in a separate request within 600 seconds of the original request or response, your instance accepts it regardless of inbound security group rules, because it’s regarded as response traffic.
* can be controlled by modifying the security group’s outbound rules to permit only certain types of outbound traffic or using NACL

## IAM with EC2

* IAM policy can be defined to allow or deny a user access to the EC2 resources and actions
* EC2 partially supports resource-level permissions. For some EC2 API actions, you cannot specify which resource a user is allowed to work with for that action; instead, you have to allow users to work with all resources for that action
* **IAM allows to control only what actions a user can perform on the EC2 resources but cannot be used to grant access for users to be able to access or login to the instances**

## EC2 with IAM Role

* EC2 instances can be launched with IAM roles so that the applications can securely make API requests from your instances,
* IAM roles prevents the need to share as well as manage, rotate the security credentials that the applications use
* IAM role can be added to an existing running EC2 instance, as per the [latest enhancement](https://aws.amazon.com/about-aws/whats-new/2017/02/new-attach-an-iam-role-to-your-existing-amazon-ec2-instance/) from AWS,
* EC2 uses an instance profile as a container for an IAM role.
  + Creation of an IAM role using the console, creates an instance profile automatically and gives it the same name as the role it corresponds to.
  + When using the AWS CLI, API, or an AWS SDK to create a role, the role and instance profile needs to be created as separate actions, and they can be given different names.
* To launch an instance with an IAM role, the name of its instance profile needs to be specified.
* An application on the instance can retrieve the security credentials provided by the role from the instance metadata item http://169.254.169.254/latest/meta-data/*iam/security-credentials/*role-name.
* Security credentials are temporary and are rotated automatically and new credentials are made available at least five minutes prior to the expiration of the old credentials.
* Best Practice: **Always launch EC2 instance with IAM role instead of hardcoded credentials**



## AWS Certification Exam Practice Questions

* Questions are collected from Internet and the answers are marked as per my knowledge and understanding (which might differ with yours).
* AWS services are updated everyday and both the answers and questions might be outdated soon, so research accordingly.
* AWS exam questions are not updated to keep up the pace with AWS updates, so even if the underlying feature has changed the question might not be updated
* Open to further feedback, discussion and correction.

1. You launch an Amazon EC2 instance without an assigned AWS identity and Access Management (IAM) role. Later, you decide that the instance should be running with an IAM role. Which action must you take in order to have a running Amazon EC2 instance with an IAM role assigned to it?
   1. Create an image of the instance, and register the image with an IAM role assigned and an Amazon EBS volume mapping.
   2. **Create a new IAM role with the same permissions as an existing IAM role, and assign it to the running instance.**(As per AWS [latest enhancement](https://aws.amazon.com/about-aws/whats-new/2017/02/new-attach-an-iam-role-to-your-existing-amazon-ec2-instance/), this is possible now)
   3. Create an image of the instance, add a new IAM role with the same permissions as the desired IAM role, and deregister the image with the new role assigned.
   4. Create an image of the instance, and use this image to launch a new instance with the desired IAM role assigned (This was correct before, as it was not possible to add an IAM role to an existing instance)
2. What does the following command do with respect to the Amazon EC2 security groups? ec2 revoke-security-group-ingress
   1. Removes one or more security groups from a rule.
   2. Removes one or more security groups from an Amazon EC2 instance.
   3. **Removes one or more rules from a security group**
   4. Removes a security group from our account.
3. Which of the following cannot be used in Amazon EC2 to control who has access to specific Amazon EC2 instances?
   1. Security Groups
   2. **IAM System**
   3. SSH keys
   4. Windows passwords
4. You must assign each server to at least \_\_\_\_\_ security group
   1. 3
   2. 2
   3. 4
   4. **1**
5. A company is building software on AWS that requires access to various AWS services. Which configuration should be used to ensure that AWS credentials (i.e., Access Key ID/Secret Access Key combination) are not compromised?
   1. Enable Multi-Factor Authentication for your AWS root account.
   2. **Assign an IAM role to the Amazon EC2 instance**
   3. Store the AWS Access Key ID/Secret Access Key combination in software comments.
   4. Assign an IAM user to the Amazon EC2 Instance.
6. Which of the following items are required to allow an application deployed on an EC2 instance to write data to a DynamoDB table? Assume that no security keys are allowed to be stored on the EC2 instance. (Choose 2 answers)
   1. **Create an IAM Role that allows write access to the DynamoDB table**
   2. **Add an IAM Role to a running EC2 instance.**(As per AWS [latest enhancement](https://aws.amazon.com/about-aws/whats-new/2017/02/new-attach-an-iam-role-to-your-existing-amazon-ec2-instance/), this is possible now)
   3. Create an IAM User that allows write access to the DynamoDB table.
   4. Add an IAM User to a running EC2 instance.
   5. Launch an EC2 Instance with the IAM Role included in the launch configuration (This was correct before, as it was not possible to add an IAM role to an existing instance)
7. You have an application running on an EC2 Instance, which will allow users to download files from a private S3 bucket using a pre-assigned URL. Before generating the URL the application should verify the existence of the file in S3. How should the application use AWS credentials to access the S3 bucket securely?
   1. Use the AWS account access Keys the application retrieves the credentials from the source code of the application.
   2. Create a IAM user for the application with permissions that allow list access to the S3 bucket launch the instance as the IAM user and retrieve the IAM user’s credentials from the EC2 instance user data.
   3. **Create an IAM role for EC2 that allows list access to objects in the S3 bucket. Launch the instance with the role, and retrieve the role’s credentials from the EC2 Instance metadata**
   4. Create an IAM user for the application with permissions that allow list access to the S3 bucket. The application retrieves the IAM user credentials from a temporary directory with permissions that allow read access only to the application user.
8. A user has created an application, which will be hosted on EC2. The application makes calls to DynamoDB to fetch certain data. The application is using the DynamoDB SDK to connect with from the EC2 instance. Which of the below mentioned statements is true with respect to the best practice for security in this scenario?
   1. **The user should attach an IAM role with DynamoDB access to the EC2 instance**
   2. The user should create an IAM user with DynamoDB access and use its credentials within the application to connect with DynamoDB
   3. The user should create an IAM role, which has EC2 access so that it will allow deploying the application
   4. The user should create an IAM user with DynamoDB and EC2 access. Attach the user with the application so that it does not use the root account credentials
9. Your application is leveraging IAM Roles for EC2 for accessing object stored in S3. Which two of the following IAM policies control access to you S3 objects.
   1. **An IAM trust policy allows the EC2 instance to assume an EC2 instance role.**
   2. **An IAM access policy allows the EC2 role to access S3 objects**
   3. An IAM bucket policy allows the EC2 role to access S3 objects. (Bucket policy is defined with S3 and not with IAM)
   4. An IAM trust policy allows applications running on the EC2 instance to assume as EC2 role (Trust policy allows EC2 instance to assume the role)
   5. An IAM trust policy allows applications running on the EC2 instance to access S3 objects. (Applications can access S3 through EC2 assuming the role)
10. You have an application running on an EC2 Instance, which will allow users to download files from a private S3 bucket using a pre-assigned URL. Before generating the URL the application should verify the existence of the file in S3. How should the application use AWS credentials to access the S3 bucket securely?
    1. Use the AWS account access Keys the application retrieves the credentials from the source code of the application.
    2. Create a IAM user for the application with permissions that allow list access to the S3 bucket launch the instance as the IAM user and retrieve the IAM user’s credentials from the EC2 instance user data.
    3. **Create an IAM role for EC2 that allows list access to objects in the S3 bucket. Launch the instance with the role, and retrieve the role’s credentials from the EC2 Instance metadata**
    4. Create an IAM user for the application with permissions that allow list access to the S3 bucket. The application retrieves the IAM user credentials from a temporary directory with permissions that allow read access only to the application user.