Notes Based on Real Exam Attempts

These notes are created by me based on both my exam attempts. Links to AWS documentation on everything I talk about will be posted under their specific section.

**IAM Conditions:**

**Condition operators-** Are the “verbs” of conditions and specify the type of comparison that IAM performs. Think of these as how IAM decides how to interpret the conditions. Condition operators can be grouped into the following categories:

* String, Numeric, Date and time, Boolean, Binary, IP address, Amazon Resource Name (ARN), ifExists, Null check.
* These are not that useful to know for the exam, but as there is one per condition it will not hurt to learn what they mean.

To better explain this let me show you an IAM policy with a condition statement and what better way to show you than with an example in ACloudGuru and one you may find on the test.



Like I said I will not dive too deep into the concept, but as show below with JSON you know they are in Key value pairs. For condition statements you actually have 3 keys.

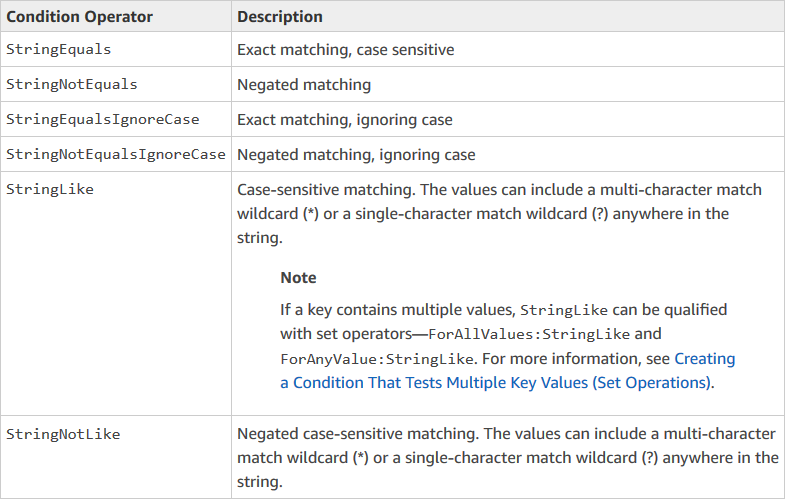
1. **“Condition”** shown to the left of the red box. This is just like the “Effect”, “Action” you know standard JSON.
2. **Condition Operator** Shown by the red box. This determines how IAM will perform the comparison. If you don’t know, in programming languages bool from what I know means True or False. This states that the condition will be based on whether the condition key is true or false.
3. **Condition Keys** give you granular control over when your JSON policy matches or does not match a specific API request. The condition key above is the “aws:SecureTransport” to the right of the red box.

This is the syntax of all condition policies in any resource policy of AWS.

Just like with every other part of policies, condition operators can easily be interpreted by reading the operator. I am going to quickly touch base on a few more things you should know.

**String Condition Operators:**

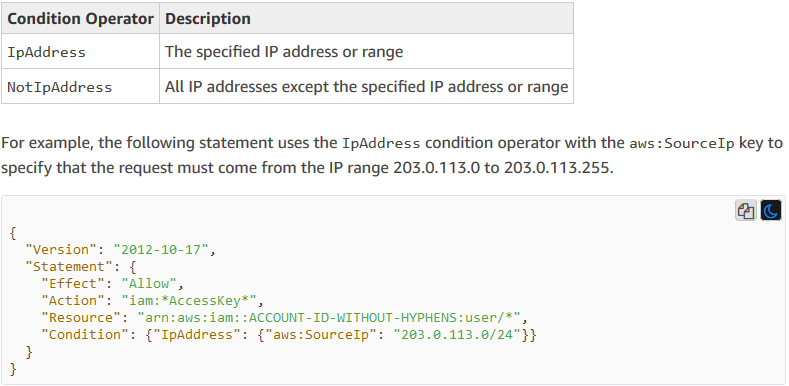
Not every condition operator has only 1 operator you can use. For example with String, which construct condition elements that restrict access based on comparing a key to a string value, can have 6 different operators. All of the operators prefix with String so it shouldn’t be that hard to interpret.



From my experience from both practice exams and the real exam you will only get the StringEquals or StringNotEquals. Just keep in mind their differences because and allow on the effect with a StringNotEquals will allow everything but the string. They might try to trip you up with that.

**IP Address:**

I have seen this condition operator too and has the same tricky principle as the String operator.



Just keep in mind if the IpAddress operator is replace with NotIpAddress, then the condition would deny only the IP address shown below and allow everyone else.

For additional reading on Condition Operators click on this link.

<https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html>

**Condition Keys-** Are the keys that match or don’t match an incoming API request.

* One thing to note is that there are different types of condition keys
  + Global condition keys (Prefixed with aws)
  + S3 condition keys (Prefixed with s3)
  + KMS condition keys (Prefixed with KMS)

And so on.

Condition keys only work with certain condition operators.

**Global Condition Keys to Study:**

* **aws:MultiFactorAuthAge**
* **aws:MultiFactorAuthPresent**
* **aws:SecureTransport**
* **aws:PrincipalOrgID**
* **aws:Referer**
* **aws:SourceVpce**
* **aws:SourceIp**

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\_policies\_condition-keys.html

**KMS Topics:**

**KMS Condition Keys to Study:**

* [kms:GrantIsForAWSResource](https://docs.aws.amazon.com/kms/latest/developerguide/policy-conditions.html#conditions-kms-grant-is-for-aws-resource)
* [kms:ViaService](https://docs.aws.amazon.com/kms/latest/developerguide/policy-conditions.html#conditions-kms-via-service) (Came up 3 times yesterday)

While we still are on a topic of policies I would like to point out a few things with key policies. Knowledge of KMS API calls were huge for my test. ACloudGuru does not cover this at all so I will point out a few API calls to get familiar with. The API calls I am referring too are the Key Usage operations. When someone is granted key usage they are given the ability to

* DescribeKey
* Encrypt
* Decrypt
* GenerateDataKey
* ReEncrypt

The main thing is you should know how the envelope encryption of KMS works. Actual CMK’s can only encrypt 4kb. This isn’t very useful so what KMS does is perform envelope encryption.

**Envelope Encryption-** When you encrypt your data, your data is protected, but you have to protect your encryption key. This is where envelope encryption comes in. The CMK creates 2 data keys when encrypting an object. One is in plaintext and one is an encrypted data key. The plaintext data key is used to encrypt the data and is then discarded, while the encrypted data key is stored with the object and that data key is used to decrypt the object.

What this means is that in order to encrypt objects above 4kb you need to perform the GenerateDataKey operation to generate a data key. Than you use the Encrypt operation to encrypt the data key. To get your data you use the Encrypt operation. Lastly the Describe operation is being used this whole time to describe the key.

https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#enveloping

<https://docs.aws.amazon.com/IAM/latest/UserGuide/list_awskeymanagementservice.html>

**Grants-** Provides permissions instead of using a key policy

* Grants are generally used to provide temp permissions or more granular permissions
* Use grants for more dynamic permission assignments

<https://docs.aws.amazon.com/kms/latest/developerguide/grants.html>

**KMS Data Key Caching-** Data key caching stores data keys and related cryptographic material in a cache.

Uses

* It can reuse data keys
* It generates numerous data keys
* Your cryptographic operations are unacceptably slow, expensive, limited, or resource-intensive

Keep in mind that this is a functionality of the KMS SDKs only.

<https://docs.aws.amazon.com/encryption-sdk/latest/developer-guide/data-key-caching.html>

Here is an extra thing you should study. This will help you understand how KMS integrates with S3. Remember that KMS integrates with ever service differently. In my exam the only questions it tested me on about how KMS integrates was with S3, but it never hurts to learn other services such as DynamoDB or EBS.

<https://docs.aws.amazon.com/kms/latest/developerguide/services-s3.html>

**Amazon Athena:**

**Athena-** An interactive query service that enables you to analyze and query data located in S3 using SQL.

Use Cases:

* Can be used to query log files stored in S3, ELB logs, S3 Access Logs, CloudTrail Logs

In the exam they usually will ask a question on how to find all events or a certain event through you CloudTrail Logs. The short answer is Athena.

**ELB Logs-** Your load balancers provide access logs that capture detailed information about requests sent to your load balancer. These contain things like Clients IP, latencies, and server response.

Use Cases:

* Analyze traffic patterns
* Troubleshoot issues

These logs are stored in S3 and can be used with Athena. I am also bringing these logs up because in a question asking how you can inspect the actual packets this along with VPC Flow Logs were some answers. **You cannot use ELB Logs or VPC Flow Logs to inspect actual packets.**

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html>

<https://docs.aws.amazon.com/athena/latest/ug/what-is.html>

This section is here for extra questions you may get on your test with links to the answers.

**What port should you open to use SES?**

<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/smtp-connect.html>

**How can you setup a write once read many policy in glacier? (WORM)**

<https://aws.amazon.com/blogs/aws/glacier-vault-lock/>

**SSM patch manager.**

<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-patch.html>

**Service Control Policies.**

<https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_scp.html>

**Permission Boundaries.**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_boundaries.html>

**A big portion of the test is understanding how to use roles in different scenarios. This could be cross-account or even ID federation. This link should prove helpful.**

<https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios.html>