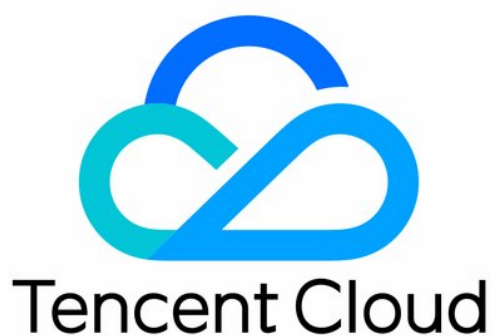


Automatic Speech Recognition

Product Introduction

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Product Introduction

Release Notes

Access Management

Overview

Authorizable Resource Types

Authorization Policy Syntax

Product Introduction

Release Notes

Last updated : 2023-04-24 17:03:35

Tencent Cloud Automatic Speech Recognition (ASR) provides highly cost-effective speech recognition services. It has been widely used by many Tencent businesses such as WeChat, Honor of Kings, and Tencent Video and has implemented multiple use cases, including recording quality inspection, real-time meeting transcription, and voice input method.

Features

Real-time speech recognition

It recognizes real-time audio streams to achieve the effect of instant speech-to-text, which is suitable for real-time audio streaming scenarios such as voice input and phone bot.

Strengths

Massive data accumulation

Based on Tencent's vast social data platform, ASR has accumulated hundreds of thousands of hours of annotated voice data in a rich and diverse corpus, laying a data foundation for a high recognition accuracy.

Industry-leading algorithms

Based on multiple sequential neural network structures (LSTM, Attention Model, and DeepCNN), ASR is trained in the multitask learning method and delivers an industry-leading recognition accuracy together with the T/S approach in general and vertical fields.

Cross-platform support

ASR provides RESTful APIs and SDKs and supports a wide variety of devices and terminals, including smart hardware, mobile application, website, desktop client, and IoT.

Support for Multiple Languages

ASR currently supports speech recognition in Mandarin and English, with more languages to come in the future.

Excellent recognition performance in noisy environment

ASR features robust recognition models, high recognition accuracy, and strong noise resistance. It can recognize audio information from noisy environments with no need of noise reduction processing.

Well proven capabilities

ASR has been fully verified by Tencent's internal businesses such as WeChat, Tencent Video, and Honor of Kings and has implemented many external use cases for customers in the internet, finance, education, and other industries, serving billions of users every day with a stable performance.

Use Cases

Voice input method

ASR makes smart voice input possible through real-time speech recognition, which saves users the input time and improves the input experience.

Meeting minutes

Audio information in conferences, court trials, and interviews can be converted to text by the real-time speech recognition service, which reduces human recording costs and improves the efficiency.

Call quality inspection

Rep conversations can be converted to text by the real-time speech recognition service, which comprehensively covers the content and improves the efficiency of quality inspection.

Access Management

Overview

Last updated : 2022-07-06 09:48:22

If you have multiple users of ASR operations, and they all share your Tencent Cloud account access key, you may face the following problems:

- The risk of your key being compromised is high since multiple users are sharing it.
- Your users might introduce security risks from maloperations due to the lack of user access control.

In this case, you can avoid the above problems by allowing different users to manage different services through sub-accounts. By default, sub-accounts don't have permissions to use ASR or relevant resources. Therefore, you need to create a policy to grant different permissions to sub-accounts.

Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage and control access permissions of your Tencent Cloud resources. Using CAM, you can create, manage, and terminate users (groups), and control the Tencent Cloud resources that can be used by the specified user through identity and policy management.

When using CAM, you can associate a policy with a user or user group to allow or forbid them to use specified resources to complete specified tasks. For more information on CAM policies, see [Element Reference](#). For more information on how to use CAM policies, see [Policy](#).

You can skip this section if you don't need to manage permissions to ASR resources for sub-accounts. This will not affect your understanding and use of the other sections of the document.

Getting Started

A CAM policy must authorize or deny the use of one or more ASR operations. At the same time, it must specify the resources that can be used for the operations (which can be all resources or partial resources for certain operations). A policy can also include the conditions set for the manipulated resources.

Some APIs of ASR use API-level permissions, for which you don't need to specify authorization for certain resources. Some other APIs use resource-level permissions, for which you can specify authorization for certain resources.

Basic policy structure	Authorization Policy Syntax
Operation definition in a policy	Authorization Policy Syntax
Resource definition in a policy	Authorization Policy Syntax

ASR authorization granularity	Authorizable Resource Types
-------------------------------	---

Authorizable Resource Types

Last updated : 2022-07-06 09:48:36

ASR supports authorization by tag.

- Authorization by tag: You can tag resources and grant sub-accounts permissions to manage resources with particular tags.

List of APIs Supporting API-Level Authorization

For API operations that support operation-level permission control, you can still authorize a user to perform it, but you must specify `*` as the resource element in the policy statement.

API	Description
CreateAsrUser	Activates the ASR service
DescribeResource	Queries resource package
DescribeStatistics	Queries statistics
DescribeUserStatus	Queries ASR service activation status
RealtimeRecognition	Performs real-time recognition
StartStreamTranscription	Performs global speech recognition

Authorization Policy Syntax

Last updated : 2022-07-06 09:48:56

Policy Syntax

CAM policy:

```
{
  "version": "2.0",
  "statement": [
    {
      "effect": "effect",
      "action": ["action"],
      "resource": ["resource"],
      "condition": {"key": {"value": {}}}
    }
  ]
}
```

- **version** is required. Currently, only the value "2.0" is allowed.
- **statement** describes the details of one or more permissions. It contains a permission or permission set of multiple other elements such as `effect` , `action` , `resource` , and `condition` . One policy has only one `statement` .
- **action** is required. It describes the allowed or denied action (operation). An operation can be an API (prefixed with "name") or a feature set (a set of specific APIs prefixed with "permid").
- **resource** is required. It describes the details of authorization. A resource is described in a six-segment format. Detailed resource definitions vary by product. For more information on how to specify resources, see the product documentation corresponding to the resource statement you are writing.
- **condition** is optional. It describes the condition for the policy to take effect. A condition consists of operator, action key, and action value. A condition value can be a client IP.
- **effect** is required. It describes the result of a statement. The result can be an "allow" or an explicit "deny".

ASR Operations

In a CAM policy statement, you can specify any API operation from any service that supports CAM. APIs prefixed with `name/asr:` should be used for ASR, such as `name/asr:CreateModel` or `name/asr:CreateAsrVocab` .

- To specify multiple operations in a single statement, separate them with commas as shown below:

```
"action": ["name/asr:action1", "name/asr:action2"]
```

You can also specify multiple operations by using a wildcard. For example, you can specify all operations beginning with "Describe" in the name as shown below:

```
"action": ["name/cvm:Describe*"]
```

- To specify all operations in ASR, use the `*` wildcard as shown below:

```
"action": ["name/asr:*"]
```

ASR Resource Path

Each CAM policy statement is resource-specific with a resource path as shown below:

```
qcs:project_id:service_type:region:account:resource
```

- project_id** describes the project information, which is only used to enable compatibility with legacy CAM logic and can be left empty.
- service_type** describes the product abbreviation such as `asr`.
- region** describes the region information, which is not required for ASR.
- account** describes the root account of the resource owner, such as `uin/164256472`.
- resource** describes the detailed resource information of each product, such as `model/model_id1` or `model/*`.

For example, you can use a specific self adaptive learning model (15b96676edb211ea9301b49691037310) by specifying it in the statement as shown below:

```
"resource": [ "qcs::asr::uin/164256472:model/15b96676edb211ea9301b49691037310" ]
```

You can also use the `*` wildcard to specify all self adaptive learning models that belong to a specific account as shown below:

```
"resource": [ "qcs::asr::uin/164256472:model/*" ]
```

If you want to specify all resources or a specific API operation supports only API-level permission control, you can use the `*` wildcard in the `resource` element as shown below:

```
"resource": ["*"]
```

To specify multiple resources in one policy, separate them with commas. In the following example, two resources are specified:

```
"resource": ["resource1", "resource2"]
```