资源治理服务

最佳实践

文档版本	01
发布日期	2025-01-22

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 使用 Config 创建合规规则

示例简介

该示例展示了如何通过Java版本SDK为Config服务创建合规规则和查询合规规则。

合规规则通过指定合规策略和合规策略所应用的范围（如：在某一区域的某些资源）来构成合规规则。

开发前准备

1. 获取华为云开发工具包（SDK），您也可以查看安装JAVA SDK。
2. 您需要拥有华为云账号以及该账号对应的 Access Key（AK）和 Secret Access Key（SK），请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见**访问密钥**。
3. **配置审计 Config SDK**，支持 Java JDK 1.8 及其以上版本。

安装 SDK

您可以通过Maven方式获取和安装SDK，您只需要在Java项目的pom.xml文件中加入相应的依赖项即可。 具体的SDK版本号请参见**SDK开发中心** 。

```
<dependency>
    <groupId>com.huaweicloud.sdk</groupId>
    <artifactId>huaweicloud-sdk-config</artifactId>
    <version>{sdk-version}</version>
</dependency>
```

代码示例

```
public class CreatePolicyAssignmentDemo {
    public static void main(String[] args) {
        // 认证用的ak和sk直接写到代码中有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");
        String regionId = "<region id>";
        HttpConfig config = HttpConfig.getDefaultHttpConfig();
        config.withIgnoreSSLVerification(true);

        ICredential auth = new GlobalCredentials().withAk(ak).withSk(sk);
        ConfigClient client = ConfigClient.newBuilder().withHttpConfig(config).withCredential(auth)
            .withRegion(ConfigRegion.valueOf(regionId)).build();
```



```
        CreatePolicyAssignmentsRequest createRequest = new CreatePolicyAssignmentsRequest()
            .withBody(new
PolicyAssignmentRequestBody().withPolicyAssignmentType(PolicyAssignmentRequestBody.PolicyAssignment
TypeEnum.BUILTIN)
                .withName("<Your policy_assignment_name>").withDescription("<Your
policy_assignment_description>")
                .withPolicyFilter(new PolicyFilterDefinition()).withPolicyDefinitionId("<Your
policy_definition_id>"));

        try {
            CreatePolicyAssignmentsResponse createResponse = client.createPolicyAssignments(createRequest);
            System.out.println(createResponse.toString());
            ShowPolicyAssignmentRequest showPolicyAssignmentRequest = new
ShowPolicyAssignmentRequest()
                .withPolicyAssignmentId(createResponse.getId());
            ShowPolicyAssignmentResponse showResponse =
client.showPolicyAssignment(showPolicyAssignmentRequest);
            System.out.println(showResponse.toString());
        } catch (ConnectionException | RequestTimeoutException | ServiceResponseException ex) {
            System.out.println(ex);
        }
    }
}
```

返回结果示例

```
class CreatePolicyAssignmentsResponse {
    policyAssignmentType: "policyAssignmentType"
    id: "id"
    name: "name"
    description: "description"
    policyFilter: class PolicyFilterDefinition {}
    period: "period"
    state: "state"
    created: "created"
    updated: "updated"
    policyDefinitionId: "policyDefinitionId"
    customPolicy: "customPolicy"
    parameters: {}
    createdBy: "createdBy"
}

class ShowPolicyAssignmentResponse {
    policyAssignmentType: "policyAssignmentType"
    id: "id"
    name: "name"
    description: "description"
    policyFilter: class PolicyFilterDefinition {}
    period: "period"
    state: "state"
    created: "created"
    updated: "updated"
    policyDefinitionId: "policyDefinitionId"
    customPolicy: "customPolicy"
    parameters: {}
    createdBy: "createdBy"
}
```

参考

更多信息请参考**资源合规概述**。

修订记录

发布日期	文档版本	修订说明
2024-12-25	1.0	文档首次发布

2 使用 Config 查询资源详情、资源关系和资源历史

示例简介

该示例展示了如何通过Java版本SDK查询资源详情、资源关系和资源历史。

1. **资源清单** 默认展示资源的部分属性，如果您需要查看某个资源的资源详情，可按如下操作查看。
2. **资源关系** 记录了您在华为云上的不同资源之间的关联情况。
3. **资源历史** 是过去某段时间内资源不同状态的集合。对接服务上报Config的资源属性和资源关系的变化，都会在资源时间线中生成一条记录，该记录会包含资源变更情况的详细信息，默认的保存期限为7年。

开发前准备

1. 获取华为云开发工具包（SDK），您也可以查看安装JAVA SDK。
2. 您需要拥有华为云账号以及该账号对应的 Access Key（AK）和 Secret Access Key（SK），请在华为云控制台“我的凭证 > 访问密钥”页面上创建和查看您的AK/SK。具体请参见**访问密钥**。
3. **配置审计 Config SDK**，支持 Java JDK 1.8 及其以上版本。

安装 SDK

您可以通过Maven方式获取和安装SDK，您只需要在Java项目的pom.xml文件中加入相应的依赖项即可。 具体的SDK版本号请参见**SDK开发中心** 。

```
<dependency>
    <groupId>com.huaweicloud.sdk</groupId>
    <artifactId>huaweicloud-sdk-config</artifactId>
    <version>{sdk-version}</version>
</dependency>
```

代码示例

```
public class ShowResourceRelationDemo {
    public static void main(String[] args) {
        // 认证用的ak和sk直接写到代码中有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
        // 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。
```



```
        String ak = System.getenv("HUAWEICLOUD_SDK_AK");
        String sk = System.getenv("HUAWEICLOUD_SDK_SK");
        String regionId = "<region id>";
        HttpConfig config = HttpConfig.getDefaultHttpConfig();
        config.withIgnoreSSLVerification(true);

        ICredential auth = new GlobalCredentials().withAk(ak).withSk(sk);
        ConfigClient client = ConfigClient.newBuilder().withHttpConfig(config).withCredential(auth)
            .withRegion(ConfigRegion.valueOf(regionId)).build();

        try {
            String resourceId = "<resource id>";
            // 查询资源详情
            ShowResourceDetailRequest resourceDetailRequest = new ShowResourceDetailRequest()
                .withResourceId(resourceId);
            System.out.println(client.showResourceDetail(resourceDetailRequest));
            // 查询资源关系
            ShowResourceRelationsRequest resourceRelationsRequest = new ShowResourceRelationsRequest()
                .withResourceId(resourceId)
                .withDirection(ShowResourceRelationsRequest.DirectionEnum.IN);
            System.out.println(client.showResourceRelations(resourceRelationsRequest).toString());
            // 查询资源历史
            ShowResourceHistoryRequest resourceHistoryRequest = new ShowResourceHistoryRequest()
                .withResourceId(resourceId);
            System.out.println(client.showResourceHistory(resourceHistoryRequest).toString());
        } catch (ConnectionException | RequestTimeoutException | ServiceResponseException ex) {
            System.out.println(ex);
        }
    }
}
```

返回结果示例

```
class ShowResourceDetailResponse {
    id: 81fi****a864
    name: zh****ng
    provider: iam
    type: users
    regionId: global
    projectId:
    projectName:
    epId: 0
    epName: default
    checksum: 522u****e689
    created: 2023-09-18T12:56:30.000Z
    updated: 2023-09-18T12:56:30.000Z
    provisioningState: Succeeded
    state: Normal
    tags: {}
    properties: {pwd_status=false, pwd_strength=high, group_list=[f588****54c5], role_list=[],
last_login_time=2023-09-18T12:57:45Z, virtual_mfa_device=false, login_protect={enabled=false},
credentials=[], policy_list=[], access_mode=default, is_root_user=false, enabled=true}
}

class ShowResourceRelationsResponse {
    relations: [class ResourceRelation {
        relationType: contains
        fromResourceType: iam.groups
        toResourceType: iam.users
        fromResourceId: f587****54c5
        toResourceId: 81fa****a864
    }]
    pageInfo: class PageInfo {
        currentCount: 1
        nextMarker: null
    }
}

class ShowResourceHistoryResponse {
```

```
   items: [class HistoryItem {
      domainId: 39f4****ea39
      resourceId: 81fa****a864
      resourceType: iam.users
      captureTime: 2023-09-21T15:39:27.632Z
      status: ResourceChanged.CREATE
      relations: [class ResourceRelation {
         relationType: isContainedIn
         fromResourceType: iam.users
         toResourceType: iam.groups
         fromResourceId: 81fa****a864
         toResourceId: b04e****8dd2
      }]
      resource: class ResourceEntity {
         id: 81fa****a864
         name: zh****ng
         provider: iam
         type: users
         regionId: global
         projectId:
         projectName:
         epId: 0
         epName: default
         checksum: 00ce****f053
         created: 2023-09-18T12:56:30Z
         updated: 2023-09-18T12:56:30Z
         provisioningState: Succeeded
         state: null
         tags: {}
         properties: {pwd_status=false, pwd_strength=high, group_list=[b04e****8dd2], role_list=[],
virtual_mfa_device=false, login_protect={enabled=false}, credentials=[], policy_list=[], access_mode=default,
enabled=true}
      }
   }]
   pageInfo: class PageInfo {
      currentCount: 1
      nextMarker: null
   }
}
```

参考

更多信息请参考**查看资源历史**。

修订记录

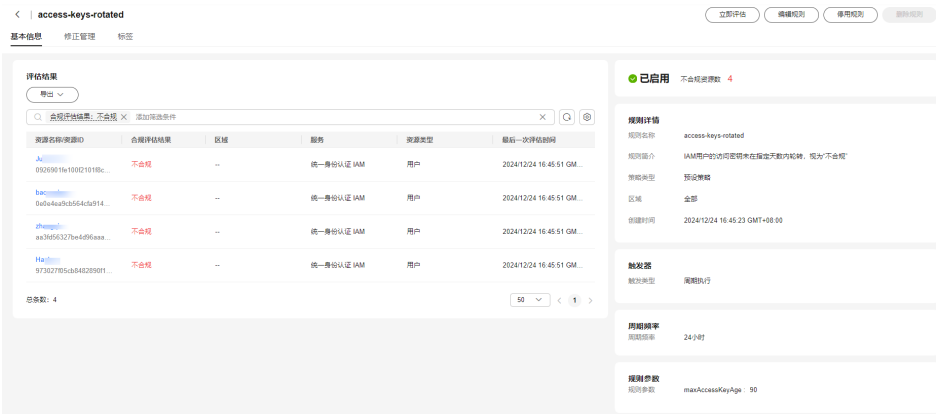
发布日期	文档版本	修订说明
2024-12-25	1.0	文档首次发布

3 通过 CES 实现不合规资源的告警通知

当资源配置不合规时，配置审计会将不合规资源自动投递到云监控服务（CES）。您可以在云监控服务中查询告警记录，还可以基于云监控服务发送消息通知。

应用场景

您在配置审计控制台通过预设策略“IAM用户的AccessKey在指定时间内轮换”创建一条合规规则。配置审计自动评估当前账号下所有IAM用户，其中部分IAM用户的合规结果为**不合规**，如下图所示。



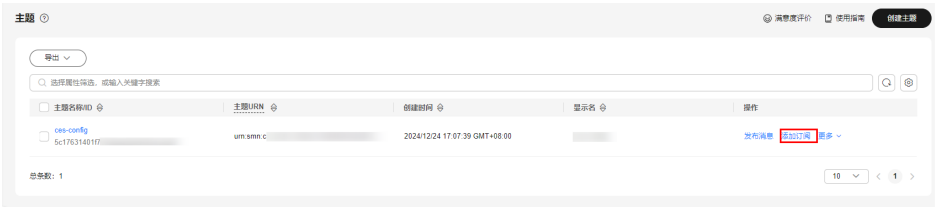
步骤一：创建规则

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在规则页面，单击“添加规则”。
4. 在基础配置页面，选择预设策略中的“**IAM用户的AccessKey在指定时间内轮换**”，单击“下一步”。
5. 在规则参数页面，选择评估全部区域的资源，AccessKey轮转时间使用默认值，单击“下一步”。
6. 确认规则配置符合预期，单击“提交”，完成规则创建。

在资源合规的规则页签，您可以查看该规则对IAM资源的检测结果。

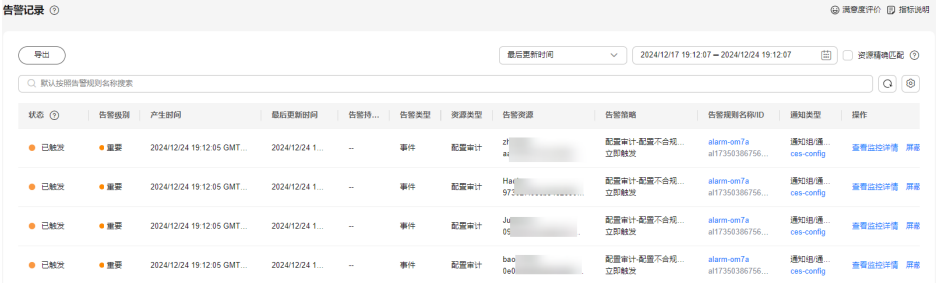
步骤二：设置消息通知主题

1. 登录**消息通知服务控制台**。
2. 在左侧导航栏，选择主题管理。
   a. 在主题页面，右上角单击“创建主题”。
   b. 设置合适的主体名称和显示名，单击“确定”，完成主题创建。
3. 以短信方式为例，添加订阅。
   a. 单击“添加订阅”。
   b. 选择协议为“短信”。
   c. 订阅终端中输入需要通知的有效手机号码。
   d. 单击“确定”。
4. 在订阅页签，点击“请求订阅”。
5. 添加订阅的手机上，会收到确认短信，按照短信指示完成订阅确认。



步骤三：设置云监控服务的告警通知

1. 登录**云监控服务控制台**，国内站用户选择北京四，国际站用户选择新加坡。
2. 在左侧导航栏，选择告警。
3. 在告警规则页面，单击“创建告警规则”。
   a. 编辑告警规则名称。
   b. 告警类型选择“事件”。
   c. 事件类型选择“系统事件”。
   d. 事件来源选择“配置审计”。
   e. 触发规则选择“自定义创建”。
   f. 打开发送通知，选择主题订阅，通知对象选择步骤二创建的SMN主题，其它配置使用默认值。
   g. 触发条件勾选“出现告警”。
   h. 点击“立即创建”，完成告警规则的创建。
4. 告警规则创建完成后，如果再评估到不合规的资源，则会通过短信方式接受到不合规的消息通知，并在告警记录中查询到相关告警。



相关文档

- 查看事件监控数据
- 创建事件监控的告警通知

4 通过高级查询自定义查询资源并下载

本文为您介绍如何通过配置审计的高级查询功能，对当前资源数据执行基于属性的自定义查询，并下载查询结果。

应用场景

Config提供高级查询能力，通过使用ResourceQL语法来自定义查询云资源。方便上云企业根据资源属性灵活导出数据。

操作步骤

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择高级查询。
3. 点击自定义查询页签，单击右上角的“新建查询”。
4. 输入如下内容查询未使用的EVS磁盘，单击“运行”。

```
SELECT *
FROM tracked_resources
WHERE provider = 'evs'
    AND type = 'volumes'
    AND properties.status != 'in-use'
```

5. 单击“导出”，选择CSV或JSON格式，将查询结果导出到本地。

相关文档

- 高级查询语法概览
- 高级查询支持的资源属性

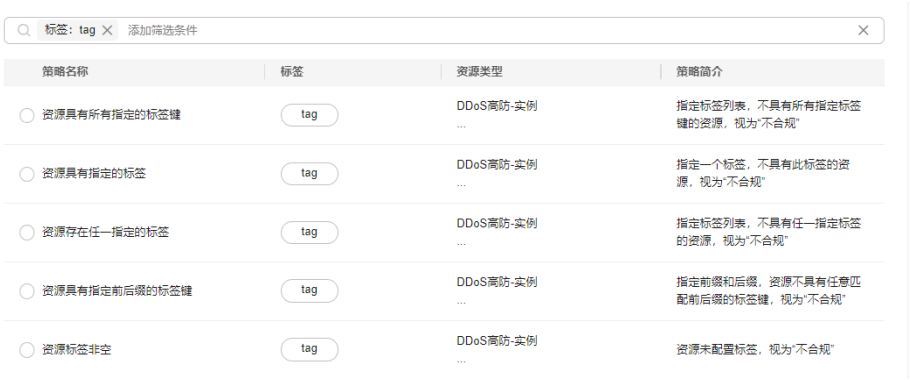
5 使用 Config 查找未绑定指定标签的资源

本文为您介绍如何通过配置审计服务查询未绑定指定标签的资源。

应用场景

企业上云后，云上创建的资源不断增加，有些大型企业资源数量达到十万、百万级别，一个账号内存在大量资源，企业需要进行分类管理。华为云推荐您使用标签对资源进行标记，进而实现资源的分组分类。通过标签对资源的业务归属、财务归属等资源属性进行标记，例如：按所属部门、地域或项目等。

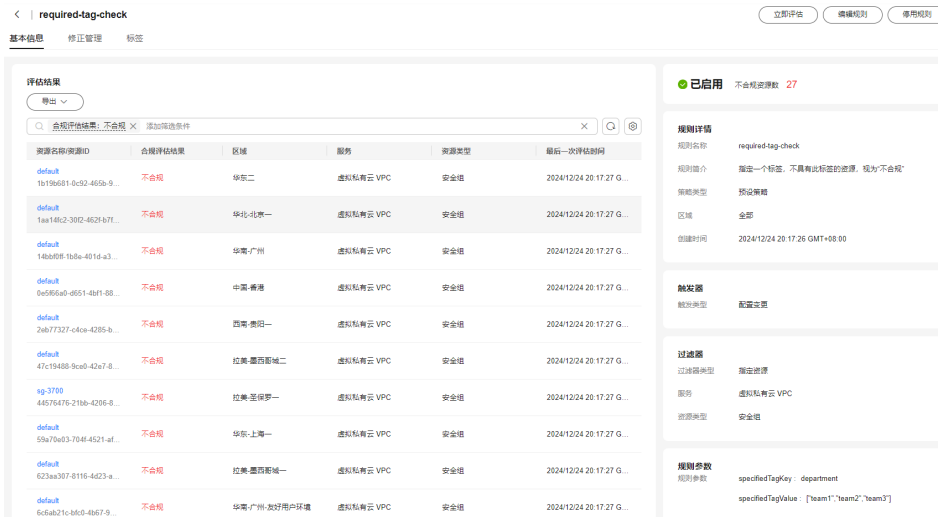
配置审计服务通过资源合规能力，可以帮助您识别未进行正确标签标记的资源。



操作步骤

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在规则页面，单击“添加规则”。
4. 在基础配置页面，选择预设策略中的“**资源具有指定的标签**”，单击“下一步”。
5. 在规则参数页面，选择服务为“虚拟私有云”、资源类型为“安全组”、区域为“全部”。
6. 设置规则参数：标签键为department，标签值列表为["team1", "team2", "team3"]。
7. 确认规则配置符合预期，单击“提交”，完成规则创建。

在资源合规的规则页签，您可以查看该规则对VPC安全组资源的检测结果。



标签审计的预设策略列表

策略名称	策略描述
required-all-tags	指定标签列表，不具有所有指定标签键的资源，视为“不合规”。
required-tag-exist	指定标签列表，不具有任一指定标签键的资源，视为“不合规”。
resource-tag-key-prefix-suffix	指定标签键的前缀和后缀，资源不具有任意匹配前后缀的标签键，视为“不合规”。
resource-tag-not-empty	资源未配置标签，视为“不合规”。
required-tag-check	指定一个标签，不具有此标签的资源，视为“不合规”。

6 通过 Config 实现资源的多维度合规审计

当企业进行云上资源合规落地时，通常会面临合规标准无法统一的问题，例如：

- 生产环境和测试环境的资源安全要求不同；
- 不同地区的法规不同，导致对资源的规范要求不同。

同时，企业内所有账号也需要配置统一的安全基准要求。配置审计通过丰富的托管规则，帮助企业在资源合规落地过程中针对不同场景，设置不同的合规策略。

基于资源标签

前提条件：请确保您已为目标资源绑定标签。具体操作，请参见**标签设计原则和命名示例**。

具体场景：根据开发环境对资源进行分类，可以实现对不同环境资源的不同审计。假设您已为生产环境中的所有资源绑定标签Env:Prod，为测试环境中的所有资源绑定标签Env:Test，标签值您可以根据实际情况自行定义，能区分不同环境中的资源即可。

操作流程：

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在规则页面，单击“添加规则”。
4. 选择您需要的预设策略，如“**ECS实例的镜像名称在指定的范围**”，单击“下一步”。
5. 在规则参数页面，保持默认的资源范围，区域选择“全部”。
6. 单击“过滤范围”，选择标签，标签键和标签值填入 Env 和 Prod。

7. 完成规则创建，则合规规则只会评估生产环境中的该类型资源。
8. 返回规则列表页面，查看新建合规规则的评估结果。

基于资源区域

具体场景：因为不同地区有不同的法律法规要求，假设您不希望您海外的OBS桶内的数据被公开访问，则可以按照如下操作步骤配置合规规则。

操作流程：

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在规则页面，单击“添加规则”。
4. 选择预设策略“**OBS桶禁止公开读**”，单击“下一步”。
5. 在规则参数页面，保持默认的资源范围，区域选择您期望检测的区域，如“亚太-新加坡”。

6. 完成规则创建，则合规规则只会评估“亚太-新加坡”的OBS桶。
7. 返回规则列表页面，查看新建合规规则的评估结果。

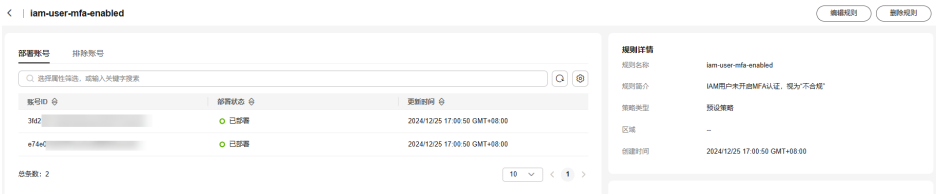
统一下发到组织

前提条件：请确保您的账号已经加入组织，且为组织管理员或者Config服务的委托管理员，请参考**组织概述**和**添加、查看和取消委托管理员**。

具体场景：多账号场景下，可以通过**组织合规规则**方式，将企业的合规要求部署到组织内的所有成员账号。下发组织合规规则的账号一般为企业的安全管理员账号，不承载具体的业务。

操作流程：

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在组织规则页面，单击“添加规则”。
4. 选择您需要的预设策略，如“**IAM用户开启MFA**”，单击“下一步”。
5. 在规则参数页面，保持默认的资源范围，单击“下一步”，完成组织规则创建。
6. 返回组织规则列表页面，查看新建的组织合规规则的部署情况。



统一查询组织成员的合规结果

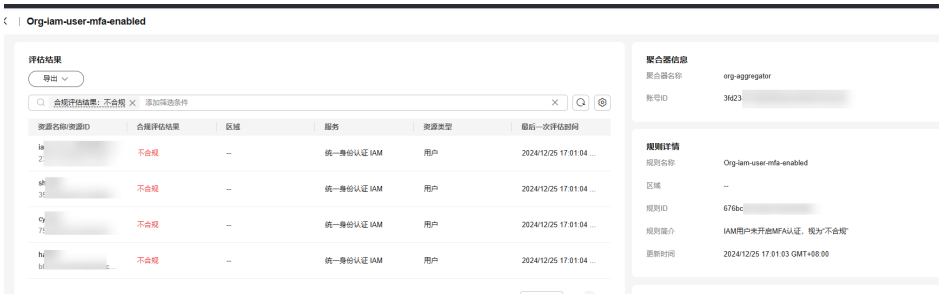
前提条件：请确保您的账号已经加入组织，且为组织管理员或者Config服务的委托管理员，请参考 **组织概述**和**添加、查看和取消委托管理员**。

具体场景：企业的安全管理员，能通过**聚合器**的能力，查询到组织内各个成员账号部署的合规规则，并查询到各个成员账号的不合规资源情况。

操作流程：

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源聚合器。
3. 在聚合器页面，单击“创建聚合器”。
4. 勾选“允许数据复制”，命名聚合器，并选择源类型为“添加组织”，完成聚合器创建。
5. 在聚合器的规则页面，选择聚合器，则可以看到聚合到的各账号的规则情况。

6. 单击规则名称，则可以看到该账号的这条规则所评估的资源的合规性情况。



常见问题

控制台为什么没有组织规则的页面?

账号是组织管理员，或加入组织并成为Config服务的委托管理员，才能看到该页面。

组织规则下发后，状态为什么是部署异常?

是因为这些组织成员账号未开启资源记录器导致的部署异常。

Config服务的相关功能均依赖于资源记录器收集的资源数据，因此账号必须开启资源记录器才可正常使用合规规则和合规规则包功能。

在组织合规规则和组织合规规则包场景下，除下发规则和合规规则包的组织管理员或Config服务的委托管理员之外，所部署的组织成员账号也需要开启资源记录器，才能正常下发合规规则和合规规则包。

7 通过 Config 实现资源自动化管理

本文为您介绍如何通过配置审计服务的资源评估和合规修正能力，自动化实现不合规资源的发现和修正。通过该流程，可以确保云上任意用户，无论有意还是无意的行为，造成了资源不合规，都可以在几分钟内被该流程自动修复，从而保证了云上的资源安全。

应用场景

具体场景：用户在云上的OBS桶，应该使用桶策略避免客户端误使用HTTP协议进行OBS业务操作，详见**确保您的数据在传输到OBS过程中不被窃取和篡改**。

操作流程

创建合规规则：

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在规则页面，单击“添加规则”。
4. 选择预设策略“**OBS桶策略授权行为使用SSL加密**”，单击“下一步”。
5. 在规则参数页面，保持默认的资源范围，区域选择“全部”，完成规则创建。
6. 返回规则列表页面，查看新建合规规则的评估结果。

配置修正函数：

以python方式为例，为您介绍如何通过FunctionGraph实现资源的自动修正。

1. 登录**函数工作流控制台**。
2. 在左侧导航栏，选择函数。
3. 在函数列表页面，单击“创建函数”。

4. 函数类型选择“事件函数”，运行时选择“Python3.9”，选择一个合适的委托，至少包含如下权限：

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "obs:bucket:PutBucketPolicy",
        "obs:bucket:GetBucketPolicy",
        "rms:resources:get"
      ]
    }
  ]
}
```

5. 完成函数创建后，为函数添加“huaweicloudsdk_obs”和“huaweicloudsdkconfig”两个依赖包。

6. 在“index.py”中贴入如下代码。

```
import json

from obs.client import ObsClient
from huaweicloudsdkcore.auth.credentials import GlobalCredentials
from huaweicloudsdkconfig.v1.region.config_region import ConfigRegion
from huaweicloudsdkconfig.v1.config_client import ConfigClient
from huaweicloudsdkconfig.v1 import ShowResourceDetailRequest

def get_resource_region(context, domain_id, resource_id):
    auth = GlobalCredentials(
        ak=context.getSecurityAccessKey(),
        sk=context.getSecuritySecretKey(),
        domain_id=domain_id
    ).with_security_token(context.getSecurityToken())
    client = ConfigClient.new_builder() \
        .with_credentials(credentials=auth) \
        .with_region(region=ConfigRegion.value_of(region_id="cn-north-4")) \
        .build()
    resource = client.show_resource_detail(ShowResourceDetailRequest(resource_id)).to_json_object()
    return resource.get("region_id")

def getBucketPolicy(obsClient, bucket_name):
    resp = obsClient.getBucketPolicy(bucket_name)
    if resp.status < 300:
        print("Get Bucket Policy Succeeded")
        return resp.body.policyJSON
    if resp.status == 404 and resp.errorCode == "NoSuchBucketPolicy":
        print("NoSuchBucketPolicy")
        return "{\"Statement\": []}"
    assert False, f"Get Bucket Policy Failed: {resp.errorCode} | {resp.errorMessage}"

def ensurePolicySSL(obsClient, bucket_name, policy):
    policy["Statement"] = policy["Statement"] + [{
        "Sid": "ensure_secure_transport",
        "Effect": "Deny",
        "Principal": {"ID": ["*"]},
        "Action": ["*"],
```



```
        "Resource": [bucket_name, bucket_name + "/*"],
        "Condition": {"Bool": {"g:SecureTransport": ["false"]}}
    }]
    resp = obsClient.setBucketPolicy(bucket_name, policy)
    if resp.status < 300:
        print("Set Bucket Policy Succeeded")
    else:
        print(policy)
        assert False, f"Set Bucket Policy Failed: {resp.errorCode} | {resp.errorMessage}"

def handler(event, context):
    domain_id = event.get("domain_id")
    bucket_name = event.get("bucket_name")
    print("domain_id", domain_id)
    print("bucket_name", bucket_name)

    region_id = get_resource_region(context, domain_id, bucket_name)
    print("region_id", region_id)

    server = f"https://obs.{region_id}.myhuaweicloud.com"
    obsClient = ObsClient(
        access_key_id=context.getSecurityAccessKey(),
        secret_access_key=context.getSecuritySecretKey(),
        server=server,
        security_token=context.getSecurityToken()
    )
    policy = getBucketPolicy(obsClient, bucket_name)
    policy = json.loads(policy)
    ensurePolicySSL(obsClient, bucket_name, policy)
    obsClient.close()
```

7. （可选）在函数的设置页面适当修改函数的“执行超时时间”和“内存”，并启用日志配置。强烈建议您执行该项操作，否则资源修正的行为有可能会失败，且您无法通过日志记录定位函数执行的错误原因。

配置合规修正：

1. 登录**配置审计控制台**。
2. 在左侧导航栏，选择资源合规。
3. 在规则页面，单击规则名称。
4. 进入规则详情页面，选择“修正管理”，单击“修正配置”。
5. 选择“手动修正”或“自动修正”，重试时间和重试次数使用默认值。
6. 选择“FGS模版”，选中前一步中所配置的函数。
7. 依赖于资源类型设置为“bucket_name”，参数中键和值分别输入domain_id 和账号ID的值。
8. 单击“保存”，完成合规修正的配置。

资源的手动修正：

如果您在上一步的修正方法中选择“手动修正”，则需要进行如下操作。

1. 重新进入修正配置的页面。
2. 在资源范围页面勾选您需要处理的资源：
   - 如果资源属于您认为需要处理的资源，则单击“执行修正”。
   - 如果资源属于您认为不需要处理的资源，则单击“加入修正例外”。
3. 登录对象存储服务控制台，进入您刚才执行修正的OBS桶的详情页。
4. 进入权限控制的桶策略页面，确认桶策略内容已经被更新。

常见问题

手动修正和自动修正的区别?

如果配置为手动修正，则需要用户主动查询不合规资源，并执行修正；如果配置为自动修正，则Config服务会自动为该合规规则下的所有不合规资源执行修正行为。

强烈建议您在第一次配置修正时，选择手动修正。手动修正可以防止资源被修改导致用户业务中断。

在您处理完所有存量的不合规资源后，再将修正配置修改为自动修正，此后所有新增的不合规资源都会被自动修复，而且不需要人为干预。

执行修正后，为什么资源未成功被修正?

这通常是因为您配置在函数工作流服务的代码存在问题，也可能是因为您赋予函数工作流的权限不足。请前往函数工作流的监控页面查询具体的失败日志。

资源修正成功后，为什么资源在合规规则页面还是不合规?

资源修正完成后，资源的变更行为通常会在5分钟之内通知Config服务，规则会自动触发该资源的合规评估并生成最新的资源合规结果。

