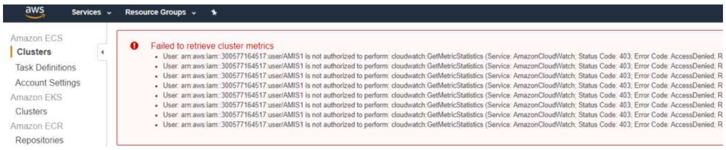
Policies in AWS (2)

0

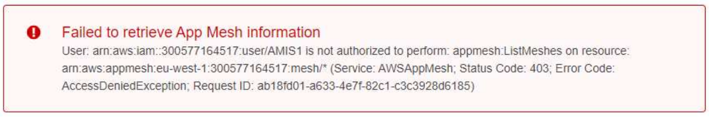
BY FREDERIQUE RETSEMA ON MARCH 8, 2020 AWS, CLOUD, DEVOPS, SECURITY

Yesterday I published a blog about AWS policies. We used the IAM wizard to create a policy. When you try to use this policy with the users we created, you will get errors like these when you go to ECS, and try to create (for example) an ECS-cluster:



Failed to retrieve cluster metrics

- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R
- User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: cloudwatch:GetMetricStatistics (Service: AmazonCloudWatch; Status Code: 403; Error Code: AccessDenied; R

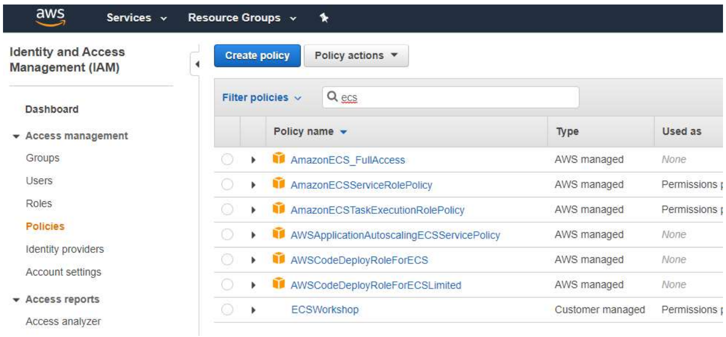


Failed to retrieve App Mesh information

User: arn:aws:iam::300577164517:user/AMIS1 is not authorized to perform: appmesh:ListMeshes on resource: arn:aws:appmesh:eu-west-1:300577164517:mesh/* (Service: AWSAppMesh; Status Code: 403; Error Code: AccessDeniedException; Request ID: ab18fd01-a633-4e7f-82c1-c3c3928d6185)

This is not as strange as it might seem: ECS uses other AWS services to do its task. But fortunately, there is a default AWS policy to grant all permissions for ECS.

Go to IAM, Policies and search for ECS. We need an ECS role where we have all ECS permissions, so click on AmazonECS_FullAccess.



ecs

		Policy name	Type	Used as
○	▸	AmazonECS_FullAccess	AWS managed	None
○	▸	AmazonECSServiceRolePolicy	AWS managed	Permissions
○	▸	AmazonECSTaskExecutionRolePolicy	AWS managed	Permissions
○	▸	AWSApplicationAutoscalingECSServicePolicy	AWS managed	None
○	▸	AWSCodeDeployRoleForECS	AWS managed	None
○	▸	AWSCodeDeployRoleForECSLimited	AWS managed	None
○	▸	ECSWorkshop	Customer managed	Permissions

When you click on JSON, you see a very long list of permissions that are required for using ECS:

ABOUT AUTHOR

Frederique Retsema

Frederique Retsema is active in IT si 1993. Senior Consultant and develop on diverse areas including SQL and Java. She likes work with automation tools like Bamboo, Jenkins, Ansible, Terraform and CloudFormation.

View all posts

FOLLOW US ON LINKEDIN

Following 2,875

POPULAR TAGS

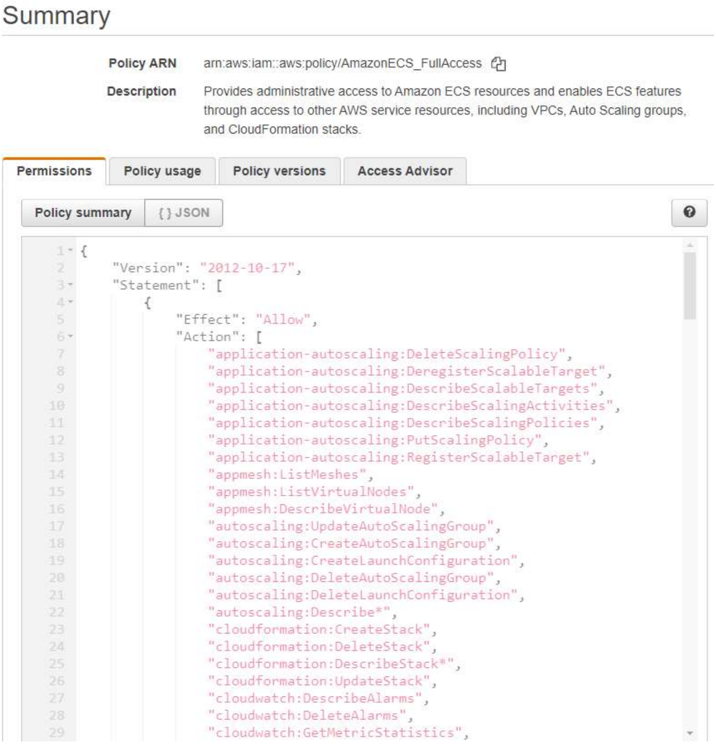
Agile analytical function apex api application container clou service Architecture Azure BPEL bpm cloud container Database docker DVT h integration iot Java javascript jms json kafka kubernetes linux maven monitoring node oci oracle cloud oracle cloud infrastructure oracle database oracle xml db OSB paas performance tuning plsql provisioning puppet push python RES saas Scrum soa sql Vagrant virtual box visualization vm XML

FOLLOW US ON TWITTER

Tweets by @AMISnl

AMIS Conclusion
@AMISnl

New Article : Creating policy's, groups and users in AV ift.tt/3aAFXkn by Frederique Retsema



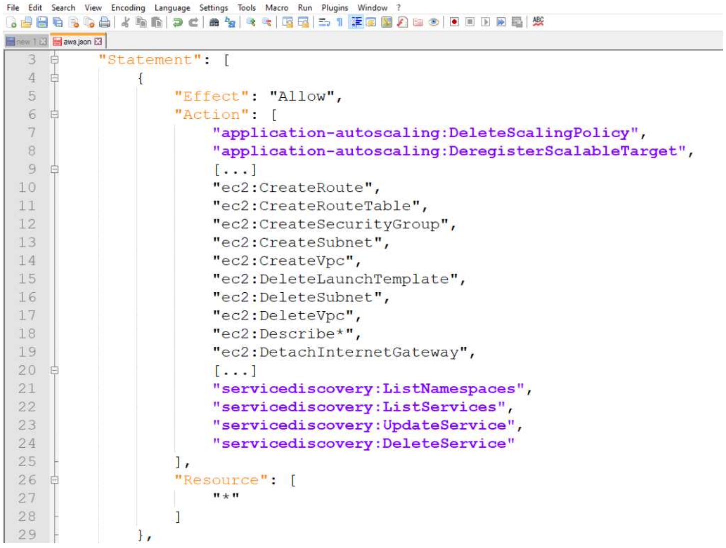
Summary

Policy ARN arn:aws:iam::aws:policy/AmazonECS_FullAccess

Description Provides administrative access to Amazon ECS resources and enables ECS features through access to other AWS service resources, including VPCs, Auto Scaling groups, and CloudFormation stacks.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "appmesh:ListMeshes",
                "appmesh:ListVirtualNodes",
                "appmesh:DescribeVirtualNode",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:Describe*",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:GetMetricStatistics",
```

Now, copy the whole JSON and put it in a text editor. When you look to this JSON, you see that there are no restrictions on region. The policy is divided in several parts: the first one doesn't have any restriction at all about the resource. This means, that you allow people to create security groups, but also delete existing VPC's (virtual networks). For me, this isn't a problem, because I will deny access to regions where I have other VPC's. In fact, in my workshop I will use a region where nothing is defined yet.

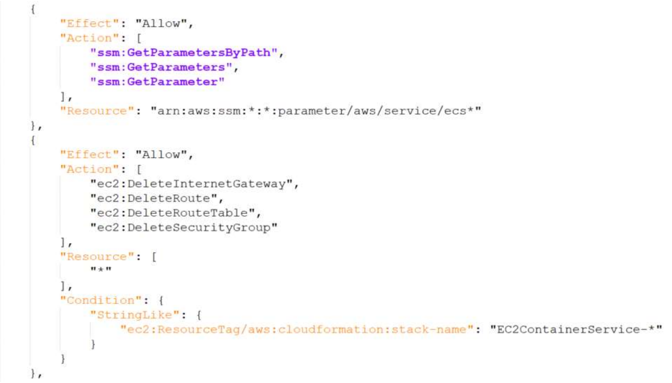


File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1
aws.json

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeregisterScalableTarget",
                [...]
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSubnet",
                "ec2:DeleteVpc",
                "ec2:Describe*",
                "ec2:DetachInternetGateway",
                [...]
                "servicediscovery:ListNamespaces",
                "servicediscovery:ListServices",
                "servicediscovery:UpdateService",
                "servicediscovery:DeleteService"
            ],
            "Resource": [
                "*"
            ]
        },
```

(please mind, that the brackets and the dots […] are put there by me, it's not part of the policy itself)

In other parts of this policies, there are restrictions, f.e. in the names of the resources:



```
{
    "Effect": "Allow",
    "Action": [
        "ssm:GetParametersByPath",
        "ssm:GetParameters",
        "ssm:GetParameter"
    ],
    "Resource": "arn:aws:ssm:*:*:parameter/aws/service/ecs*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DeleteInternetGateway",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DeleteSecurityGroup"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "ec2:ResourceTag/aws:cloudformation:stack-name": "EC2ContainerService-*"
        }
    }
},
```

For our goal, we need to add the region restriction to all these parts. There is one catch, though: some AWS services are global, for example IAM and Route53. You cannot restrict the region here; you will have to split up the first block in services that are global and services that are region dependent. The full policy can be found in the appendix of this blog and in my github repository as well [1].

Creating policy's, groups and users in AWS - …
Today, I'll demonstrate how you can add policy's, groups and users within AWS. In a couple of days,
technology.amis.nl

AMIS Conclusion
@AMISnl

Gecontroleerd en herhaalbaar IT-omgevingen opbouw en bijwerken met de schaalbaarheid en flexibiliteit van gevirtualiseerde omgevingen, zoals Cloud. De voordelen? 👉Lees ze in de #knowwhyweekly van An van Dalen.bit.ly/2xd98vz

Automation van Infrastructure as Code: gecon…
Automation van Infrastructure as Code: gecontroleerd en herhaalbaar IT-omgevingen
amis.nl

Mar 5, 20

AMIS Conclusion
@AMISnl

The latest The AMIS Oracle Technology Weekly! paper.li/AMIS_Services/… #java #word

Getting started with NoSQL Database Service …
technology.amis.nl This morning I discovered a new entry in the menu on my Oracle Cloud Infrastructure
paper.li

Mar 2, 20

AMIS Conclusion
@AMISnl

New Article : Getting started with NoSQL Database Service on Oracle Cloud Infrastructure ift.tt/2I8RACQ b Lucas Jellema

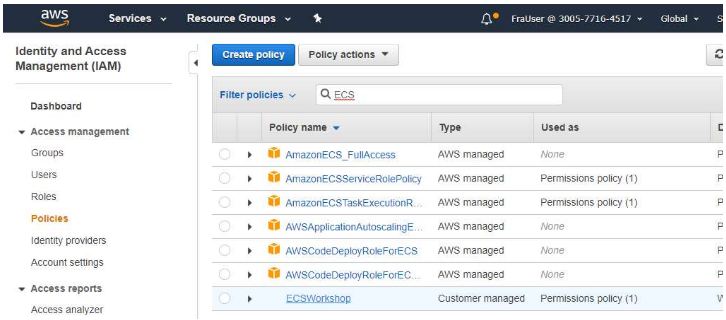
Getting started with NoSQL Database Service …
This morning I discovered a new entry in the menu on my Oracle Cloud Infrastructure Tenancy (on
technology.amis.nl

Feb 29, 20

AMIS Conclusion
@AMISnl

Data is het nieuwe goud. Of het nieuwe Uranium? Van ruwe data naar de beloofde toegevoegde waarde? Va technische know-how naar praktische know-why; lees het in de nieuwe #knowwhyweekly van Andre van Dal 👉 bit.ly/32zjbql#AMISTechnologyBlog #datascience

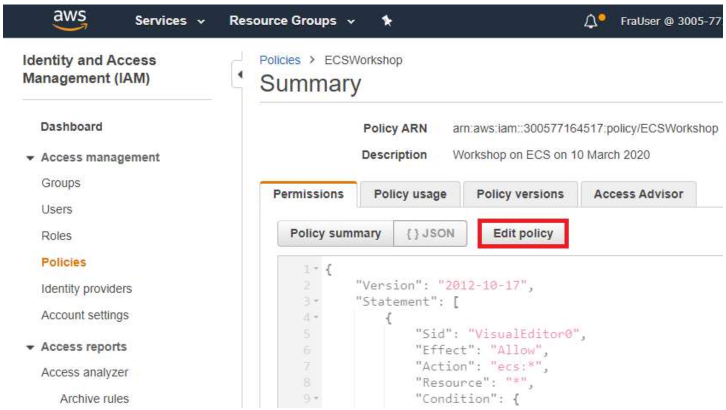
Now we have the changed policy, go to IAM > Policies and search for ECS. Then click on ECSWorkshop:



ECS

Policy name	Type	Used as	D
AmazonECS_FullAccess	AWS managed	None	P
AmazonECSServiceRolePolicy	AWS managed	Permissions policy (1)	P
AmazonECSTaskExecutionR...	AWS managed	Permissions policy (1)	P
AWSApplicationAutoscalingE...	AWS managed	None	P
AWSCodeDeployRoleForECS	AWS managed	None	P
AWSCodeDeployRoleForEC...	AWS managed	None	P
ECSWorkshop	Customer managed	Permissions policy (1)	V

Click on the Edit policy button, and then on JSON:

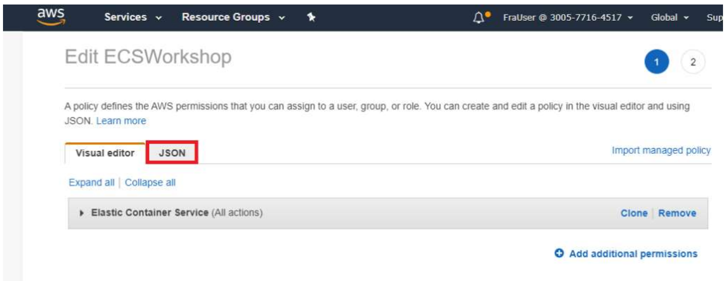


Summary

Policy ARN arn:aws:iam::300577164517:policy/ECSWorkshop

Description Workshop on ECS on 10 March 2020

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ecs:*",
            "Resource": "*",
            "Condition": {
```

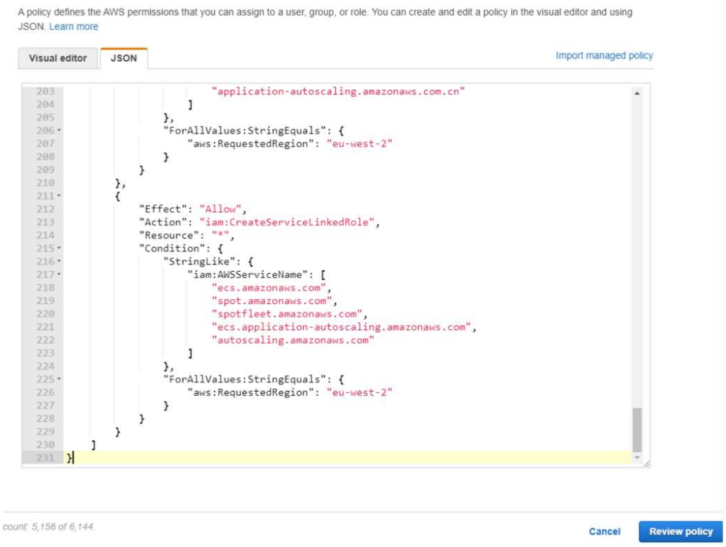


Edit ECSWorkshop

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

Elastic Container Service (All actions)

Now, copy and paste the text that you have in your text editor to this window and click on the Review policy button:



A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

```
                        "application-autoscaling.amazonaws.com.cn"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:RequestedRegion": "eu-west-2"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": [
                        "ecs.amazonaws.com",
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "ecs.application-autoscaling.amazonaws.com",
                        "autoscaling.amazonaws.com"
                    ]
                },
                "ForAllValues:StringEquals": {
                    "aws:RequestedRegion": "eu-west-2"
                }
            }
        }
    ]
}
```

count: 5,156 of 6,144

Click on the Save changes button to save this policy:

Is data het nieuwe uranium?

Is data het nieuwe uranium?

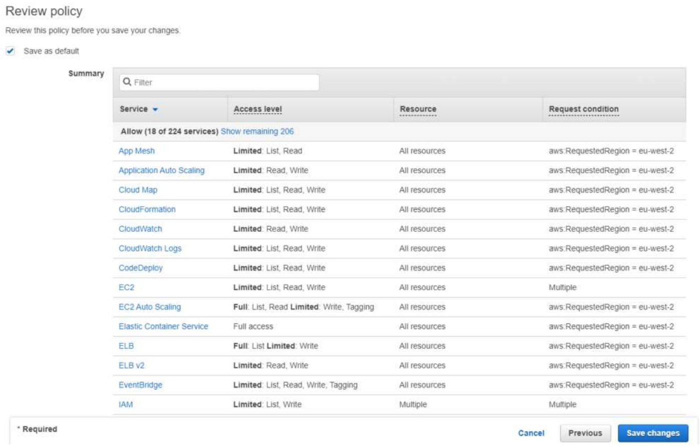
amis.nl

Feb 28, 2(

Embed View on Twit

META

- Site Admin
- Log out
- Entries feed
- Comments feed
- WordPress.org



Review policy

Review this policy before you save your changes.

☑ Save as default

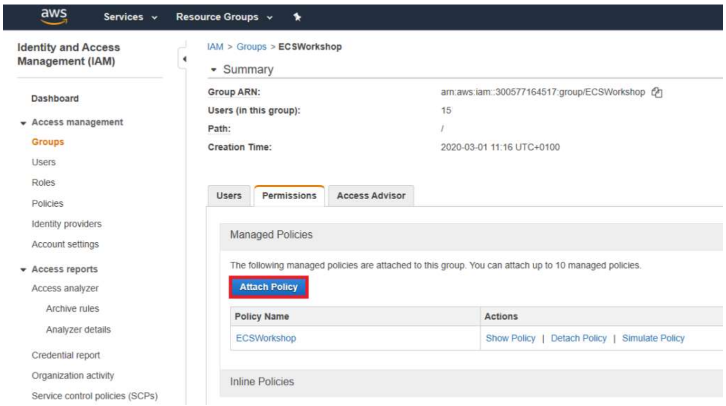
Summary

Service	Access level	Resource	Request condition
Allow (18 of 224 services) Show remaining 206			
App Mesh	**Limited**: List, Read	All resources	aws:RequestedRegion = eu-west-2
Application Auto Scaling	**Limited**: Read, Write	All resources	aws:RequestedRegion = eu-west-2
Cloud Map	**Limited**: List, Read, Write	All resources	aws:RequestedRegion = eu-west-2
CloudFormation	**Limited**: List, Read, Write	All resources	aws:RequestedRegion = eu-west-2
CloudWatch	**Limited**: Read, Write	All resources	aws:RequestedRegion = eu-west-2
CloudWatch Logs	**Limited**: List, Read, Write	All resources	aws:RequestedRegion = eu-west-2
CodeDeploy	**Limited**: List, Read, Write	All resources	aws:RequestedRegion = eu-west-2
EC2	**Limited**: List, Read, Write	All resources	Multiple
EC2 Auto Scaling	**Full**: List, Read **Limited**: Write, Tagging	All resources	aws:RequestedRegion = eu-west-2
Elastic Container Service	Full access	All resources	aws:RequestedRegion = eu-west-2
ELB	**Full**: List **Limited**: Write	All resources	aws:RequestedRegion = eu-west-2
ELB v2	**Limited**: Read, Write	All resources	aws:RequestedRegion = eu-west-2
EventBridge	**Limited**: List, Read, Write, Tagging	All resources	aws:RequestedRegion = eu-west-2
IAM	**Limited**: List, Write	Multiple	Multiple

Other changes

When you have this policy, it's time to test. Go to all the services that are in the list and try to create resources. Is it possible to use resources that are well beyond the scope of the workshop? In my case, I tried to launch an i3en.metal virtual machine using EC2 – which was possible. I think the users of my workshop don't need such an expensive type of virtual machines, I therefore limited the types that can be used (see f.e. this site [2] for inspiration) to only the default that the EC2 wizard will show, m5a.large.

When I tried to add that part to my policy, the policy became too long (> 6144 characters). I therefore had to use two policies: one for the ECSWorkshop (see above), another one called "EC2LimitToM5ALarge": this will limit the creation of EC2 instances and autoscaling instances to m2a.large (which is the smallest type that can be choosed in the ECS wizard). You can connect multiple policies to one group, deny will take precedence above allow. Go to IAM > Groups, edit the ECSWorkshop group, and click on the button "Attach Policy":



IAM > Groups > ECSWorkshop

Summary

Group ARN: arn:aws:iam::300577164517:group/ECSWorkshop

Users (in this group): 15

Path: /

Creation Time: 2020-03-01 11:16 UTC+0100

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Policy Name	Actions
ECSWorkshop	Show Policy | Detach Policy | Simulate Policy

Inline Policies

Both polities are at the end of this blog and in my github repository.

Conclusion

Once you know where to look, creating a new policy, group and new users isn't that hard in AWS. It can be hard to find back your own policy in the list of both AWS policies and your own policies. Try to use a strict naming convention and document the policies you add to your environment.

Curious what I will tell during the workshop? I wrote another AMIS blog about that [3].

Footnotes

[1] https://technology.amis.nl/2020/03/07/creating-policys-groups-and-users-in-aws/

[2] https://github.com/FrederiqueRetsema/AMIS-Blog-AWS

[3] https://blog.vizuri.com/limiting-allowed-aws-instance-type-with-iam-policy

Appendix: ECSWorkshop policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "iam:ListAttachedRolePolicies",
                "iam:ListInstanceProfiles",
                "route53:GetHostedZone",
                "route53:ListHostedZonesByName",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetHealthCheck"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "appmesh:ListMeshes",
                "appmesh:ListVirtualNodes",
                "appmesh:DescribeVirtualNode",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:Describe*",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:PutMetricAlarm",
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:GetApplication",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:ListApplications",
                "codedeploy:ListDeploymentGroups",
                "codedeploy:ListDeployments",
                "codedeploy:StopDeployment",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:ListDeploymentTargets",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetApplicationRevision",
                "codedeploy:RegisterApplicationRevision",
                "codedeploy:BatchGetApplicationRevisions",
                "codedeploy:BatchGetDeploymentGroups",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetApplications",
                "codedeploy:ListApplicationRevisions",
                "codedeploy:ListDeploymentConfigs",
                "codedeploy:ContinueDeployment",
                "sns:ListTopics",
                "lambda:ListFunctions",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotFleetRequests",
                "ec2:CreateInternetGateway",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSubnet",
                "ec2:DeleteVpc",
                "ec2:Describe*",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:RunInstances",
                "ec2:RequestSpotFleet",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateRule",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteListener",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteRule",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeTargetGroups",
                "ecs:*",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:ListRuleNamesByTarget",
                "events:ListTargetsByRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "logs:CreateLogGroup",
                "logs:DescribeLogGroups",
                "logs:FilterLogEvents",
                "servicediscovery:CreatePrivateDnsNamespace",
                "servicediscovery:CreateService",
                "servicediscovery:GetNamespace",
                "servicediscovery:GetOperation",
                "servicediscovery:GetService",
                "servicediscovery:ListNamespaces",
                "servicediscovery:ListServices",
                "servicediscovery:UpdateService",
                "servicediscovery:DeleteService"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/aws/service/ecs*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteInternetGateway",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSecurityGroup"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-name": "EC2ContainerService-*"
                },
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                },
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/ecsInstanceRole*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn"
                    ]
                },
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/ecsAutoscaleRole*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "application-autoscaling.amazonaws.com",
                        "application-autoscaling.amazonaws.com.cn"
                    ]
                },
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": [
                        "ecs.amazonaws.com",
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "ecs.application-autoscaling.amazonaws.com",
                        "autoscaling.amazonaws.com"
                    ]
                },
                "StringEquals": {
                    "aws:RequestedRegion": "eu-central-1"
                }
            }
        }
    ]
}
```

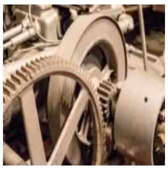
Appendix: EC2LimitToM5ALarge policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "ec2:InstanceType": [
                        "m5a.large"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "autoscaling:InstanceType": [
                        "m5a.large"
                    ]
                }
            }
        }
    ]
}
```

Related Posts:

Creating policy's, groups and users in

Oracle Compute Cloud – Uploading



Differences between CloudFormation,

Application deployment to Oracle

Single-Sign-On to Oracle ERP Cloud

AWS IAM instance type policies region restrictions

ABOUT AUTHOR

FREDERIQUE RETSEMA

Frederique Retsema is active in IT since 1993. Senior Consultant and developer on diverse areas including SQL and Java. She likes to work with automation tools like Bamboo, Jenkins, Ansible, Terraform and CloudFormation.

RELATED POSTS

MARCH 7, 2020 0

Creating policy's, groups and users in AWS

FEBRUARY 26, 2020 0

Setup and use of oVirt on CentOS7

FEBRUARY 23, 2020 0

Secure browsing using a local SOCKS proxy server (on desktop or mobile) and an always free OCI compute instance as SSH server

Leave a Reply

Enter your comment here...