IAM Deep Dive

Custom IAM Policies with Conditions

July 24, 2017

BIO

- Worked in IT for 15 years
- Large scale projects to startups
- Virtualization, Storage, Network Platforms
- Started using Public cloud about 3 year ago
- Currently Work Full Time as a Cloud Architect

LinkedIn
https://www.linkedin.com/in/bryant-poush/

amazon web services | Certified
Solutions Architect - Associate

amazon web services | Certified
SysOps Administrator - Associate

Overview

Review IAM Policy structure

Gain deeper understanding of IAM Policies

Review IAM Conditions

Create custom policies with Conditions

Testing and Debugging IAM Policies

Everyone’s first IAM Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

IAM Policy Structure

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SidIdentifier",
      "Effect": "Allow",
      "Action":   ["ec2:StartInstances"],
      "Resource": "*"
    }
  ]
}
```

Version: Policy Language Version, always use the newest version.

Sid: Optional Identifier

Effect: Allow or Explicit Deny

Action: What you are trying to do.

Resource: What you are trying to your action against.

Multiple Actions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2StartStop",
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

More that one Action.
Combine to reduce policy size!

Multiple Actions with Multiple Statements

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadAllS3Buckets",
      "Effect": "Allow",
      "Action": ["s3:ListBucket", "s3:GetObject"],
      "Resource": "*"
    },
    {
      "Sid": "DenyAuditLogsS3Buckets",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": ["arn:aws:s3:::aws-cloudtrail-logs*", "arn:aws:s3:::aws-config-logs*"]
    }
  ]
}
```

Allow Read of all S3 bucket.

Explicit Deny

Specific Resource

Everyone’s second IAM Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2StopStartRebootTerminateInstances",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:RebootInstances"
      ],
      "Resource": "*"
    },
  ]
}
```

Everyone’s third IAM Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadAllS3Buckets",
      "Effect": "Allow",
      "Action": ["s3:ListBucket", "s3:GetObject"],
      "Resource": "*"
    },
    {
      "Sid": "DenyAuditLogsS3Buckets",
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": ["arn:aws:s3:::aws-cloudtrail-logs*", "arn:aws:s3:::aws-config-logs*"]
    }
  ]
}
```

I love IAM...

Wow...This IAM stuff is easy…

IAM is AWESOME!

I Love IAM

The Challenge….Wait….what!?!

Launch Instance | Connect | Actions

search : Jmeter | Add filter

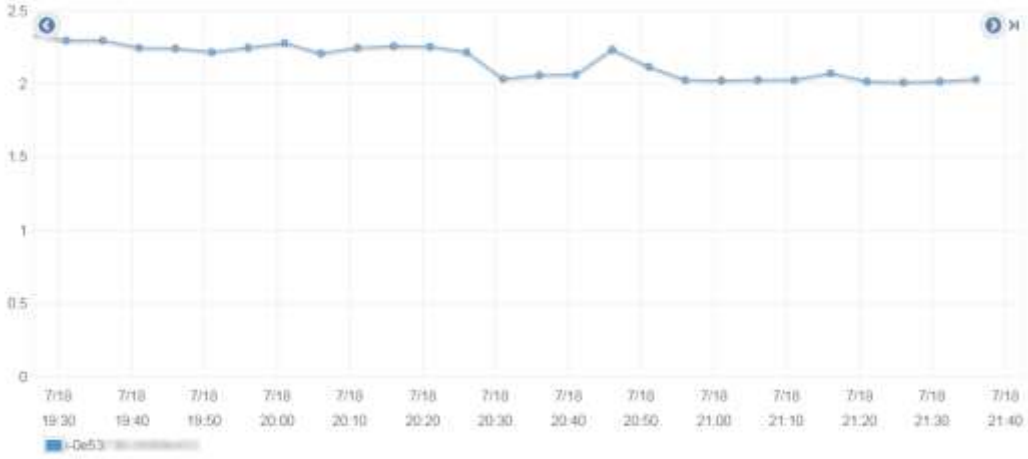
Name	Instance ID	Instance Type
[redacted]	i-0e53[redacted]	m4.10xlarge

Description | Status Checks | Monitoring | Tags

Volume ID	vol-0425[redacted]
Size	500 GiB
Created	[redacted]
State	in-use
Attachment information	i-0e53[redacted] (attached)
Volume type	io1
Product codes	-
IOPS	10000

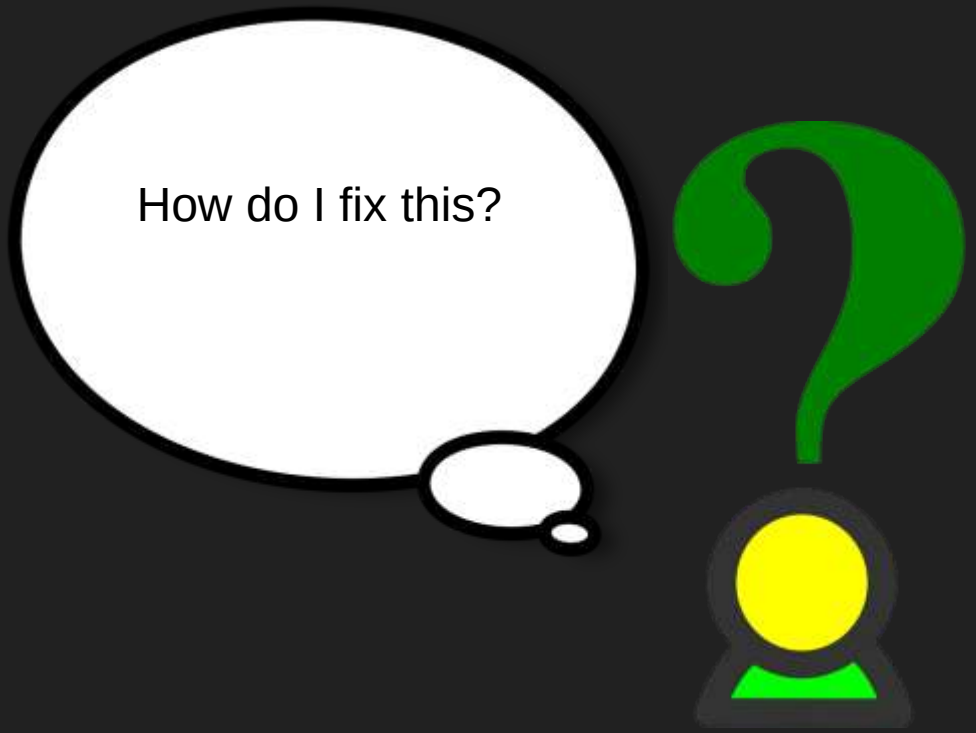
CloudWatch Monitoring Details

CPU Utilization (Percent) | Statistic: Average | Time Range: Last 3 Hours | Period: 5 Minutes



Close

Aftermath


How do I fix this?

Everyone’s second IAM Policy

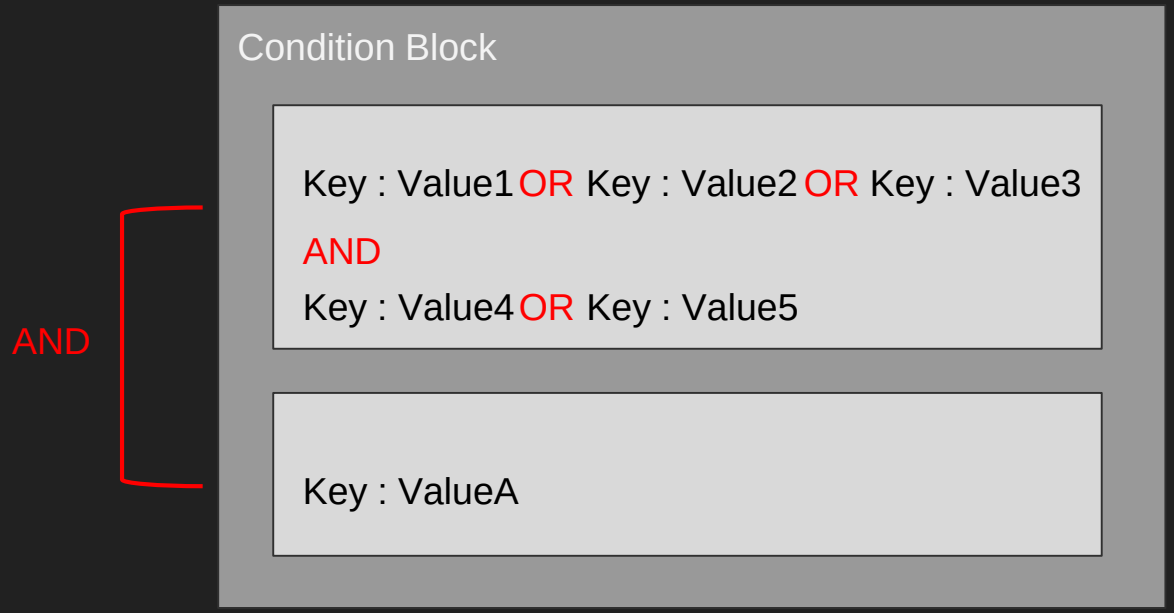
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2StopStartRebootTerminateInstances",
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:RebootInstances"
      ],
      "Resource": "*"
    },
  ]
}
```

ec2:RunInstances = Create and Launch Instance

Enter...IAM Conditions

- Optional block to specify conditions
- All Conditions must evaluate to true for the entire policy to pass as true
- Use standard operators

  - Simple Key : Value Pair
  - Can have Multiple OR
  - Can use AND
  - Multiple Conditions



IAM Condition Block

```
"Condition": {
  "ConditionOperator" : {
    "ContextKey" : "Value"
  }
}
```

Condition Operator: Standard Operators

Value: What do you want evaluate against?

Context Key: AWS Service Action you want to evaluate.

IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SidIdentifier",
      "Effect": "Allow",
      "Action": ["ec2:StartInstances"],
      "Resource": "*",
                "Condition": {
                        "StringEquals": {
                                "ec2:Region":"us-
east-1"
                        }
                }
    }
  ]
}
```

Version: Policy Language Version, always use the newest version.

Sid: Optional Identifier

Effect: Allow or Explicit Deny

Action: What you are trying to do.

Resource: What you are trying to do with.

Condition Block

Condition Block : Operators Examples

String : `StringEquals, StringLike, StringEquals`

Numeric : `NumberiEquals, NumericGrearThan, NumericLessThan`

Data & Time : `DateEquals, DateLessThan, DataGreaterThan`

Boolean : `Boolean Matching`

IP Address : `IpAddress, NotIpAddress`

Amazon Resource Name : `ArnEquals, ArnNotLike, ArnLike`

Condition Block : Context Keys Examples

AWS Identity and Access Management

User Guide

Documentation - This Guide

Search

- What Is IAM?
- Getting Set Up
- Getting Started
- Tutorials
- Best Practices and Use Cases
- IAM Console and Sign-in Page
- Identities
- Access Management
- Logging IAM Events with AWS CloudTrail
- Troubleshooting IAM
- Reference

- `ec2:UnmonitorInstances`

Condition context keys for Amazon EC2

For information about using the following Amazon EC2 conditions in
EC2 in the *Amazon EC2 User Guide for Linux Instances*.

Amazon EC2 has the following service-specific context keys that can
condition context keys that are available to all services, see Available
Reference.

- `ec2:AccepterVpc`
- `ec2:AvailabilityZone`
- `ec2:EbsOptimized`
- `ec2:ImageType`
- `ec2:InstanceProfile`
- `ec2:InstanceType`
- `ec2:Owner`
- `ec2:ParentSnapshot`
- `ec2:PlacementGroup`
- `ec2:PlacementGroupStrategy`
- `ec2:Public`
- `ec2:Region`

Condition Block : Context Keys Examples

Amazon Elastic Compute Cloud

User Guide for Linux Instances

Condition Keys for Amazon EC2

In a policy statement, you can optionally specify conditions that control when it is in effect. Each condition contains one or more key-value pairs. Condition keys are not case sensitive. We've defined AWS-wide condition keys, plus

s in a single condition, we evaluate them using a logical AND

Condition Key	Key-Value Pair	Evaluation Types
`ec2:AccepterVpc`	"ec2:AccepterVpc":"*vpc-arn*" Where *vpc-arn* is the VPC ARN for the accepter VPC in a VPC peering connection	ARN, Null
`ec2:AvailabilityZone`	"ec2:AvailabilityZone":"*az-api-name*" Where *az-api-name* is the name of the Availability example, `us-east-2a`) To list your Availability Zones, use describe-availat	
`ec2:CreateAction`	"ec2:CreateAction":"*api-name*" Where *api-name* is the name of the resource-crea example, `RunInstances`)	
`ec2:EbsOptimized`	"ec2:EbsOptimized":"*optimized-flag*" Where *optimized-flag* is `true` | `false` (for an insta	
`ec2:Encrypted`	"ec2:Encrypted":"*encrypted-flag*" Where *encrypted-flag* is `true` | `false` (for an EBS	
`ec2:InstanceType`	"ec2:InstanceType":"*instance-type-api-name*" Where *instance-type-api-name* is the name of the instance type.	String, Null
`ec2:Owner`	"ec2:Owner":"*account-id*" Where *account-id* is `amazon` | `aws-marketplace` | *aws-account-id*	String, Null
`ec2:ParentSnapshot`	"ec2:ParentSnapshot":"*snapshot-arn*" Where *snapshot-arn* is the snapshot ARN	ARN, Null
`ec2:ParentVolume`	"ec2:ParentVolume":"*volume-arn*" Where *volume-arn* is the volume ARN	ARN, Null
`ec2:PlacementGroup`	"ec2:PlacementGroup":"*placement-group-arn*" Where *placement-group-arn* is the placement group ARN	ARN, Null

Condition Block : Context Keys & Actions

- Not All AWS Service Actions Support Conditions
- Conditions support one or more operators, not all
- Review Resource Level Permissions for specific custom resources actions

AWS Identity and Access Management

User Guide

Documentation - This Guide

Search

AWS Documentation » AWS Identity and Access Management » User Guide » Reference Information for AWS Identity and Access Management » AWS IAM Policy Reference » AWS Service Actions and Condition Context Keys for Use in IAM Policies » Actions and Condition Context Keys for Auto Scaling

Actions and Condition Context Keys for Auto Scaling

Auto Scaling (service prefix: autoscaling) provides the following service-specific actions and condition context keys for use in IAM policies.

Condition context keys for Auto Scaling

For more information about using condition keys in an IAM policy for Auto Scaling, see Auto Scaling Keys in the *Auto Scaling User Guide*.

Auto Scaling has no service-specific context keys that can be used in an IAM policy. For the list of the global condition context keys that are available to all services, see Available Global Condition Keys in the *IAM Policy Elements Reference*.

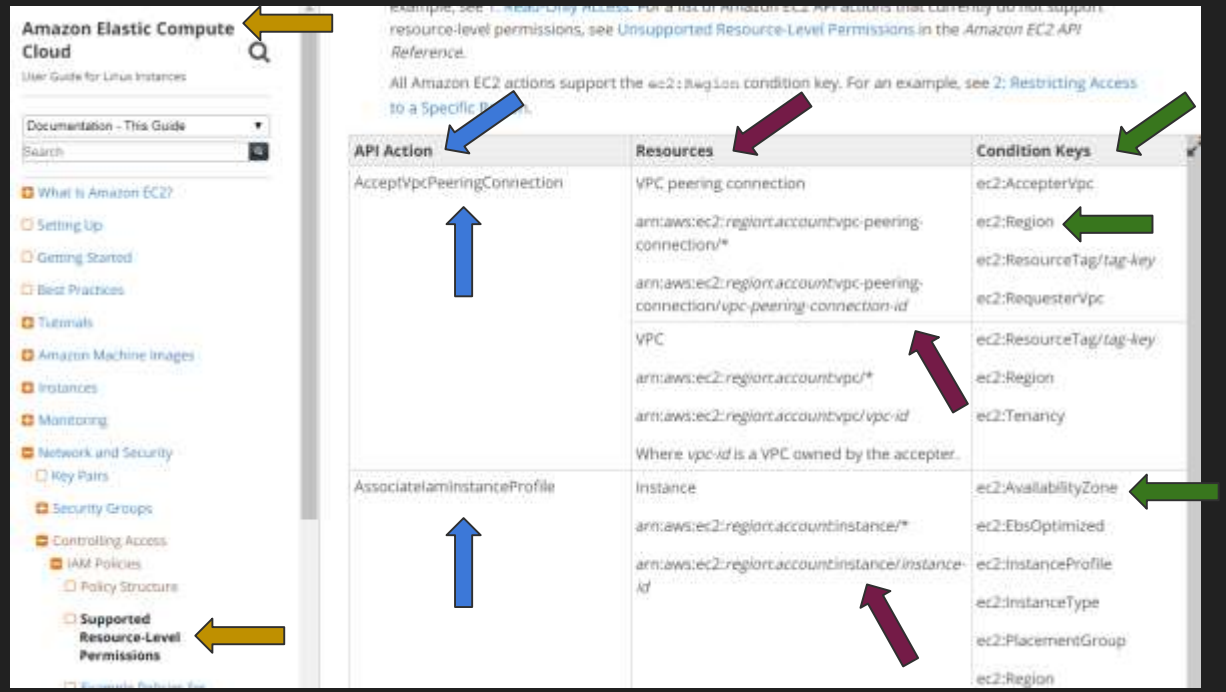
IAM Policy Structure with Resource Level Permissions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SidIdentifier",
      "Effect": "Allow",
      "Action": ["ec2:StartInstances"],
      "Resource": "*",
                "Condition": {
                        "StringEquals": {
                                "ec2:Region":"us-
east-1"
                        }
                }
    }
  ]
}
```

Resource: What you are trying to do with.

Resource-Level Permission

- Specify which resources you can perform actions against
- Some resource level permission support Condition Keys



Creating Custom IAM Policies

Objective 1

Limit EC2 Instance Type

Limit EC2 in specific Region

Allow any AMI

Allow any Network Interface

Allow any Key Pair

Allow any Security Group

Launch in specific subnet, do not allow public subnets

Resource-Level Permission for RunInstances

RunInstances	Image arn:aws:ec2:*region*:image/* arn:aws:ec2:*region*:image/*image-id*	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/*tag-key*
	Instance arn:aws:ec2:*region*:*account*:instance/*	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy

IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
    "Sid": "EC2SpecificInstanceTypes",
    "Effect": "Allow",
    "Action": ["ec2:RunInstances"],
    "Resource": ["*"]
        "Condition": {
                "StringEquals": {
                 "ec2:InstanceType": [
                    "t2.nano",
                    "t2.micro",
                    "t2.small",
                    "t2.medium",
                    "t2.large"
                    ]
                }
            }
      }
  ]
}
```

Condition Block

Condition Operator

Condition Contect Key

Condition Values

arn:aws:ec2:*region*:*account*:instance/*”

Limiting access with Conditions

Objective 1

Limit EC2 Instance Type

Limit EC2 in specific Region

Allow any AMI

Allow any Network Interface

Allow any Key Pair

Allow any Security Group

Launch in specific subnet, do not allow public subnets

Condition Block : Context Keys Examples

RunInstances	Key pair arn:aws:ec2:*region*:*account*:key-pair/* arn:aws:ec2:*region*:*account*:key-pair/*key-pair-name*	ec2:Region
	Network interface arn:aws:ec2:*region*:*account*:network-interface/* arn:aws:ec2:*region*:*account*:network-interface/*eni-id*	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/*tag-key* ec2:Vpc
	Subnet arn:aws:ec2:*region*:*account*:subnet/* arn:aws:ec2:*region*:*account*:subnet/*subnet-id*	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc

IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*",
        "arn:aws:ec2:region:account:subnet/subnet-ec71c7a4",
        "arn:aws:ec2:region:account:subnet/subnet-791fda55",
        "arn:aws:ec2:region:account:subnet/subnet-4e2cd76e"
      ]
    }
  ]
}
```

Resource Level Permissions

Limiting access with Resource Level Permissions & Conditions

Objective 1

Limit EC2 Instance Type

Limit EC2 in specific Region

Allow any AMI

Allow any Network Interface

Allow any Key Pair

Allow any Security Group

Launch in specific subnet, do not allow public subnets

Objective 2

Limit EBS volume type to gp2 only volumes

Limit size of EBS volume to no greater than 50GB

IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
     "Sid": "EC2EBSType",
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
      "Resource": ["arn:aws:ec2:region:account:volume/*"],
      "Condition": {
        "StringEquals": {"ec2:VolumeType": ["gp2"]},
        "NumericLessThanEquals": {"ec2:VolumeSize": ["50"]}
      }
    }
  ]
}
```

Condition AND

Limiting access with Resource Level Permissions & Conditions

Objective 1

Limit EC2 Instance Type

Limit EC2 in specific Region

Allow any AMI

Allow any Network Interface

Allow any Key Pair

Allow any Security Group

Launch in specific subnet, do not allow public subnets

Objective 2

Limit EBS volume type to gp2 only volumes

Limit size of EBS volume to no greater than 50GB

Objective 3

Limit Security Group to Region

Limit Security Group to specific VPC

Limit Security Group Ingress, Egress rules to specific VPC

IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2SecurityGroupsinVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
      ],
      "Resource": ["arn:aws:ec2:region:account:security-group/*"],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": ["arn:aws:ec2:region:account:vpc/vpc-abc1234"]
        }
      }
    }
  ]
}
```

Limiting access with Resource Level Permissions & Conditions

Objective 1

Limit EC2 Instance Type

Limit EC2 in specific Region

Allow any AMI

Allow any Network Interface

Allow any Key Pair

Allow any Security Group

Launch in specific subnet, do not allow public subnets

Objective 2

Limit EBS volume type to gp2 only volumes

Limit size of EBS volume to no greater than 50GB

Objective 3

Limit Security Group to Region

Limit Security Group to specific VPC

Limit Security Group Ingress, Egress rules to specific VPC

Testing Custom Policies with Conditions

- Test Custom Policies in a test account or test User/Group/Roles
- Try to test every action the User/Group/Role may use

Testing IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
     "Sid": "EC2EBSType",
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
      "Resource": ["arn:aws:ec2:region:account:volume/*"],
      "Condition": {
        "StringEquals": {"ec2:VolumeType": ["gp2"]},
        "NumericLessThanEquals": {"ec2:VolumeSize": ["50"]}
      }
    }
  ]
}
```

Action | ec2:RunInstance

EC2 Service Actions for IAM Policies

RunInstances

Launches the specified number of instances using an AMI for which you have permissions.

You can specify a number of options, or leave the default options. The following rules apply:

CreateVolume

Creates an EBS volume that can be attached to an instance in the same Availability Zone. The volume is created in the regional endpoint that you send the HTTP request to. For more information see Regions and Endpoints.

You can create a new empty volume or restore a volume from an EBS snapshot. Any AWS Marketplace product codes from the snapshot are propagated to the volume.

You can create encrypted volumes with the `Encrypted` parameter. Encrypted volumes may only be attached to instances that support Amazon EBS encryption. Volumes that are created from encrypted snapshots are also automatically encrypted. For more information, see Amazon EBS Encryption in the *Amazon Elastic Compute Cloud User Guide*.

You can tag your volumes during creation. For more information, see Tagging Your Amazon EC2 Resources.

For more information, see Creating an Amazon EBS Volume in the *Amazon Elastic Compute Cloud User Guide*.

Testing IAM Policy Structure with Condition Block - Before

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2EBSType",
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
      "Resource": ["arn:aws:ec2:region:account:volume/*"],
      "Condition": {
        "StringEquals": {"ec2:VolumeType": ["gp2"]},
        "NumericLessThanEquals": {"ec2:VolumeSize": ["50"]}
      }
    }
  ]
}
```

Testing IAM Policy Structure with Condition Block - After

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2EBSType",
      "Effect": "Allow",
      "Action": ["ec2:CreateVolume"],
      "Resource": ["arn:aws:ec2:region:account:volume/*"],
      "Condition": {
        "StringEquals": {"ec2:VolumeType": ["gp2"]},
        "NumericLessThanEquals": {"ec2:VolumeSize": ["50"]}
      }
    }
  ]
}
```

Testing IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EBSVolumeType",
      "Effect": "Allow",
      "Action": ["ec2:CreateVolume"],
      "Resource": ["arn:aws:ec2:region:account:volume/*"],
      "Condition": {
        "StringEquals": {"ec2:VolumeType": "gp2"},
        "NumericLessThanEquals": {"ec2:VolumeSize": ["50"]}
      }
    },
    {
      "Sid": "EC2EBSType",
      "Effect": "Allow",
      "Action": ["ec2:RunInstances"],
      "Resource": ["arn:aws:ec2:region:account:volume/*"],
      "Condition": {
        "StringEquals": {"ec2:VolumeType": ["gp2"]},
        "NumericLessThanEquals": {"ec2:VolumeSize": ["50"]}
      }
    }
  ]
}
```

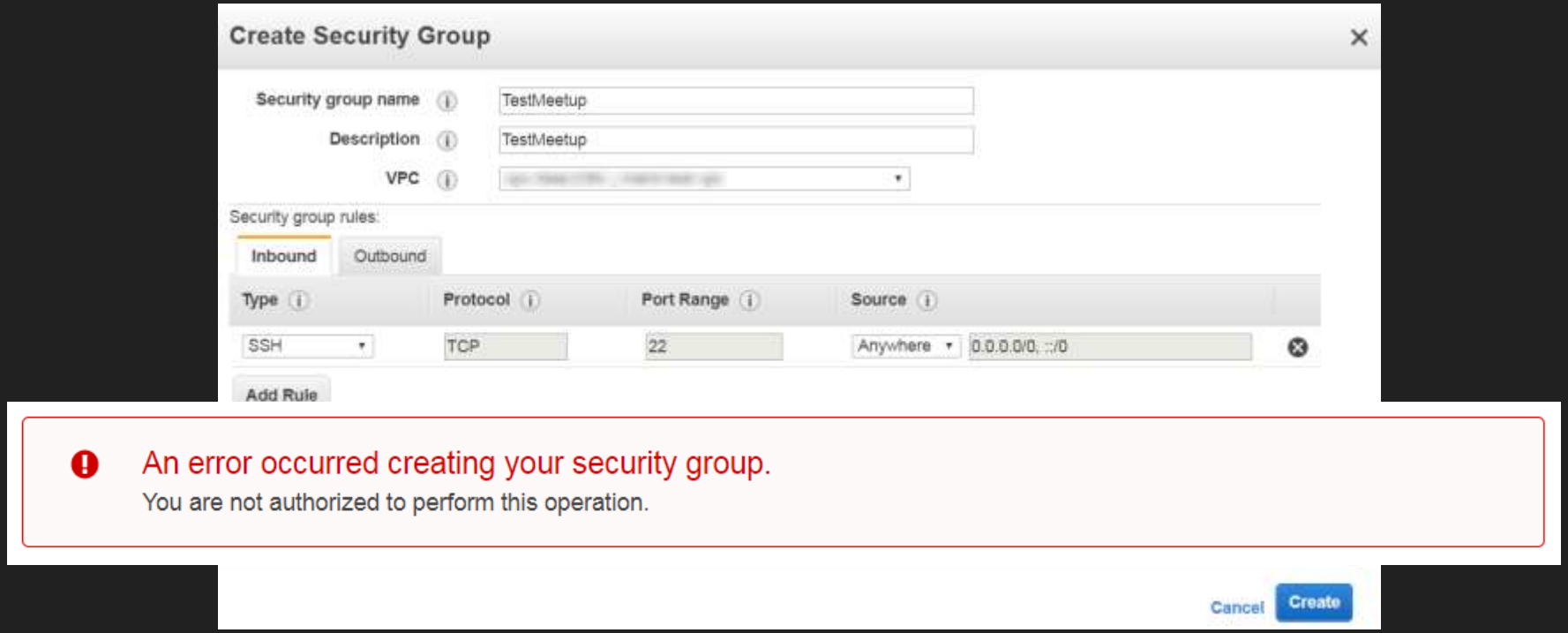
Debugging IAM Policy Structure with Condition Block - Correct

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2SecurityGroupsinVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
      ],
      "Resource": ["arn:aws:ec2:region:account:security-group/*"],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": ["arn:aws:ec2:region:account:vpc/vpc-abc1234"]
        }
      }
    }
  ]
}
```

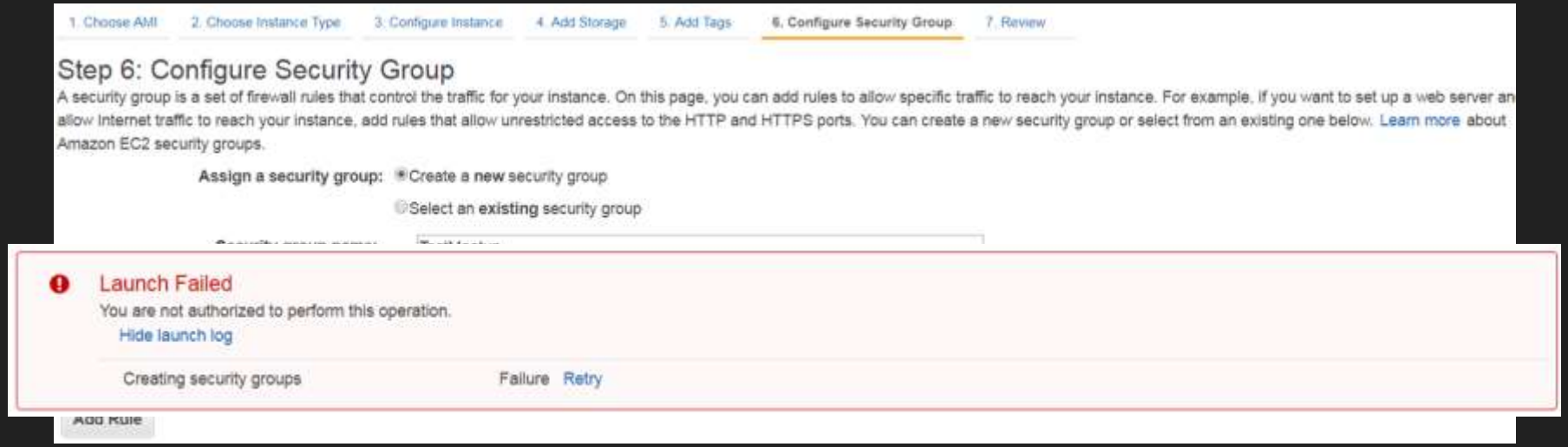
Debugging IAM Policy Structure with Condition Block - Incorrect

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2SecurityGroupsinVPC",
      "Effect": "Allow",
      "Action": [
        “ec2:CreateSecurityGroup”,
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
      ],
      "Resource": ["arn:aws:ec2:region:account:security-group/*"],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": ["arn:aws:ec2:region:account:vpc/vpc-abc1234"]
        }
      }
    }
  ]
}
```

Debugging IAM Policy Structure with Condition Block



Debugging IAM Policy Structure with Condition Block



Debugging IAM Policy Structure with Condition Block

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
     "Sid": "EC2SecurityGroupsinVPC",
      "Effect": "Allow",
      "Action": [
        “ec2:CreateSecurityGroup”,
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
      ],
      "Resource": ["arn:aws:ec2:region:account:security-group/*"],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": ["arn:aws:ec2:region:account:vpc/vpc-abc1234"]
        }
      }
    }
  ]
}
```

Action | ec2:CreateSecurityGroup

Debugging IAM Policy Structure with Condition Block

AuthorizeSecurityGroupEgress	Security group arn:aws:ec2:*region*:*account*:security-group/* arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
AuthorizeSecurityGroupIngress	Security group arn:aws:ec2:*region*:*account*:security-group/* arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
DeleteSecurityGroup	Security group	ec2:Region
RevokeSecurityGroupEgress	Security group arn:aws:ec2:*region*:*account*:security-group/* arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc
RevokeSecurityGroupIngress	Security group arn:aws:ec2:*region*:*account*:security-group/* arn:aws:ec2:*region*:*account*:security-group/*security-group-id*	ec2:Region ec2:ResourceTag/*tag-key* ec2:Vpc

Debugging IAM Policy Structure with Condition Block – Incorrect Resource

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2SecurityGroupsinVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
      ],
      "Resource": ["arn:aws:ec2:region:account:security-group/*"],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": ["arn:aws:ec2:region:account:vpc/vpc-abc1234"]
        }
      }
    }
  ]
}
```

Debugging IAM Policy Structure with Condition Block – Resource *

```
{
    "Sid": "EC2CreateSecurityGroups",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateSecurityGroup"
    ],
    "Resource": "*"
  },
```

Debugging IAM Policy Structure with Condition Block - Final

```
{
  "Version": "2012-10-17",
  "Statement": [
                                {
      "Sid": "EC2CreateSecurityGroups",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2SecurityGroupsinVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
      ],
      "Resource": [
        "arn:aws:ec2:region:account:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:Vpc": [
            "arn:aws:ec2:region:account:vpc/vpc-abc1234"
          ]
        }
      }
    }
  ]
}
```

THANK YOU!

- Slide Deck will be posted on SlideShare
- Examples IAM Policies @ https://github.com/bryantpoush/meetups/tree/master/2017/July
- Feel free to reach out on Meetup