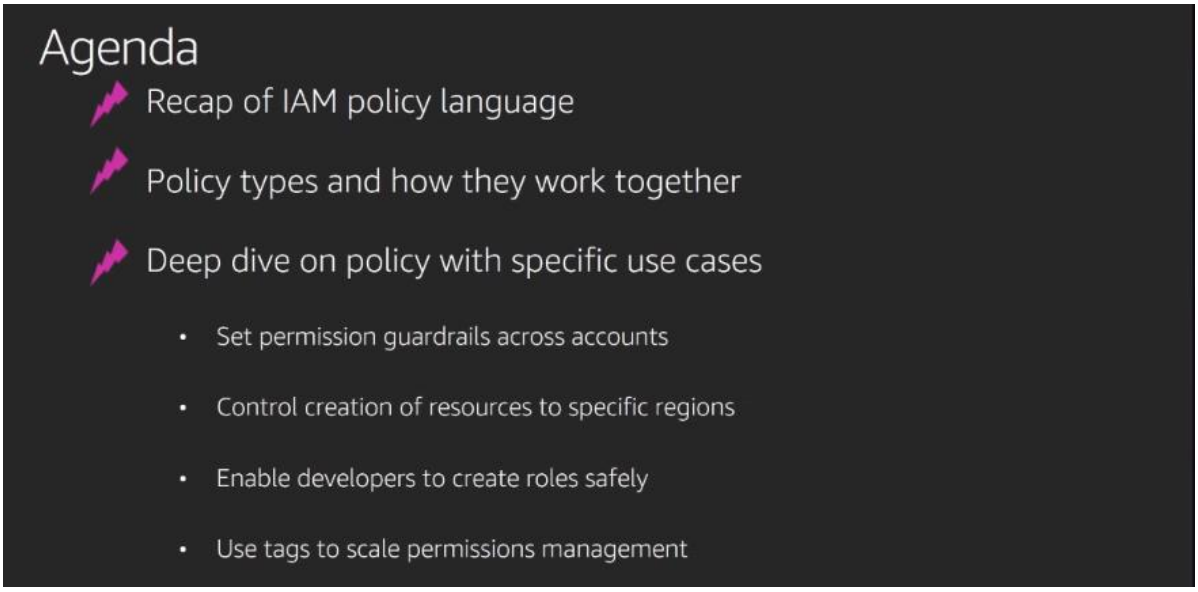
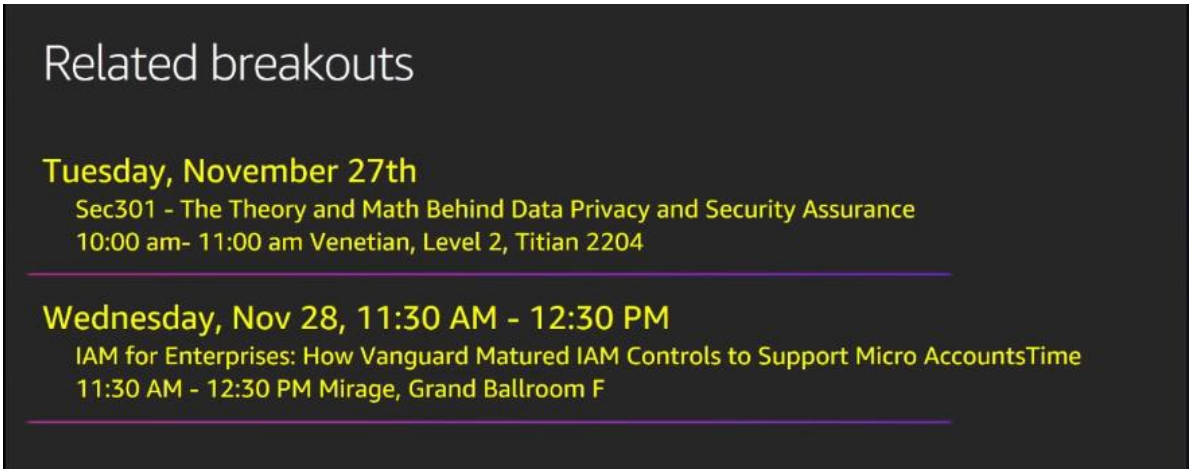
Are you interested in becoming an IAM policy master and learning about powerful techniques for controlling access to AWS resources? If your answer is “yes,” this session is for you. Join us as we cover the different types of policies and describe how they work together to control access to resources in your account and across your AWS organization. We walk through use cases that help you delegate permission management to developers by demonstrating IAM permission boundaries. We take an in-depth look at controlling access to specific AWS regions using condition keys. Finally, we explain how to use tags to scale permissions management in your account. This session requires you to know the basics of IAM policies.





Recap of IAM policy language

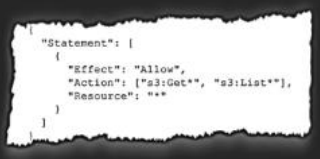
Quick recap of IAM policies

- What are IAM policies?
- IAM policy structure
- IAM policy evaluation rules
- How to think about IAM policy evaluation (new!)

What are IAM policies?

Policies provide authorization to AWS services and resources

Two parts:
- Specification: *Defining* access policies
- Enforcement: *Evaluating* policies



When you *define* access policies. You specify which IAM principals are allowed to perform which actions on specific AWS resources and under which conditions.

IAM enforces this access by *evaluating* the AWS request and the policies you defined and returns either yes or no answer.

IAM policy structure

```
{
 "Statement":[{
  "Effect":"effect",
  "Principal":"principal",
  "Action":"action",
  "Resource":"arn",
  "Condition":{
   "condition":{
    "key":"value" }
   }
  }
 ]
}
```

Principal – The entity that is allowed or denied access

"Principal":"AWS":"arn:aws:iam::123456789012:user/username"

Action - Type of access that is allowed or denied access

"Action":"s3:GetObject"

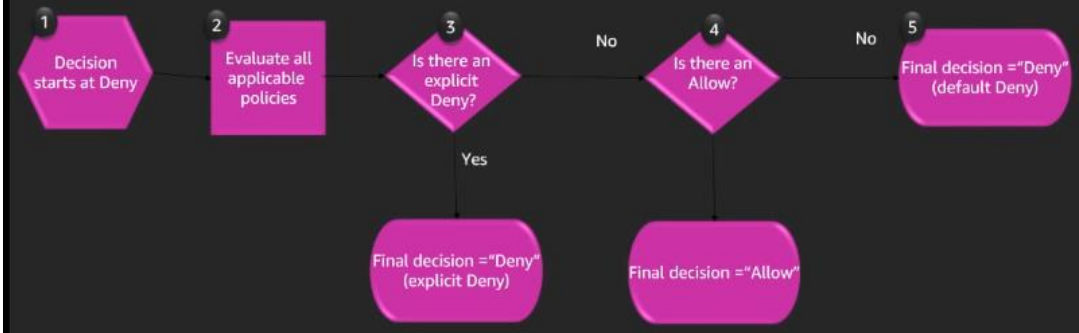
Resource – The Amazon resource(s) the action will act on

"Resource":"arn:aws:sqs:us-west-2:123456789012:queue1"

Condition – The conditions under the access defined is valid

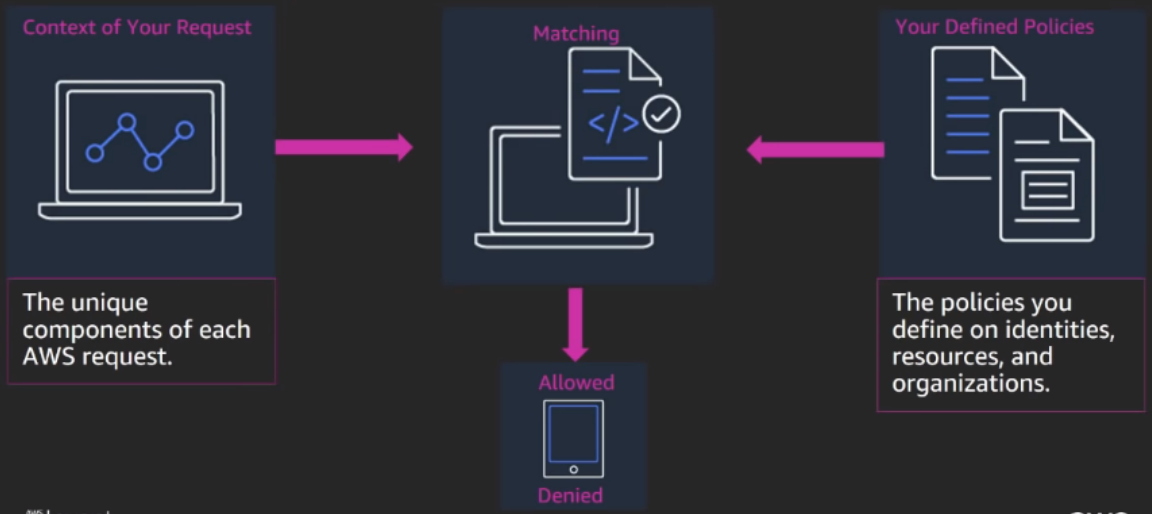
"StringEqualsIfExists": {"aws:RequestTag/project": ["Pickles"]}

IAM policy evaluation



Policy enforcement

Context and policies – *a new way to think about evaluation*



Policy types and how they work together

Policy types and core use cases

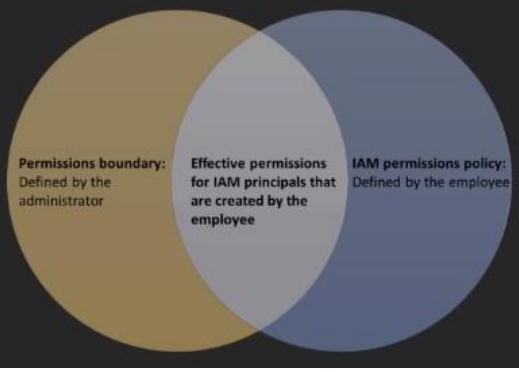
AWS Organizations Service control policies (SCPs)	*Guardrails to disable service access on the principals in the account*
AWS Identity and Access Management (IAM) As Permission Policies and Permission Boundaries	*Grant granular permissions on IAM principals (users and roles) and control the maximum permissions they can set*
AWS Security Token Service (AWS STS) Scoped-down policies	*Reduce general shared permissions further*
Specific AWS services Resource-based policies	*Cross-account access and to control access from the resource*
VPC Endpoints Endpoint Policies	*Controls access to the service with a VPC endpoint*

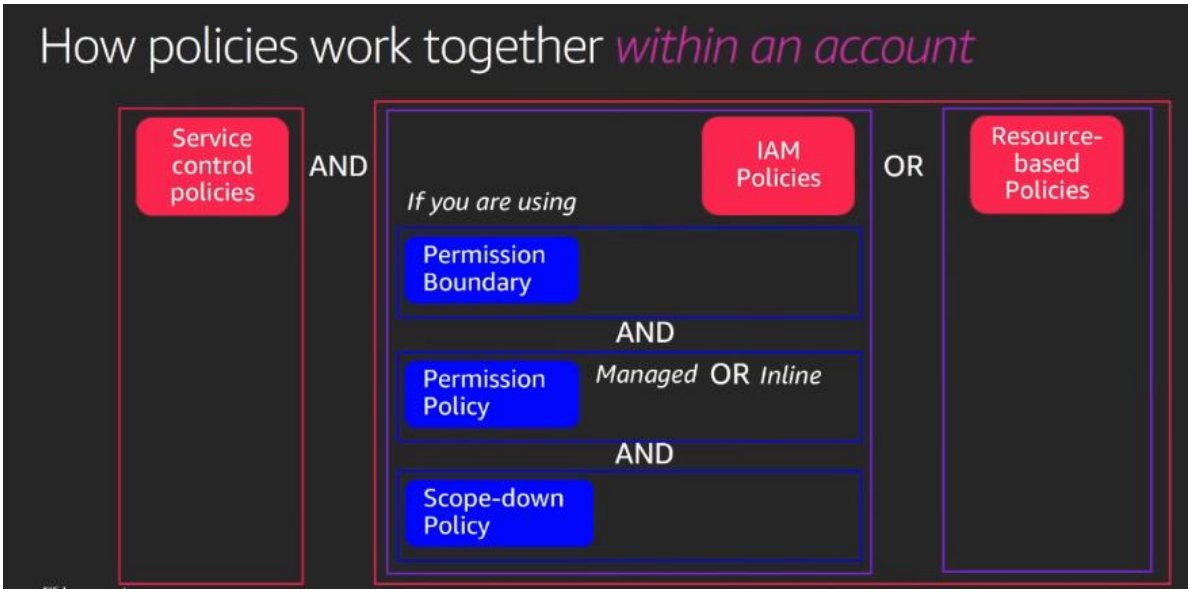
All use the same policy language

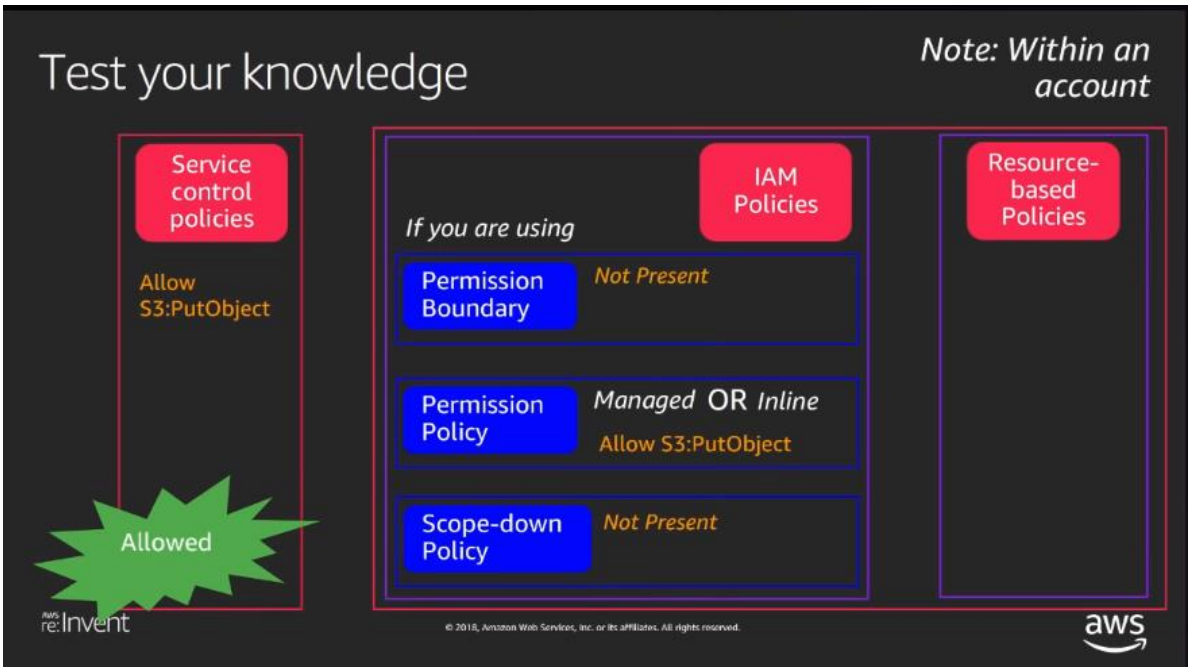
Permission boundaries – new this year!

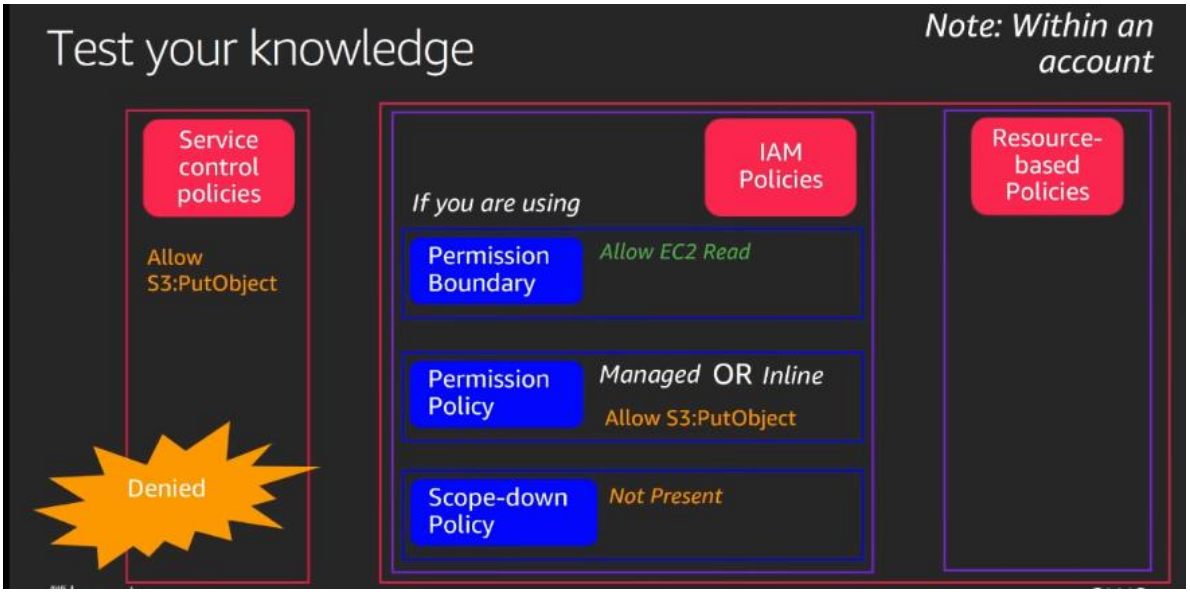
Scale and delegate permission management to developers safely

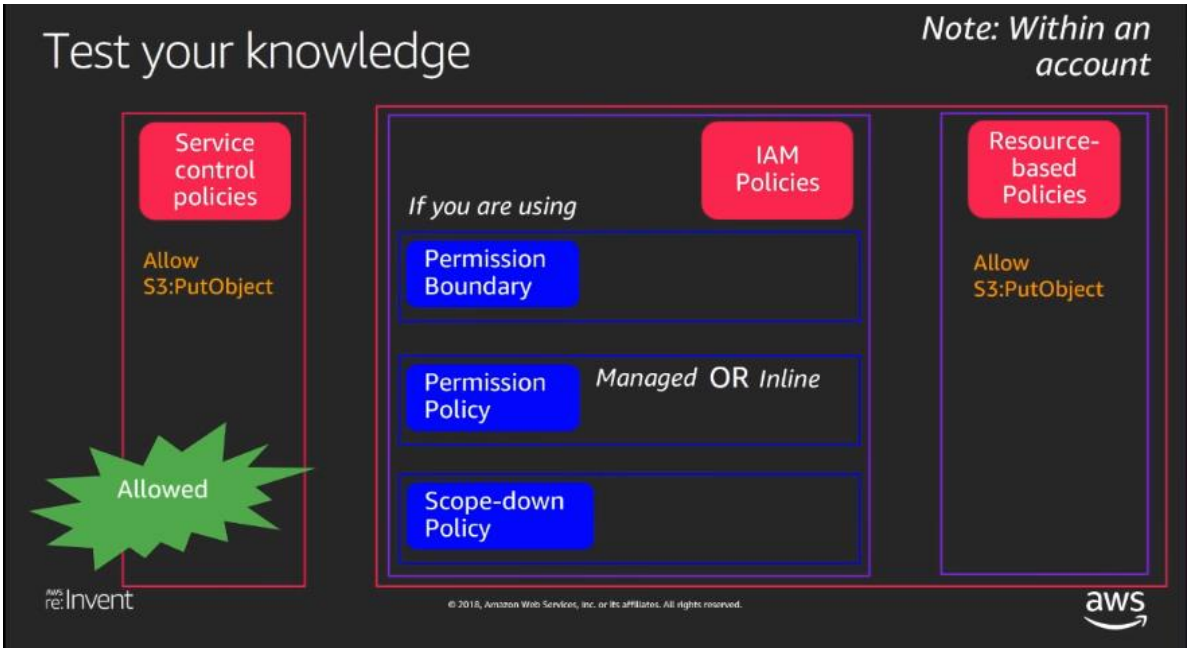
Control the maximum permissions employees can grant

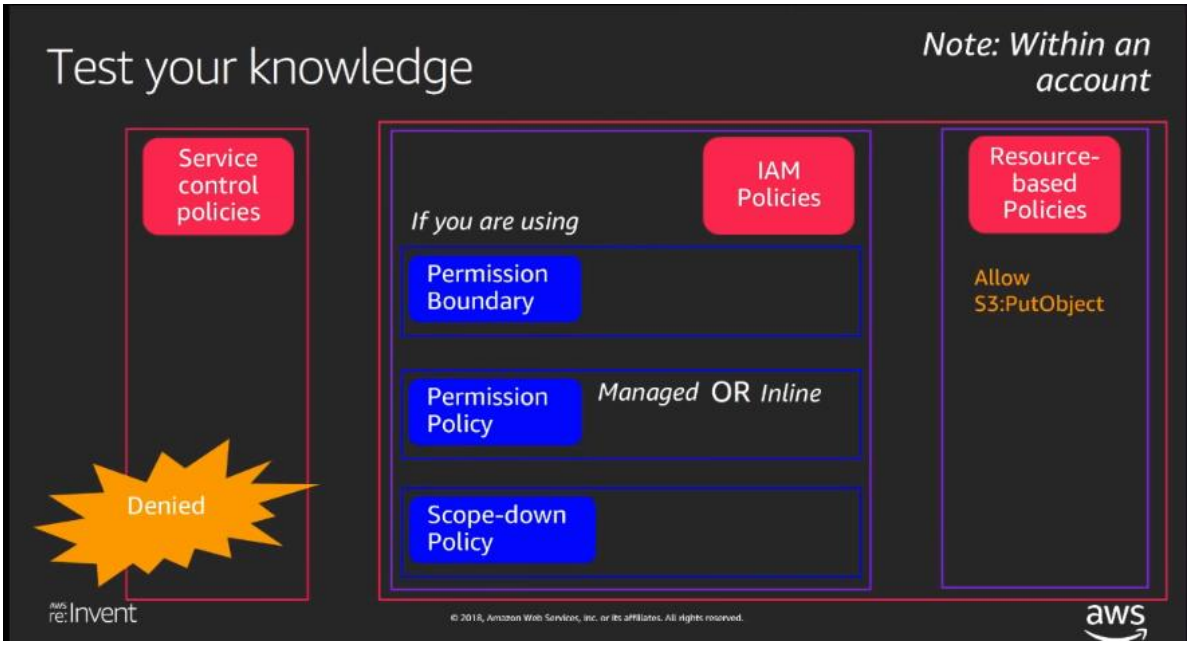


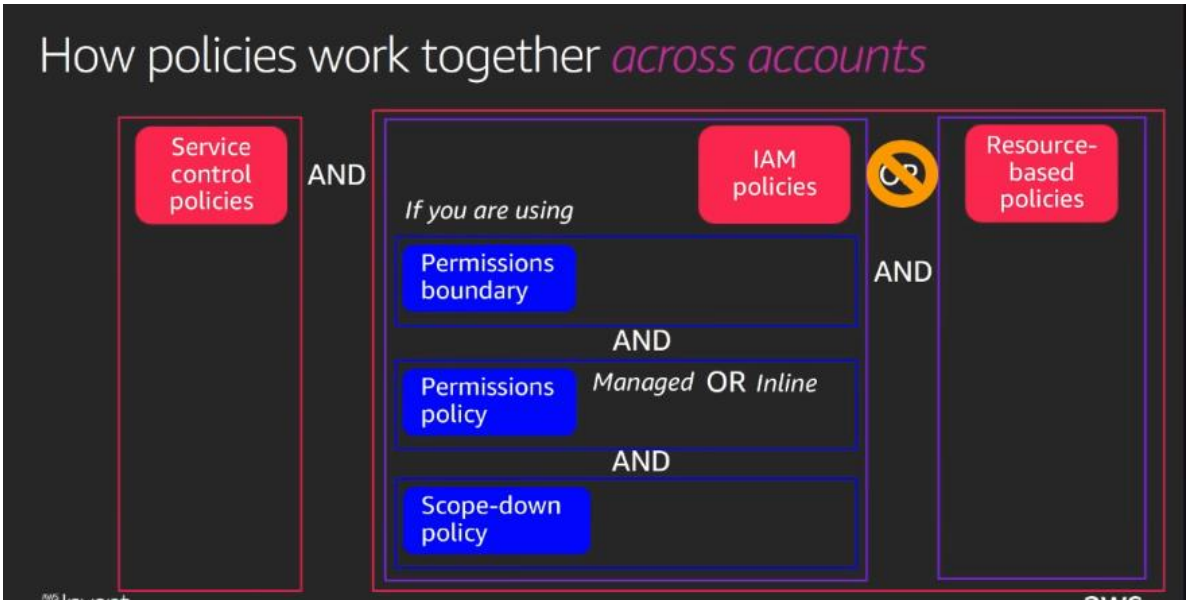

How policies work together within an account
Service control policies
AND
IAM Policies
If you are using
Permission Boundary
AND
Permission Policy
Managed OR Inline
AND
Scope-down Policy
OR
Resource-based Policies


Test your knowledge
Note: Within an account
Service control policies
Allow S3:PutObject
Allowed
IAM Policies
If you are using
Permission Boundary
Not Present
Permission Policy
Managed OR Inline
Allow S3:PutObject
Scope-down Policy
Not Present
Resource-based Policies
aws re:Invent
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
aws

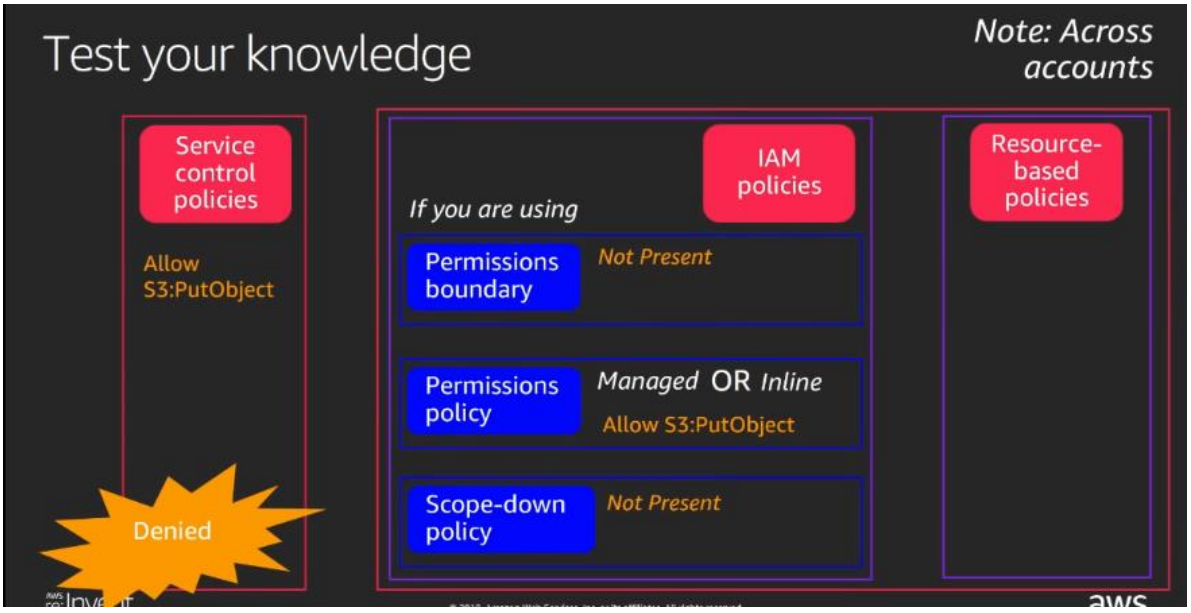

Test your knowledge
Note: Within an account
Service control policies
Allow S3:PutObject
Denied
IAM Policies
If you are using
Permission Boundary
Allow EC2 Read
Permission Policy
Managed OR Inline
Allow S3:PutObject
Scope-down Policy
Not Present
Resource-based Policies

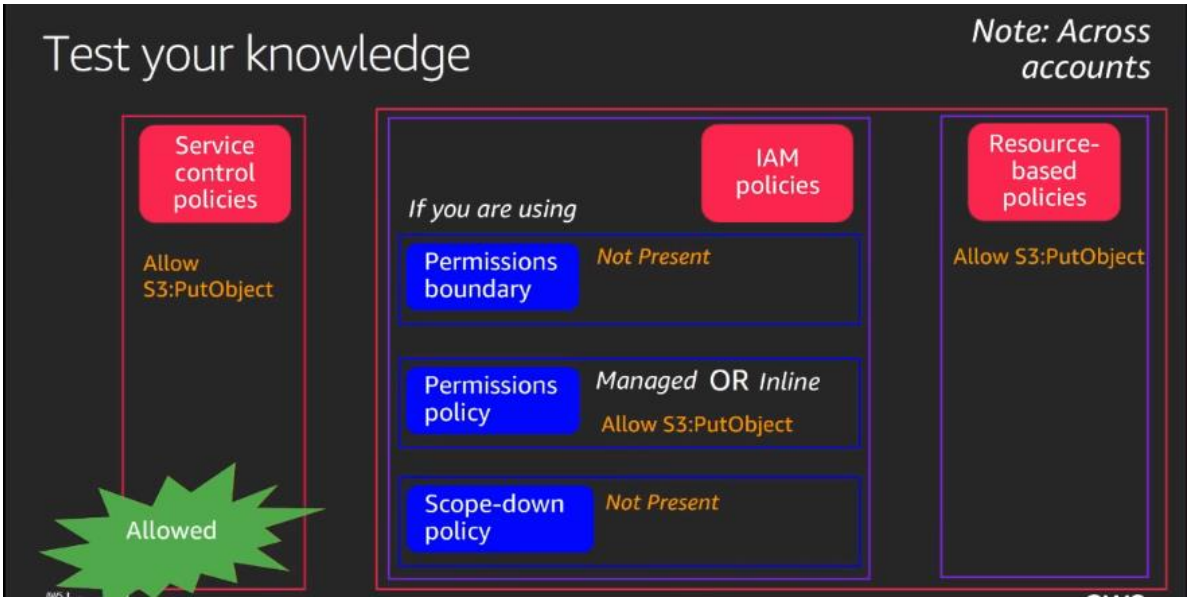

Test your knowledge
Note: Within an account
Service control policies
Allow S3:PutObject
Allowed
IAM Policies
If you are using
Permission Boundary
Permission Policy
Managed OR Inline
Scope-down Policy
Resource-based Policies
Allow S3:PutObject
aws re:Invent
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
aws


Test your knowledge
Note: Within an account
Service control policies
Denied
IAM Policies
If you are using
Permission Boundary
Permission Policy
Managed OR Inline
Scope-down Policy
Resource-based Policies
Allow S3:PutObject
aws re:Invent
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
aws



Both sides now have to allow for it to hold





Deep dive on policy with specific use cases

Congratulations – you just got a new job!

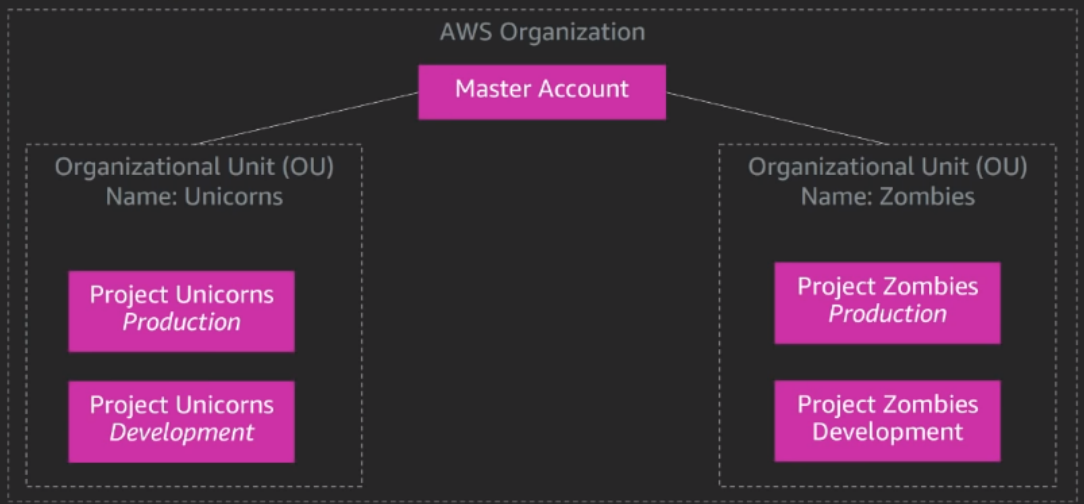
Your new position

Let's imagine you are now the lead of a central security team

Your first mission

Prevent developers from reverting setting in your AWS accounts and onboard a new two new teams!

AWS Organization and account structure



Services your organization uses

- Amazon EC2 to run workloads
- AWS Lambda for serverless applications
- AWS Secrets Manager to store secrets for database access and third party API keys
- Amazon S3 to store content objects

#1 - Set permission guardrails across accounts

#1 - Situation

Your team has gone through and set up AWS CloudTrail in all accounts. Your company also requires users to authenticate with their existing identity provider.

#1 - Challenge

Ensure developers cannot turn off CloudTrail, create IAM users, or set up AWS Directory Service.

#2 - Control creation of resources to specific regions

#2 - Situation

You've learned that you can trust your development team to create resources in AWS, however your leadership is concerned about creating resources in unapproved regions.

#2 - Challenge

Ensure your developers can create resources, but only in approved regions.

#3 - Enable developers to create roles safely

#3 - Situation

Your developers know their stuff! They mentioned they can build on AWS more quickly if they can create their own roles without going through your central security team.

#3 - Challenge

Enable your developers to create IAM roles to pass to Amazon EC2 and AWS Lambda, but ensure they cannot exceed their own permissions.

#4 - Use tags to scale permissions management

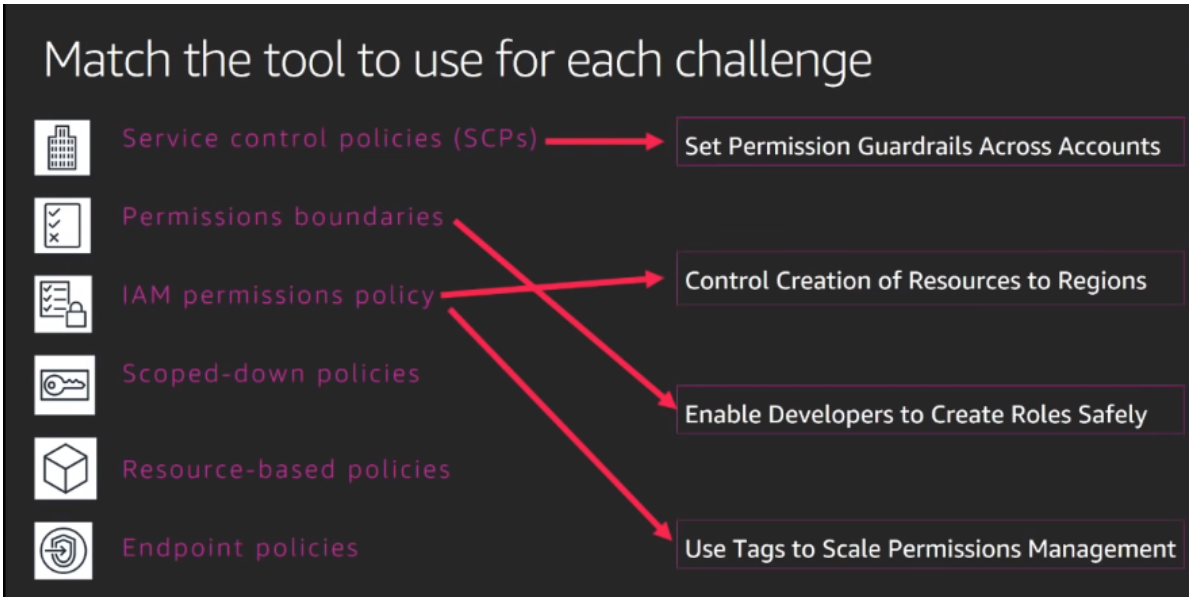
#4 - Situation

The Unicorns project has been split into two projects. Dorky Unicorns and Sneaky Unicorns. They still share an account and keep stepping on each other toes.

#4 - Challenge

Update permissions to enable developers working on Dorky Unicorns and Sneaky Unicorns to manage their own resources without managing the other project's.

Match the tool to use for each challenge



Challenge #1

Ensure developers cannot turn off CloudTrail, create IAM users,

or set up AWS Directory Service.

Pro Tip: Rely on deny statements when restricting access to accounts to reduce blast radius.

We will use a service control policy SCP.

SCP for challenge #1

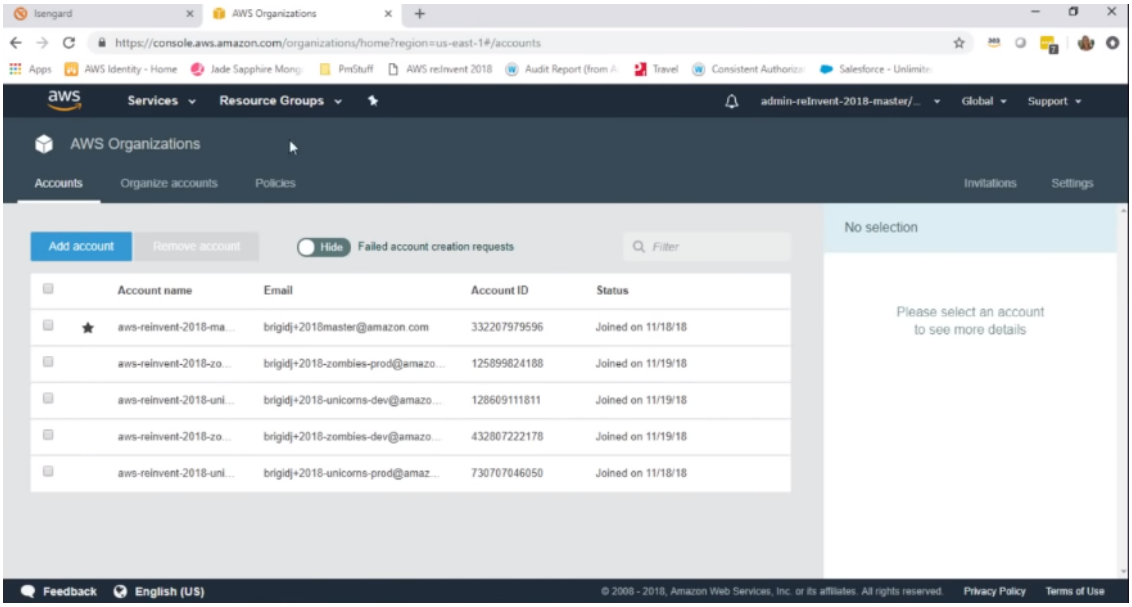
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyUnapprovedAction",
            "Effect": "Deny",
            "Action": [
                "ds:*",
                "iam:CreateUser",
                "cloudtrail:StopLogging"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Don't Forget!

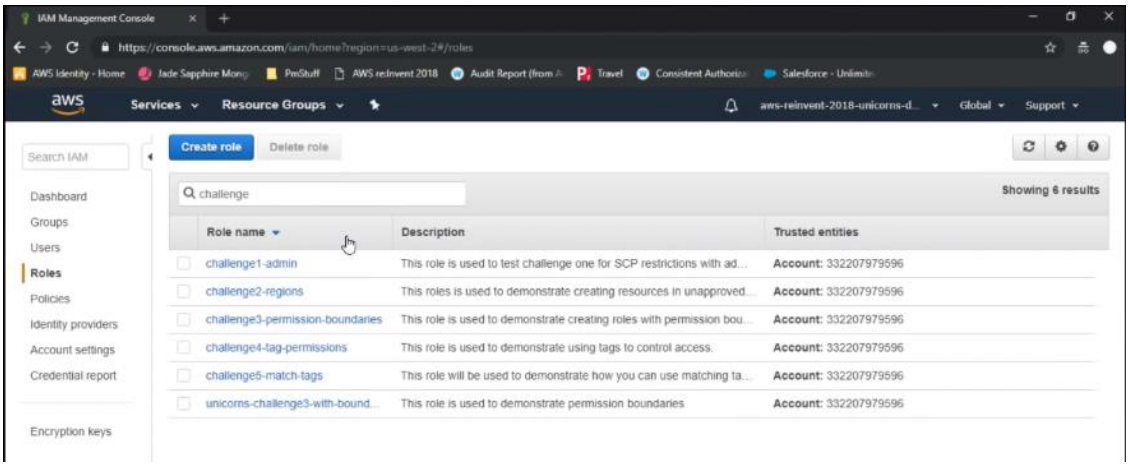
*We also have an Allow *.* policy attached to this OU*

Let's see SCPs in action

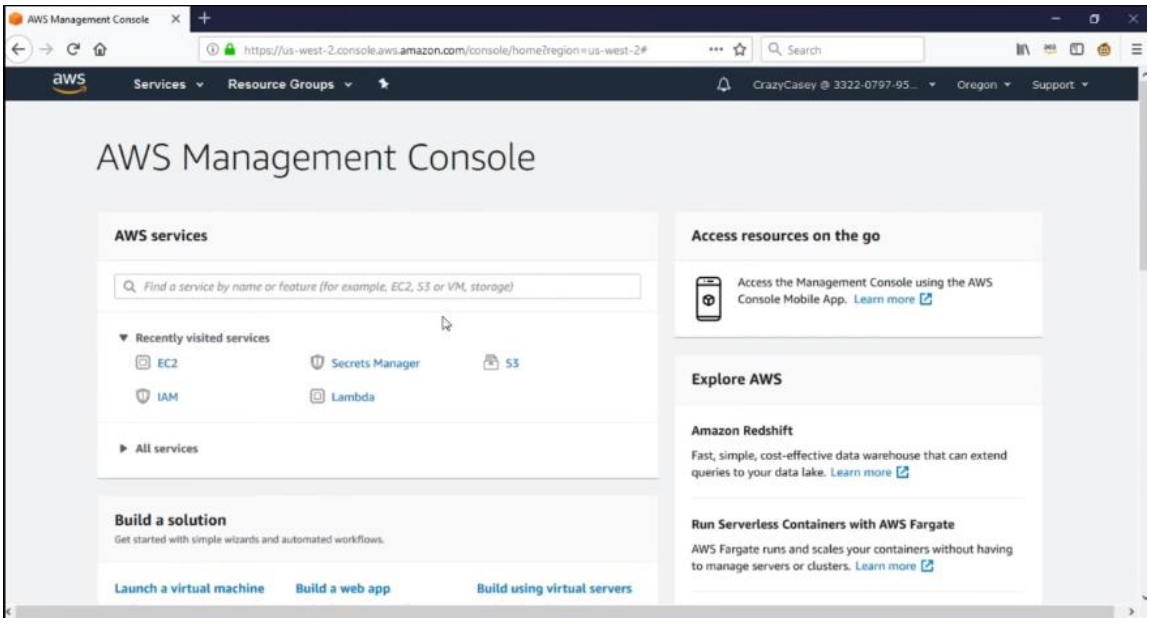
- Show SCP to deny access to modify AWS CloudTrail, Create IAM users, and AWS Directory Services
- Use the CLI as an administrator in the Unicorn-dev account and try to:
  - Create an IAM user
  - List roles
  - Stop logging in AWS CloudTrail



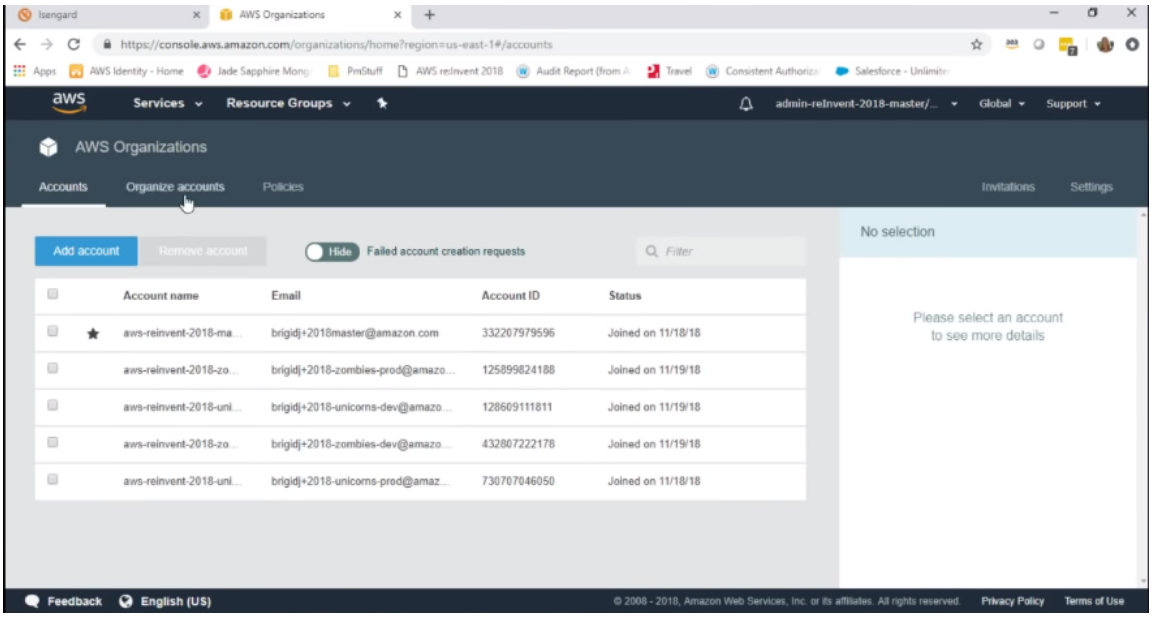
This dark skin mode is our master account

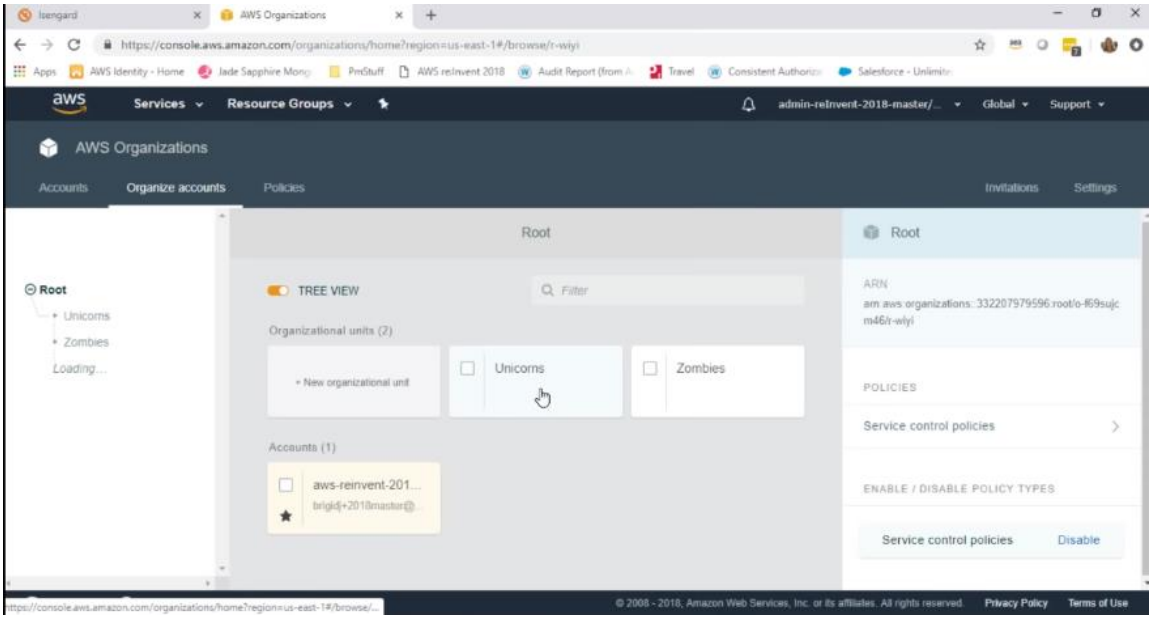


This light skin mode is our unicorn's dev account where we will see a bunch of the activities

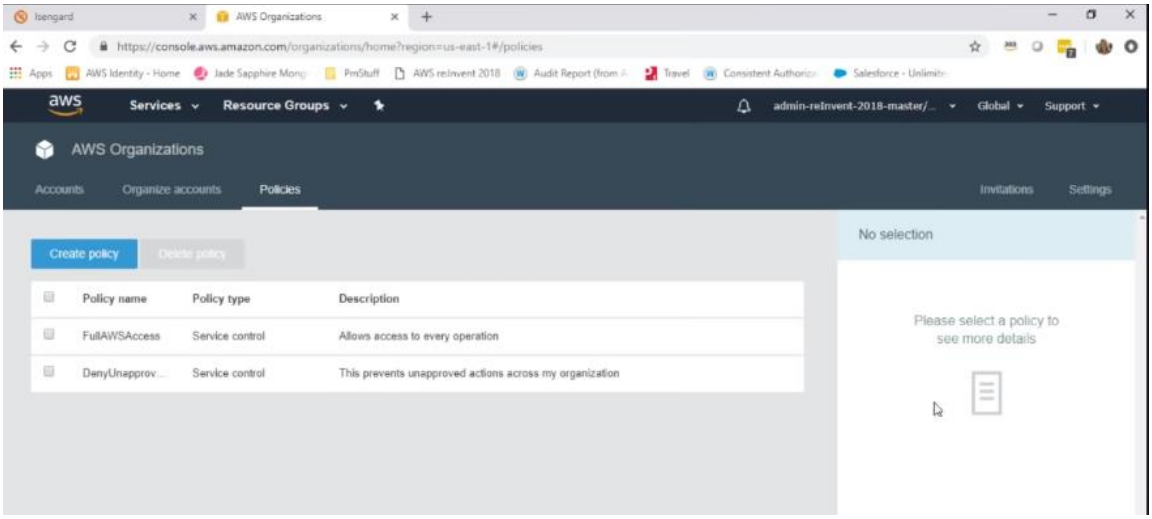


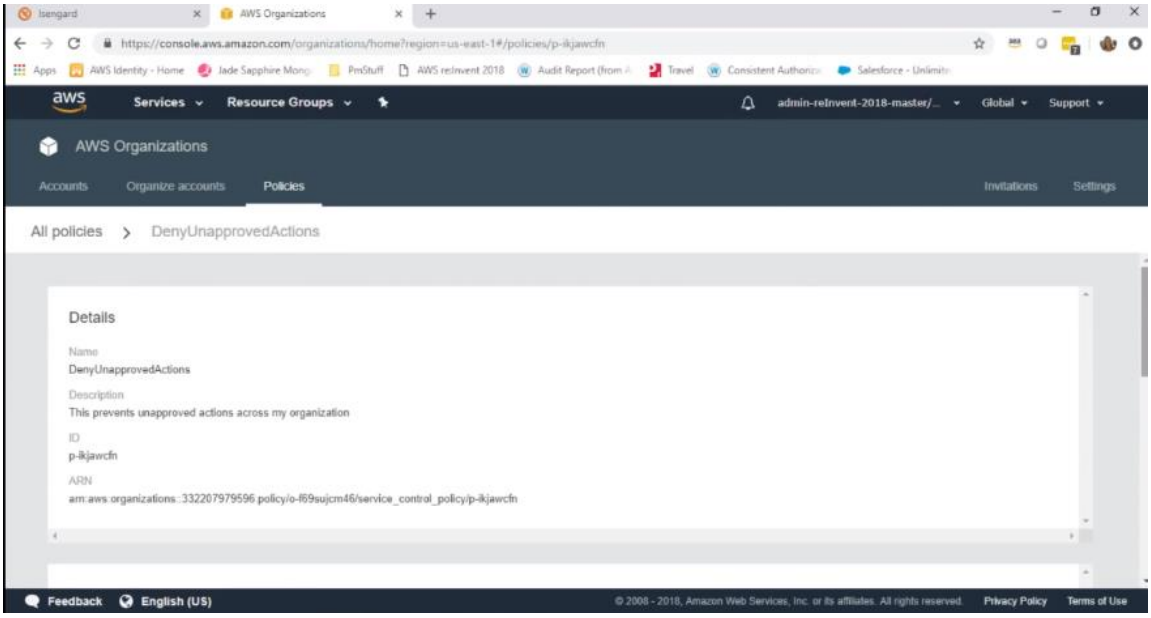
This Firefox page will be the developer that will assume different roles for each challenge

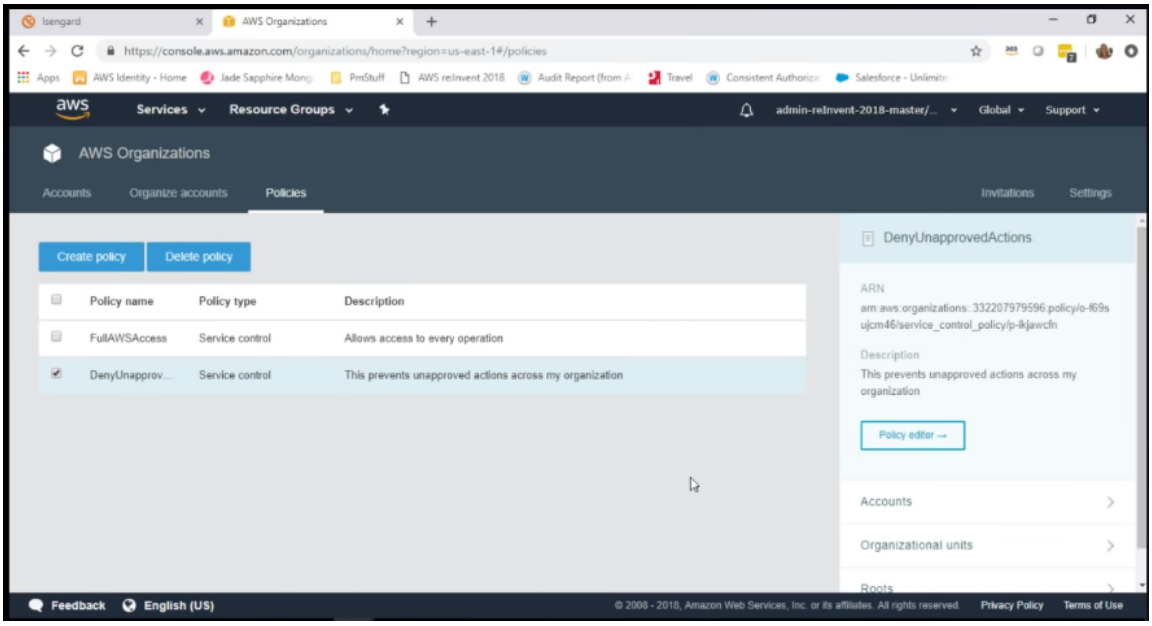


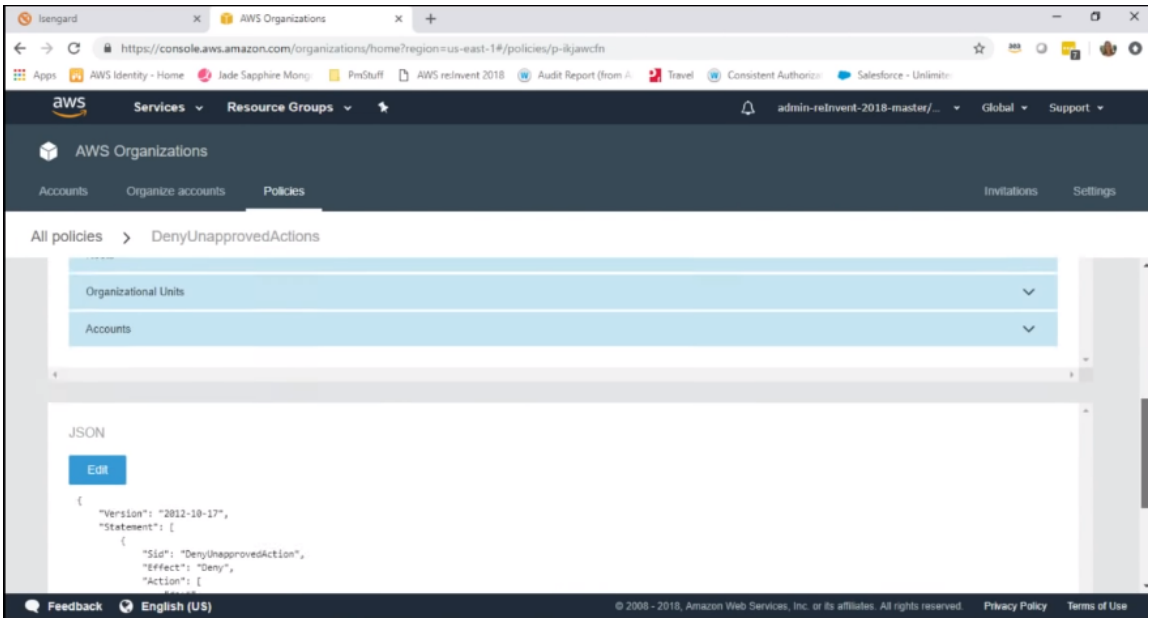


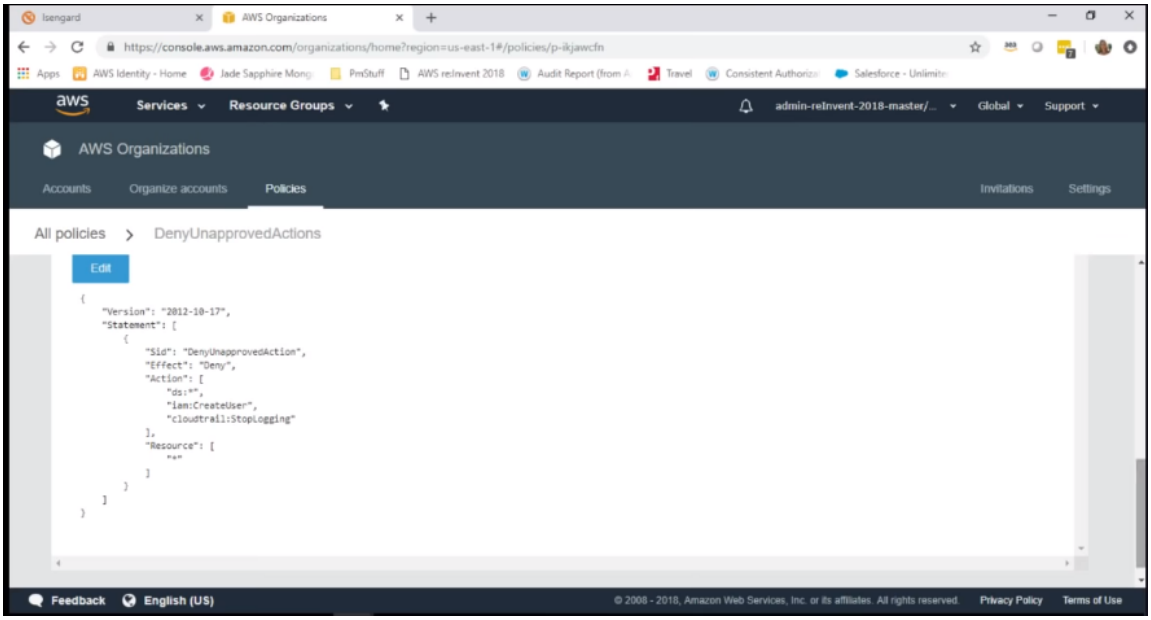
We have attached the **DenyUnapprovedActions** policy to the root of our master account here, we can also see the 2 organizational units OUs under this master account.

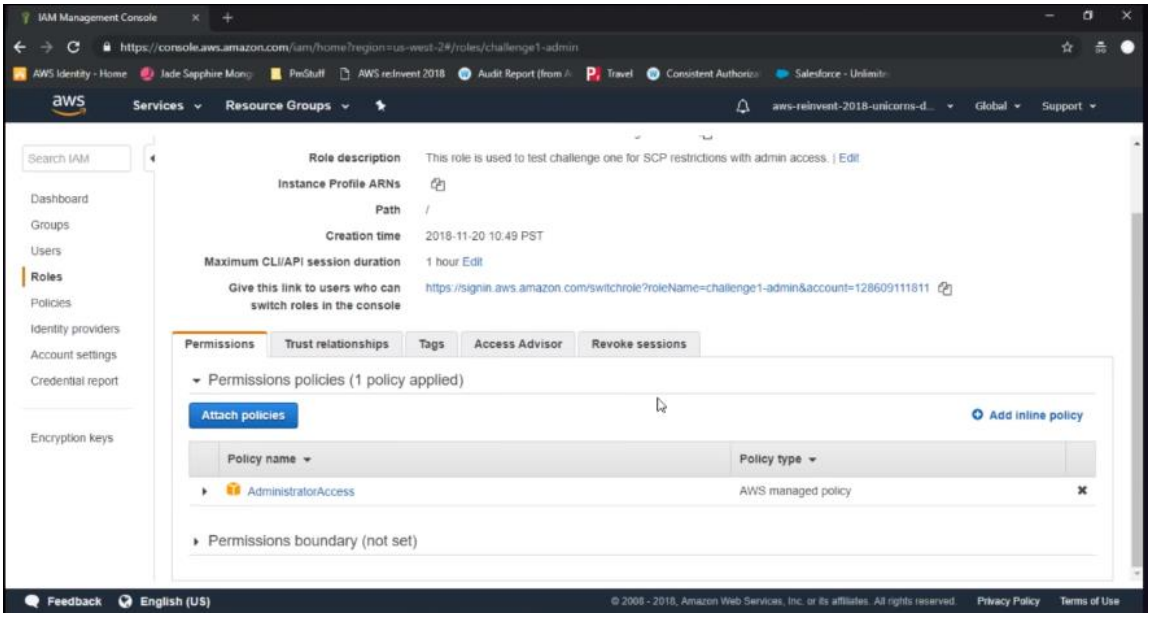



Isengard
AWS Organizations
https://console.aws.amazon.com/organizations/home?region=us-east-1#/policies/p-ikjawcfn
Services
Resource Groups
admin-reInvent-2018-master/...
Global
Support
AWS Organizations
Accounts
Organize accounts
Policies
Invitations
Settings
All policies
DenyUnapprovedActions
Details
Name
DenyUnapprovedActions
Description
This prevents unapproved actions across my organization
ID
p-ikjawcfn
ARN
arn:aws:organizations::332207979596:policy/o-f69sujcm46/service_control_policy/p-ikjawcfn
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

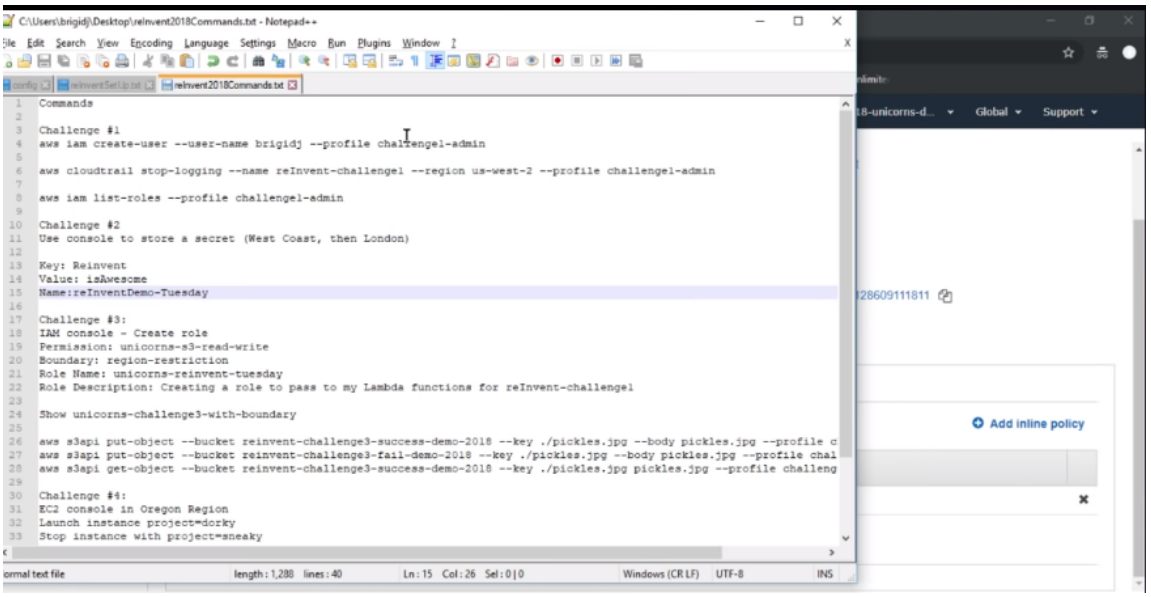

Isengard
AWS Organizations
https://console.aws.amazon.com/organizations/home?region=us-east-1#/policies
Services
Resource Groups
admin-reInvent-2018-master/...
Global
Support
AWS Organizations
Accounts
Organize accounts
Policies
Invitations
Settings
Create policy
Delete policy
Policy name
Policy type
Description
FullAWSAccess
Service control
Allows access to every operation
DenyUnapprov...
Service control
This prevents unapproved actions across my organization
DenyUnapprovedActions
ARN
arn:aws:organizations::332207979596:policy/o-f69s
ujcm46/service_control_policy/p-ikjawcfn
Description
This prevents unapproved actions across my organization
Policy editor →
Accounts
Organizational units
Roots
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

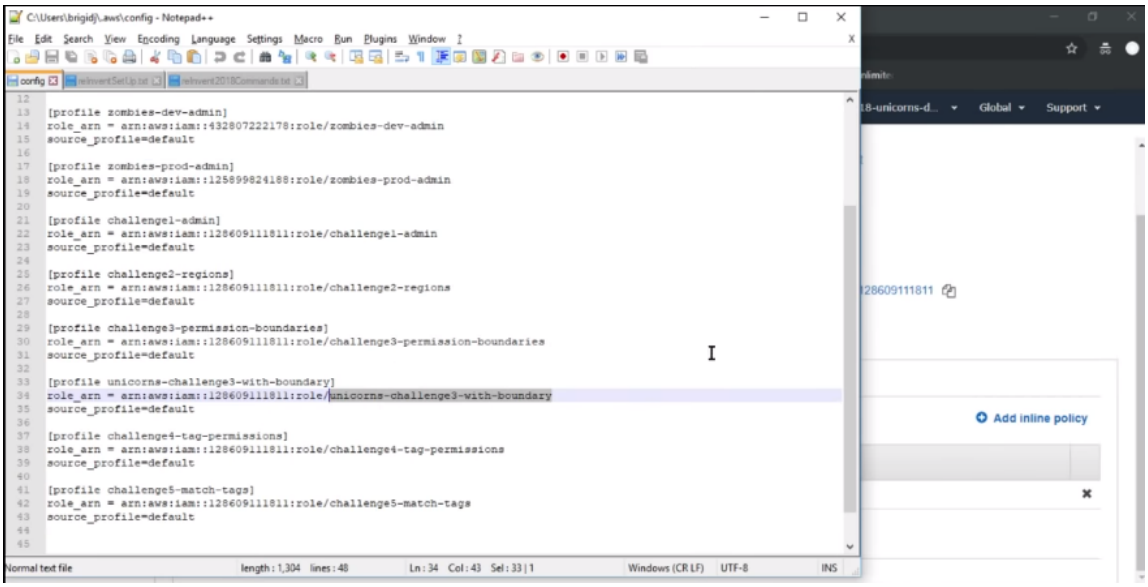

Isengard
AWS Organizations
https://console.aws.amazon.com/organizations/home?region=us-east-1#/policies/p-ikjawcfn
Services
Resource Groups
admin-reInvent-2018-master/...
Global
Support
AWS Organizations
Accounts
Organize accounts
Policies
Invitations
Settings
All policies
DenyUnapprovedActions
Organizational Units
Accounts
JSON
Edit
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "DenyUnapprovedAction",
"Effect": "Deny",
"Action": [
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



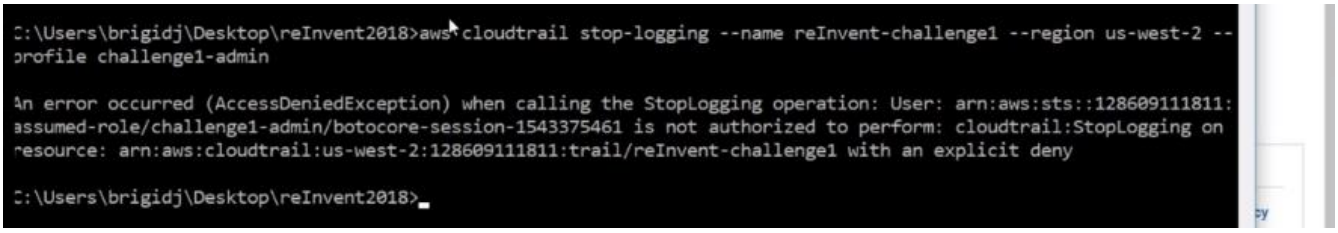


We will now work as an admin user here with the role above

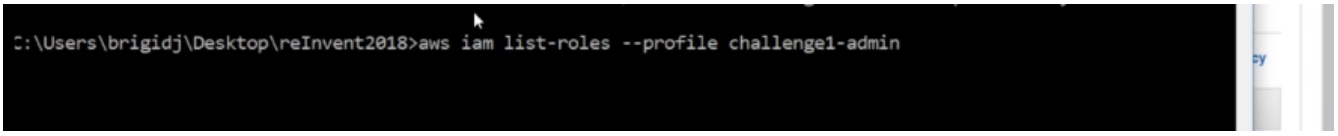




When the admin user tries to create a new IAM user, it gets denied because of the SCP



Trying to turn off CloudTrail logging also fails because of the SCP



Listing of roles in the account should pass because it is not denied in the SCP

Let's see SCPs in action

- Show SCP to deny access to modify AWS CloudTrail, Create IAM users, and AWS Directory Services
- Use the CLI as an administrator in the Unicorn-dev account and try to:
  - Create an IAM user
  - List roles
  - Stop logging in AWS CloudTrail

Challenge #2

Ensure your developers can create resources, but only in approved regions.

Pro Tip: Use the RequestedRegion AWS condition

Policy for challenge #2

```
{
        "Effect": "Allow",
        "Action": [
            "secretsmanager:*",
            "lambda:*",
            "s3:PutObject",
            "s3:GetObject",
            "s3:DeleteObject"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestedRegion": [
                    "us-west-1",
                    "us-west-2"
                ]
            }
        }
    }
```

This is what the policy looks like and we have specified the allowed regions in the policy

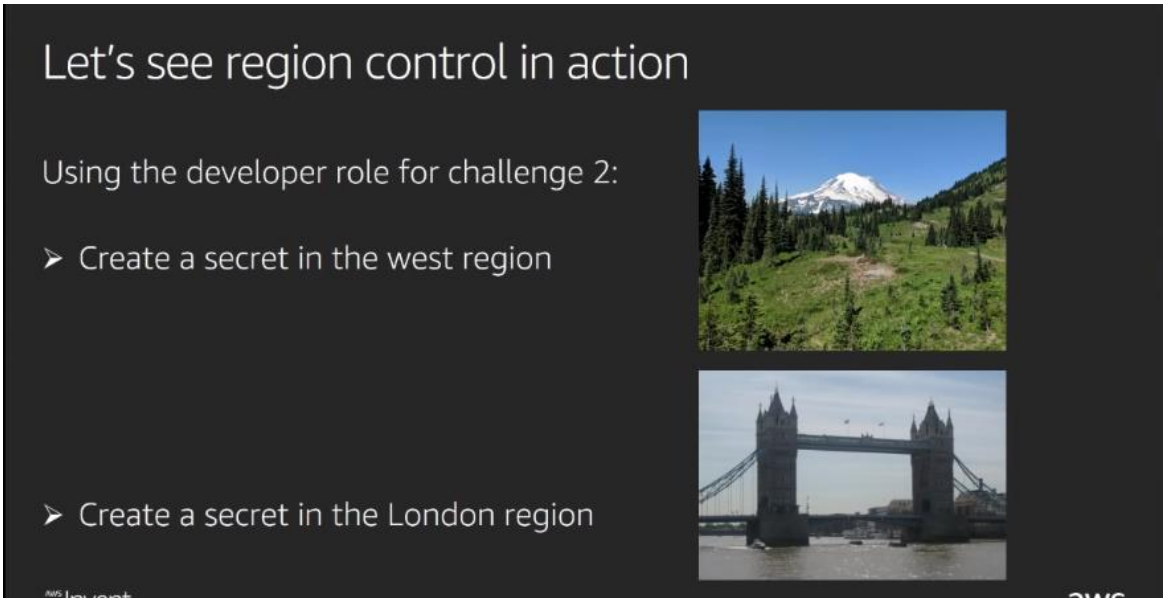
Policy for challenge #2

```
{
        "Effect": "Allow",
        "Action": "ec2:RunInstances",
        "Resource": [
            "arn:aws:ec2:*:*:subnet/*",
            "arn:aws:ec2:*:*:key-pair/*",
            "arn:aws:ec2:*:*:instance/*",
            "arn:aws:ec2:*::snapshot/*",
            "arn:aws:ec2:*:*:launch-template/*",
            "arn:aws:ec2:*:*:volume/*",
            "arn:aws:ec2:*:*:security-group/*",
            "arn:aws:ec2:*:*:placement-group/*",
            "arn:aws:ec2:*:*:network-interface/*",
            "arn:aws:ec2:*::image/*"
        ],
        "Condition": {
            "StringEquals": {
                "aws:RequestedRegion": ["us-west-1","us-west-2"]
            }
        }
    }
```

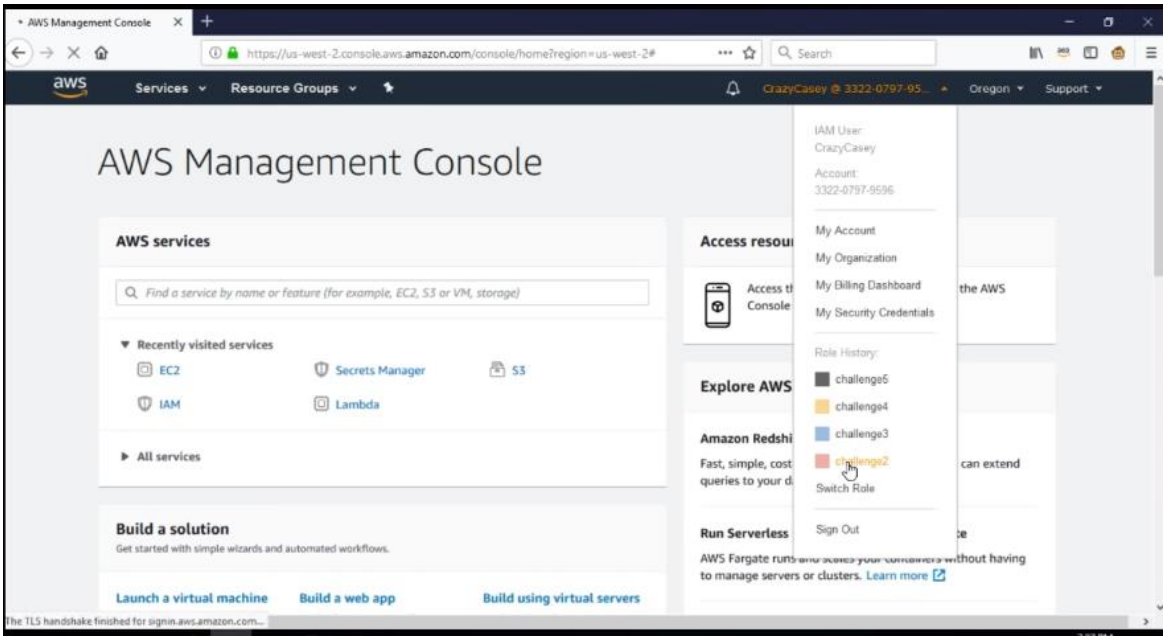

We also specify that EC2 instances should be allowed only in the 2 specified regions

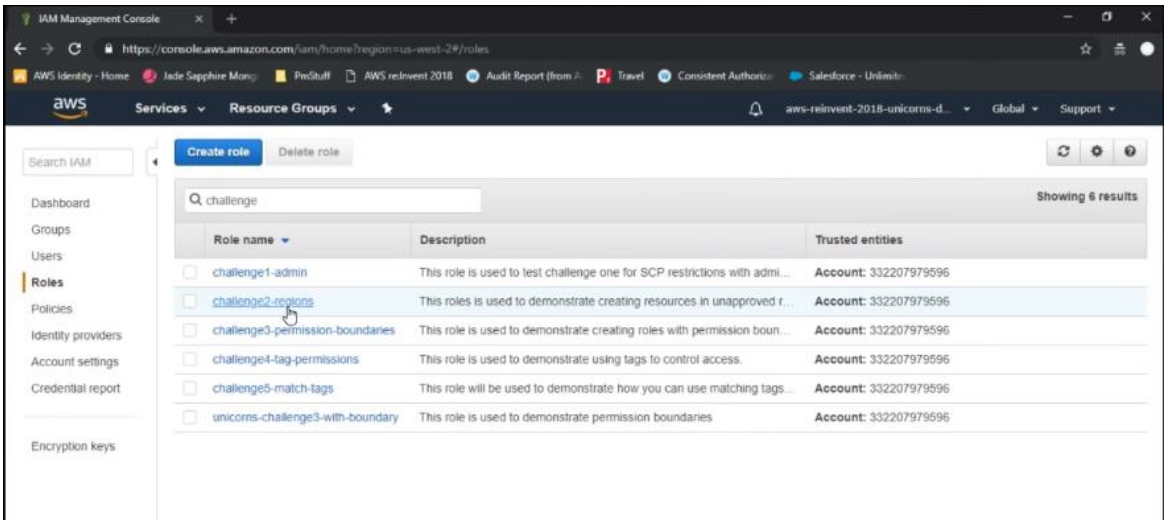
Policy for challenge #2

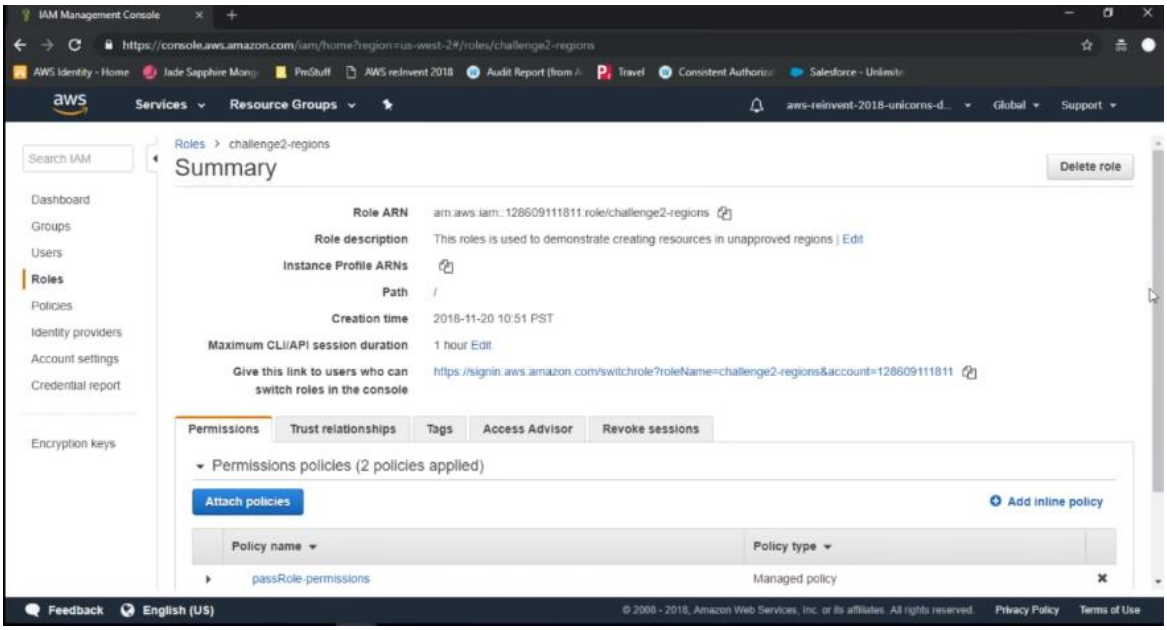
```
{
        "Effect": "Allow",
        "Action": [
            "ec2:Describe*",
            "ec2:Get*",
            "s3:ListBucket",
            "s3:ListAllMyBuckets",
            "iam:list*"
        ],
        "Resource": "*"
    }
```

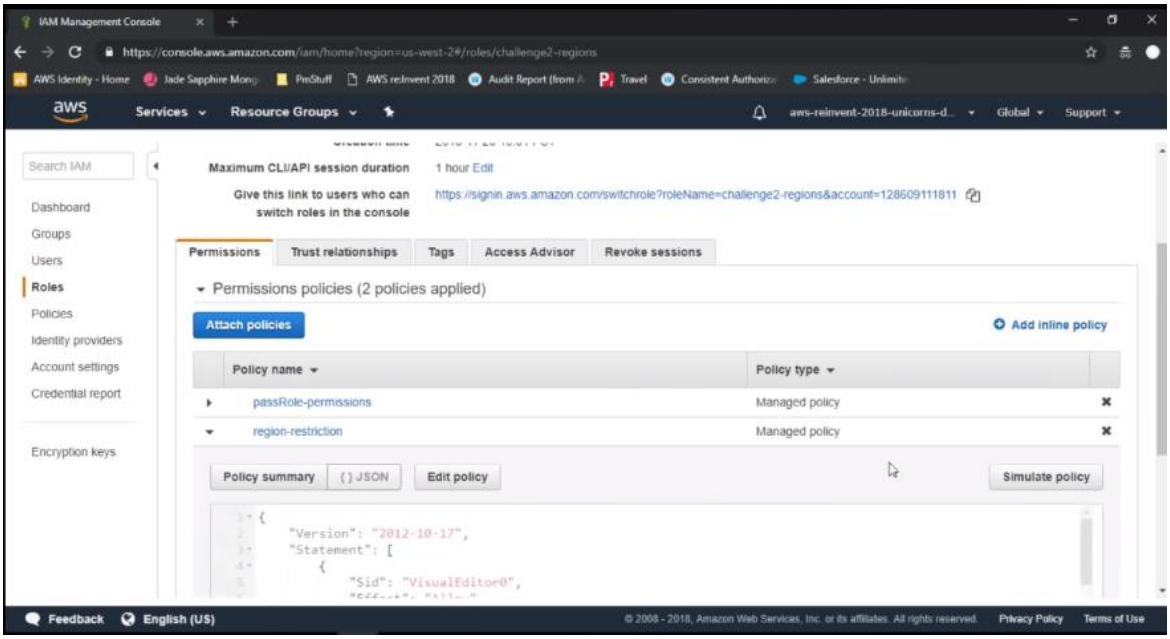


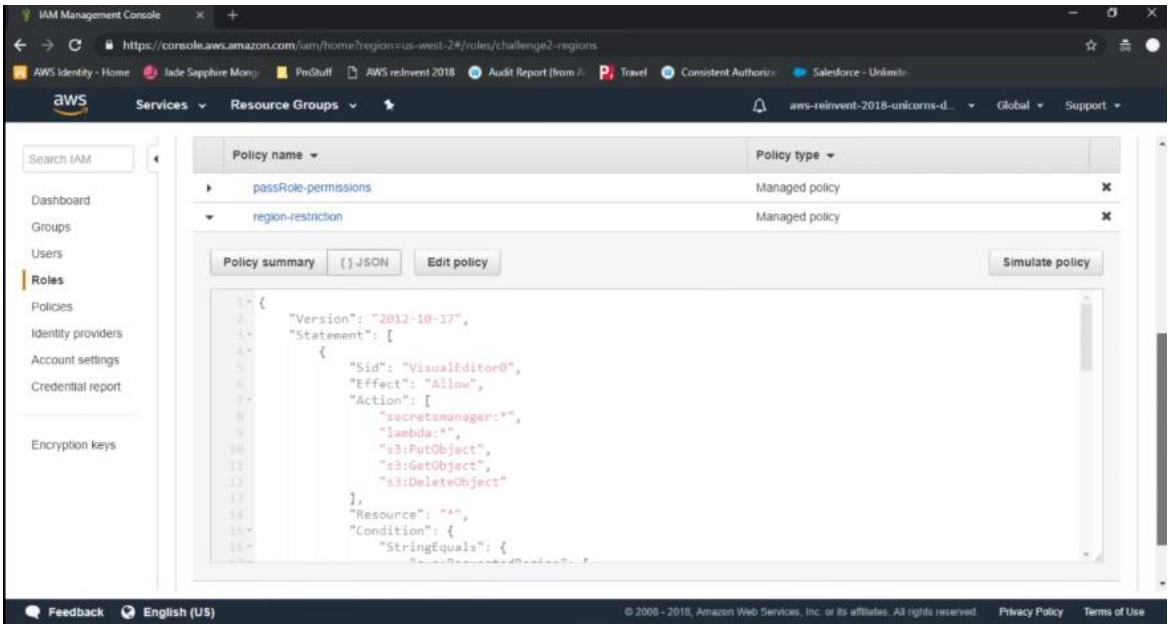
We will try and create a secret using the AWS Secrets Manager in the west region and also in the London region which should not work

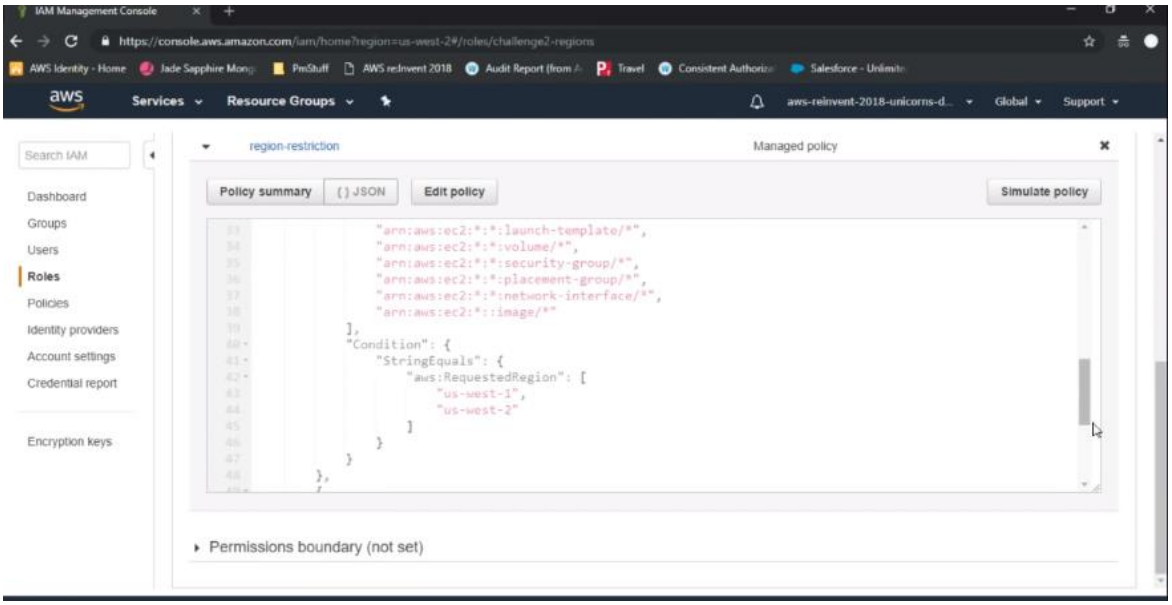


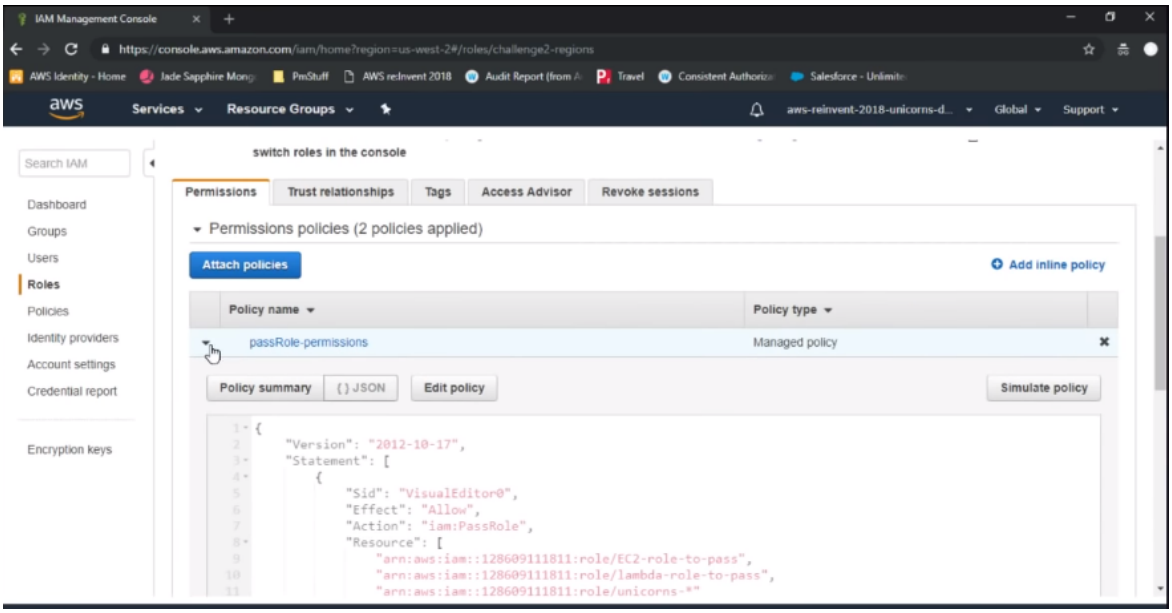


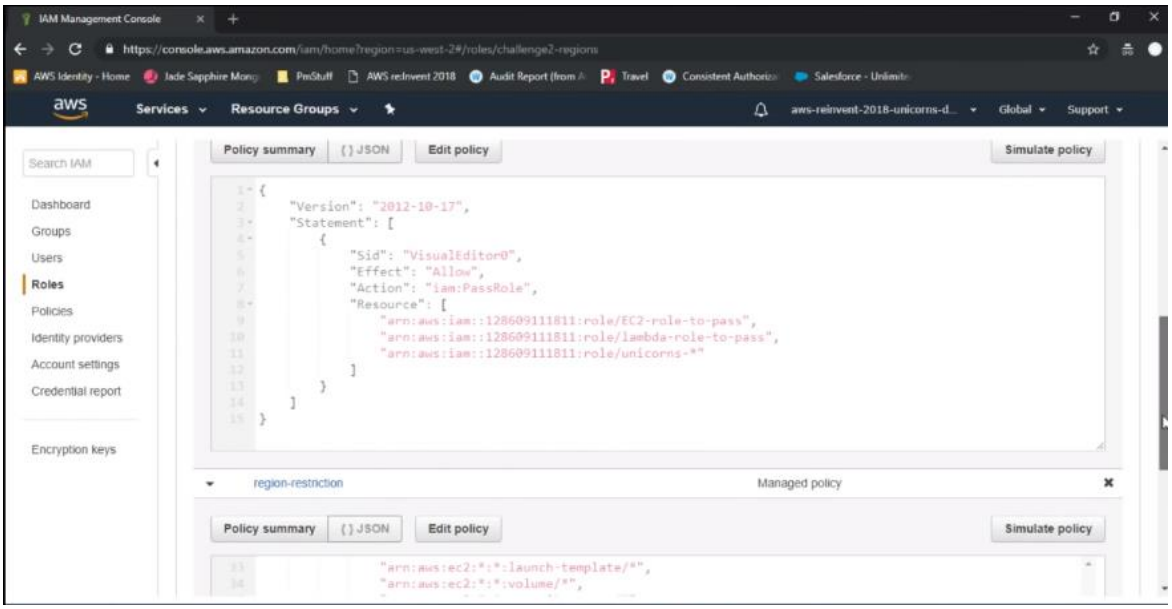

IAM Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge2-regions
Services
Resource Groups
Roles > challenge2-regions
Summary
Delete role
Role ARN
arn:aws:iam::128609111811:role/challenge2-regions
Role description
This roles is used to demonstrate creating resources in unapproved regions | Edit
Instance Profile ARNs
Path
/
Creation time
2018-11-20 10:51 PST
Maximum CLI/API session duration
1 hour Edit
Give this link to users who can switch roles in the console
https://signin.aws.amazon.com/switchrole?roleName=challenge2-regions&account=128609111811
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (2 policies applied)
Attach policies
Add inline policy
Policy name
Policy type
passRole-permissions
Managed policy
Feedback
English (US)
Privacy Policy
Terms of Use


IAM Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge2-regions
Maximum CLI/API session duration
1 hour Edit
Give this link to users who can switch roles in the console
https://signin.aws.amazon.com/switchrole?roleName=challenge2-regions&account=128609111811
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (2 policies applied)
Attach policies
Add inline policy
Policy name
Policy type
passRole-permissions
Managed policy
region-restriction
Managed policy
Policy summary
{} JSON
Edit policy
Simulate policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
Feedback
English (US)
Privacy Policy
Terms of Use

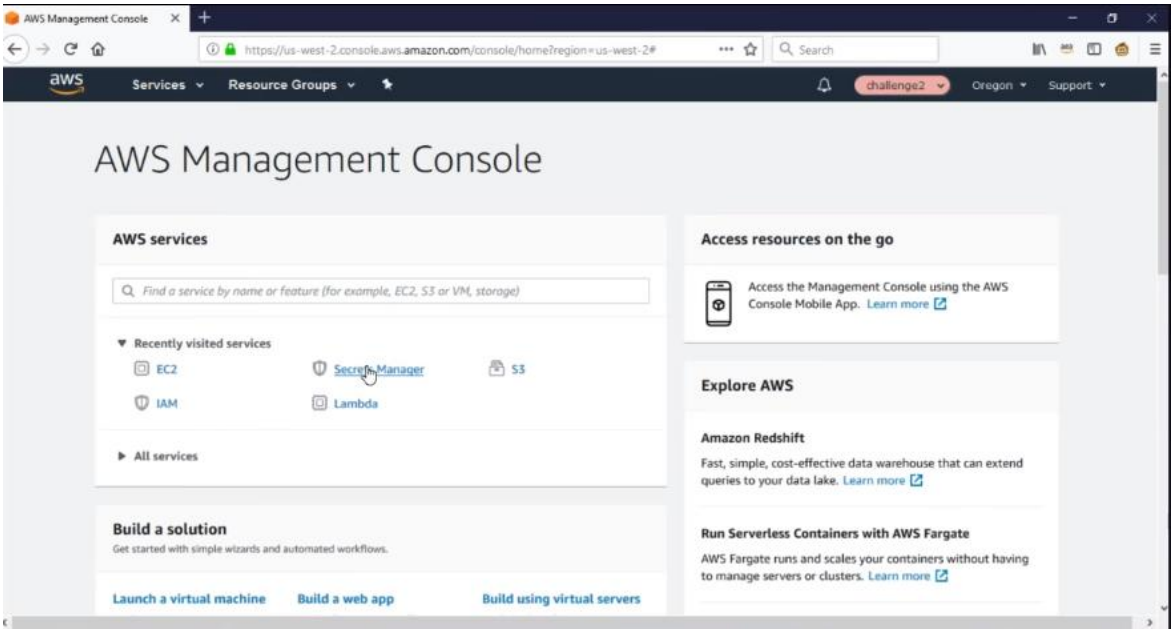

IAM Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge2-regions
Policy name
Policy type
passRole-permissions
Managed policy
region-restriction
Managed policy
Policy summary
{} JSON
Edit policy
Simulate policy
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "VisualEditor0",
"Effect": "Allow",
"Action": [
"secretsmanager:*",
"lambda:*",
"s3:PutObject",
"s3:GetObject",
"s3:DeleteObject"
],
"Resource": "*",
"Condition": {
"StringEquals": {
Feedback
English (US)
Privacy Policy
Terms of Use

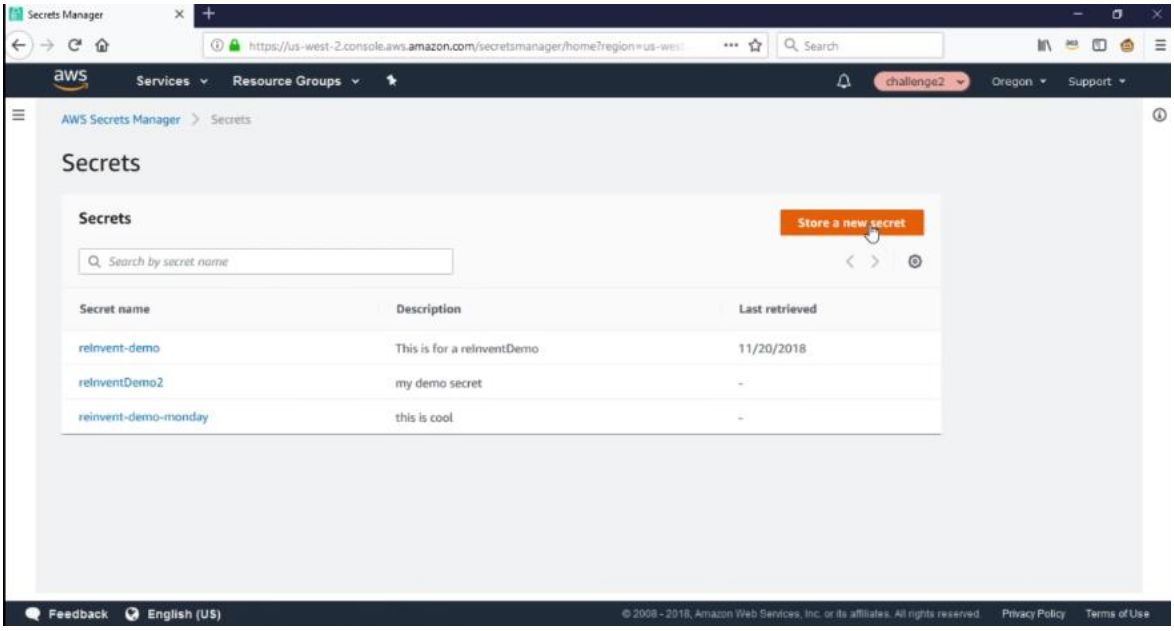


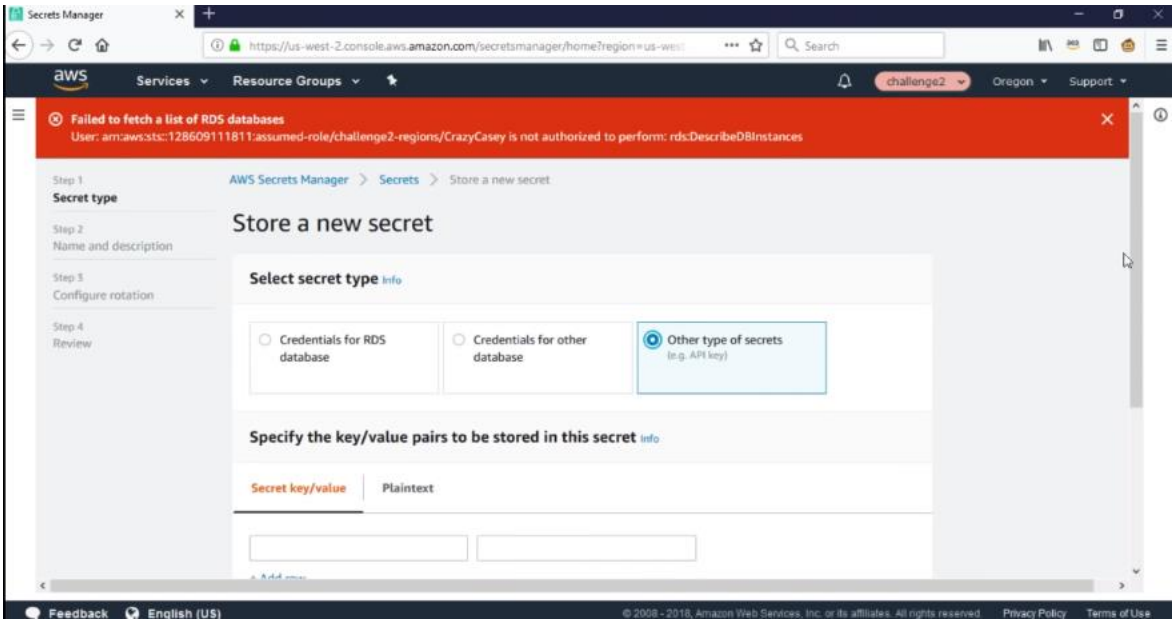


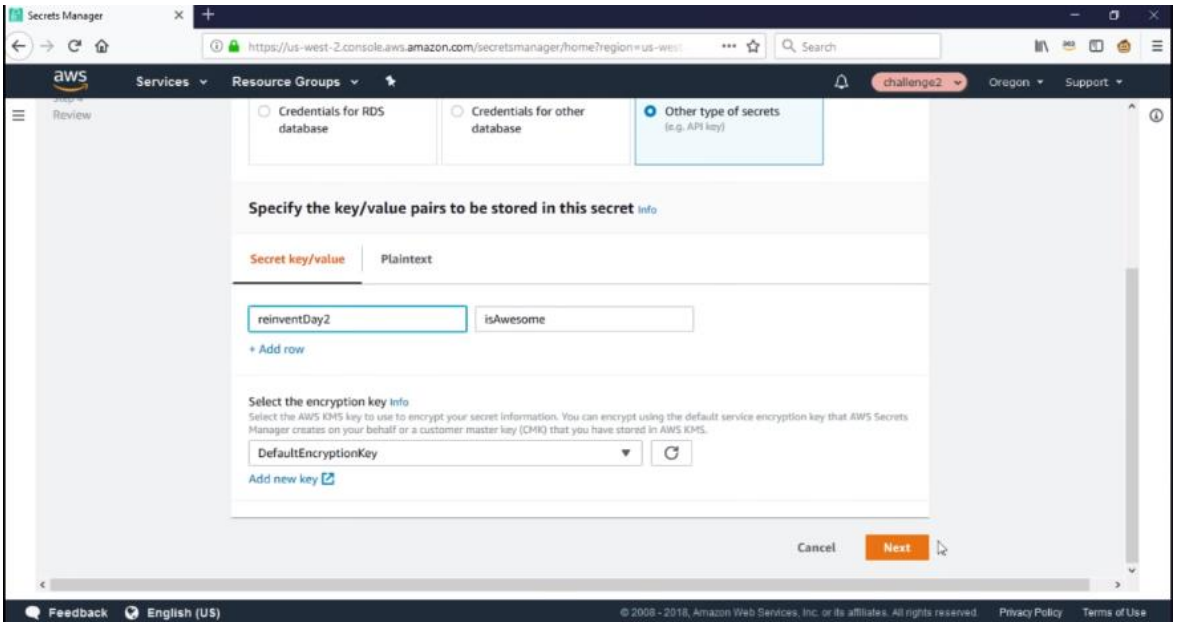


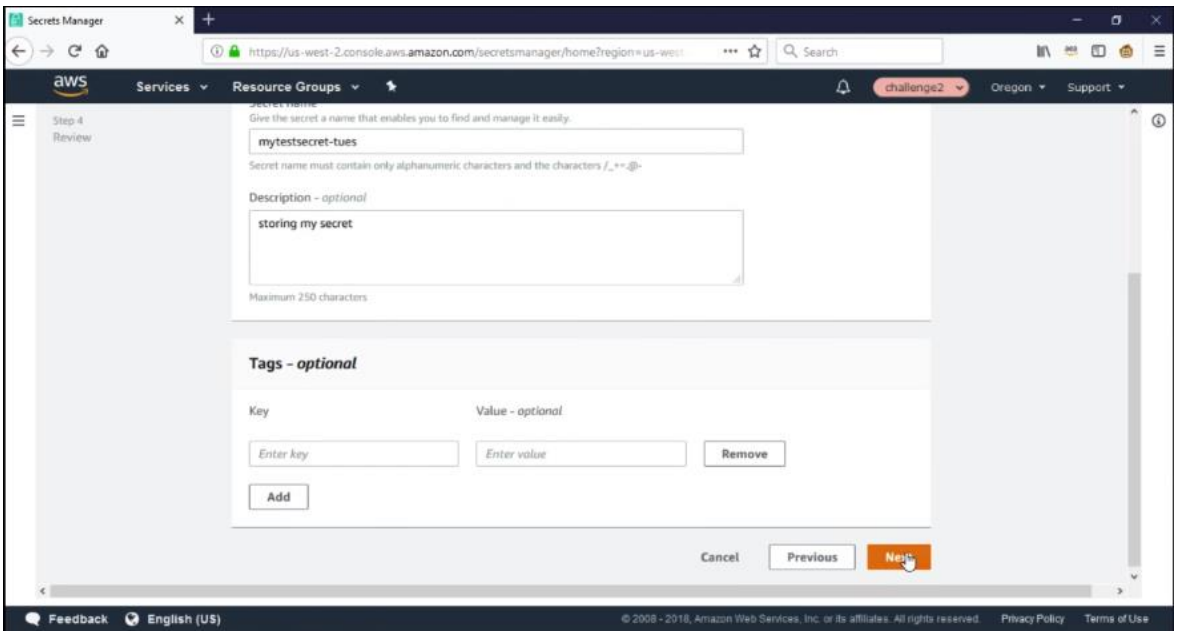
This **passRole** policy will allow us to test launching some EC2 instances or some Lambda functions.

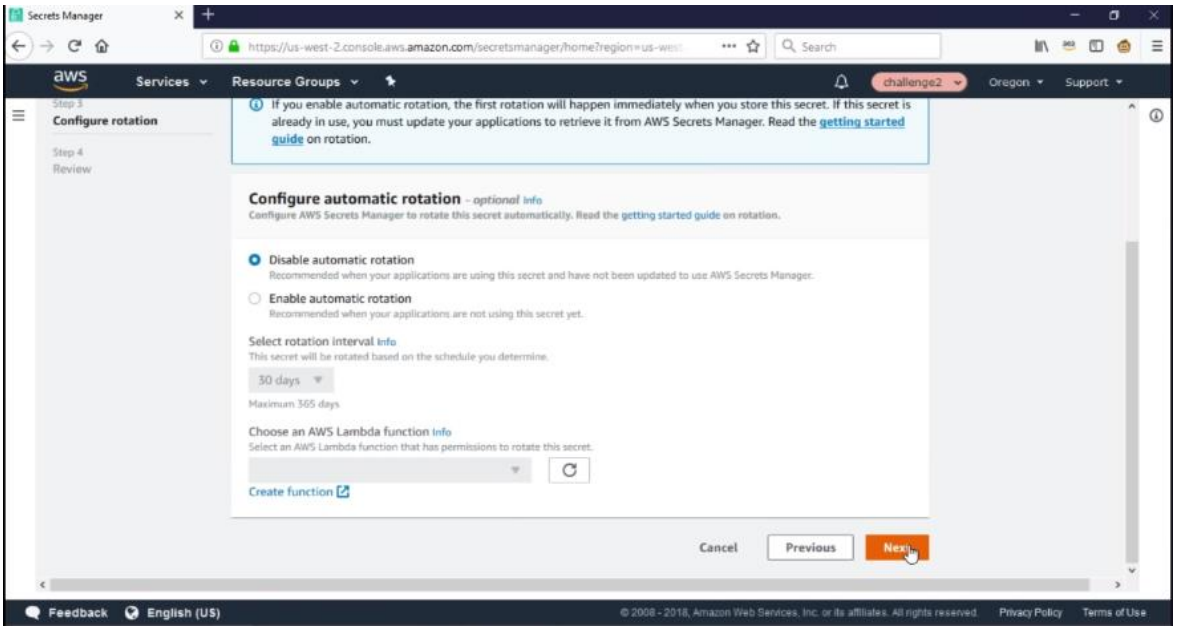

AWS Management Console
https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2#
Search
Services
Resource Groups
challenge2
Oregon
Support
AWS Management Console
AWS services
Find a service by name or feature (for example, EC2, S3 or VM, storage)
Recently visited services
EC2
Secrets Manager
S3
IAM
Lambda
All services
Build a solution
Get started with simple wizards and automated workflows.
Launch a virtual machine
Build a web app
Build using virtual servers
Access resources on the go
Access the Management Console using the AWS Console Mobile App. Learn more
Explore AWS
Amazon Redshift
Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more
Run Serverless Containers with AWS Fargate
AWS Fargate runs and scales your containers without having to manage servers or clusters. Learn more

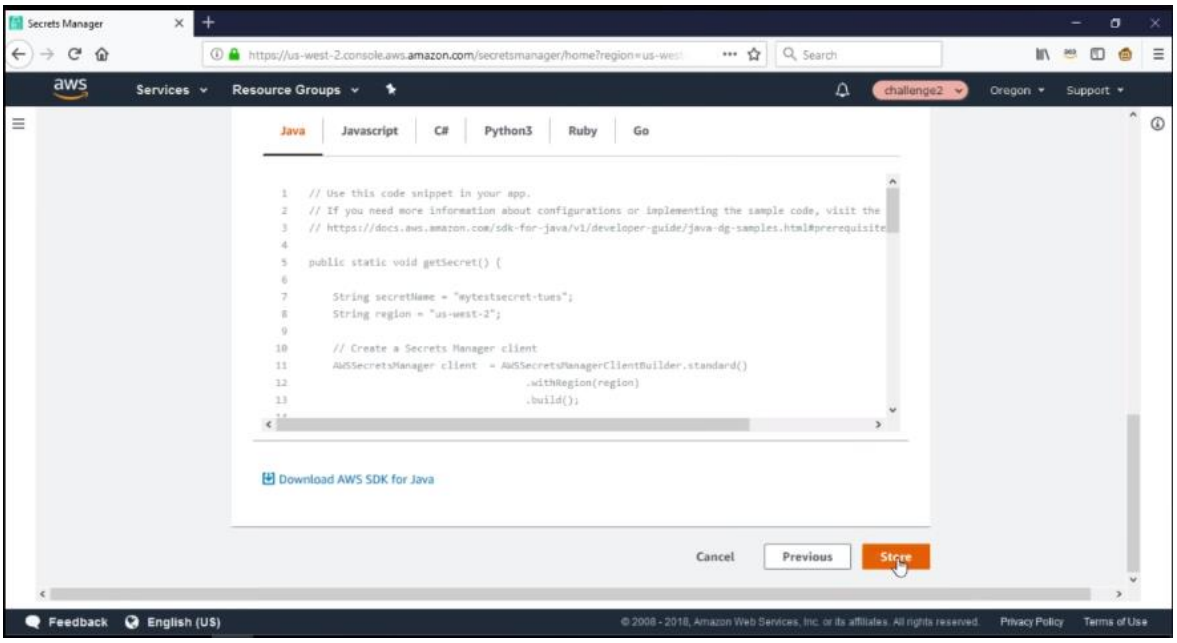

Secrets Manager
https://us-west-2.console.aws.amazon.com/secretsmanager/home?region=us-west
Search
Services
Resource Groups
challenge2
Oregon
Support
AWS Secrets Manager > Secrets
Secrets
Secrets
Store a new secret
Search by secret name
Secret name
Description
Last retrieved
reInvent-demo
This is for a reInventDemo
11/20/2018
reInventDemo2
my demo secret
-
reinvent-demo-monday
this is cool
-
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Secrets Manager
https://us-west-2.console.aws.amazon.com/secretsmanager/home?region=us-west
Search
Services
Resource Groups
challenge2
Oregon
Support
Failed to fetch a list of RDS databases
User: arn:aws:sts::128609111811:assumed-role/challenge2-regions/CrazyCasey is not authorized to perform: rds:DescribeDBInstances
Step 1
Secret type
Step 2
Name and description
Step 3
Configure rotation
Step 4
Review
AWS Secrets Manager > Secrets > Store a new secret
Store a new secret
Select secret type Info
Credentials for RDS database
Credentials for other database
Other type of secrets
(e.g. API key)
Specify the key/value pairs to be stored in this secret Info
Secret key/value
Plaintext
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

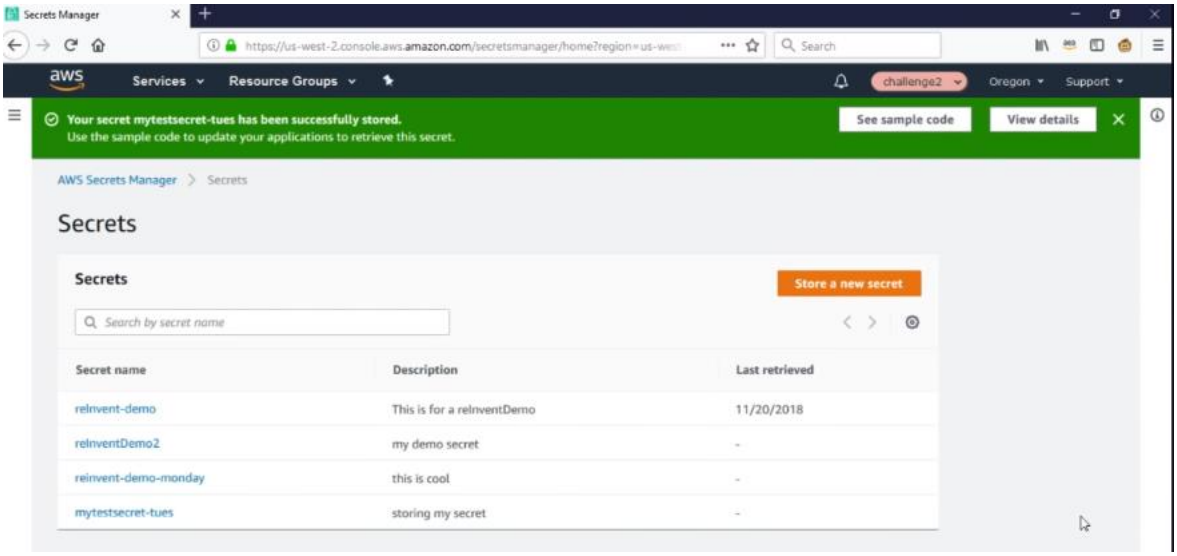

Secrets Manager
Services
Resource Groups
challenge2
Oregon
Support
Credentials for RDS database
Credentials for other database
Other type of secrets
(e.g. API key)
Specify the key/value pairs to be stored in this secret Info
Secret key/value
Plaintext
reinventDay2
isAwesome
+ Add row
Select the encryption key Info
DefaultEncryptionKey
Add new key
Cancel
Next
Feedback
English (US)
Privacy Policy
Terms of Use


Secrets Manager
Services
Resource Groups
challenge2
Oregon
Support
Step 4
Review
mytestsecret-tues
Description - optional
storing my secret
Maximum 250 characters
Tags - optional
Key
Value - optional
Enter key
Enter value
Remove
Add
Cancel
Previous
Next
Feedback
English (US)
Privacy Policy
Terms of Use

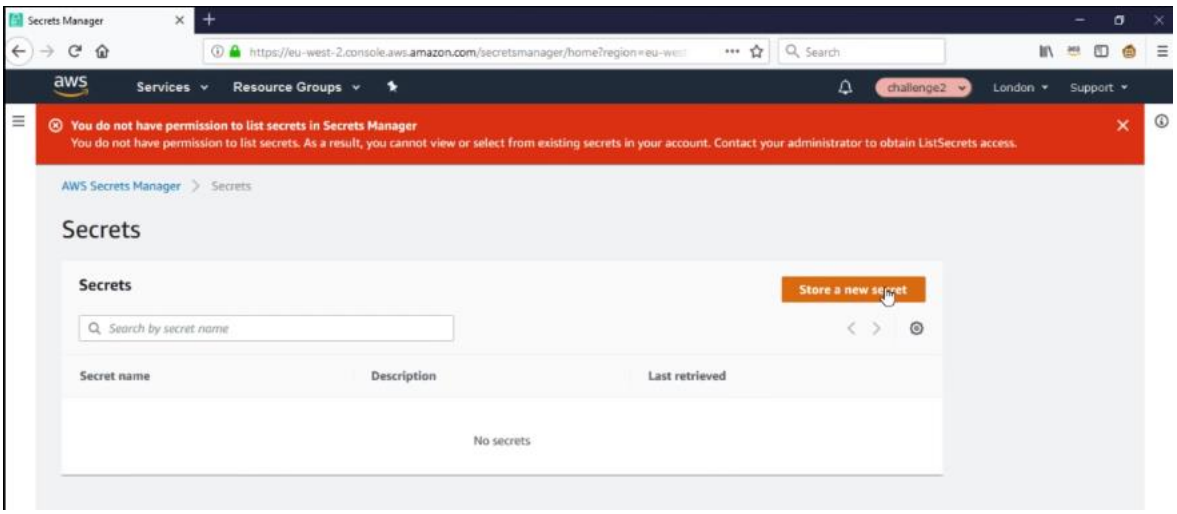

Secrets Manager
Services
Resource Groups
challenge2
Oregon
Support
Configure rotation
Step 4
Review
If you enable automatic rotation, the first rotation will happen immediately when you store this secret. If this secret is already in use, you must update your applications to retrieve it from AWS Secrets Manager. Read the getting started guide on rotation.
Configure automatic rotation - optional Info
Disable automatic rotation
Enable automatic rotation
Select rotation interval Info
30 days
Choose an AWS Lambda function Info
Create function
Cancel
Previous
Next
Feedback
English (US)
Privacy Policy
Terms of Use



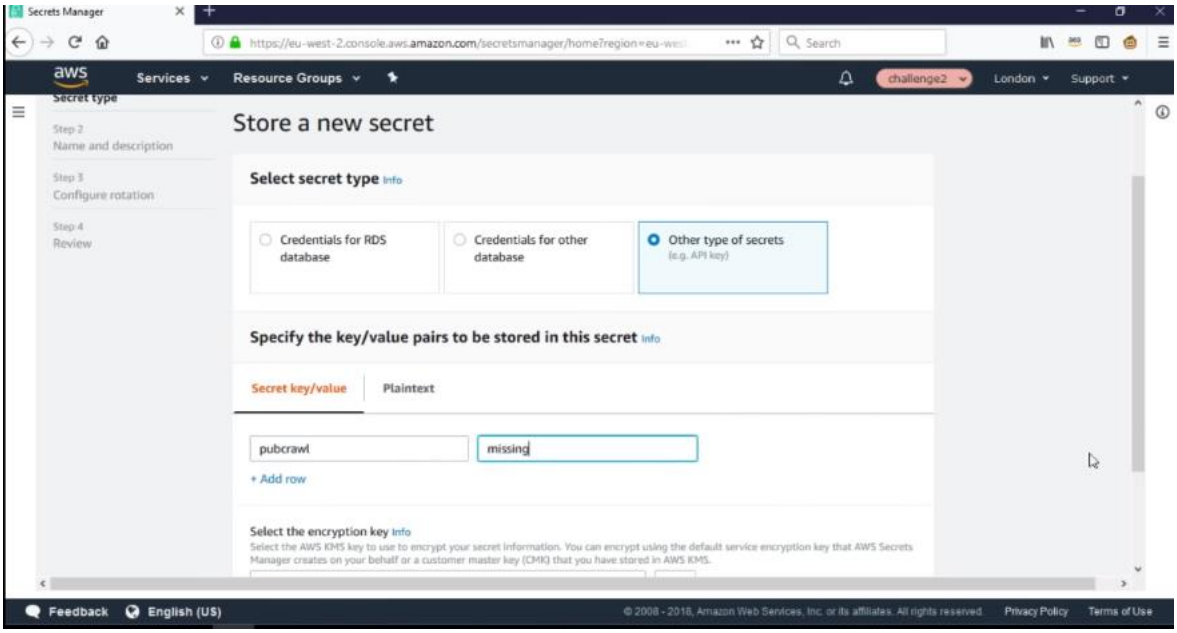
This is the code we can use in our app to retrieve the secret

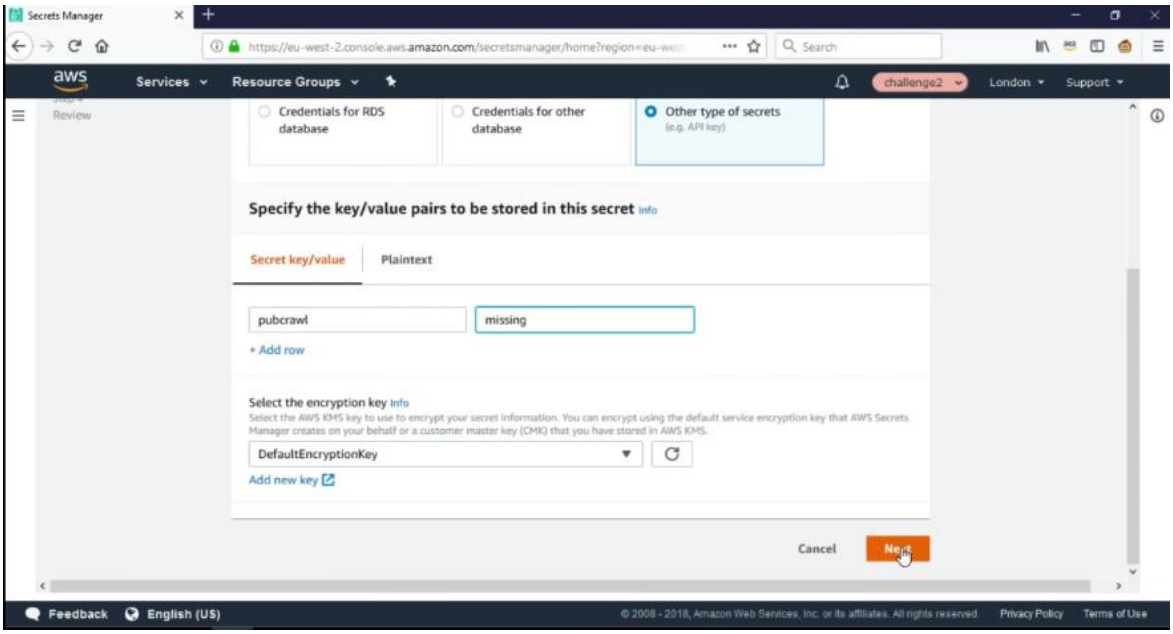


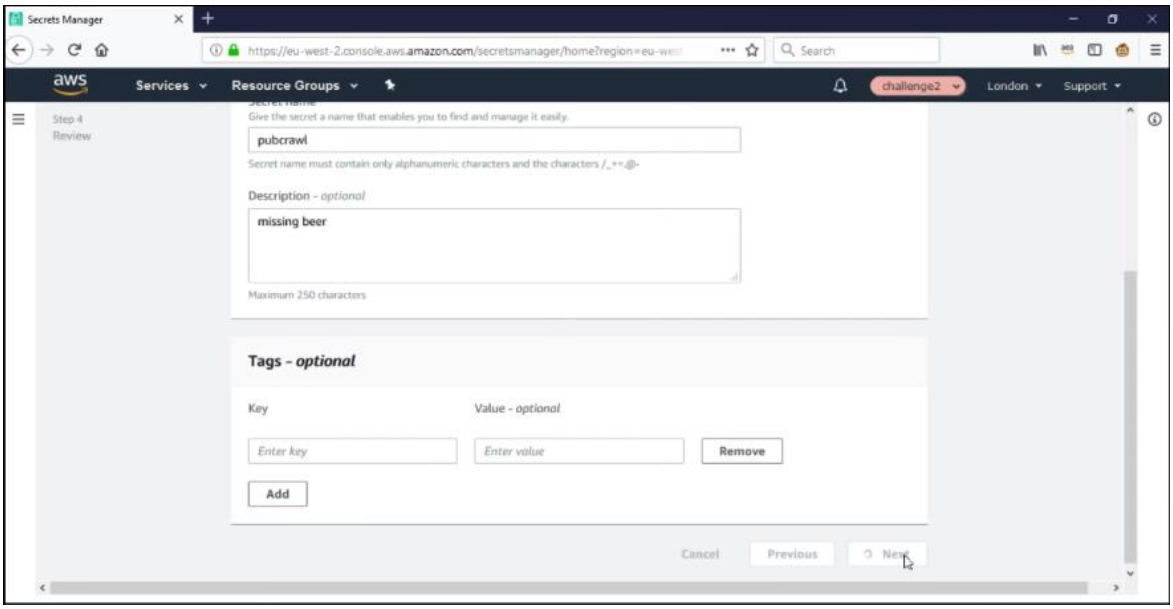
We are successfully allowed to create a secret in the west region due to our policy.

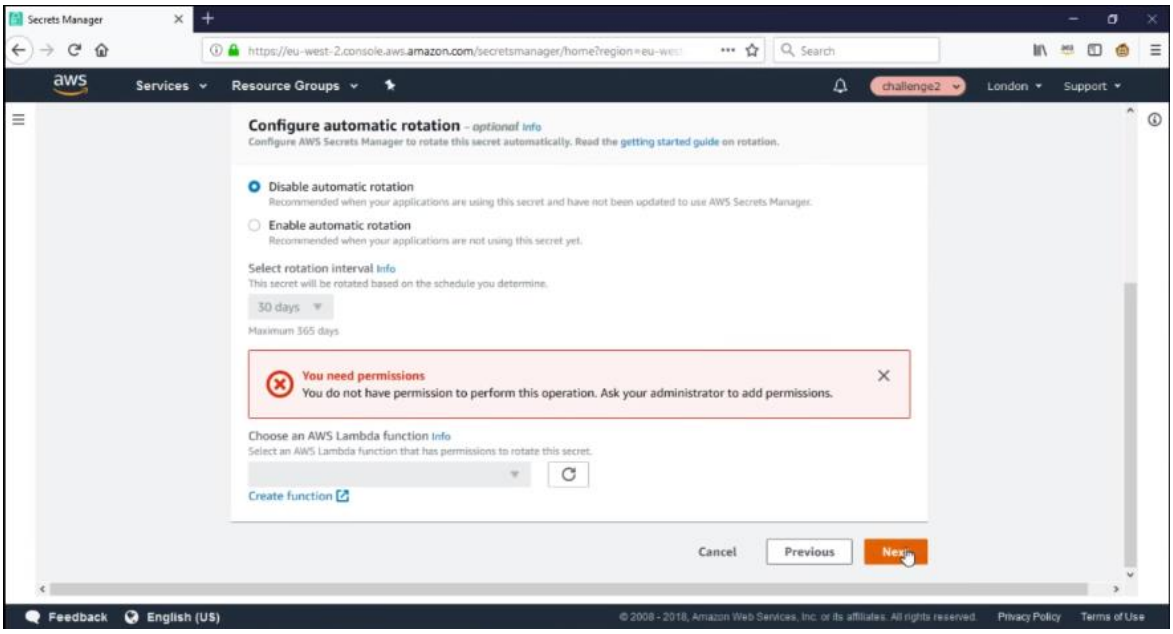


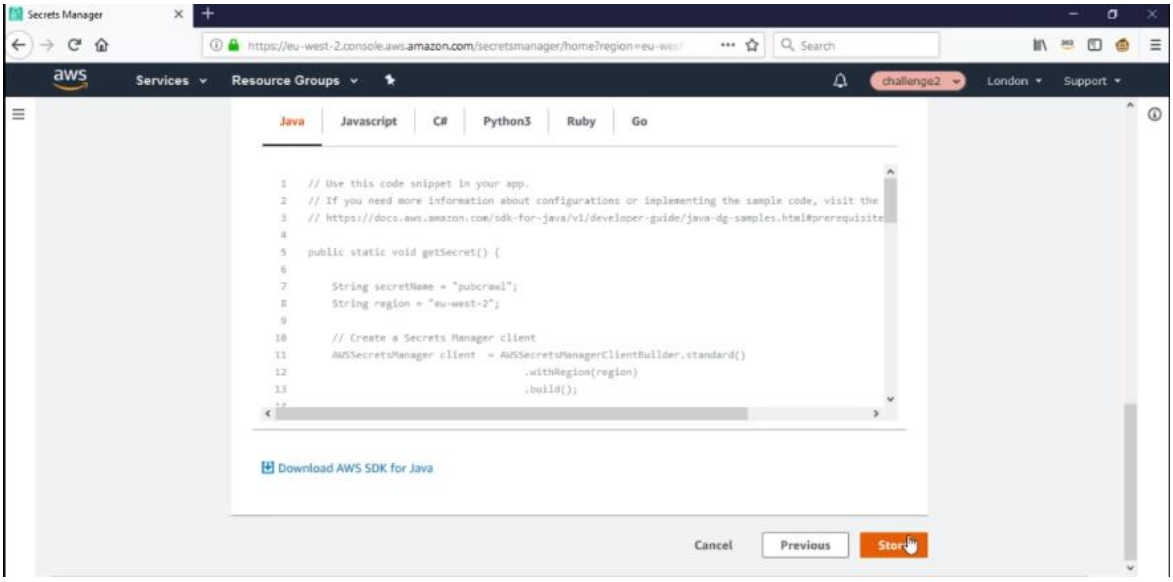
We are now in the London region and cannot even list the available secrets, we will now try to create a secret here

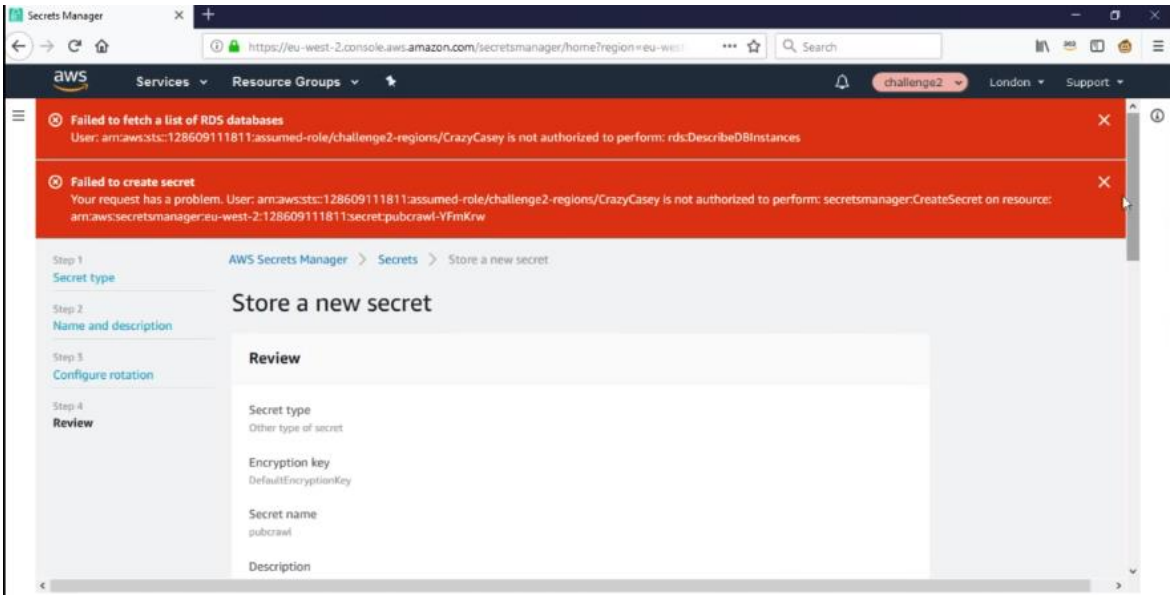

Secrets Manager
https://eu-west-2.console.aws.amazon.com/secretsmanager/home?region=eu-west
Search
aws
Services
Resource Groups
challenge2
London
Support
Secret type
Step 2
Name and description
Step 3
Configure rotation
Step 4
Review
Store a new secret
Select secret type Info
Credentials for RDS database
Credentials for other database
Other type of secrets
(e.g. API key)
Specify the key/value pairs to be stored in this secret Info
Secret key/value
Plaintext
pubcrawl
missing
+ Add row
Select the encryption key Info
Select the AWS KMS key to use to encrypt your secret information. You can encrypt using the default service encryption key that AWS Secrets Manager creates on your behalf or a customer master key (CMK) that you have stored in AWS KMS.
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Secrets Manager
https://eu-west-2.console.aws.amazon.com/secretsmanager/home?region=eu-west
Search
aws
Services
Resource Groups
challenge2
London
Support
Review
Credentials for RDS database
Credentials for other database
Other type of secrets
(e.g. API key)
Specify the key/value pairs to be stored in this secret Info
Secret key/value
Plaintext
pubcrawl
missing
+ Add row
Select the encryption key Info
Select the AWS KMS key to use to encrypt your secret information. You can encrypt using the default service encryption key that AWS Secrets Manager creates on your behalf or a customer master key (CMK) that you have stored in AWS KMS.
DefaultEncryptionKey
Add new key
Cancel
Next
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Secrets Manager
https://eu-west-2.console.aws.amazon.com/secretsmanager/home?region=eu-west
Search
aws
Services
Resource Groups
challenge2
London
Support
Step 4
Review
Give the secret a name that enables you to find and manage it easily.
pubcrawl
Secret name must contain only alphanumeric characters and the characters /_+=.@-
Description - optional
missing beer
Maximum 250 characters
Tags - optional
Key
Value - optional
Enter key
Enter value
Remove
Add
Cancel
Previous
Next







This failed as expected in the London region

Let's see region control in action

Using the developer role for challenge 2:

- Create a secret in the west region

- Create a secret in the London region

Challenge #3

Enable your developers to create IAM roles to pass to EC2 and Lambda, but ensure they cannot exceed their own permissions.

Pro Tip: Require and use role naming conventions to control the roles developers can manage.

We want to allow the developers to create their IAM roles to pass to EC2 or Lambda, but we want to make sure they cannot escalate their privileges beyond this scope. We have to constrain the developers with the use of permission boundaries and also to their own little world using naming-prefixes.

Four parts required for permission boundaries

- Allow create managed policies
- Allow create role, but only with a specific permission boundary
  - *This is a condition with a pointer to an existing managed policy*
- Allow attach managed policies, but only to roles with a specific boundary
- Allow passRole for these roles using a naming requirement

Policy for Challenge #3

Allow create managed policies

```
{
      "Effect": "Allow",
      "Action": [
            "iam:CreatePolicy",
            "iam:CreatePolicyVersion",
            "iam:DeletePolicyVersion"
      ],
      "Resource": "arn:aws:iam::128609111811:policy/unicorns-*"
}
```

This is how we can constrain the developer to the unicorn-* prefixes.

Policy for Challenge #3

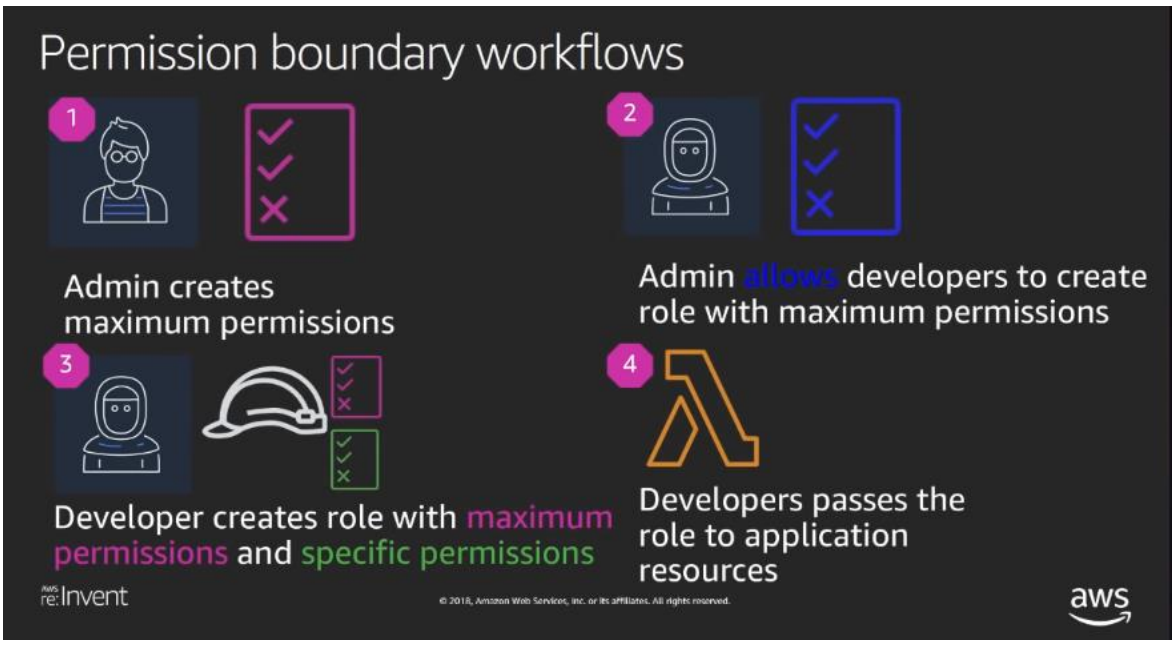
Allow create role, but only with a specific permission boundary

Allow attach managed policies, but only to roles with a specific boundary

```
{
      "Effect": "Allow",
      "Action": [
            "iam:DetachRolePolicy",
            "iam:CreateRole",
            "iam:AttachRolePolicy"
      ],
      "Resource": "arn:aws:iam::128609111811:role/unicorns-*",
      "Condition": {
            "StringEquals": {
                  "iam:PermissionsBoundary":
      "arn:aws:iam::128609111811:policy/region-restriction"
            }
      }
}
```

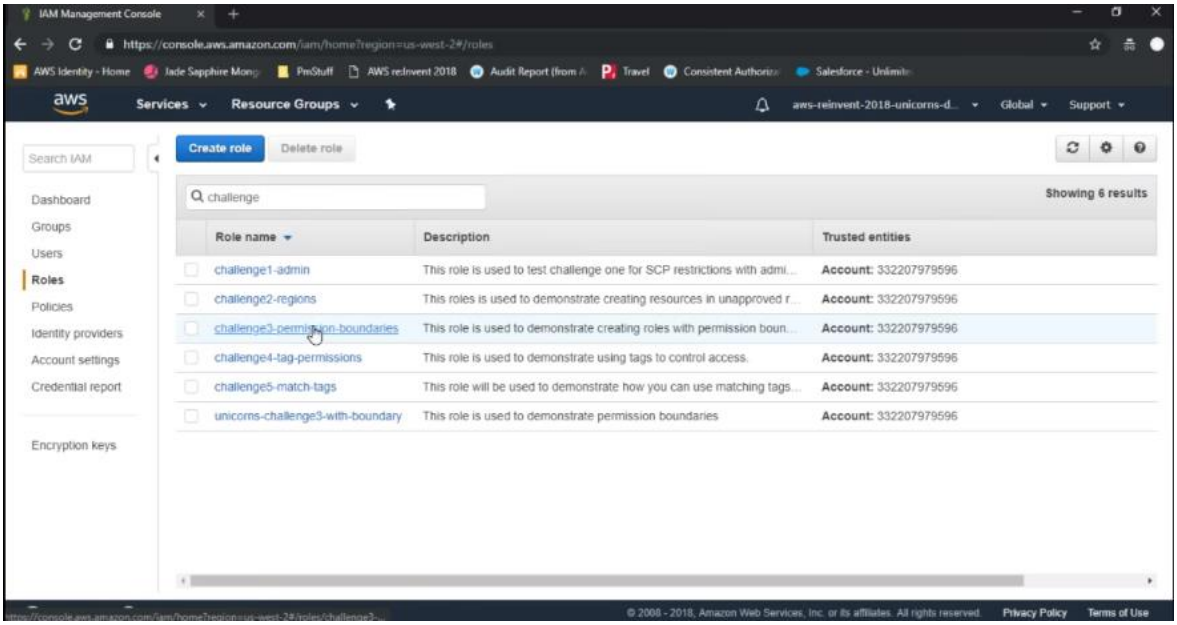
The **StringEquals** condition uses a pointer to a managed policy that provides the region restriction to only the 2 regions allowed as the permission boundary.

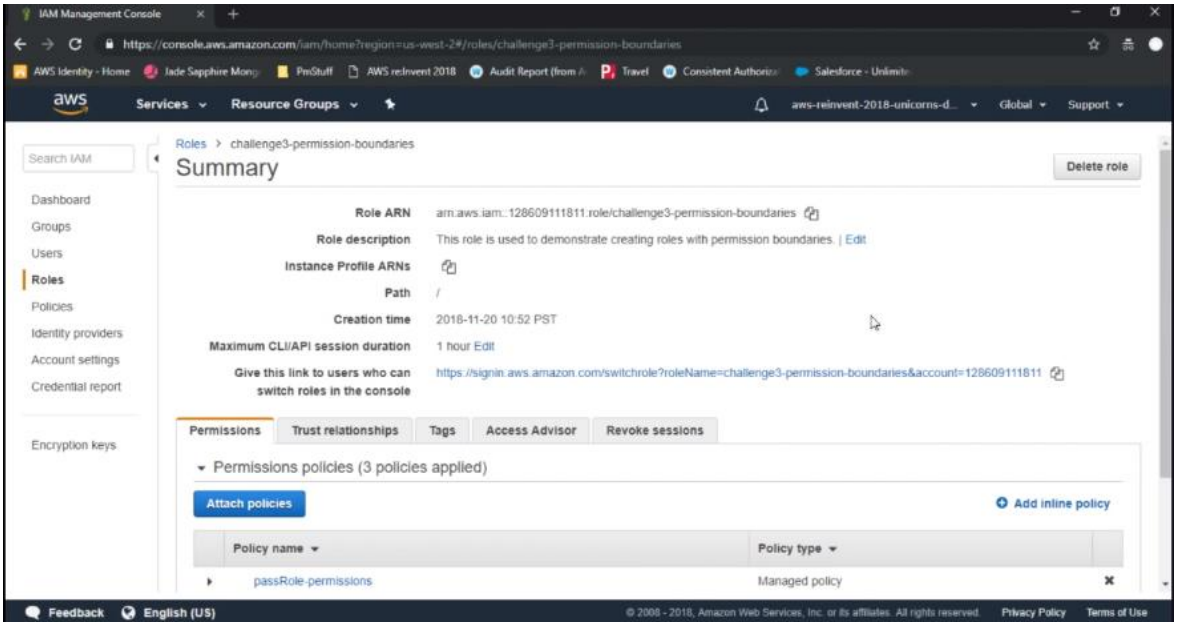
Permission boundary workflows

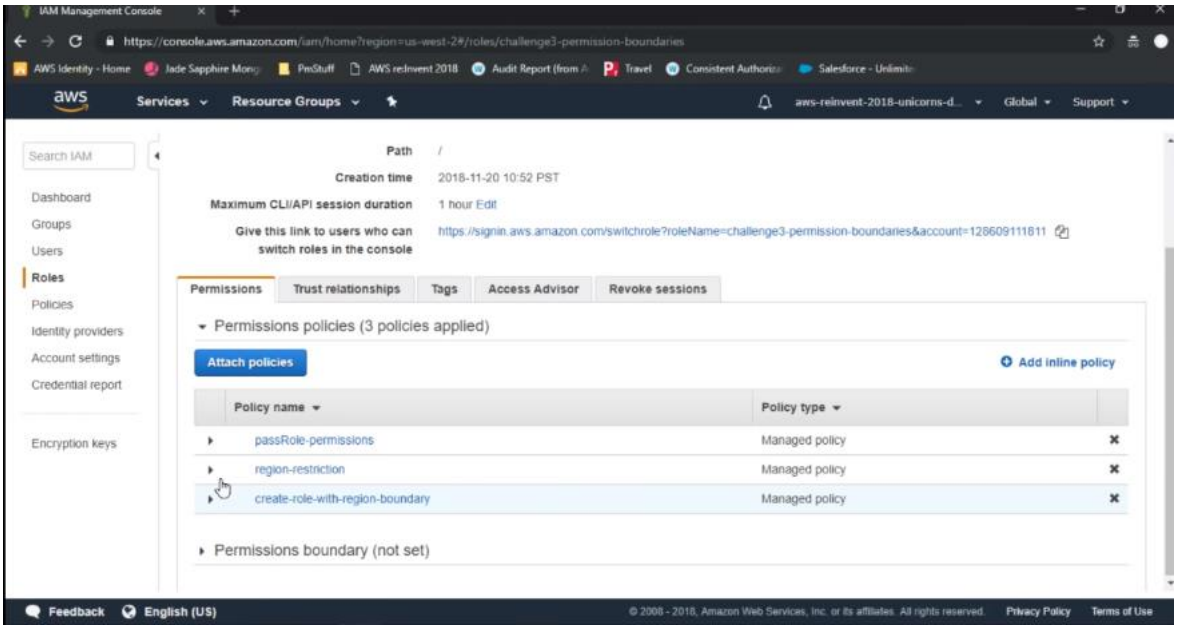


Let's see permission boundaries in action

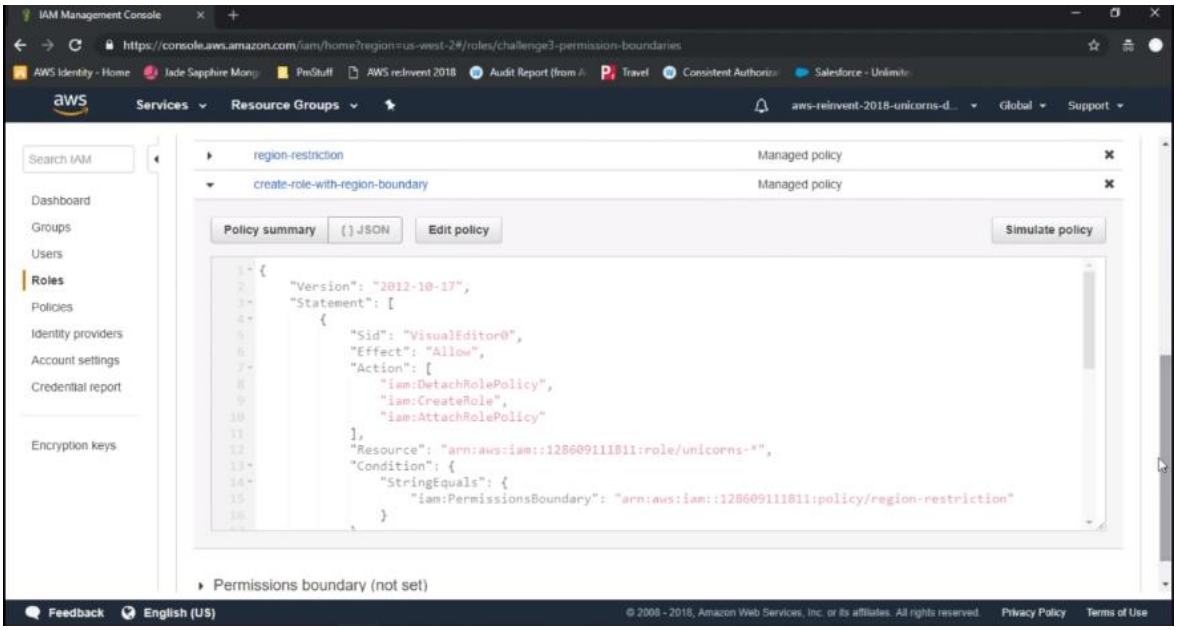
- Using the developer role, create a role with a permission boundary
- Use a role with a permission boundary to put data in S3 for approved regions and for unapproved regions.

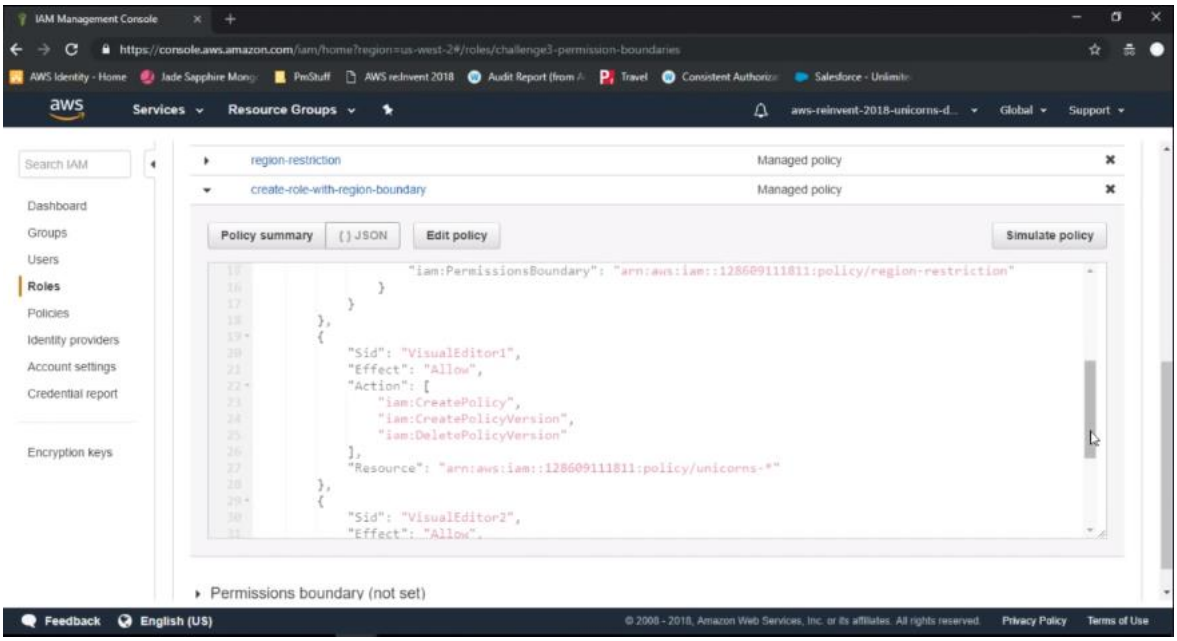


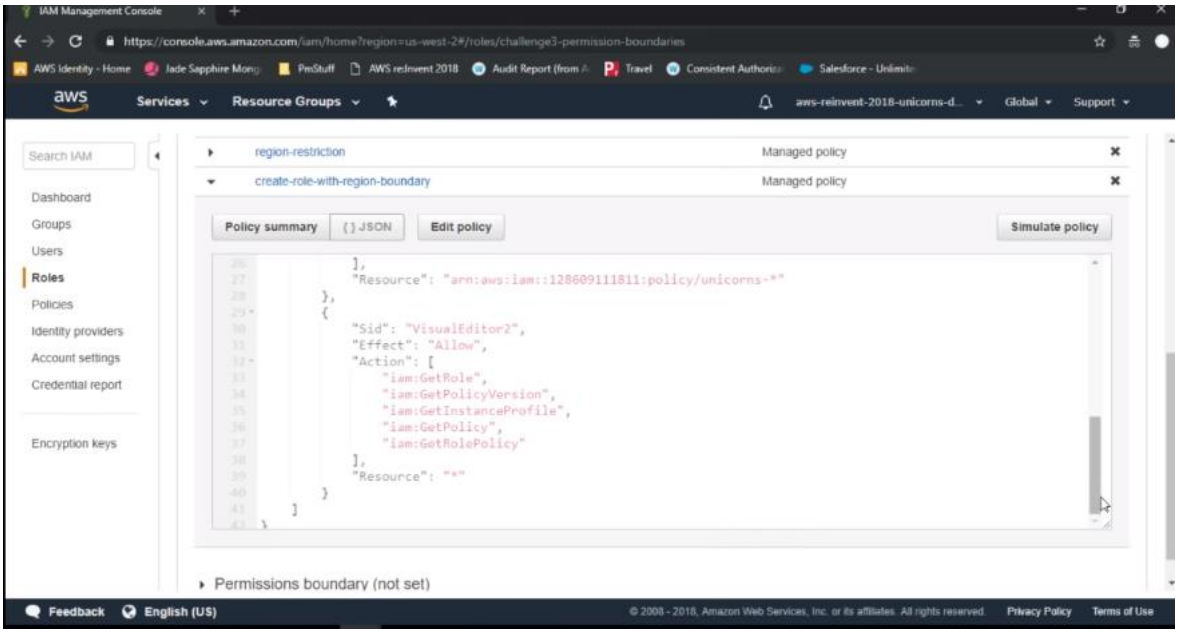


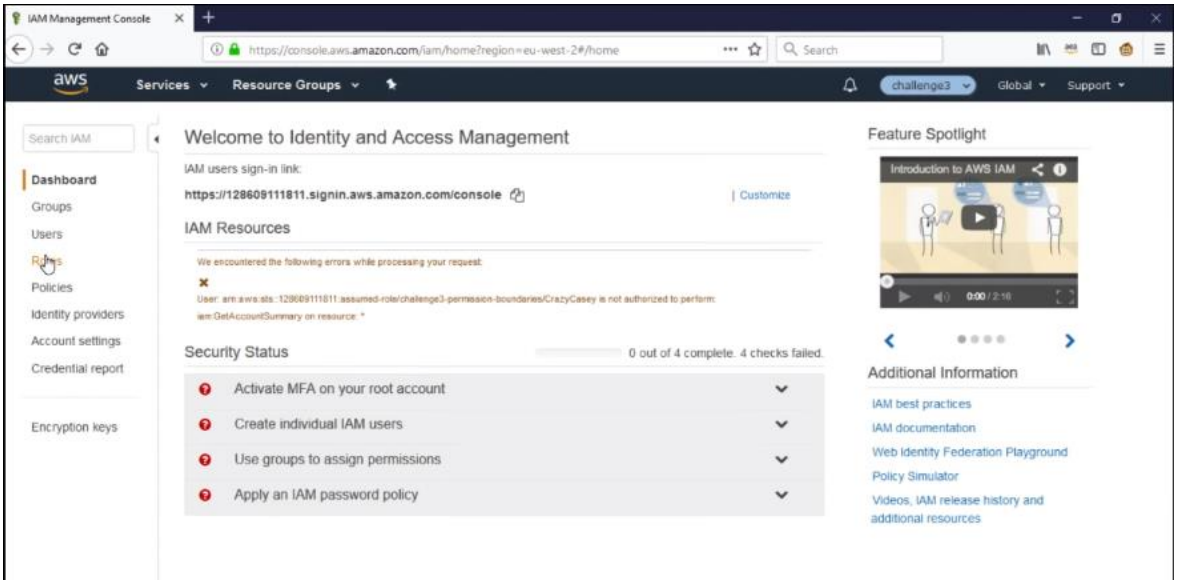


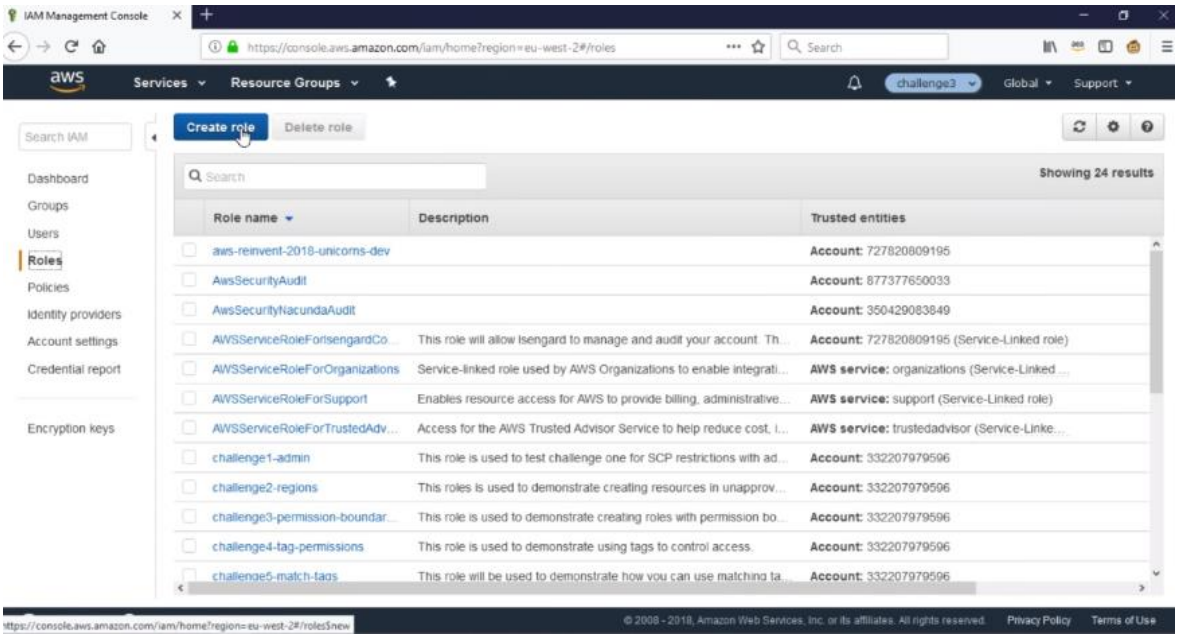
We already had the **passRole** policy and the region restriction policy **region-restriction** that restricts the developer to be able to create resources in only 2 regions. Then we just added the new policy **create-role-with-region-boundary** that adds the boundary to what they can actually do.

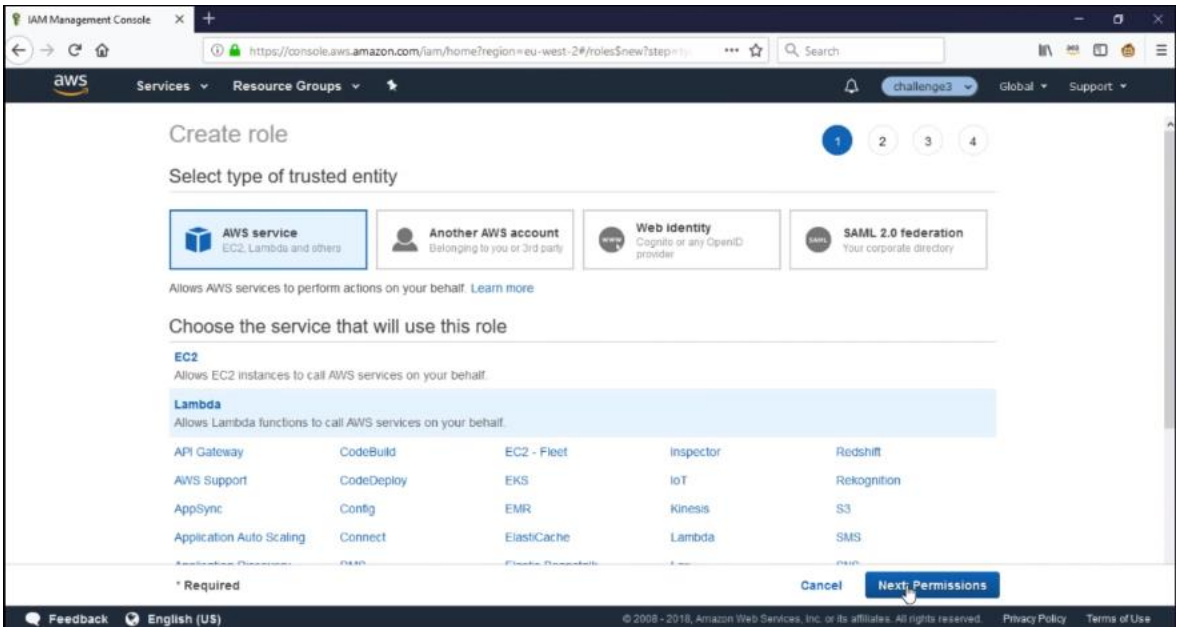


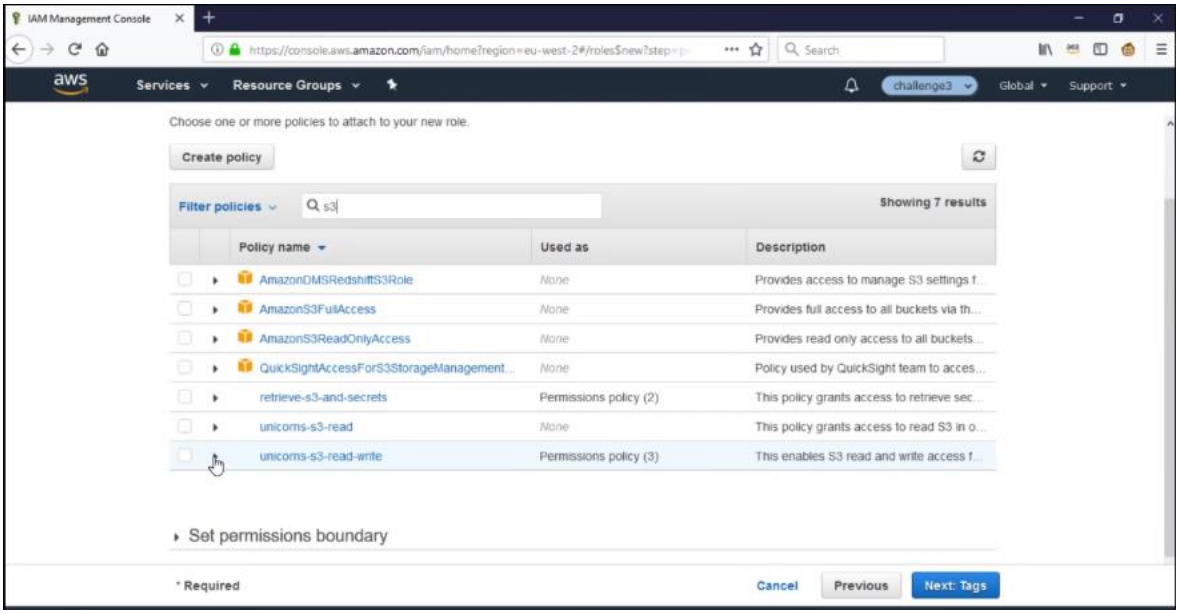

IAM Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge3-permission-boundaries
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
region-restriction
Managed policy
create-role-with-region-boundary
Managed policy
Policy summary
{} JSON
Edit policy
Simulate policy
"iam:PermissionsBoundary": "arn:aws:iam::128609111811:policy/region-restriction"
"Sid": "VisualEditor1",
"Effect": "Allow",
"Action": [
"iam:CreatePolicy",
"iam:CreatePolicyVersion",
"iam:DeletePolicyVersion"
"Resource": "arn:aws:iam::128609111811:policy/unicorns-*"
"Sid": "VisualEditor2",
"Effect": "Allow",
Permissions boundary (not set)
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

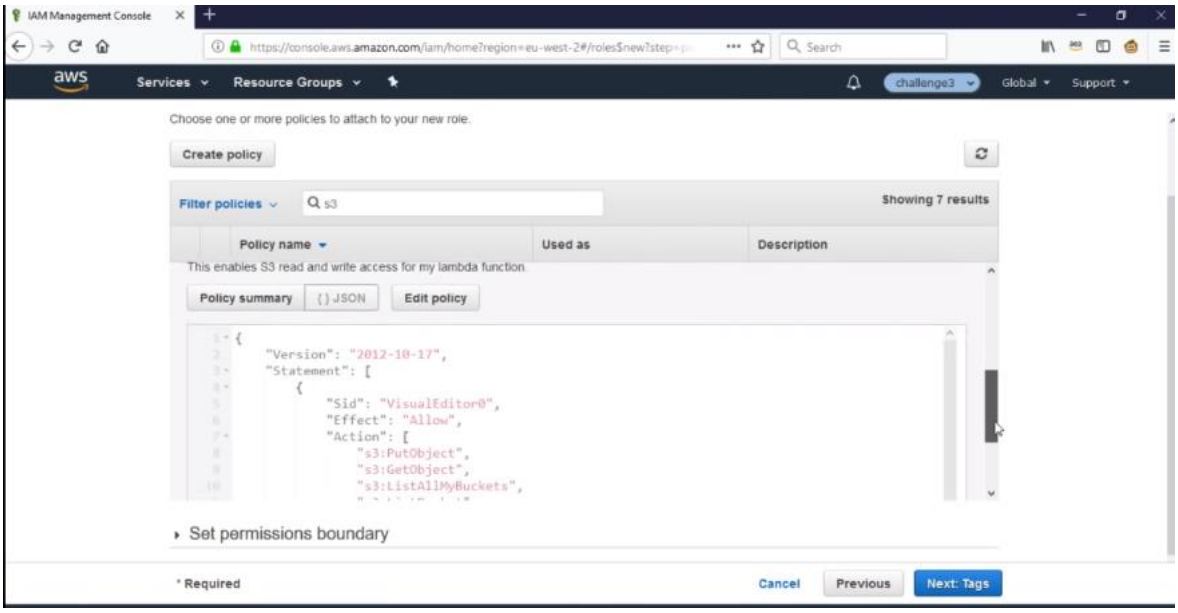

IAM Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge3-permission-boundaries
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
region-restriction
Managed policy
create-role-with-region-boundary
Managed policy
Policy summary
{} JSON
Edit policy
Simulate policy
"Resource": "arn:aws:iam::128609111811:policy/unicorns-*"
"Sid": "VisualEditor2",
"Effect": "Allow",
"Action": [
"iam:GetRole",
"iam:GetPolicyVersion",
"iam:GetInstanceProfile",
"iam:GetPolicy",
"iam:GotRolePolicy"
"Resource": "*"
Permissions boundary (not set)
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


IAM Management Console
https://console.aws.amazon.com/iam/home?region=eu-west-2#/home
Search
Services
Resource Groups
challenge3
Global
Support
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Welcome to Identity and Access Management
IAM users sign-in link:
https://128609111811.signin.aws.amazon.com/console
Customize
IAM Resources
We encountered the following errors while processing your request:
User: arn:aws:sts::128609111811:assumed-role/challenge3-permission-boundaries/CrazyCasey is not authorized to perform:
iam:GetAccountSummary on resource: *
Security Status
0 out of 4 complete. 4 checks failed.
Activate MFA on your root account
Create individual IAM users
Use groups to assign permissions
Apply an IAM password policy
Feature Spotlight
Introduction to AWS IAM
Additional Information
IAM best practices
IAM documentation
Web Identity Federation Playground
Policy Simulator
Videos, IAM release history and additional resources

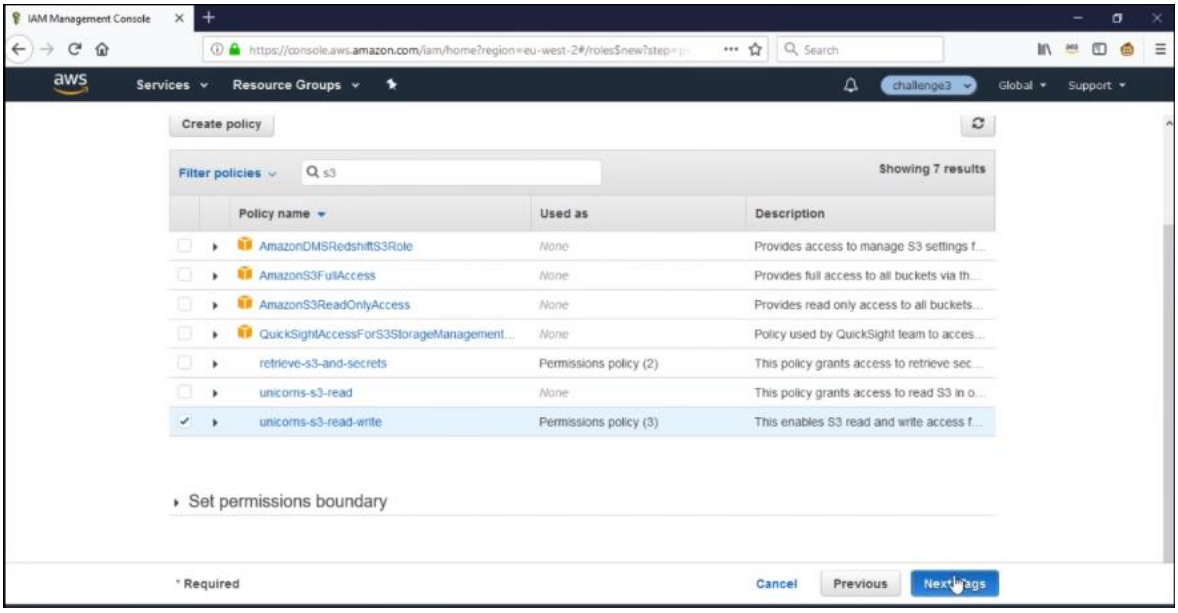

IAM Management Console
https://console.aws.amazon.com/iam/home?region=eu-west-2#/roles
Search
aws
Services
Resource Groups
challenge3
Global
Support
Search IAM
Create role
Delete role
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Search
Showing 24 results
Role name
Description
Trusted entities
aws-reinvent-2018-unicorns-dev
Account: 727820809195
AwsSecurityAudit
Account: 877377650033
AwsSecurityNacundaAudit
Account: 350429083849
AWSServiceRoleForIsengardCo...
This role will allow Isengard to manage and audit your account. Th...
Account: 727820809195 (Service-Linked role)
AWSServiceRoleForOrganizations
Service-linked role used by AWS Organizations to enable integrati...
AWS service: organizations (Service-Linked ...
AWSServiceRoleForSupport
Enables resource access for AWS to provide billing, administrative...
AWS service: support (Service-Linked role)
AWSServiceRoleForTrustedAdv...
Access for the AWS Trusted Advisor Service to help reduce cost, i...
AWS service: trustedadvisor (Service-Linke...
challenge1-admin
This role is used to test challenge one for SCP restrictions with ad...
Account: 332207979596
challenge2-regions
This roles is used to demonstrate creating resources in unapprov...
Account: 332207979596
challenge3-permission-boundar...
This role is used to demonstrate creating roles with permission bo...
Account: 332207979596
challenge4-tag-permissions
This role is used to demonstrate using tags to control access.
Account: 332207979596
challenge5-match-tags
This role will be used to demonstrate how you can use matching ta...
Account: 332207979596
https://console.aws.amazon.com/iam/home?region=eu-west-2#/roles$new
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

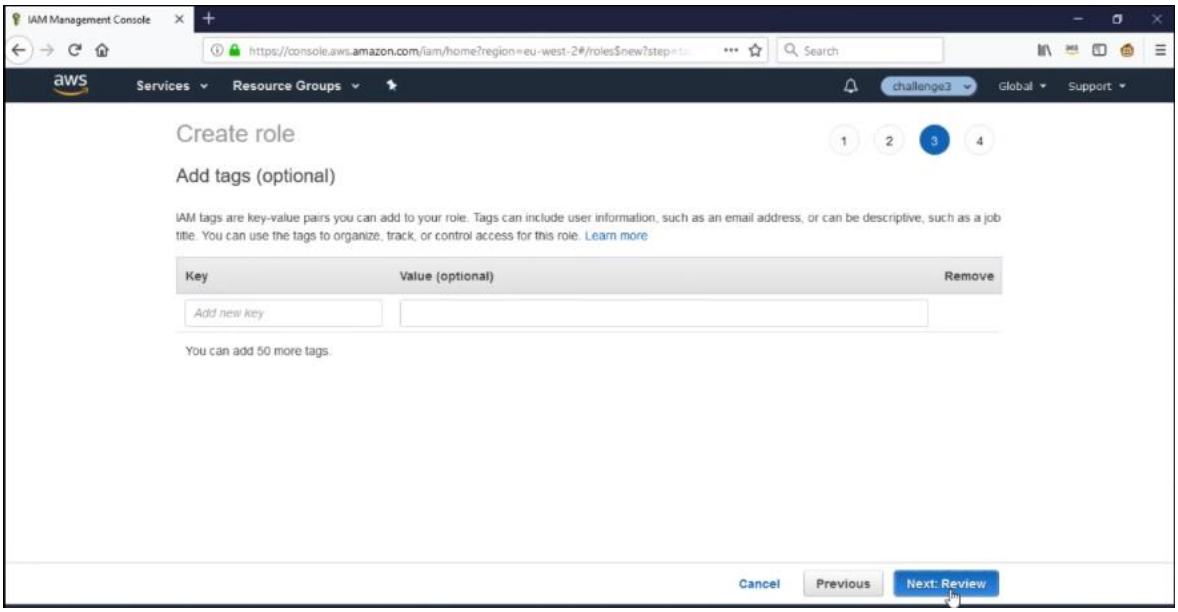

IAM Management Console
https://console.aws.amazon.com/iam/home?region=eu-west-2#/roles$new?step=
Search
aws
Services
Resource Groups
challenge3
Global
Support
Create role
1
2
3
4
Select type of trusted entity
AWS service
EC2, Lambda and others
Another AWS account
Belonging to you or 3rd party
Web identity
Cognito or any OpenID provider
SAML 2.0 federation
Your corporate directory
Allows AWS services to perform actions on your behalf. Learn more
Choose the service that will use this role
EC2
Allows EC2 instances to call AWS services on your behalf.
Lambda
Allows Lambda functions to call AWS services on your behalf.
API Gateway
CodeBuild
EC2 - Fleet
Inspector
Redshift
AWS Support
CodeDeploy
EKS
IoT
Rekognition
AppSync
Config
EMR
Kinesis
S3
Application Auto Scaling
Connect
ElastiCache
Lambda
SMS
* Required
Cancel
Next: Permissions
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

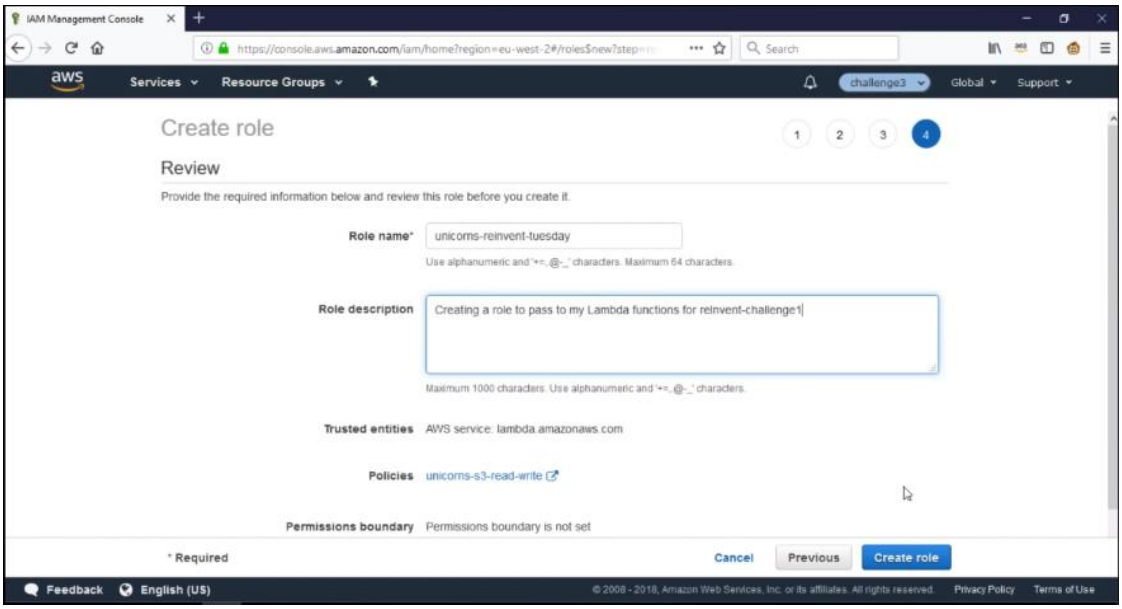

IAM Management Console
https://console.aws.amazon.com/iam/home?region=eu-west-2#/roles$new?step=p
Search
aws
Services
Resource Groups
challenge3
Global
Support
Choose one or more policies to attach to your new role.
Create policy
Filter policies
s3
Showing 7 results
Policy name
Used as
Description
AmazonDMSRedshiftS3Role
None
Provides access to manage S3 settings f...
AmazonS3FullAccess
None
Provides full access to all buckets via th...
AmazonS3ReadOnlyAccess
None
Provides read only access to all buckets...
QuickSightAccessForS3StorageManagement...
None
Policy used by QuickSight team to acces...
retrieve-s3-and-secrets
Permissions policy (2)
This policy grants access to retrieve sec...
unicorns-s3-read
None
This policy grants access to read S3 in o...
unicorns-s3-read-write
Permissions policy (3)
This enables S3 read and write access f...
Set permissions boundary
* Required
Cancel
Previous
Next: Tags

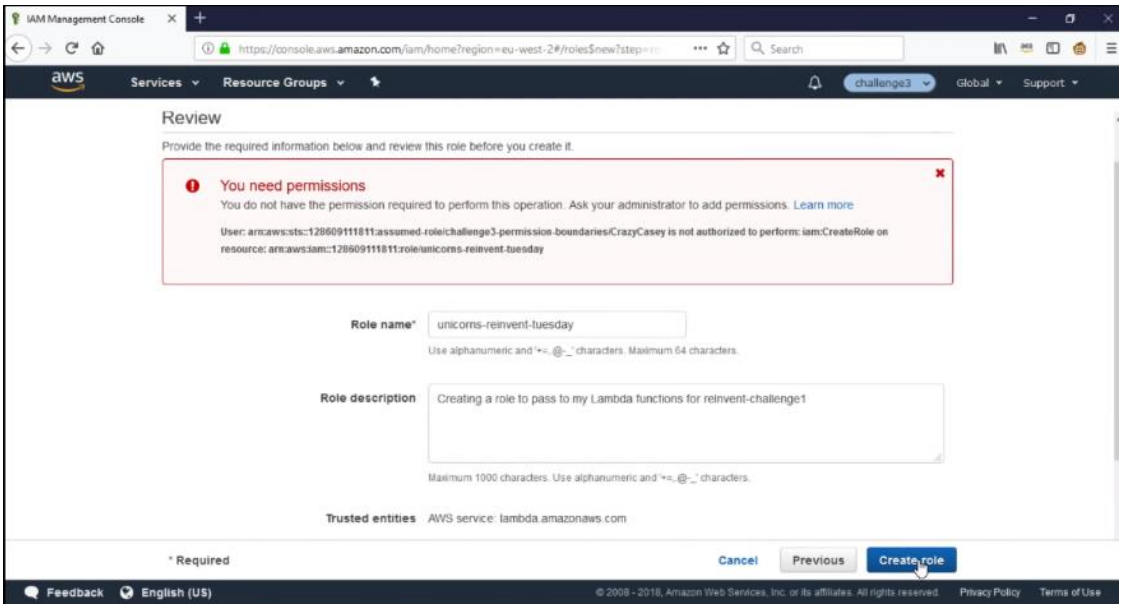


This is a very general S3 policy that has no region restriction on it.

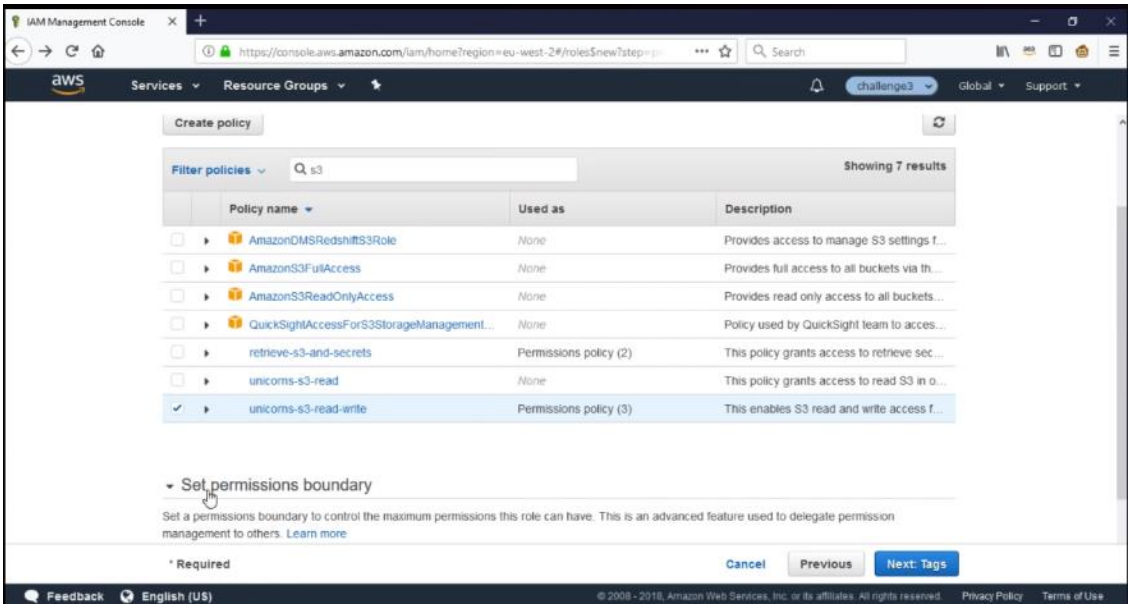


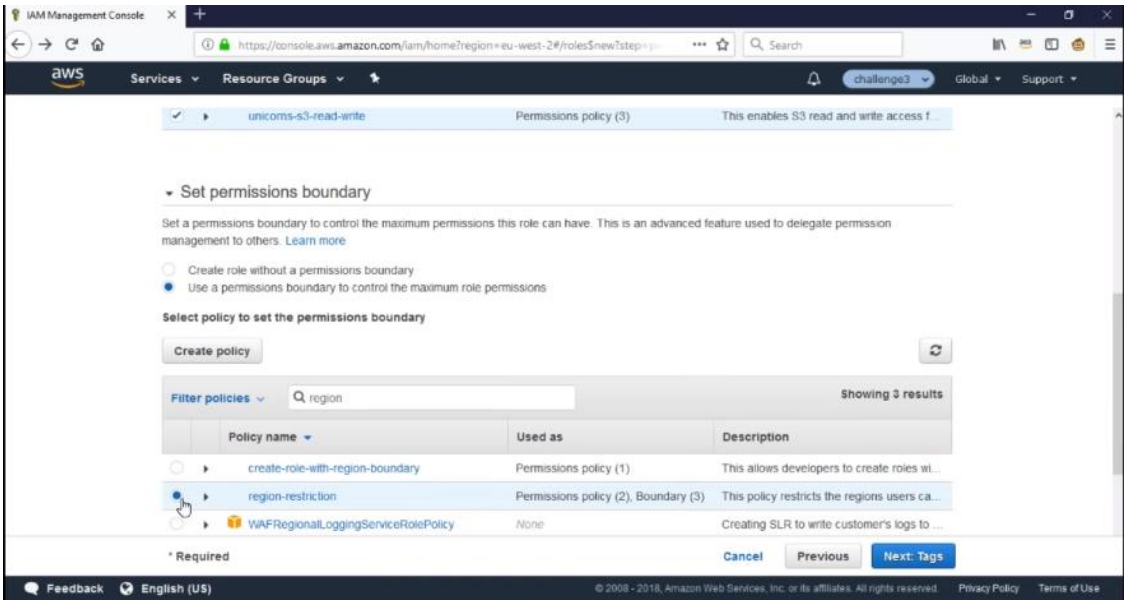


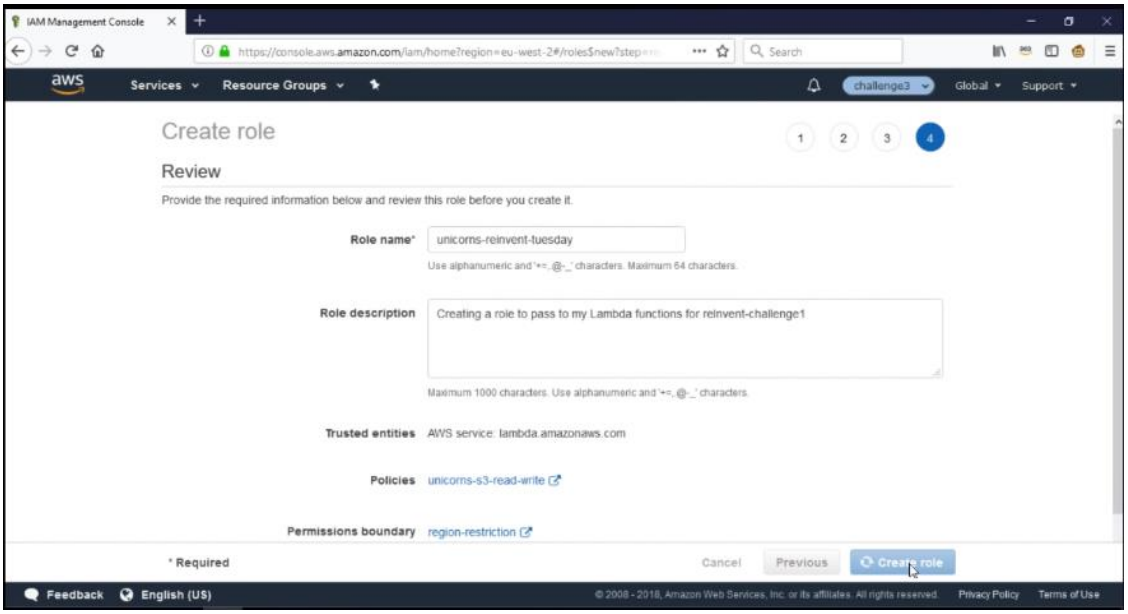


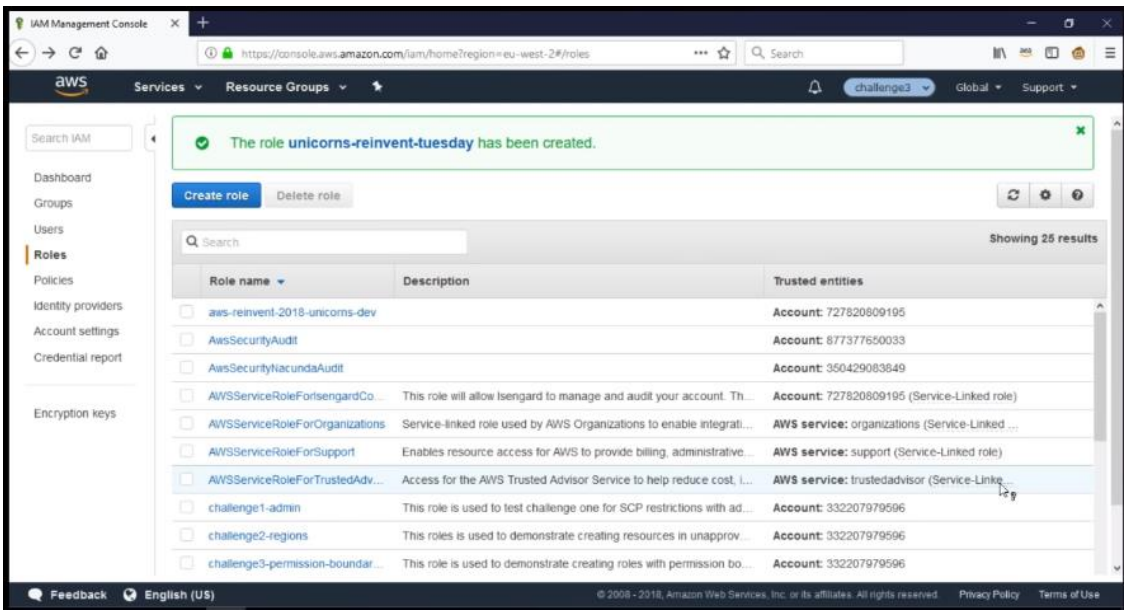


The create role is denied because the developer does not have access to create roles in this region

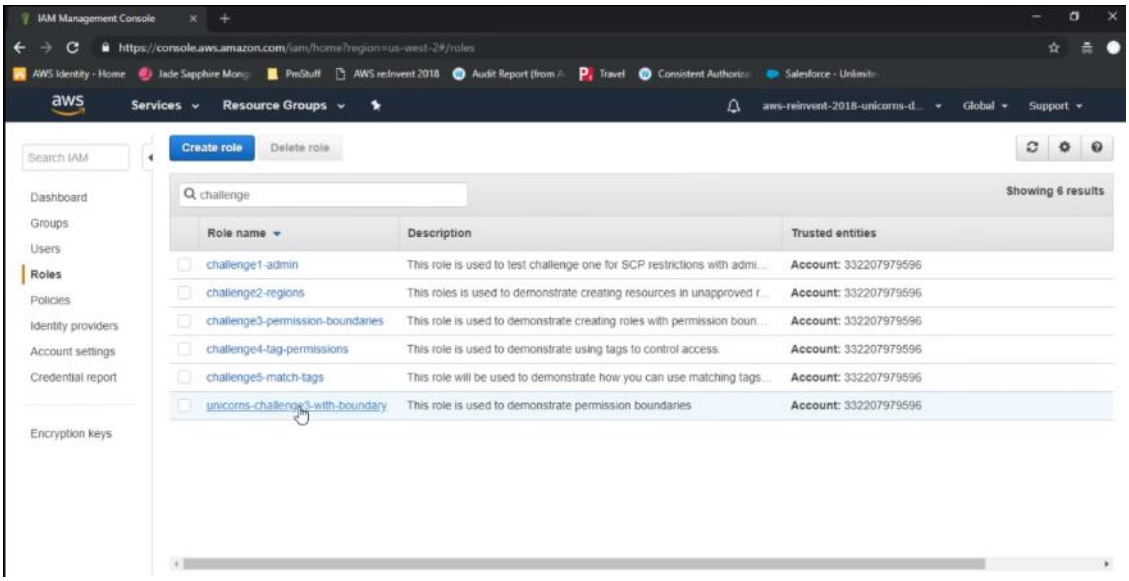


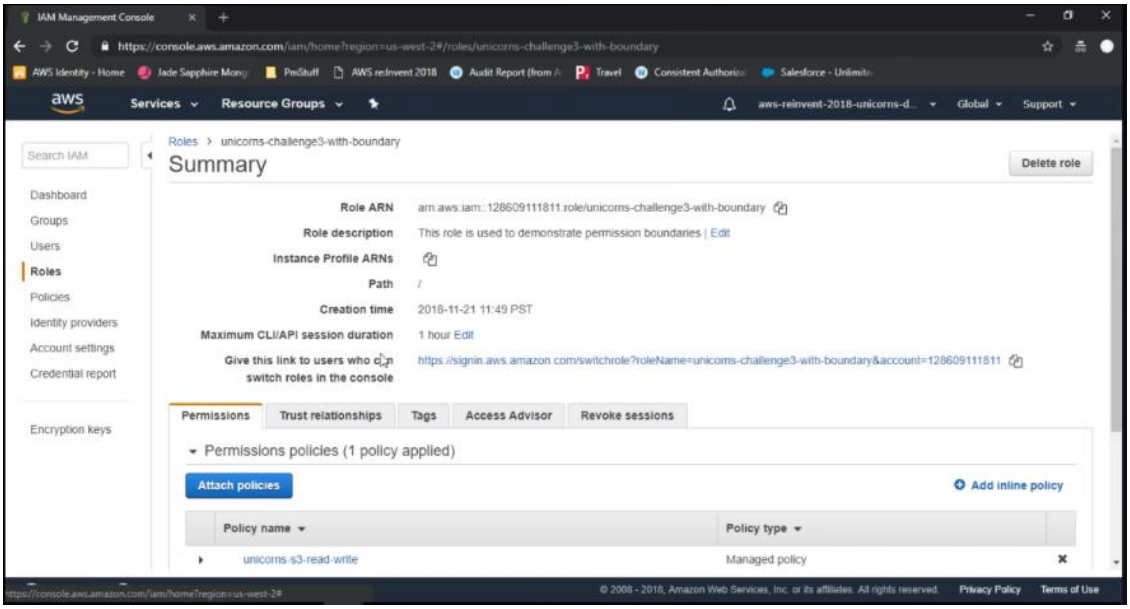


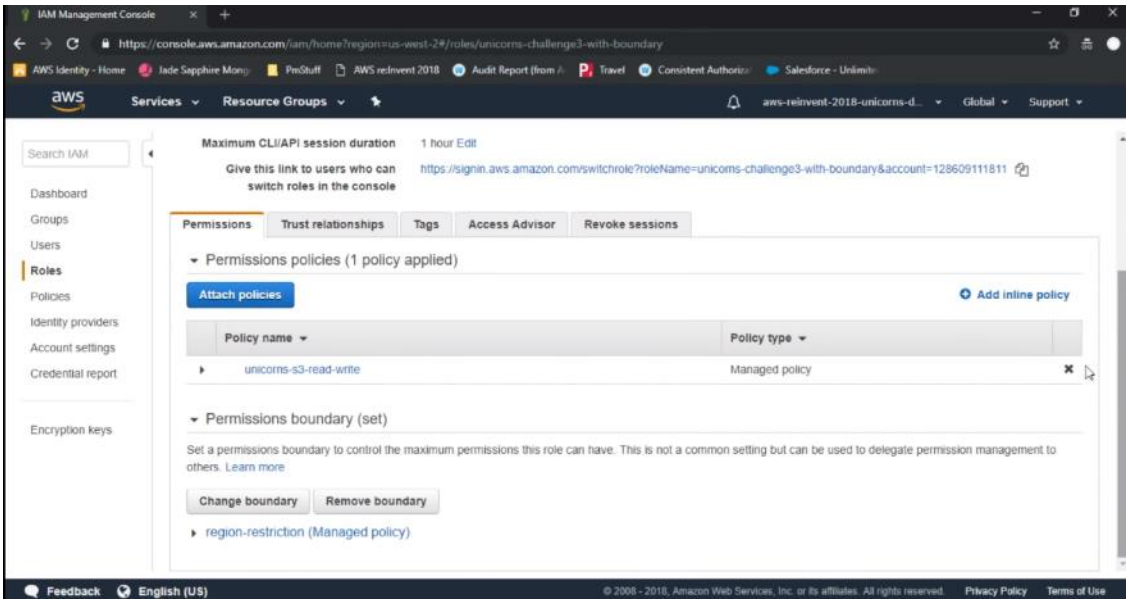




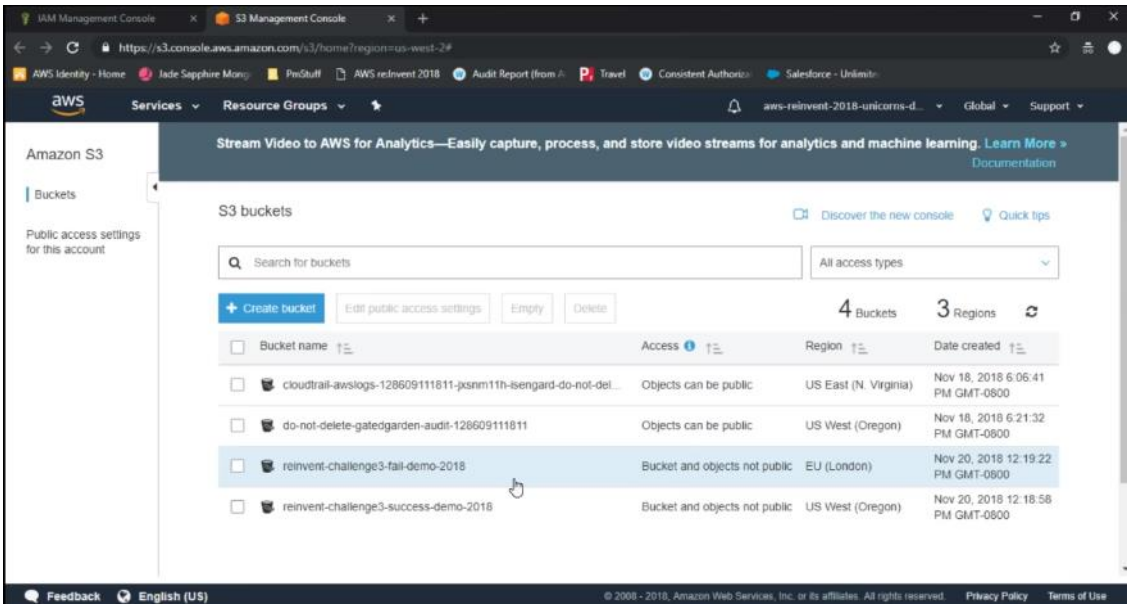
Now when we use the correct permission boundary SCP, the developer can now create the role successfully







We are using the region-restriction policy and the general s3-read-write policy.



We also have 2 S3 buckets for the fail and the success scenarios

```
C:\Users\brigidj\Desktop\reInvent2018>aws s3api put-object --bucket reinvent-challenge3-success-demo-2018 --key ./
pickles.jpg --body pickles.jpg --profile challenge2-regions
{
    "ETag": "\"d45a3ae9dc9ac57a5471281d4d4080d9\""
}
```

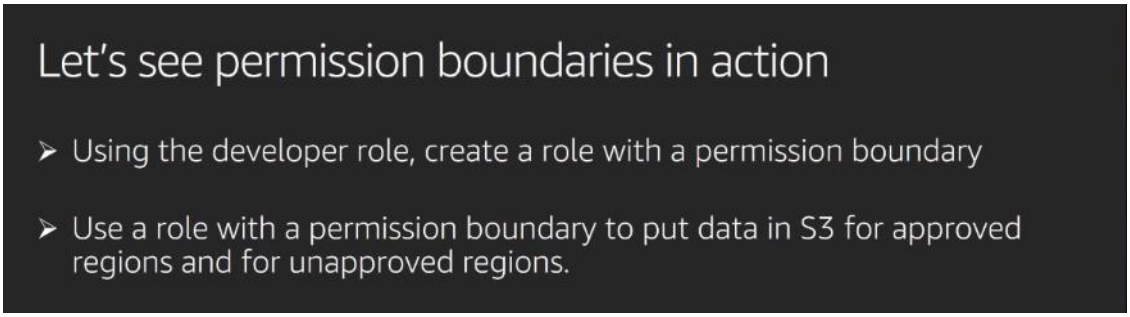
We will use the CLI to PUT object in the success object in the west region, it worked.

```
C:\Users\brigidj\Desktop\reInvent2018>aws s3api put-object --bucket reinvent-challenge3-fail-demo-2018 --key ./pic
kles.jpg --body pickles.jpg --profile challenge2-regions

An error occurred (AccessDenied) when calling the PutObject operation: Access Denied

C:\Users\brigidj\Desktop\reInvent2018>
```

When we try to PUT into the fail bucket in the London region, it fails because of the permission boundary is preventing us from calling outside of the US west region.



Challenge #4

Enable developers working on the Dorky project and the Sneaky project to manage their own resources without also managing the other project's.

Pro Tip: Carefully consider the tag keys you want to use for authorization

We are still inside the same account but the unicorn team is broken into 2 different teams for the Dorky and Sneaky projects that need to manage their own resources and not share resources. We are going to **use tag-based access control** in this case to scale our permissions management. You need to control your tags management.

Three parts required for tag-based access control

- Allow users to create tags when creating resources, but require specific tags when users create resources
  - RequestTag condition to require specific tag value during create actions
- Control which existing resources and values developers can tag
  - Use a combination of RequestTag and ResourceTag control access
- Control resources users can manage based on tag values
  - ResourceTag to control access to resources based on a tag that exists on a resource

Policy for challenge #4

```
"Effect": "Allow",
          "Action": [
              "ec2:RunInstances"
          ],
          "Resource": [
              "arn:aws:ec2:*:*:subnet/*",
              "arn:aws:ec2:*:*:key-pair/*",
              "arn:aws:ec2:*::snapshot/*",
              "arn:aws:ec2:*:*:launch-template/*",
              "arn:aws:ec2:*:*:volume/*",
              "arn:aws:ec2:*:*:security-group/*",
              "arn:aws:ec2:*:*:placement-group/*",
              "arn:aws:ec2:*:*:network-interface/*",
              "arn:aws:ec2:*::image/*"
          ],
          "Condition": {
              "StringEquals": {
                  "aws:RequestedRegion": ["us-west-1", "us-west-2" ]
              }
          }
```

Policy for challenge #4

Allow for creation of tags when creating new resources, but…

```
"Effect": "Allow",
"Action": "ec2:CreateTags",
"Resource": "*",
"Condition": {
        "StringEquals": {
                "ec2:CreateAction": "RunInstances"
        }
}
```

Allows creation of tags

But only during RunInstances calls

Policy for challenge #4

…require specific tags when users create new resources

```
"Effect": "Allow",
"Action": [
  "ec2:RunInstances"
],
"Resource": [
  "arn:aws:ec2:*:*:instance/*"],
"Condition": {
  "ForAllValues:StringEquals": {
      "aws:TagKeys": ["project","name"]
  },
  "StringEquals": {
      "aws:RequestTag/project": ["dorky"],
      "aws:RequestedRegion": ["us-west-1","us-west-2"]
  }}
```

Allows project and/or name, but nothing else

Requires project tag and must be this value

Requires instance to be in approved region

Policy for challenge #4

Control which existing resources and values developers can tag

```
"Effect": "Allow",
"Action": "ec2:CreateTags",
"Resource": "*",
"Condition": {
        "StringEquals": {
                "ec2:ResourceTag/project": ["dorky"]
        },
        "ForAllValues:StringEquals": {
                "aws:TagKeys": ["project","name"]
        },
        "StringEqualsIfExists": {
                "aws:RequestTag/project": ["dorky"]
        }
}
```

Only tag resources with these tags

Tag with either of these keys

For *project*, you specify only these values

Policy for challenge #4

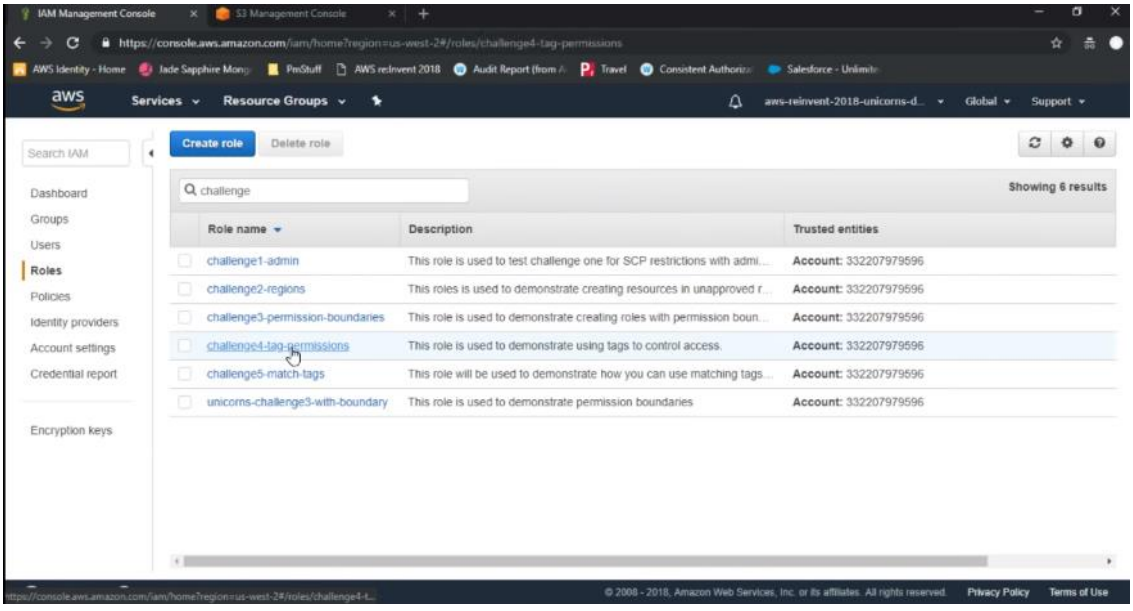
Control resources users can manage based on tag values

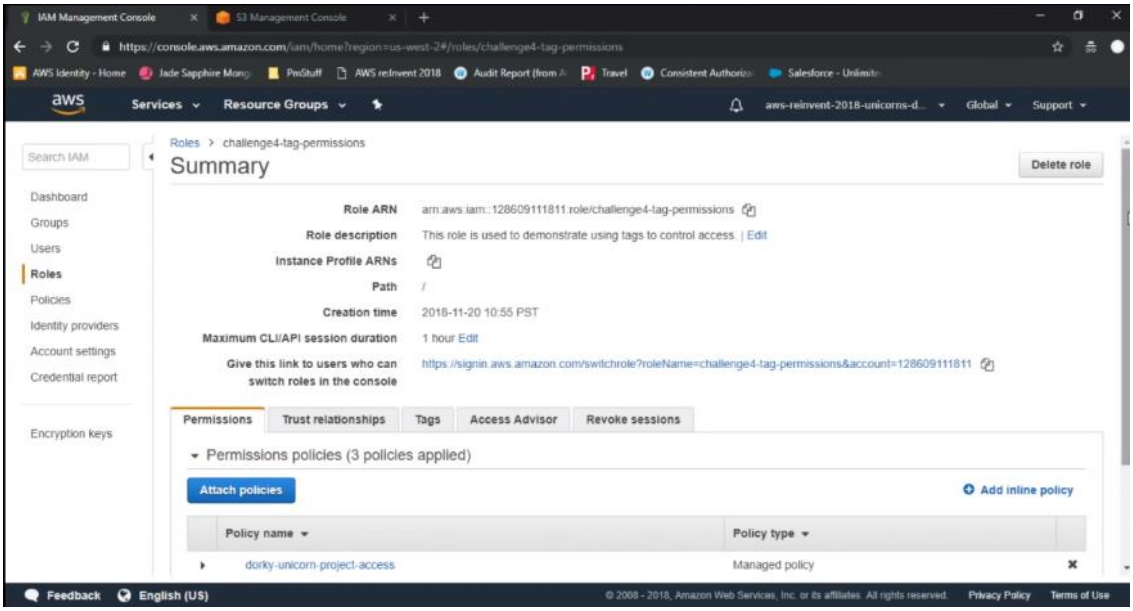
```
"Effect": "Allow",
"Action": [
       "ec2:StartInstances",
       "ec2:StopInstances"],
"Resource": "*",
"Condition": {
       "StringEquals": {
              "ec2:ResourceTag/project": "dorky"
       }
}
}
```

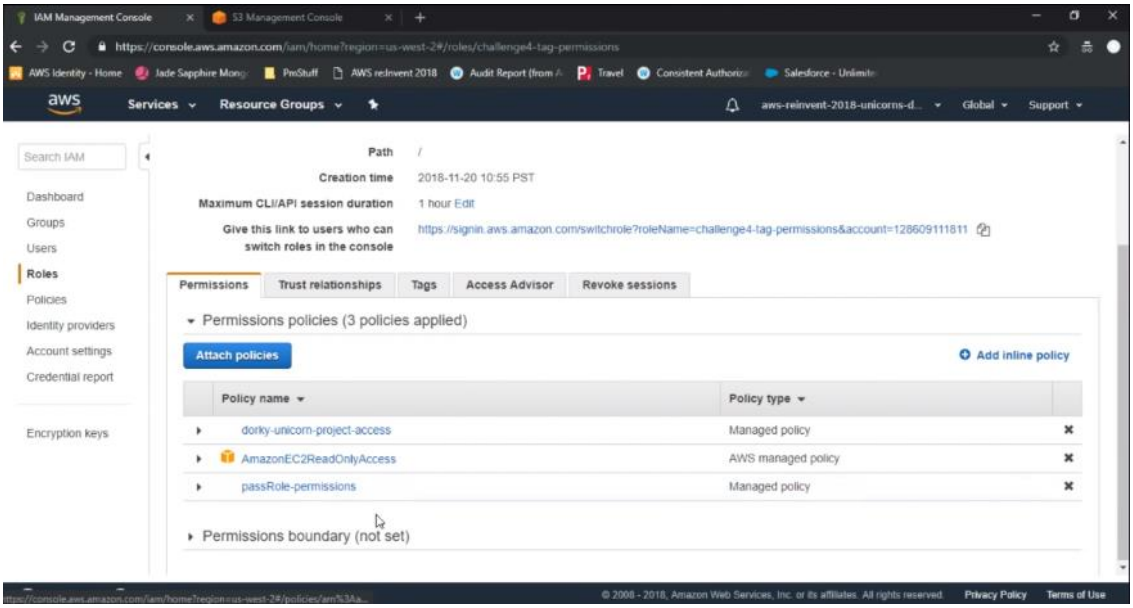
Only manage resources with these tags

Let's see tag-based access control in action

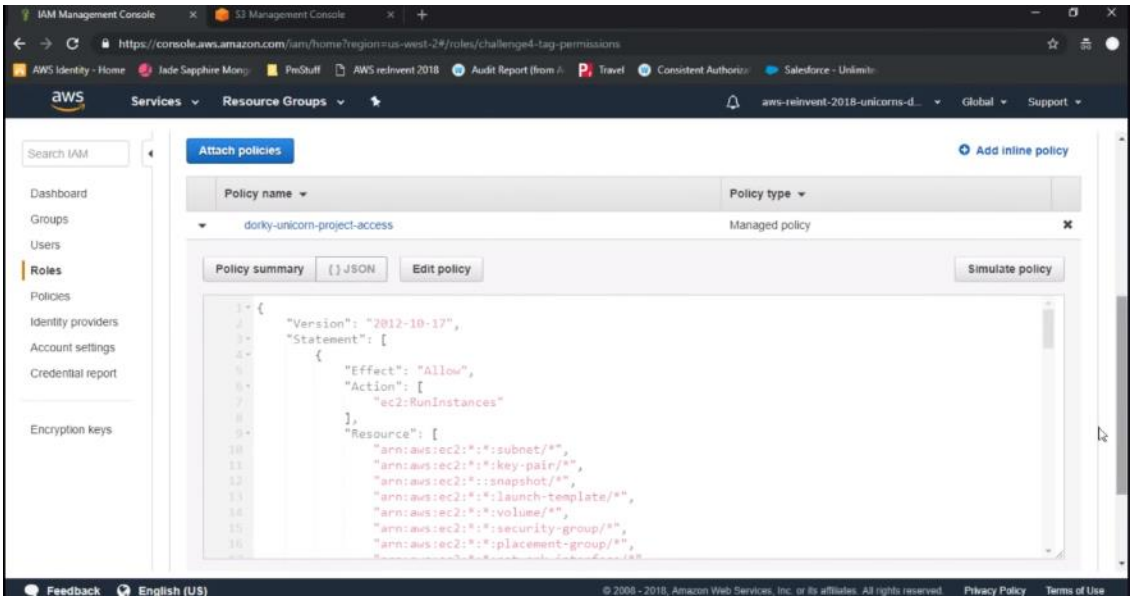
- Launch instances for project dorky
- Try to launch instances for project sneaky
- Modify tags on existing instances (dorky and sneaky)
- Manage existing instances

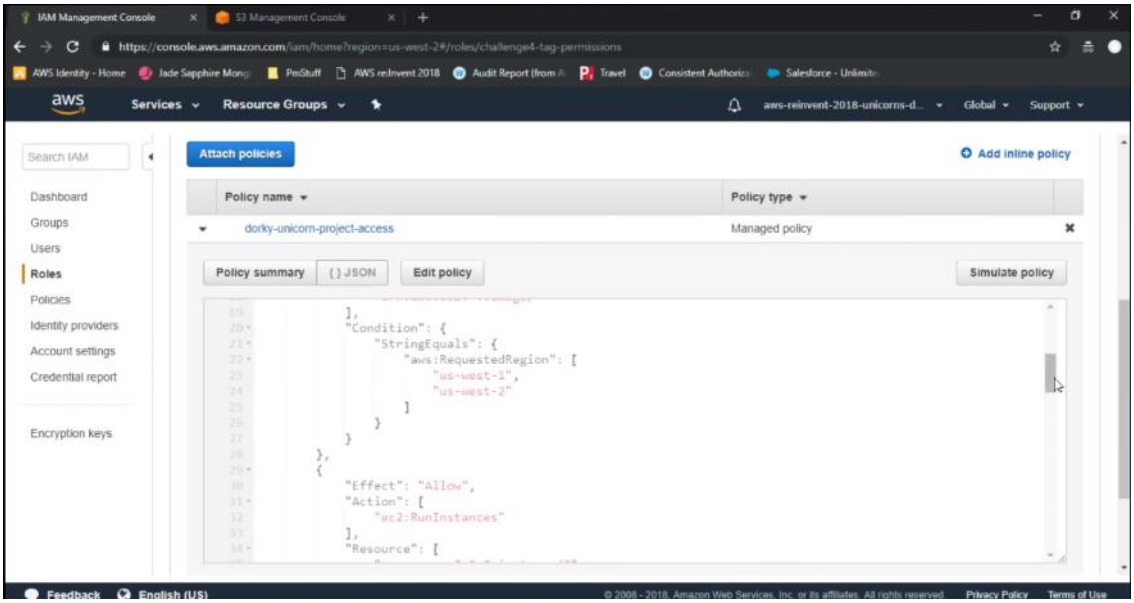


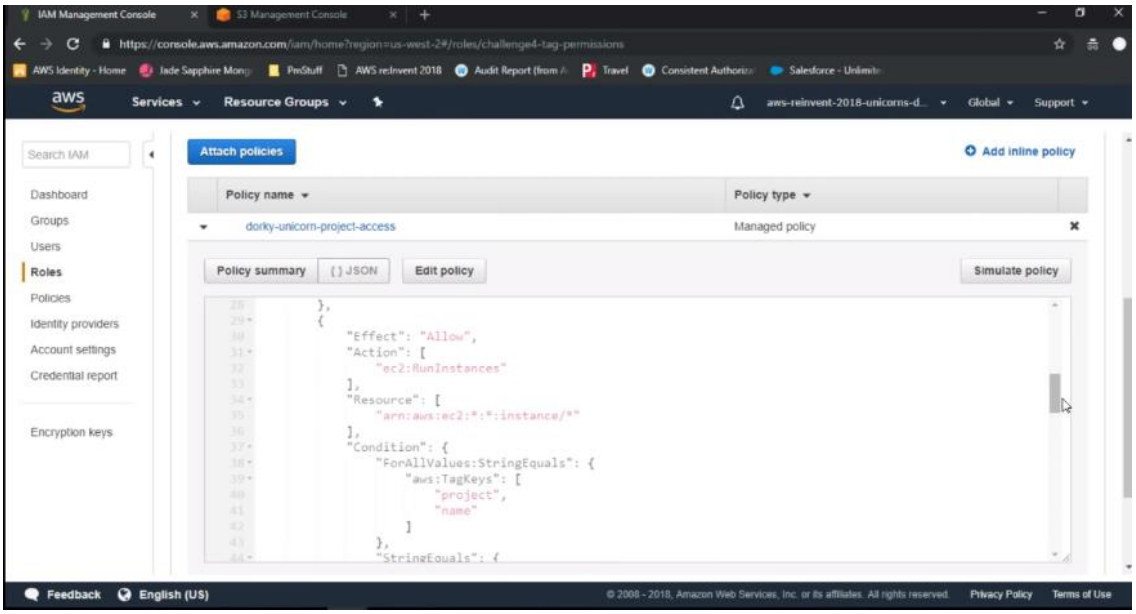


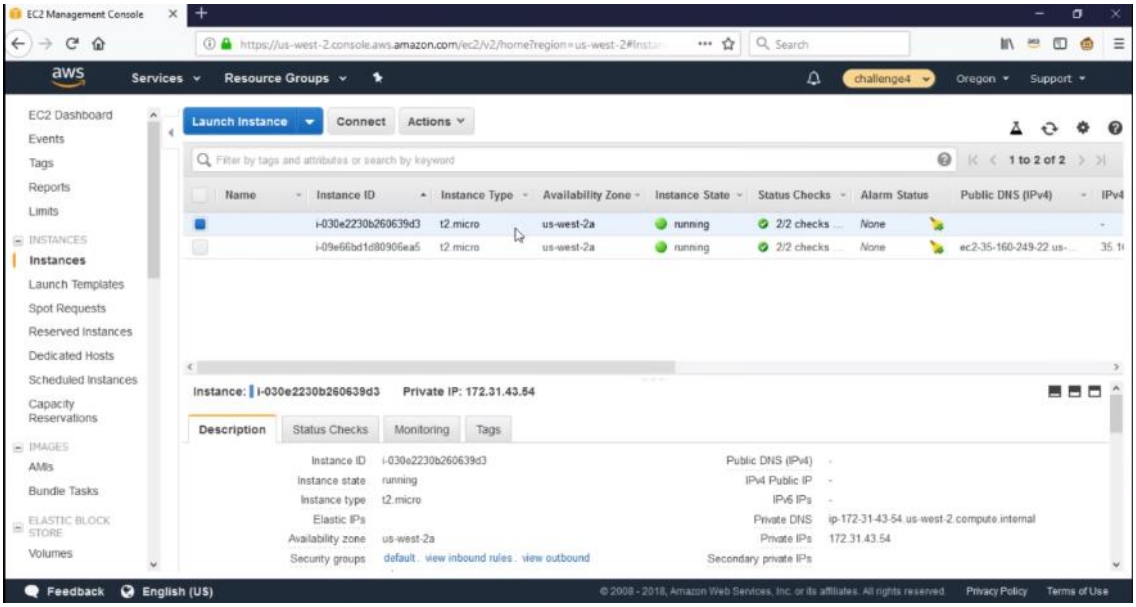


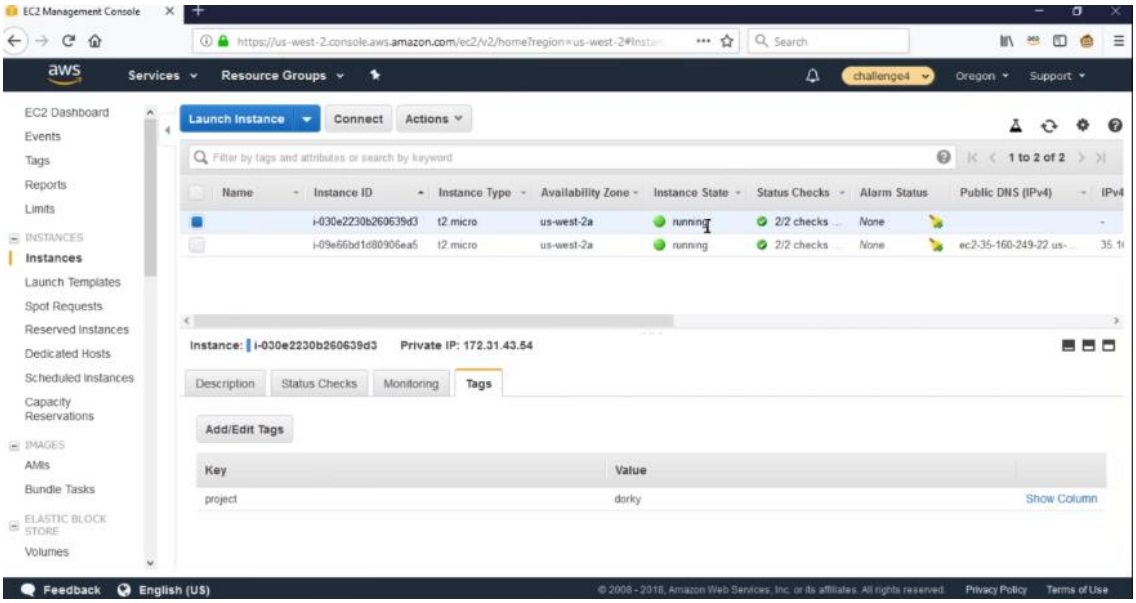
We now have the **dorky-unicorn-project-access** policy added to allow just the Dorky developers do things in their account only. We then also gave the general **AmazonEC2ReadOnlyAccess** policy that allows to do EC2 related things like list or runInstances from the EC2 console.

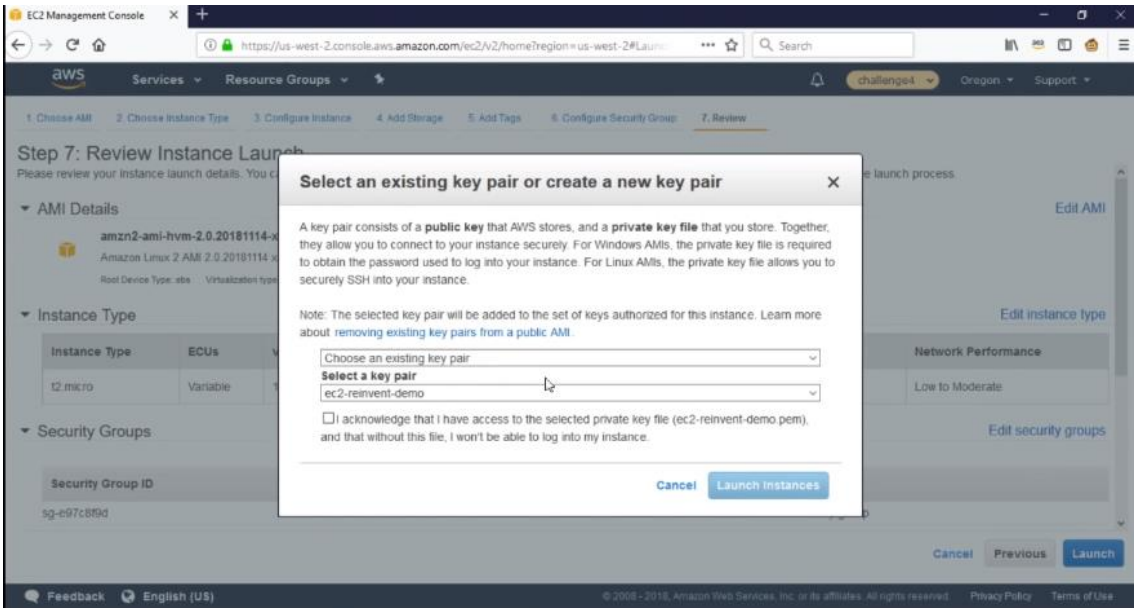

IAM Management Console
S3 Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge4-tag-permissions
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Attach policies
Add inline policy
Policy name
Policy type
dorky-unicorn-project-access
Managed policy
Policy summary
{ } JSON
Edit policy
Simulate policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"ec2:RunInstances"
],
"Resource": [
"arn:aws:ec2:*:*:subnet/*",
"arn:aws:ec2:*:*:key-pair/*",
"arn:aws:ec2:*::snapshot/*",
"arn:aws:ec2:*:*:launch-template/*",
"arn:aws:ec2:*:*:volume/*",
"arn:aws:ec2:*:*:security-group/*",
"arn:aws:ec2:*:*:placement-group/*",
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

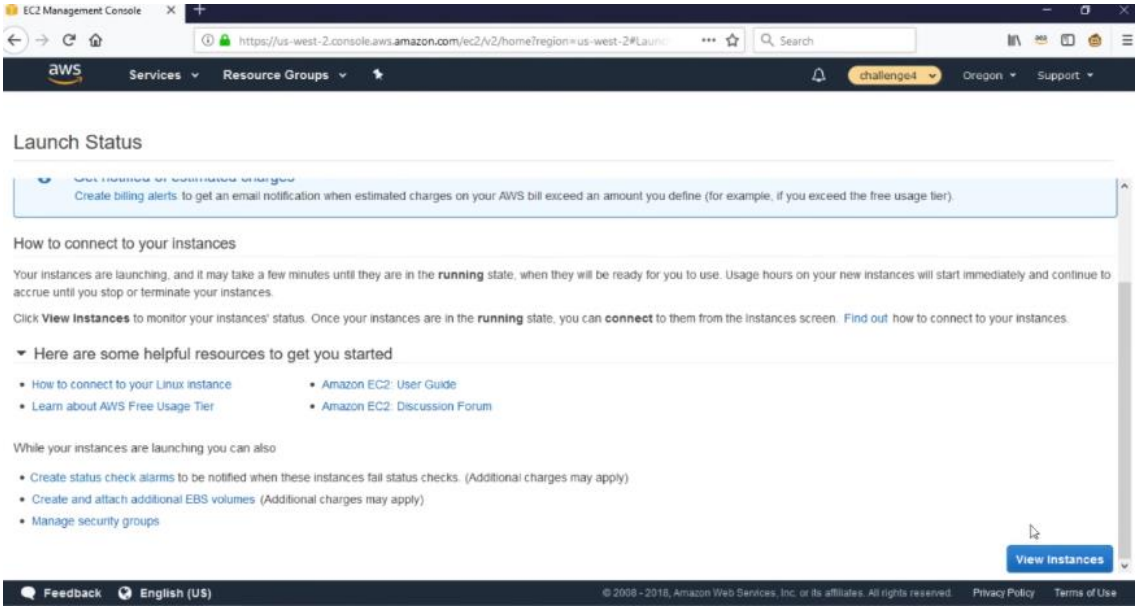

IAM Management Console
S3 Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge4-tag-permissions
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Attach policies
Add inline policy
Policy name
Policy type
dorky-unicorn-project-access
Managed policy
Policy summary
{ } JSON
Edit policy
Simulate policy
],
"Condition": {
"StringEquals": {
"aws:RequestedRegion": [
"us-west-1",
"us-west-2"
]
}
}
},
{
"Effect": "Allow",
"Action": [
"ec2:RunInstances"
],
"Resource": [
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

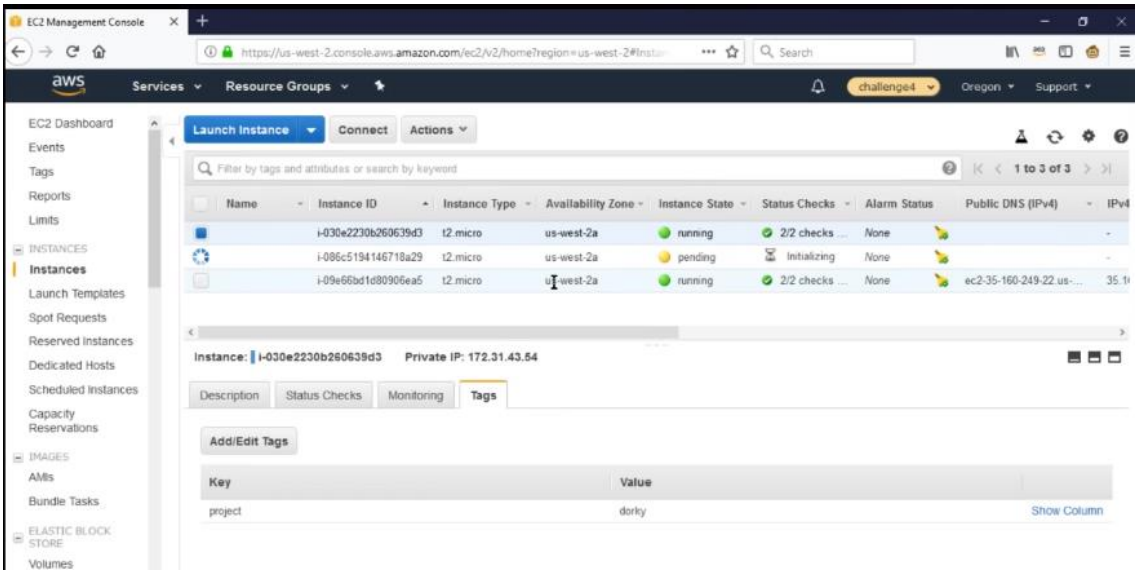

IAM Management Console
S3 Management Console
https://console.aws.amazon.com/iam/home?region=us-west-2#/roles/challenge4-tag-permissions
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Attach policies
Add inline policy
Policy name
Policy type
dorky-unicorn-project-access
Managed policy
Policy summary
{ } JSON
Edit policy
Simulate policy
},
{
"Effect": "Allow",
"Action": [
"ec2:RunInstances"
],
"Resource": [
"arn:aws:ec2:*:*:instance/*"
],
"Condition": {
"ForAllValues:StringEquals": {
"aws:TagKeys": [
"project",
"name"
]
},
"StringEquals": {
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
Services
Resource Groups
challenge4
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Launch Instance
Connect
Actions
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
i-030e2230b260639d3
t2.micro
us-west-2a
running
2/2 checks
None
i-09e66bd1d80906ea5
Instance: i-030e2230b260639d3 Private IP: 172.31.43.54
Description
Status Checks
Monitoring
Tags
Instance ID i-030e2230b260639d3
Instance state running
Instance type t2.micro
Elastic IPs
Availability zone us-west-2a
Security groups default. view inbound rules. view outbound
Public DNS (IPv4)
IPv4 Public IP
IPv6 IPs
Private DNS ip-172-31-43-54.us-west-2.compute.internal
Private IPs 172.31.43.54
Secondary private IPs
Feedback
English (US)
Privacy Policy
Terms of Use

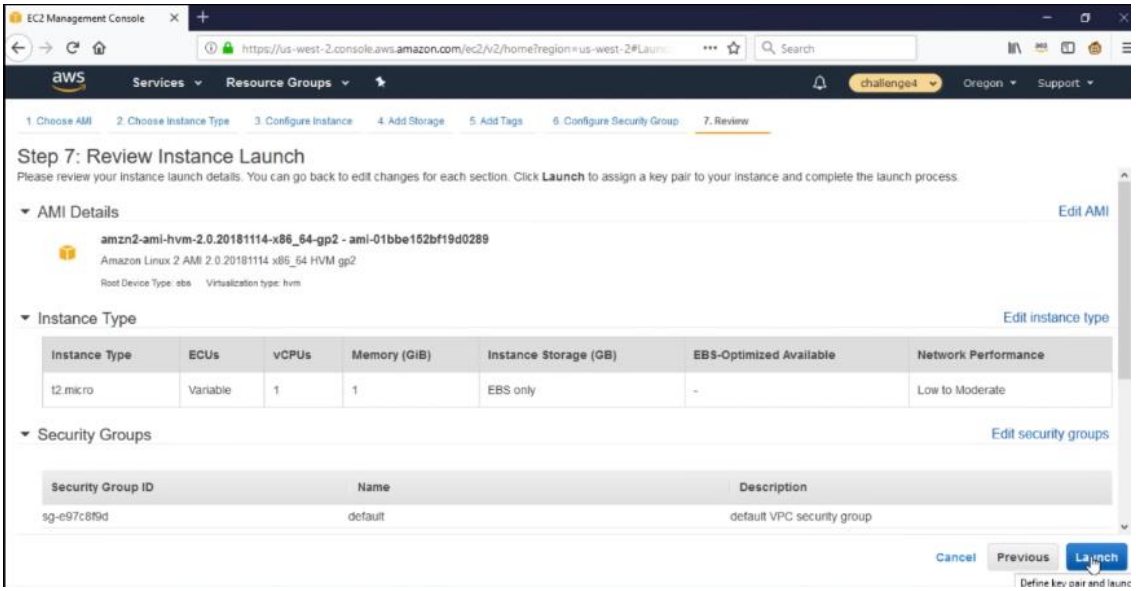

EC2 Management Console
Services
Resource Groups
challenge4
Oregon
Support
Launch Instance
Connect
Actions
Instance: i-030e2230b260639d3 Private IP: 172.31.43.54
Description
Status Checks
Monitoring
Tags
Add/Edit Tags
Key
Value
project
dorky
Show Column
Feedback
English (US)
Privacy Policy
Terms of Use

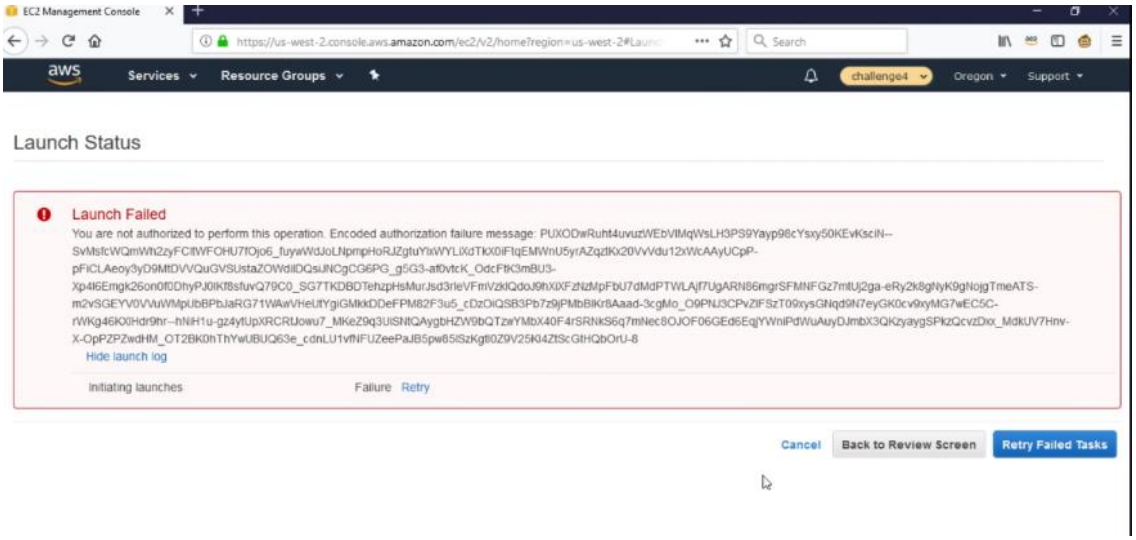

EC2 Management Console
Services
Resource Groups
challenge4
Oregon
Support
Step 7: Review Instance Launch
AMI Details
Edit AMI
Instance Type
Edit instance type
Security Groups
Edit security groups
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Choose an existing key pair
Select a key pair
ec2-reinvent-demo
I acknowledge that I have access to the selected private key file (ec2-reinvent-demo.pem), and that without this file, I won't be able to log into my instance.
Cancel
Launch Instances
Cancel
Previous
Launch
Feedback
English (US)
Privacy Policy
Terms of Use



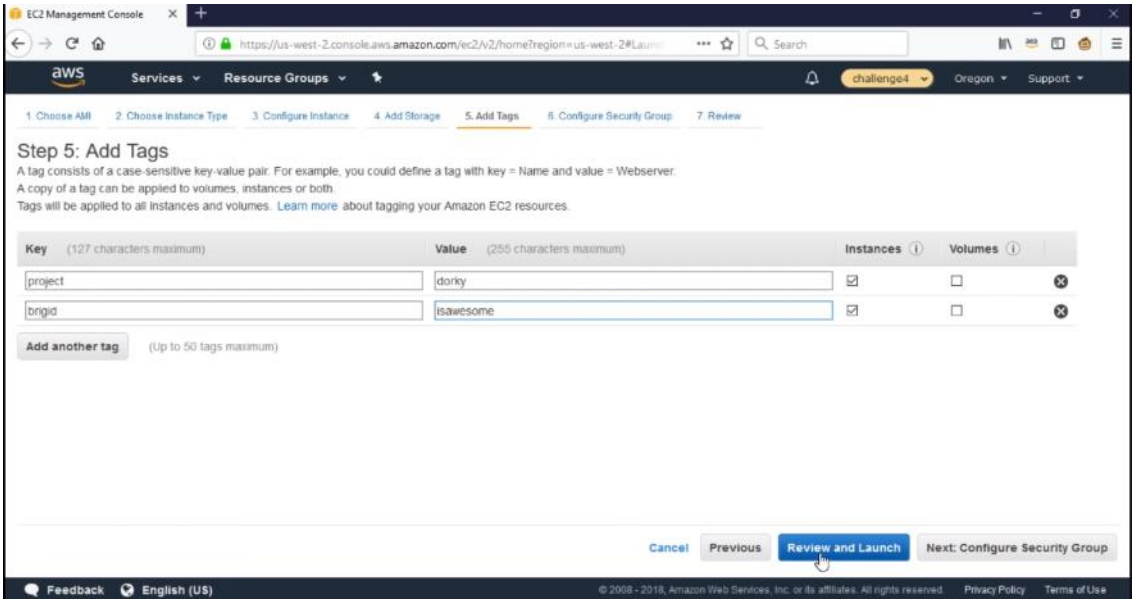


We are allowed to launch instances within the US-WEST region specified in the policy. We then try to launch another instance with the tag **sneaky** as above

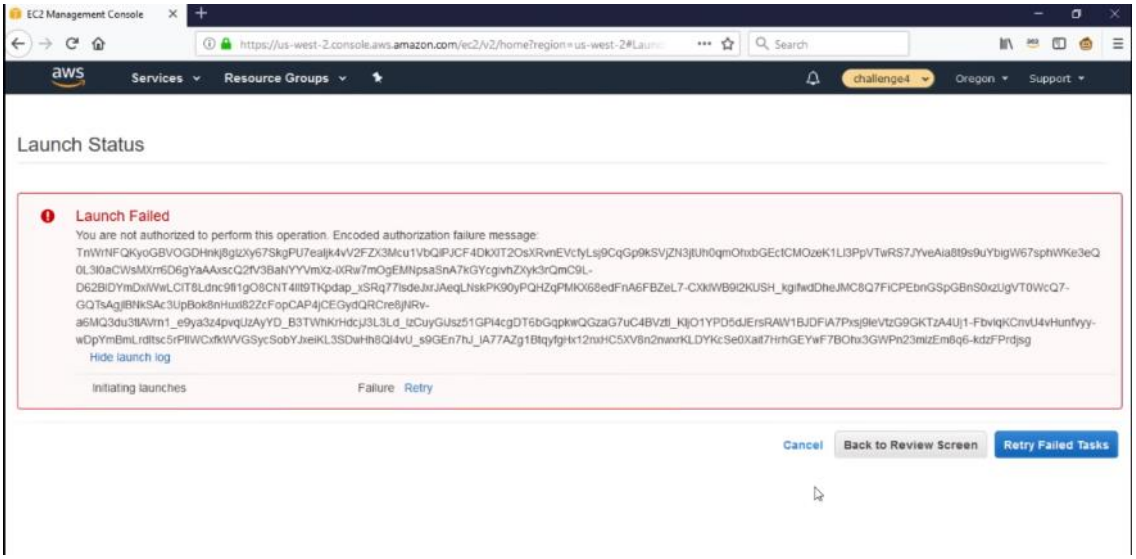




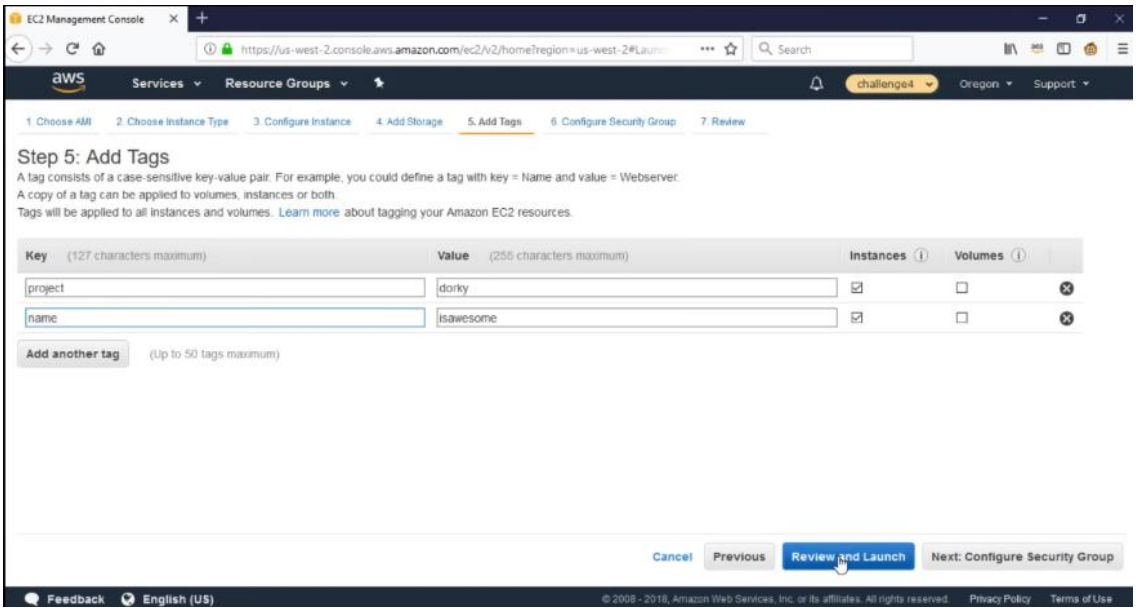
It failed because the Dorky team is not allowed to launch instances with tags starting with ‘sneaky’.

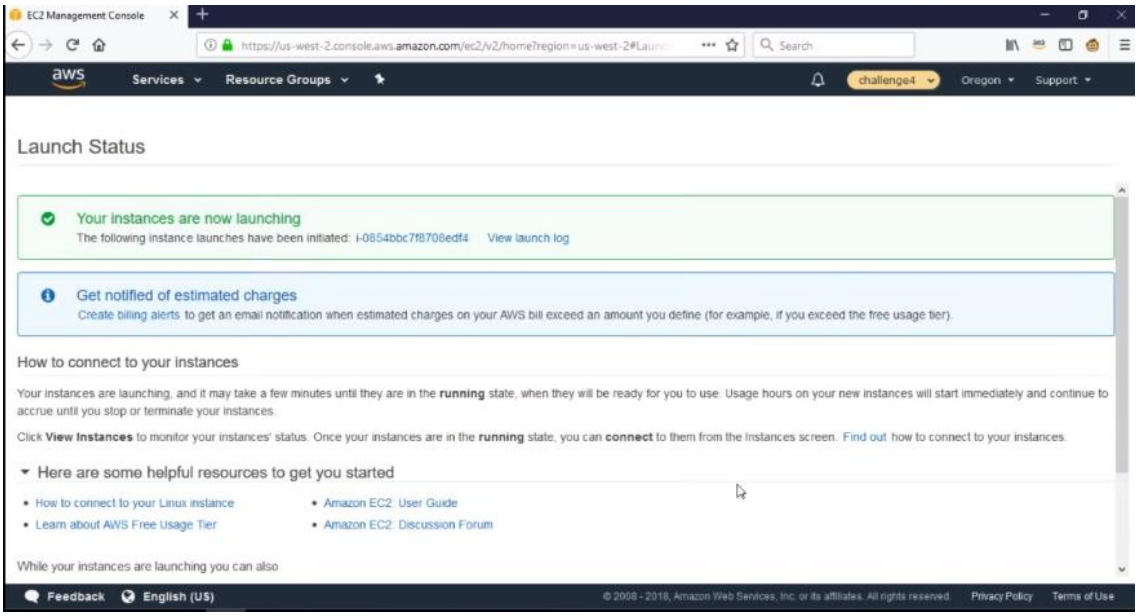


We try to launch another instance with the above tags again

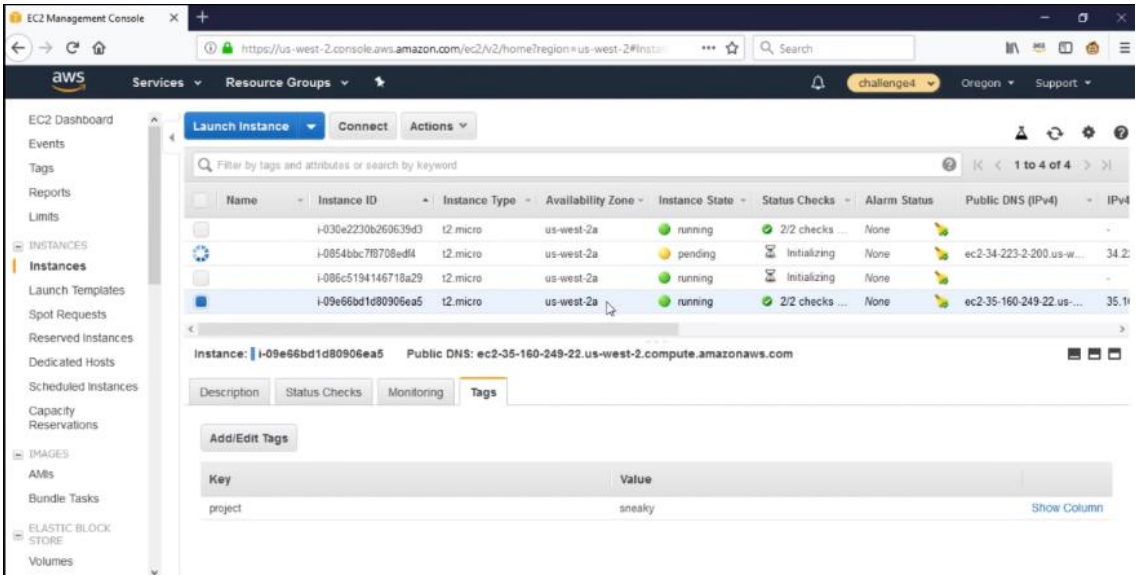


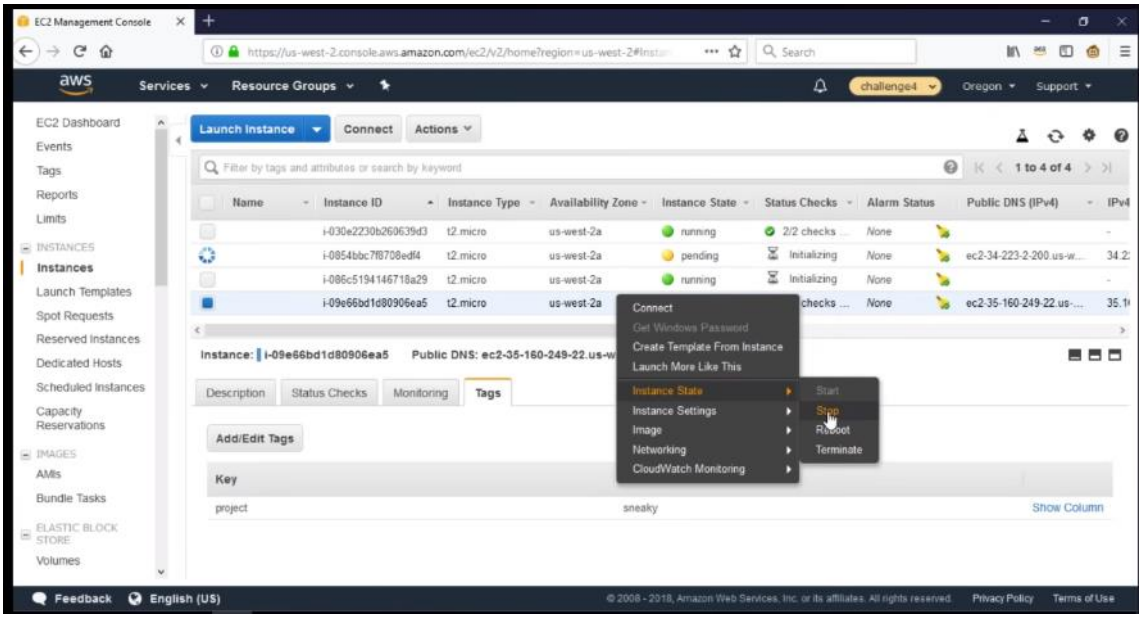
This failed because we can only have **project** and **name** tags and the project tag must have the value of **dorky** only.


EC2 Management Console
Services
Resource Groups
challenge4
Oregon
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 5: Add Tags
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.
Key (127 characters maximum)
Value (255 characters maximum)
Instances
Volumes
project
dorky
name
isawesome
Add another tag
(Up to 50 tags maximum)
Cancel
Previous
Review and Launch
Next: Configure Security Group
Feedback
English (US)
Privacy Policy
Terms of Use

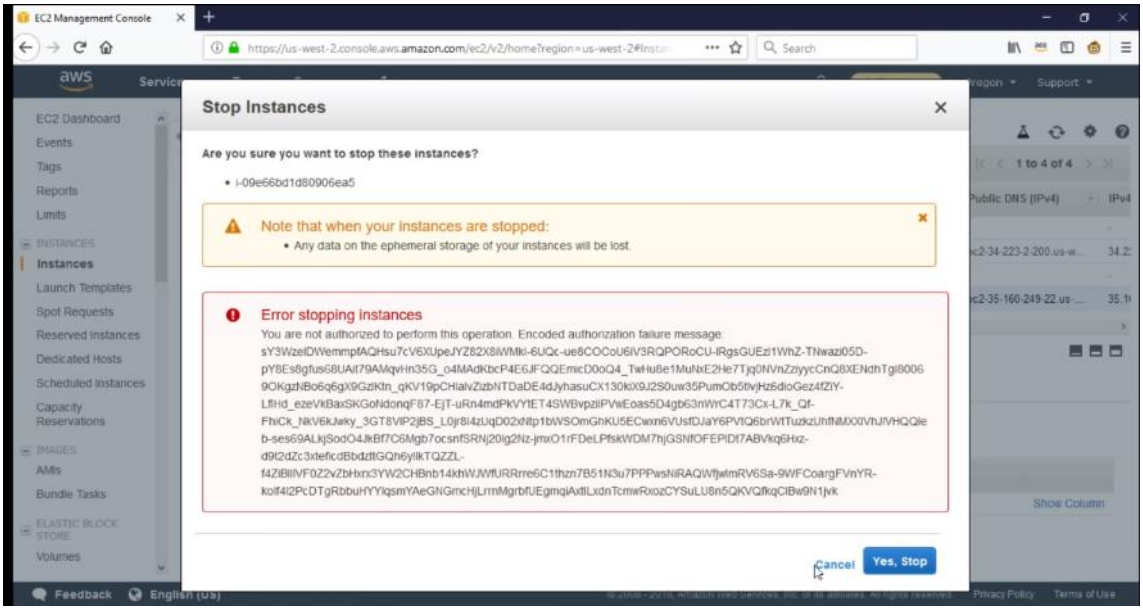

EC2 Management Console
Services
Resource Groups
challenge4
Oregon
Support
Launch Status
Your instances are now launching
The following instance launches have been initiated: i-0854bbc7f8708edf4
View launch log
Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).
How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click View Instances to monitor your instances' status. Once your instances are in the running state, you can connect to them from the Instances screen. Find out how to connect to your instances.
Here are some helpful resources to get you started
How to connect to your Linux instance
Learn about AWS Free Usage Tier
Amazon EC2: User Guide
Amazon EC2: Discussion Forum
While your instances are launching you can also
Feedback
English (US)
Privacy Policy
Terms of Use

It worked.

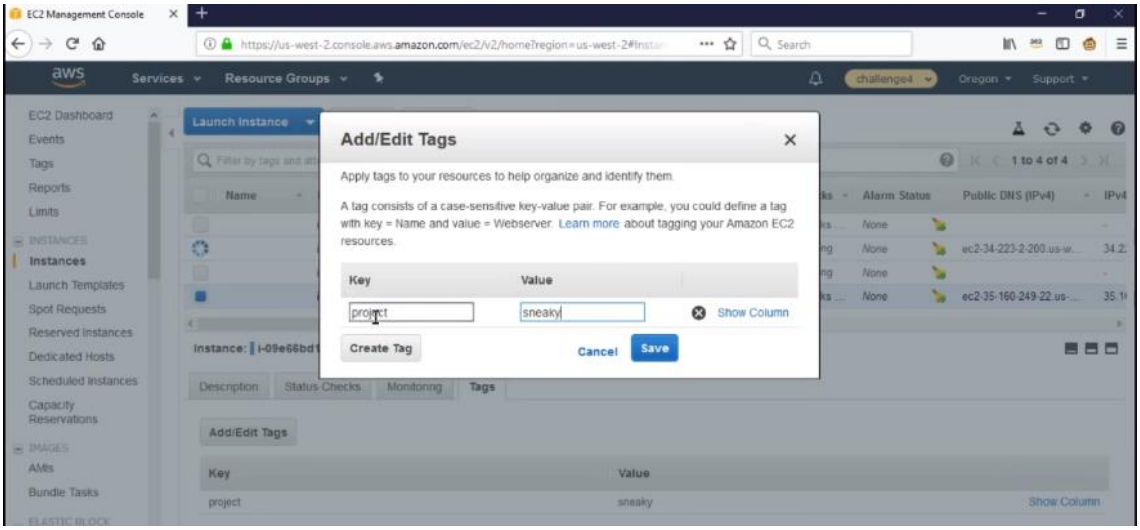

EC2 Management Console
Services
Resource Groups
challenge4
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Launch Instance
Connect
Actions
1 to 4 of 4
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
i-030e2230b260639d3 t2.micro us-west-2a running 2/2 checks ... None
i-0854bbc7f8708edf4 t2.micro us-west-2a pending Initializing None ec2-34-223-2-200.us-w...
i-086c5194146718a29 t2.micro us-west-2a running Initializing None
i-09e66bd1d80906ea5 t2.micro us-west-2a running 2/2 checks ... None ec2-35-160-249-22.us-...
Instance: i-09e66bd1d80906ea5 Public DNS: ec2-35-160-249-22.us-west-2.compute.amazonaws.com
Description
Status Checks
Monitoring
Tags
Add/Edit Tags
Key
Value
project
sneaky
Show Column

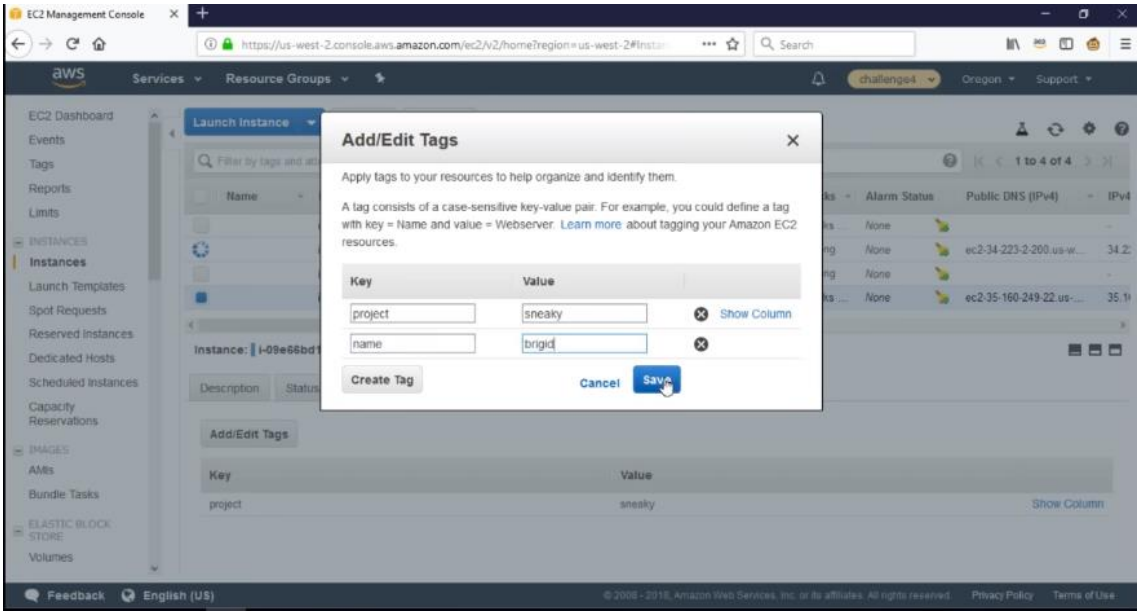


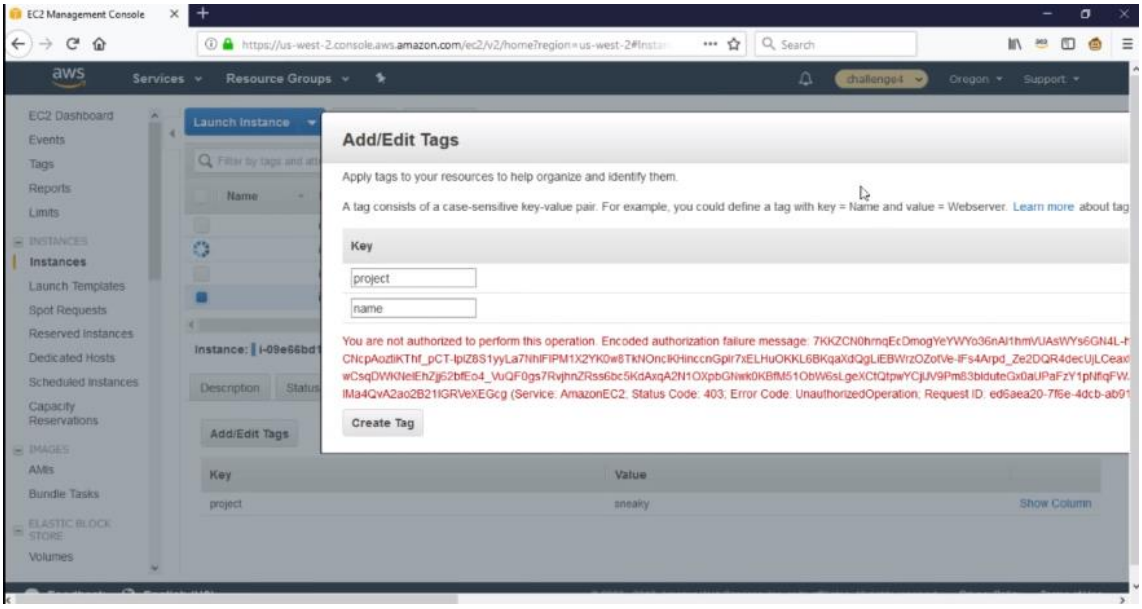
Let us try and stop an instance that belongs to the Sneaky team



We cannot stop instances that do not belong to the Dorky team that always has the project equals dorky tags.



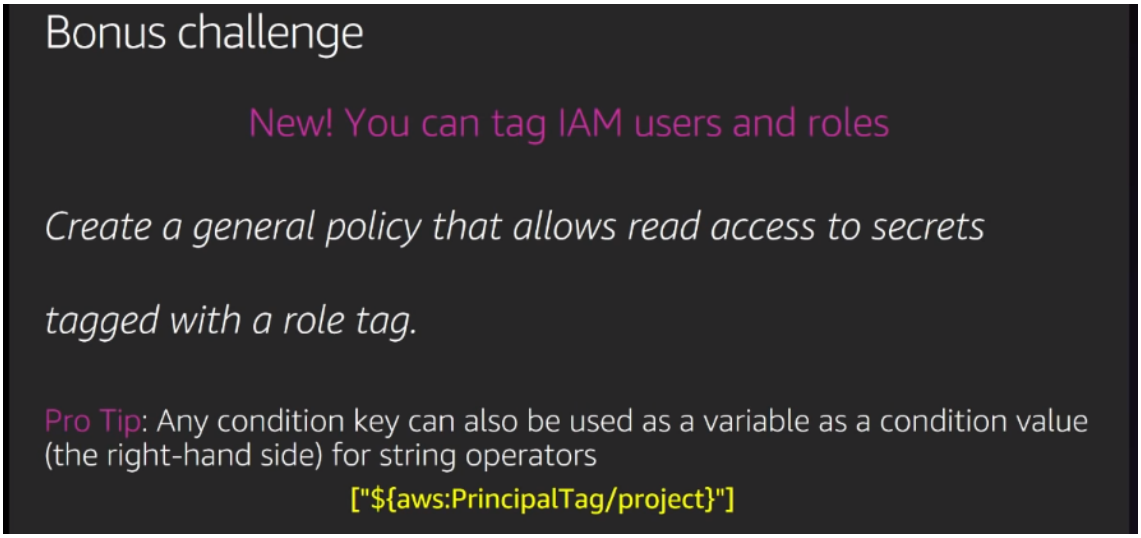




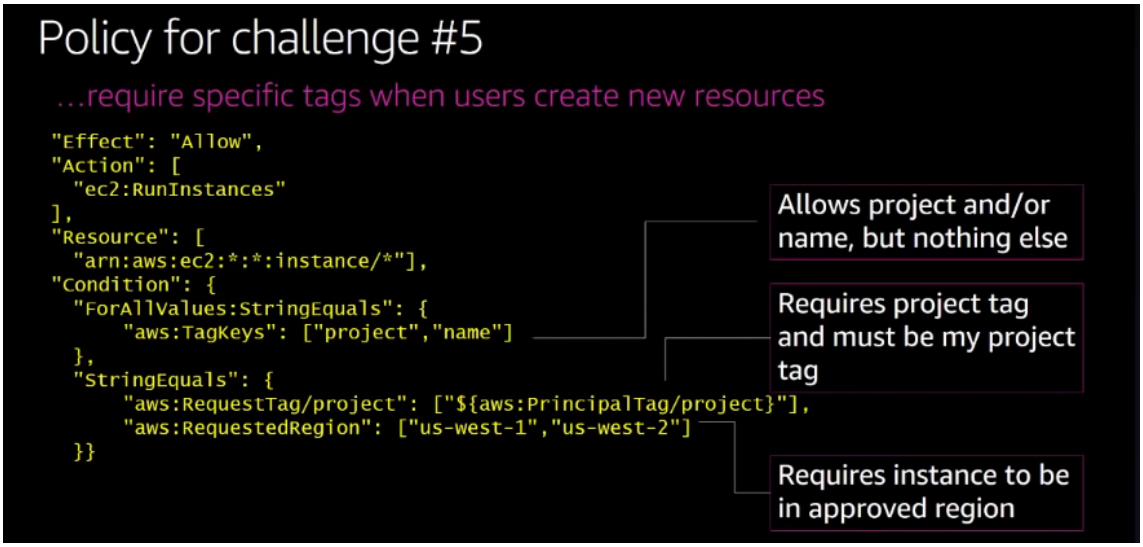
We also cannot rename or add tags to an instance that does not belong to our Dorky team.

Let's see tag-based access control in action

- Launch instances for project dorky
- Try to launch instances for project sneaky
- Modify tags on existing instances (dorky and sneaky)
- Manage existing instances



This means that you can use the tag on your user like dorky and sneaky to do things in the policy.



We are using the same previous policy except that we are now requiring that the project tag (dorky or sneaky) must be passed into the policy using **${aws:PrincipalTag/project}**

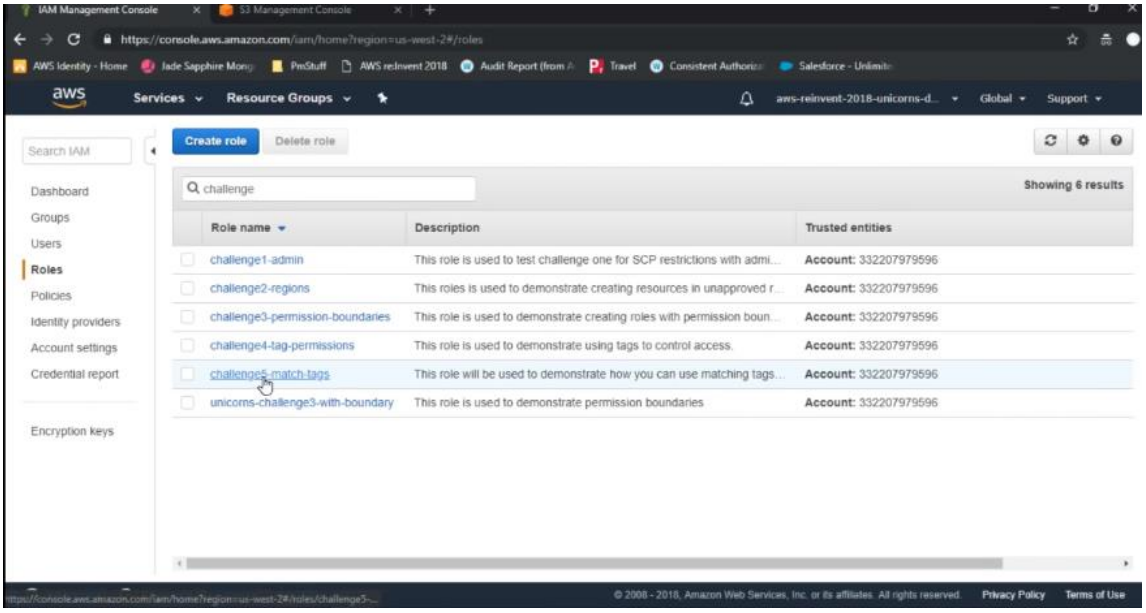
You can also now create tags only on existing resources that have your project tag only, if you pass in a project tag, it must also be your project tag only.

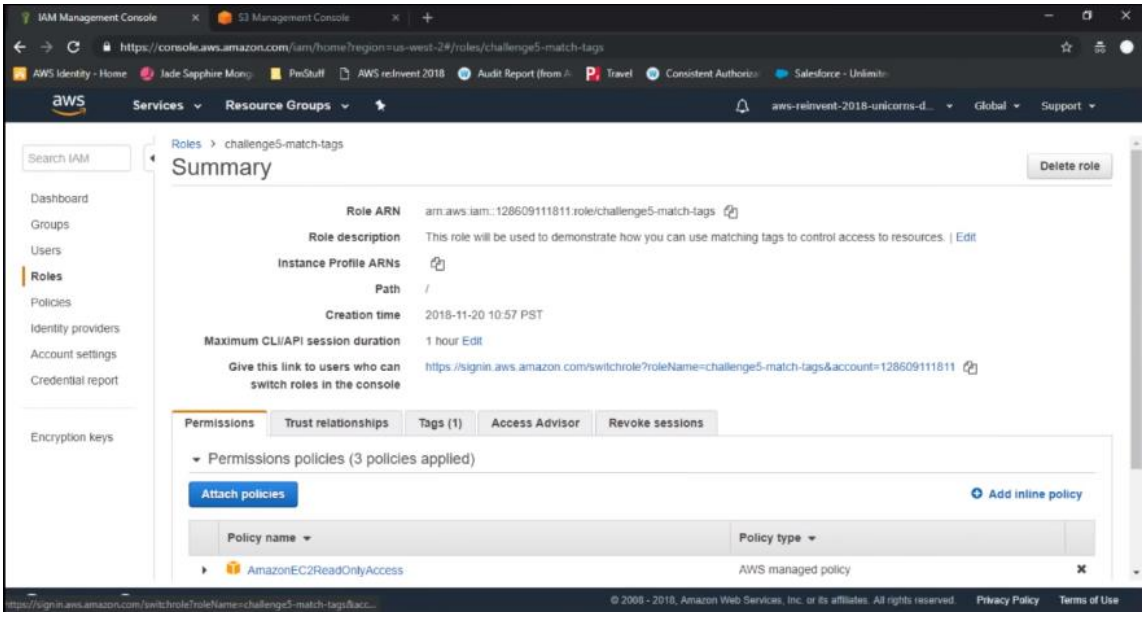
Policy for challenge #5

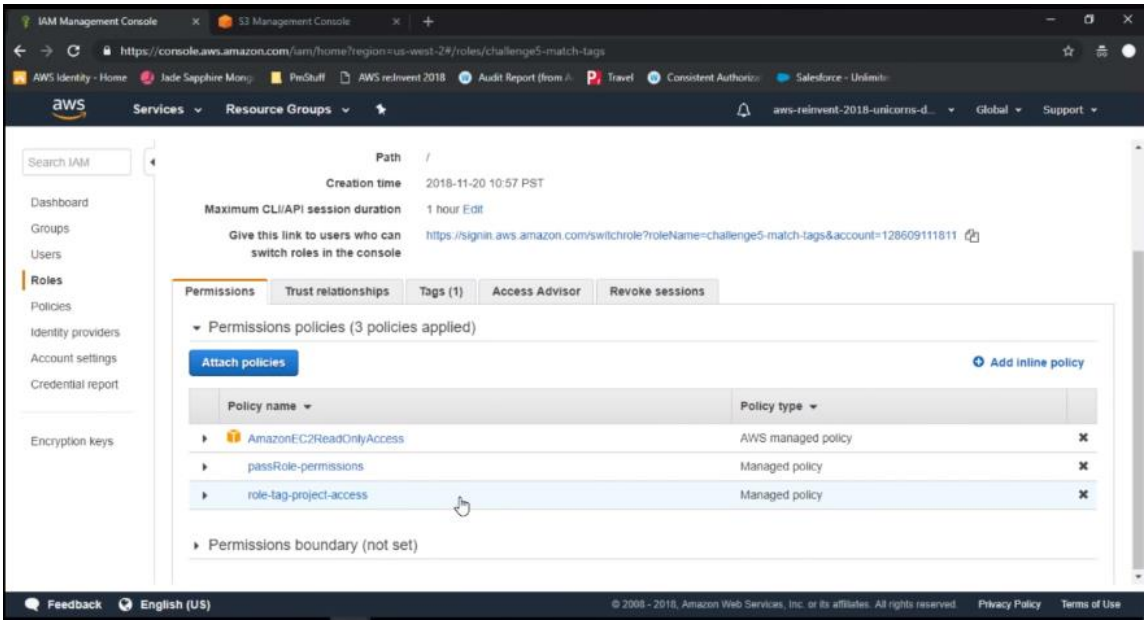
Control resources users can manage based on tag values

```
"Effect": "Allow",
"Action": [
       "ec2:StartInstances",
       "ec2:StopInstances"],
"Resource": "*",
"Condition": {
       "StringEquals": {
              "ec2:ResourceTag/project": "${aws:PrincipalTag/project}"
       }
}
}
```

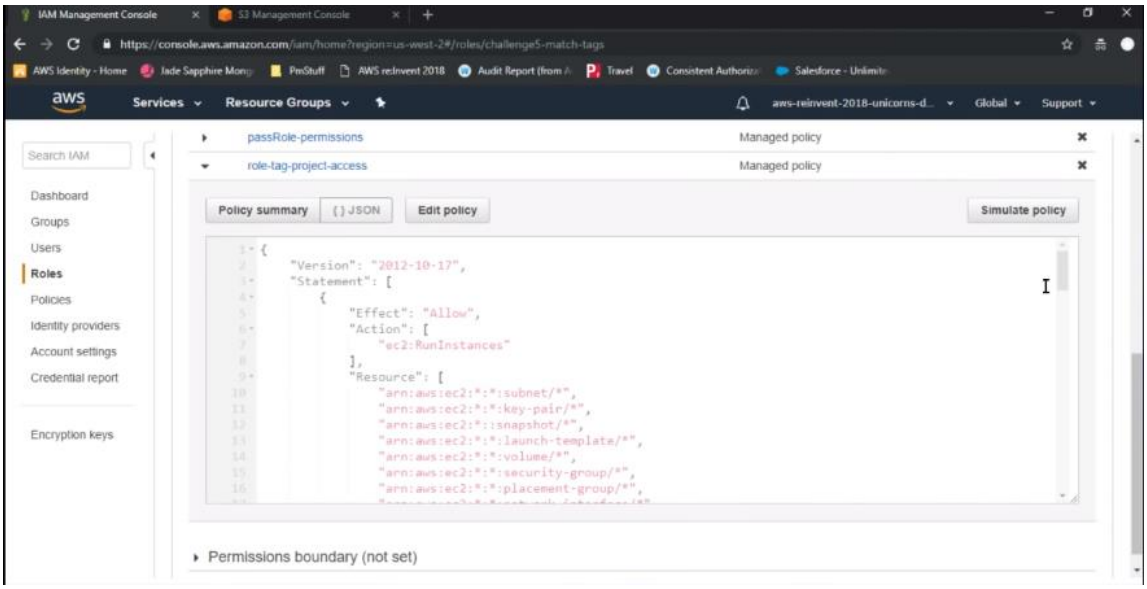
Only manage resources with my project tag

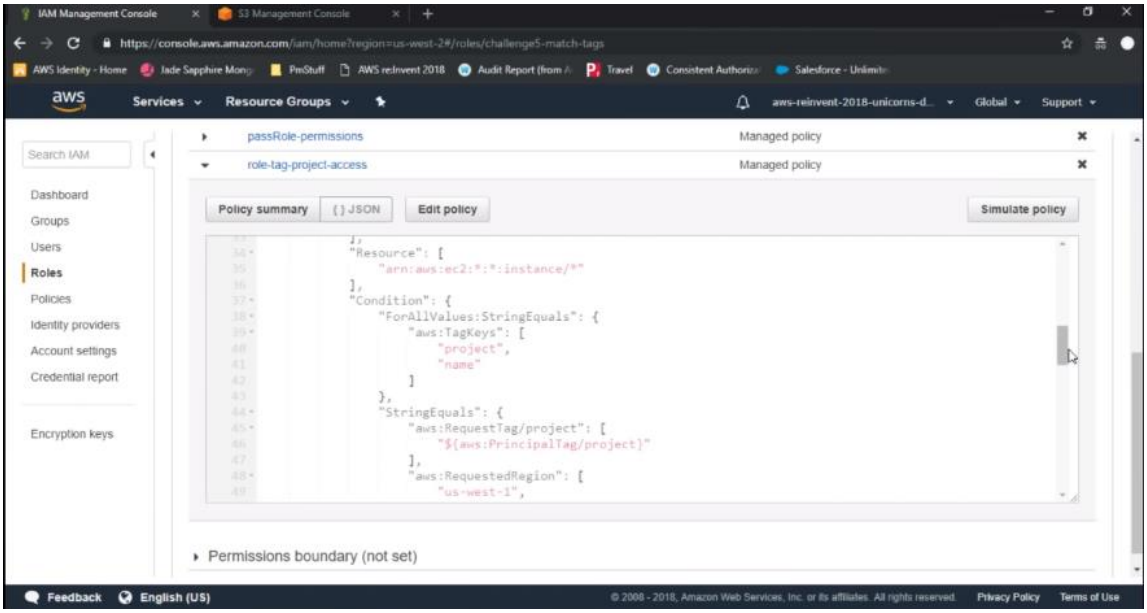


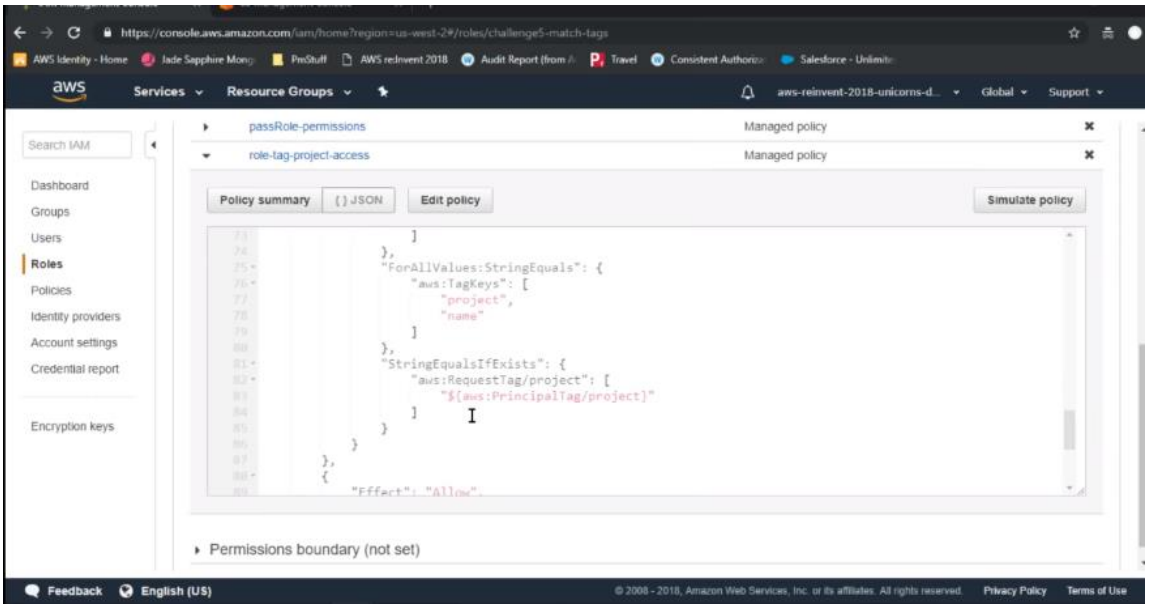


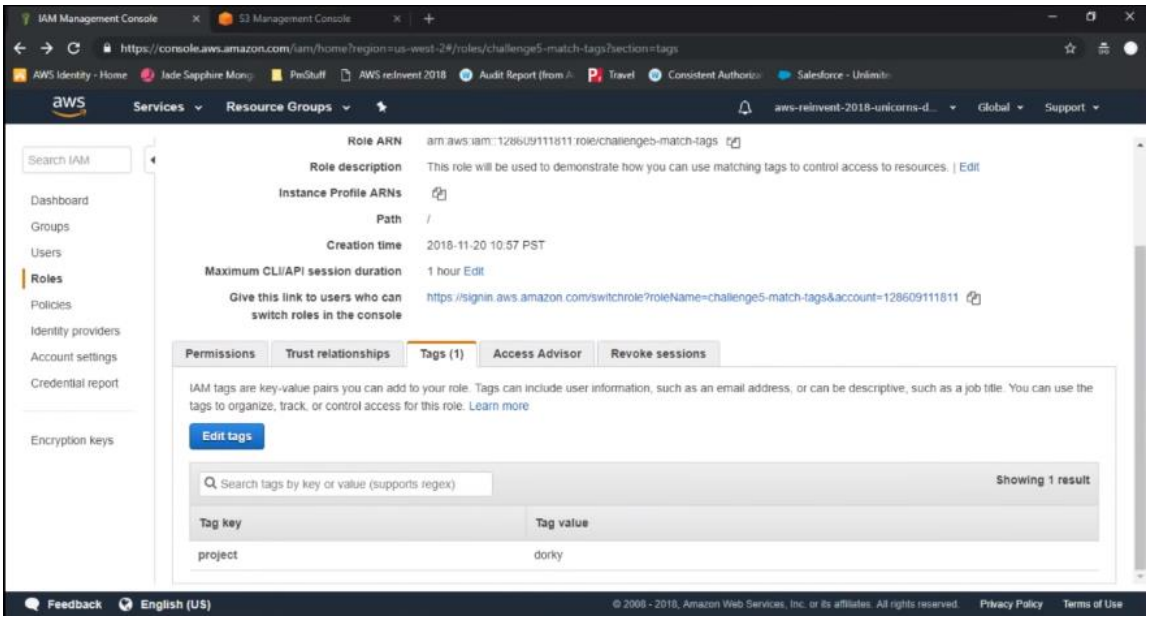


We now have this as the **role-tag-project-access** policy

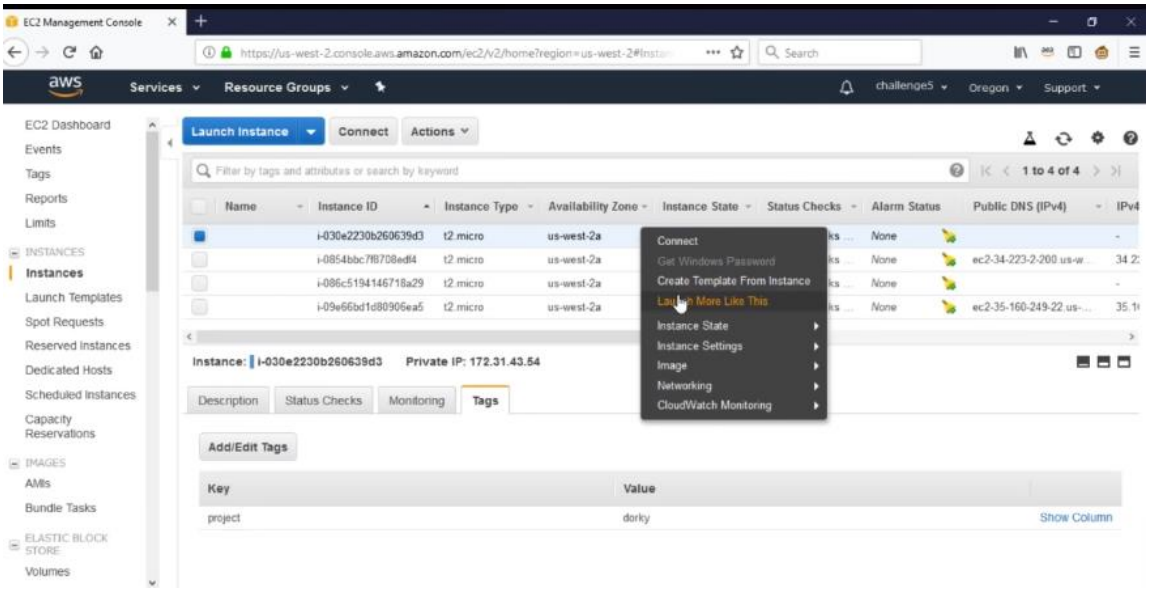


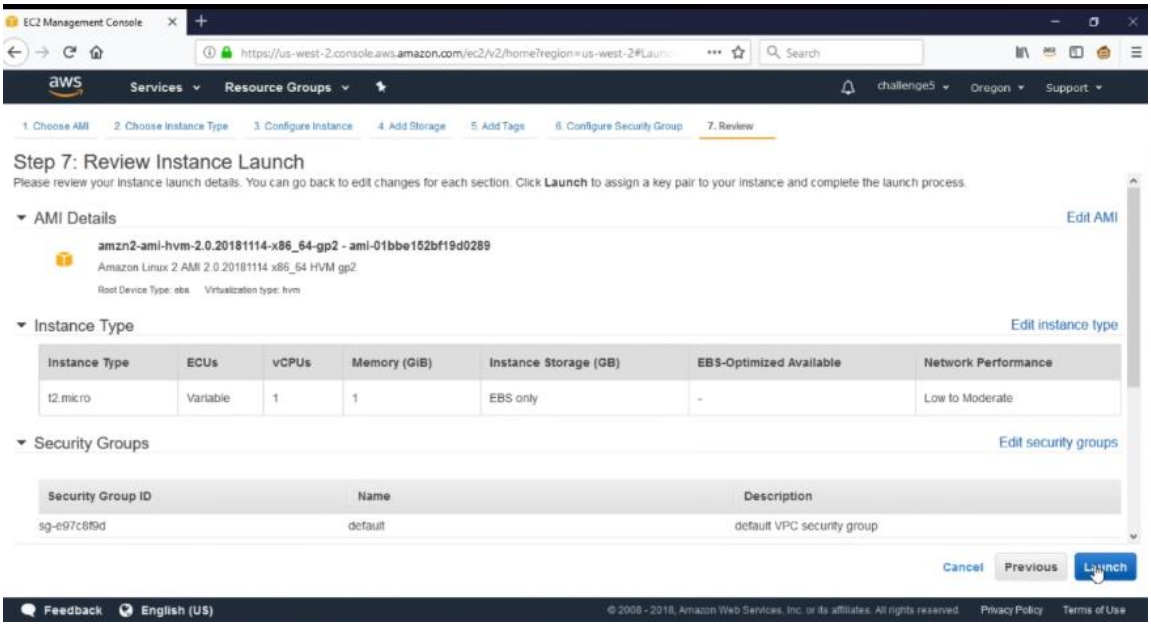


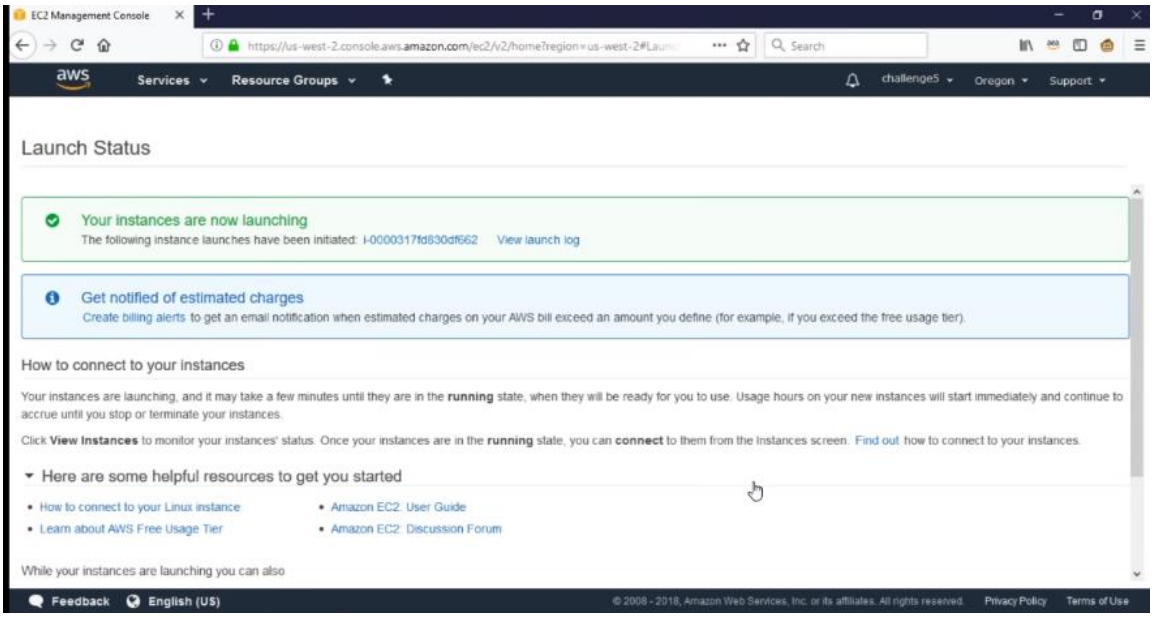




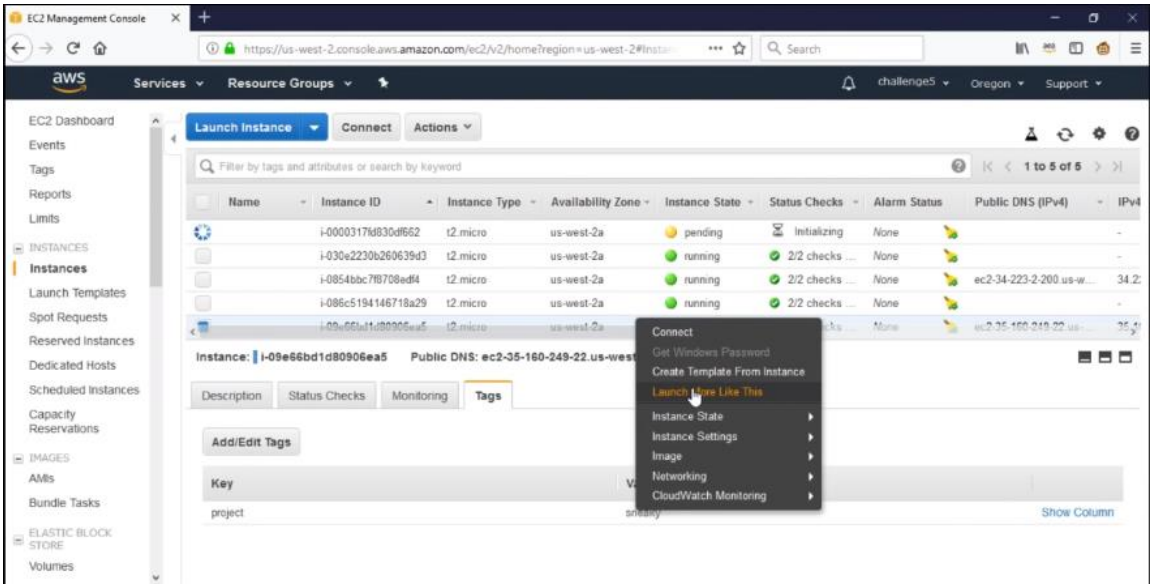
But we have added a tag for the project value to be dorky for this policy.

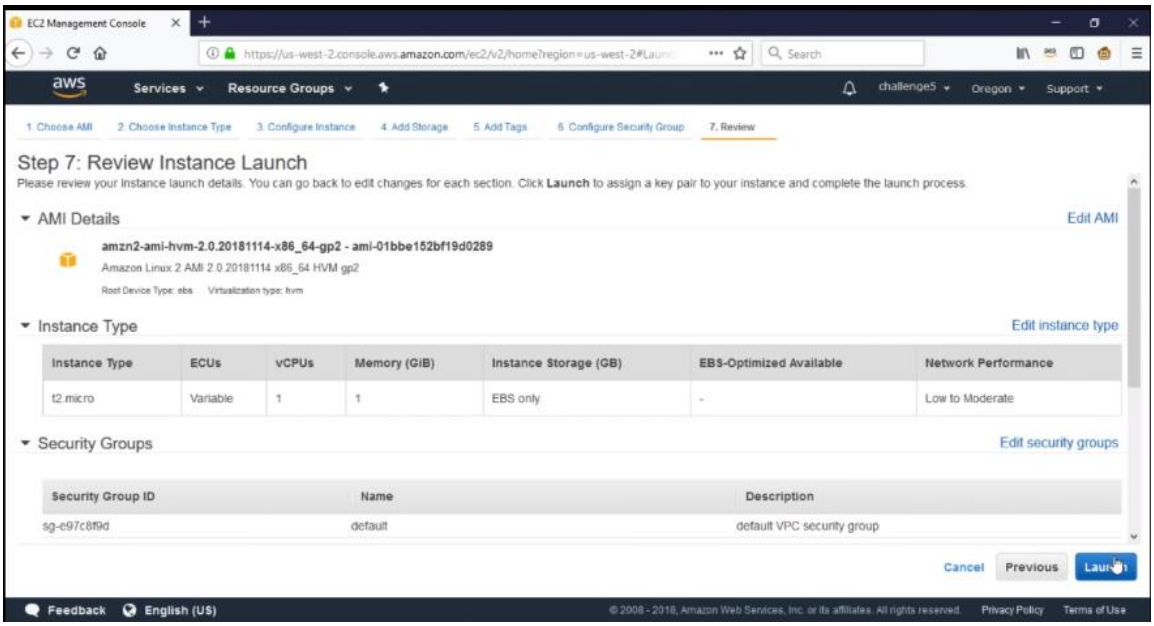


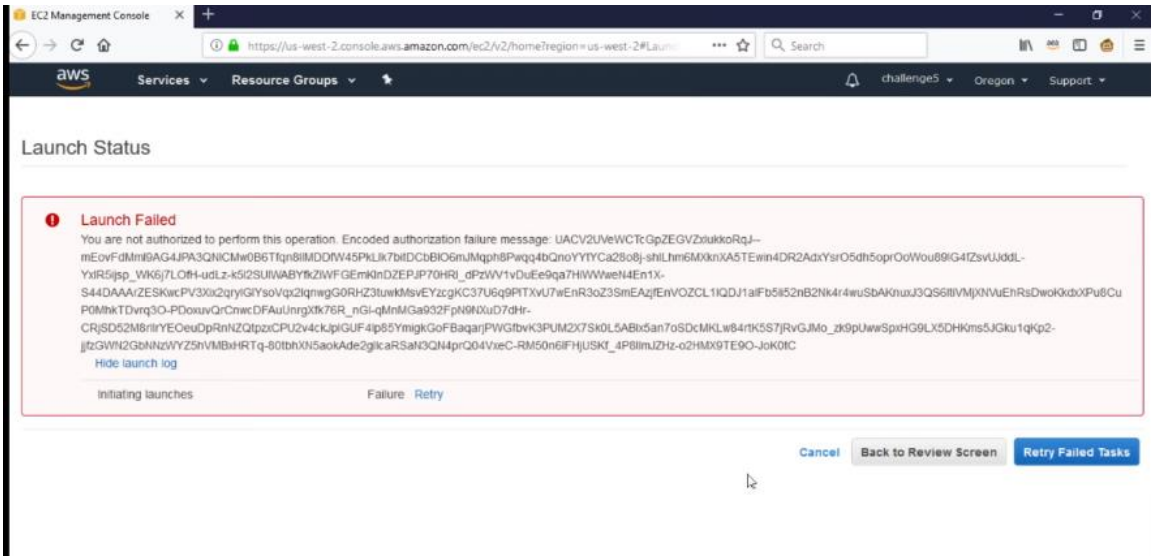




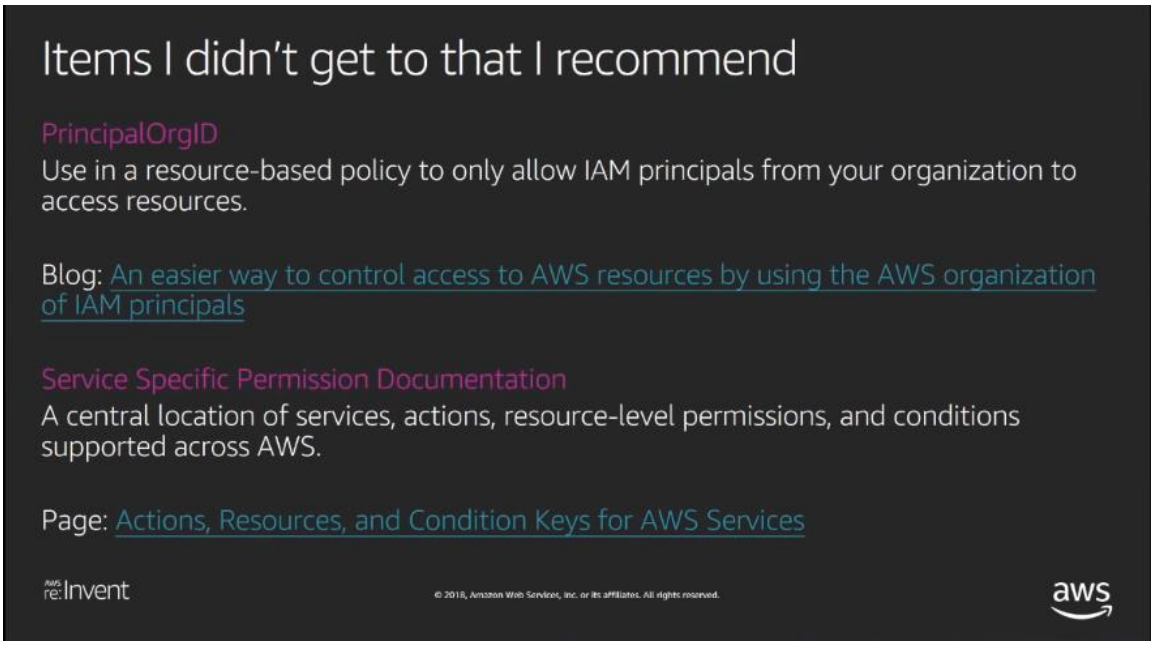
We can successfully launch instances with the dorky project tag.

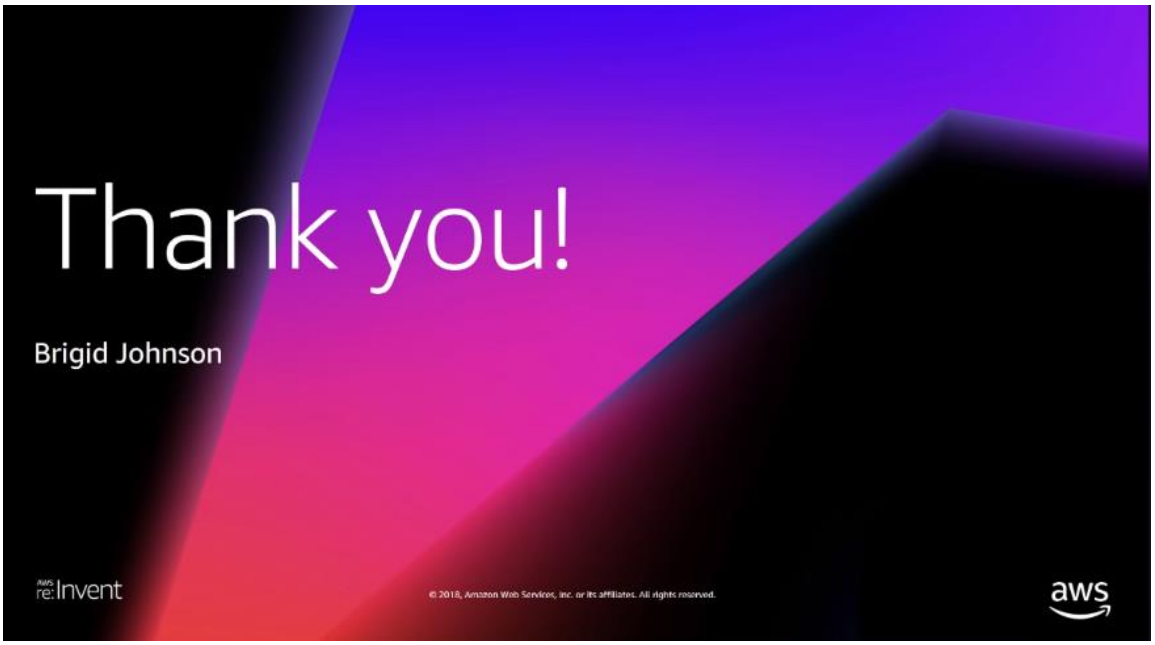






We cannot launch sneaky instances.




Thank you!
Brigid Johnson
aws re:Invent
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
aws