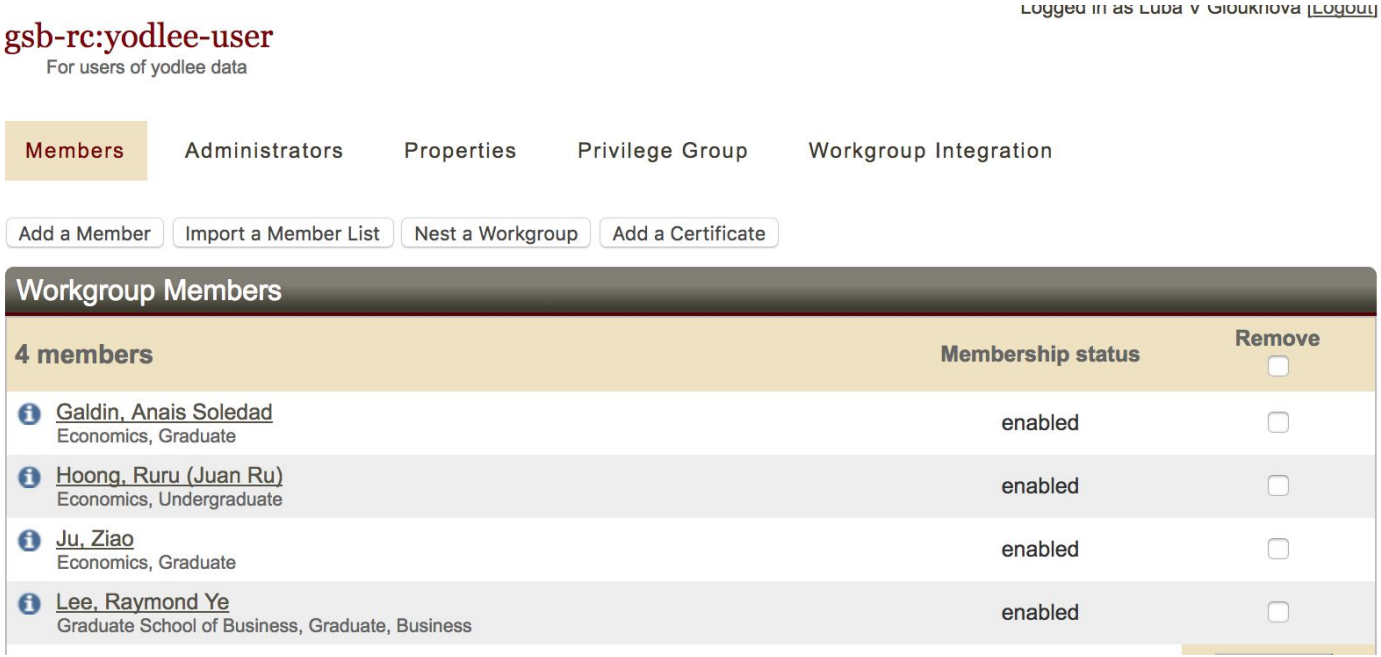
AWS + SSO

overcoming challenges

AWS CLI w/ Roles

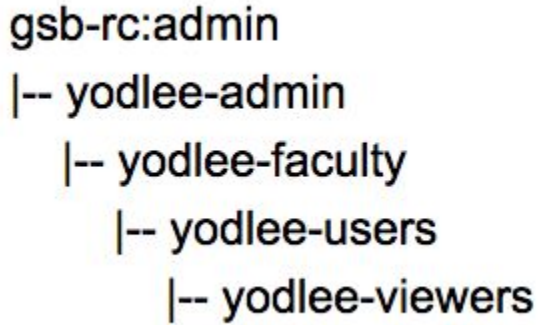
Motivation

An SSO user (SUNet ID) inherits permissions via a an AWS Role through their membership to a Stanford Workgroup.



Motivation

Good news: instead of administering an account and/or IAM for each faculty member and/or RA, we just administer a single level of permission. Plus, we get all the extra security associated with leveraging SUNet (MFA, id expiration).



Motivation

Bad news: These Roles do not have keys associated with them making aws cli use impossible without some backend engineering

Quick Configuration

For general use, the `aws configure` command is the fastest way to set up your AWS CLI installation.

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

The AWS CLI will prompt you for four pieces of information. AWS Access Key ID and AWS Secret Access Key are your account credentials.

Solution

Programmatically create a temporary IAM role by following the instructions here:

https://aws.amazon.com/blogs/security/how-to-implement-a-general-solution-for-federated-apicli-access-using-saml-2-0/

Our code:

https://code.stanford.edu/morrowwr/awscli-console

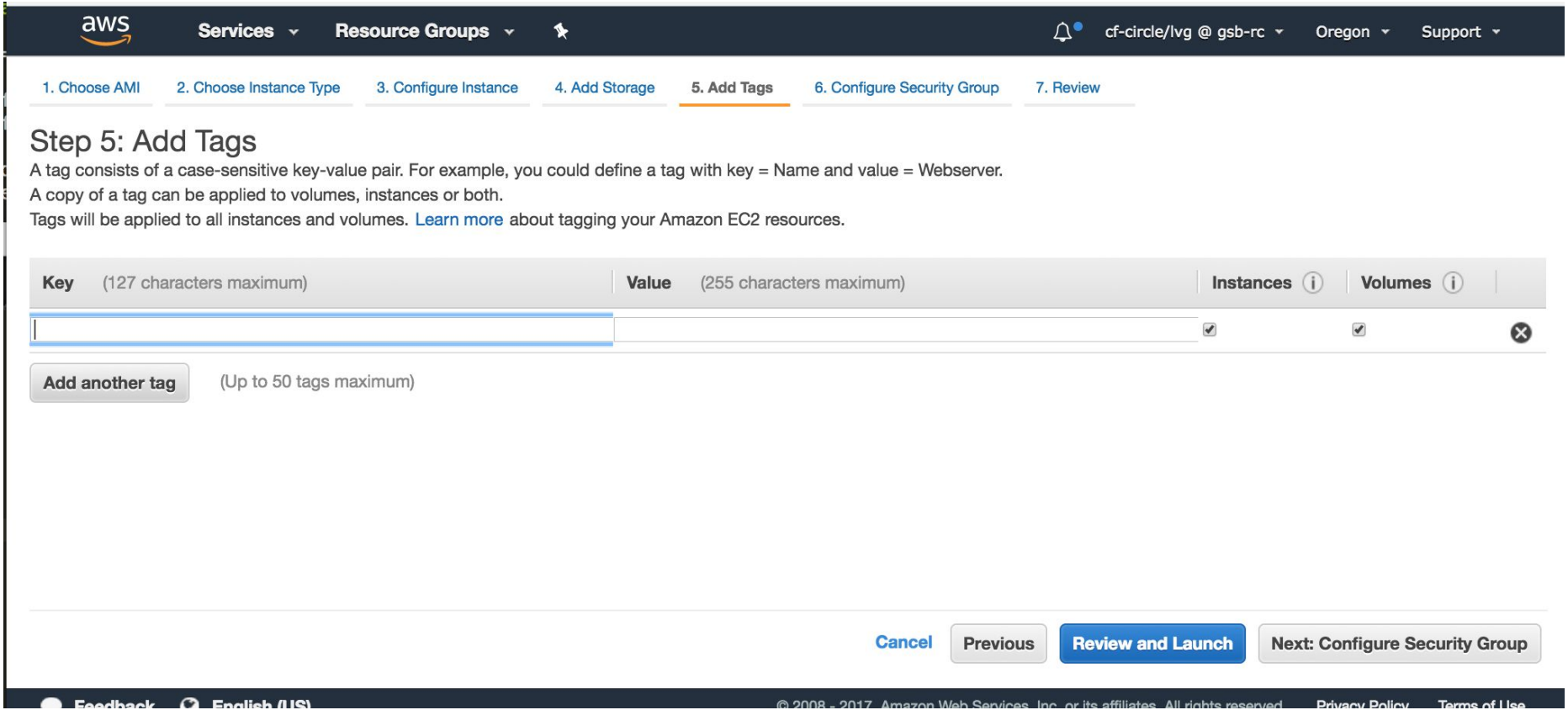
```
[lvg@yen4:/ifs/gsb/aws_rolecli$ ./awsconsole.py                                                    ]

--------------------------------------------------------------------------------------------------
 STANFORD GSB AWS CLI CONSOLE
--------------------------------------------------------------------------------------------------

  Version 0.1
  For support, email gsb_circle_research@stanford.edu

--------------------------------------------------------------------------------------------------

  Stanford WebAuth


  Enter your SUNet credentials:
[ Username: lvg                                                                                    ]
[ Password:                                                                                        ]

  Submitting credentials...
  Response did not contain a valid SAML assertion... checking for MFA...

--------------------------------------------------------------------------------------------------

  Please choose the role you would like to assume:

  [0]: arn:aws:iam::929035564788:role/comscore-admin
  [1]: arn:aws:iam::637853756653:role/cf-aaggar
  [2]: arn:aws:iam::175825637954:role/yodlee-admin
  [3]: arn:aws:iam::637853756653:role/cf-circle
  [4]: arn:aws:iam::637853756653:role/rss_practicum
  [5]: arn:aws:iam::385850523699:role/appnexus-admin
[ Selection:  2                                                                                    ]


--------------------------------------------------------------------------------------------------

  Your new access key pair has been stored in the AWS configuration file
  /afs/.ir/users/l/v/lvg/.aws/credentials under the saml profile.
  Note that it will expire in 1 hour -- Mon Nov 13 20:58:39 2017 UTC.
  After this time, you may safely rerun this script to refresh your access key pair.
  To use this credential call the AWS CLI with the --profile option, as in

        aws --profile saml ec2 describe-instances

--------------------------------------------------------------------------------------------------

[lvg@yen4:/ifs/gsb/aws_rolecli$ aws --profile saml s3 cp /ifs/gsb/diamondr/yshare/data/panel_m_members.dta s3://yodlee-diamondr/An]
ais/panel_m_members.dta
```

Auto-Tagging EC2

Motivation

As is, if an SSO user were to spin up an EC2 instance, the cost associated with that instance would be difficult to parse out of total costs across all instances in the account.

However, tagging resources (EC2 instances) by unique identifier of the creator for allows for improved cost allocation purposes


aws
Services
Resource Groups
cf-circle/lvg @ gsb-rc
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
Add another tag
(Up to 50 tags maximum)
Cancel
Previous
Review and Launch
Next: Configure Security Group
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

```
aws ec2 run-instances --image-id ami-c3b8d6aa --count 1 --instance-type t2.medium
 --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
--associate-public-ip-address
```

`--tag-specifications` (list)

> The tags to apply to the resources during launch. You can tag instances and volumes. The specified tags are applied to all instances or volumes that are created during launch.

Shorthand Syntax:

```
ResourceType=string,Tags=[{Key=string,Value=string},{Key=string,Value=string}] ...
```

... granting users the permissions to manually assign tags

does not solve the problem

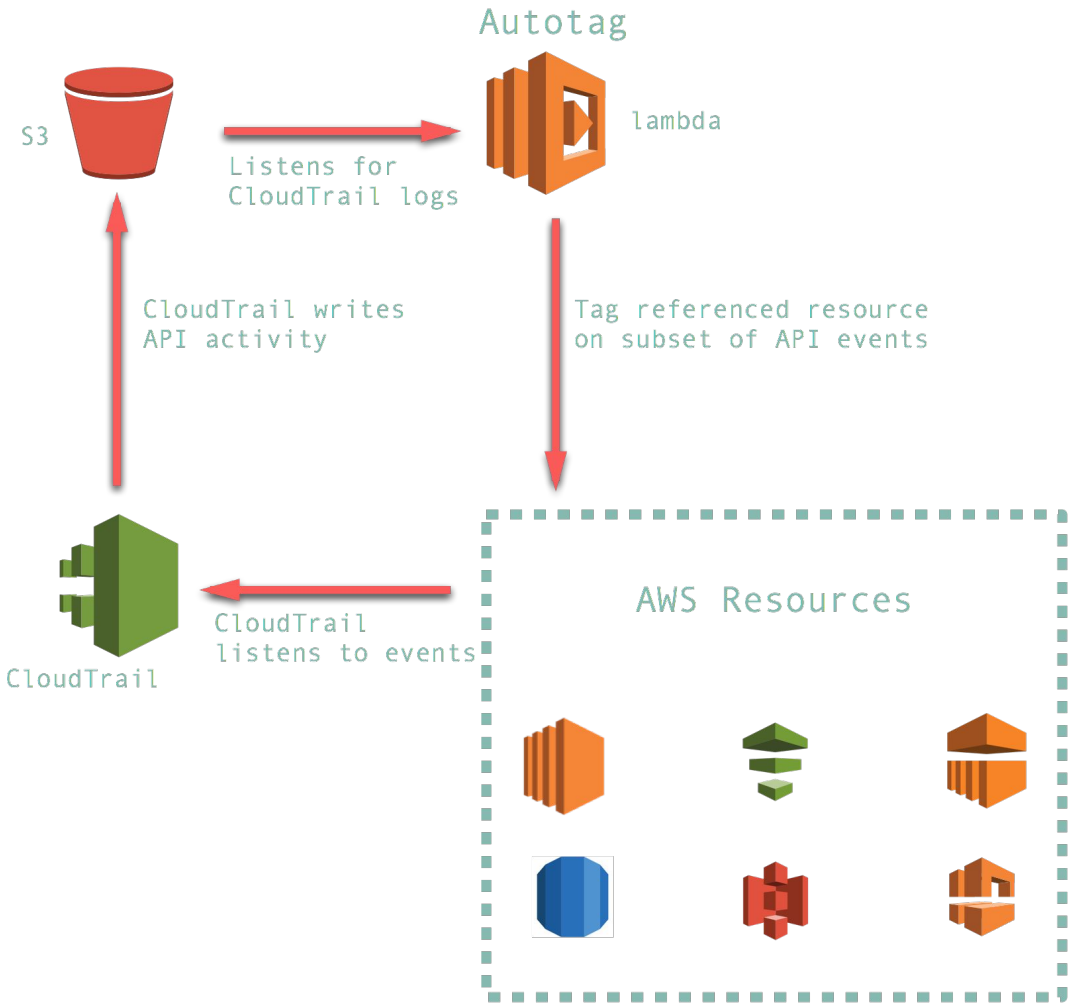
1. users may fail to tag
2. users may tag incorrectly

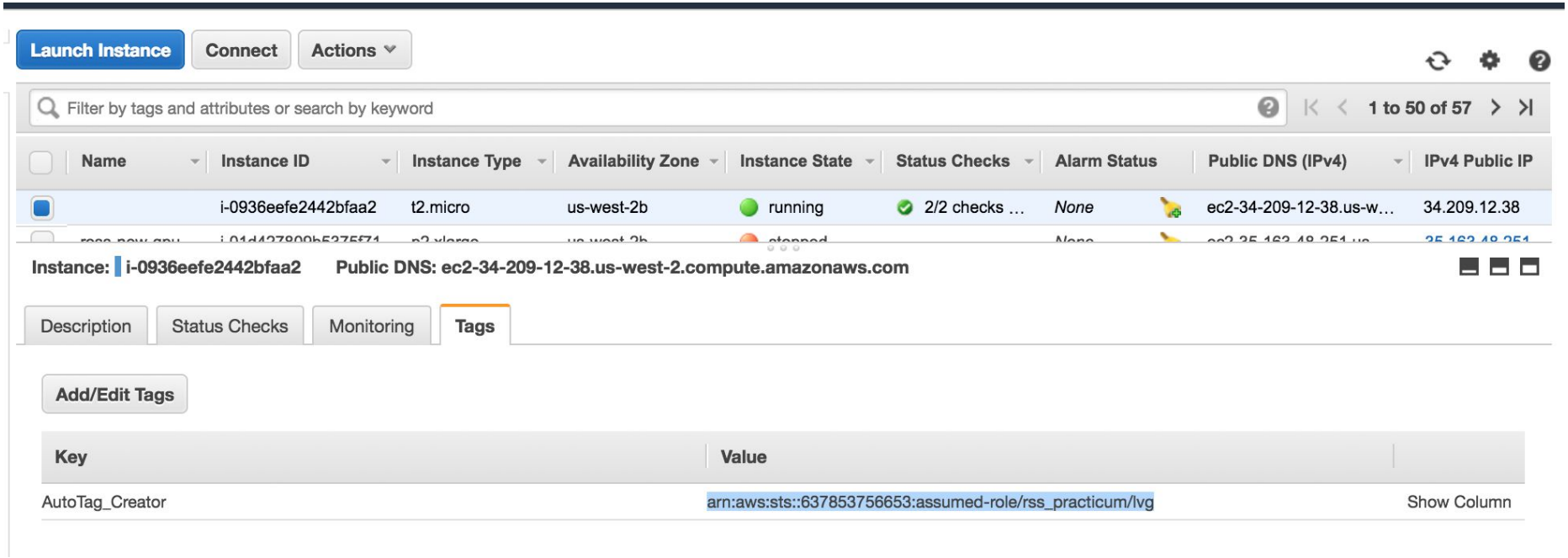
Solution

Programmatically tag EC2 Resources

AWS Lambda+S3+Cloudtrail application via AWS CloudFormation template

https://github.com/GorillaStack/auto-tag


Autotag
S3
lambda
Listens for
CloudTrail logs
CloudTrail writes
API activity
Tag referenced resource
on subset of API events
CloudTrail
CloudTrail
listens to events
AWS Resources


Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 50 of 57
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
i-0936eefe2442bfaa2
t2.micro
us-west-2b
running
2/2 checks ...
None
ec2-34-209-12-38.us-w...
34.209.12.38
Instance: i-0936eefe2442bfaa2
Public DNS: ec2-34-209-12-38.us-west-2.compute.amazonaws.com
Description
Status Checks
Monitoring
Tags
Add/Edit Tags
Key
Value
AutoTag_Creator
arn:aws:sts::637853756653:assumed-role/rss_practicum/lvg
Show Column

Cost Allocation Tags

AWS-Generated Cost Allocation Tags

A *resource created by tag* is an AWS-generated cost allocation tag containing resource creator information that is automatically applied to the resources that you create. This feature is only available in the Billing & Cost Management console, and will not appear anywhere else in the AWS console, including the Tag Editor.

Activate

User-Defined Cost Allocation Tags

✔ Finished loading tags.

Activating tags for cost allocation tells AWS that the associated cost data for these tags should be made available throughout the billing pipeline. Once activated, cost allocation tags can be used as a dimension of grouping and filtering in Cost Explorer, as well as for refining AWS budget criteria.

Clicking the Refresh button will prioritize your account for updates, so that tags from your linked accounts are visible to you sooner. Please note that the Refresh operation can only be triggered once every 24 hours.

Activate | Deactivate | Undo | Refresh

Filter: All tags | AutoTag_Creator | Tags per page: 100

Tag key*	Status
AutoTag_Creator	Inactive

Nov 22, 2017 - Nov 29, 2017 | Daily | Group by: TagKey: AutoT... | Stack