

Secure multi-account AWS with aws-vault

Why multi-account?

- **Separate concerns (by product, business unit, etc.)**
- **Consolidated billing (specified by account)**
- **Security**

The bastion/security/identity account

- **AWS account with no compute resources, only IAM.**
- **A single place to manage IAM users.**
- **A single set of IAM credentials per user.**
- **Users *assume roles* in other accounts.**

Examples

Group *Developers* can access all resources in the *Product* account.

Group *IT* can access all resources in the *Corporate* account and IAM in the *Bastion* account.

Group *Finance* can access billing details in the *Master* account.

**This is useful even if you only have
a Master account and a Bastion
account**

- **Manage *users* in the Bastion account.**
- **Manage *roles* in your Master account.**

Setting up

In your *Master* account, create a new IAM role. Make the trusted entity your Bastion account.

console.aws.amazon.com/iam/home?region=eu-west-1#/role

aws Services Resource Groups martin @ 6528-9943-2631 Global Support

Create role

1 2 3

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 entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID* ⓘ

Options

- ☐ Require external ID (Best practice when a third party will assume this role)
- ☒ Require MFA ⓘ

* Required

Cancel **Next: Permissions**

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

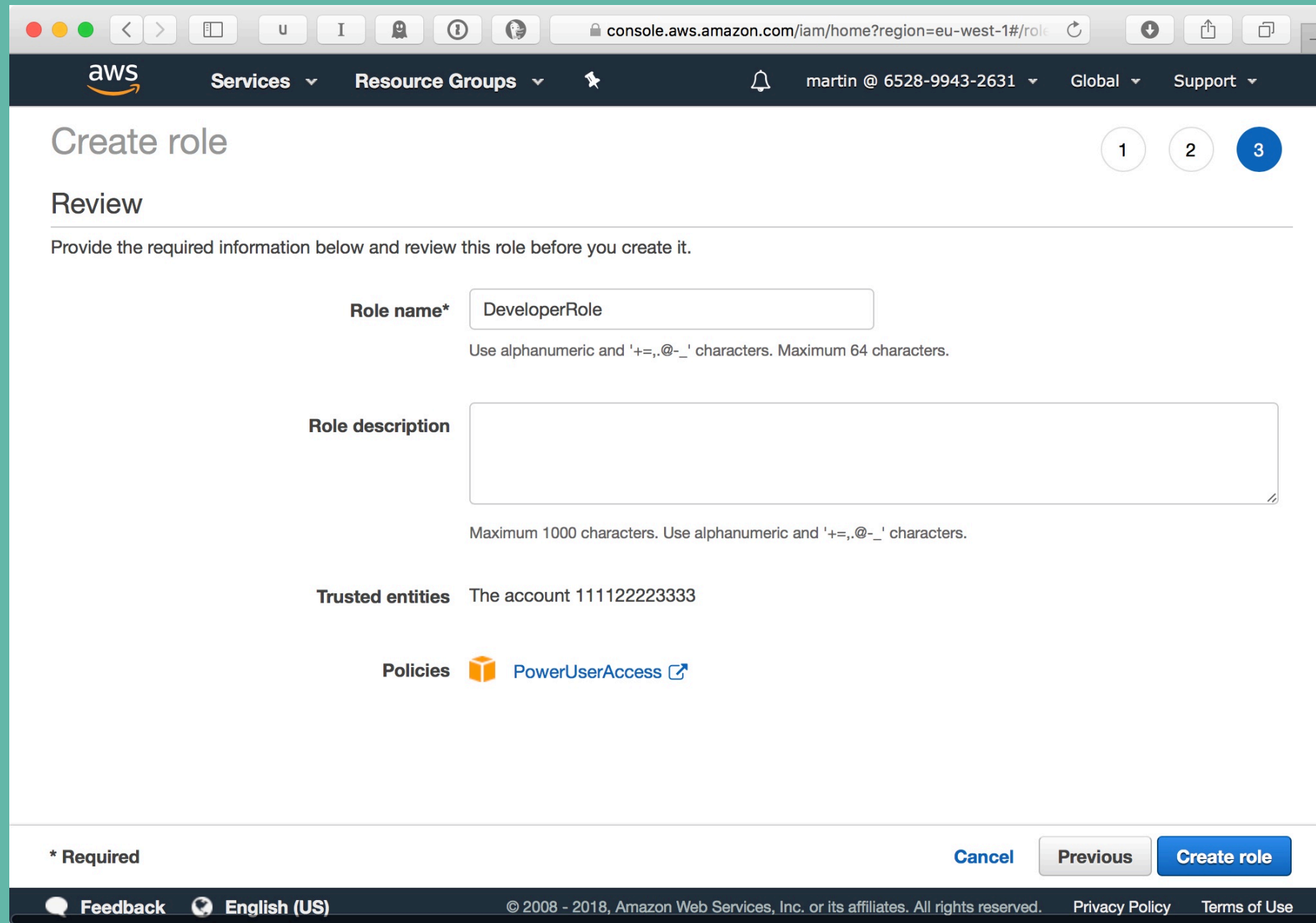
Attach permissions.

The screenshot shows the AWS IAM console interface for creating a new role. The page is titled 'Create role' and is on step 2 of a 3-step process. The current step is 'Attach permissions policies', where the user is prompted to 'Choose one or more policies to attach to your new role.' There are buttons for 'Create policy' and 'Refresh'. A search bar shows 'power' and 'Showing 6 results'. A table lists the following policies:

	Policy name	Attachments	Description
<input type="checkbox"/>	AmazonCognitoPowerUser	0	Provides administrative access to existing Amazon Cognito r...
<input type="checkbox"/>	AmazonEC2ContainerRegistryPowerUser	0	Provides full access to Amazon EC2 Container Registry repo...
<input type="checkbox"/>	AWSCodeCommitPowerUser	0	Provides full access to AWS CodeCommit repositories, but d...
<input type="checkbox"/>	AWSDataPipeline_PowerUser	0	Provides full access to Data Pipeline, list access for S3, Dyn...
<input type="checkbox"/>	AWSKeyManagementServicePowerUser	0	Provides access to AWS Key Management Service (KMS).
<input checked="" type="checkbox"/>	PowerUserAccess	1	Provides full access to AWS services and resources, but doe...

At the bottom, there are buttons for 'Cancel', 'Previous', and 'Next: Review'. The footer includes 'Feedback', 'English (US)', and copyright information.

Name and create the role.



The screenshot shows the AWS IAM console interface for creating a new role. The browser address bar indicates the URL is `console.aws.amazon.com/iam/home?region=eu-west-1#/role`. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a notification bell, the user 'martin @ 6528-9943-2631', 'Global', and 'Support'.

The main heading is 'Create role', with three numbered steps (1, 2, 3) in the top right, where step 3 is currently active. Below this is the 'Review' section, which includes the instruction: 'Provide the required information below and review this role before you create it.'

The form contains the following fields:

- Role name***: A text input field containing 'DeveloperRole'. Below it, a note states: 'Use alphanumeric and '+=,.-_' characters. Maximum 64 characters.'
- Role description**: A large text area. Below it, a note states: 'Maximum 1000 characters. Use alphanumeric and '+=,.-_' characters.'
- Trusted entities**: A label followed by the text 'The account 111122223333'.
- Policies**: A label followed by a cube icon and a link to 'PowerUserAccess'.

At the bottom of the form, there is a legend for '* Required', and three buttons: 'Cancel', 'Previous', and 'Create role'.

The footer of the console includes a 'Feedback' link, 'English (US)' language selection, and copyright information: '© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.' It also includes links for 'Privacy Policy' and 'Terms of Use'.

Trust policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBastionAccountUserWithMFA",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<BASTION-ACCOUNT-ID>:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "Bool": {
          "aws:MultiFactorAuthPresent": "true"
        }
      }
    }
  ]
}
```

In your *Bastion* account, create a policy that allows users to assume the role in the *Master* account.

The screenshot shows the AWS IAM console 'Create policy' page. The browser address bar shows 'console.aws.amazon.com/iam/home?region=eu-west-1#/policy'. The page header includes the AWS logo, 'Services', 'Resource Groups', a user profile 'root @ identity', and 'Global' and 'Support' links. The main heading is 'Create policy' with two numbered steps (1 and 2). Below the heading is a description: 'A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)'. There are two tabs: 'Visual editor' (selected) and 'JSON'. A link 'Import managed policy' is on the right. Below the tabs are links 'Expand all' and 'Collapse all'. The policy configuration section is titled 'STS (1 action)' with 'Clone' and 'Remove' links. It contains three sections: 'Service' with 'STS', 'Actions' with 'Write' and 'AssumeRole', and 'Resources' with radio buttons for 'Specific' (selected) and 'All resources'. Below the 'Resources' section is a 'role' field with a dropdown showing 'arn:aws:iam::222233334444:role/Dev' and an 'EDIT' button. There is also an 'Any' checkbox. At the bottom of the policy configuration area is a link 'Add ARN to restrict access'. The footer includes 'Feedback', 'English (US)', copyright information '© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.', and links for 'Privacy Policy' and 'Terms of Use'.

Documentation

Create policy

1 2

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

Visual editor JSON [Import managed policy](#)

[Expand all](#) [Collapse all](#)

▼ STS (1 action) [Clone](#) [Remove](#)

Service	STS
Actions	Write AssumeRole
Resources	<input checked="" type="radio"/> Specific close <input type="radio"/> All resources
role ?	<input type="text" value="arn:aws:iam::222233334444:role/Dev"/> EDIT <input type="checkbox"/> Any

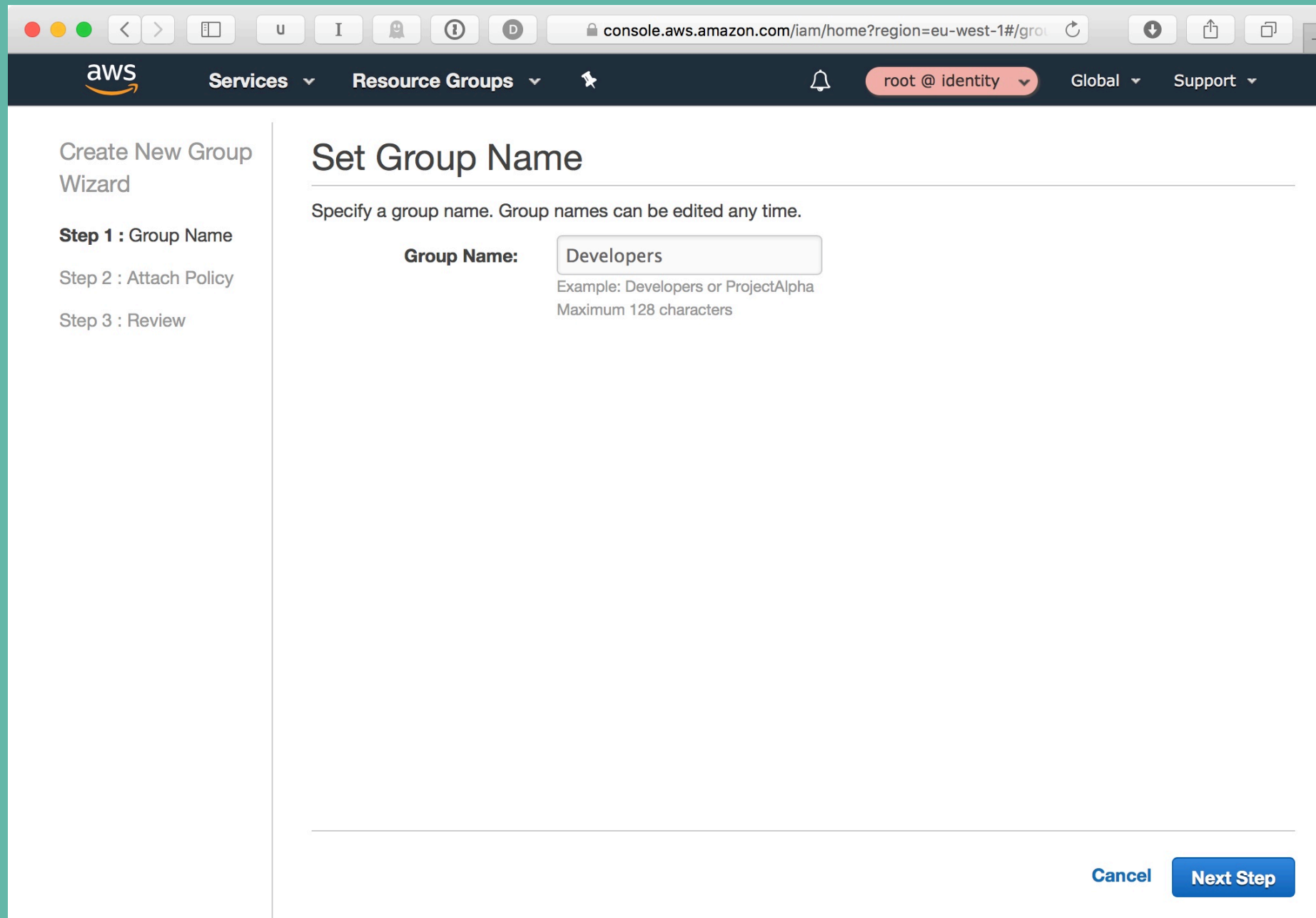
[Add ARN to restrict access](#)

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::<MASTER-ACCOUNT-ID>:role/DeveloperRole"
    }
  ]
}
```

Create a *Developers* group.



The screenshot shows the AWS IAM console interface for creating a new group. The browser address bar shows the URL: `console.aws.amazon.com/iam/home?region=eu-west-1#/groups`. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a user icon, a notification bell, 'root @ identity', 'Global', and 'Support'. On the left sidebar, under 'Create New Group Wizard', the steps are listed: 'Step 1 : Group Name' (selected), 'Step 2 : Attach Policy', and 'Step 3 : Review'. The main content area is titled 'Set Group Name' and contains the instruction: 'Specify a group name. Group names can be edited any time.' Below this, there is a 'Group Name:' label and a text input field containing 'Developers'. A hint below the input field reads: 'Example: Developers or ProjectAlpha' and 'Maximum 128 characters'. At the bottom right of the main content area, there are two buttons: 'Cancel' and 'Next Step'.

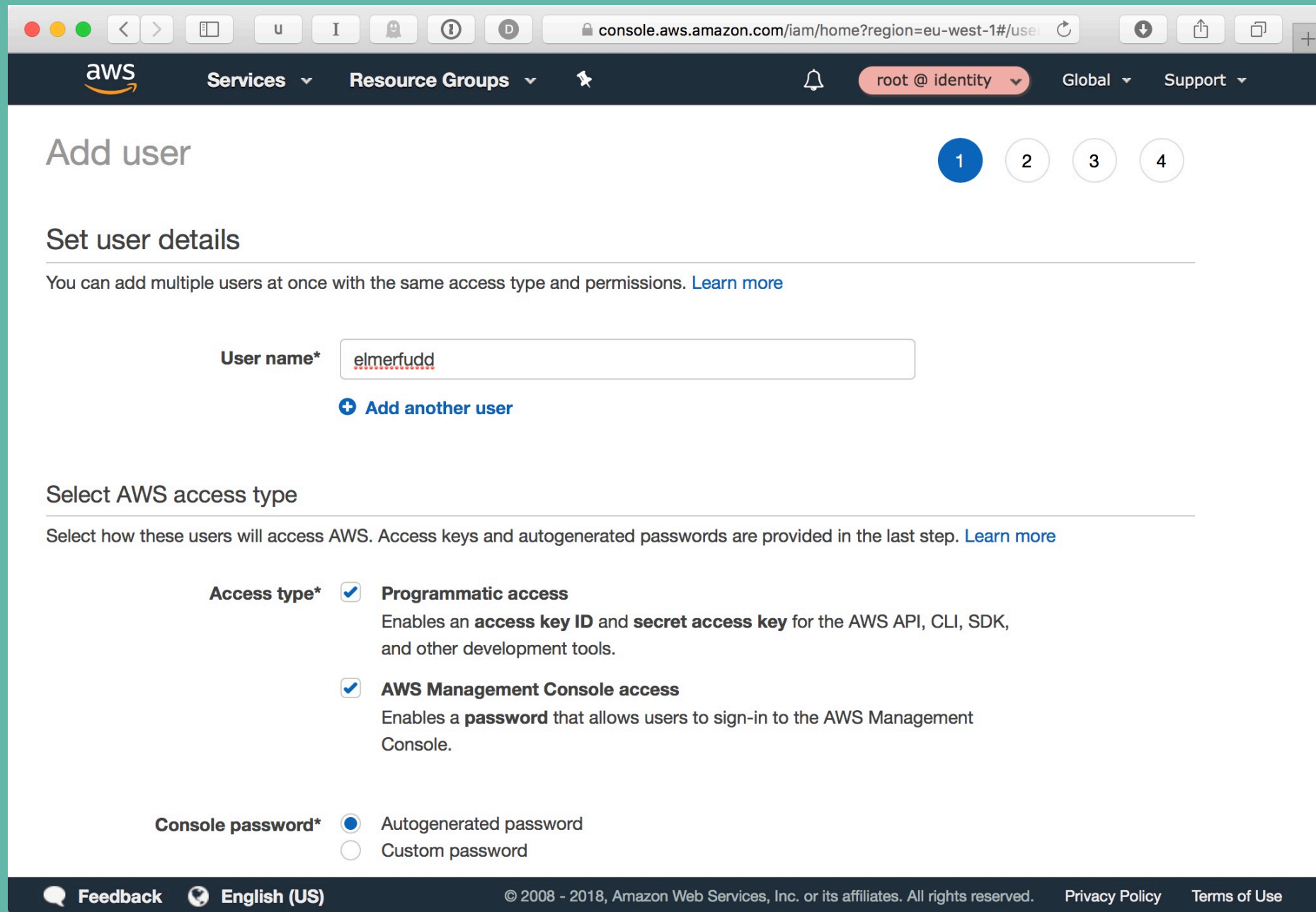
And attach the policy.

The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a user profile 'root @ identity', and links for 'Global' and 'Support'. On the left sidebar, the 'Create New Group Wizard' is shown with three steps: 'Step 1 : Group Name', 'Step 2 : Attach Policy' (which is the active step), and 'Step 3 : Review'. The main content area is titled 'Attach Policy' and contains the instruction: 'Select one or more policies to attach. Each group can have up to 10 policies attached.' Below this is a search bar with the filter 'Policy Type' and a search input containing 'assumemaster', which has yielded 'Showing 1 results'. A table lists the available policies:

	Policy Name ↕	Attached Entities ↕	Creation Time ↕	Edited Time ↕
<input checked="" type="checkbox"/>	AssumeMasterDeve...	0	2018-05-07 10:07 UT...	2018-05-07 10:0...

At the bottom right of the main content area, there are three buttons: 'Cancel', 'Previous', and 'Next Step'.

Create a user.



The screenshot shows the AWS IAM console interface for creating a new user. The browser address bar displays `console.aws.amazon.com/iam/home?region=eu-west-1#/users`. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', a notification bell, the current user 'root @ identity', and links for 'Global' and 'Support'. The main heading is 'Add user', followed by a progress indicator with four steps; step 1 is active. Below this is the 'Set user details' section, which includes a text input for 'User name*' containing 'elmerfudd' and a link to 'Add another user'. The 'Select AWS access type' section follows, with instructions on how users will access AWS. Under 'Access type*', both 'Programmatic access' and 'AWS Management Console access' are selected with checkboxes. The 'Console password*' section shows 'Autogenerated password' selected with a radio button. The footer contains a 'Feedback' link, 'English (US)' language selection, copyright information for 2008-2018, and links for 'Privacy Policy' and 'Terms of Use'.

console.aws.amazon.com/iam/home?region=eu-west-1#/users

aws Services Resource Groups root @ identity Global Support

Add user

1 2 3 4

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* elmerfudd

+ Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☒ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* ☒ Autogenerated password
☐ Custom password

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

And add them to the Developers group.

The screenshot shows the AWS IAM console interface for adding a new user. The breadcrumb trail indicates the user is in the 'Add user' section, specifically at step 2 of 4. The first step, 'Set permissions for elmerfudd', is completed. Three options are presented: 'Add user to group' (selected), 'Copy permissions from existing user', and 'Attach existing policies directly'. Below these, a note explains that adding a user to a group is a best practice. The 'Add user to group' section contains a 'Create group' button and a 'Refresh' button. A search bar with the text 'dev' shows 'Showing 1 result'. The results table lists the 'Developers' group, which is checked, and shows it has the 'AssumeMasterDeveloperRole' policy attached.

console.aws.amazon.com/iam/home?region=eu-west-1#/user

aws Services Resource Groups root @ identity Global Support

Add user

1 2 3 4

Set permissions for elmerfudd

Add user to group

Copy permissions from existing user

Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

Create group Refresh

Showing 1 result

Group	Attached policies
<input checked="" type="checkbox"/> Developers	AssumeMasterDeveloperRole

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Remember this?

```
{  
  "Condition": {  
    "Bool": {  
      "aws:MultiFactorAuthPresent": "true"  
    }  
  }  
}
```

The user must activate an MFA device¹ on their account before they can switch roles.

¹ **IAM User Guide: Allow IAM Users to Self-Manage an MFA Device**

Once signed in to the Bastion account, the user can *switch role* and use the *AWS console* as usual.

aws

Switch Role

Allows management of resources across AWS accounts using a single user ID and password. You can switch roles after an AWS administrator has configured a role and given you the account and role details. [Learn more.](#)

Account* ⓘ

Role* ⓘ

Display Name ⓘ

Color a a a a a a

***Required** [Cancel](#) [Switch Role](#)

English ⓘ

[Terms of Use](#) [Privacy Policy](#) © 1996-2018, Amazon Web Services, Inc. or its affiliates.

CLI setup 🧐

~/.aws/config

```
[profile bastion]
mfa_serial = arn:aws:iam::<BASTION-ACCOUNT-ID>:mfa/elmerfudd

[profile master]
source_profile = bastion
mfa_serial = arn:aws:iam::<BASTION-ACCOUNT-ID>:mfa/elmerfudd
role_arn = arn:aws:iam::<MASTER-ACCOUNT-ID>:role/DeveloperRole
```

~/.aws/config

```
[profile bastion]
mfa_serial = arn:aws:iam::<BASTION-ACCOUNT-ID>:mfa/elmerfudd

[profile master]
source_profile = bastion
mfa_serial = arn:aws:iam::<BASTION-ACCOUNT-ID>:mfa/elmerfudd
role_arn = arn:aws:iam::<MASTER-ACCOUNT-ID>:role/DeveloperRole
```

~/.aws/credentials

```
[bastion]
aws_access_key_id = AKIAI6ZCA7DBKEXAMPLE
aws_secret_access_key = A94L...

# No credentials for the master profile,
# it inherits from bastion.
```


Cool, but we still have credentials in clear text on disk.

Enter **aws-vault**.

AWS Vault stores IAM credentials in your operating system's secure keystore and then generates temporary credentials from those to expose to your shell and applications.

Add credentials for the *bastion* profile to aws-vault.

```
$ aws-vault add bastion
```

```
Enter Access Key ID: AKIAI6ZCA7DBKEXAMPLE
```

```
Enter Secret Access Key: A94L...
```

```
Added credentials to profile "bastion" in vault
```

And remove them from ~/.aws/credentials.

```
--- a/credentials
+++ b/credentials
@@ -1,3 +0,0 @@
-[bastion]
-aws_access_key_id = AKIAI6ZCA7DBKEXAMPLE
-aws_secret_access_key = A94L...
```

Use `aws-vault exec` to generate temporary credentials.

```
$ aws-vault exec bastion -- aws sts get-caller-identity
Enter token for arn:aws:iam::111122223333:mfa/elmerfudd: 123456
{
  "UserId": "AIDAIMBDIHHP70EXAMPLE",
  "Account": "111122223333",
  "Arn": "arn:aws:iam::111122223333:user/elmerfudd"
}
```

Use `aws-vault exec` to generate temporary credentials.

```
$ aws-vault exec master -- aws sts get-caller-identity
Enter token for arn:aws:iam::111122223333:mfa/elmerfudd: 123456
{
  "UserId": "AROAI4MZALGSVCEXAMPLE:1525697128262099000",
  "Account": "222233334444",
  "Arn": "arn:aws:sts::222233334444:assumed-role/DeveloperRole/1525697128262099000"
}
```

Use `aws-vault rotate` to rotate your credentials.

```
$ aws-vault rotate bastion
```

Rotating credentials `for` profile `"bastion"` (takes 10-20 seconds)

Done!

The credentials you had in clear text on disk have now been destroyed and new credentials have been stored in your OS keychain.

Not even you know your credentials now.

What else can aws-vault do?

List your profiles and their credentials sources with `aws-vault list`

```
$ aws-vault list
```

Profile	Credentials	Sessions
=====	=====	=====
bastion	bastion	1525711415
master	bastion	1525711528

Run a local EC2 instance metadata server on your machine with `aws-vault` server.

Useful if your are developing apps that use an AWS SDK that reads credentials from instance metadata.

I could not get this to work 😞

Last but not least, aws-vault login.

```
$ aws-vault login master
```

```
Enter token for arn:aws:iam::111122223333:mfa/elmerfudd: 123456
```

**This opens the AWS console in your web browser,
signed in with temporary credentials and the correct
role assumed.**

It's magic.

Takeaways

- **Multiple AWS accounts can help you separate concerns and make your AWS bills a little clearer.**
- **A bastion AWS account can give you an extra layer of security and make it easier to manage users across many accounts.**
- **aws-vault makes it easier and safer to use the AWS CLI with one or many AWS accounts.**

Questions?

aws-vault: <https://github.com/99designs/aws-vault>

Slides: <https://github.com/masv/multi-account-with-aws-vault>

E-mail: martin@forzafootball.com