

Managed Cloud Services

When you don't want to run it yourself

Managed Docker Repository

Elastic Container Service Repository (ECS Repository)

ECS Repository

Store our Docker Images in the Cloud

- What if we want to store our built docker image somewhere other than our laptop?
- What if we don't want our image to be “public” on hub.docker.com?
- AWS has a managed Docker Image Repository: ECS Repository

ECS Repository

- Get into your AWS account
- Search for “ECS”

The screenshot shows the AWS Elastic Container Service (ECS) console interface. At the top, there's a navigation bar with tabs for 'Services' and a search bar containing the text 'ecs'. Below the search bar, a sidebar on the left lists various AWS services: 'Amazon ECS', 'Clusters', 'Task Definitions', 'Account Settings', 'Amazon EKS', 'Clusters', 'Amazon ECR', 'Repositories', 'AWS Marketplace', 'Discover software', and 'Subscriptions'. To the right of the sidebar, the main content area displays search results for 'ecs'. The results are categorized into 'Services' (21), 'Features' (58), and 'Blogs' (2,425). The 'Services' section highlights 'Elastic Container Service' as a top feature, describing it as a 'Highly secure, reliable, and scalable way to run containers'. Other listed services include 'Batch', 'EFS', and 'EC2'. The 'Features' section highlights 'Clusters' and 'Task definitions'.

Search results for 'ecs'

Services

- Elastic Container Service** ★
Highly secure, reliable, and scalable way to run containers
- Top features**
Clusters Task definitions
- Batch** ★
Fully managed batch processing at any scale
- EFS** ★
Managed File Storage for EC2
- EC2** ★
Virtual Servers in the Cloud

Features

- Clusters**
Elastic Container Service feature
- Task definitions**

<https://console.aws.amazon.com/ecs/home?region=us-east-1> [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Print](#)

ECS Repository

- Get into your AWS account
- Search for “ECS”

The screenshot shows the AWS Elastic Container Registry (ECR) interface. The top navigation bar includes tabs for Services, Elastic Container Service, Elastic Container Registry, CloudFormation, VPC, EC2, and RDS. The main content area is titled "Amazon ECR > Repositories". It features two tabs: "Private" (which is selected) and "Public". Below these tabs is a search bar with the placeholder "Find repositories". A table header is present with columns for "Repository name", "URI", and "Created at". A message "No repositories" is displayed below the table, followed by the sub-message "No repositories were found". On the right side of the table, there are buttons for "View push commands", "Delete", "Actions", and a large orange "Create repository" button.

ECS Repository

- Create a private repository

The screenshot shows the AWS Elastic Container Registry (ECR) interface. At the top, there are tabs for 'Learner Lab' and 'Elastic Container Registry'. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/ecr/create-repository?region=us-east-1>. The navigation bar includes links for 'Services', 'Search for services [Option+S]', 'Amazon ECR', 'Repositories', and 'Create repository'. Below the navigation, the breadcrumb path is 'Amazon ECR > Repositories > Create repository'. The main section is titled 'Create repository' and contains a 'General settings' panel. Under 'Visibility settings', the 'Private' option is selected, with the note: 'Access is managed by IAM and repository policy permissions.' The 'Public' option is also available. A 'Repository name' input field contains the value '561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app'. A note below the input field states: '15 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.' Under 'Tag immutability', the 'Disabled' option is selected, with the note: 'Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.' A warning message at the bottom states: 'Once a repository is created, the visibility setting of the repository can't be changed.' At the bottom of the page, there are links for 'Feedback', 'Unified Settings', 'Privacy', 'Terms', and 'Cookie preferences'.

Once a repository is created, the visibility setting of the repository can't be changed.

ECS Repository

- Create a private repository
- Now we can push docker images from our laptop to this repository
- From there, we can pull them down to an EC2 instance, or to Elastic Container Service to run

The screenshot shows the AWS Elastic Container Registry (ECR) interface. The top navigation bar includes tabs for 'Services' (highlighted), 'Elastic Container Service', 'Elastic Container Registry', 'CloudFormation', 'VPC', 'EC2', and 'RDS'. The main heading is 'Amazon ECR > Repositories'. Below this, there are two tabs: 'Private' (selected) and 'Public'. A search bar at the top right contains the placeholder 'Find repositories'. The main content area displays a table titled 'Private repositories (1 of 1)'. The table has columns for 'Repository name', 'URI', and 'Created at'. One repository is listed: 'csc346-chat-app' with URI '561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app' and creation date 'October 27, 2022, 20:03:55 (UTC-07)'. Action buttons include 'View push commands', 'Delete', 'Actions ▾', and 'Create repository'.

Repository name	URI	Created at
csc346-chat-app	561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app	October 27, 2022, 20:03:55 (UTC-07)

Feedback Looking for language selection? Find it in the new Unified Settings [\[link\]](#)

© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy Terms Cookie preferences

ECS Repository

- Create a private repository
- Now we can push docker images from our laptop to this repository
- From there, we can pull them down to an EC2 instance, or to Elastic Container Service to run
- View the push commands

The screenshot shows a web browser window titled "Elastic Container Registry" with the URL <https://us-east-1.console.aws.amazon.com/ecr/repositories?region=us-east-1>. The main content is titled "Push commands for csc346-chat-app". It provides push commands for both macOS/Linux and Windows. The macOS/Linux section is active, showing the command:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin  
561707296892.dkr.ecr.us-east-1.amazonaws.com
```

A note below the command states: "Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed." The Windows section is also present.

The page also lists four steps for pushing the image:

1. Retrieve an authentication token and authenticate your Docker client to your registry.
Use the AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin  
561707296892.dkr.ecr.us-east-1.amazonaws.com
```
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

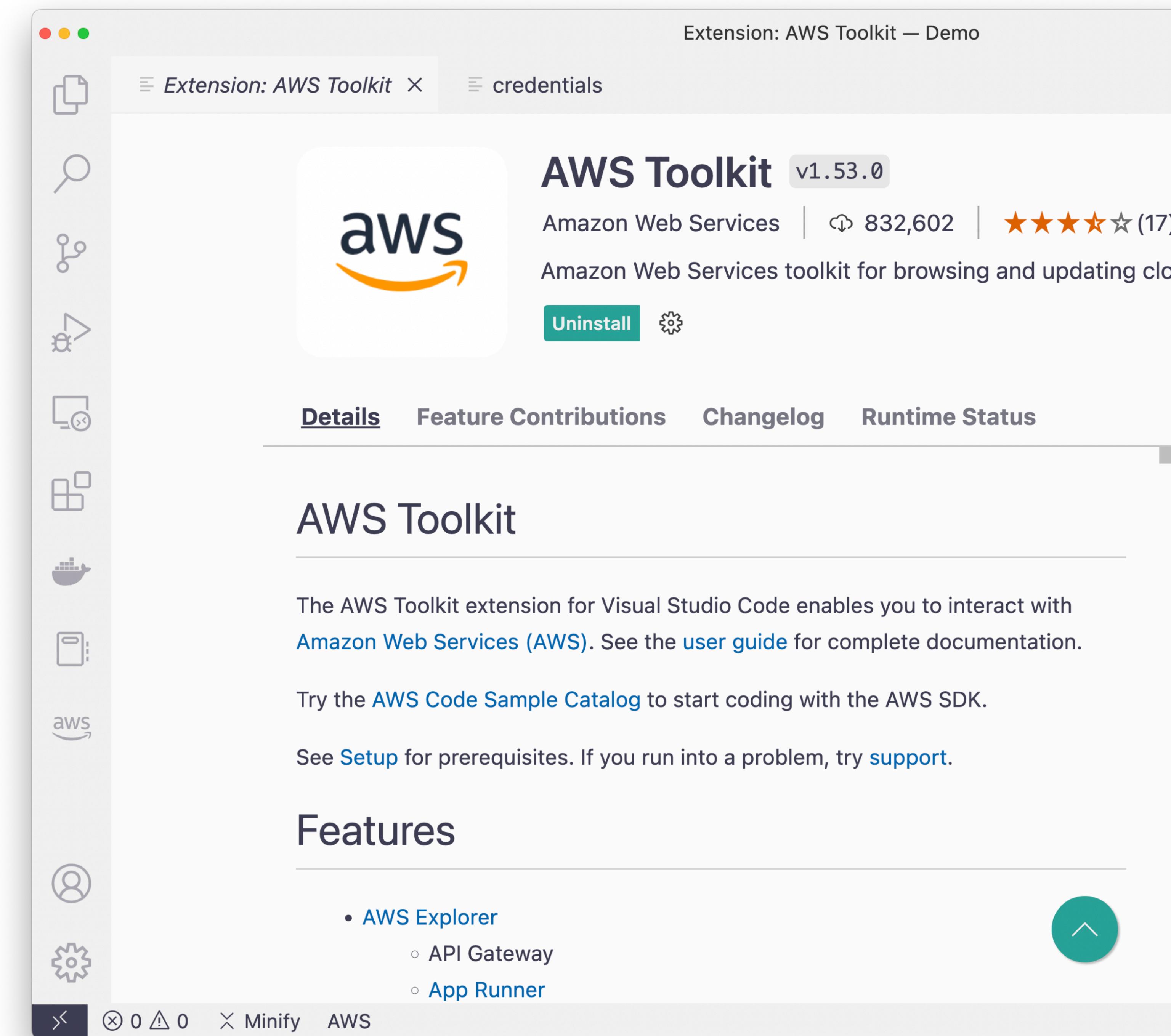
```
docker build -t csc346-chat-app .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag csc346-chat-app:latest 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
```

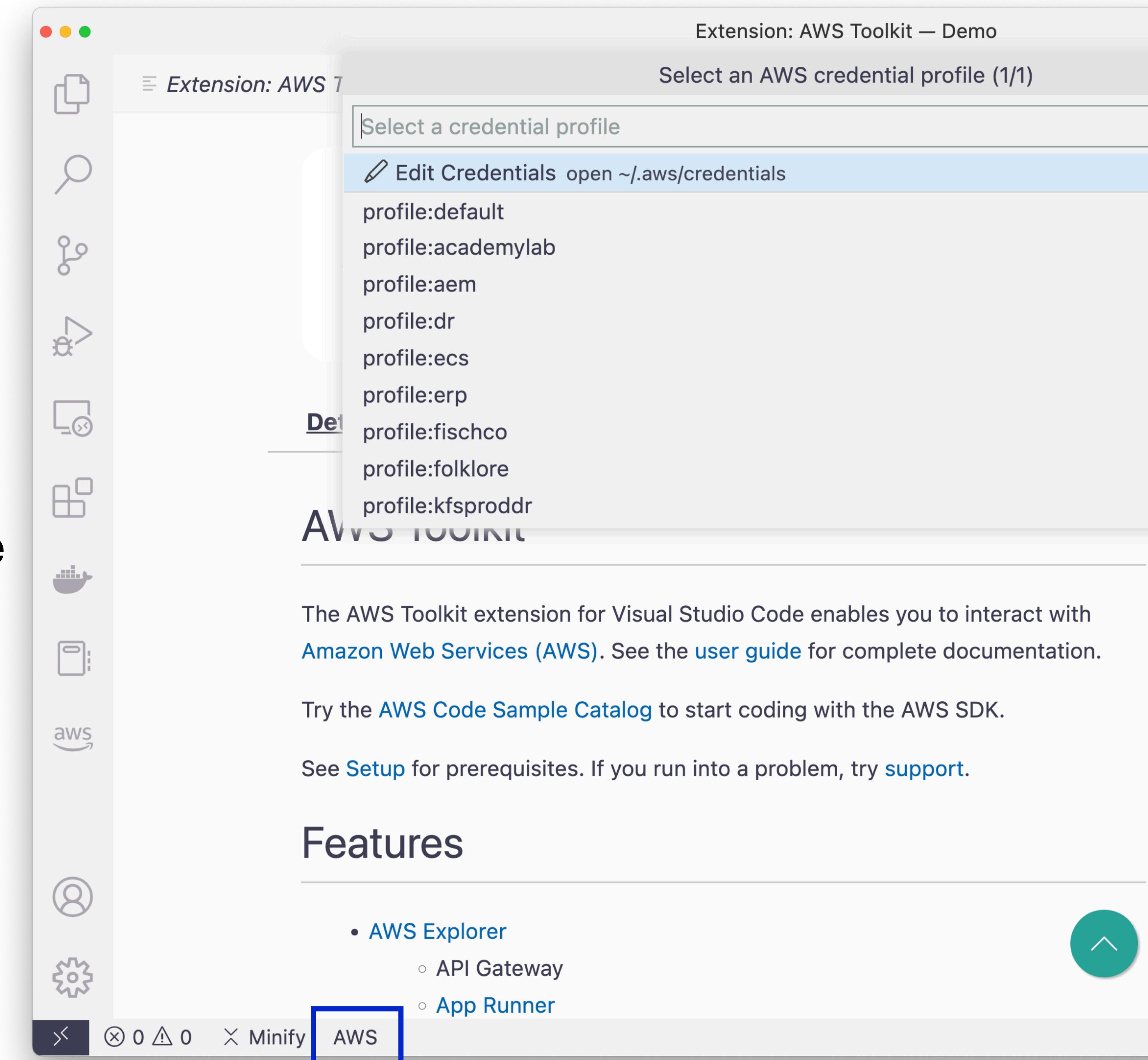
ECS Repository

- There's a really great “AWS Toolkit” extension for VS Code that Amazon supports



ECS Repository

- There's a really great “AWS Toolkit” extension for VS Code that Amazon supports
- Clicking on the “AWS” in the window footer will bring up the AWS commands
- Easily access your credentials file



ECS Repository

- In order to push images to ECR, you need to have current AWS IAM credentials
- Copy them from the AWS Academy site and update your credentials file

The screenshot shows the AWS Toolkit extension for Visual Studio Code. The Explorer pane on the left displays a project structure for a 'DEMO' folder, containing 'html', 'Dockerfile', and 'run.sh' files. The Timeline section shows a message: 'No timeline information was provided.' The Terminal pane on the right is titled 'credentials — Demo' and contains the following AWS credentials:

```
1 [default]
2 aws_access_key_id=ASIAYFSC5FB6MTPG4T7S
3 aws_secret_access_key=LKgbseayEbS5B1Ry1M4cT1ReKkDzH4
4 aws_session_token=FwoGZXIvYXdzEEwaDD5ISLm1LGZ9ujTqsS
5
6
7
8
9
10
11
```

The Terminal tab is currently selected, showing the command prompt: ~ /Demo \$.

ECS Repository

- Build your image

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a project structure with a folder named "DEMO" containing "html", "Dockerfile", and "run.sh".
- Dockerfile Editor:** The "Dockerfile" tab is active, displaying the following code:

```
1 FROM httpd:2.4-alpine
2 COPY ./html /usr/local/apache2/htdocs/
3
```
- Terminal View:** The terminal tab shows the output of a Docker build command:
 - Line 1: `~/.Demo $ docker build -t csc346-chat-app .`
 - Line 2: `[+] Building 0.1s (7/7) FINISHED`
 - Line 3: `=> [internal] load build definition from Dockerfile`
 - Line 4: `=> => transferring dockerfile: 103B`
 - Line 5: `=> [internal] load .dockerignore`
 - Line 6: `=> => transferring context: 2B`
 - Line 7: `=> [internal] load metadata for docker.io/library/httpd:2.4-alpi`
 - Line 8: `=> [internal] load build context`
 - Line 9: `=> => transferring context: 32.38kB`
 - Line 10: `=> CACHED [1/2] FROM docker.io/library/httpd:2.4-alpine`
 - Line 11: `=> [2/2] COPY ./html /usr/local/apache2/htdocs/`
 - Line 12: `=> exporting to image`
 - Line 13: `=> => exporting layers`
 - Line 14: `=> => writing image sha256:14d36ba18ca2868860521dfbdd499d155dd14`
 - Line 15: `=> => naming to docker.io/library/csc346-chat-app`
- Bottom Status Bar:** Shows icons for Minify, AWS, and other status information.

ECS Repository

- Build your image
- Login to ECR
-

The screenshot shows a code editor interface with the title "Dockerfile — Demo". On the left is the Explorer sidebar, which lists a folder named "DEMO" containing "html", "Dockerfile", and "run.sh". The "Dockerfile" file is currently selected. The main area displays the contents of the Dockerfile:

```
FROM httpd:2.4-alpine
COPY ./html /usr/local/apache2/htdocs/
```

Below the code editor is a terminal window titled "TERMINAL". It shows the command `~/Demo $ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin` being run, followed by the output "561707296892.dkr.ecr.us-east-1.amazonaws.com" and "Login Succeeded". The terminal also includes a note about logging in with a password and a link to [docker.com/go/access-tokens/](https://docs.docker.com/go/access-tokens/).

ECS Repository

- Build your image
- Login to ECR
- Tag your local image with the ECR host name that matches your repository
 - This is what tells the docker push **command** where to send your image

The screenshot shows the AWS Lambda CodeWhisperer extension integrated into the Visual Studio Code interface. The Explorer sidebar on the left lists a project named 'DEMO' containing files 'html', 'Dockerfile', and 'run.sh'. The 'Dockerfile' tab is active, displaying the following code:

```
FROM httpd:2.4-alpine
COPY ./html /usr/local/apache2/htdocs/
```

Below the code editor, the terminal window shows the command used to tag the Docker image:

```
~/Demo $ docker tag csc346-chat-app:latest 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
```

The terminal also displays a reference log entry:

- ~/Demo \$ docker tag csc346-chat-app:latest 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
- ~/Demo \$

The status bar at the bottom indicates the current line and column: Ln 3, Col 1.

ECS Repository

- Build your image
- Login to ECR
- Tag your local image with the ECR host name that matches your repository
- Push your image up to ECR

The screenshot shows the AWS Lambda CodeWhisperer extension integrated into the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with a 'DEMO' folder containing 'html', 'Dockerfile', and 'run.sh'. The 'Dockerfile' tab in the main editor area displays the following code:

```
FROM httpd:2.4-alpine
COPY ./html /usr/local/apache2/htdocs/
```

Below the editor, a terminal window shows the command-line history for pushing the Docker image to ECR:

- ~/Demo \$ docker tag csc346-chat-app:latest 561707296892.dkr.ecr.us-east-1.amazonaws.com:latest
- ~/Demo \$ docker push 561707296892.dkr.ecr.us-east-1.amazonaws.com:latest
- The push refers to repository [561707296892.dkr.ecr.us-east-1.amazonaws.com:latest]
- e59a6fd22816: Pushed
- 3519fde520d1: Pushed
- 8fe52be198ba: Pushed
- 60a823ff2ab1: Pushed
- 182eb4edc47e: Pushed
- 87e6e9d87ec7: Pushed
- 5d3e392a13a0: Pushed
- latest: digest: sha256:c11f233b6856b7d24c1c5e99c1af39c39f46c18bf9

A large black box at the bottom contains the command:

```
docker push 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
```

The bottom status bar shows the AWS extension icon, a timeline icon, and other status indicators.

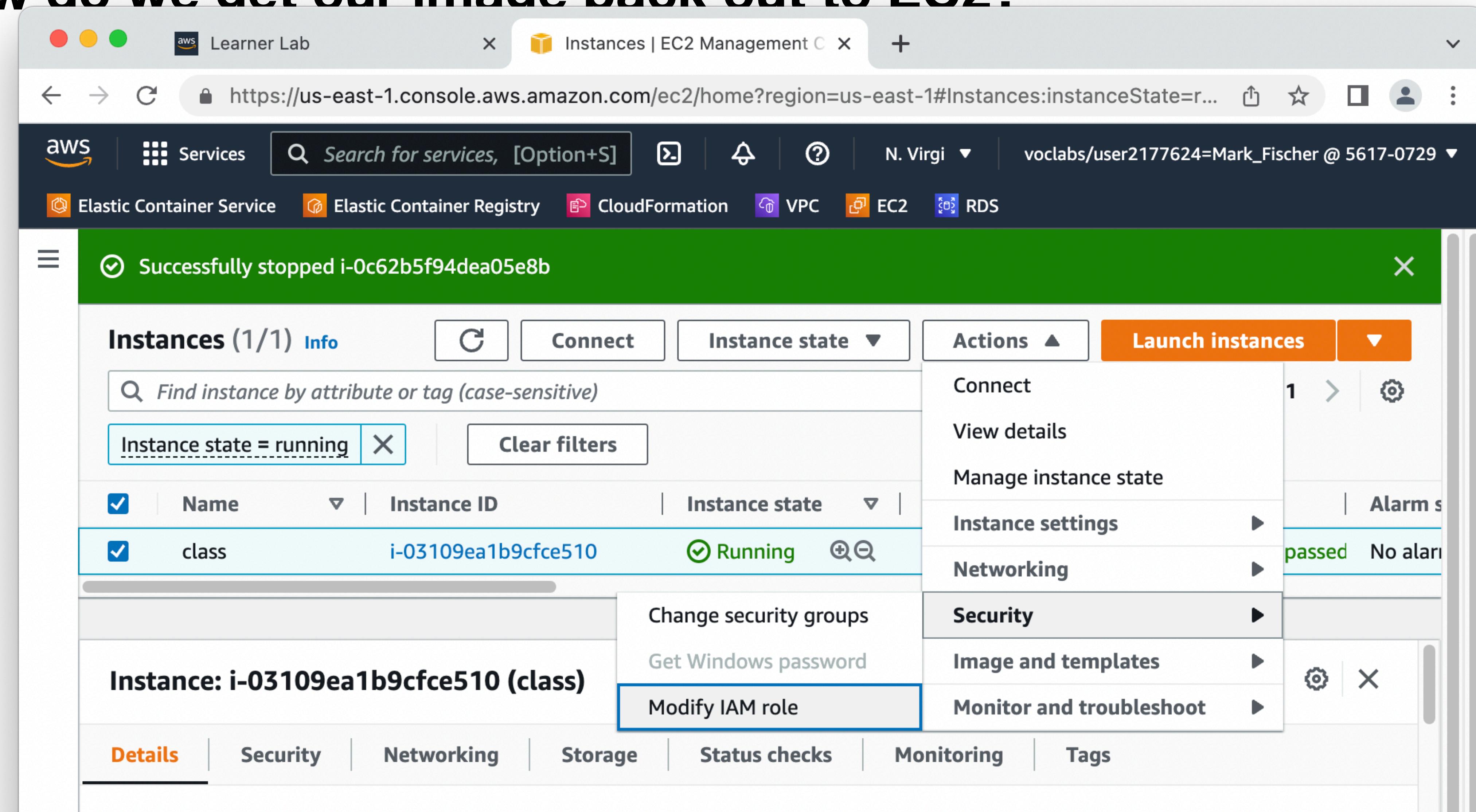
ECS Repository

How do we get our image back out to EC2?

- We still need permissions on our EC2 instance to pull an image back down
- We could copy IAM credentials to our EC2 host just like we do for our laptop
- However within AWS you can leverage IAM Roles
 - A role defines a set of permissions that an actor can take on resources
 - We can attach an Role Profile to our instance

ECS Repository

How do we get our image back out to EC2?



EC2 How

Learner Lab Modify IAM role | EC2 Manager

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyIAMRole:instanceId... Cancel Update IAM role

aws Services Search for services, [Option+S] + N. Virgi voclabs/user2177624=Mark_Fischer @ 5617-0729

Elastic Container Service Elastic Container Registry CloudFormation VPC EC2 RDS

Successfully stopped i-0c62b5f94dea05e8b X

EC2 > Instances > i-03109ea1b9cfce510 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID i-03109ea1b9cfce510

lab

LabInstanceProfile
arn:aws:iam::561707296892:instance-profile/**LabInstanceProfile**

Choose IAM role ▲ Create new IAM role

The role you select replaces any roles that are attached to this instance.

⚠️ If you choose **No IAM Role**, any IAM role that is currently attached to the instance will be removed. Are you sure you want to remove from the selected instance?

ECS Repository

- With an IAM role attached we can now do our docker login on the EC2 instance

The screenshot shows the VS Code interface running on an EC2 instance. The terminal tab is active, displaying the following command and its output:

```
[ec2-user@ip-172-31-84-94 ~]$ aws ecr get-login-password --region us-east-1
username AWS --password-stdin 561707296892.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

The terminal also shows a success message:

```
Login Succeeded
```

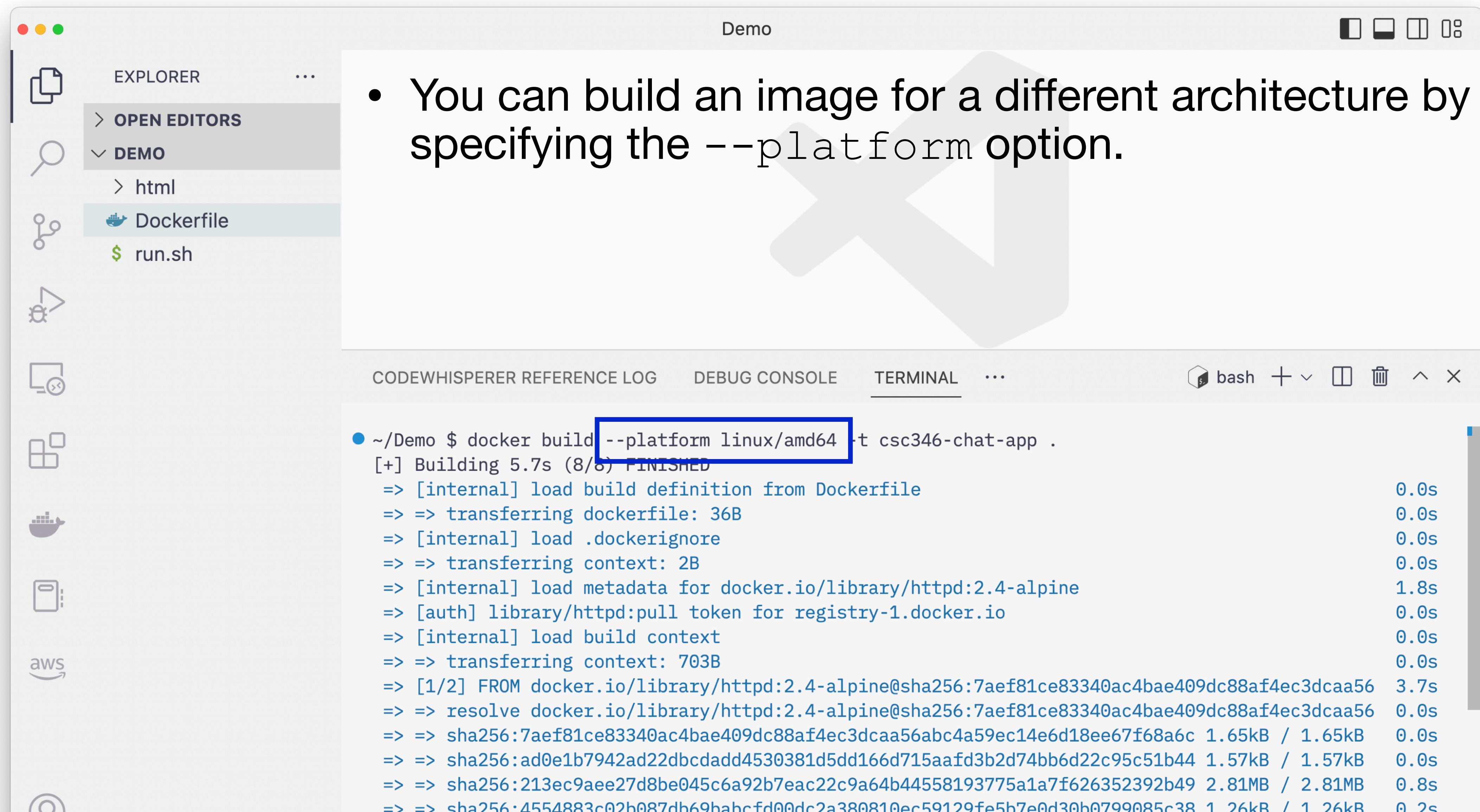
The status bar at the bottom indicates the connection is via SSH to 35.173.191.131.

ECS Repository

The screenshot shows a terminal window titled "ec2-user [SSH: 35.173.191.131]" with a large red hammer icon overlaid on the center of the screen. On the left is a vertical toolbar with icons for file, search, share, and other developer tools. Below the toolbar are tabs for PORTS, DEBUG CONSOLE, TERMINAL (which is selected), PROBLEMS, and OUTPUT. At the bottom is a command-line interface.

```
[ec2-user@ip-172-31-84-94 ~]$ sudo docker run -d --rm -p80:80 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
st
WARNING: The requested image's platform (linux/arm64/v8) does not match the detected host platform (linux/amd64) and no specific platform was requested
396047640ee76144276c4d2b2f309145edbf60b82d6c078c2ca7a3bb70690827
[ec2-user@ip-172-31-84-94 ~]$
```

ECS Repository



ECS Repository

- Build, tag, push the updated image
- Now we can run the image on our EC2 instance directly from the ECR repository

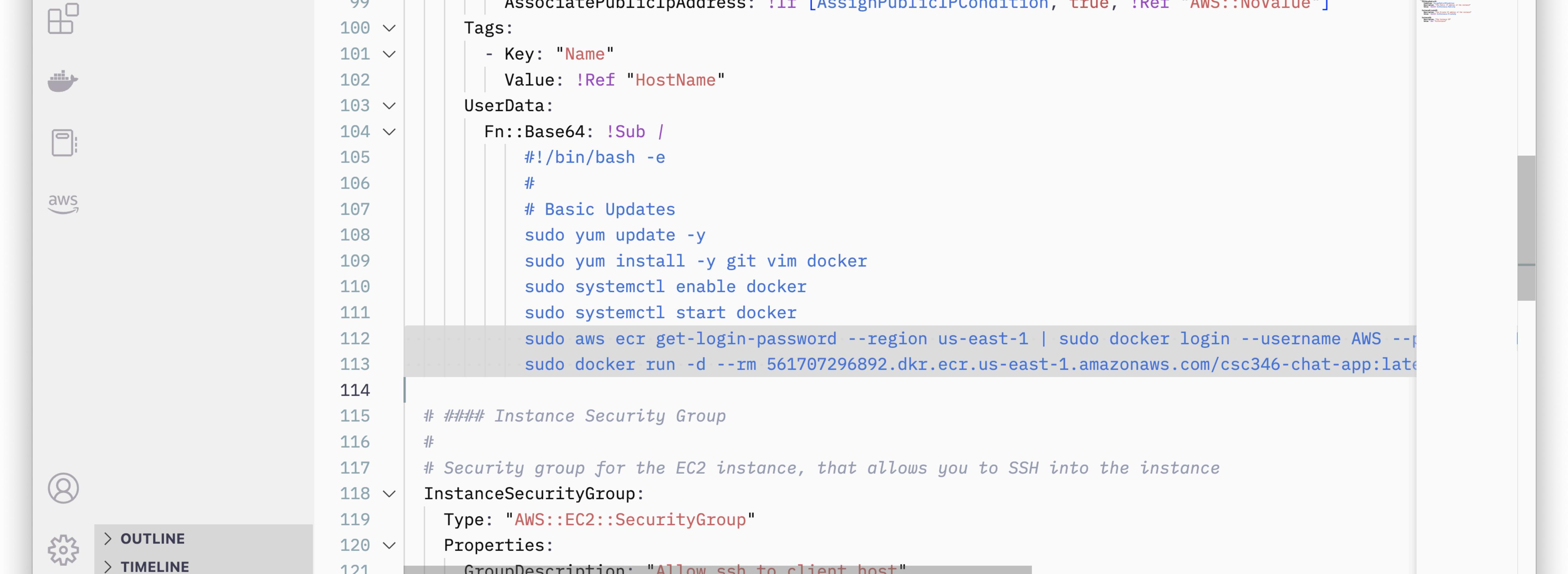
The screenshot shows a terminal window with a sidebar containing various icons. The terminal itself has tabs for PORTS, DEBUG CONSOLE, TERMINAL (which is selected), PROBLEMS, and OUTPUT. The terminal output is as follows:

```
[ec2-user@ip-172-31-84-94 ~]$ sudo docker run -d --rm -p80:80 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
Unable to find image '561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest' latest: Pulling from csc346-chat-app
213ec9aee27d: Already exists
4554883c02b0: Pull complete
b43fba6eeb4f: Pull complete
014f3125c597: Pull complete
8f50db9e03b2: Pull complete
8682bf7183ff: Pull complete
6df563375ab3: Pull complete
Digest: sha256:d46c91fbfd784d6665a98497d8e9cde9b3a47d4629137c50306f44cde991d57
Status: Downloaded newer image for 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
4737519e18885632a0741bdcc27dfaf5459dda3dc13c530002f88bf92ddda9ff
[ec2-user@ip-172-31-84-94 ~]$
```

At the bottom of the terminal window, there is a footer bar with the text "SSH: 35.173.191.131" and some status indicators.

More Automation

- Combine with CloudFormation to automatically login and start the image at boot time



The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar with icons for stacks, AWS Lambda, CloudWatch Metrics, and CloudWatch Logs. Below that, there are links for 'aws' and a user icon. At the bottom, there are buttons for 'OUTLINE' and 'TIMELINE'. The main area displays a CloudFormation template with the following content:

```
99   AssociatePublicIpAddress: !If [AssignPublicIPCondition, true, !Ref "AWS::NoValue"]
100  Tags:
101    - Key: "Name"
102      Value: !Ref "HostName"
103  UserData:
104    Fn::Base64: !Sub /
105      #!/bin/bash -e
106      #
107      # Basic Updates
108      sudo yum update -y
109      sudo yum install -y git vim docker
110      sudo systemctl enable docker
111      sudo systemctl start docker
112      sudo aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin
113      sudo docker run -d --rm 561707296892.dkr.ecr.us-east-1.amazonaws.com/csc346-chat-app:latest
114
115      # ##### Instance Security Group
116      #
117      # Security group for the EC2 instance, that allows you to SSH into the instance
118  InstanceSecurityGroup:
119    Type: "AWS::EC2::SecurityGroup"
120    Properties:
121      GroupDescription: "Allow ssh to client host"
```

