

Lab

Deploy your app to AWS

Software Studio

DataLab, CS, NTHU

What is AWS



What kind of service AWS provide

Compute EC2 Lightsail ↗ Lambda Batch Elastic Beanstalk Serverless Application Repository AWS Outposts EC2 Image Builder	Blockchain Amazon Managed Blockchain	Analytics Athena EMR CloudSearch Elasticsearch Service Kinesis QuickSight ↗ Data Pipeline AWS Data Exchange AWS Glue AWS Lake Formation MSK	End User Computing WorkSpaces AppStream 2.0 WorkDocs WorkLink
Storage S3 EFS FSx S3 Glacier Storage Gateway AWS Backup	Satellite Ground Station	Management & Governance AWS Organizations CloudWatch AWS Auto Scaling CloudFormation CloudTrail Config OpsWorks Service Catalog Systems Manager	Internet Of Things IoT Core FreeRTOS IoT 1-Click IoT Analytics IoT Device Defender IoT Device Management IoT Events IoT Greengrass IoT SiteWise IoT Things Graph
Database RDS DynamoDB ElastiCache Neptune Amazon Redshift Amazon QLDB Amazon DocumentDB Managed Cassandra Service	Quantum Technologies Amazon Braket ↗	Security, Identity, & Compliance IAM Resource Access Manager Cognito Secrets Manager GuardDuty Inspector Amazon Macie ↗ AWS Single Sign-On Certificate Manager Key Management Service CloudHSM Directory Service WAF & Shield AWS Firewall Manager Artifact Security Hub	Game Development Amazon GameLift
			Containers Elastic Container Registry Elastic Container Service Elastic Kubernetes Service

IAM

- Identity and Access Management
 - Enables you to manage access to AWS services and resources securely.
 - Create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

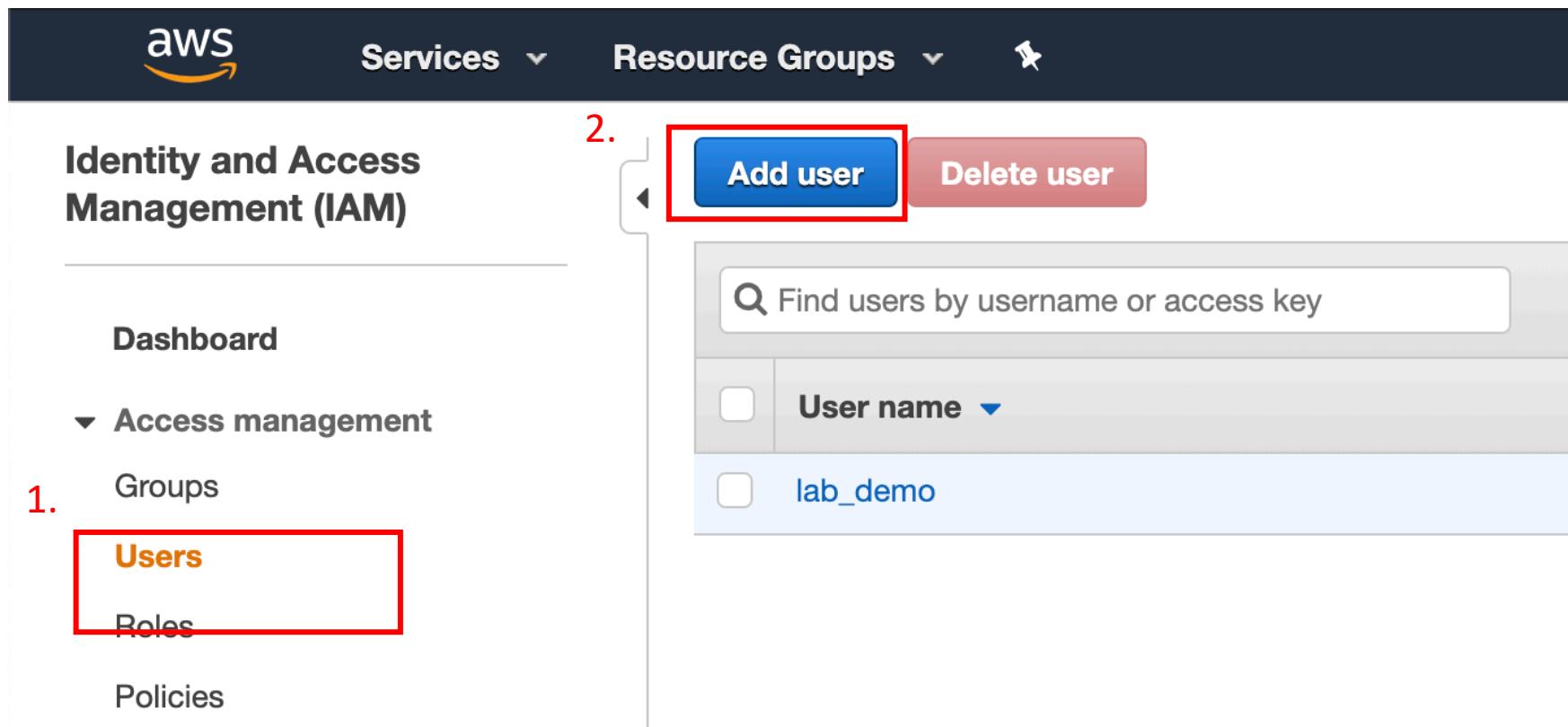
IAM

- Find IAM Services

The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with the AWS logo, a "Services" dropdown, "Resource Groups" dropdown, and a user profile section for "alanlin" in "Oregon". Below the navigation is the title "AWS Management Console". On the left, there's a sidebar titled "AWS services" with a search bar containing "IAM". A list of services is shown, with "IAM" selected. Other recently visited services include Billing, EC2, RDS, Elastic Beanstalk, and another IAM entry. Below the sidebar, there's a "Build a solution" section with options like "Launch a virtual machine", "Build a web app", "Build using virtual servers", "Register a domain", "Connect an IoT device", and "Start migrating to AWS". To the right, there are two main columns: "Access resources on the go" (with a link to the AWS Console Mobile App) and "Explore AWS" (with sections for Amazon DocumentDB, AMD Powered EC2 Instances, EMR Migration Guide, AWS IQ, and Have feedback?).

IAM

- Click User on the dashboard
- Click Add user button



IAM

- Check “programmatic access” and ‘AWS Management Console access’.
- Uncheck “Require password reset” if you want to keep the custom password.
- Finish the form then click Next

Add user

1 2 3 4 5

Set user details

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

User name* lab_demo_2

[+ 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

.....

Show password

Require password reset User must create a new password at next sign-in

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

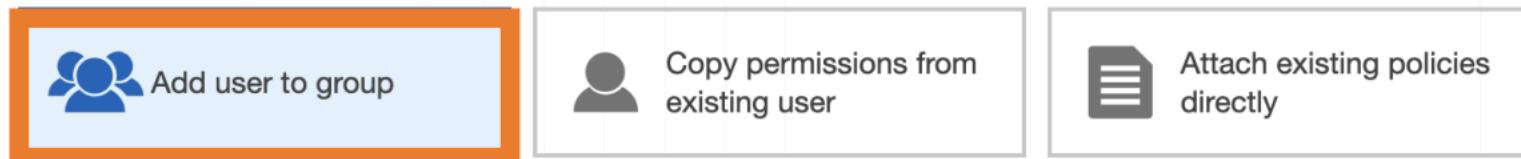
IAM

- Create group

Add user

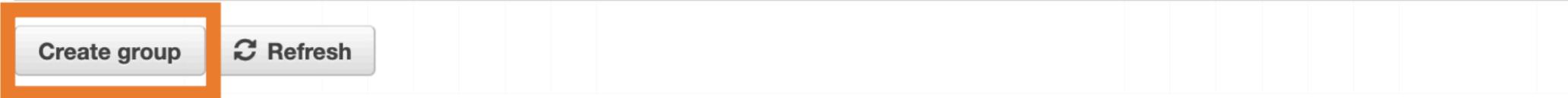
1 2 3 4 5

▼ Set permissions



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



IAM

- Search “AWSElasticBeanstalkFullAccess”

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name lab_demo

[Create policy](#) [Refresh](#)

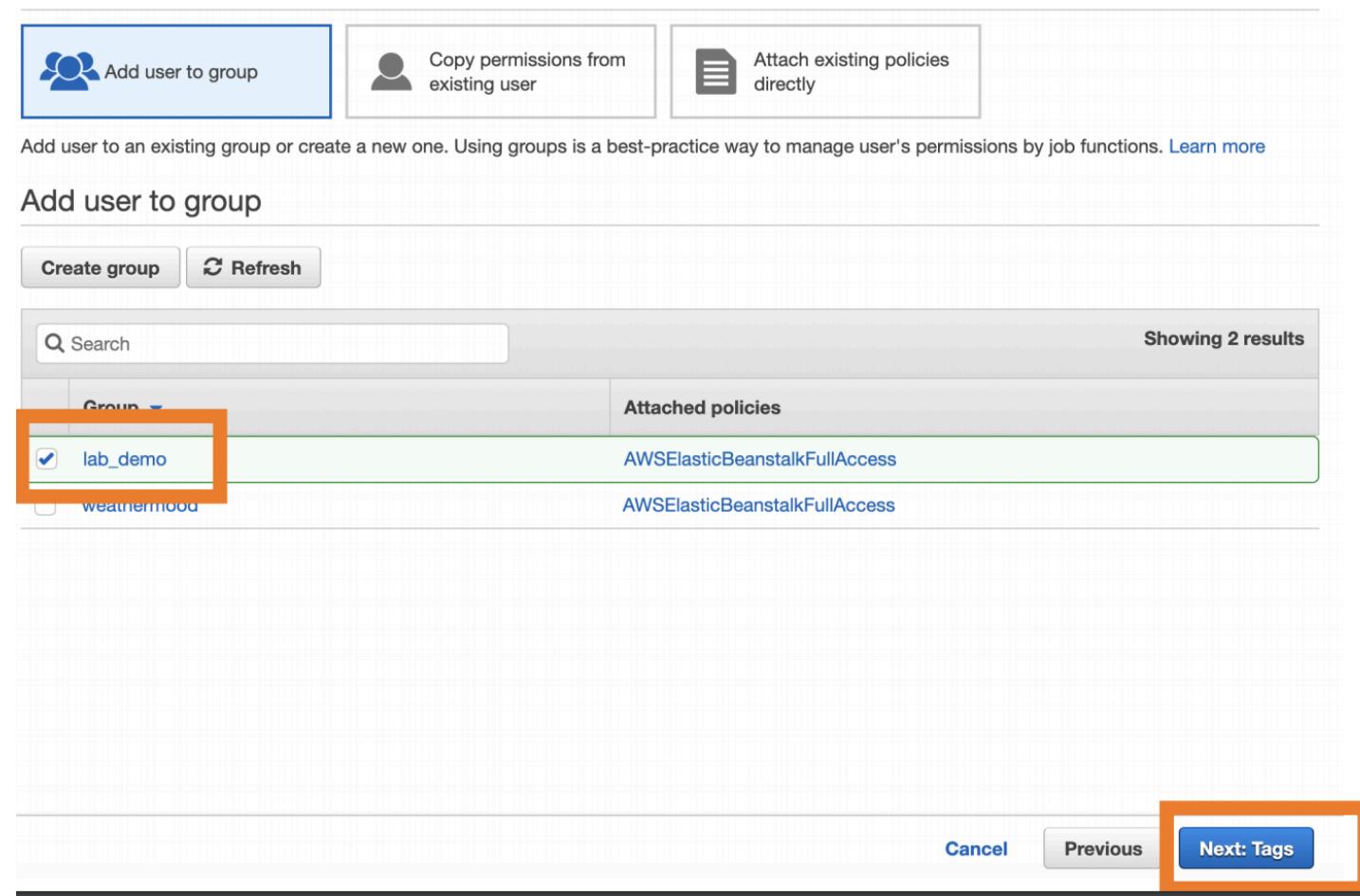
[Filter policies](#) Showing 8 results

	Policy name	Type	Used as	Description
<input type="checkbox"/>	AWSElasticBeanstalkCustomPlatformforE...	AWS managed	None	Provide the instance in your custom platform builder environment permission to launch EC2 inst...
<input type="checkbox"/>	AWSElasticBeanstalkEnhancedHealth	AWS managed	Permissions policy (1)	AWS Elastic Beanstalk Service policy for Health Monitoring system
<input checked="" type="checkbox"/>	AWSElasticBeanstalkFullAccess	AWS managed	Permissions policy (1)	Provides full access to AWS Elastic Beanstalk and underlying services that it requires such as S...
<input type="checkbox"/>	AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy (1)	Provide the instances in your multicontainer Docker environment access to use the Amazon EC...
<input type="checkbox"/>	AWSElasticBeanstalkReadOnlyAccess	AWS managed	None	Provides read only access to AWS Elastic Beanstalk via the AWS Management Console.
<input type="checkbox"/>	AWSElasticBeanstalkService	AWS managed	Permissions policy (1)	AWS Elastic Beanstalk Service role policy which grants permissions to create & manage resourc...
<input type="checkbox"/>	AWSElasticBeanstalkWebTier	AWS managed	Permissions policy (1)	Provide the instances in your web server environment access to upload log files to Amazon S3.

[Cancel](#) [Create group](#)

IAM

- Add user to the group you just create
- Click Next



IAM

- Tags is optional
- You can use the tags to organize, track, or control access for this user

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	<input type="button" value="Remove"/>

You can add 50 more tags.

Cancel Previous **Next: Review**

IAM

- Create User

Add user

1 2 3 4 5

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	lab_demo
AWS access type	Programmatic access and AWS Management Console access
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	lab_demo

Tags

No tags were added.

Cancel

Previous

Create user

IAM

- Download .csv file.

Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://124817121290.signin.aws.amazon.com/console>

 Download .csv

	User	Access key ID	Secret access key	Email login instructions
▶	lab_demo		***** Show	Send email ↗

Get project from Gitlab

- Clone project from GitLab
 - The client side code is in the ***weathermood*** project
 - The server side code is in the ***weathermood-server*** project
- Checkout branch
 - The client side code is in ***weathermood*** project, ***server-file*** branch
 - The server side code is in the ***weathermood-server*** project, ***file*** branch

Elastic Beanstalk

- Easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.



awsebcli

- AWS Elastic Beanstalk Command Line Interface (EB CLI)
 - create, configure, and manage Elastic Beanstalk environments
- Please install awsebcli first
 - <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb-cli3-install.html>

awsebcli

- Under weathermood-server
 - eb init

```
→ weathermood-server git:(file) eb init -i
```

```
Select a default region
```

- 1) us-east-1 : US East (N. Virginia)
 - 2) us-west-1 : US West (N. California)
 - 3) us-west-2 : US West (Oregon)
 - 4) eu-west-1 : EU (Ireland)
 - 5) eu-central-1 : EU (Frankfurt)
 - 6) ap-south-1 : Asia Pacific (Mumbai)
 - 7) ap-southeast-1 : Asia Pacific (Singapore)
 - 8) ap-southeast-2 : Asia Pacific (Sydney)
 - 9) ap-northeast-1 : Asia Pacific (Tokyo)
 - 10) ap-northeast-2 : Asia Pacific (Seoul)
 - 11) sa-east-1 : South America (Sao Paulo)
 - 12) cn-north-1 : China (Beijing)
 - 13) cn-northwest-1 : China (Ningxia)
 - 14) us-east-2 : US East (Ohio)
 - 15) ca-central-1 : Canada (Central)
 - 16) eu-west-2 : EU (London)
 - 17) eu-west-3 : EU (Paris)
 - 18) eu-north-1 : EU (Stockholm)
 - 19) ap-east-1 : Asia Pacific (Hong Kong)
 - 20) me-south-1 : Middle East (Bahrain)
- (default is 3):

```
Select an application to use
```

- 1) weathermood-server-file
 - 2) [Create new Application]
- (default is 2): 1

```
It appears you are using Docker. Is this correct?
```

```
(Y/n): Y
```

```
Select a platform branch.
```

- 1) Docker running on 64bit Amazon Linux 2
 - 2) Multi-container Docker running on 64bit Amazon Linux
 - 3) Docker running on 64bit Amazon Linux
 - 4) (BETA) Docker running on 64bit Amazon Linux 2 (Beta)
- (default is 1):

```
Do you wish to continue with CodeCommit? (y/N) (default is n): n
```

```
Do you want to set up SSH for your instances?
```

```
(Y/n): n
```

awsebcli

- eb create --single
- Enter **weathermood-{group_id}** for DNS CNAME prefix.
- Ex: **weathermood-1** for group 1

```
→ weathermood-server git:(file) eb create --single
Enter Environment Name
(default is weathermood-server-file-dev):
Enter DNS CNAME prefix
(default is weathermood-server-file-dev):
Would you like to enable Spot Fleet requests for this environment?
(y/N): N
```

Elastic Beanstalk

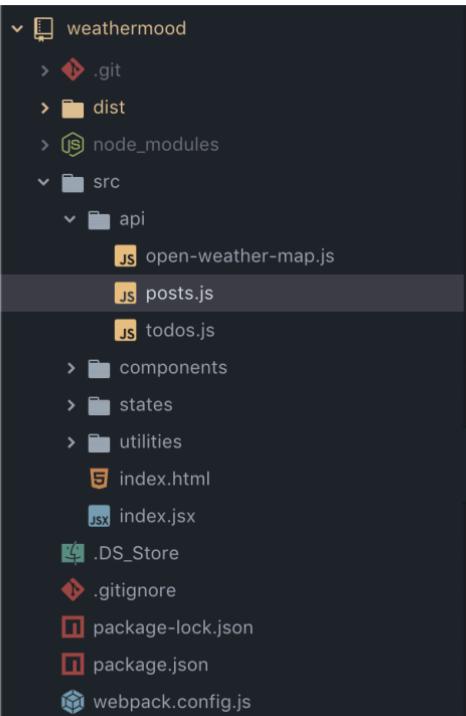
- Find Elastic Beanstalk service
- Click the Applications you just create
- Get the URL

The screenshot shows the AWS Elastic Beanstalk console interface. On the left, a sidebar navigation bar includes links for Environments, Applications (which is currently selected), and Recent environments (with one entry: weathermood-server-file-dev). The main content area has a header "Elastic Beanstalk > Applications" and displays the "All applications" page. It features a search bar labeled "Filter results matching the display values" and a table with columns "Application name" and "Recent environments". A single application row is visible: "weathermood-server-file" (underlined in blue). To the right, a modal window titled "Application 'weathermood-server-file' environments" is open, showing a table with columns "Environment name", "Health", "Date created", "Last modified", and "URL". One environment row is listed: "weathermood-server-file-dev" with status "Ok", creation date "2020-04-17 12:26:24 UTC+0800", last modified "2020-04-17 13:03:45 UTC+0800", and URL "weathermood-server-file-dev.us-west-2.elasticbeanstalk.com".

Environment name	Health	Date created	Last modified	URL
weathermood-server-file-dev	Ok	2020-04-17 12:26:24 UTC+0800	2020-04-17 13:03:45 UTC+0800	weathermood-server-file-dev.us-west-2.elasticbeanstalk.com

Project Code

- Go to client project
 - weathermood -> src -> api -> post.js
- Paste Elastic Beanstalk Url to postBaseUrl



```
import axios from 'axios';

// Develop server URL
// const postBaseUrl = 'http://localhost:8080/api';

// Staging server URL
// const postBaseUrl = 'http://weathermood-staging.us-west-2.elasticbeanstalk.com/api';

// Production server URL
const postBaseUrl = 'http://weathermood-server-dev2.us-west-2.elasticbeanstalk.com/api';

export function listPosts(searchText = '') {
    let url = `${postBaseUrl}/posts`;
    if (searchText)
        url += `?searchText=${searchText}`;

    console.log(`Making GET request to: ${url}`);

    return axios.get(url).then(function(res) {
        if (res.status !== 200)
            throw new Error(`Unexpected response code: ${res.status}`);

        return res.data;
    });
}
```

Project Code

- Build client project
 - npm run build
- Copy client project dist folder to server dist folder
 - Weathermood project dist -> weathermood-serveproject dist

awsebcli

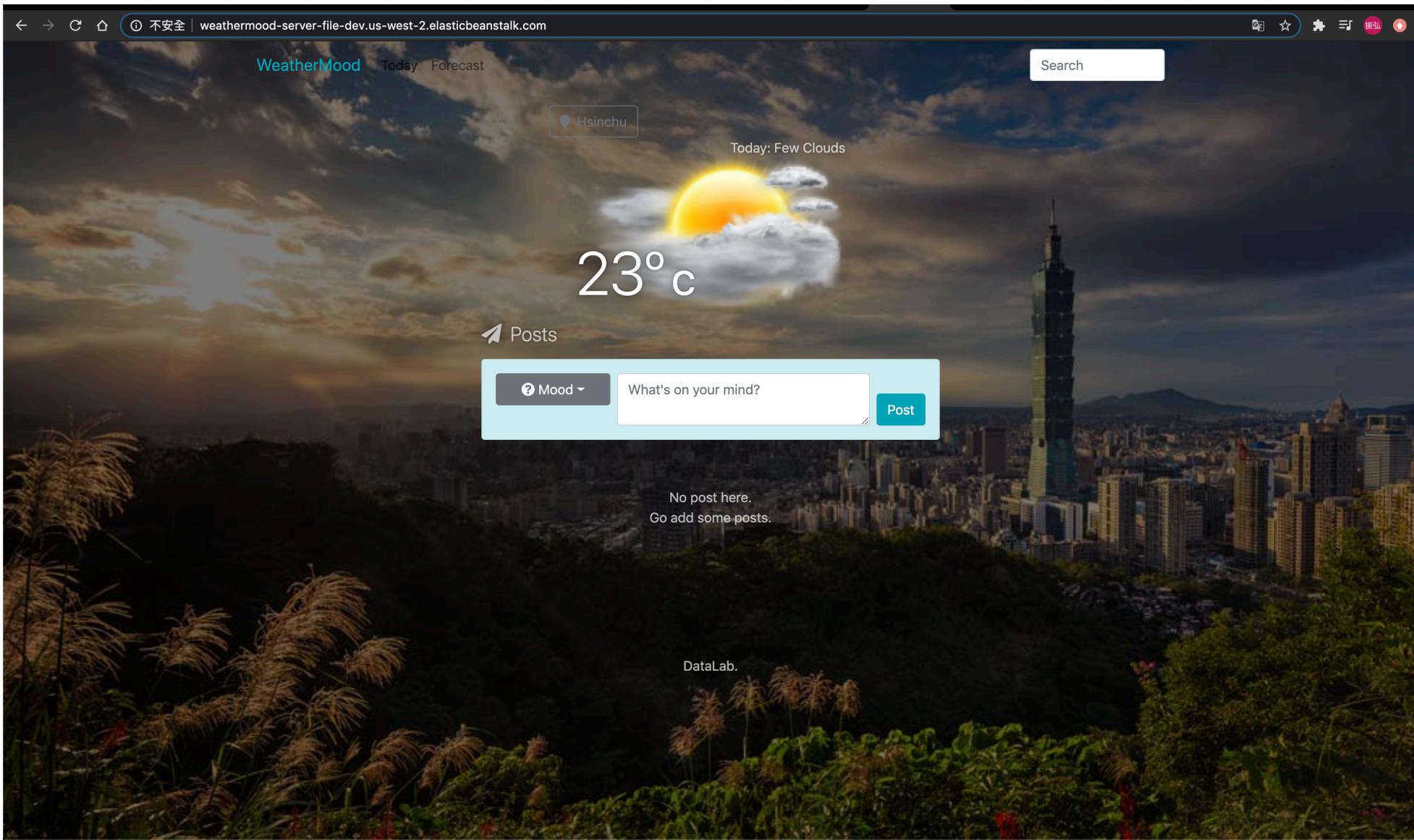
- eb deploy <env>
- It will deploy committed change only. Therefore, you have to commit before you deploy.

```
→ weathermood-server git:(file) ✘ git add .
→ weathermood-server git:(file) ✘ git commit -m "for lab demo"
[file 0312b98] for lab demo
4 files changed, 5 insertions(+), 6 deletions(-)
create mode 100644 dist/.DS_Store
rewrite dist/index.bundle.js (62%)
rewrite dist/index.bundle.js.map (60%)
→ weathermood-server git:(file) eb deploy
```

```
→ weathermood-server git:(file) eb deploy
Creating application version archive "app-c335-190427_202417".
Uploading weathermood-server/app-c335-190427_202417.zip to S3. This may take a while.
Upload Complete.
2019-04-27 12:24:30    INFO  Environment update is starting.
2019-04-27 12:24:33    INFO  Deploying new version to instance(s).
2019-04-27 12:24:40    INFO  Successfully pulled node:6.10
2019-04-27 12:24:40    INFO  Successfully built aws(beanstalk/staging-app
2019-04-27 12:24:49    INFO  Docker container 66d894332056 is running aws(beanstalk/current-app.
2019-04-27 12:24:56    INFO  New application version was deployed to running EC2 instances.
2019-04-27 12:24:56    INFO  Environment update completed successfully.

Alert: An update to the EB CLI is available. Run "pip install --upgrade awsebcli" to get the latest
version.
```

Enter URL into Browser



Lab assignment

- Deploy project to AWS
- After you deploy leave your URL on discord and TA will check