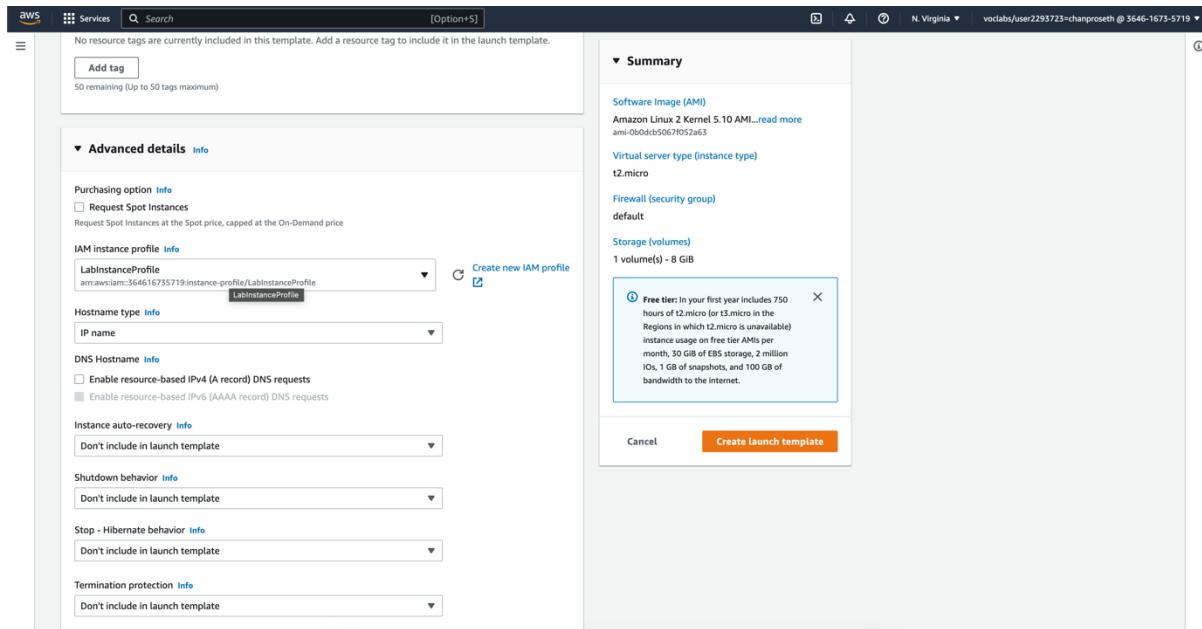


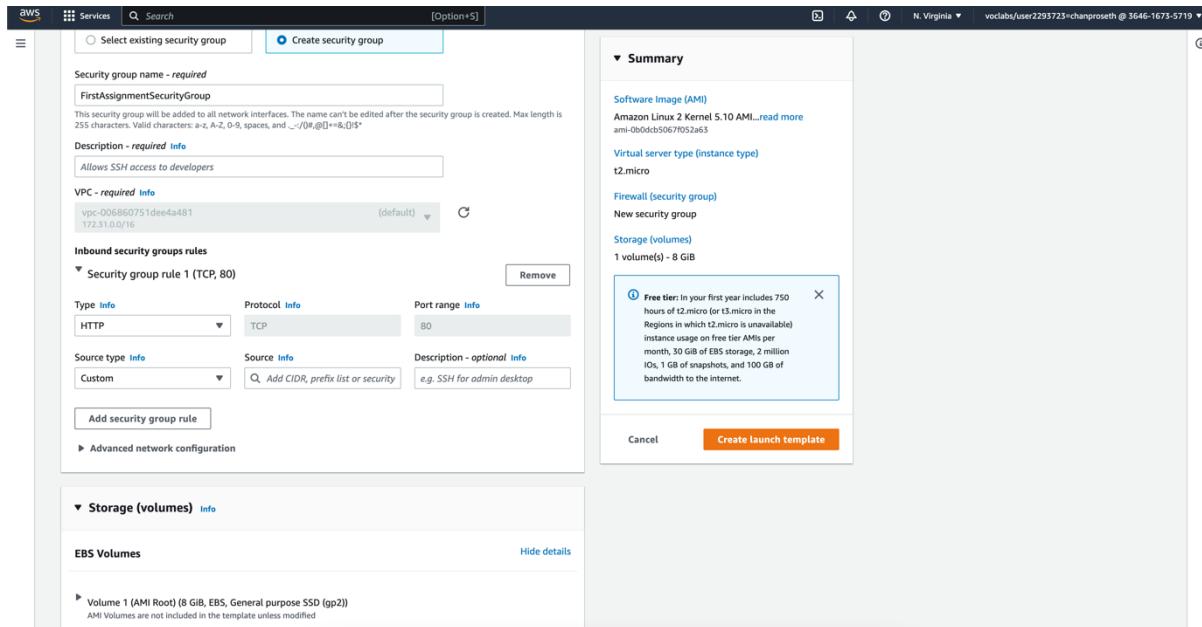
Part I: Spin up an EC2 instance.

1. Allow HTTP:80 port from the world (0.0.0.0/0) in the Network Setting panel.
2. Expand Advanced Settings, and select the LabInstanceProfile.

→ Create Launch template



→ Allow http port 80 to public



→ Success spin up

The screenshot shows the AWS EC2 console with the path: EC2 > Launch templates > Create launch template. A success message is displayed: "Successfully created FirstAssignment (lt-04dde0c5a80ea2a5c)". Below this, an "Actions log" table shows the following steps and their status:

Action	Status
Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Create Launch Template	Succeeded

On the right, there is a "Description" field with the placeholder "A description to help you identify the security group." Below the log, a "Next steps" section provides links to "Launch an instance", "Create an Auto Scaling group from your template", and "Create a Spot Fleet". At the bottom right is an orange "View launch templates" button.

Part 2: Configure a web server on EC2.

1. Select the instance
2. Hit Connect
3. Select the “Session Manager” tab and hit Connect.
4. To install and customize a web server:

```
sudo -s => Logging as a root user so you can start the HTTPD service  
yum install httpd -y => Installing a web server  
service httpd start => Starting the server  
cd /var/www/html => Changing the directory to customize the default Apache page. nano  
index.html => Create the index.html and write your name here as HTML.
```

→ Use http public address : <http://54.144.118.38/> to check result

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with a 'New' link), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (with a 'New' link), Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, and Placement Groups.

The main content area displays the 'Instance summary for i-0b86a8e71ca41097c' for instance i-0b86a8e71ca41097c. It includes fields for Instance ID, Public IPv4 address (54.144.118.38), Private IPv4 address (172.31.91.29), Public IP DNS name (ip-172-31-91-29.ec2.internal), Private IP DNS name (ip-172-31-91-29.ec2.internal), Instance type (t2.micro), Auto-assigned IP address (54.144.118.38 [Public IP]), VPC ID (vpc-006860751dee4a481), IAM Role (LabRole), Subnet ID (subnet-0dfcb01060789e8ca), and various status indicators like Running and Opt-in to AWS Compute Optimizer finding.

Below the summary, there are tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under the Security tab, there are sections for Security details (IAM Role: LabRole, Owner ID: 364616735719, Security groups: sg-049a032dc8d3c43b0 (FirstAssignmentSecurityGroup)) and Inbound rules (a table showing one rule: Name: sgr-0e9d8063dad5aca64, Port range: 80, Protocol: TCP, Source: 0.0.0.0/0, Security groups: FirstAssignmentSecurityGroup).

→ Select session manager and connect to jump to session manager console

The screenshot shows the 'Connect to instance' dialog box. At the top, it says 'Connect to instance Info' and 'Connect to your instance i-0b86a8e71ca41097c using any of these options'. Below this, there are four tabs: EC2 Instance Connect (selected), Session Manager (highlighted in orange), SSH client, and EC2 serial console. A section titled 'Session Manager usage:' lists the following points:

- Connect to your instance without SSH keys or a bastion host.
- Sessions are secured using an AWS Key Management Service key.
- You can log session commands and details in an Amazon S3 bucket or CloudWatch Logs log group.
- Configure sessions on the Session Manager [Preferences](#) page.

At the bottom of the dialog, there are 'Cancel' and 'Connect' buttons. The 'Connect' button is highlighted with a yellow background.

→ Install server

```

Session ID: user2293723=chanproseth-007d3a7d9ae5dc7cc Instance ID: i-0b86a8e71ca41097c Terminate
sh-4.2$ sudo -s
[root@ip-172-31-91-29 bin]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-fsleytem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package httpd-fsleytem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/pam.d/system-auth for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package httpd-fsleytem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/pam.d/system-auth for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository        Size
=====
Installing:
httpd             x86_64    2.4.54-1.amzn2          amzn2-core       1.4 M
Installing for dependencies:
apr               x86_64    1.7.0-9.amzn2          amzn2-core       122 k
apr-util          x86_64    1.6.1-5.amzn2.0.2       amzn2-core       99 k
apr-util-bdb      x86_64    1.6.1-5.amzn2.0.2       amzn2-core       19 k
generic-logos-httd noarch   18.0.0-4.amzn2          amzn2-core       19 k
httpd-fsleytem   noarch   2.4.54-1.amzn2          amzn2-core       24 k
httpd-tools       x86_64    2.4.54-1.amzn2          amzn2-core       50 k
mailcap           noarch   2.1.41-2.amzn2          amzn2-core       31 k
mod_http2         x86_64    1.15.19-1.amzn2.0.1     amzn2-core       149 k

Transaction Summary
=====
Install 1 Package (+8 Dependent packages)

Total download size: 1.9 M
Installed size: 5.2 M
Downloading packages:

Session ID: user2293723=chanproseth-007d3a7d9ae5dc7cc Instance ID: i-0b86a8e71ca41097c Terminate
=====
Transaction Summary
=====
Install 1 Package (+8 Dependent packages)

Total download size: 1.9 M
Installed size: 5.2 M
Downloading packages:
(1/9): apr-1.7.0-9.amzn2.x86_64.rpm | 122 kB 00:00:00
(2/9): apr-util-1.6.1-5.amzn2.0.2.x86_64.rpm | 99 kB 00:00:00
(3/9): apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64.rpm | 19 kB 00:00:00
(4/9): generic-logos-httd-18.0.0-4.amzn2.noarch.rpm | 19 kB 00:00:00
(5/9): httpd-fsleytem-2.4.54-1.amzn2.noarch.rpm | 24 kB 00:00:00
(6/9): httpd-2.4.54-1.amzn2.x86_64.rpm | 1.4 MB 00:00:00
(7/9): httpd-tools-2.4.54-1.amzn2.x86_64.rpm | 88 kB 00:00:00
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm | 31 kB 00:00:00
(9/9): mod_http2-1.15.19-1.amzn2.0.1.x86_64.rpm | 149 kB 00:00:00

Total: 8.2 MB/s | 1.9 MB 00:00:00

=====
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.7.0-9.amzn2.x86_64                               1/9
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64                  2/9
  Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64                  3/9
  Installing : httpd-tools-2.4.54-1.amzn2.x86_64                   4/9
  Installing : httpd-fsleytem-2.4.54-1.amzn2.noarch                 5/9
  Installing : generic-logos-httd-18.0.0-4.amzn2.noarch              6/9
  Installing : mailcap-2.1.41-2.amzn2.noarch                         7/9
  Installing : mod_http2-1.15.19-1.amzn2.0.1.x86_64                  8/9
  Installing : httpd-2.4.54-1.amzn2.x86_64                           9/9
  Verifying : apr-1.7.0-9.amzn2.x86_64                               1/9
  Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64                  2/9
  Verifying : httpd-tools-2.4.54-1.amzn2.x86_64                   3/9
  Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64                  4/9
  Verifying : httpd-2.4.54-1.amzn2.x86_64                           5/9
  Verifying : mailcap-2.1.41-2.amzn2.noarch                         6/9
  Verifying : generic-logos-httd-18.0.0-4.amzn2.noarch              7/9
  Verifying : httpd-fsleytem-2.4.54-1.amzn2.noarch                 8/9
  Verifying : apr-1.7.0-9.amzn2.x86_64                               9/9

Installed:
  httpd.x86_64 0:2.4.54-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.0-9.amzn2           apr-util.x86_64 0:1.6.1-5.amzn2.0.2           generic-logos-httd.noarch 0:18.0.0-4.amzn2
  httpd-fsleytem.noarch 0:2.4.54-1.amzn2   httpd-tools.x86_64 0:2.4.54-1.amzn2           mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-172-31-91-29 bin]#

```

→ Cd to path and create html to display my name

```
Session ID: user2293723=chanproseth-007d5a7d9ae5dc7cc           Instance ID: i-0b86a8e71ca41097c
Terminat
Installing : mod_http2-1.15.19-1.amzn2.0.1.x86_64          8/9
Installing : httpd-2.4.54-1.amzn2.x86_64                  9/9
Verifying  : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64          1/9
Verifying  : apr-util-1.6.1-5.amzn2.0.2.x86_64          2/9
Verifying  : mod_http2-1.15.19-1.amzn2.0.1.x86_64          3/9
Verifying  : httpd-2.4.54-1.amzn2.x86_64                  4/9
Verifying  : mailcap-2.1.41-2.amzn2.noarch                5/9
Verifying  : generic-logos-httpd-18.0.0-4.amzn2.noarch    6/9
Verifying  : httpd-filesystem-2.4.54-1.amzn2.noarch       7/9
Verifying  : apr-1.7.0-9.amzn2.x86_64                      8/9
Verifying  : apr-1.7.0-9.amzn2.x86_64                      9/9

Installed:
httpd.x86_64 0:2.4.54-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.0-9.amzn2      apr-util.x86_64 0:1.6.1-5.amzn2.0.2      apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2      generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.54-1.amzn2      httpd-tools.x86_64 0:2.4.54-1.amzn2      mailcap.noarch 0:2.1.41-2.amzn2      mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-172-31-91-29 bin]# service httpd -d
The service command supports only basic LSB actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.
[root@ip-172-31-91-29 bin]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@ip-172-31-91-29 bin]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-91-29 bin]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
     Active: active (running) since Mon 2022-11-28 20:33:16 UTC; 68 ago
       Docs: man:httpd.service(8)
   Main PID: 3441 (httpd)
     Status: "processing requests..."
  CGroup: /system.slice/httpd.service
          └─3441 /usr/sbin/httpd -DFOREGROUND
              ├─3442 /usr/sbin/httpd -DFOREGROUND
              ├─3443 /usr/sbin/httpd -DFOREGROUND
              ├─3444 /usr/sbin/httpd -DFOREGROUND
              ├─3445 /usr/sbin/httpd -DFOREGROUND
              ├─3446 /usr/sbin/httpd -DFOREGROUND

Nov 28 20:33:16 ip-172-31-91-29.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 28 20:33:16 ip-172-31-91-29.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-91-29 bin]# cd /var/www/html
[root@ip-172-31-91-29 html]# nano index.html
[root@ip-172-31-91-29 html]#
```



Chanproseth Kan

Part 3: Creating a lambda function returns an array of strings. Make it an API by enabling the public URL :

- a) Choose the LabRole as IAM
- b) Enable URL and enable CORS

→ Create lambda function

The screenshot shows the 'Create function' wizard in the AWS Lambda console. The 'Basic information' step is selected. In the 'Function name' field, 'FirstAssignmentLambda' is entered. The 'Runtime' is set to 'Node.js 18.x'. Under 'Architecture', 'x86_64' is chosen. In the 'Permissions info' section, the 'Change default execution role' button is expanded, showing the 'Create a new role with basic Lambda permissions' option selected. At the bottom right, there is a 'Create function' button.

→ Enable function url and cors

The screenshot shows the 'Advanced settings' configuration page for the Lambda function. Under 'Auth type', 'NONE' is selected. A note states that when choosing 'NONE', Lambda automatically creates a resource-based policy. Under 'Function URI permissions', 'Configure cross-origin resource sharing (CORS)' is checked. Other options like 'Enable tags' and 'Enable VPC' are also listed. At the bottom right, there is a 'Create function' button.

→ Successfully create function

The screenshot shows the AWS Lambda console. At the top, a green banner says "Successfully created the function FirstAssignmentlambda. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the function name "FirstAssignmentlambda" is displayed. The "Function overview" tab is selected. On the left, there's a "Layers" section with "(0)" and a "+ Add trigger" button. On the right, there's a "Description" field (empty), "Last modified" (4 seconds ago), and a "Function ARN" field containing "arn:aws:lambda:us-east-1:364616735719:function:FirstAssignmentlambda". Below that is a "Function URL" field with a link: "https://a4krbrjilhk3h6wwd7qto4wgau0pqmw.lambda-url.us-east-1.on.aws". At the bottom, tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions" are visible, with "Code" being the active tab.

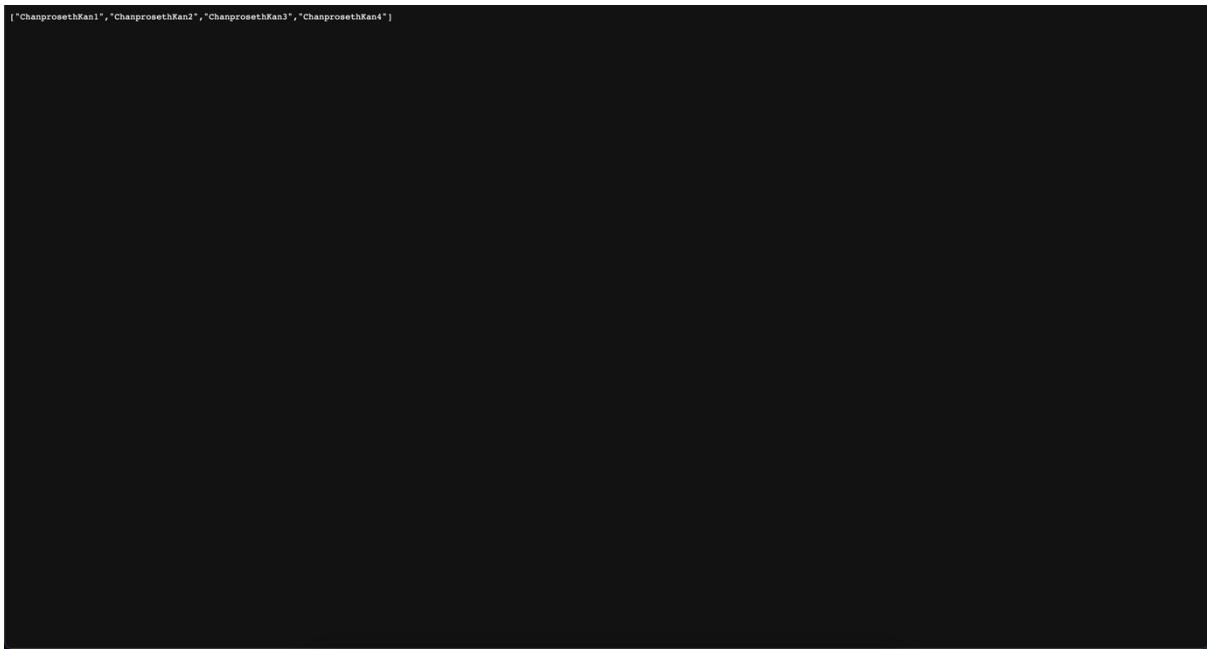
→ Custom response to return name array

The screenshot shows the AWS Lambda code editor for the "index.js" file under the "FirstAssignmentlambda" function. The "Code" tab is selected. The code is as follows:

```
1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify(['ChandreshKan1', 'ChandreshKan2', 'ChandreshKan3', 'ChandreshKan4']),
6     };
7     return response;
8 };
9 
```

The code editor includes a toolbar with "File", "Edit", "Find", "View", "Go", "Tools", "Window", "Test", and "Deploy". The status bar at the bottom indicates "9:1 JavaScript Spaces: 4".

➔ Result



Part4: Call the API in React and deploy the front-end app in S3.

- A. Install NodeJS on your laptop
- B. npx create-react-app appname
- C. npm install axios
- D. npm start – to start your front-end app
- E. npm run build – after testing, build the app
- F. create a bucket and deselect “Block public access”
- G. drop all files inside the build folder into the bucket.
- H. Write a policy that makes all objects in the bucket public. Refer to the next section.
- I. Enable “static website hosting” and define the index.html as the index and error page.

→ Step A to D complete

```
Compiled successfully!

You can now view first-assignment-react in the browser.

  Local:          http://localhost:3000
  On Your Network:  http://10.200.29.77:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
npm run build
^C
+ first-assignment-react npm run build

> first-assignment-react@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.

File sizes after gzip:

 46.61 kB  build/static/js/main.cb21e5ba.js
 1.79 kB   build/static/js/787.b43cd43f.chunk.js
 541 B     build/static/css/main.073c9b0a.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server.

  npm install -g serve
  serve -s build

Find out more about deployment here:
  https://cra.link/deployment

+ first-assignment-react open .
+ first-assignment-react
```

→ Step F

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' step is active, showing fields for 'Bucket name' (set to 'FirstAssignmentBucket') and 'AWS Region' (set to 'US East (N. Virginia) us-east-1'). Below these, there's a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button. The 'Object Ownership' step is shown below, with the 'ACLs enabled (recommended)' option selected. At the bottom, there are 'Feedback' and 'Cookie preferences' links.

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

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer
The object writer remains the object owner.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to buckets and its objects is blocked, turn on block all public access. These settings apply only to this bucket and its objects. AWS recommends that you turn off all public access before applying any of these settings, ensure that your application will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific usage cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

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

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

→ Successfully create bucket

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Successfully created bucket "kc-first-assignment-bucket"
To upload files and folders, or to configure additional bucket settings choose [View details](#).

Replicate data within and between AWS Regions using Amazon S3 Replication.
[View tutorial](#)

Amazon S3 > Buckets

Account snapshot
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

Buckets (1) [Info](#)
Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
kc-first-assignment-bucket	US East (N. Virginia) us-east-1	Objects can be public	November 28, 2022, 16:26:22 (UTC-06:00)

<https://s3.console.aws.amazon.com/s3/buckets/kc-first-assignment-bucket?region=us-east-1>

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

→ Step H

The screenshot shows the AWS S3 Bucket Policy configuration page. At the top, there are three informational banners: "We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.", "Successfully edited bucket policy.", and "Replicate data within and between AWS Regions using Amazon S3 Replication." Below these, a section titled "Individual Block Public Access settings for this bucket" is shown. The main area displays the JSON-based bucket policy:

```
{
  "Version": "2012-10-17",
  "Id": "Policy1650912821527",
  "Statement": [
    {
      "Sid": "Stmt1650912820312",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::kc-first-assignment-bucket/*"
    }
  ]
}
```

At the bottom, there is an "Object Ownership" section with a note about controlling ownership of objects written to the bucket from other AWS accounts.

→ Step E (add code for all Api in react project) and npm build

The screenshot shows a code editor with the file "App.js" open. The code uses React hooks to fetch data from a Lambda function via Axios. The code is as follows:

```
import axios from "axios";
import { useEffect, useState } from "react";

export default function App() {
  const [students, setStudents] = useState([]);

  useEffect(() => {
    async function fetchStudents() {
      const studentsFromLambda = await axios.get(
        "https://4ykrurj1hh3hwq7qto-wgqubpcjnw.lambda-url.us-east-1.on.aws/"
      );
      setStudents(studentsFromLambda);
      console.log(studentsFromLambda);
    }
    fetchStudents();
  }, []);

  return (
    <div>
      Cloud Computing course
      <ol>
        {students.map((student) => (
          <li>{student}</li>
        ))}
      </ol>
    </div>
  );
}


```

The code editor interface includes a sidebar for "EXPLORER" showing the project structure, a bottom navigation bar with "OUTLINE", "TIMELINE", "SONARLINT RULES", "SONARLINT ISSUE LOCATIONS", and "SONARLINT CONNECTED MODE", and a status bar at the bottom indicating "Ln 5, Col 3 Spaces: 2 UTF-8 LF () JavaScript".

→ Step G (Upload file in build folder of react project to S3 bucket)

The screenshot shows the AWS S3 console with the following details:

- Upload succeeded:** View details below.
- Summary:**
 - Destination:** s3://kc-first-assignment-bucket
 - Succeeded:** 14 files, 735.6 KB (100.00%)
 - Failed:** 0 files, 0 B (0%)
- Files and folders:** (14 Total, 735.6 KB)

Name	Folder	Type	Size	Status	Error
787.b43cda3fchunk.js	static/js/	text/javascript	4.5 KB	Succeeded	-
787.b43cda3fchunk.js.map	static/js/	-	10.3 KB	Succeeded	-
asset-manifest.json	-	application/json	517.0 B	Succeeded	-
favicon.ico	-	image/x-icon	3.8 KB	Succeeded	-
index.html	-	text/html	644.0 B	Succeeded	-
logo192.png	-	image/png	5.2 KB	Succeeded	-
logo512.png	-	image/png	9.4 KB	Succeeded	-

→ Enable static website hosting to specified index.html

The screenshot shows the AWS S3 console with the following steps for enabling static website hosting:

- Edit static website hosting:**
 - Static website hosting:** Use this bucket to host a website or redirect requests. [Learn more](#)
 - Hosting type:**
 - Host a static website:** Use the bucket endpoint as the web address. [Learn more](#)
 - Redirect requests for an object:** Redirect requests to another bucket or domain. [Learn more](#)
 - Index document:** Specify the home or default page of the website.
 - Error document - optional:** This is returned when an error occurs.
 - Redirection rules - optional:** Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)
- Edit static website hosting:** When you enable static website hosting, you can configure your Amazon S3 bucket to host a static website or redirect requests.
- Host a static website:** The process of configuring your bucket to host a static website includes the following steps:
 - Enabling static website hosting
 - Configuring an index document and an error document
 - Configuring permissions to allow public read access
- Redirect requests:** To redirect all requests to your bucket's website endpoint or custom domain, configure your bucket to redirect all requests. The Host name is the website address that you want to redirect requests to. To maintain your website configuration and redirect requests for specific content, configure optional JSON redirection rules.
- Learn more:**
 - Hosting a static website with a custom domain
 - Hosting a static website
 - Configuring a webpage redirect
 - Blocking public access

The screenshot shows the AWS S3 Bucket Properties page for a bucket named 'kc-first-assignment-bucket'. A green success message at the top states 'Successfully edited static website hosting.' The main content area is divided into three sections: 'Object Lock', 'Requester pays', and 'Static website hosting'. The 'Object Lock' section shows 'Object Lock' is disabled. The 'Requester pays' section shows 'Requester pays' is disabled. The 'Static website hosting' section shows 'Static website hosting' is enabled, 'Hosting type' is 'Bucket hosting', and the 'Bucket website endpoint' is listed as <http://kc-first-assignment-bucket.s3-website-us-east-1.amazonaws.com>. On the right side, there are several informational panels: 'Bucket properties' (with a note about Object Lock support), 'Protect your storage' (about Amazon S3 Versioning), 'Secure your storage' (about automatic encryption), and 'Archive rarely accessed objects' (about S3 Intelligent-Tiering). The bottom of the page includes standard AWS footer links like Feedback, Privacy, Terms, and Cookies preferences.

Part 5: [personal account] Create your own AWS account. You must add your payment method. Otherwise, services will be unavailable. Do the following 3 tasks. Then don't do anything else without my approval. Otherwise, It will charge you. If you want to practice something in AWS, use the Academy account.

1. Enable MFA for the root user.
2. Create an admin group with an administrator policy. Create a user for yourself in that group. Always use that IAM user. Not your root user.
3. Set up a billing alarm.
 - i. Make sure the region is North Virginia
 - ii. Go to CloudWatch
 - iii. In Alarms, you will see billing that selects the billing metric automatically.

→ After successfully create account

→ Create MFA

The screenshot shows the 'Your Security Credentials' page in the AWS IAM console. On the left, a sidebar lists various IAM management options like Dashboard, Access management, and Access reports. The main content area is titled 'Your Security Credentials' and includes sections for 'Password' and 'Multi-factor authentication (MFA)'. Under 'Multi-factor authentication (MFA)', there's a table with one row for a Virtual TOTP device. The table has columns for 'Device type', 'Serial number', and 'Actions'. A blue 'Activate MFA' button is visible above the table.

→ Create user MFA

The screenshot shows the 'Users' list page in the AWS IAM console. The sidebar on the left shows the 'Users' option under 'Access management' is selected. The main content area displays a table with one user entry: 'chanproseth.kan'. The table columns include 'User name', 'Groups', 'Last activity', 'MFA', 'Password age', and 'Active key age'. The user 'chanproseth.kan' is listed under the 'Admin' group, has never been active, uses a virtual MFA device, and their password was updated 3 minutes ago.

→ Setting up billing alarm in Cloud watch

The screenshot shows the AWS CloudWatch Metrics Insights interface for creating a new alarm. The top navigation bar includes 'Services' and a search bar. The main area is titled 'estimatedcharges' and shows a 'Period' of '6 hours'. The 'Conditions' section is expanded, showing the 'Threshold type' set to 'Static' (selected) and 'Anomaly detection' (unselected). Under 'Whenever EstimatedCharges is...', the condition 'Greater than threshold' is selected. The threshold value is set to '1 USD'. Below this, the 'Additional configuration' section includes 'Datapoints to alarm' (set to 1 out of 5), 'Missing data treatment' (set to 'Treat missing data as missing'), and 'Alarm name' (set to 'Billing Alarm'). At the bottom right are 'Cancel' and 'Next' buttons.

The screenshot shows the first step of the AWS CloudWatch Alarms creation wizard, titled 'Specify metric and conditions'. The left sidebar lists steps: Step 1 (Selected), Step 2, Step 3 (Selected), and Step 4. The main form is titled 'Add name and description' and contains fields for 'Name and description': 'Alarm name' (set to 'Billing Alarm') and 'Alarm description - optional' (set to 'Billing getting more than ur limit'). A note at the bottom states 'Up to 1024 characters (\$4/1024)'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

AWS Services Search [Option+S] N. Virginia chanproseth.kan

CloudWatch > Alarms > Create alarm

Step 1 Specify metric and conditions

Step 2 Configure actions

Step 3 Add name and description

Step 4 Preview and create

Preview and create

Step 1: Specify metric and conditions

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 6 hours.

2
1.5
1
0.5
0

11/23 11/25 11/27

EstimatedCharges

Namespace AWS/Billing
Metric name EstimatedCharges
Currency USD
Statistic Maximum
Period 6 hours

Conditions

Threshold type Static
Whenever EstimatedCharges is Greater/Equal (\geq) than...

Feedback Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences