YIYANG WANG

CSC309 Week 11

DATABASE DEPLOYMENT & CORS & CLOUD INFRASTRUCTURE

HTTPS://WWW.CS.TORONTO.EDU/~KIANOOSH/COURSES/CSC309H5/

26 MARCH, 2025

Please join the

Zoom for polls

Meeting Code:

2210147631

Password:

123456

CORS – Why Does It Matter?

- **CORS** = Cross-Origin Resource Sharing
- **Purpose:** Protects browsers from malicious cross-site requests.
- **How it works:**
  - Browser sends a “preflight” request (OPTIONS)
  - Server responds with `Access-Control-Allow-Origin`, `Access-Control-Allow-Methods`, etc.
- **Practical Tips:**
  - For Next.js API routes on Vercel/Netlify, set headers in your API route or in vercel.json.
  - For Express servers, use cors npm package.

```
fetch('http://localhost:5000/data', {
  method: 'GET',
})
  .then(response => response.json())
  .then(data => {
    console.log('Received data:', data);
  })
  .catch(error => {
    console.error('Error fetching data:', error);
  });
```

a frontend running on 3000 send a request

```
const express = require('express');
const cors = require('cors');

const app = express();

// Enable CORS for a specific origin
app.use(cors({
  origin: 'http://localhost:3000', // frontend origin
}));

app.get('/data', (req, res) => {
  res.json({ message: 'Hello from server' });
});

app.listen(5000, () => {
  console.log('Server running on http://localhost:5000');
});
```

response headers

HTTP/1.1 200 OK

Access-Control-Allow-Origin: http://localhost:3000

Access-Control-Allow-Methods: GET,POST,PUT,DELETE

Access-Control-Allow-Headers: Content-Type

Access-Control-Allow-Credentials: true

Next.js

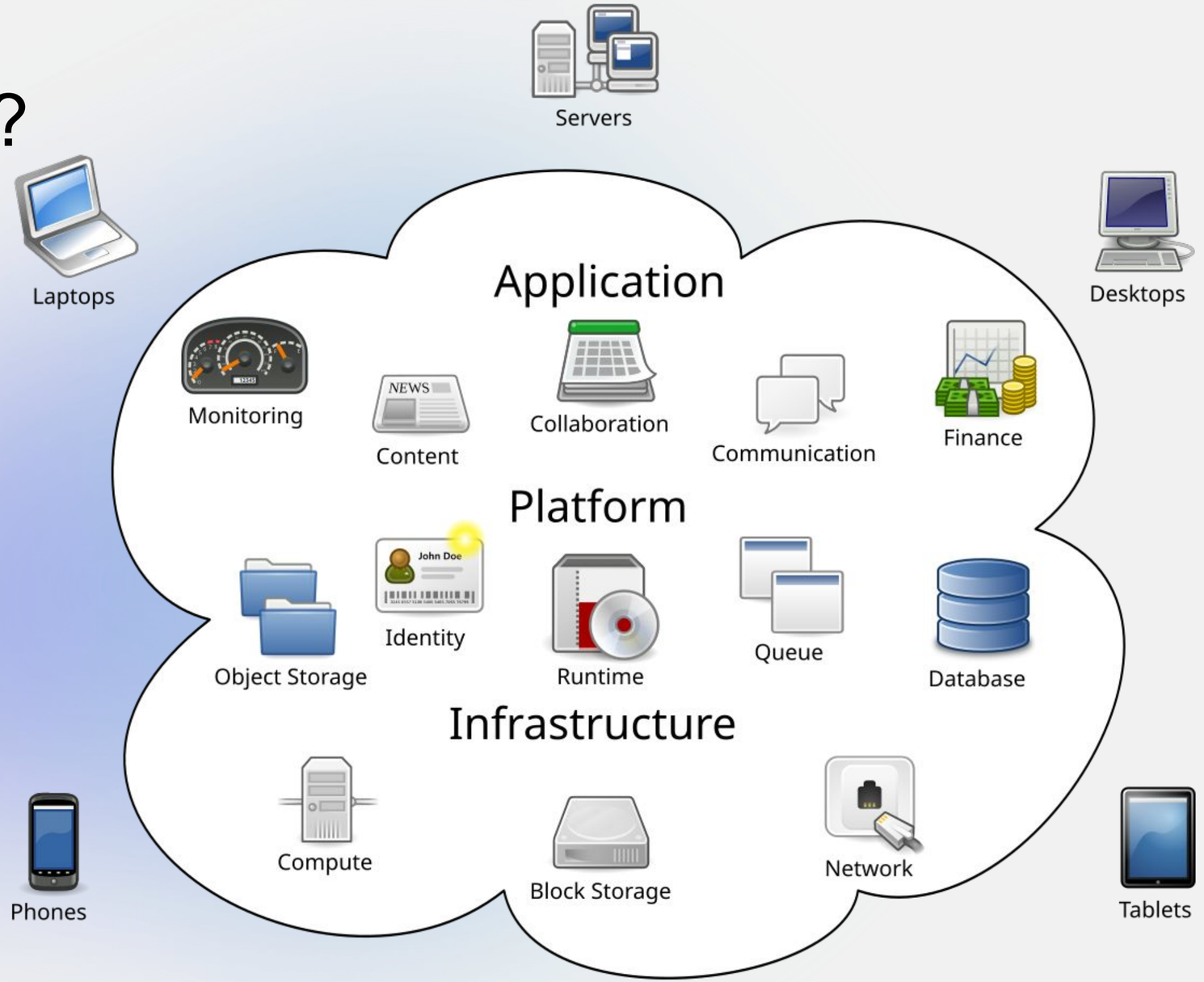
```jsx
// pages/index.js
import { useEffect, useState } from 'react';

export default function Home() {
  const [data, setData] = useState(null);

  useEffect(() => {
    fetch('http://localhost:5000/data')
      .then(response => response.json())
      .then(data => {
        console.log('Received data:', data);
        setData(data);
      })
      .catch(error => {
        console.error('Error fetching data:', error);
      });
  }, []);

  return (
    <div>
      <h1>Next.js Frontend</h1>
      <p>{data ? data.message : 'Loading...'}</p>
    </div>
  );
}
```

```jsx
const cors = require('cors');
// allow Next.js frontend to access
app.use(cors({
  origin: 'http://localhost:3000',
}));
```

Cloud Infrastructure?

computing resources:
servers, storage,
databases, networking

→ provided via the internet



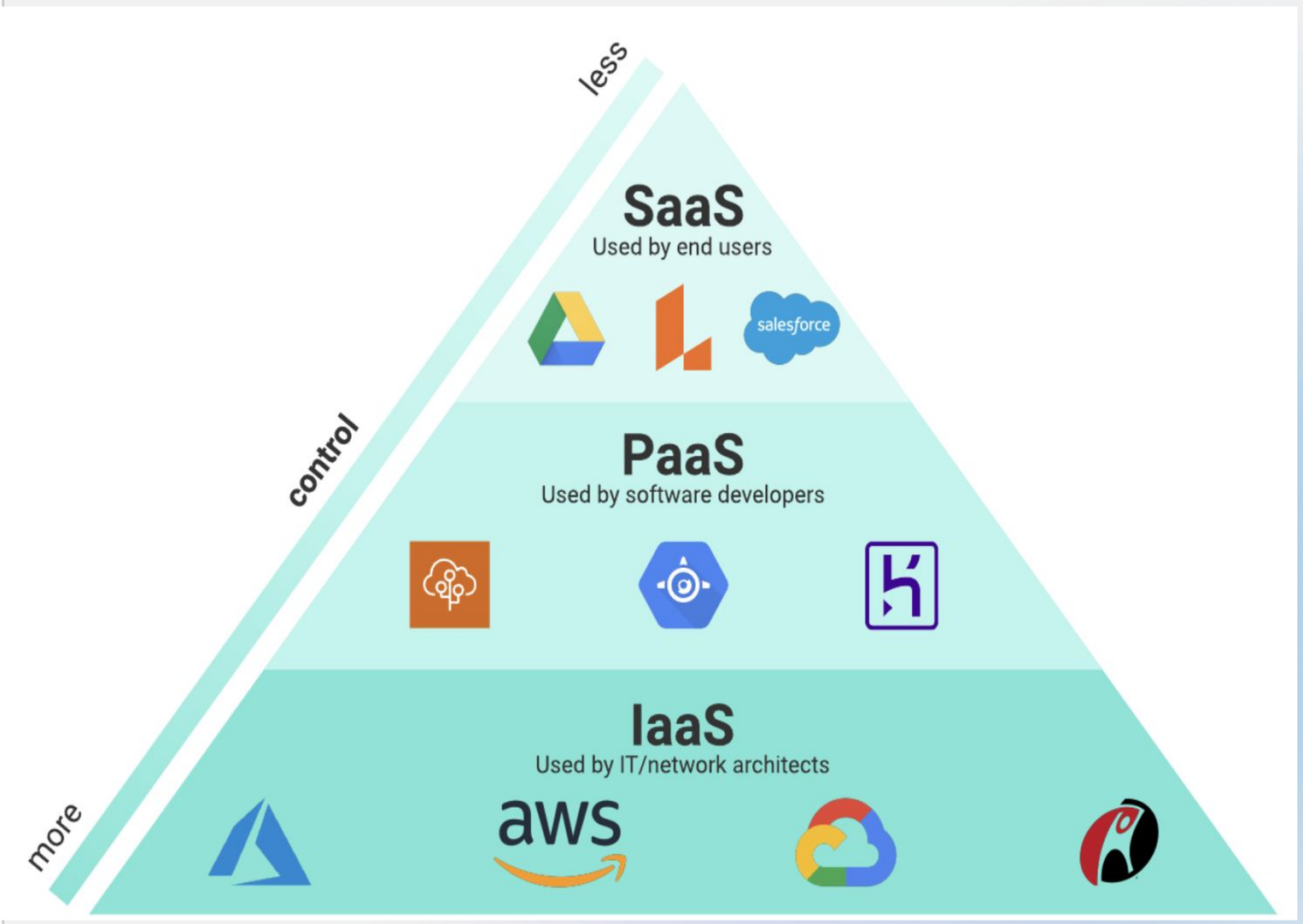
Common Cloud Services

compute → virtual computer to run programs, applications and codes

storage → virtual hard drives that can store files

networking → virtual network to define internet connections or network locations

databases → virtual databases to storing reporting data



Serverless / Function as a Service (FaaS): e.g., AWS Lambda, Vercel, Netlify functions

PaaS (Platform as a Service): cloud-based platforms for users to develop and deliver applications

IaaS (Infrastructure as a Service): e.g., AWS EC2, Azure VMs, Google Compute Engine

https://www.lucidchart.com/blog/cloud-computing-basics

Deploy - Where Does the Database Go?

Local DB: Good for development, not for production.

Managed DB Services (e.g., RDS on AWS, Azure SQL, Google Cloud SQL):

- Pros: Scalability, backups, high availability
- Cons: Might be more complex to configure, cost considerations

External DB Providers: Supabase（Firebase), PlanetScale, Neon, MongoDB Atlas, etc.

- Easy to integrate with Next.js
- Simple, free-tier options for students/small projects

Hosting Platform Comparison

Platform	Next.js Support	Backend/API Hosting	Database Hosting	Docker Support	Free Tier
Vercel	✅ Excellent	✅ (API routes)	❌ (use external)	❌ (no Docker)	✅ Yes
Railway	✅ Good	✅ Full backend	✅ PostgreSQL	✅ (Nixpacks/Docker)	✅ Yes
Fly.io	✅ (via Docker)	✅ Full backend	❌ (use external)	✅ Full Docker	✅ Yes
Supabase	❌ (not for frontend)	❌	✅ PostgreSQL	❌	✅ Yes
AWS	✅ (manual setup)	✅ EC2 / Lambda / Fargate	✅ RDS / DynamoDB	✅ Full Docker	⚠️ Limited
Azure	✅ (manual setup)	✅ App Service / VM	✅ Azure SQL / Cosmos	✅ Full Docker	⚠️ Limited

Deploy Your Personal Website

Third-party that offer free website hosting:
github.io
netlify
vercel
and more…

Deploy a Next.js Project to Vercel

login with your github account

then

→ easy, just import your project repo!

Deploying Projects to Railway

1. Create Railway account & login
2. Create new project → Deploy from GitHub repo
3. Add PostgreSQL database service
4. Set DATABASE_URL in Variables tab

AWS, Azure, Google Cloud

There are many online resources(tutorials, certification courses…)

Today’s bonus

1. An URL to your personal website

   can be your existing portfolio link, or just create a new one with a simple message like “Hi, this is my cool website!”

2. An URL to your deployed application (piazza post # 348)

   1).complete the Dockerfile of last lab’s quiz app and deploy 2) OR your group project(it doesn’t need to be perfect, one functional feature is enough)

   https://drive.google.com/file/d/1NfwlopSFaa4YC_EfoQlbkgSpLzdrOfeD/view?usp=sharing

YIYANG WANG

Thank You

CSC309 Week 11