Deploying Backends

17-316/616 Fall 2025

AI Tools for Software Development

https://ai-developer-tools.github.io

Austin Henley and Andrew Begel

Administrivia

- P6: Deployment released today, due Fri. Nov 20, 11:59pm

Recap: How did deployment go on Monday?

Serverless computing

- No servers that you must manage, just deploy code
- Scales automatically
- Pay for what you use
- Runs your code on trigger events
  - Timer
  - HTTP events
  - File upload

How does it actually work?

- Your code + packages are packaged together
- Runs inside a lightweight VM
- Isolated environment, can’t access other resources
- Other services are used to trigger your serverless function
- After execution, the VM will stay for a bit to be reused
  - Cold start is “slow”, warm start is very fast
- Environment will be destroyed
- Scales easily: spin up many VMs for simultaneous execution

AWS CDK tool

- Instead of using the web app, you can write scripts
- “Infrastructure as code”
- Reuse scripts, code review changes, and keep in GitHub

- npm install –g aws-cdk
- https://docs.aws.amazon.com/cdk/v2/guide/hello-world.html

Activity: Deploy Calculator Backend with AWS Lambda

- Install the AWS CLI
- Install AWS Toolkit extension into VS Code (or Cursor)
- From the AWS Console, open up AWS Lambda
- Choose Create Function
  - Choose Author from scratch and enter calculate and choose runtime.
  - Choose Create function
- Copy the index.mjs file to your calculator backend codebase.
- Connect the handler to your calculate() function.
- Deploy your Lambda function with a .zip file archive
  - https://docs.aws.amazon.com/lambda/latest/dg/nodejs-package.html

Activity: Connect REST API to Calculator

- Modify the Lambda code so it is invoked by HTTP POST requests to your calculate() function.
- Step 1: Open the AWS API Gateway console.
  - Create API
  - In the REST API box, choose Build
  - Under API details, enter Calculator
- Step 2: In the Resources page for your API, choose Create Resource
  - ResourceName is CalculatorManager
- Step 3: Create an HTTP POST method
  - In Resources, highlight CalculatorManager. Choose Create Method
    - Method Type: POST
    - Integration type: Lambda
    - Lambda: enter LambdaFunctionOverHttps
- Step 4: Deploy the API
  - In Resources, choose Deploy API
    - Stage: New stage: test
  - Copy the invoke URL into your calculator frontend.
- This tutorial may help: https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway-tutorial.html

Practice Good Git Hygiene

1. Whenever working on a new task, create a new *feature* branch. *Never* work directly on the main branch.
2. Only when your feature is committed to the feature branch and fully tested, create a pull request (PR) to push the changes to the main branch.
3. Create a GitHub action that runs your integration tests on PR creation.
4. If your integration tests pass, have someone else on your team approve the PR.
5. Create a GitHub action to deploy to AWS on PR approval.

Activity: Automate Deployment

- Get with your team
- Create a GitHub Action that deploys your app to AWS after each commit has been successfully tested and PR approved.

Next week

- How do we know our deployment worked/is working?
- Cloud monitoring services
- E.g., Prometheus, Grafana, notifications