DS 4300

Amazon EC2 & Lambda

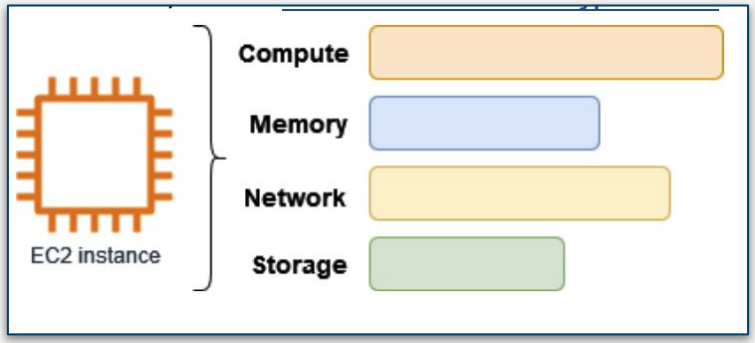
Mark Fontenot, PhD
Northeastern University

Based in part on material from Gareth Eagar's *Data Engineering with AWS*, Packt Publishing

EC2

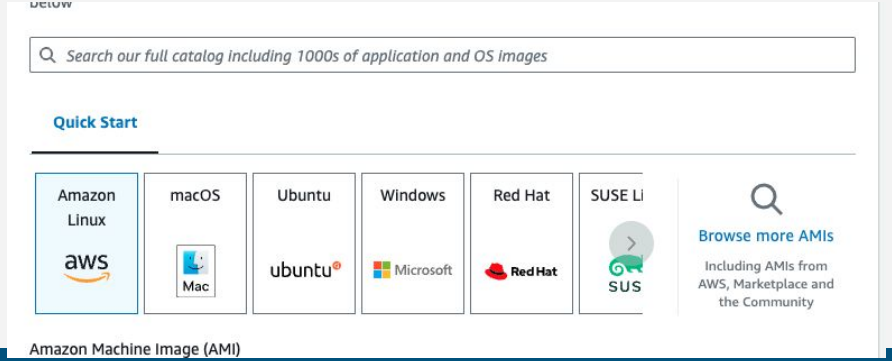
EC2

- EC2 → Elastic Cloud Compute
- Scalable Virtual Computing in the Cloud
- Many (Many!!) instance types available
- Pay-as-you-go model for pricing
- Multiple different Operating Systems



Features of EC2

- **Elasticity** - easily (and programmatically) scale instances up or down as needed
- You can use one of the standard AMIs OR provide your own AMI if pre-config is needed
- Easily integrates with many other services such as S3, RDS, etc.



AMI = Amazon Machine Image

EC2 Lifecycle

- **Launch** - when starting an instance for the first time with a chosen configuration
- **Start/Stop** - Temporarily suspend usage without deleting the instance
- **Terminate** - Permanently delete the instance
- **Reboot** - Restart an instance without sling the data on the root volume

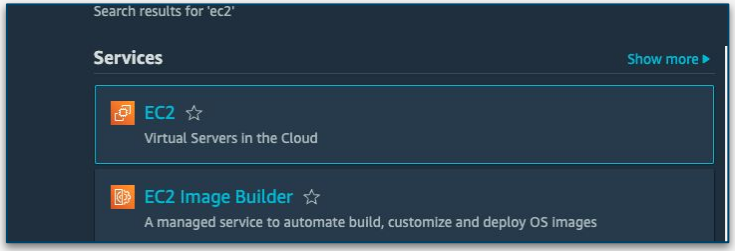
Where Can You Store Data?

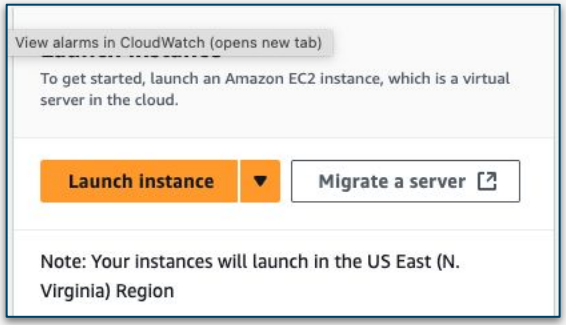
- **Instance Store**: Temporary, high-speed storage tied to the instance lifecycle
- **EFS** (Elastic File System) Support - Shared file storage
- **EBS** (Elastic Block Storage) - Persistent block-level storage
- **S3** - large data set storage or EC2 backups even

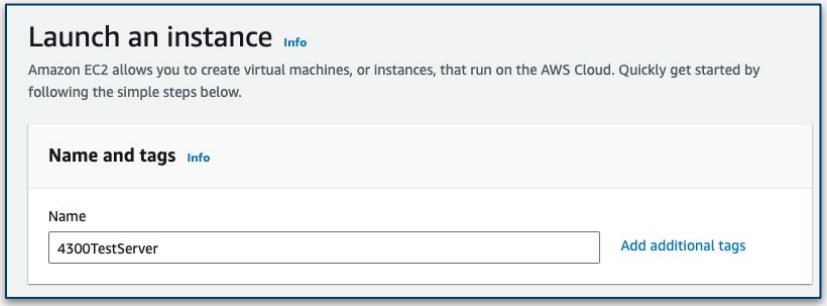
Common EC2 Use Cases

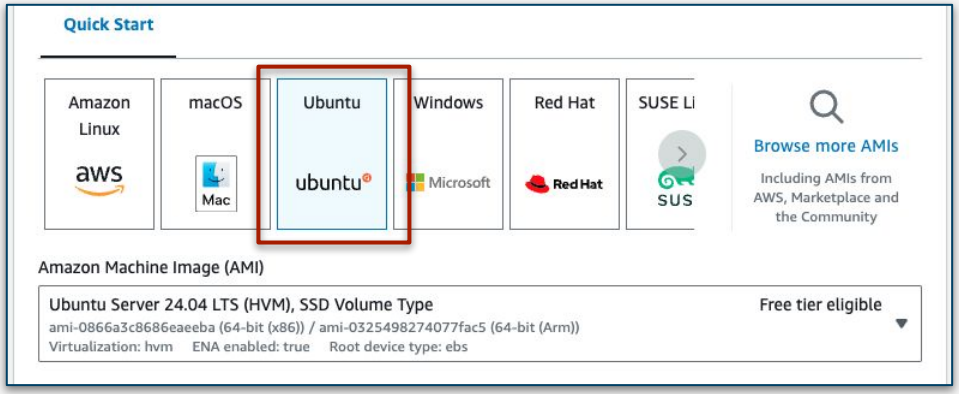
- Web Hosting - Run a website/web server and associated apps
- Data Processing - It’s a VM… you can do anything to data possible with a programming language.
- Machine Learning - Train models using GPU instances
- Disaster Recovery - Backup critical workloads or infrastructure in the cloud

Let’s Spin Up an EC2 Instance

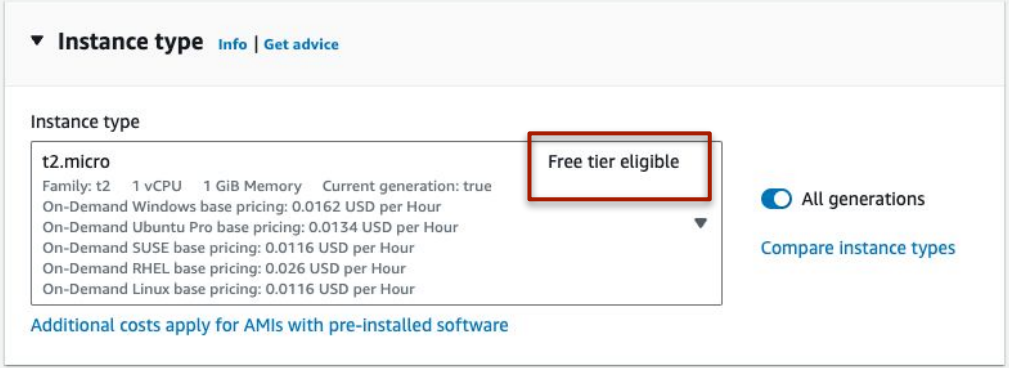


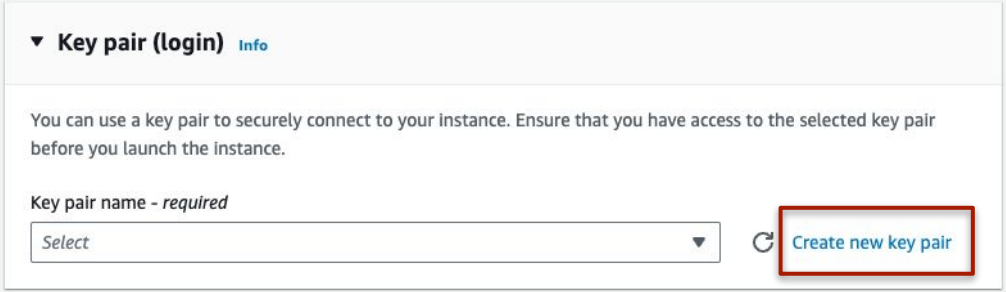


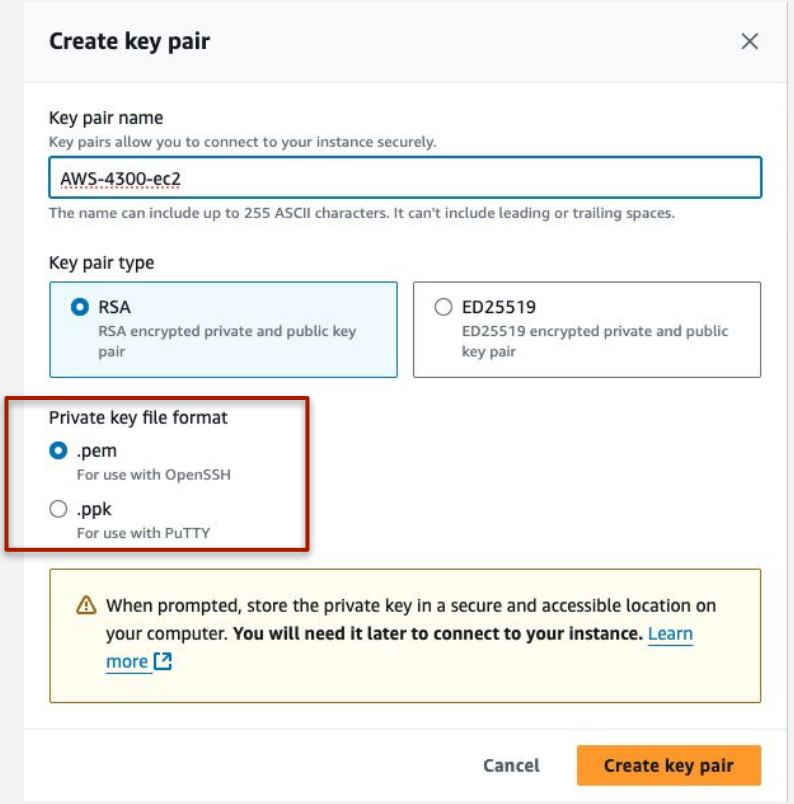




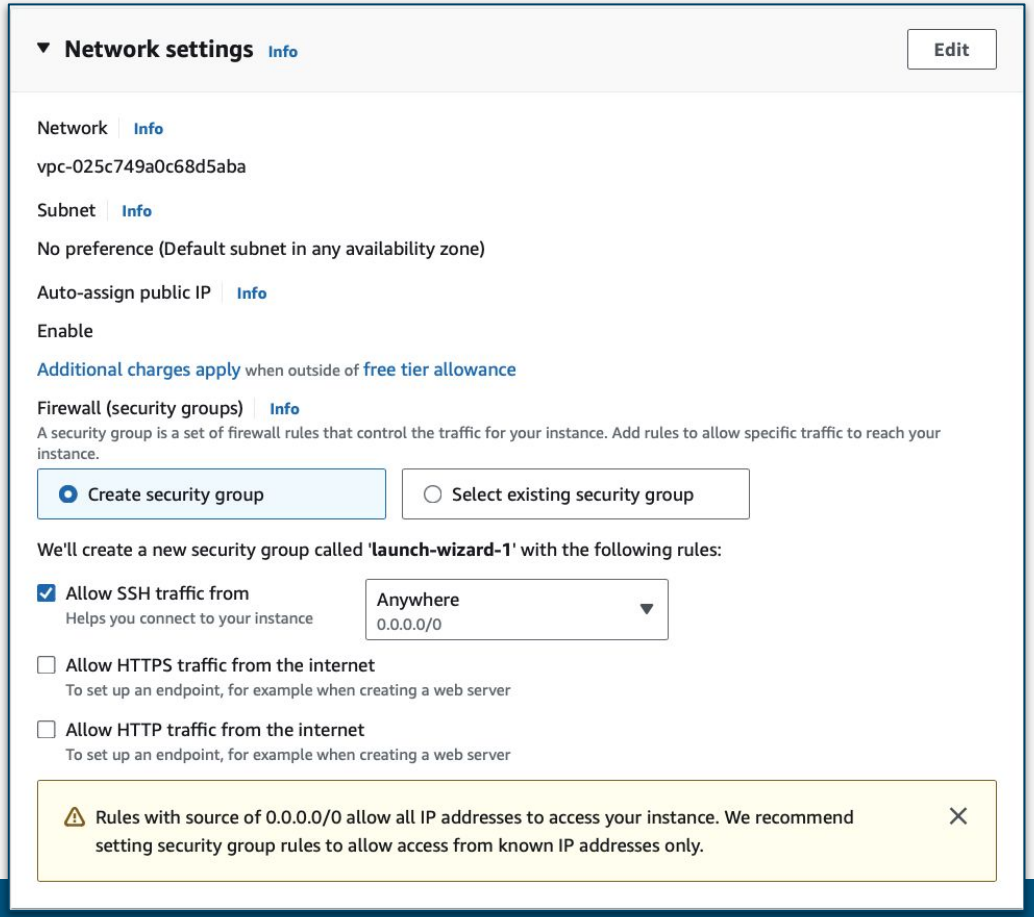
Let’s Spin Up an EC2 Instance

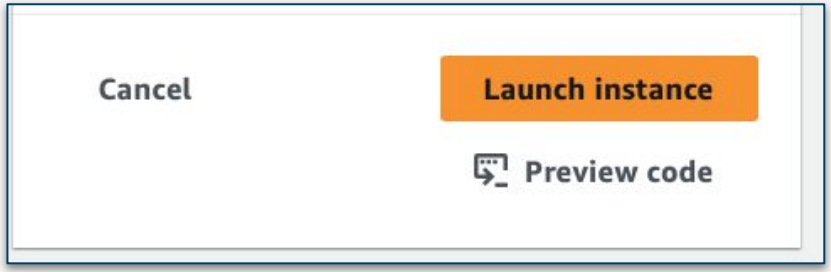






Let’s Spin Up an EC2 Instance





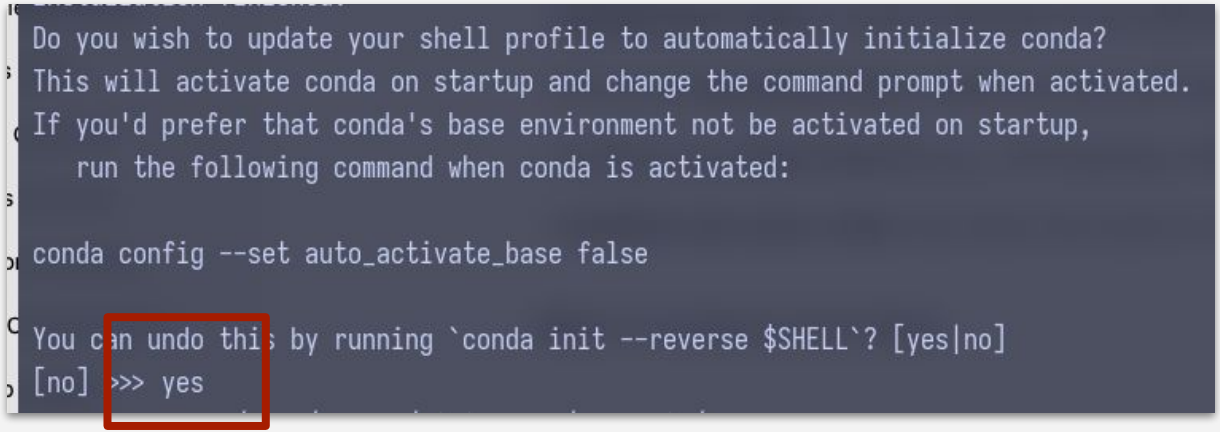
Ubuntu VM Commands

- Initial user is **ubuntu**
- Access super user commands with **sudo**
- Package manager is **apt**
  - kind of like Homebrew or Choco
- Update the packages installed
  - sudo apt update; sudo apt upgrade

MiniConda on EC2

Make sure you're logged in to your EC2 instance

- Let's install MiniConda
  - curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
  - bash ./Miniconda3-latest-Linux-x86_64.sh



Installing & Using Streamlit

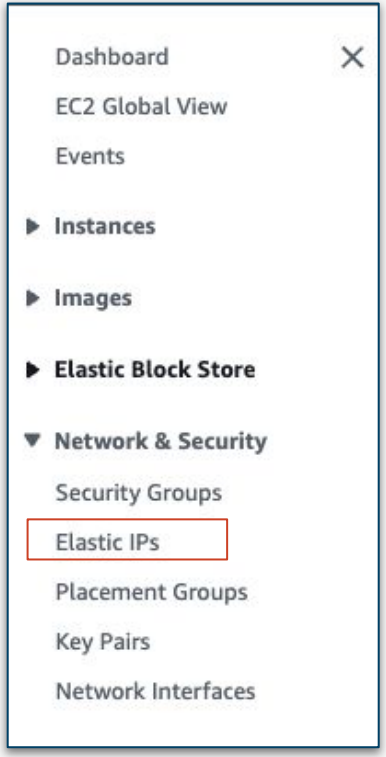
- Log out of your EC2 instance and log back in
- Make sure pip is now available:
  - pip --version
- Install Streamlit and sklearn
  - pip install streamlit scikit-learn
- Make a directory for a small web app
  - mkdir web
  - cd web

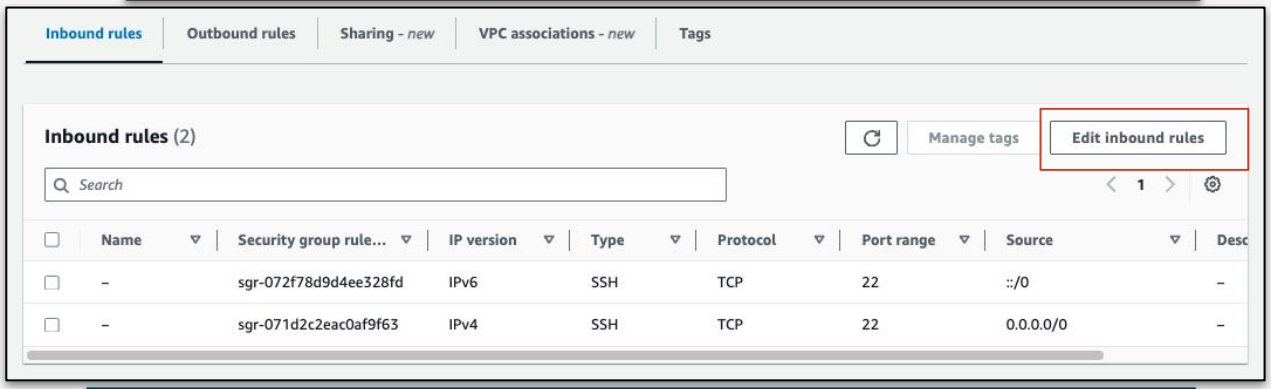
Basic Streamlit App

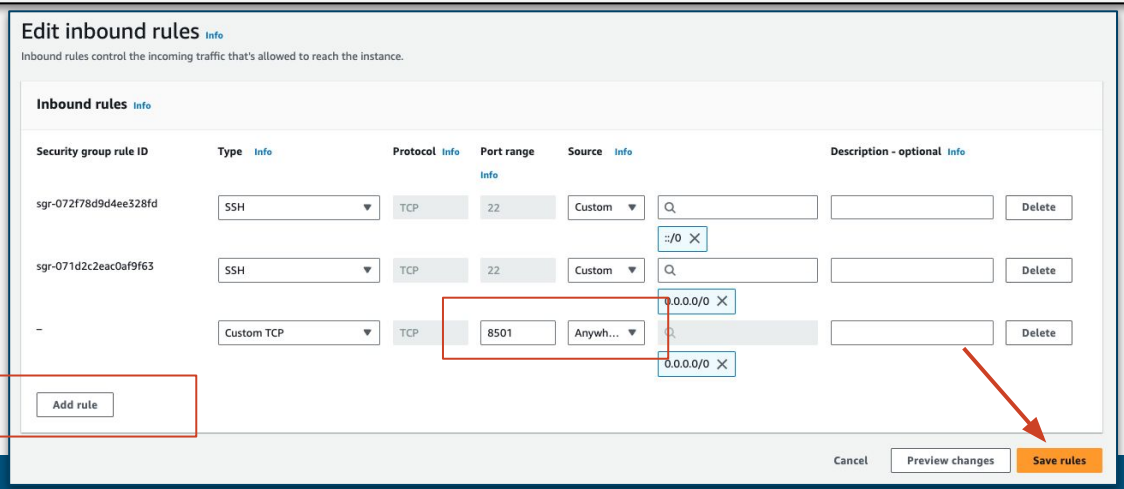
```
import streamlit as st

def main():
    st.title("Welcome to my Streamlit App")

    st.write("## Data Sets")
    st.write("""
        - data set 01
        - data set 02
        - data set 03
    """)

    st.write("\n")
    st.write("## Goodbye!")

if __name__ == "__main__":
    main()
```

- `nano test.py`
- Add code on left
- `ctrl-x` to save and exit
- `streamlit run test.py`

Opening Up The Streamlit Port

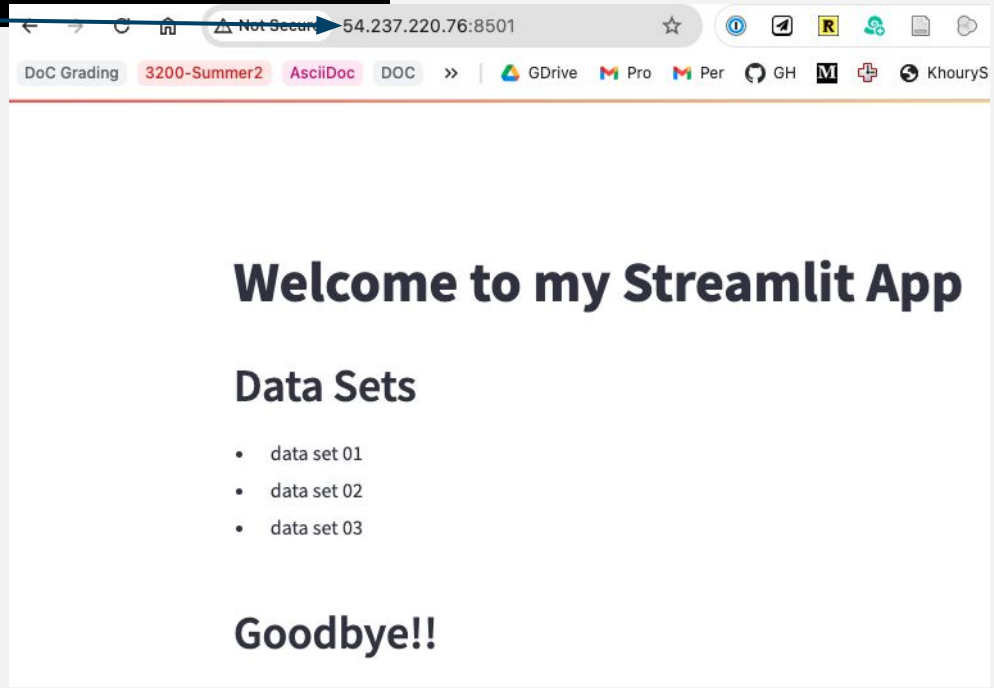






In a Browser

```
(base) ubuntu@ip-172-31-91-161:~/web$ streamlit run test.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.31.91.161:8501
  External URL: http://54.237.220.76:8501
```



AWS Lambda

Lambdas

- Lambdas provide *serverless computing*
- Automatically run code in response to **events**.
- Relieves you from having to manager servers - only worry about the code
- You only pay for **execution time**, not for idle compute time (different from EC2)

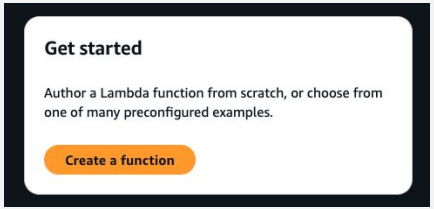
Lambda Features

- **Event-driven execution** - can be triggered by many different events in AWS
- Supports a large number of runtimes… Python, Java, Node.js, etc
- HIGHLY integrated with other AWS services
- Extremely scalable and can rapidly adjust to demands

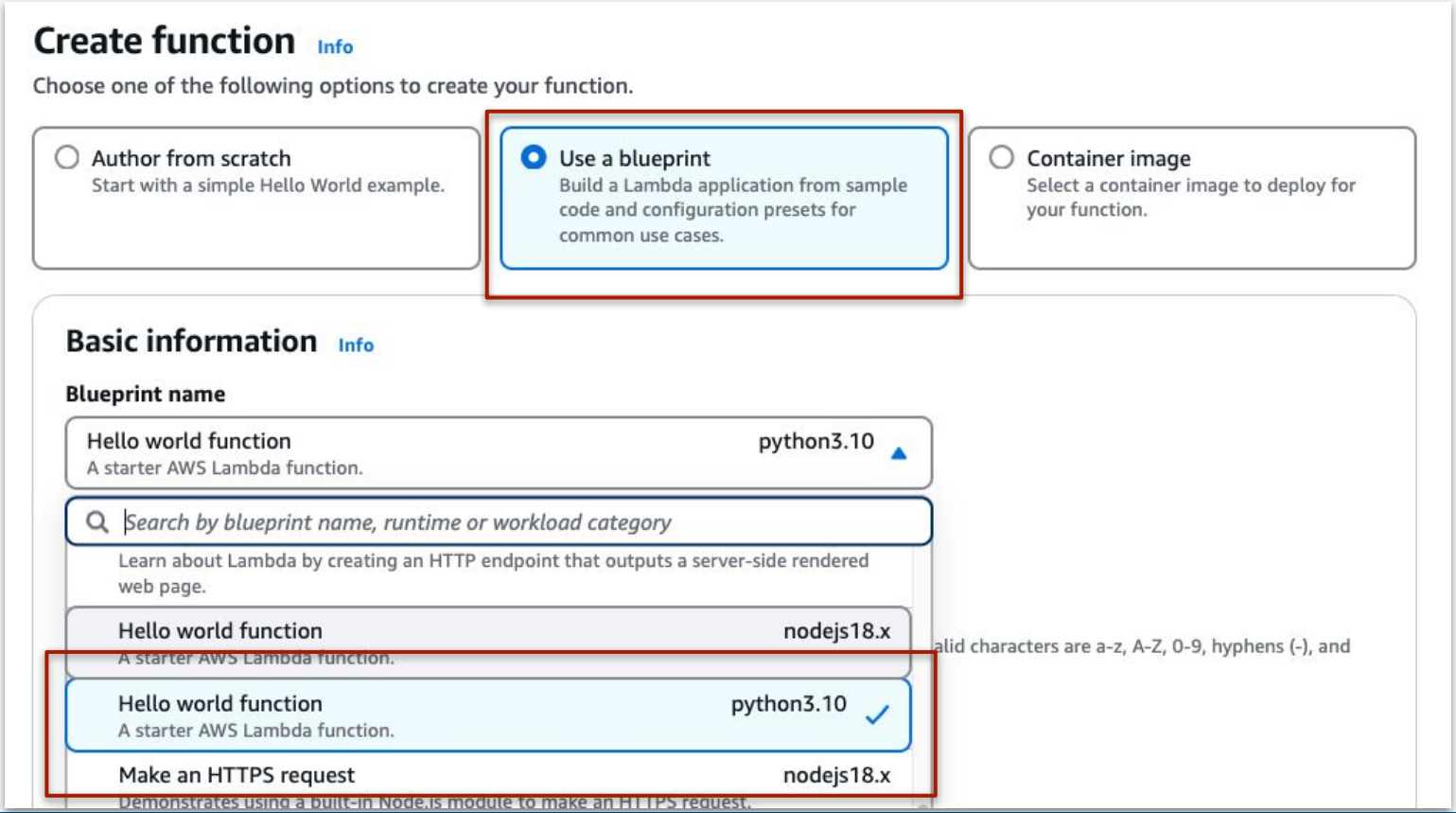
How it Works

- Add/upload your code through AWS MGMT console
- Configure event source(s)
- Watch your Lambda run when one of the event sources fires an event

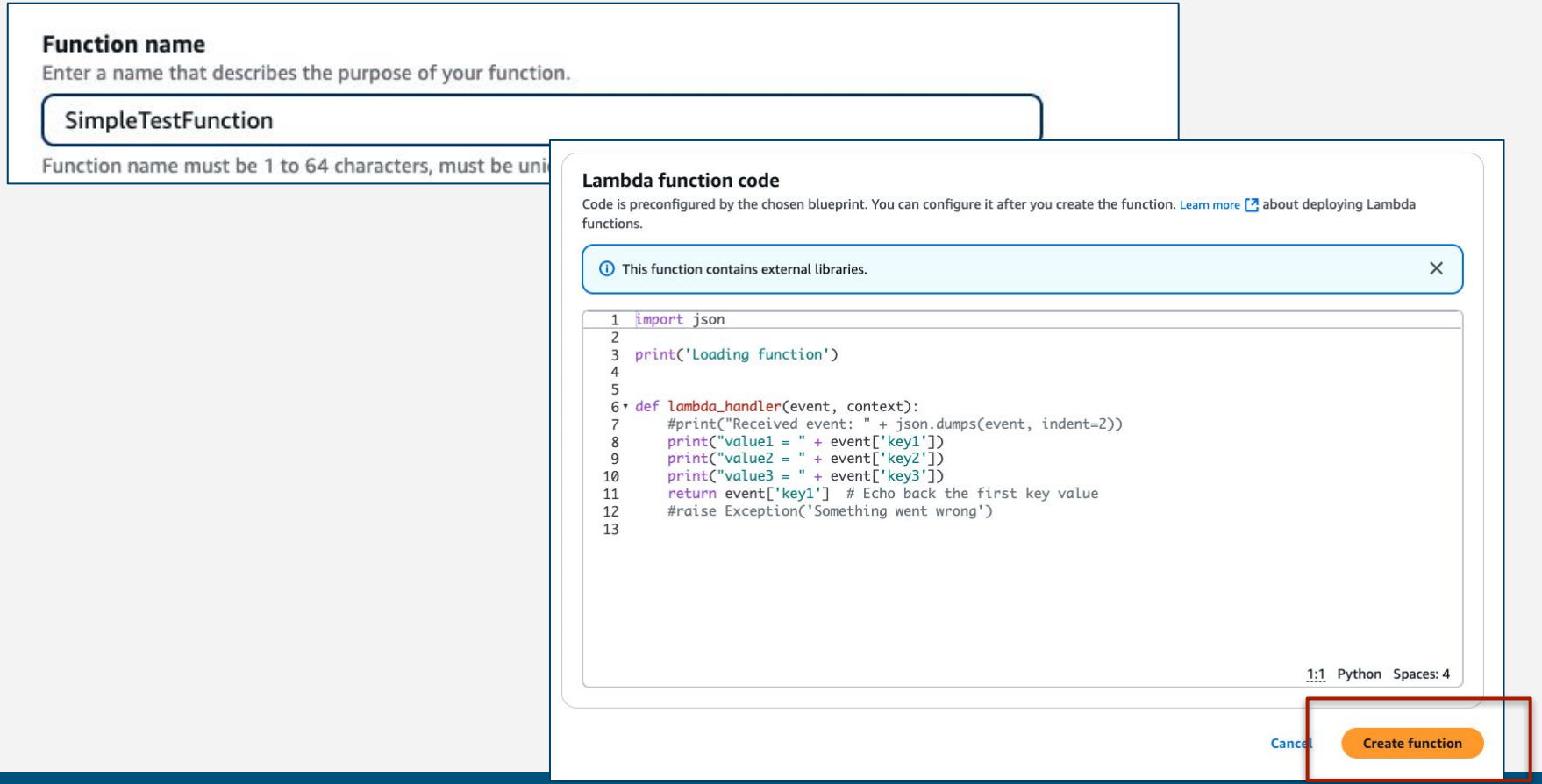
Let’s Make One



Making a Lambda

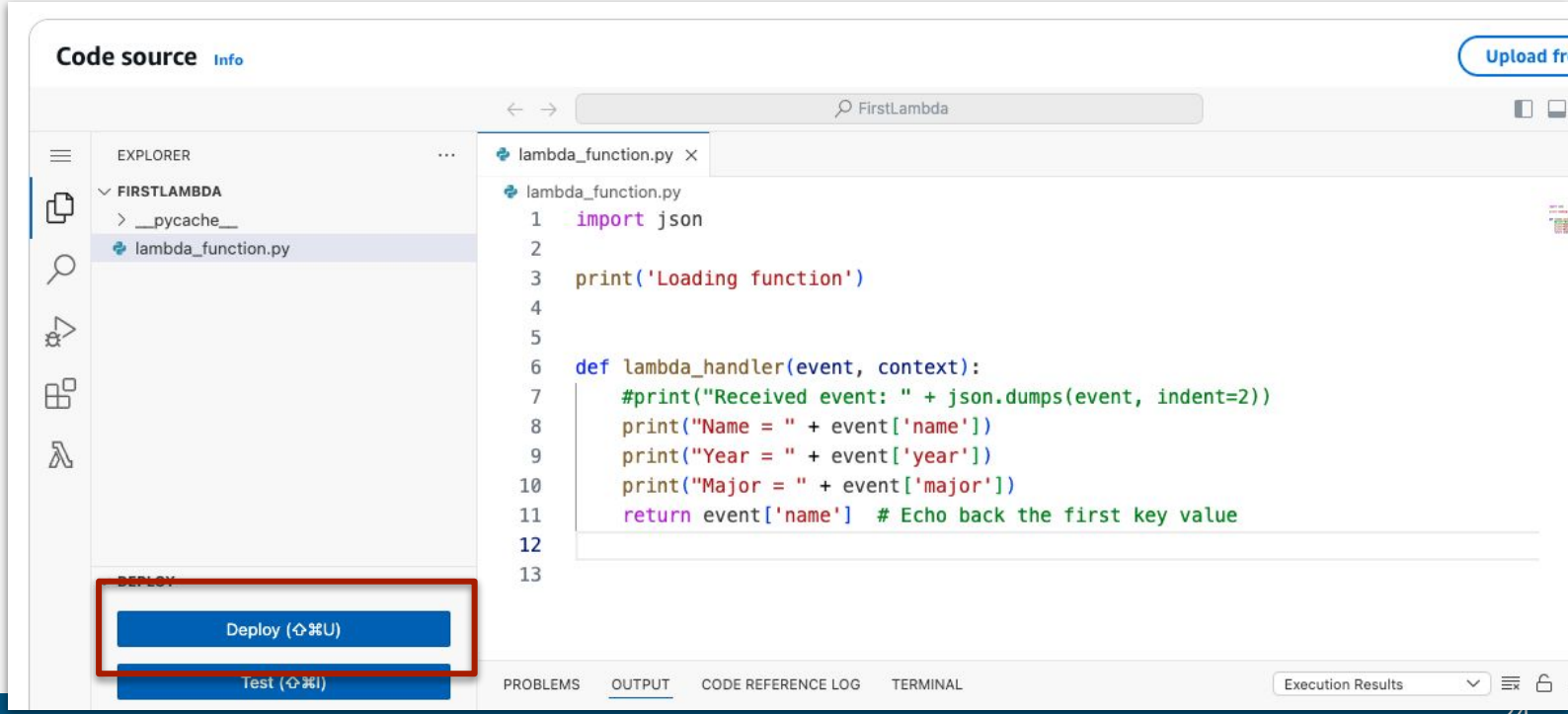


Creating a Function

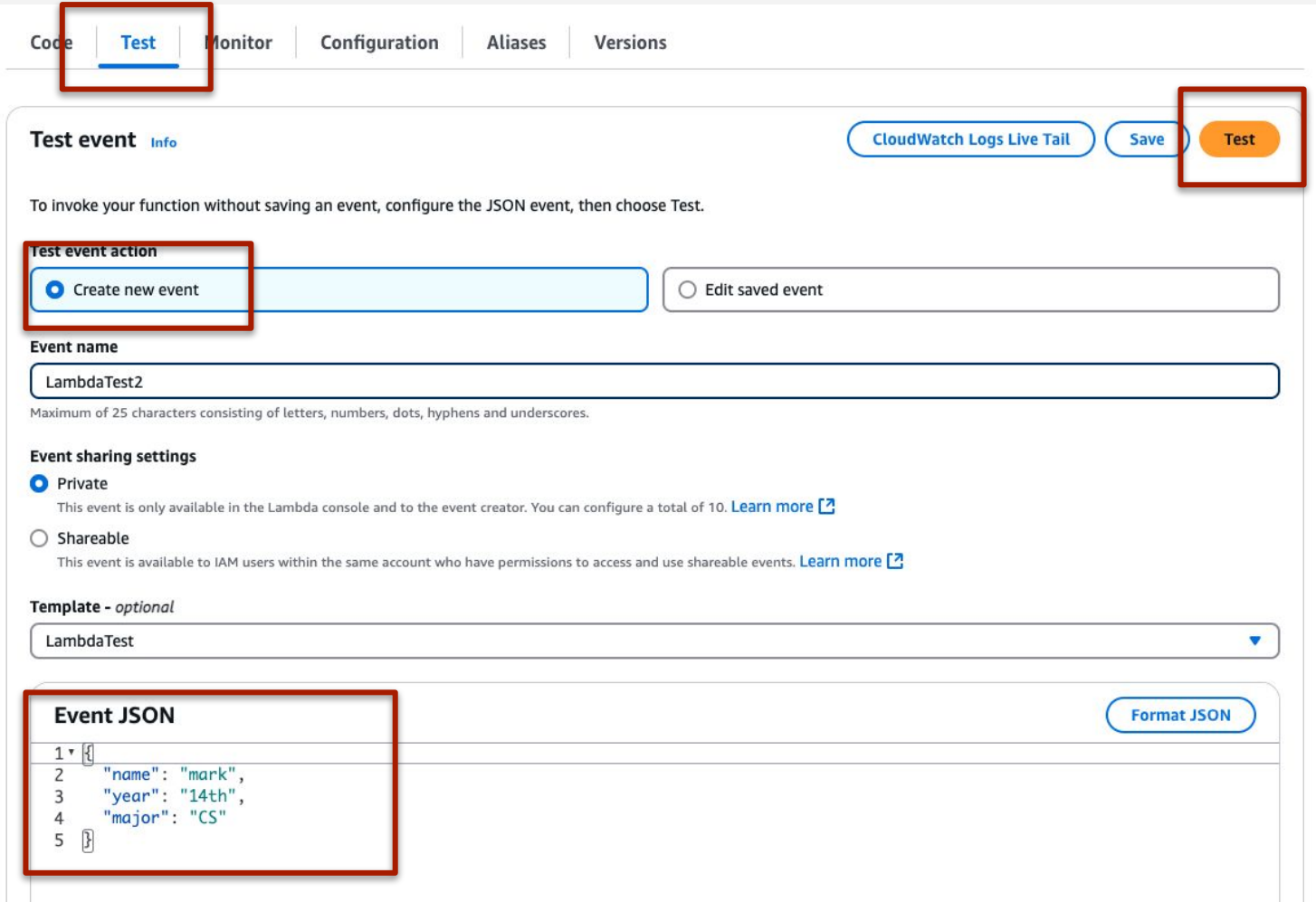


Sample Code

- Edit the code
- Deploy the code!



Test It


Code
Test
Monitor
Configuration
Aliases
Versions
Test event Info
CloudWatch Logs Live Tail
Save
Test
To invoke your function without saving an event, configure the JSON event, then choose Test.
Test event action
Create new event
Edit saved event
Event name
LambdaTest2
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more
Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more
Template - optional
LambdaTest
Event JSON
Format JSON
1 {
2 "name": "mark",
3 "year": "14th",
4 "major": "CS"
5 }

??