Designing an Autonomous Workbench for Data Science on AWS

Dipen Chawla

Data Engineer, Episource LLC

Need for a dedicated ML Workbench

Shortcomings of shared dev environments

Jupyter notebook servers on EC2

- Limited scalability
- Requires DevOps intervention
- Cannot access secure data with role-based access
- No auto-shutdowns
- Difficult to personalise

What we needed in an ML workbench

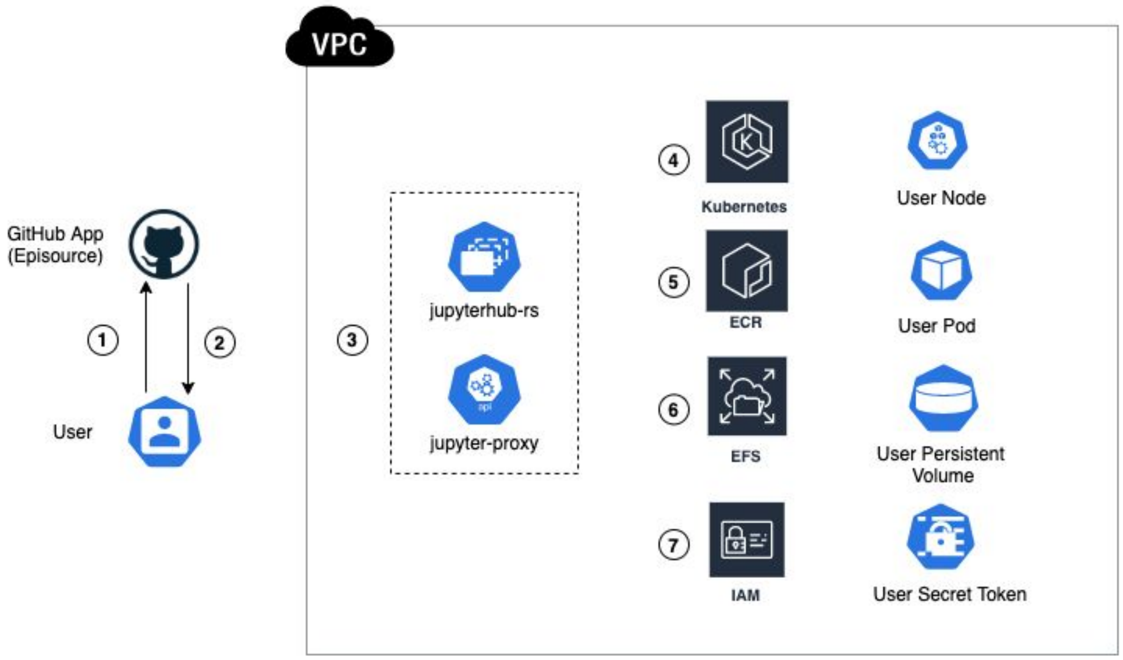
Key Features

- Must not reinvent the wheel
- Scale sensibly
- Personalised user pods
- Users could be given access to shared datasets
- Encryption and closed-door security

Arriving at the right picture

Using Jupyterhub on Kubernetes

- Data Scientists accustomed to Jupyter ecosystem + lots of ML extensions!
- Provided extensive number of integrations via spawners
- Decided to go ahead with AWS EKS
- Kubespawner in zero-to-jupyterhub fit our use-case
- Added customisations like Auto-Shutdown, IAM roles for service accounts and volume mounting via AWS EFS



1. User logs into VPN and sends authenticates via Github App
2. GitHub App checks validity of user and redirects user to Server Selection Screen.
3. User directly interacts with the frontend of Jupyterhub - size selection, manual shutdown.
4. EKS Autoscaler provisions an EKS node
5. EKS uses ECR to pull the image for user pod
6. User folder from EFS to setup user filesystem
7. IAM secret token is embedded to use AWS services

User Storage

- EFS storage mounted to each user pod at startup for user filesystem
- User files persisted even after each session terminated
- Additional datasets (present as Persistent Volumes) can be mounted to user pods, if required
- EFS and Persistent Volumes are encrypted at all times

User Management

- Workbench authentication via GitHub
- Ensures only organization members of Episource can access
- Further, an app-level firewall to restrict the access to members of ML team
- Jupyterhub admins can view activity and have access to individual folders

User Access Control

- Configure user pods to use Kubernetes Service Accounts
- Each SA is configured to use a dedicated IAM role
- Ensures that each user can access only the data they are authorised to

Impact of the workbench on ML development cycle

How the workbench has helped ML teams

- Work with familiar ecosystem across all stages of development
- Can perform high compute analysis jobs
- Hypotheses take lesser time to turn into features
- Auto shutdown and RBAC = less monitoring overhead
- Can share files / data by moving them to shared folders

Key Takeaways

What we’ve learned so far

- Standing on the shoulders of giants - ❤️ for open source
- Autonomy is the key when building tools to help your ML team to succeed
- Your design may not work at first - keep iterating !
- Never look past security aspects of your architecture

Thank you!

- @dipen_chawla