

PyData Indy

The Current State of Software Development

Justin Bolles, @jbolles | @KSM_Consulting

Cross-Functional Code

The Peak of Productivity

- First answer was Java
 - Write once; Run everywhere.
 - JVM was resource hog.
 - Required correct version of Java Client.
 - Didn't make move to Internet well.

Cross Functional Code

Peak of Productivity

- iOS vs Android
 - Cross platform development with ReactNative, Appcellerator, etc
 - Required teams to maintain three code bases
 - Cross-platform code
 - Native code for iOS and Android
 - Large organizations dumped cross functional codebases for native (AirBNB: <https://medium.com/airbnb-engineering/sunsetting-react-native-1868ba28e30a>)

Cross Functional Code

Peak of Productivity

- JavaScript for Everyone!
 - Make JavaScript both front-end and back-end language so front-end devs can do back-end work!
 - Back-end is different mindset.
 - JavaScript dependencies are complex.
 - Still too many JavaScript Frameworks (ViewJS, Angular, ReactJS, Meteor, Ember, Backbone, Aurelia, Polymer, Mithrill)
 - JavaScript is not a backend language! Promises are terrible!

Cross-Functional Code

Why?

- Deployments and Maintenance were expensive.
- More environments complicate deployments.
- There is a better way!

I NEED TO KNOW WHY MOVING
OUR APP TO THE CLOUD DIDN'T
AUTOMATICALLY SOLVE ALL OUR
PROBLEMS.



Dilbert.com @ScottAdamsSays

YOU WOULDN'T
LET ME RE-
ARCHITECT THE
APP TO BE
CLOUD-NATIVE.

JUST PUT IT
IN
CONTAINERS.



11-08-17 © 2017 Scott Adams, Inc. Dist. by Andrews McMeel

YOU CAN'T
SOLVE A
PROBLEM JUST
BY SAYING
TECHY THINGS. KUBERNETES.



Containers!

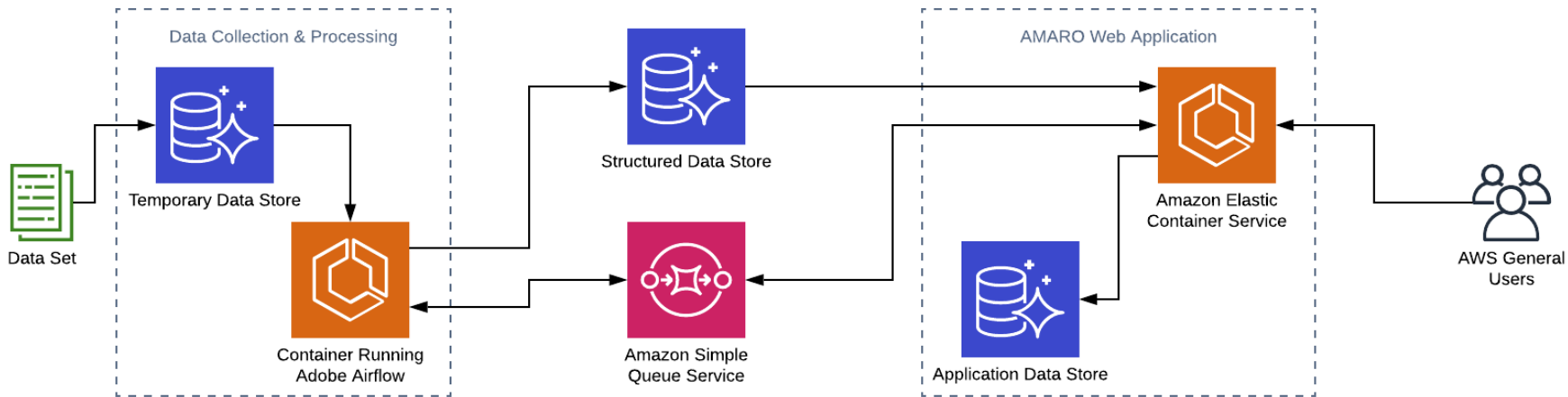
Your boss's new favorite buzzword!

Containers!

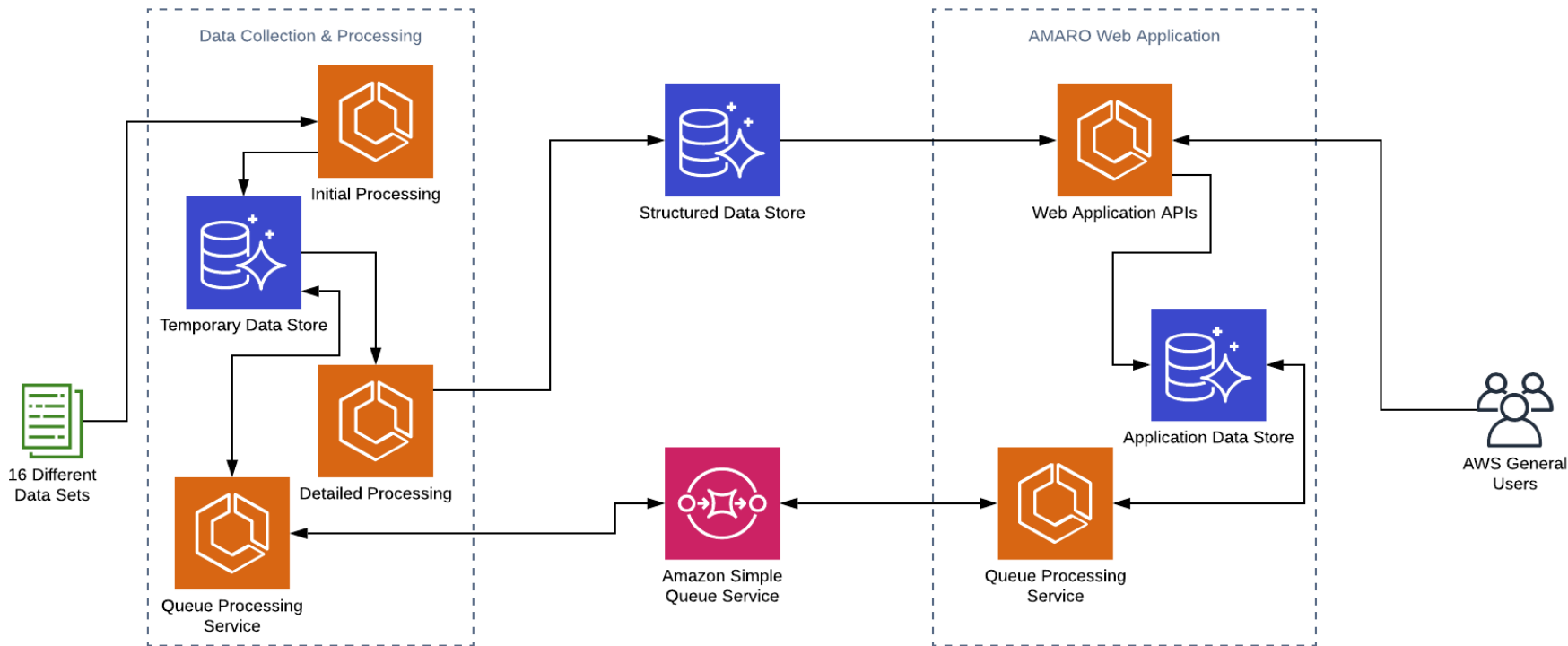
Let Languages Specialize

- JavaScript is great for front-end.
- Ruby on Rails, .NET great for middle-tier APIs
- Python is great for data collection and processing.
- **NOT EVERYONE HAS TO DO EVERYTHING!**

Example Infrastructure



Actual Infrastructure



Reusable Code

Ideal State

Write Once, Use Everywhere

- Writing good code is time consuming and therefore expensive.
- Constant Refactoring of Object-Oriented Code
- DRY development methodology
- Create library of reusable objects.

Pipe and Filter

Ultimate DRY Methodology

- Keep snippets of code small and focused.
- Standardize inputs and outputs of all snippets of code.
- String the snippets together in different arrangements to accomplish goals.
- Share snippets across different projects for ultimate reusability.








































































































Apache Airflow

The scheduler does not appear to be running. Last heartbeat was received 18 seconds ago.

The DAGs list may not update, and new tasks will not be scheduled.

DAGs

Search:

		DAG	Schedule	Owner	Recent Tasks 	Last Run 	DAG Runs 	Links
		bloomberg	0:01:00	me	 	2019-10-08 00:00 	 	         
		ciqBackup	0:01:00	me		2019-10-10 18:06 		         
		welcome_to_amaro	0 * * * *	me	 	2019-10-08 21:51 	  	         
		welcome_to_amaro_100	1 day, 0:00:00	me	 	2017-06-17 02:00 	 	         
		welcome_to_amaro_3	0 * * * *	me	 		 	         
		welcome_to_amaro_44	0 * * * *	me	 		 	         

Continuous Deployment

The DevOps Buzzword

What is DevOps?

- Continuous Delivery
 - New features rolled out daily.
 - Bugs are fixed before a majority of users find them.
- Reduce cost of implementing new features.
- Increase **customer happiness** by increasing usability of application.

DevOps can do MORE

Dynamic Environments

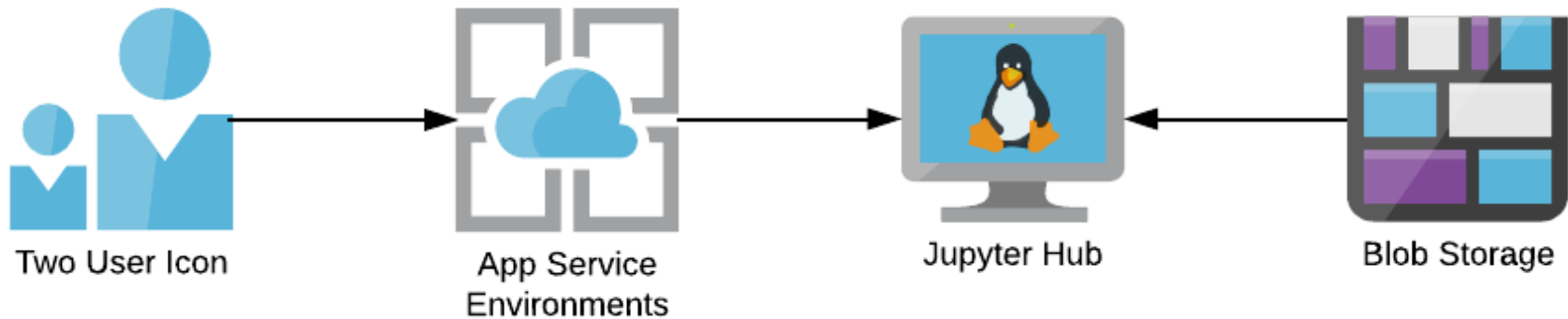
- Every action taken by every worker in today's workplace is a data generation event.
- Today's workforce creates an enormous amount of data from which organizations can make decisions.
- Infrastructure as code allows environments to be created on demand.
- Datasets that were previously off limits can be analyzed in a secure environment, allowing organizations to gain insights.

Dynamic Environments

Allowing analysts to interact with sensitive data

- Client requirements:
 - Allow users to request access to data.
 - If approved, allow dataset(s) to be hooked to custom Jupyter Notebook instance for analysis.
 - No data leaves the platform.

Solution

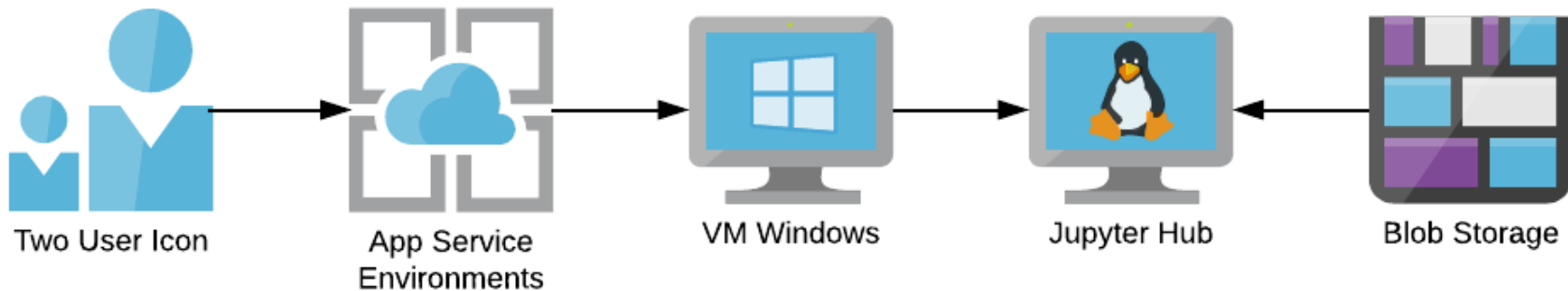


Dynamic Environments

That was too easy.

- Actually, users would like to use tools like Excel and Tableau.
- Also, they want to use their own licenses.
- Also, the client would like to approve any reports created from the data so the users can export results.

Diagram of Final Solution



Wrapping Up!

Better Code of All!

- Containers reduce cost of deployment and allow for language interaction. Let languages specialize.
- Apache Airflow allows for small snippets of code to be reusable, even across projects.
- DevOps is more than a buzzword! Infrastructure as code unlocks the power of big data.

Questions?

THANK YOU!

Justin Bolles

Twitter: @jbolles