

CLOUDCAMP: AUTOMATING THE DEPLOYMENT AND MANAGEMENT OF CLOUD SERVICES

Anirban Bhattacharjee*, Yogesh Barve*, Aniruddha Gokhale* &
Takayuki Kuroda†



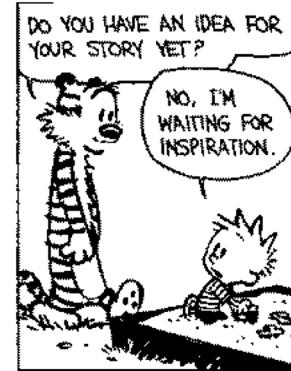
*Vanderbilt University, USA & †NEC Corporation, Kawasaki, Kanagawa, Japan

Work Supported by NEC Corporation, Japan and NSF US Ignite CNS #1531079

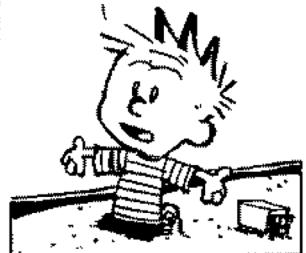
D g r i u C o P

ROADMAP

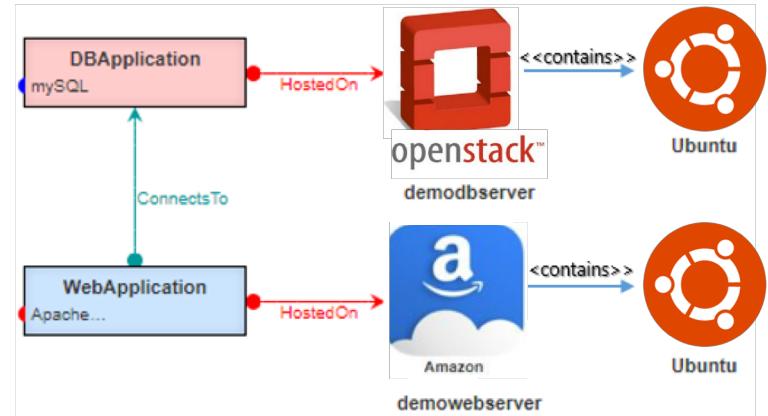
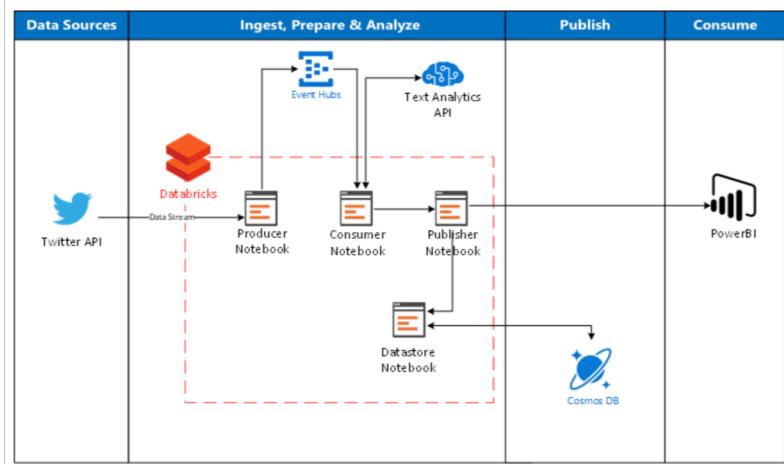
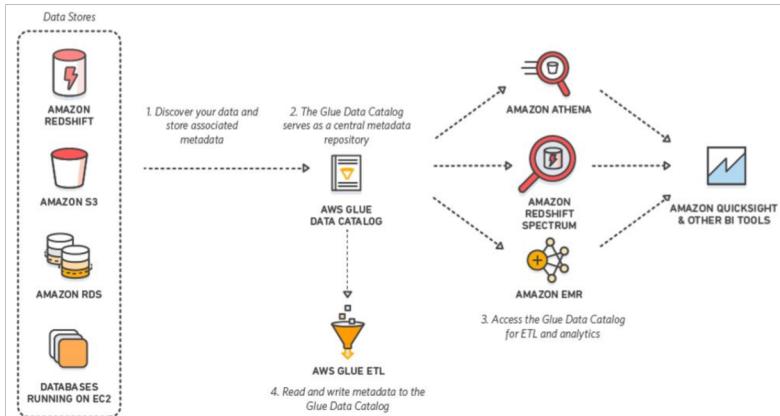
- Motivation
- Challenges
- Objectives
- CloudCAMP Solution
- CloudCAMP Architecture and Design
- Evaluation for Service Deployment
- Conclusion



YOU CAN'T JUST TURN ON CREATIVITY LIKE A FAUCET. YOU HAVE TO BE IN THE RIGHT MOOD.



MOTIVATION



e.g., Figure 1 shows a service topology consisting of two connected software stacks, i.e., Web front-end and MySQL database

Composite Services need be deployed on heterogeneous cloud platform.



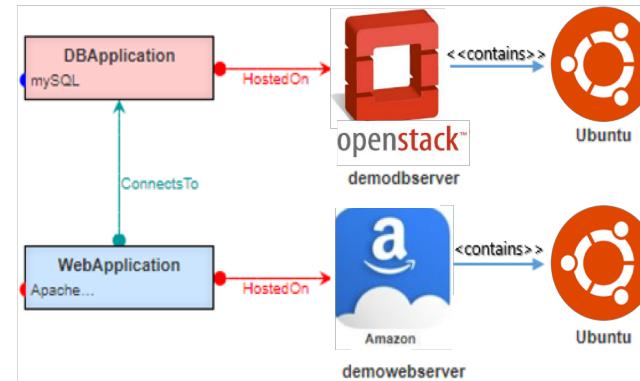
CHALLENGES & SOLUTION NEEDS

- Reduction in specification details of Cloud Infrastructure and Application Components
 - Identifying the commonalities of the cloud provisioning stacks.
 - The minimal number of variability points are the user inputs e.g. VM name, service name, VM flavor etc.
- Auto-completion for Infrastructure Provisioning using model transformation
- Capture variability and complexities for self-service provisioning automation in Knowledge Base
- Support for Continuous Delivery, and Migration of service components
- Complexities in Automated Migration of the Service Components across heterogeneous Providers

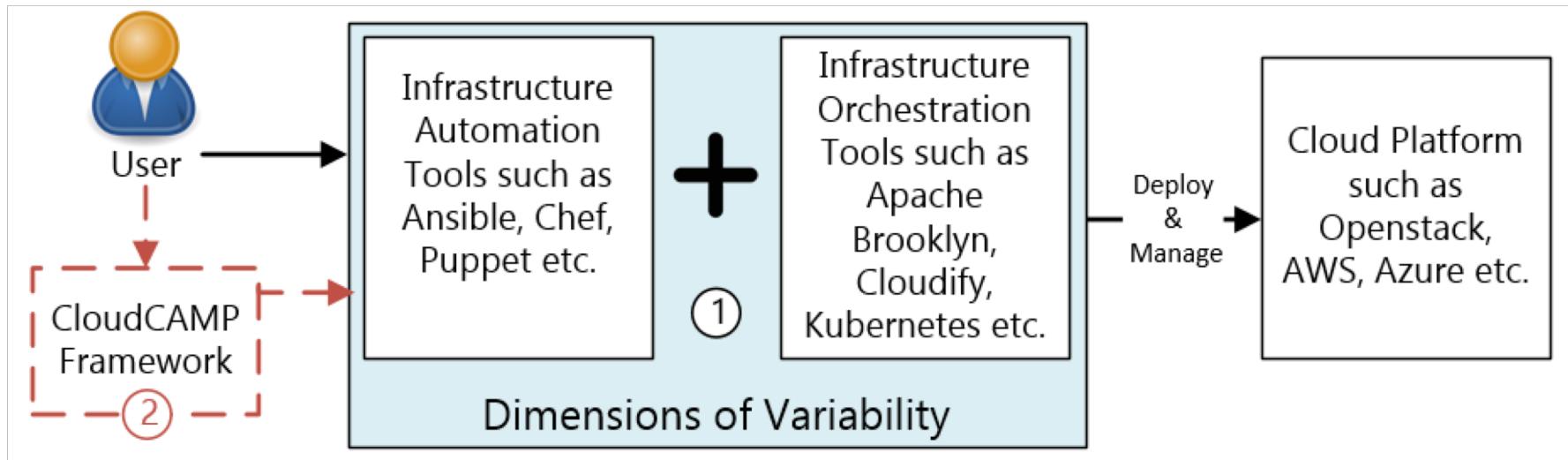


OBJECTIVES

- Web-based self-service Infrastructure Provisioning System
 - Model Business logic of the service in WebGME tool as shown in Figure 1
- Automated provisioning of full-blown applications infrastructure
- Generate the Infrastructure-as-Code (IAC) solution from business model
- Ensure Interoperability using OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA)
- Verify the correctness of the IAC



CLOUDCAMP SOLUTION

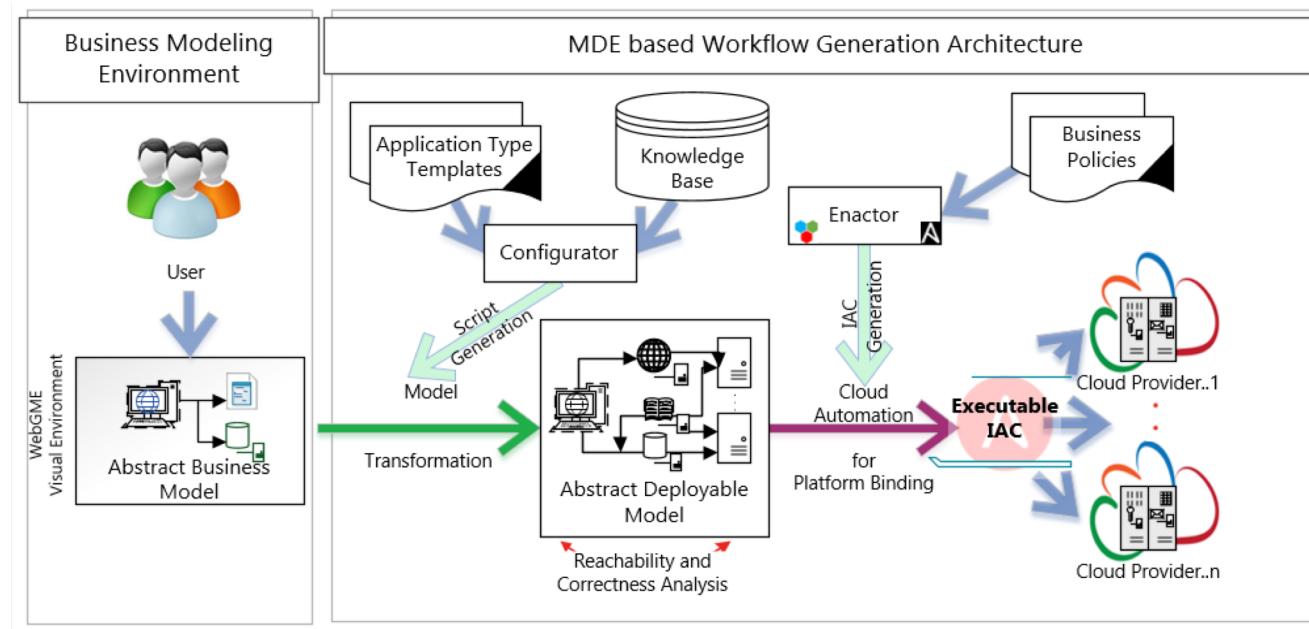


Box 1 depicts the responsibilities of deployers to write scripts in automation tools to provision infrastructure for service components.

Box 2 depicts CloudCAMP contribution to abstract and automate whole infrastructure design solutions using MDE and UI.



CLOUDCAMP ARCHITECTURE & DESIGN

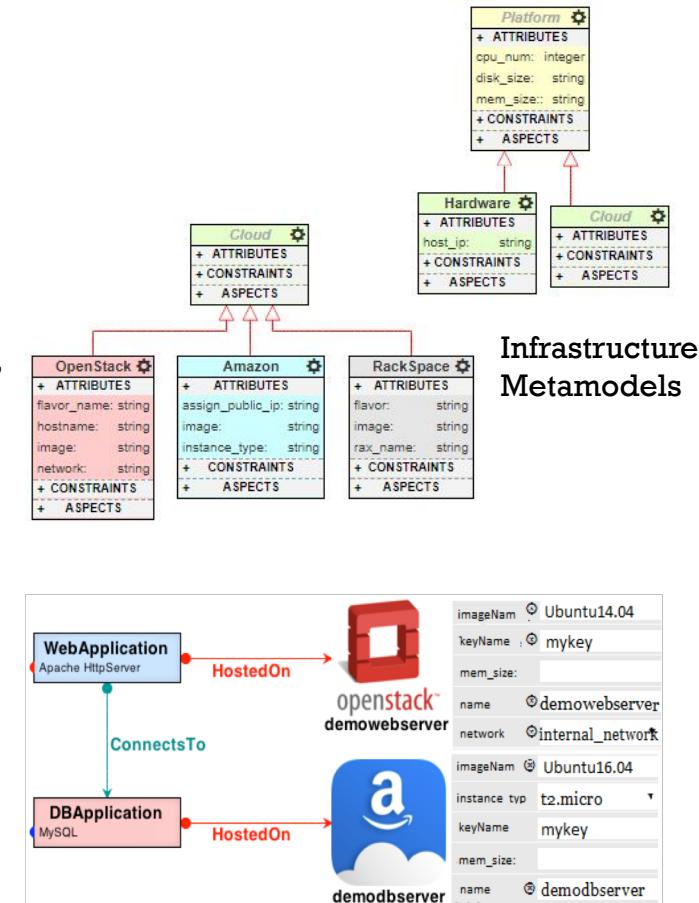


- **Abstraction of Model:** Deployers provision by selecting only business-relevant components.
- **Configurator:** Transforms abstract service components to Ansible-specific automation tasks using DSML.
- **Enactor:** Generates IAC by integrating automation code, cloud-specs & inter-component connection types.
- **Knowledge Base:** Software dependencies for service component types are stored in RDBMS table.



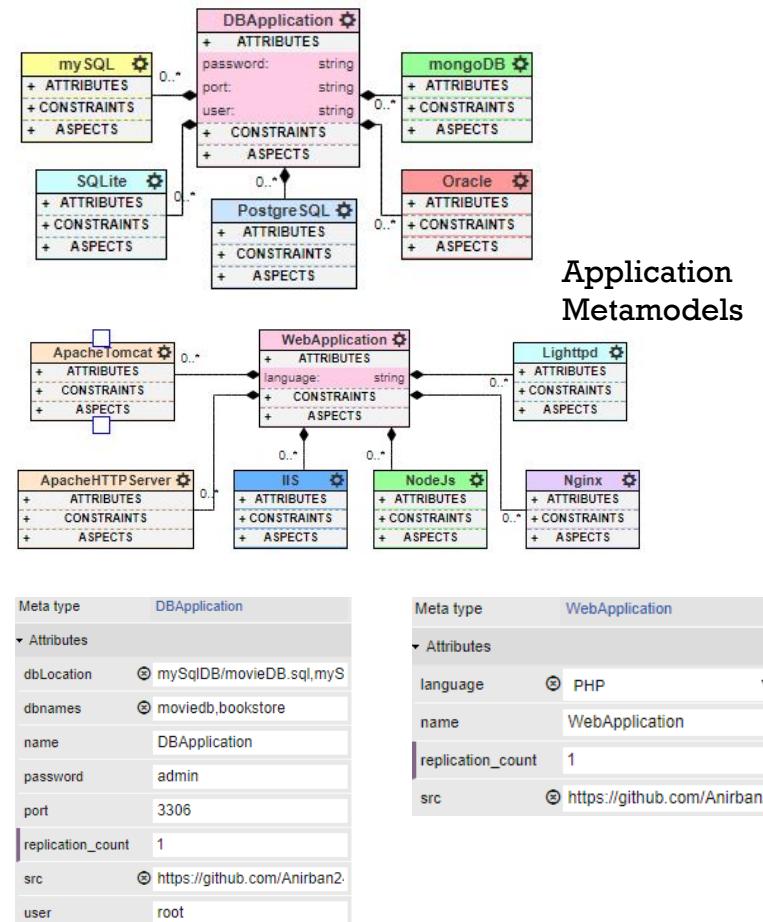
METAMODEL FOR THE CLOUD PLATFORMS

- We reverse engineered the process of hosting applications across different cloud environments
 - Captured all the commonalities and variabilities.
 - The specifications for different cloud platforms such as OpenStack, Amazon AWS, Microsoft Azure, etc for provisioning VMs
 - Various operating systems (OS) are captured.
- The variability points are also captured.
 - The deployer can select a pre-defined VM flavor, available networks, security groups, roles, and the available images, all of which are defined as variabilities in our metamodel.
 - A pre-deployed machine can be specified by providing the IP address and OS.
 - Available services and VM types for cloud platforms are pre-defined in the metamodel.

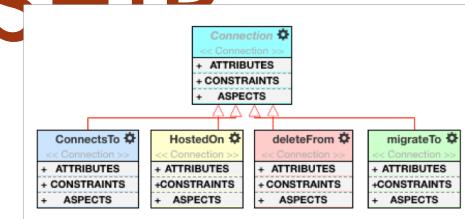


METAMODEL FOR APPLICATION COMPONENTS

- CloudCAMP provides different node types for application components.
 - Web Application,
 - Database Application,
 - DataAnalytics Application, etc.
- Metamodel enables a deployer to choose the web server attribute, language for the code, the database server attribute or the NoSQL database attributes from the provided list.
- The deployer has to specify the variable attributes to deploy the desired application component type.



DEFINING THE RELATIONSHIPS AMONG COMPONENTS



- hostedOn relationship type implies the source node type is required to be deployed on the destination node type.
- connectsTo relationship type is used for deployment ordering to relate the source node type's endpoint to the required target node type endpoint if they are dependent.
- deleteFrom connection type defines the source node type is required to be removed from the end node type.
- migrateTo connection type defines the source node type that is to be migrated to the end node type. The migrateTo relation type cannot be defined without a deleteFrom connection type.



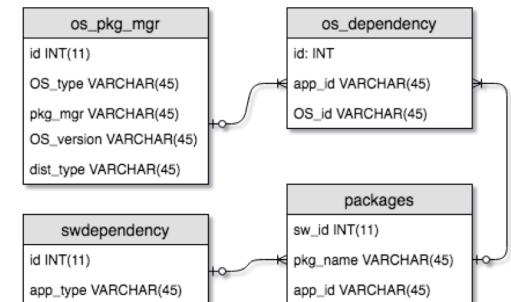
KNOWLEDGE BASE DESIGN

▪ Knowledge Base Database Design:

- The ER diagram of the knowledge base database stores the artifact sets in the knowledge base.
- We have structured it as four tables:
 1. All the operating systems, their distributions, package manager and versions in the os_pkg_manager table,
 2. All available application component types, e.g., PHP based web application, MySQL based DB applications, etc. in the swdependency table, and
 3. All the software packages needed for a particular application type are stored in the packages table.

▪ Knowledge Base Template Design:

- The knowledge base templates are designed by capturing the commonality in the application components, and comprises placeholders
- Placeholders need to be filled up by the CloudCAMP DSML by querying the knowledge base database.



sw_id	pkg_name	app_id	id	AppType
1	mysql-server	2	1	php
2	mysql-client	2	2	mysql
3	python-mysqldb	2		
4	libmysqldclient-dev	2		swdependency

packages				
id	OS_type	pkg_mngr	OS_version	distribution
1	ubuntu	apt	14.04	linux
2	centos	yum	7	linux
3	rhel	yum	7.3	linux

os_pkg_mgr				
1	ubuntu	apt	14.04	linux



EVALUATION FOR SERVICE DEPLOYMENT

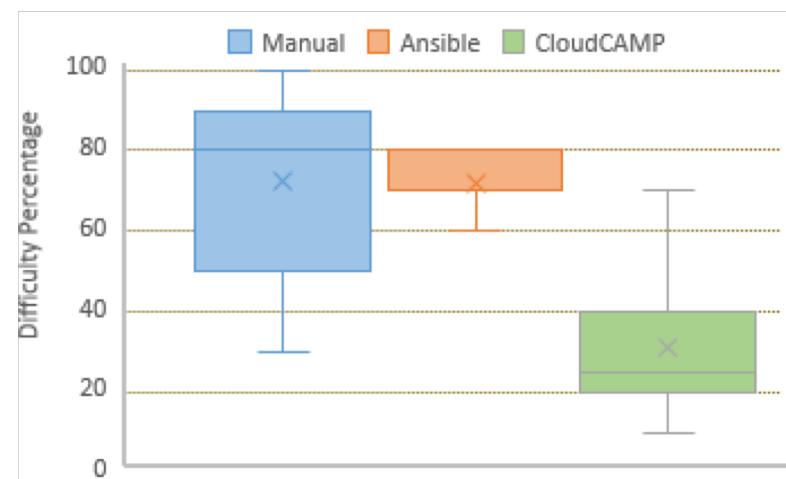
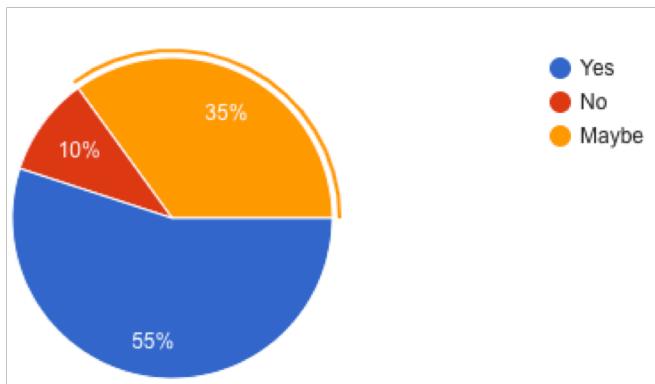
- Deployment Scenario: A simple Web and DB (Figure 1)
- Migration Scenario: Migrate the DB service component from one cloud platform to another
- A naïve user using CloudCAMP takes about 15-20 mins. Below table shows the first time scripting & manual effort:

	Lines to Write For Deployment	Deployment Time(using Scripts)(mins)	Lines to Write For Migration	Migration Time (using Scripts) (mins)
Median	300	300	550	720
Mean \pm SD	315 ± 47	516 ± 244	553 ± 142	653 ± 231

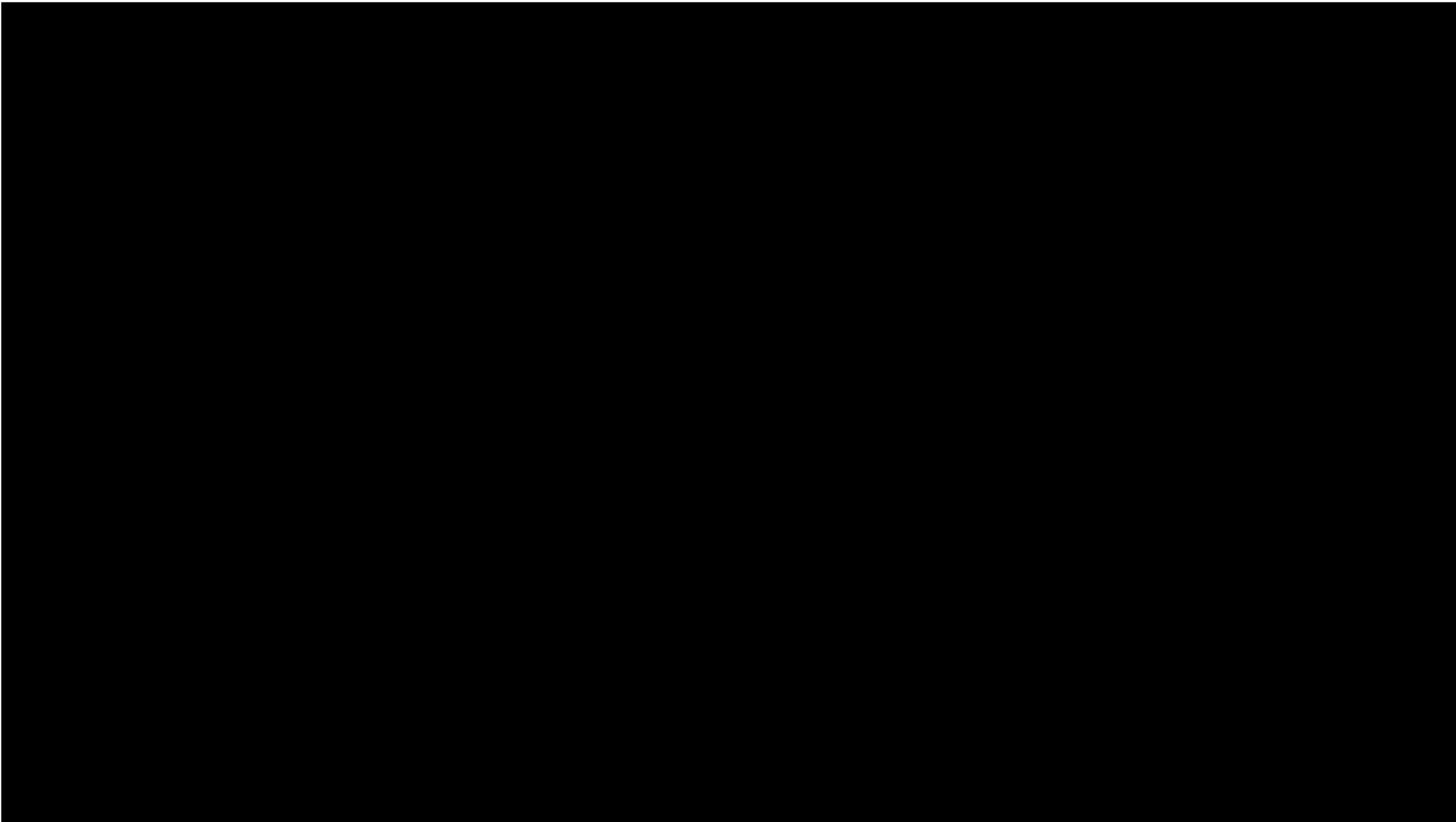


CLOUDCAMP SURVEY

- Will you use a tool like CLOUDCAMP to deploy applications in future?
- How easy was deploying PHPMySQL using a MDE tool like CLOUDCAMP compared to other approaches?



DEMO



<https://youtu.be/Dc8rfNQIo3k>



QUESTIONS?



Thank You!

