

Multi-cloud management with IBM Cloud Automation Manager

Strategy and Offering

Danny Mace, Director of Offering Management

October 10, 2017

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

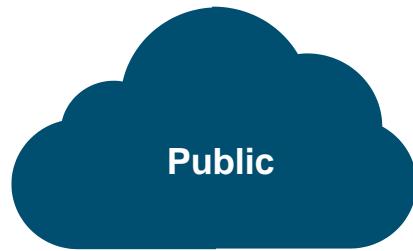
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

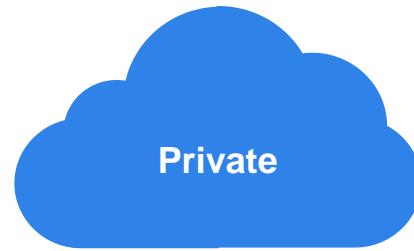
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Enterprises are increasingly turning to multi-cloud strategies to accelerate innovation

The average enterprise is running 5 or more clouds ⁽¹⁾



Significant growth in hybrid cloud adoption ⁽²⁾



Challenges

- Managing complexity - multiple clouds, tools, and services
- Optimizing solutions - workload placement, costs
- Managing compliance & security
- Maintaining low friction oversight, governance and control

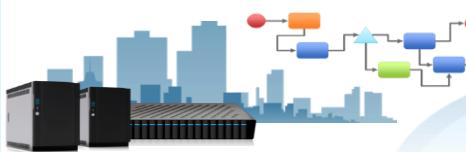
(1) Forbes & Gartner

(2) RightScale Cloud Computing Trends: 2016 State of the Cloud Survey

Clients require multi-cloud management across a spectrum of workloads

Organizations are embracing new business models and disruptive technologies to be competitive, meet business need, and innovate

Cloud Enabled



Extend

Optimize Investments

Focus on Integration & Scale

- Expose & monetize existing services
- Extend existing workloads on Cloud
- Operational automation
- Analytics-based visibility on progress

Cloud Native



Drives Need

Innovate

Integration



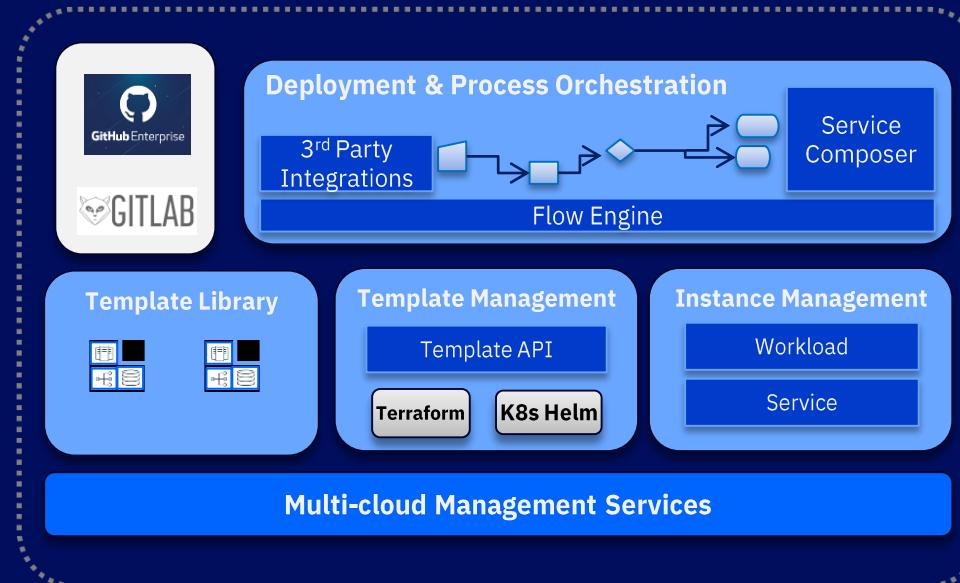
Hybrid Cloud

Focus on Speed & Agility

- Cloud First assembly of services
- Agile DevOps model
- Continuous delivery
- User first delivery model

Cloud Automation Manager: Use cases

- **Compose & Orchestrate complex application environments** into easy to consume cloud services that can be delivered to a DevOps tool chain or published into a catalog
- **Automate and standardize** delivery of infrastructure and application environments consistently in the clouds of your choice
- **Manage workload and service instance lifecycle** across multiple clouds with topology centric controls



Full stack multi-cloud automation built with open technology

- Automate provisioning of **bare-metal servers, VMs, cloud native services, Docker containers and complex application stacks**
- Provision workloads with **Terraform in IBM Cloud, VMware, AWS EC2, Azure, PowerVC and OpenStack**
- **Increase productivity** with a **consistent, easy to use and easy to understand user interface**
- Or use developer friendly scriptable **REST APIs**

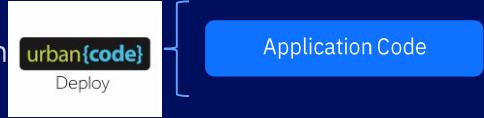
Process Integration

Business processes
automation



Application Delivery

Integration with application
delivery tool chains



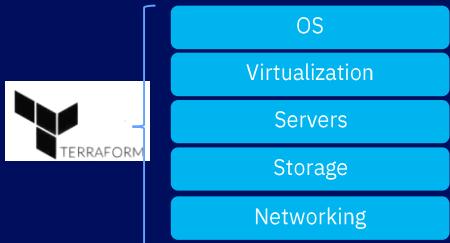
Software Configuration Management

Based on Chef industry
standard to leverage skills
and existing automation



Infrastructure automation

Based on Terraform open
source for a broad ecosystem

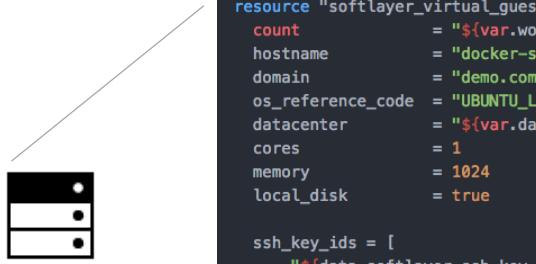


Automating Infrastructure with Terraform

Terraform codifies cloud APIs into a declarative text language used to describe your cloud resources

- Describe cloud state in a text based configuration file
 - *Store in a version control system*
- ‘Plan’ the configuration to do a test dry run
 - *The plan tells you what will be destroyed, added, or modified*
- ‘Apply’ the configuration to make the configuration real
 - *The apply causes the Terraform engine to invoke cloud provider APIs*
- Use *plan* and *apply* to update running infrastructure
 - *Modify configuration file (new version, cpu, memory, number of VMs, etc)*
 - *Manually ‘taint’ selected resources to force full restart*
 - *Plan then Apply to make the change real*

Managing Cloud Infrastructure as Code



```
resource "softlayer_virtual_guest" "worker" {  
    count          = "${var.worker_count}"  
    hostname       = "docker-swarm-worker${count.index}"  
    domain         = "demo.com"  
    os_reference_code = "UBUNTU_LATEST"  
    datacenter     = "${var.datacenter}"  
    cores          = 1  
    memory         = 1024  
    local_disk     = true  
  
    ssh_key_ids = [  
        "${data.softlayer_ssh_key.my_key.id}"  
    ]  
}
```



Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools

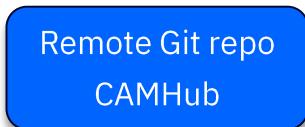
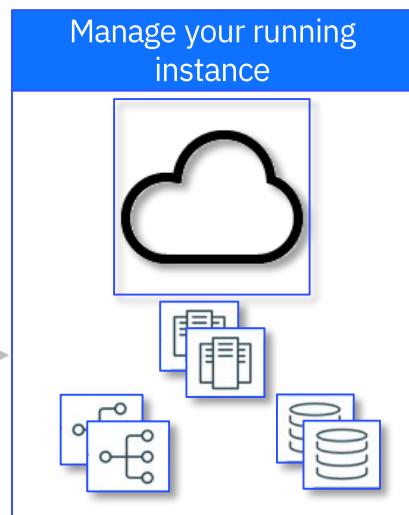
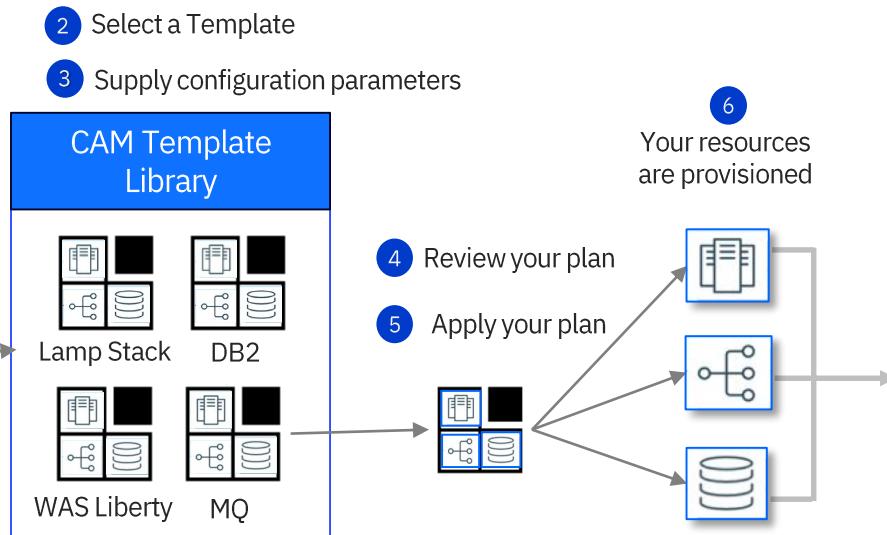
Source: https://en.wikipedia.org/wiki/Infrastructure_as_Code

Store Terraform configuration in Git, manage infrastructure as code

- **Accelerate development velocity** with reusable infrastructure based on open source Terraform
- **Improve governance and transparency** by tracking the ‘who’, ‘what’ and ‘when’ of all environment changes
- **Improve development team collaboration** by enabling team members to easily share application environments
- **Reduce configuration drift** by making it easy to track changes to your running environment

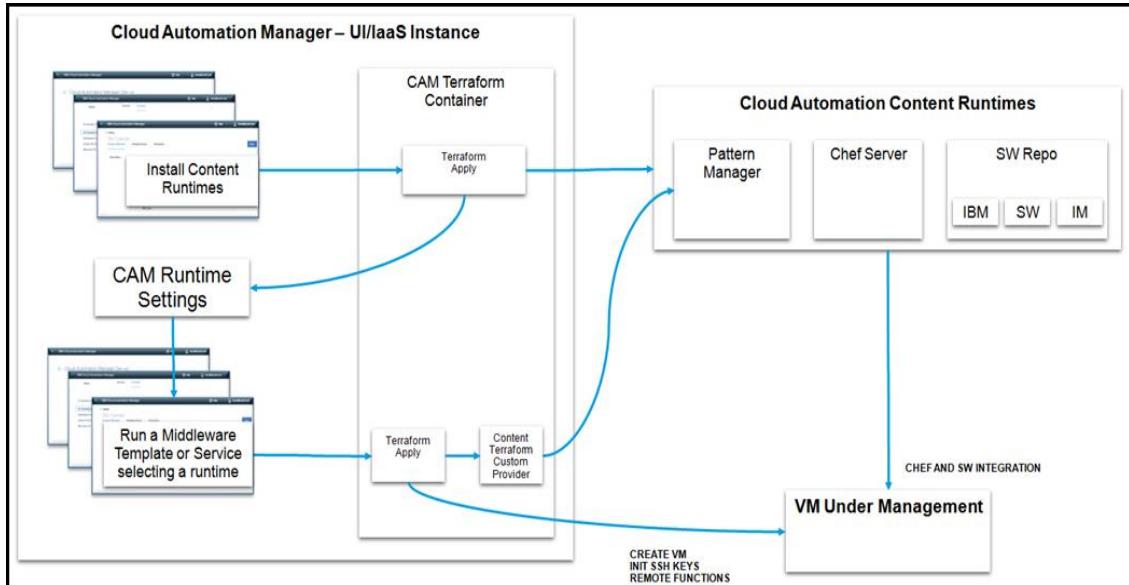
Using Terraform Configurations is simple and easy

| Terraform Configurations | |
|--|--|
| Stored in | Created by |
| <ul style="list-style-type: none"> • GitLabs • GitHub Enterprise • GitHub | <ul style="list-style-type: none"> • You • Your org • IBM • 3rd party |



Cloud Automation Manager Chef Content Runtime Support

Built-in Chef Server runtime



- Optional Chef runtime that can be easily deployed into your provider cloud to support IBM Chef enabled content (VMware only)
- You supply the virtual server, let CAM stand-up a pre-configured Chef runtime
- Fully containerized
- Integrated software repository

Pre-built automation content jumpstart deployments



Terraform configurations:

- WAS ND, WAS Liberty, DB2, MQ, MEAN stack, LAMP stack, Strongloop stack, Kubernetes clusters, MongoDB, Node.js, Amazon EC2 virtual servers, IBM Cloud virtual servers, VMware virtual servers, MySQL

Chef cookbooks:

- WAS ND, WAS Liberty, DB2, MQ, MySQL

Content Marketplace

- **Accelerate projects into production** by leveraging pre-build curated Helm Chart, Terraform Configurations and Chef scripts
- **Catalog of IBM Middleware content**, built to best practices for consistent high quality
- **Interoperable in IBM Cloud** (private, dedicated, public)
- **Reusable and customizable** to adapt to specific business requirements
- **Fully supported with updates** and distribution

IBM provided Starter Library for popular opens source environments

Starter Library

1. MEAN stack, multi-cloud deployment template
2. LAMP stack, single VM deployment template
3. LAMP stack, multi-cloud deployment template
4. Strongloop Stack on a single VM template
5. Kubernetes cluster with NGINX template
6. MongoDB on a single VM template
7. Strongloop 3 tier deployment template
8. Strongloop Kubernetes cluster template
9. Node.js on a single VM template
10. Amazon EC2 virtual server with SSH key
11. 2 virtual servers with SSH keys, 1 on private network template
12. 1 SoftLayer virtual server with SSH key template
13. 1 Virtual Server with SSH key template
14. 2 Virtual Servers with SSH keys template

Supported OS: CentOS, Ubuntu (used in LAMP and MEAN, in addition to CentOS)

Target cloud: Varies by template

The screenshot shows the IBM Cloud Automation Manager interface with the 'Starterpacks' library selected. The library contains 27 templates categorized into several groups:

- Kubernetes Cluster with Nginx Deploy**: Kubernetes - An orchestration system to containerize applications, e.g., Nginx. (VMware vSphere)
- Kubernetes Cluster with Strongloop 3 Tier**: Kubernetes - An orchestration system to containerize applications, e.g., Strong Loop. (VMware vSphere)
- LAMP stack deployment**: LAMP - A fully-integrated environment for full stack PHP web development. (VMware vSphere)
- MariaDB on a Single VM**: MariaDB - An open-source cross-platform relational database. (VMware vSphere)
- MEAN stack deployment**: MEAN - A simple and scalable starting point for full stack JavaScript web development. (VMware vSphere)
- MongoDB on a Single VM**: MongoDB - An open-source cross-platform document-oriented database. (VMware vSphere)
- Node.js on a Single VM**: Node.js - An execution environment for event-driven server-side applications. (VMware vSphere)
- SingleVirtualMachine**: Create Virtual Machine with single vnic. (VMware vSphere)
- Strongloop Stack on a Single VM**: Strongloop - An Easy starting point for full stack JavaScript web development. (VMware vSphere)
- Tomcat on a Single VM**: Tomcat - An Easy starting point for Java web development. (VMware vSphere)
- LAMP stack deployment**: LAMP - A fully-integrated environment for full stack PHP web development. (Amazon EC2)
- Amazon EC2 Virtual Server with SSH Key**: Amazon EC2 Ubuntu 14.04 virtual server under an existing Virtual Private Cloud (Amazon EC2)
- Kubernetes Cluster with Nginx Deploy**: Kubernetes - An orchestration system to containerize applications, e.g., Nginx. (Amazon EC2)
- LAMP stack deployment (hybrid-cloud)**: LAMP - A fully-integrated environment for full stack PHP web development. (IBM Cloud)
- MEAN stack deployment (hybrid-cloud)**: MEAN - A simple and scalable starting point for full stack JavaScript web development. (IBM Cloud)
- MongoDB on a Single VM**: MongoDB - An open-source cross-platform document-oriented database. (IBM Cloud)
- Node.js on a Single VM**: Node.js - An execution environment for event-driven server-side applications. (IBM Cloud)
- 1 Virtual Server with SSH Key**: A sample cloud resource template that creates a Debian virtual server and ssh key. (IBM Cloud)
- 2 Virtual Servers with SSH keys with one on a private network**: A sample cloud resource template with 2 virtual servers, one on a private network. (IBM Cloud)
- Strongloop 3 Tier Deployment**: Strongloop - An Easy starting point for full stack JavaScript web development. (Microsoft Azure)
- LAMP stack deployment on Microsoft Azure**: LAMP - A fully-integrated environment for full stack PHP web development. (Microsoft Azure)
- MEAN stack deployment on Microsoft Azure**: MEAN - A simple and scalable starting point for full stack JavaScript web development. (Microsoft Azure)

IBM Enterprise Content for Middleware, Analytics, and Data

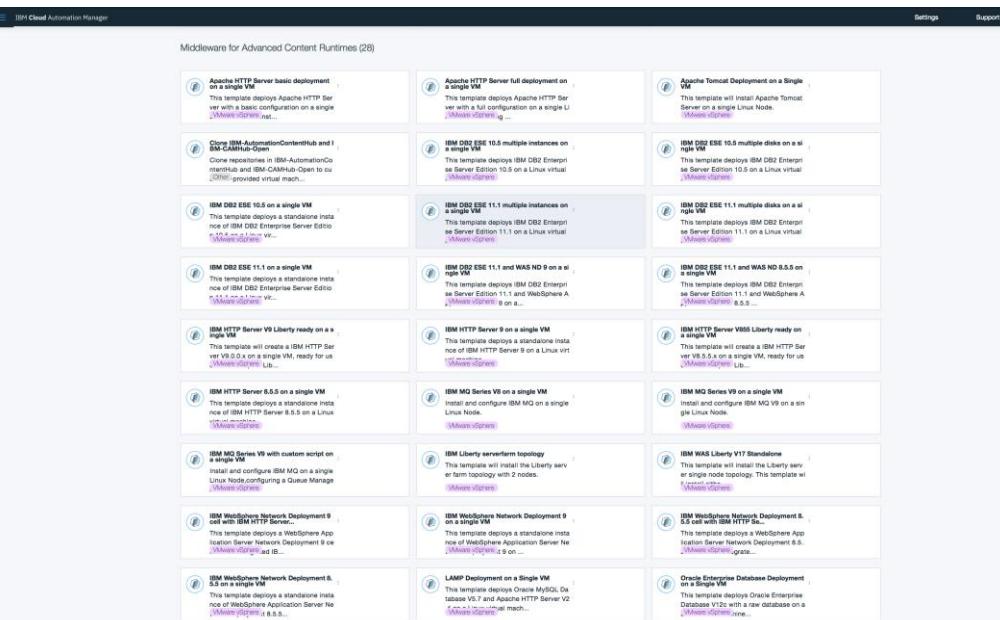
IBM Enterprise Middleware Library

1. Apache HTTP Server Standalone Terraform configuration
2. IBM DB2 Standalone Server Terraform configuration
3. IBM HTTP Server Standalone Terraform configuration
4. WebSphere Liberty Single Server Terraform configuration
5. MQ Basic Single Queue Terraform configuration
6. WebSphere Application Server ND Standalone Server Terraform configuration
7. WebSphere ND Basic Cluster Terraform configuration
8. Oracle MySQL Standalone Terraform configuration
9. Apache Chef Automation Cookbook
10. Apache Tomcat Chef Automation Cookbook
11. Oracle MySQL Community Chef Automation Cookbook
12. MQ Chef Automation Cookbook
13. DB2 Chef Automation Cookbook
14. WebSphere Application Server ND Chef Automation Cookbook
15. WebSphere Liberty Chef Automation Cookbook
16. IBM Installation Manager Chef Automation Cookbook
17. IBM HTTP Server Chef Automation Cookbook

Supported OS: RHEL, Ubuntu (some payloads)

Target: VMWare

Roadmap: IBM Cloud, AWS EC2



Leverage the Terraform public community for more pre-built content

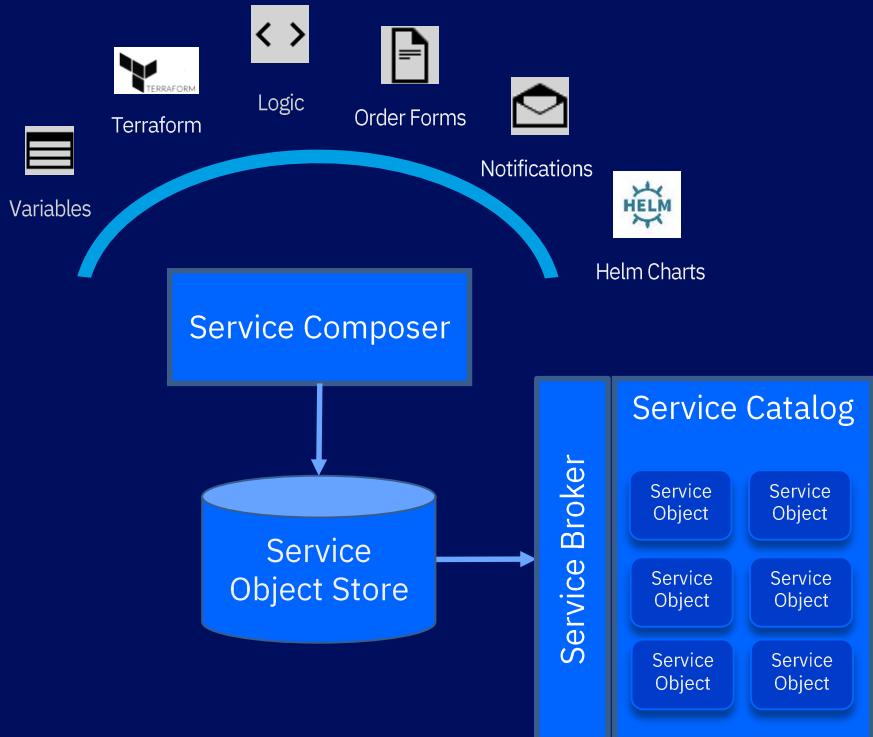
The screenshot shows the HashiCorp Terraform Module Registry interface. At the top, there's a search bar, a 'Browse' button, a 'Publish' button, and a 'Sign-in' button. Below the header, there are filters for 'Provider' and a 'Verified' badge. The main content area displays a grid of nine Terraform modules:

| Provider | Name | Description | Version | Author |
|----------|-------------|---------------------------------|-------------------------------|--------------------|
| aws | autoscaling | Terraform module which... | Version 1.0.3 | By terraform-a... |
| azurerm | consul | A Terraform Module for how... | Version 0.0.3 | By hashicorp |
| azurerm | vault | A Terraform Module for how... | Version 0.0.1 | By hashicorp |
| google | sql-db | Modular Cloud SQL... | Version 1.0.0 | By GoogleClou... |
| google | vault | A Terraform Module for how... | Version 0.0.1 | By hashicorp |
| aws | vault | A Terraform Module for how... | Version 0.0.6 | By hashicorp |
| azurerm | kubernetes | Install a Kubernetes cluster... | Version 1.7.5-tectonic.1-rc.5 | By tectonic.1-rc.5 |
| google | consul | A Terraform Module for how... | Version 0.0.1 | By hashicorp |
| aws | kubernetes | Install a Kubernetes cluster... | Version 1.7.5-tectonic.1-rc.5 | By tectonic.1-rc.5 |

At the bottom of the page, there are 'PREVIOUS' and 'NEXT' navigation buttons.

DevOps ready **application environments as a service** to improve developer velocity

- **Graphically compose services** to hide automation complexity, lock down configurations and simplify end-user consumption
- **Publish composed services** through a self service catalog
- **Consume services via the catalog or REST API**



[← Library](#)

Edit “MEAN Service”

[Overview](#) [Parameters & Forms](#)**Composition**[Source Code](#)

All change saved

Publish

Search...

**Base**

Operation Switch

Approval

Email Notification

Script Execution

Templates**Template Category 1**

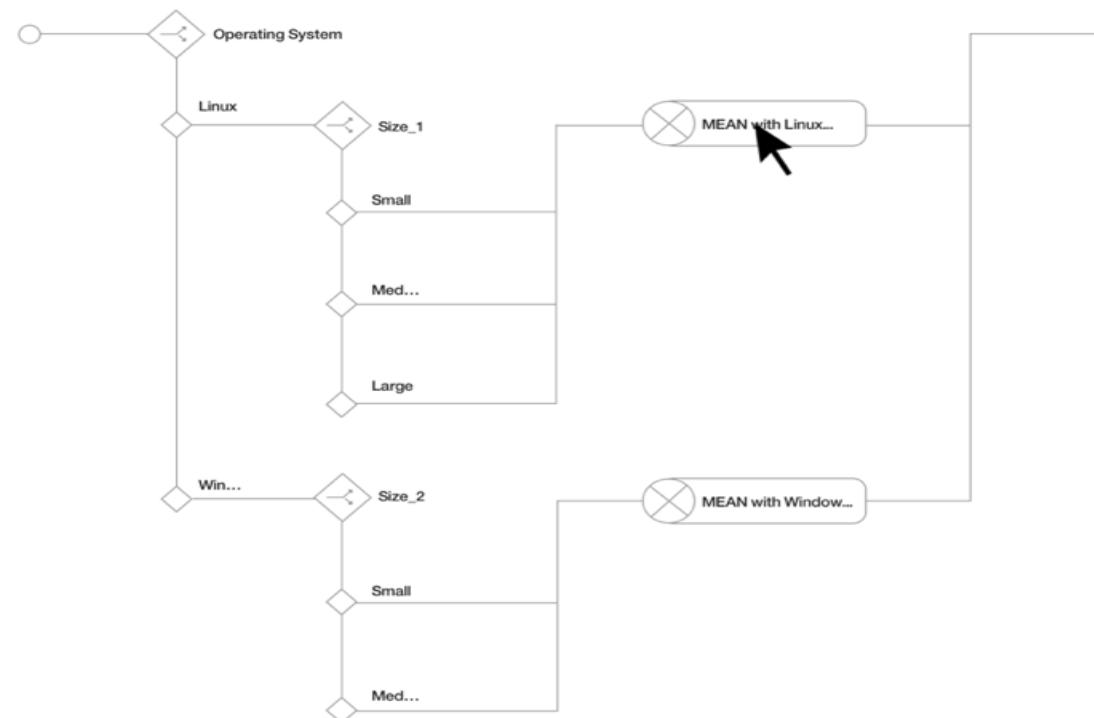
MEAN with Linux OS

Here is service description sample.
Delivers 100x the performance of Apach...[Details](#)

MEAN with Win OS

Here is service description sample.
Delivers 100x the performance of Apach...[Details](#)

LAMP 3 Tiers

Here is service description sample.
Delivers 100x the performance of Apach...[Details](#)[Template Category 2](#)[Template Category 3](#)

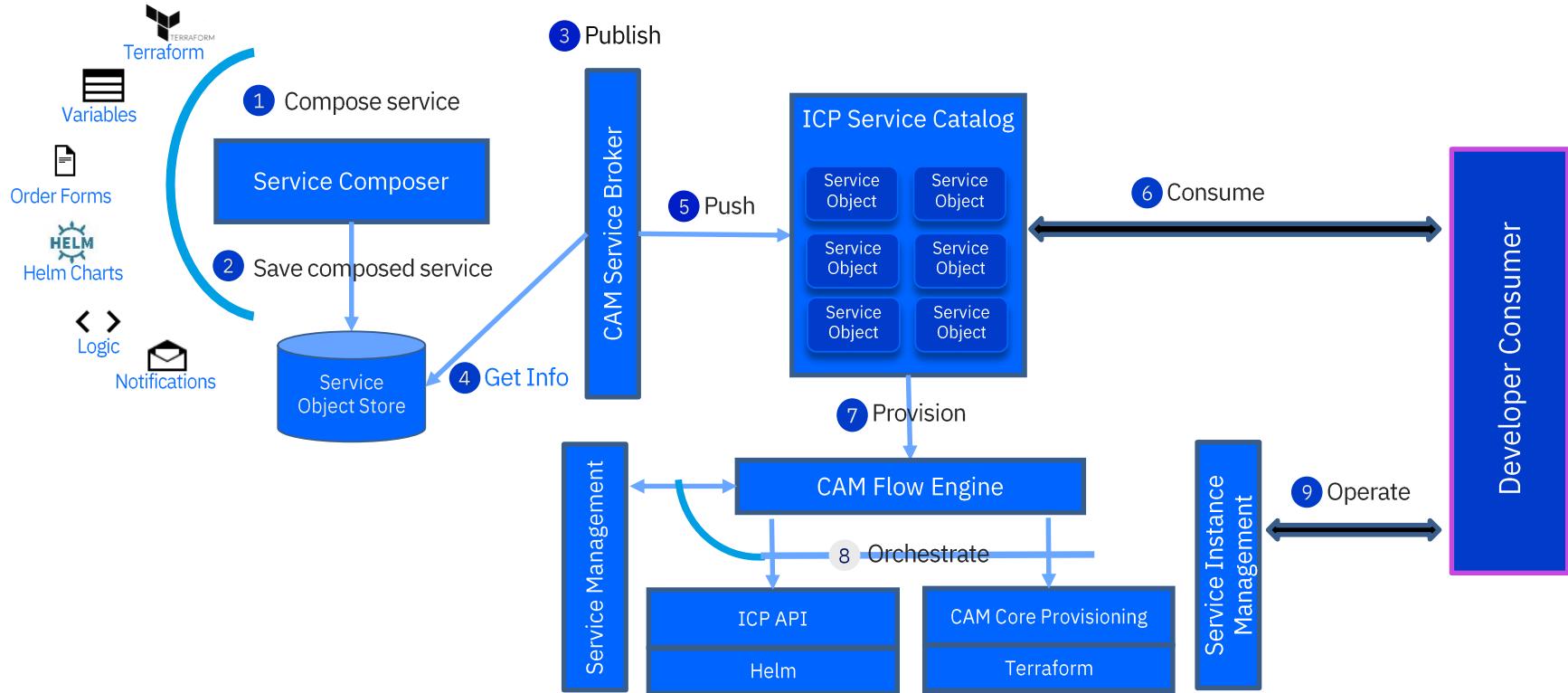
Self Service Catalog – The services you need when you need them

Self-service Catalog for Templates and Services

The screenshot shows the IBM Cloud Automation Manager interface. At the top, there's a navigation bar with 'IBM Cloud Automation Manager' and icons for 'Settings', 'Help', and 'William'. Below the navigation bar, there are tabs for 'Library', 'Services', and 'Templates', with 'Templates' being the active tab. On the left, a sidebar lists categories: 'All Templates (77) >', 'My Templates (11)', 'Middleware (52)', 'Starterpacks (14)', 'Amazon EC2 (20)', 'IBM (28)', and 'VMware vSphere (29)'. A search bar labeled 'Search Templates' is positioned above the main content area. The main content area is titled 'My Templates (11)' and contains a grid of 11 template cards. Each card has a thumbnail icon, a title, a brief description, and a 'VMware vSphere' tag. The templates listed are: 'LAMP stack deployment' (description: 'LAMP - A fully-integrated environment for full stack PHP web development.'), 'MongoDB on a Single VM' (description: 'MongoDB - An open-source cross-platform document-oriented database.'), 'SingleVirtualMachine' (description: 'Create Virtual Machine with single vnic'), 'Tomcat on a Single VM' (description: 'Tomcat - An easy starting point for Java web development.'), 'Node.js on a Single VM' (description: 'Node.js - An execution environment for event-driven server-side applications.'), 'Strongloop 3 Tier Deployment' (description: 'Strongloop - An easy starting point for full stack JavaScript web development.'), 'MEAN stack deployment' (description: 'MEAN - A simple and scalable starting point for full stack JavaScript web development.'), 'Strongloop Stack on a Single VM' (description: 'Strongloop - An easy starting point for full stack JavaScript web development.'), and 'Kubernetes Cluster with Strongloop 3 Tier Deployment' (description: 'Kubernetes - An orchestration system for containerized applications, e.g., Strongloop').

- **Rapidly deliver applications** and infrastructure on multi-cloud environments - IBM and 3rd party, private and public, VM's and containers
- **Simplify delivery of application environments** by hiding automation complexity and locking down environment configurations
- **Streamline fulfilment processes** to deliver application and infrastructure workloads aligned to the needs of the business
- **Deliver application environments as a service** to CI/CD tool chains

Self Service Catalog - Compose, Publish, Consume and Operate



Workload and Service Instance View

Manage workload and services instances from an integrated console

The screenshot shows the IBM Cloud Automation Manager interface with two main sections:

- Deployed Instances:** This section displays a table of deployed instances. A sidebar on the left shows filter options for Status (Active, In Progress, Error, Terminated) and a search bar at the top.

| NAME | STATUS | SERVICE OFFERING | USER | ORDERED TIME |
|---------------|--------|-------------------------|------------------|---------------------|
| test-rv-aug18 | Active | Stable Deploy Virtua... | Rasika A. Rasika | 08/18/2017 09:33 AM |

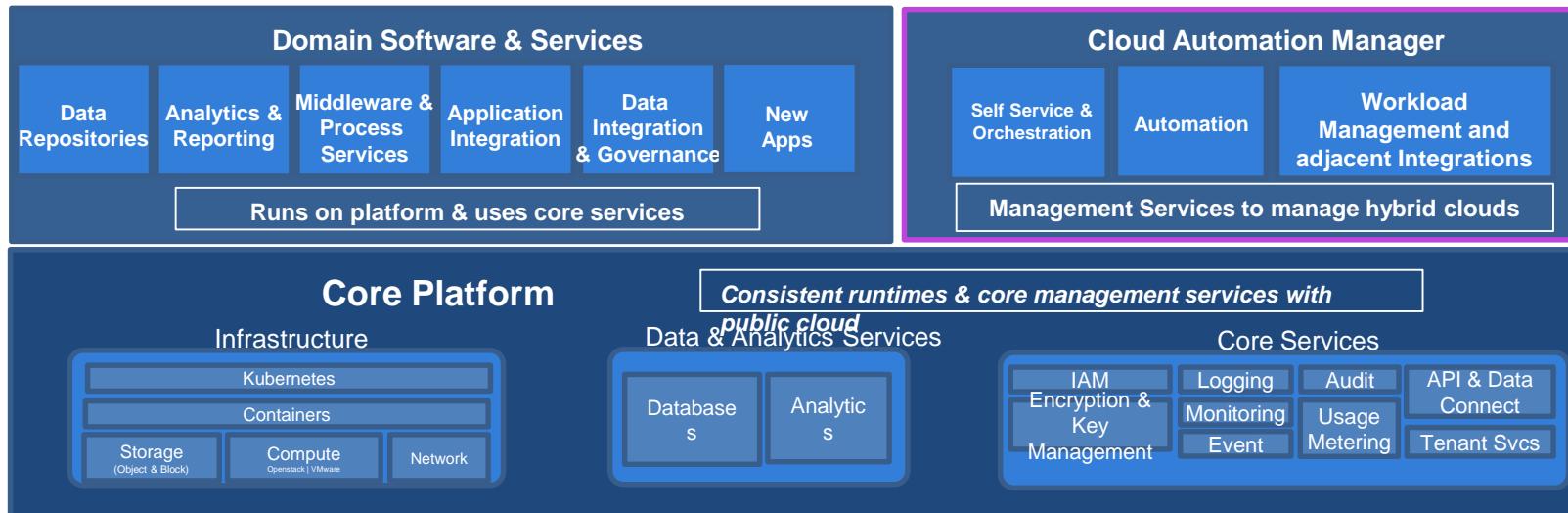
- All Service Instances (1):** This section shows a single instance named "test-rv-aug18" with status "Active".
- Deployed Instances (16):** This section displays a larger table of 16 instances across four pages. The columns include NAME, TEMPLATE, PROVIDER, STATUS, and ACTIONS.

| NAME | TEMPLATE | PROVIDER | STATUS | ACTIONS |
|-------------------------|------------------------------|----------------|---------|---------|
| demo1 | demo1 | | Running | ⋮ |
| demo1 | demo1 | | Running | ⋮ |
| Jeff-gitlab-wrongacc... | Jeff-gitlab-wrong-acesstokey | VMware vSphere | Failed | ⋮ |
| jeff-gitlab-wrongacc... | Jeff-gitlab-wrong-acesstokey | VMware vSphere | Failed | ⋮ |
| Jeff-gitlab-wrongsub... | Jeff-gitlab-wrong-subdir | VMware vSphere | Failed | ⋮ |

- **Manage lifecycle** of VM and cloud resources
- **Workload and Service** instance views
- **Rollout & rollback infrastructure updates** with Terraform Plan & Apply
- **Integrate** with monitoring and logging tools, native cloud consoles and day 2 management tools (UCD, APM, ServiceNow, Splunk, Cloud Broker, etc.)

Cloud Automation Manager works with IBM Cloud Private

- CAM runs on **IBM Cloud Private** as a lightweight container runtime
- CAM is packaged as a Helm Chart that installs into **IBM Cloud Private** through the ICP AppCenter or with a single CLI command
- CAM leverages **IBM Cloud Private** for common management services
- CAM extends **IBM Cloud Private** with multi-cloud management capabilities
- CAM services are published into the **IBM Cloud Private** catalog



DEMO!



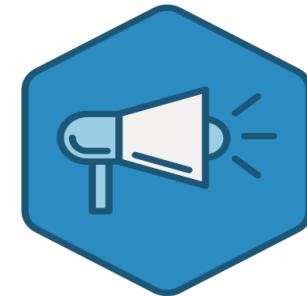
Learn More!

External

1. Learn more about CAM + IBM Cloud Private
2. Request a briefing and demo

Resources

- Marketplace: [**IBM Cloud Automation Manager**](#)
 - Architectural deep dive: [**Behind the scenes with CAM**](#)
 - Complete [**Multi-Cloud maturity assessment**](#) for your organization
- IBM Announces [**Kubernetes-based Cloud private platform**](#)
 - Join the conversation: [**IBM Cloud Private Community**](#)
 - Try [**IBM Cloud Private**](#) today!





Hello Admin!

Accelerate application delivery in the cloud of your choice with cloud agnostic automation.

Develop Automation Contents

Create & Operate IT Services



Connect to a Cloud

To start using Cloud Automation Manager, connect to a cloud in the [Cloud Connections](#) page.



Create Your Template

Quickly find the perfect infrastructure template, or import your own template in the [Library](#) page.



Deploy Your Template

It's as easy as entering a few parameters to deploy your template in the [Library](#) page.



Manage Template Instances

View and manage your template instances in the [Deployed Instances](#) page.

 Search

Cloud Connections (6)

Create Connection 

| NAME | CLOUD PROVIDER | STATUS | ACTIONS |
|-------------------------|-----------------|--------|---------|
| aws-connection | Amazon EC2 | ✓ | ⋮ |
| fit3rack-vmware-conn... | VMware vSphere | ✓ | ⋮ |
| ibmcloud-connection | IBM | ✓ | ⋮ |
| ms-azure-connection | Microsoft Azure | ✓ | ⋮ |
| vmware-nsxv-connecti... | VMware NSXv | ✓ | ⋮ |

[All Templates \(56\) >](#) Search Templates[My Templates \(1\)](#)[Middleware \(28\)](#)[Starterpacks \(27\)](#)[VMware vSphere \(38\)](#)[Amazon EC2 \(4\)](#)[IBM \(11\)](#)[Microsoft Azure \(2\)](#)[Other \(1\)](#)[Create Template +](#)

My Templates (1)

**Bills AWS LAMP Stack in GitLab**

Template contents are hosted in a git repository.

⋮

[Amazon EC2](#)

Middleware for Advanced Content Runtimes (28)

**Apache HTTP Server basic deployment on a single VM**

⋮

This template deploys Apache HTTP Server with a basic configuration on a single Linux node. The inst...

[VMware vSphere](#)**Apache HTTP Server full deployment on a single VM**

⋮

This template deploys Apache HTTP Server with a full configuration on a single Linux node including ...

[VMware vSphere](#)**Apache Tomcat Deployment on a Single VM**

⋮

This template will install Apache Tomcat Ser

**Clone IBM-AutomationContentHub and IBM-CAMHub-Open**

⋮

Clone repositories in IBM-AutomationConte

All Templates (56) >

 mean ×

My Templates (1)

Middleware (28)

Starterpacks (27)

VMware vSphere (38)

Amazon EC2 (4)

IBM (11)

Microsoft Azure (2)

Other (1)

My Templates (0)

Middleware for Advanced Content Runtimes (0)

Starterpacks (4)

Create Template +

**MEAN stack deployment**MEAN - A simple and scalable starting point
for full stack JavaScript web development.

⋮

VMware vSphere

**MEAN stack deployment**MEAN - A simple and scalable starting point
for full stack JavaScript web development.

⋮

Amazon EC2

**MEAN stack deployment (hybrid-cloud)**MEAN - A simple and scalable starting point
for full stack JavaScript web development.

⋮

IBM

**MEAN stack deployment on Microsoft
Azure**MEAN - A simple and scalable starting point
for full stack JavaScript web development.

⋮

Microsoft Azure

[← Template Library](#)

MEAN stack deployment

[Overview](#) [Template Source](#) [Parameters](#)

Deploys a MEAN stack (Mongo, Express, Angular, Node) in AWS providing a complete development environment. This deployment also deploys a simple sample application to validate the stack is operational.

GIT URL :<https://github.com/IBM-CAMHub-Open/starterlibrary/tree/master/AWS/terraform/hcl/meanstack>

Features

| | |
|---------|-------------------|
| AUTHOR | IBM |
| TYPE | Terraform |
| CREATED | 10/08/2017 7:12PM |
| CLOUD | Amazon EC2 |

- **Clouds**
Amazon Web Services (AWS)
- **Template version**
V1.0
- **Operating systems supported**
AWS: Ubuntu 16.04
- **Topology**
AWS: 2 virtual machines:
 - Node.js, Angular.js, Express.js;
 - MongoDB
- **Software deployed**
 1. MongoDB: a NoSQL database
 2. Express.js: a web application framework that runs on Node.js
 3. Angular.js: a JavaScript MVC framework that runs in browser JavaScript engines
 4. Node.js: an execution environment for event-driven server-side and networking applications
- **Default virtual machine settings**

AWS:
Flavor: Size = t2.medium,
Cores = 2, Memory = 4GB, Storage = EBS Only
Network: Private IP - in 10.0.1.0/24, Public IP - Computed
Firewall:
Mongo:
Ingress - Only allows SSH and ICMP for all sources, and TCP 27017 for 10.0.1.0/24 only;

[Deploy Template](#)

[Template Library](#)

Deploy a Template

MEAN stack deployment

Deploys a MEAN stack (Mongo, Express, Angular, Node) in AWS providing a complete development environment. This deployment also deploys a simple sample application to validate the stack is operational.

| | |
|---------|-------------------|
| AUTHOR | IBM |
| TYPE | Terraform |
| CREATED | 10/08/2017 7:12PM |
| CLOUD | Amazon EC2 |

1. Enter a unique Instance Name

*** Instance Name:**

Enter Instance name

2. Select a Cloud Connection

***Cloud Connection:**

aws-connection

3. Fill in Template Parameters

*** AWS Region:** ⓘ

US East (N. Virginia)

*** Network Name Prefix:** ⓘ

opencontent-meanstack

*** Public SSH Key Name:** ⓘ

cam-public-key-meanstack

*** Public SSH Key:** ⓘ*** Hostname of MongoDB Server:** ⓘ

meanstack-db

*** Hostname of Node.js Server:** ⓘ

meanstack-nodejs



Deployed Instances

Services

Templates

Search Instances



Deployed Instances (5)

| 5 instances per page 1-5 of 5 instances | | 1 of 1 pages | | | |
|---|---|----------------|--------------------------|---------------------|---------|
| Name | Template | Provider | Status | Created | Actions |
| AndreStrongloop | Strongloop Stack on a Single VM | IBM | ● Running | 10/09/2017, 7:52PM | |
| AndreStrongloopServi... | SingleVirtualMachine | VMware vSphere | ● Failed | 10/10/2017, 10:26AM | |
| AndreWASVMware | IBM HTTP Server 9 on a single VM - VMware vSphere | VMware vSphere | ● Failed | 10/10/2017, 9:39AM | |
| awstest | MEAN stack deployment | Amazon EC2 | ● Running | 10/09/2017, 2:17PM | |
| BillsLAMPStack | Bills AWS LAMP Stack in GitLab | Amazon EC2 | ● Destroyed | 10/09/2017, 4:20PM | |

[← Deployed Instances](#)awstest ● Running[Overview](#) [Modify](#) [Log File](#)

⋮

Template File i

 MEAN stack deployment

Instance Details i

| | |
|----------|--------------------|
| TEMPLATE | Terraform |
| CLOUD | Amazon EC2 |
| CREATED | 10/09/2017, 2:17PM |

Activity i

All (1) Plan (0) Apply (1) Resources (0)

✓ Deployment is Successful. 10/09/2017 2:17:47 PM | [View Log](#)

| Resource Details | | | | | | i |
|--|---|---------------|-------------------|-------|---------|-------------------------------------|
| NAME | CONSOLE | IP ADDRESS | CREATED | STATE | ACTIONS | |
| aws_instance.mongodb_server | https://console.aws.amazon.com/ec2/v2/home#MongoDB:instance:aws_instance.mongodb_server | 34.233.71.127 | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_instance.nodejs_server | https://console.aws.amazon.com/ec2/v2/home#Node.js:instance:aws_instance.nodejs_server | 34.206.52.81 | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_internet_gateway.default | https://console.aws.amazon.com/ec2/v2/home#InternetGateway:aws_internet_gateway.default | -- | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_key_pair.cam_public_key | https://console.aws.amazon.com/ec2/v2/home#KeyPair:aws_key_pair.cam_public_key | -- | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_key_pair.temp_public_key | https://console.aws.amazon.com/ec2/v2/home#KeyPair:aws_key_pair.temp_public_key | -- | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_route_table.default | https://console.aws.amazon.com/ec2/v2/home#RouteTable:aws_route_table.default | -- | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_route_table_association.default... | https://console.aws.amazon.com/ec2/v2/home#RouteTableAssociation:aws_route_table_association.default... | -- | 10/09/2017 2:17PM | -- | ⋮ | |
| aws_security_group.meanstack_mongo | https://console.aws.amazon.com/ec2/v2/home#SecurityGroup:aws_security_group.meanstack_mongo | -- | 10/09/2017 2:17PM | -- | ⋮ | |



Open-Source Full-Stack Solution For MEAN Applications

Learn more

Congrats! You've configured and run the sample application.

MEAN.JS is a web application boilerplate, which means you should start changing everything :-)

This sample application tracks users and articles.

- Click [Sign Up](#) to get started.
- Configure your app to work with your social accounts, by editing the `/config/env/*.js` files.
- Edit your users module.
- Add new CRUD modules.
- Have fun...

MongoDB

[MongoDB](#) is a NoSQL database.

MongoDB's [great manual](#) is the place to get started with NoSQL and MongoDB.

Express

[Express](#) is a NodeJS server-side application web framework. Check out [The ExpressJS API reference](#) for more information or [StackOverflow](#) for more info.

AngularJS

[AngularJS](#) is client-side web application framework. The [Thinkster Popular Guide](#) and [Egghead Videos](#) are great resources.

Node.js

[Node.js](#) is a JavaScript run-time, popular for being a web server platform. Node's website and this [stackOverflow thread](#) offer excellent starting points to get to grasps with node.

[Library](#)[Services](#)[Templates](#)[All Services \(3\) >](#) Search Services

AWS (2)

VMware Services (1)

[Create a Service +](#)[Add Category](#)**Status** Draft Published Retired

AWS (1)



SimpleAWSMeanStack-Danny

Two VM Mean stack on AWS

Draft Ordered

VMware Services (1)



AndreVMware

Draft Ordered



SimpleAWSMeanStack-Danny

[Save](#)[Publish](#)[Overview](#)[Parameters](#)[Composition](#)[Source Code](#)

Quick Overview

*** Service Display Name**[Change icon](#)

SimpleAWSMeanStack-Danny

Short Description

Two VM Mean stack on AWS

24/150

Detail Overview

Long Description

This service deploys a fully functional MEAN development stack on AWS. There are no parameters and everything is fully automated.

Plan: Standard

A default plan for this service. Adding more plans will be available soon.

Features

- Clouds

[Add a Feature](#)



SimpleAWSMeanStack-Danny

Save

Publish



Overview Parameters

Composition

Source Code



Flow Components



Decision

Notification



Email Notification

Integration



Rest Hook

Templates



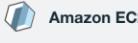
1 Virtual Server with SSH Key



2 Virtual Servers with SSH keys



2 Virtual Servers with SSH keys w...



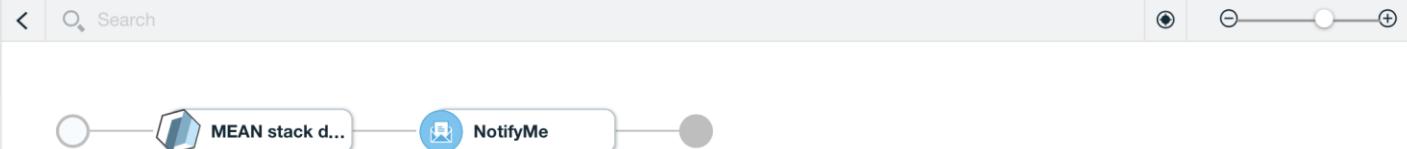
Amazon EC2 Virtual Server with S...



Apache HTTP Server basic deplo...



Apache HTTP Server full deploym...



Library**Services****Templates**

All Services (3) >

Search Services

AWS (2)

VMware Services (1)

Create a Service +

AWS (1)

[Add Category](#)**Status** Draft Published Retired

VMware Services (1)



AndreVMware

Draft

Ordered



SimpleAWSMeanStack-Danny

:

Two VM Mean stack

Edit

Move to ...

Order

Publish

Delete

Deployed Instances[Services](#)[Templates](#)[All Service Instances \(2\) >](#)

Search Deployed Instances

[Clear Terminated/Error Instances](#)**Status**

- Active
- In Progress
- Error
- Terminated

| NAME | STATUS | SERVICE OFFERING | USER | ORDERED TIME | |
|-------------------------|--------|-------------------------|-------|---------------------|--|
| DannysAWSOrder | Error | SimpleAWSMeanStack-D... | admin | 10/10/2017 10:41 AM | |
| AndreStrongloopServi... | Error | AndreVMware | admin | 10/10/2017 10:26 AM | |