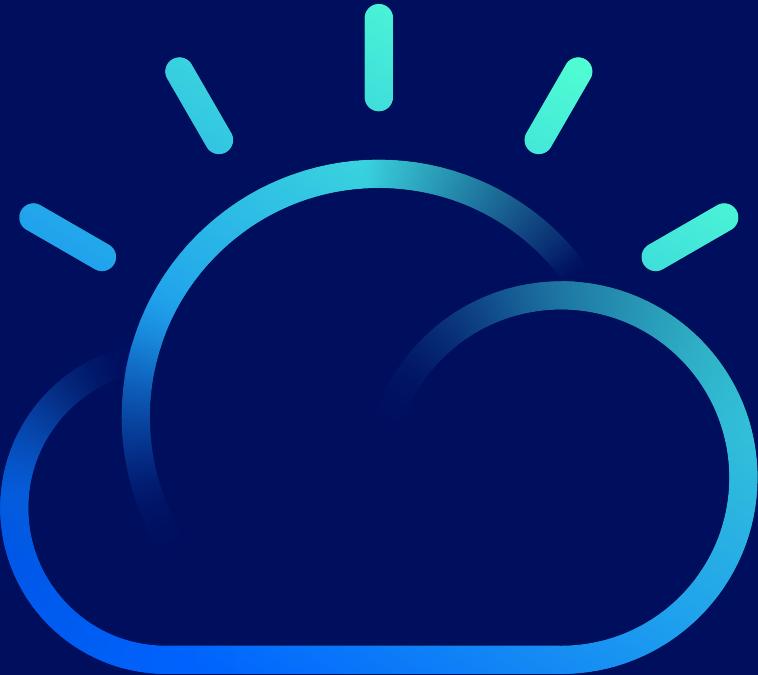


Introduction to CAM



Complexity is the new reality

“ *The cloud market has evolved to meet the needs of clients who want to maintain on-premises systems while tapping a multitude of cloud platforms and vendors...*

*...IT and business executives are looking for multicloud capabilities that **reduce the risks**, and **deliver more automation throughout their cloud journeys***

— Stephen Elliott, IDC





>60%

of customers state they
don't have the tools and
procedures to manage and
operate in a complex
multicloud environment

As organizations embrace hybrid multicloud - new challenges quickly emerging

Visibility Automation



Visibility

- Where are the failed components?
- Where are my services running?
- How can I monitor applications across clusters and clouds?
- How can I manage clusters as if they were one environment ?
- How do I monitor usage across clouds ?

Governance

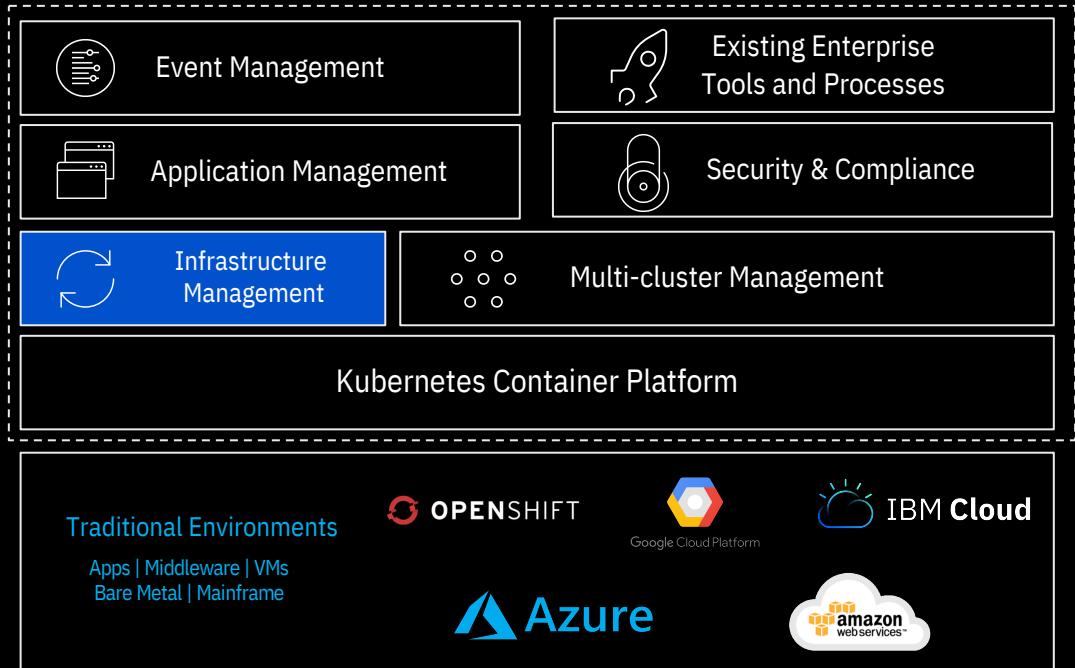
- How do I set consistent security policies across environments ?
- Which clusters are compliant?
- How can I manage configuration across this large environment ?
- How can I place workloads based on capacity, policy?
- How can I track and optimize cloud spend and trends?

Automation

- How do I deploy applications across these environments?
- How do I move workloads across these environments?
- How can I backup my applications?
- How can I do Disaster Recovery ?

IBM Cloud Pak for Multicloud Management

A complete solution for managing traditional and cloud native based applications across public and private clouds from a single, integrated, management control plane.



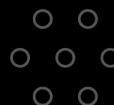
IBM Cloud Automation Manager

Infrastructure Automation powered by Terraform



Application
Management

Event
Management



Multicloud
Management

Infrastructure
Management



Existing Tools
& Processes

Security &
Compliance
Management

Manage Infrastructure-as-Code

Automate application provisioning with a consistent process
that works the same in public, private, and hybrid clouds

-

Operationalize with ‘Git’

Efficiently manage infrastructure at scale

-

Graphical infrastructure automation

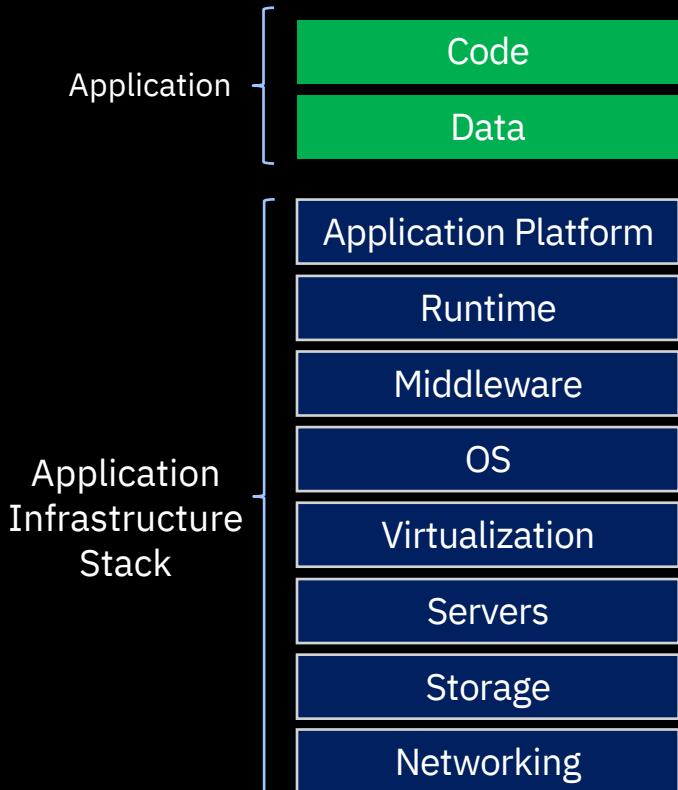
Graphical drag-n-drop tools to design cloud orchestration
workflows and Terraform automation

Application Developers need
Application Environments & Cloud Services

- Self served from a catalog
- Simple to consume
 - request, bind, operate
- Cloud agnostic
- Architecture agnostic
- Compliant

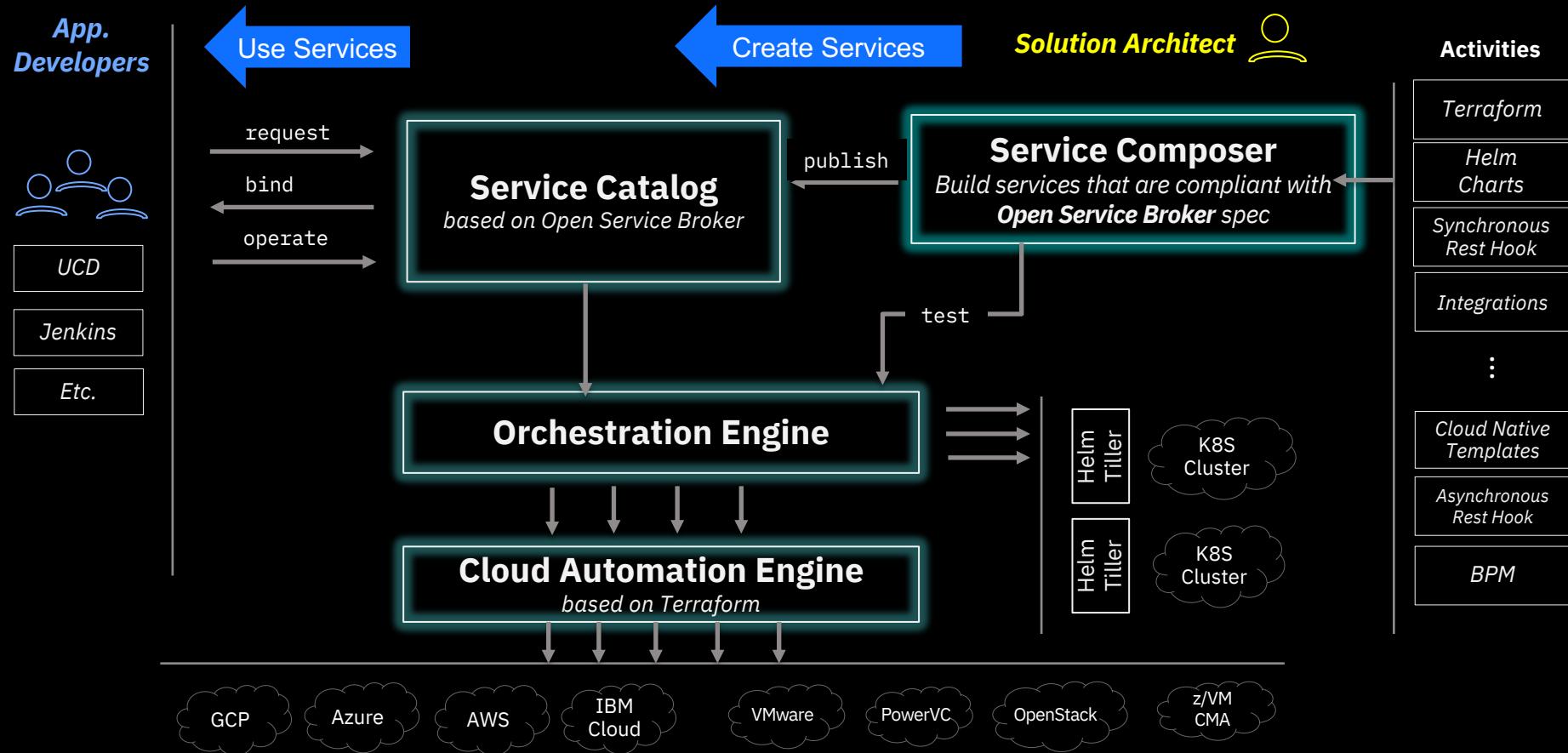
Application Delivery is the goal

Use **Cloud Automation Manager** to build infrastructure stacks for your applications in **all clouds⁽¹⁾**, public, private, and hybrid



- (1) Supports over 100 different clouds and cloud tools with Terraform
- Public clouds: IBM, AWS, GCP, Azure, Ali, Huawei and more
 - Private clouds: IBM, VMware, OpenStack, PowerVC, zVMM, and more
 - Other: <https://www.terraform.io/docs/providers/>

CAM components to deliver application environments



CAM *Service Composer*

Visually build cloud service orchestrations

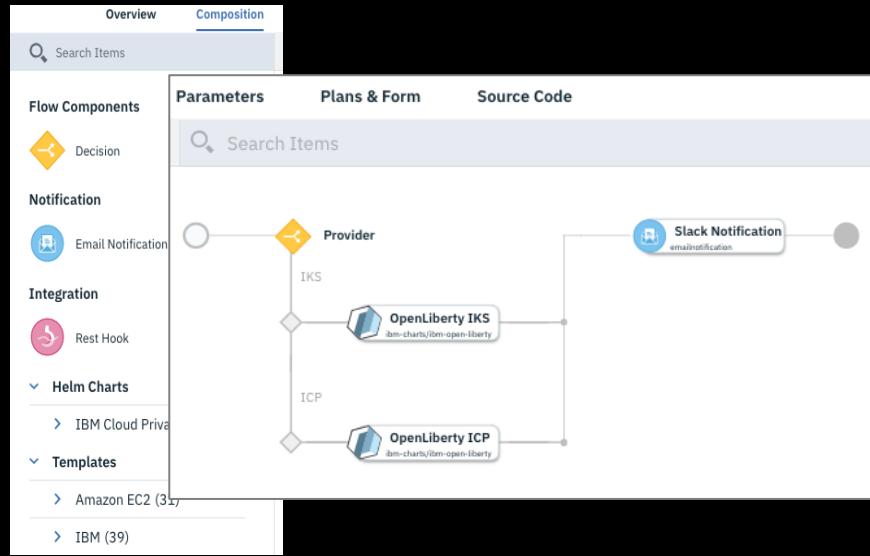
- Package services into Open Service brokers
- Publish service brokers into the Service catalog

Build services from Helm and Terraform

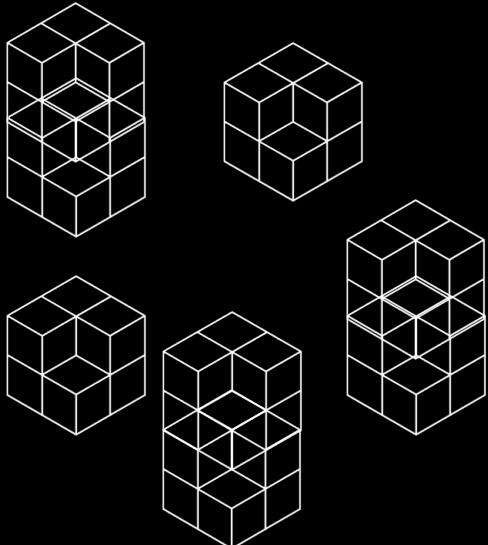
- Mixed clouds - public, private, hybrid
- Mixed arch. - containers, VMs, services

Integrate with adjacent services

- CMDBs, Ticketing, etc. (any REST based call)
- Use out of the box or custom integrations



CAM Automation Engine



- **Based on open technology Terraform**
enterprise hardened by IBM
- **Graphical Terraform *Template Designer***
build automation with 10x productivity
- **Single way to build and manage *all infrastructure* in *all clouds***
optimize skills, standardize processes
- **Manage infrastructure as code**
declarative - focus on the what not the how

CAM *Template Designer*

Develop Terraform visually

- Integrated source code editing
- Terraform helpers (variables, resources, modules, etc.)

Maintain Infrastructure as code

- Save infrastructure in Git VCS
- Declarative - *focus on the what, not the how*
- Manage configuration drift

Get started fast

- Leverage pre-built content from IBM
- Leverage open community Terraform assets

The screenshot shows the Cloud Automation Manager SDK (CMH) interface. At the top, there's a toolbar with buttons for New..., Save, Undo, Redo, Copy, Paste, Find, and Replace. Below the toolbar is a navigation bar with tabs for Diagram and Source. The main area has two sections: 'Diagram' and 'Source'. In the Diagram section, there's a cloud icon labeled 'existing-vpc' connected to two rectangular boxes: 'jke-db-cmh' and 'jke-web-cmh'. Each box contains icons for 'chef_bootstrap_comp' (with 'Chef Bootstrap' and 'Chef Role' sub-options), 'was_create_dmgr' (with 'Chef Role'), and a lock icon. In the Source section, the code editor displays a Terraform template named 'cam-infra-template'. The code defines two AWS resources: 'aws_instance' and 'aws_vpc'. The 'aws_instance' resource is configured with variables like 'ami', 'key_name', 'instance_type', 'availability_zone', 'subnet_id', 'vpc_security_group_ids', and 'tags'. The 'aws_vpc' resource is defined with 'cidr_block' set to '0.0.0.0/0', 'enable_dns_hostnames' set to true, and 'tags' including 'Name' and 'network_name_prefix'.

```
graph TD; existing-vpc --- jke-db-cmh; existing-vpc --- jke-web-cmh;
jke-db-cmh[jke-db-cmh] --- chef_bootstrap_comp1[chef_bootstrap_comp];
jke-db-cmh --- was_create_dmgr1[was_create_dmgr];
jke-db-cmh --- lock1[lock];
jke-web-cmh[jke-web-cmh] --- chef_bootstrap_comp2[chef_bootstrap_comp];
jke-web-cmh --- wasV9_install1[wasV9_install];
jke-web-cmh --- lock2[lock];
graph TD; subgraph cam-infra-template [cam-infra-template]; resource1[resource "aws_instance" "web-server"]; resource2[resource "aws_vpc" "default"]; end;
resource1 --- ami1["ami = \"${var.web-server_ami}\""];
resource1 --- key_name1["key_name = \"${aws_key_pair.auth.id}\""];
resource1 --- instance_type1["instance_type = \"${var.web-server_aws_instance_type}\""];
resource1 --- availability_zone1["availability_zone = \"${var.availability_zone}\""];
resource1 --- subnet_id1["subnet_id = \"${aws_subnet.subnet.id}\""];
resource1 --- vpc_security_group_ids1["vpc_security_group_ids = [\"${aws_security_group.group_name.id}\"]"];
resource1 --- tags1["tags { Name = \"${var.web-server_name}\" }"];
resource2 --- cidr_block2["cidr_block = \"0.0.0.0/0\""];
resource2 --- enable_dns_hostnames2["enable_dns_hostnames = true"];
resource2 --- tags2["tags { Name = \"${var.network_name_prefix}\" }"];
```

CAM component deep dive

Service Composer

Build Open Service Broker specification compliant service brokers

Cloud Automation Engine

Workflow Orchestration

For Service Administrators

- **Build custom services fast** with a graphical design tool - turn months into days
- **Build Hybrid Services** - one process, any architecture, any cloud

For Developers

- **Self service** IaaS++ on demand
- **Low learning curve** - OSB abstracts away cloud platform details
- **Simple to use** - Curated services remove unnecessary detail
- **Use with DevOps toolchains** through the broker's bind interface

For Cloud Automation Developers

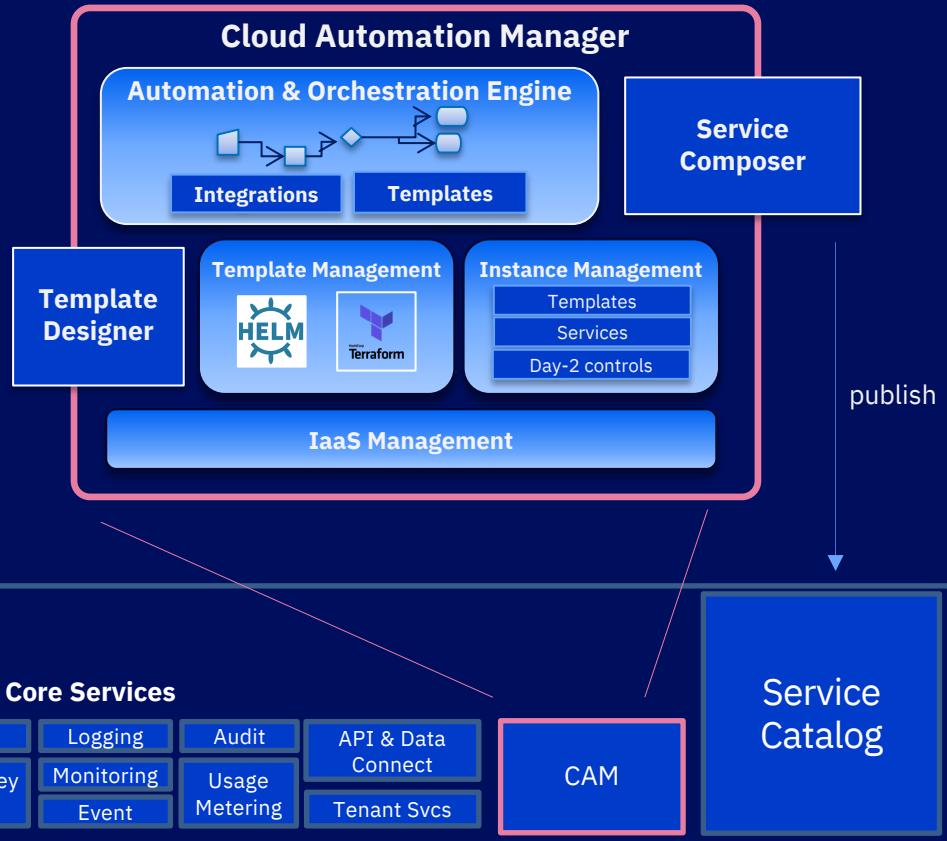
- **Open technology Terraform**, enterprise hardened by IBM
- **One automation language** & one management plane for all clouds
- **Supports multiple private and public clouds** - load new providers in-field
- **Graphical Template Designer** to build Terraform with 10x productivity
- **Get started fast** with 100's of high quality pre-built modules and integrations from the [open community](#) and [IBM](#).
- **Fully compatible** with open source Terraform
- **Manage infrastructure as code** with built-in integration with GitHub and GitLab VCS

For Service Administrators

- **Build custom workflows** for service provisioning
- **Dynamically adjust workflows** based on user & user provided attributes (decision branch, service plans)
- **Integrate** with **social media**
- **Integrate** with **ticketing & CMDBs**
- **Easily build custom integrations** with any REST API based service
- **Efficiently manage parameters** passed between orchestration activities

How is IBM Cloud Automation Manager Delivered?

- Containerized cloud native application
- Installed into a worker node via a Helm chart
- Leverages IBM core services for enterprise capabilities & services
- Built with open source technology



Cloud Automation Manager Key Capabilities

	<p>Assemble & curate cloud services from components in any cloud and expose them in the service catalog</p>
	<p>Automate & orchestrate cloud service provisioning consistently and repeatedly with a common approach in any cloud</p>
	<p>Build fit-for-purpose cloud management solutions from best of breed performance and management tools</p>

Automation

The ability to automate the provisioning of resources across multiple clouds, including management of deployments in a single pane of glass, with support of a Software Configuration Manager for software install and configuration.

Terraform - Declarative Cloud Automation



I want a VMware VM:

- Base image
- Size (vcpu, memory)
- IP address
- Ephemeral Storage



```
resource "vsphere_virtual_machine" "vm_1" {  
  depends_on = ["vsphere_folder.folder_vm_1"]  
  name      = "${var.name}"  
  folder    = "${var.folder}"  
  datacenter = "${var.datacenter}"  
  vcpu      = "${var.vcpu}"  
  memory    = "${var.memory}"  
  cluster   = "${var.cluster}"  
  network_interface {  
    label = "${var.network_label}"  
    ipv4_gateway = "${var.ipv4_gateway}"  
    ipv4_address = "${var.ipv4_address}"  
    ipv4_prefix_length = "${var.ipv4_prefix_length}"  
  }  
  disk {  
    datastore = "${var.storage}"  
    template = "${var.vm_template}"  
  }  
}
```

Common approach in all clouds

- VMs, containers, cloud native services

Open source

- 10,000+ GitHub stars; 1,153 contributors
- Supported by major cloud vendors

Large & growing ecosystem

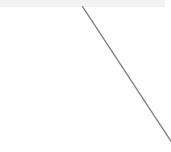
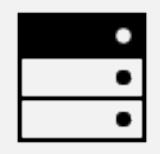
- Terraform Module Registry
- Many OEM providers and provisioners

Enterprise hardened by IBM

- Secrets management
- Role based provisioning
- Team development

Managing cloud infrastructure as code

```
resource "ibm_compute_vm_instance" "vm" [
  hostname      = "${var.hostname}"
  os_reference_code = "${var.os_reference_code}"
  domain        = "${var.domain}"
  datacenter     = "${var.datacenter}"
  network_speed = "${var.network_speed}"
  hourly_billing = true
  private_network_only = "${var.private_network_only}"
  cores          = "${var.cores}"
  memory         = "${var.memory}"
  disks          = ["${var.disk_size}"]
  dedicated_acct_host_only = true
  local_disk     = false
  ssh_key_ids   = ["${ibm_compute_ssh_key.ssh_key.id}"]
  tags           = ["${var.tags}"]
  user_metadata  = "${file("install.yml")}"
]
```



Store Terraform configuration in Git and manage infrastructure as code

Accelerate development velocity with reusable infrastructure based on open source Terraform

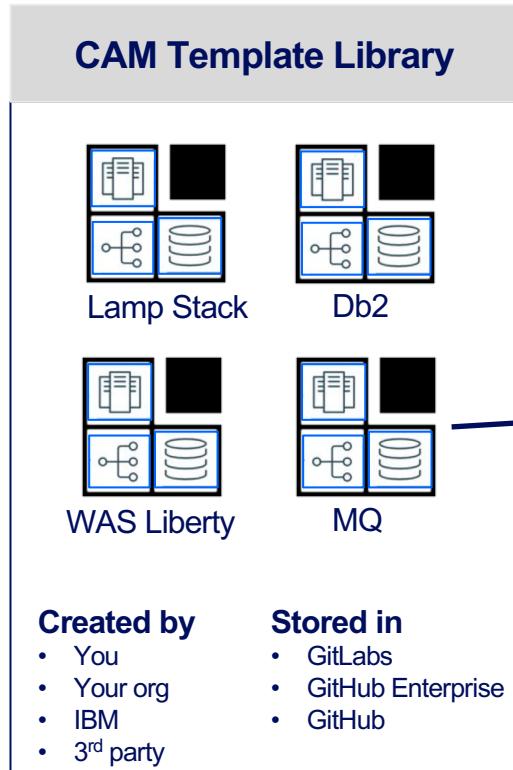
Improve governance and transparency by tracking the ‘who’, ‘what’ and ‘when’ of all environment changes

Improve development team collaboration by enabling team members to easily share application environments

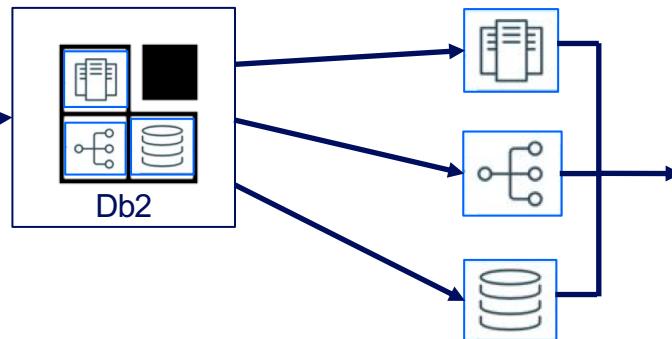
Reduce configuration drift by making it easy to track changes to your running environment

Using Terraform

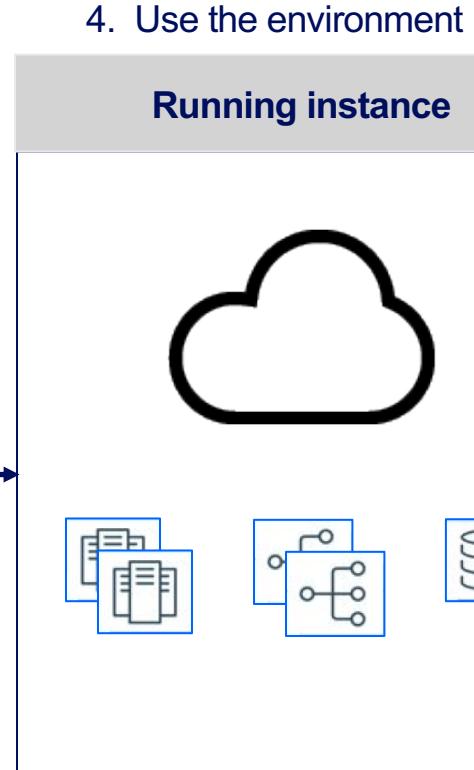
1. Select a template



2. Review and apply plan



3. Resources are provisioned



4. Use the environment

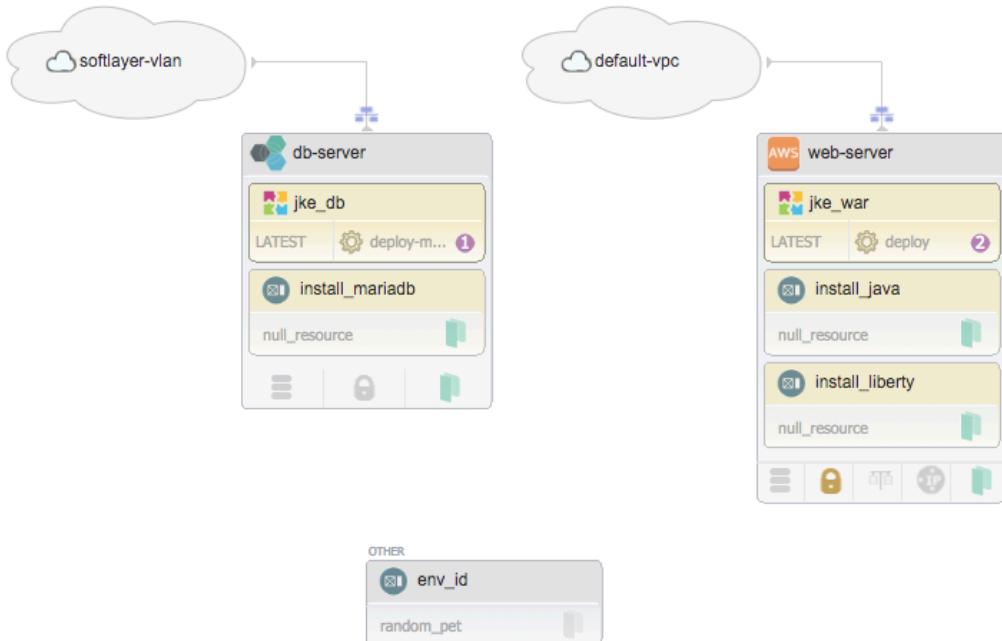
Creating Terraform templates – CAM Template Designer

Graphical and text-based
Terraform template designer

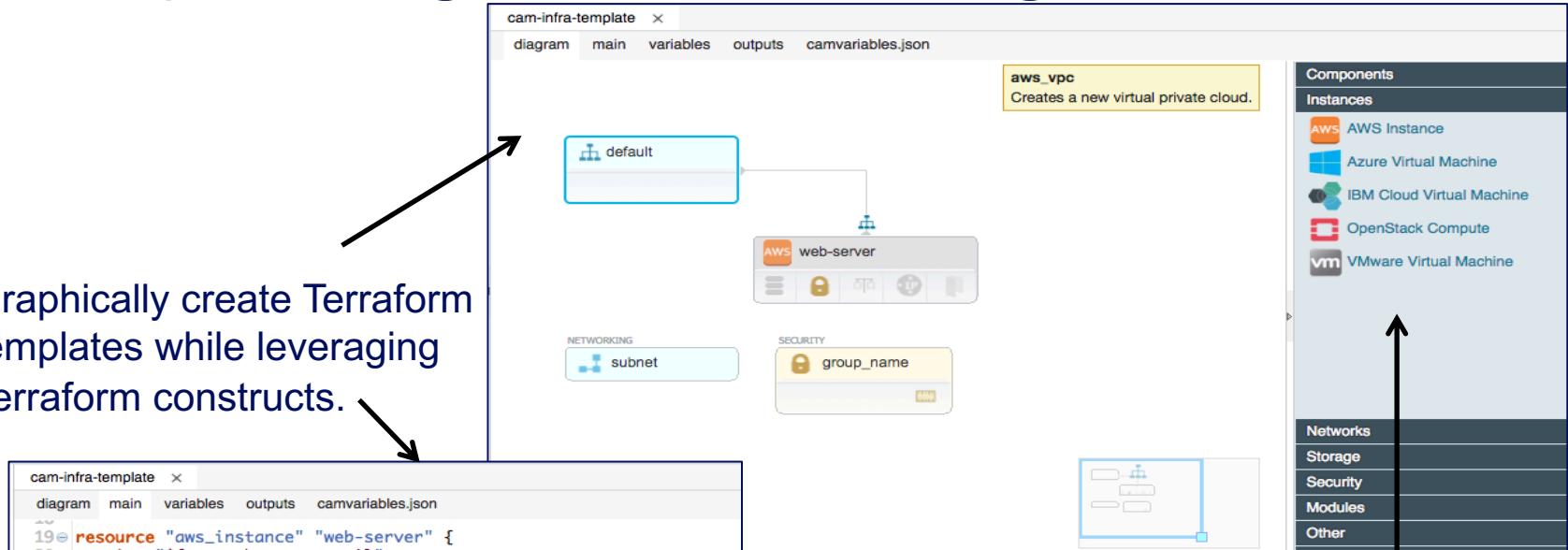
Built-in integration with Git to
easily share and re-use templates

Rich cloud support to create
complex Terraform content via
drag and drop

Easily create and publish CAM
templates



CAM Template Designer – Terraform Design

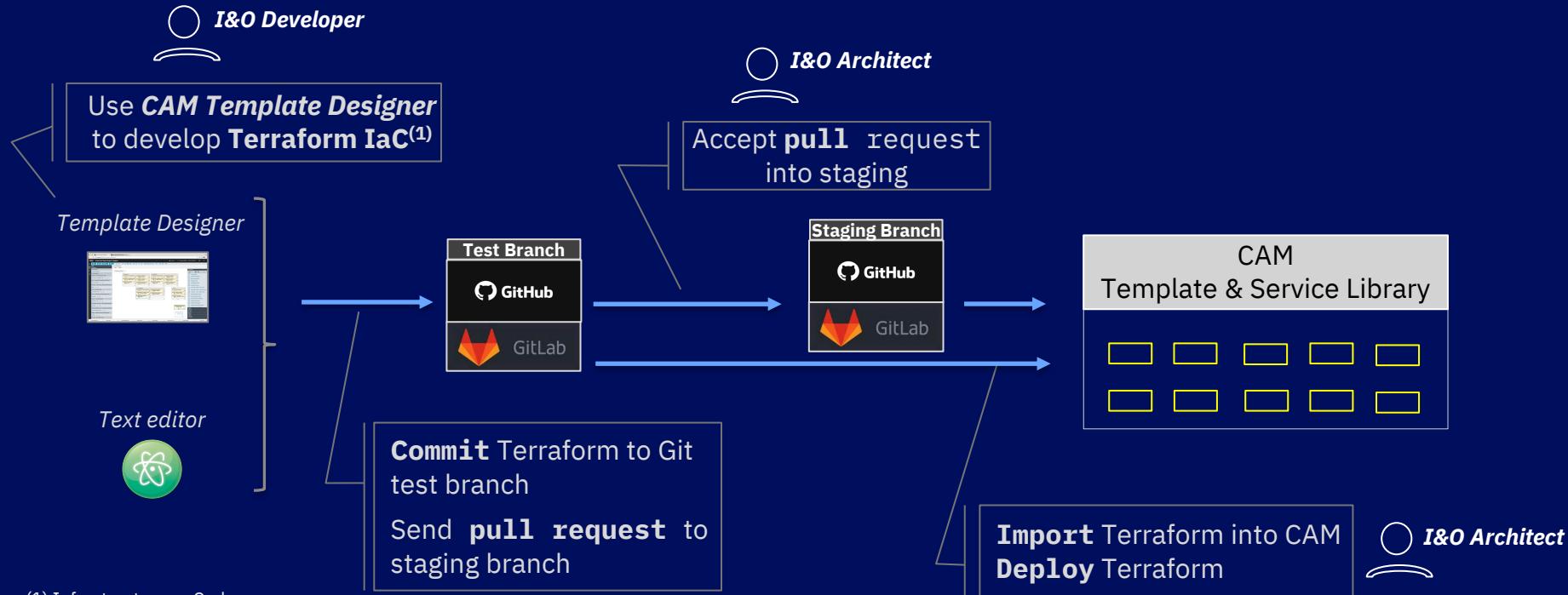


Graphically create Terraform templates while leveraging Terraform constructs.

Rich cloud support

```
cam-infra-template ×  
diagram main variables outputs camvariables.json  
19resource "aws_instance" "web-server" {  
20  ami = "${var.web-server_ami}"  
21  key_name = "${aws_key_pair.auth.id}"  
22  instance_type = "${var.web-server_aws_instance_type}"  
23  availability_zone = "${var.availability_zone}"  
24  subnet_id = "${aws_subnet.subnet.id}"  
25  vpc_security_group_ids = ["${aws_security_group.group_name.id}"]  
26  tags {  
27    Name = "${var.web-server_name}"  
28  }  
29}  
30  
31resource "aws_vpc" "default" {  
32  cidr_block      = "0.0.0.0/0"  
33  enable_dns_hostnames = true  
34  tags {  
35    Name = "${var.network_name_prefix}"  
36  }  
37}
```

Use Case: As an I&O Architect, I can automate infrastructure provisioning with a declarative language and use DevOps practices to manage cloud infrastructure-as-code with the same process in any cloud for consistency, repeatability, and efficiency



[← Template Library](#)

Dynatrace OneAgent Installation Template

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

This template will install Dynatrace OneAgent on a target node. It requires a Dynatrace SaaS account and internet access from the target node.

AUTHOR	IBM
TYPE	Terraform
CREATED	05/04/2018 1:51PM
CLOUD	Other
VERSION	master

Template Version:

master(default)

GIT URL :https://github.com/millerkc/template_integration_dynatrace/tree/master/other/terraform

To clone the template click the Git repository link, clone the repository and then create a template with the new repository information.

Features

- **Clouds**

All Clouds

- **Topology**

Deployed on a single Linux node. Script is executed remotely.

- **Default Virtual Machine Settings**

vCPU 1, Memory (GB) 2, Disk (GB) 10, 150MB free in /tmp

- **Operating Systems Supported**

Ubuntu 16
Ubuntu 14
Redhat 6.x
Redhat 7.x

- **Software Deployed**

Dynatrace OneAgent for Linux

- **Usage and Special Notes**

1. You must already have a Dynatrace SaaS account (trial accounts are OK).
2. The supplied API Token must have InstallerDownload scope (can be found by default at Settings > Integration > Platform as a Service).
3. This integration template is run remotely on the Target Virtual

[Deploy Template](#)

[← Template Library](#)

Deploy a Template

Dynatrace OneAgent Installation Template

This template will install Dynatrace OneAgent on a target node. It requires a Dynatrace SaaS account and internet access from the target node.

AUTHOR	IBM
TYPE	Terraform
CREATED	05/04/2018 1:51PM
CLOUD	Other
VERSION	master

1. Enter a unique Instance Name

*** Instance Name:**

Enter Instance name

2. Select a Cloud Connection

***Cloud Connection:**

Other

3. dynatrace_oneagent_linux

*** IP address:** *** Operating system user:** 

root

Operating system password: **Operating system private key:** *** Dynatrace Environment ID:** [Cancel](#)[Deploy](#)

Automate any cloud service, in any cloud, consistently and repeatably

Common approach in all clouds

- Standardize provisioning for efficiency
- Built on open technology, based on Terraform, Chef, and Helm

Manage infrastructure as code

- Declarative - *focus on the what, not the how*
- Repeatable, reduces configuration errors

Pre-built content to get started fast

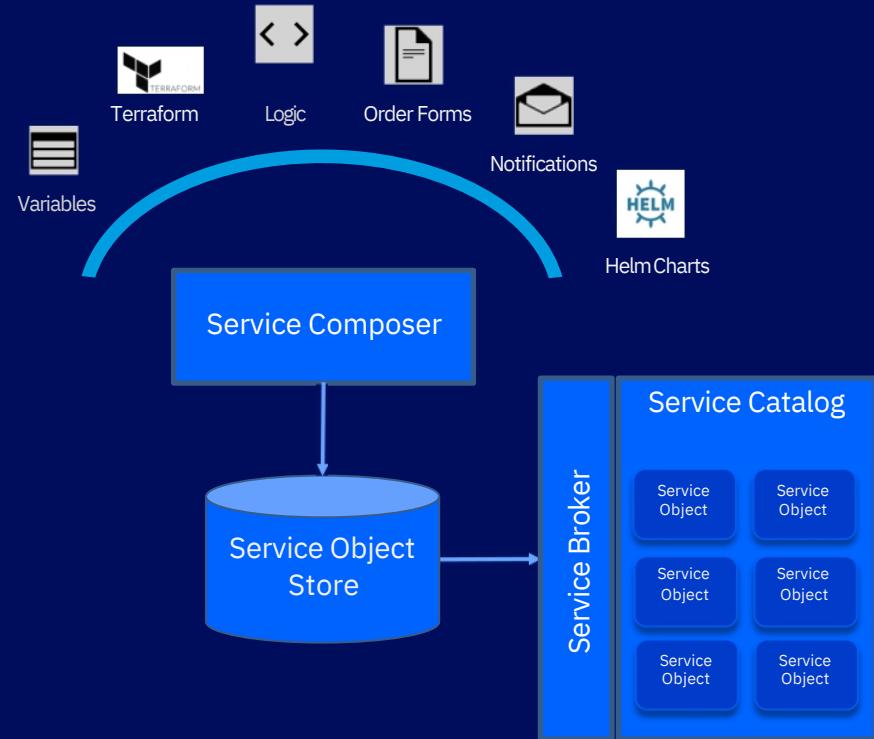
- Extensive catalog of IBM Terraform & Helm charts
- Open source & open community assets
- Customize with Template Designer

Orchestration

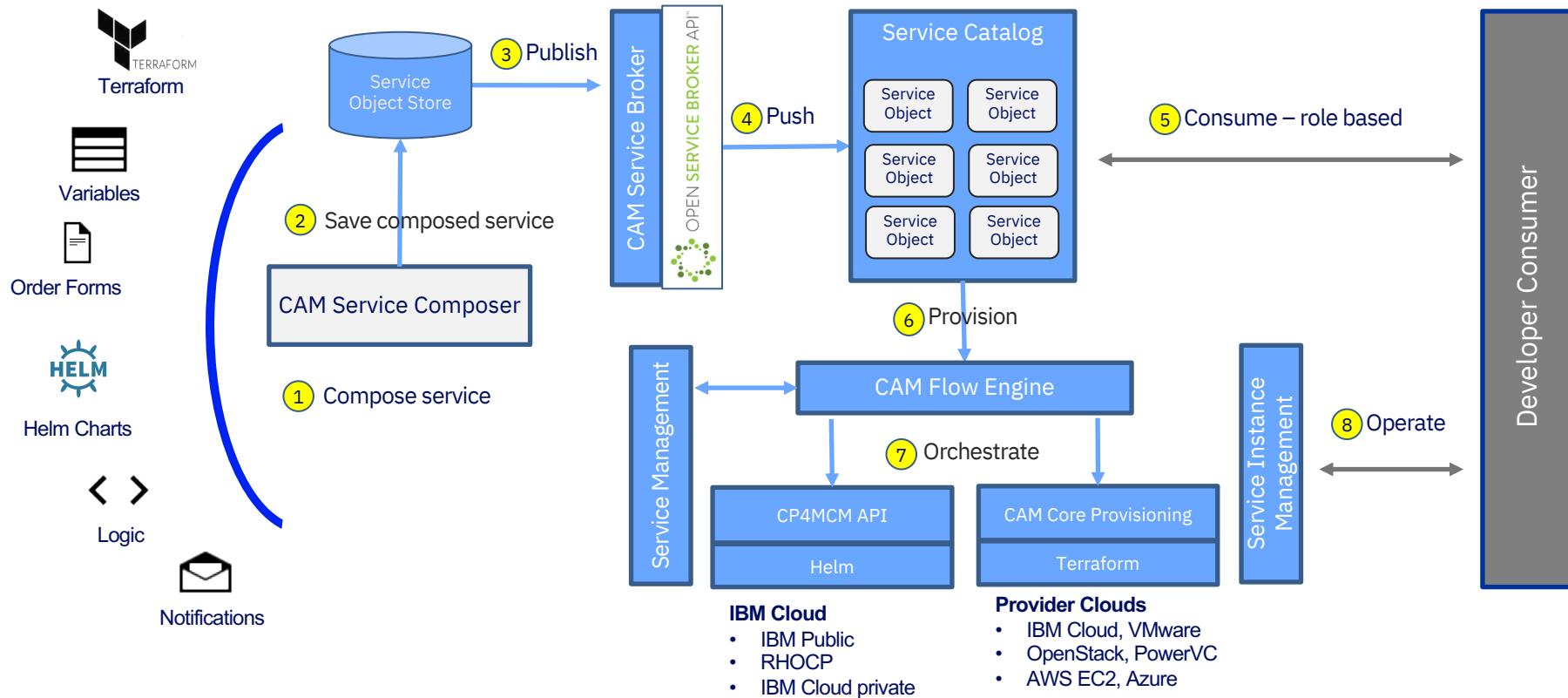
Assembling automations into a flow that represents the end-to-end lifecycle of a service and supports publication into self service catalogs.

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

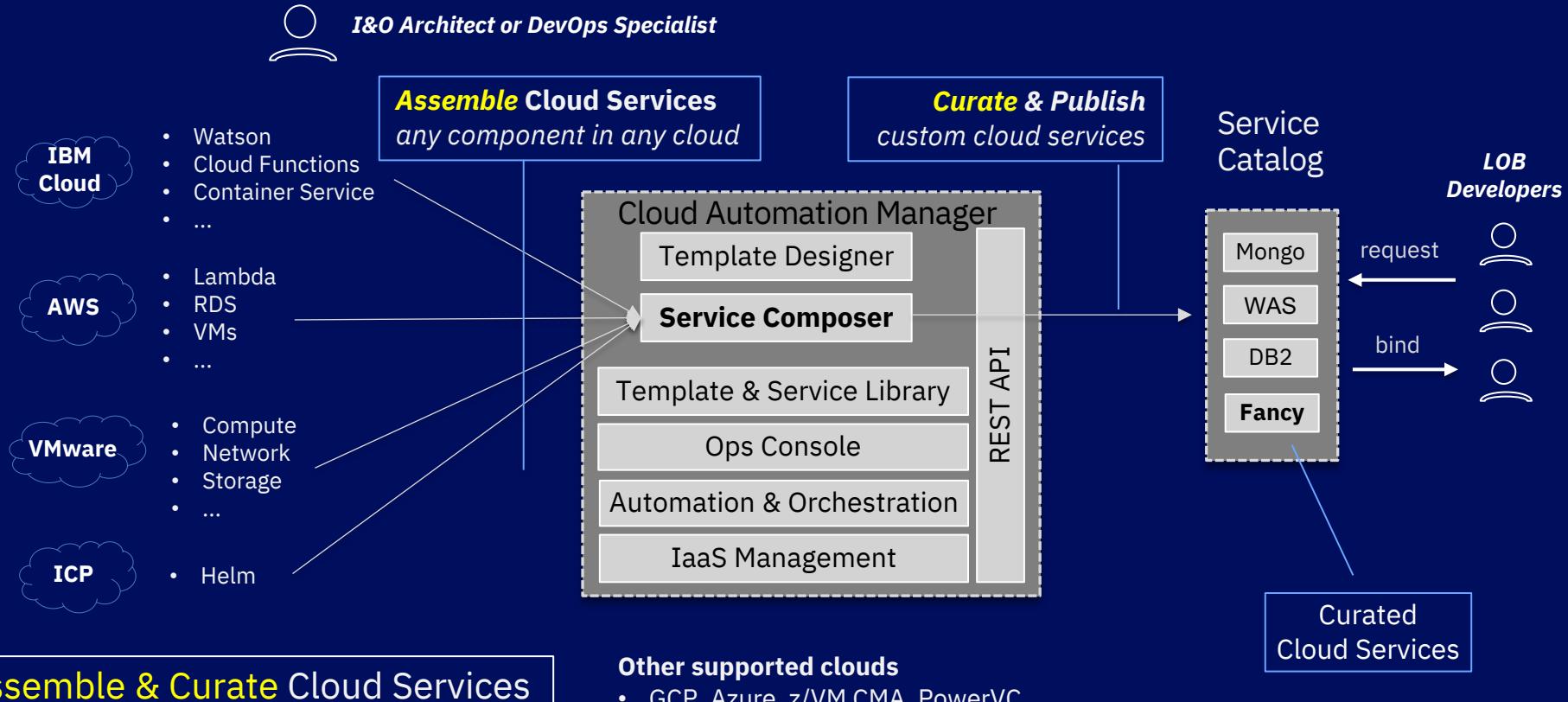
- ***Graphically compose services*** to hide automation complexity, lock down configurations and simplify end-user consumption
- ***Publish composed services*** to self service catalogs
- ***Consume services via REST API***



Service Management Lifecycle: Compose, Publish, Consume and Operate



Use Case: As an I&O Architect, I can collaborate with my DevOps colleague to assemble a sophisticated cloud service from VMs, containers and cloud native services located in any cloud and publish it into the Service catalog for my LOB developers



Assemble, Curate, Publish - with Service Composer

IBM Cloud Automation Manager Docs Support

WASNDService

Overview **Composition** Parameters Plans & Form Source Code

(1) Assemble the service
Drag activities from the palette to
• Define 3 plans
• Get an IP address
• Integrate an ELK stack
• Integrate a performance monitor

(2) Curate the service by locking down configuration variables

(3) Publish the service into the MCM Catalog

Save Publish

Development Plan

Test Plan

Production Plan

IBM WebSphere Network Deployment V9 on a single ... X

Basic Information Parameters

Search Parameters

INPUT PARAMETERS	VALUE
WASN01_dns_servers	1 Items
WASN01_dns_suffixes	1 Items
WASN01_domain	<code>\$(templates.infobloxcd464.output.associated_domain)</code>
WASN01-image	Content/ContentRH_Template_2018_1Q
WASN01-os_admin_user	root
WASN01-os_password	Op3nPatterns
user_public_ssh_key	None
WASN01_root_disk_size	100
WASN01-name	<code>\$(templates.infobloxcd464.output.associated_</code>

A Published Service



IBM Cloud Private

Create resource

Docs

Support



ibm-transadv-dev

IBM Cloud Product Insights Transformation Advisor

ibm-charts



ibm-transadv-dev

IBM Cloud Product Insights Transformation Advisor

ibm-charts-internal



ibm-voice-gateway-dev

IBM Voice Gateway Helm chart (Developer Trial)

ibm-charts



ibm-websphere-liberty

WebSphere Liberty for Linux on amd64, ppc64le and s390x

ibm-charts



ibm-websphere-liberty

WebSphere Liberty for Linux on amd64, ppc64le and s390x

ibm-charts-internal



ibm-webterminal-dev

A browser-based full xterm terminal.

ibm-charts



ibmcloudcloudantservice

ibmcloudcloudantservice

service



icsservice

icsservice

service



wasapplicationservice

wasapplicationservice

service



wasndservice

wasndservice

service

Example: Ordering the WAS ND Service from the Service Catalog

[View all](#)

wasndservice

wasndservice

WASNDService

VERSION 2458111
PUBLISHED Mar 7th 2018
TYPE Service

Details

Helps user deploy WAS ND in multiple formats and connected to different systems

Useful Links

Documentation:

Support link:

Plans

PLAN	FEATURES	PRICING
development	To deploy a Standard plan	Free
staging	Staging	Free
production	Production	Free

The user can select from three service plans

Configure

Normalize delivery of cloud services to your developers *seamlessly* on your terms

Assemble cloud services

- Graphical drag & drop UX
- Containers, VMs, cloud native services
- Hybrid cloud, multi-cloud, any cloud

Curate cloud services

- Governance, control, visibility
- Pre-set configuration to hide complexity
- Orchestrate execution order & flows

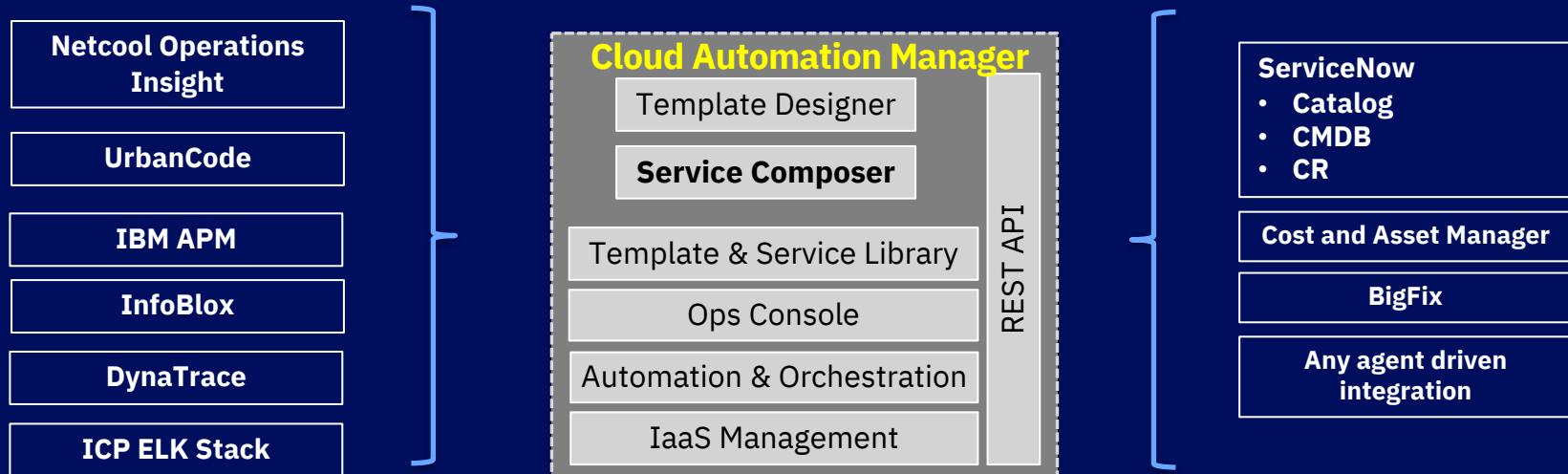
Customize cloud services

- Define multiple service plans
- Define bind interface for CI/CD toolchains
- Integrate with heritage ITSM tools

Publish cloud services into the MCM catalog

- Simple cloud services or compete solutions
- 3rd party catalogs (eg ServiceNow)
- *Platform independent cloud development*

Use Case: As an I&O Architect, I can use Cloud Automation Manager as the central brain of a fit-for-purpose multi-cloud Cloud Management Platform by integrating additional cloud management capabilities



Use Terraform modules to inject agents
Compose custom day-2 actions

Fit-for-purpose all cloud management

Purpose built for integration

- Flexible, extensible, modern architecture
- Fully scriptable with REST API
- Leverages Terraform for automation & workflow orchestration

Open Source Integrations

- Terraform modules
- Distributed from github, supported by IBM
- Easy to extend, fork and customize

Leverage CP4MCM native services

- CP4MCM native services available out of the box
- CAM HA provided by CP4MCM

Bridge traditional and cloud native

- Land traditional, expand to containers

Cloud Pak for MCM with Cloud Automation Manager Ecosystem

Pre-built Terraform for IBM Cloud Automation Manager



Open Source Terraform

MEAN stack – VMware, Azure, AWS, IBM Cloud
LAMP stack – VMware, Azure, AWS, IBM Cloud
Node.js – VMware, IBM Cloud
Strongloop – VMware, IBM Cloud
MariaDB - VMware
MongoDB – VMware, IBM Cloud
MongoDB Strongloop 3 tier – VMware, IBM Cloud
Virtual Servers with SSH key – AWS, IBM Cloud
Apache HTTP Server - VMware
Apache Tomcat - VMware

Enterprise Middleware Catalog

IBM DB2 EE
IBM MQ
IBM WebSphere Application Server ND
IBM WebSphere Liberty
IBM HTTP Server
Oracle DB Enterprise
Oracle MySQL

Available for VMware, IBM Cloud and AWS

Available in the CAMHub catalog @ <https://github.com/IBM-CAMHub-Open/>

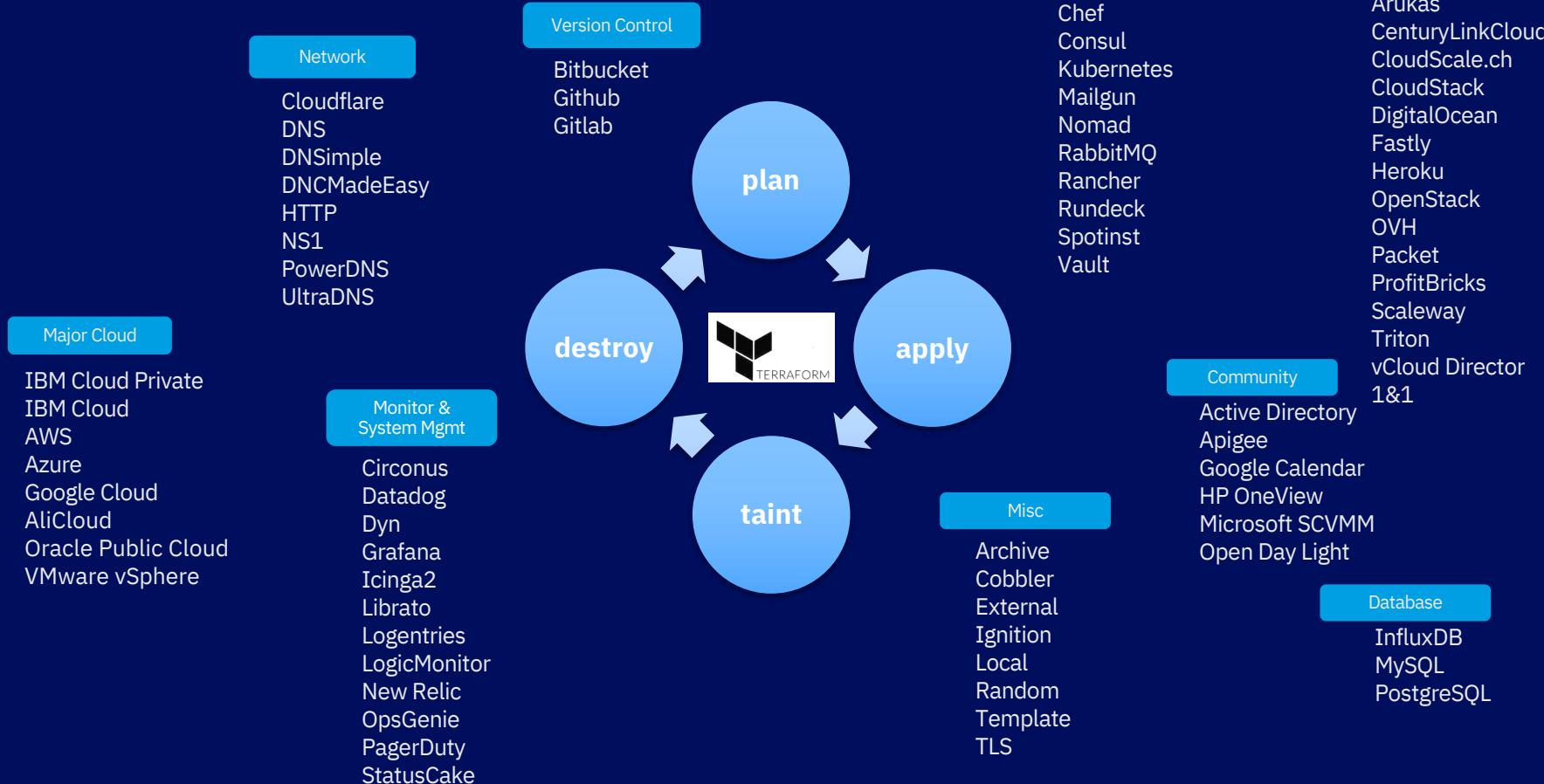


Community Templates
Terraform Module Registry



Bring your own templates
Self written, IBM Cloud Schematics, etc

Terraform open community eco-system



The Cloud Automation Manager Business

Accelerate Delivery of *Application Services*

All clouds, one tool

- Built with open technology
- Hybrid cloud

Any workload architecture

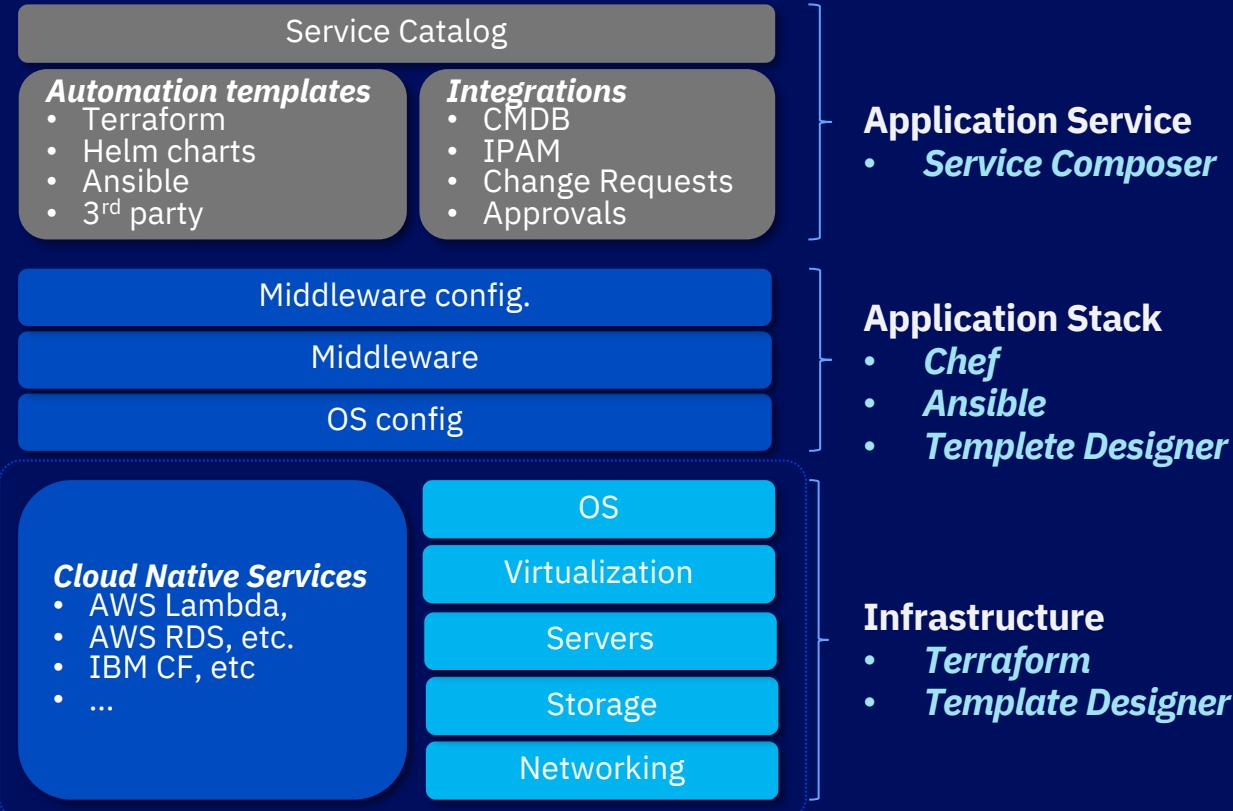
- Traditional
- Containers
- Cloud services
- Hybrid architecture

Graphical experience

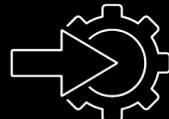
- Blueprint Designer
- Service Composer
- CAM UI

Purpose build for integration

- Leverage existing investments
- Adopt incrementally



Simplify cloud management, Speed innovation, Reduce costs with IBM **Cloud** Automation Manager in IBM **Cloud** Pak for Multicloud Manager



Rapid Innovation

Accelerate delivery of applications into production

Improve productivity of

- cloud automation developers
- application service architects
- application developers



Efficient Operations

Standardize delivery of cloud infrastructure and application services

Manage cloud infrastructure as code (IaC) with DevOps practices

Enterprise grade cloud management with clear separation of roles for policy enforcement



Investment Leverage

Purpose built for integration with existing cloud management services

Integrate with existing services

- Application management services
- Data center services

Protect existing investments, leverage existing skills



Accommodate Future Needs

Modern micro-service architecture; extensible, flexible, agile

Hybrid ready; Support hybrid cloud and hybrid architecture applications

Compose bespoke cloud management solutions; Adopt incrementally



Let's put
smart
to work.

IBM

Thank you