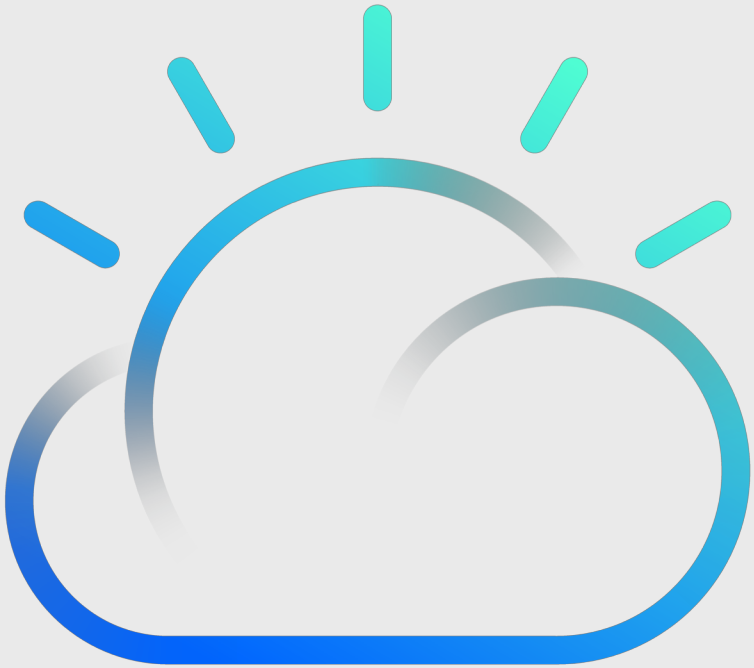
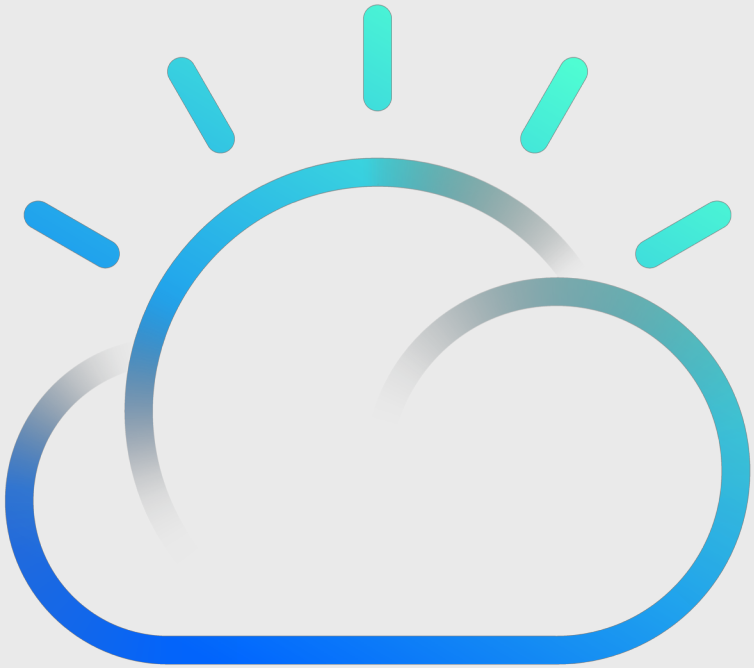
Cloud Automation Manager



IBM Cloud

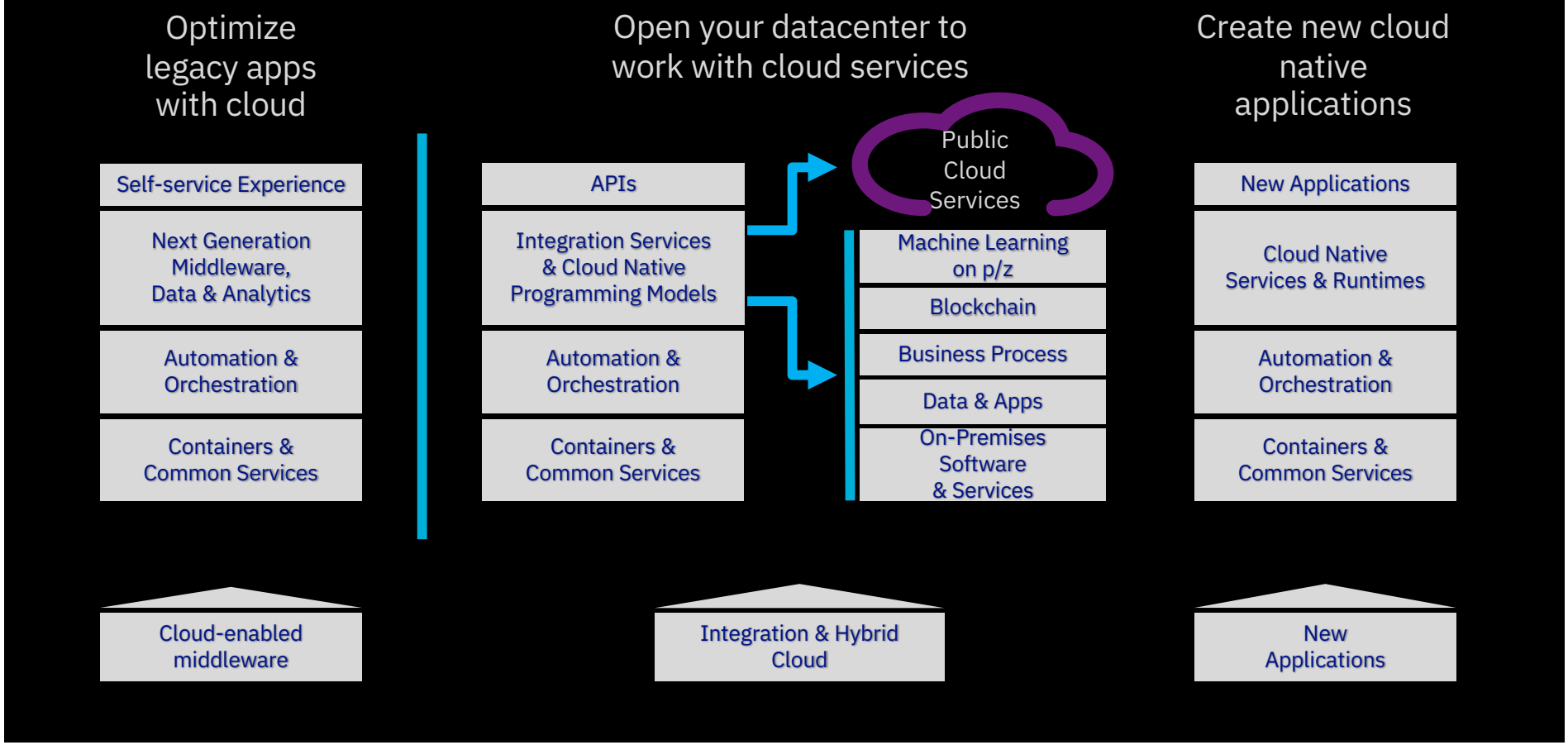
Level Set

The use cases and market forces, the architectural context within the overall IBM Cloud strategy and the functional components of the solution.



IBM Cloud

Use cases driving private cloud adoption



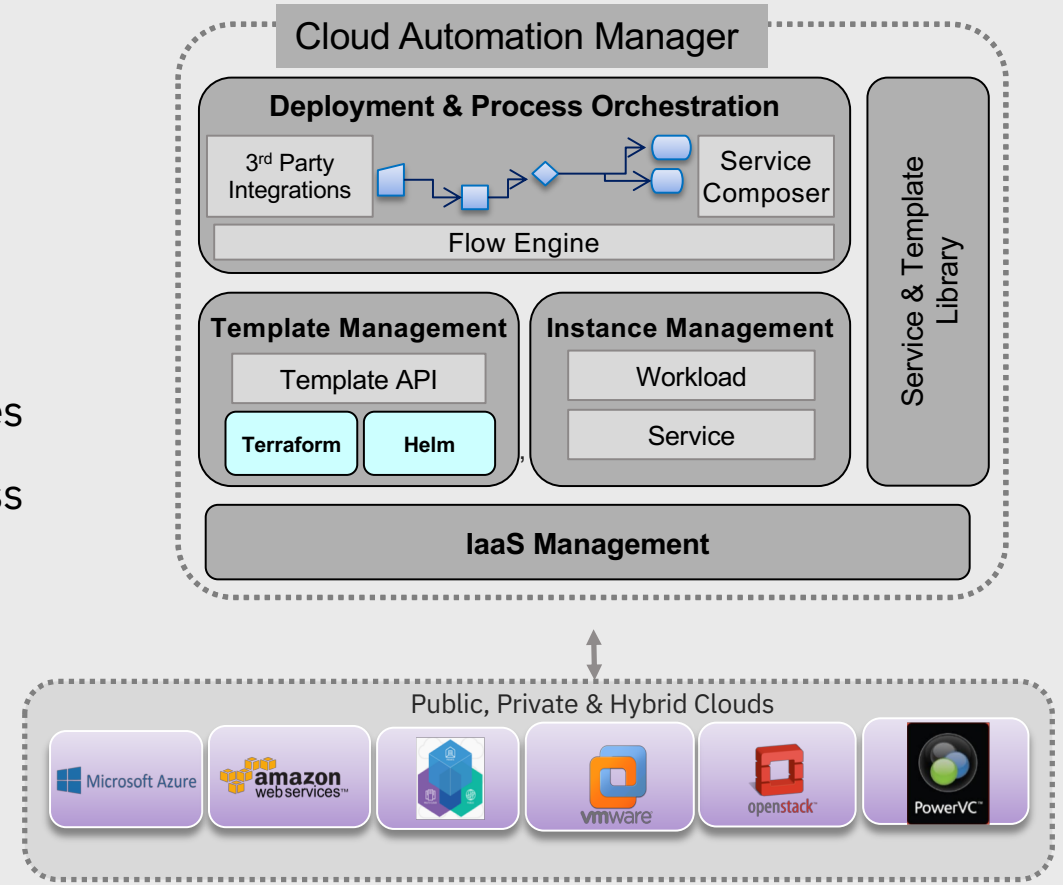
IBM Cloud Automation Manager: Full stack automation and service orchestration

Automated provisioning of infrastructure and applications with workflow orchestration

Self-service access to cloud infrastructure and application services

Manage and govern workloads across multiple and hybrid clouds

Built with **open technology** to avoid vendor lock-in



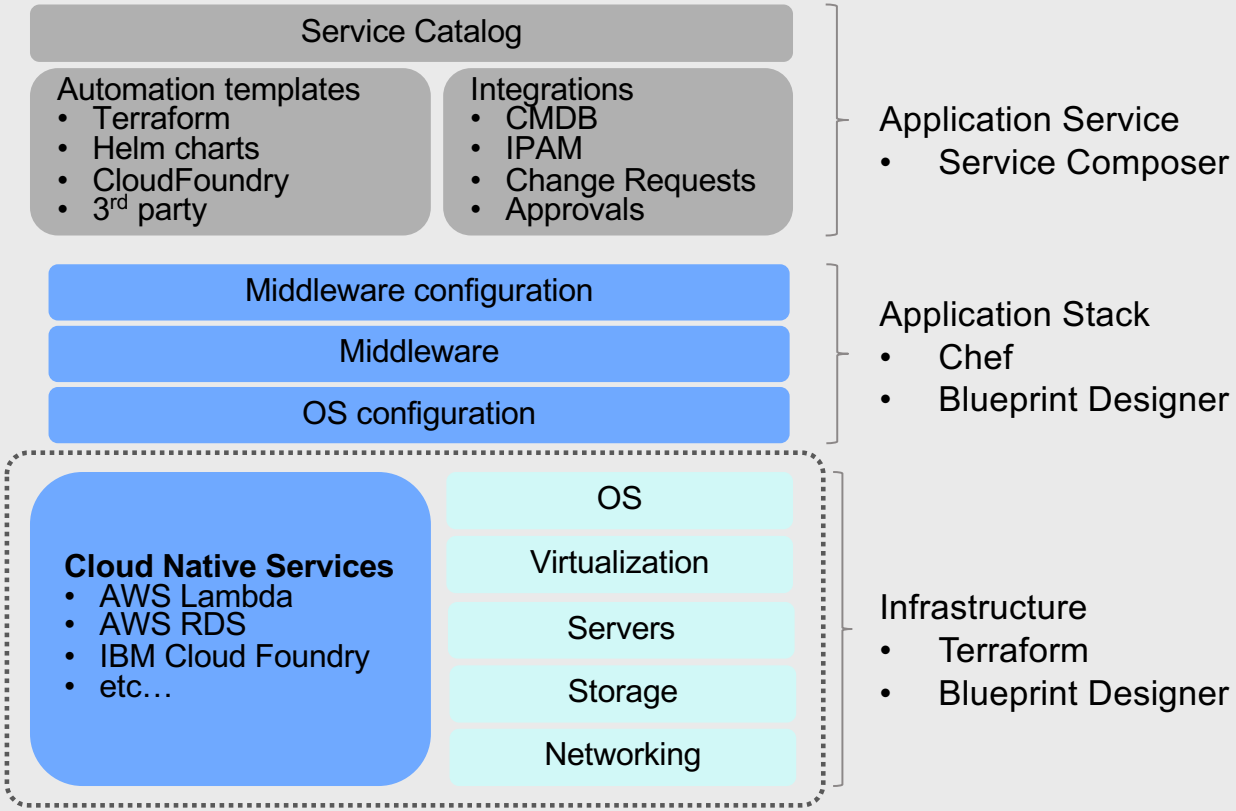
The Cloud Automation Manager business

All clouds, one tool

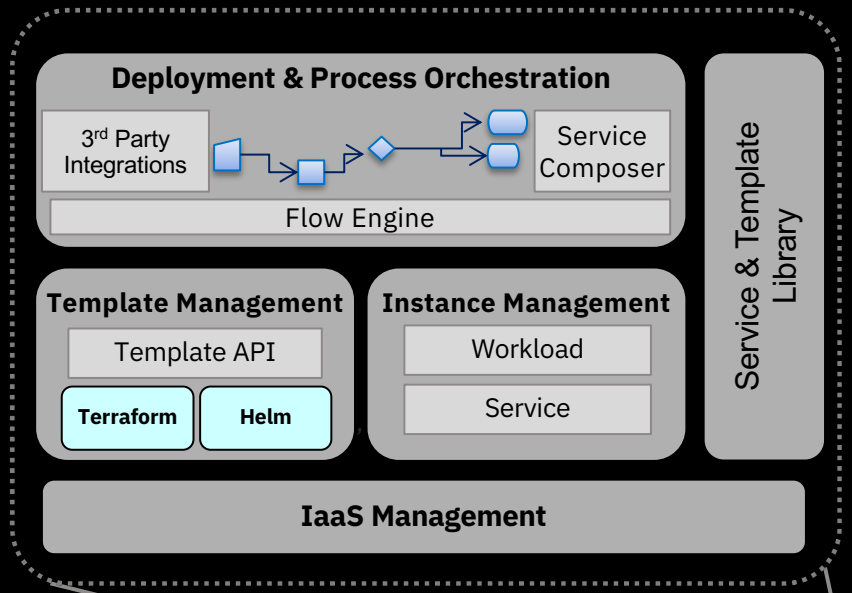
Any workload architecture

Graphical experience

Purpose build for integration



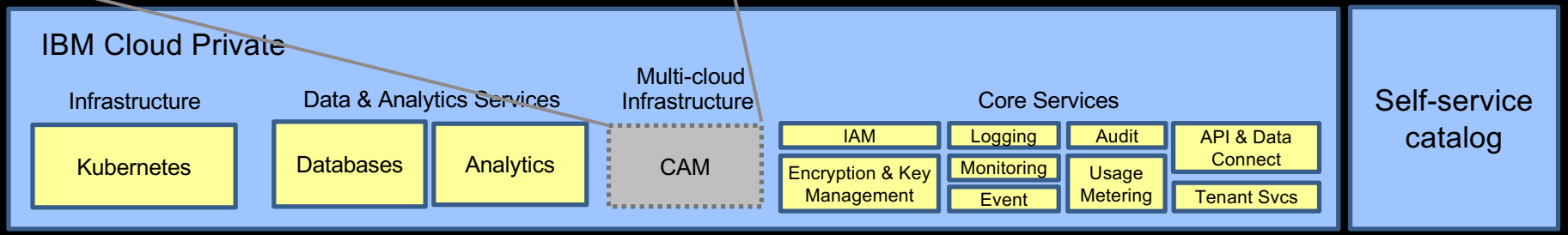
Cloud Automation Manager in IBM Cloud Private



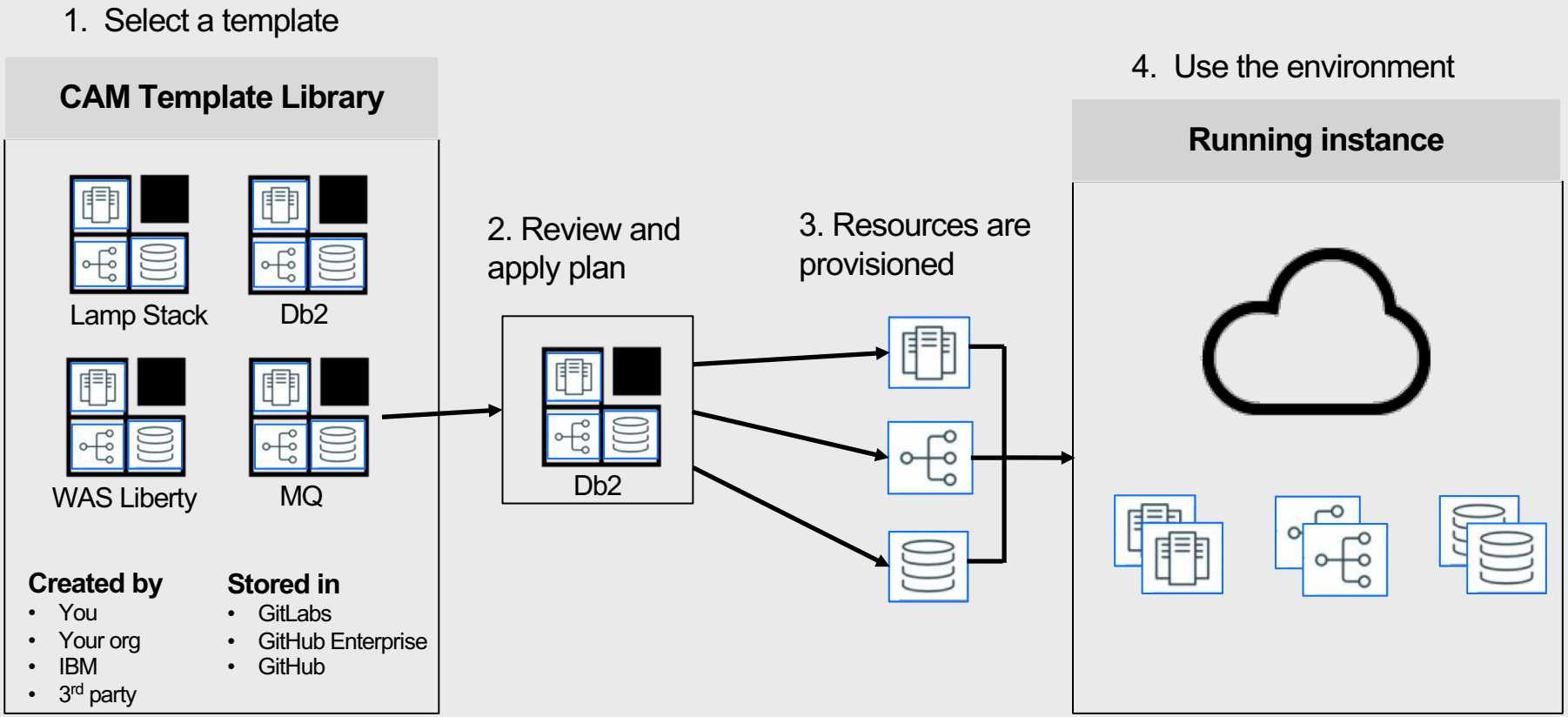
Containerized cloud native application

Installed into an IBM Cloud Private worker node with a Helm chart

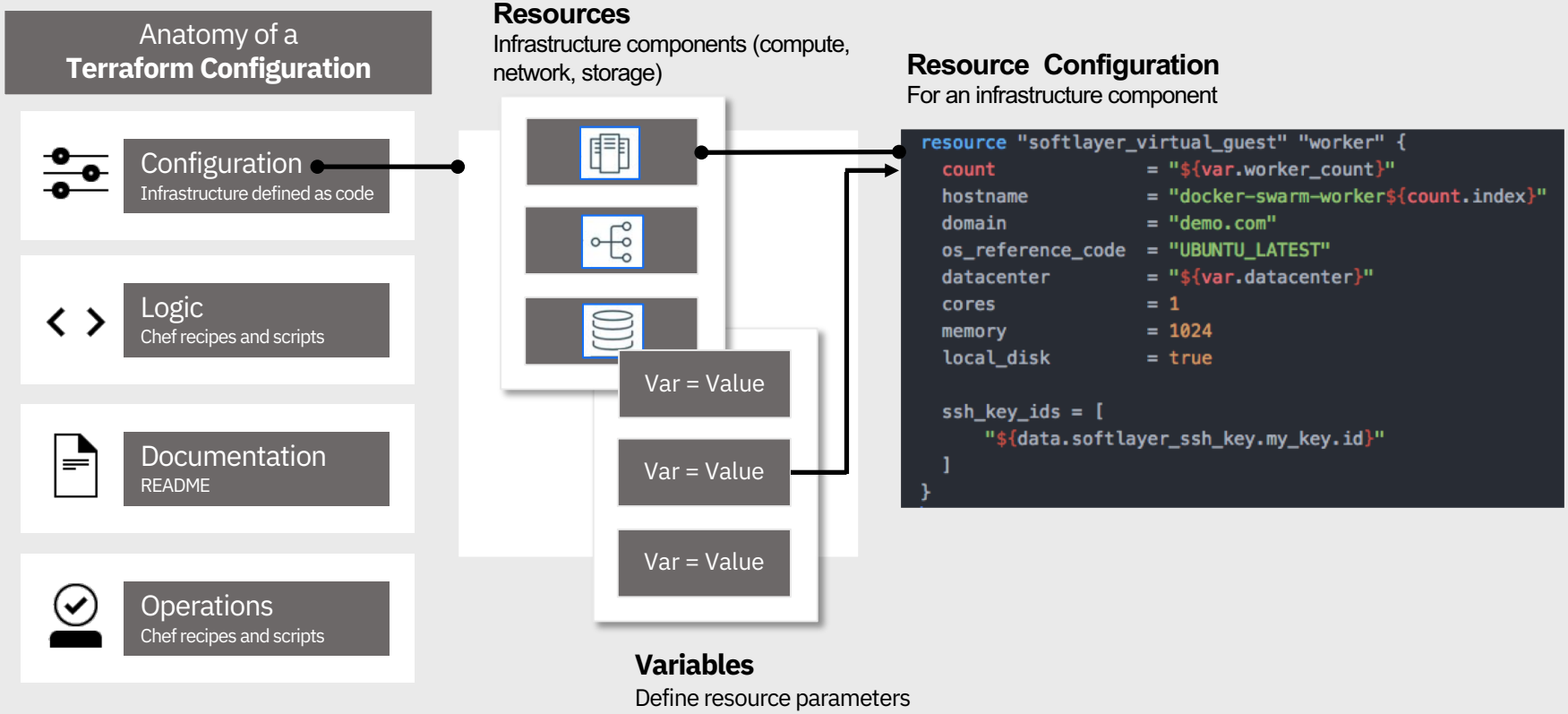
Leverages IBM Cloud Private services for enterprise capabilities



Using Terraform Configurations



Anatomy of a Terraform Configuration



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}"
  ]
}
```


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

Template lifecycle – Plan/Apply

Allows user to make changes on running instances

Two major scenarios:

1. User want to change a parameter on an instance
    - Example – change memory/cpu
2. User wants to update to a new version of the template used to create their instance
    - Example – add more VMs or other components

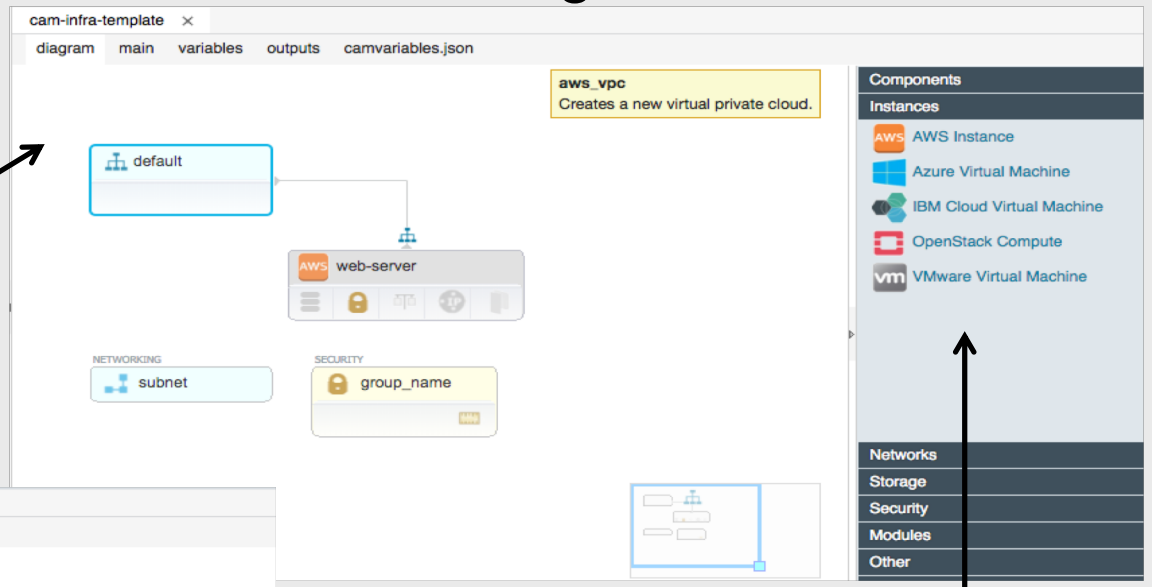
Instance modification/update in a two-step process:

1. Plan - allows user to select the version and /or change parameters and execute a phase to see what will change in the environment
2. Apply – executes the changes presented on plan action on the environment

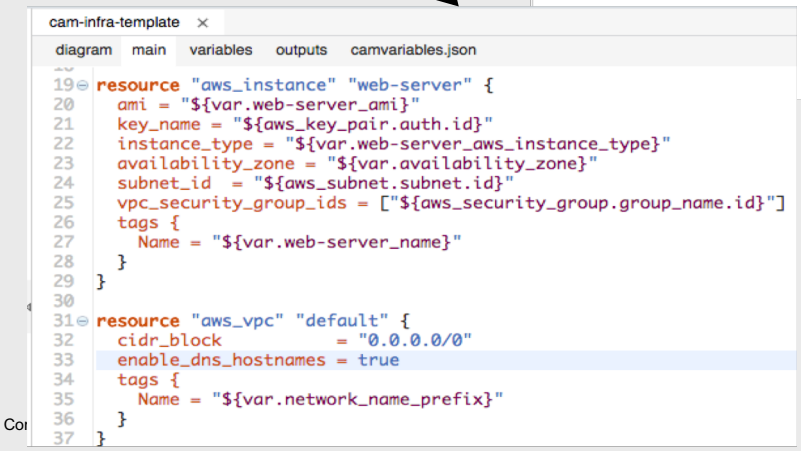
Creating Terraform templates – CAM Template Designer

- 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
- Integration with UrbanCode Deploy to automate full-stack application deployments

CAM Template Designer – Terraform Design



Graphically create Terraform templates while leveraging Terraform constructs.



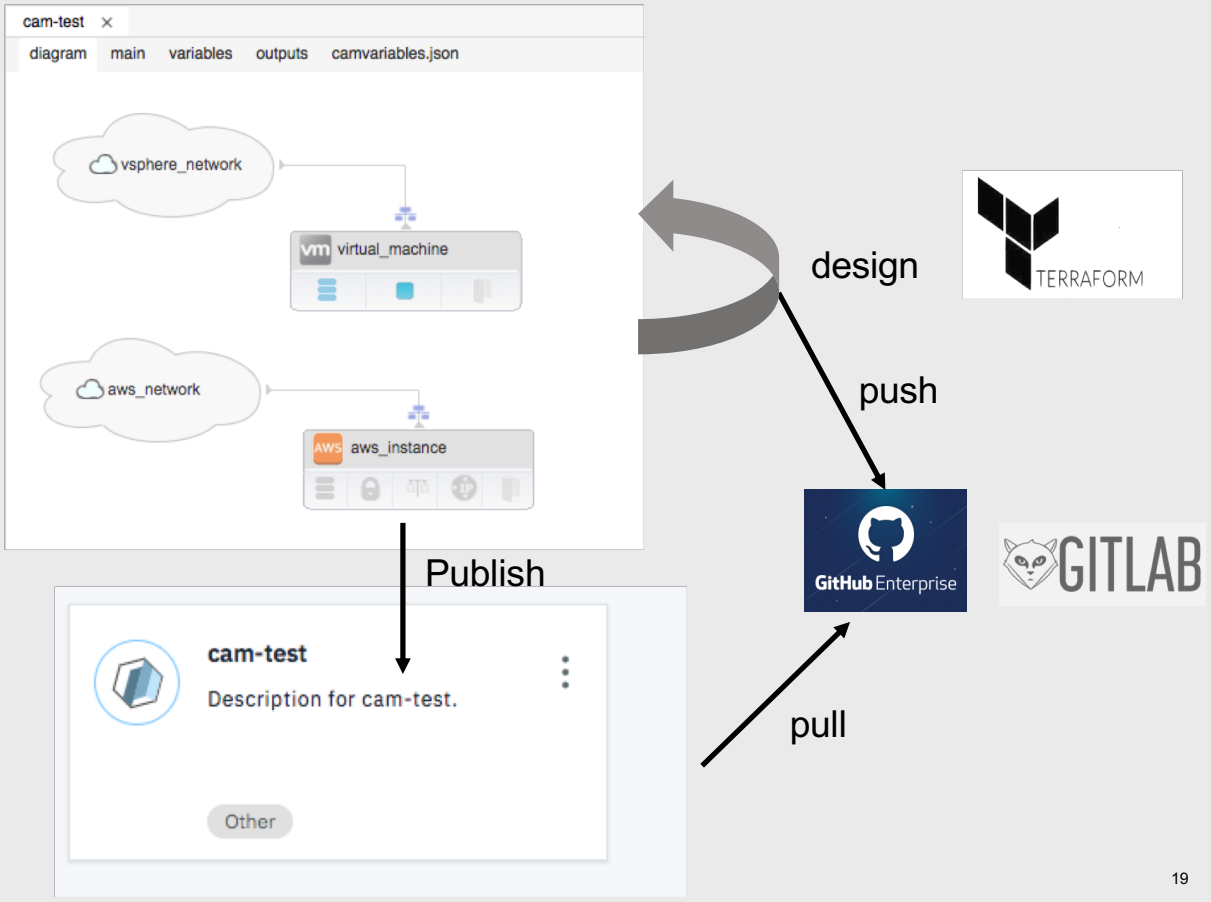

```
resource "aws_instance" "web-server" {
  ami = "${var.web-server_ami}"
  key_name = "${aws_key_pair.auth.id}"
  instance_type = "${var.web-server_aws_instance_type}"
  availability_zone = "${var.availability_zone}"
  subnet_id  = "${aws_subnet.subnet.id}"
  vpc_security_group_ids = ["${aws_security_group.group_name.id}"]
  tags {
    Name = "${var.web-server_name}"
  }
}

resource "aws_vpc" "default" {
  cidr_block           = "0.0.0.0/0"
  enable_dns_hostnames = true
  tags {
    Name = "${var.network_name_prefix}"
  }
}
```

Rich cloud support

CAM Template Designer – Create CAM Templates

1. Create CAM project
2. Edit Terraform
3. Publish to CAM



CAM Template Designer – UrbanCode Deploy Integration

UCD Terraform provisioner

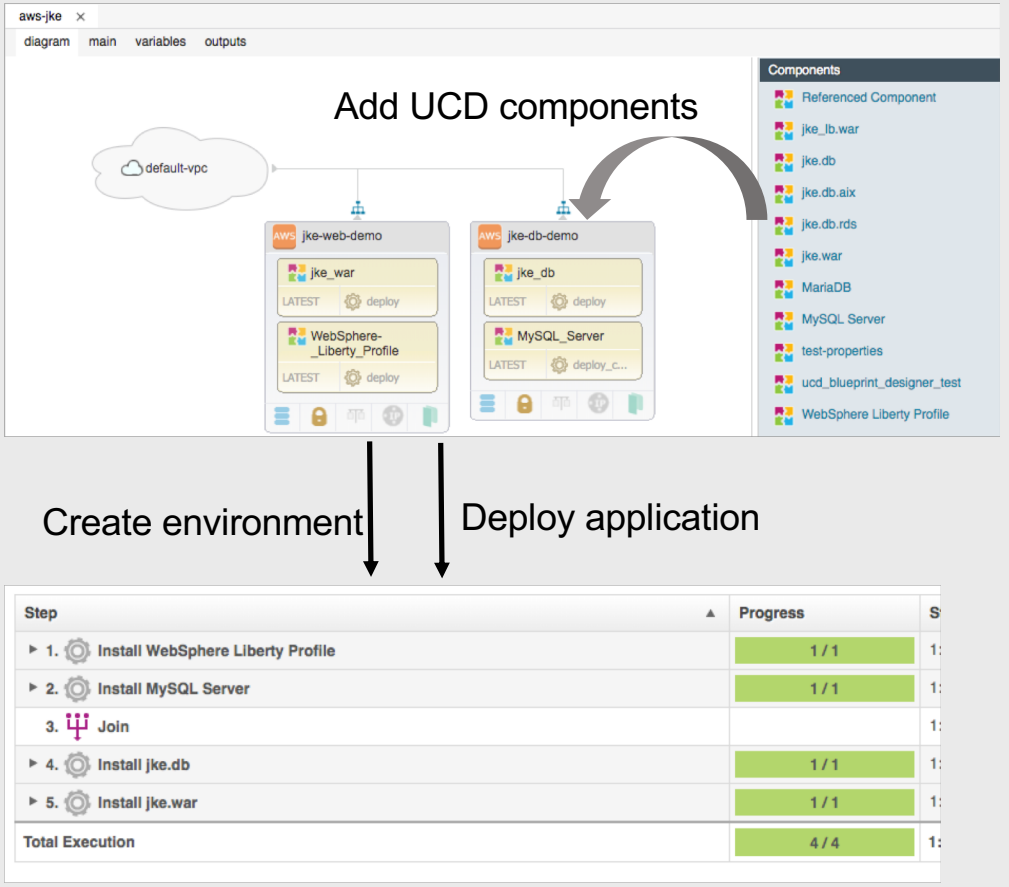
- Install UCD agent
- Included in CAM Terraform runtime

UCD Terraform provider

- Resources to automate UCD
- Create resource tree, environment, mappings
- Execute application and component processes
- Included in CAM Terraform runtime

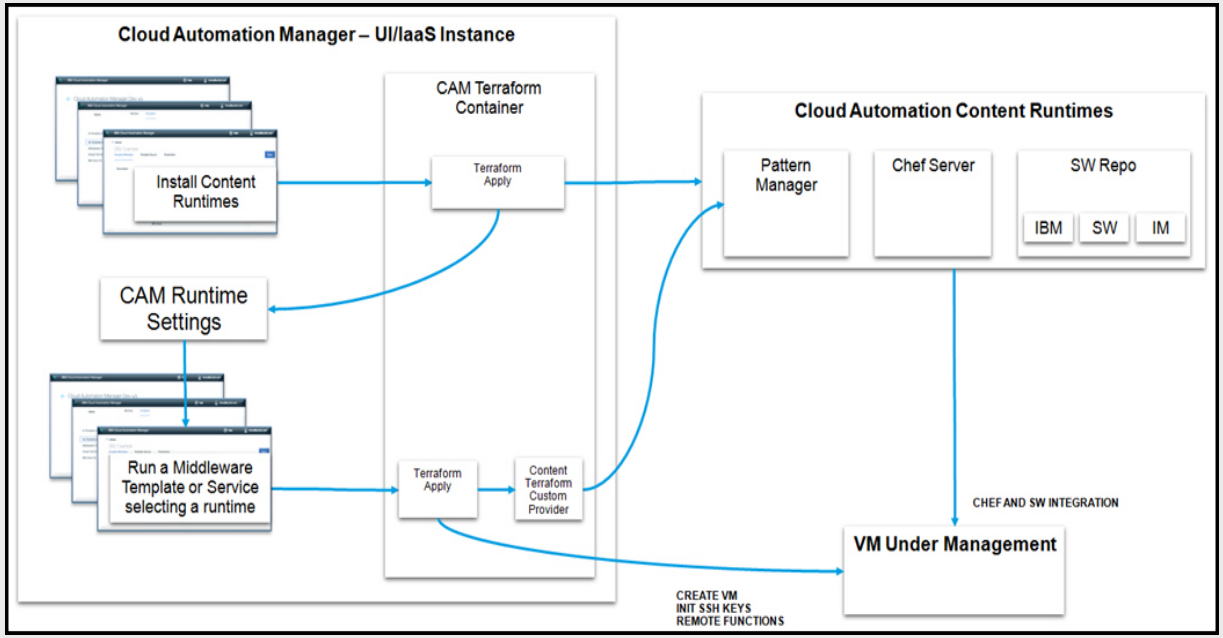
UCD Designer integration

- View UCD components in palette
- Drag and drop components onto diagram
- Automatically create UCD Terraform resources



Cloud Automation Manager: Chef content runtime support

Built-in Chef Server runtime



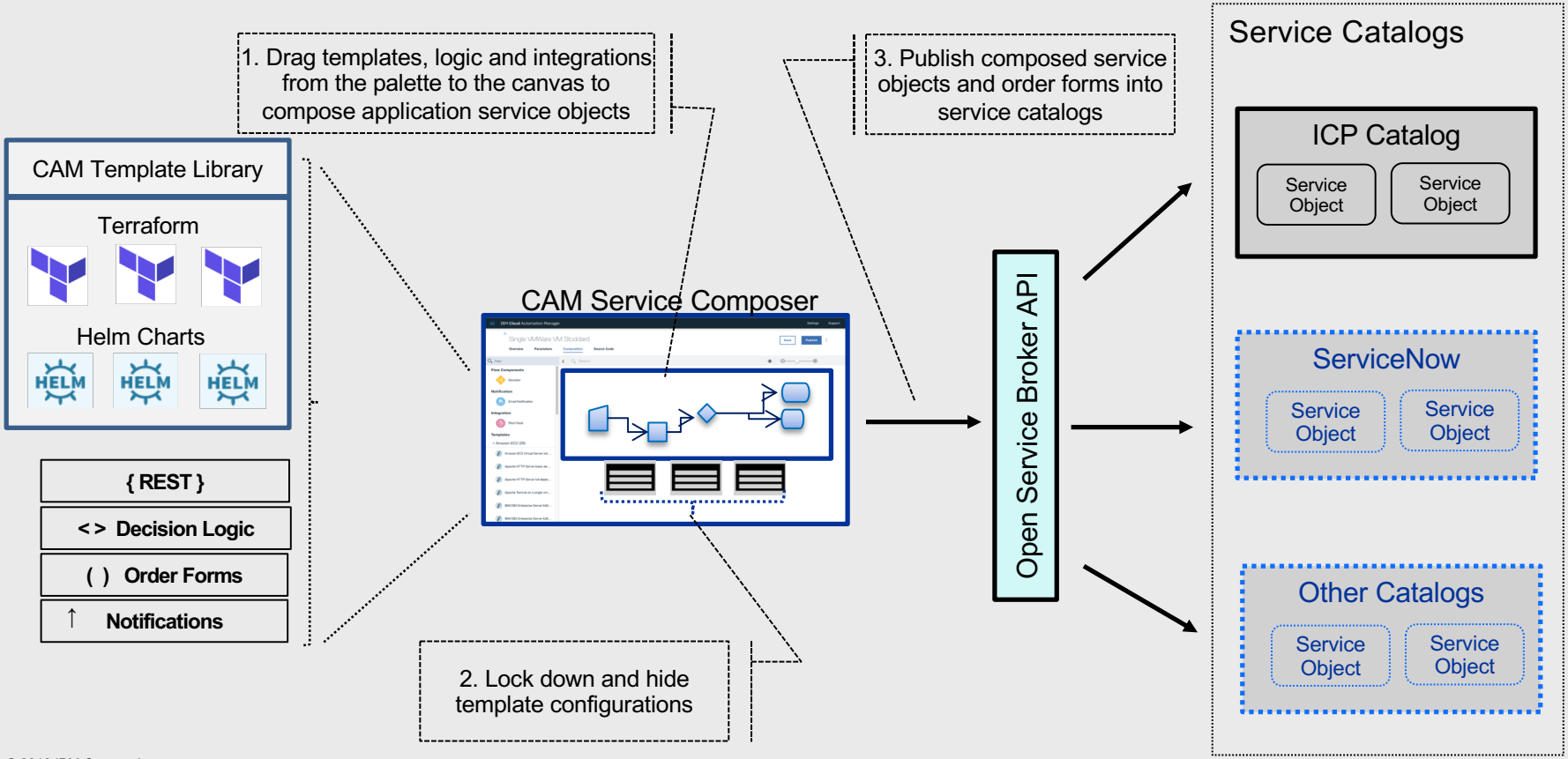
Optional Chef runtime that can be easily deployed into your provider cloud to support IBM Chef enabled content

You supply the virtual server, let CAM stand-up a pre-configure Chef runtime

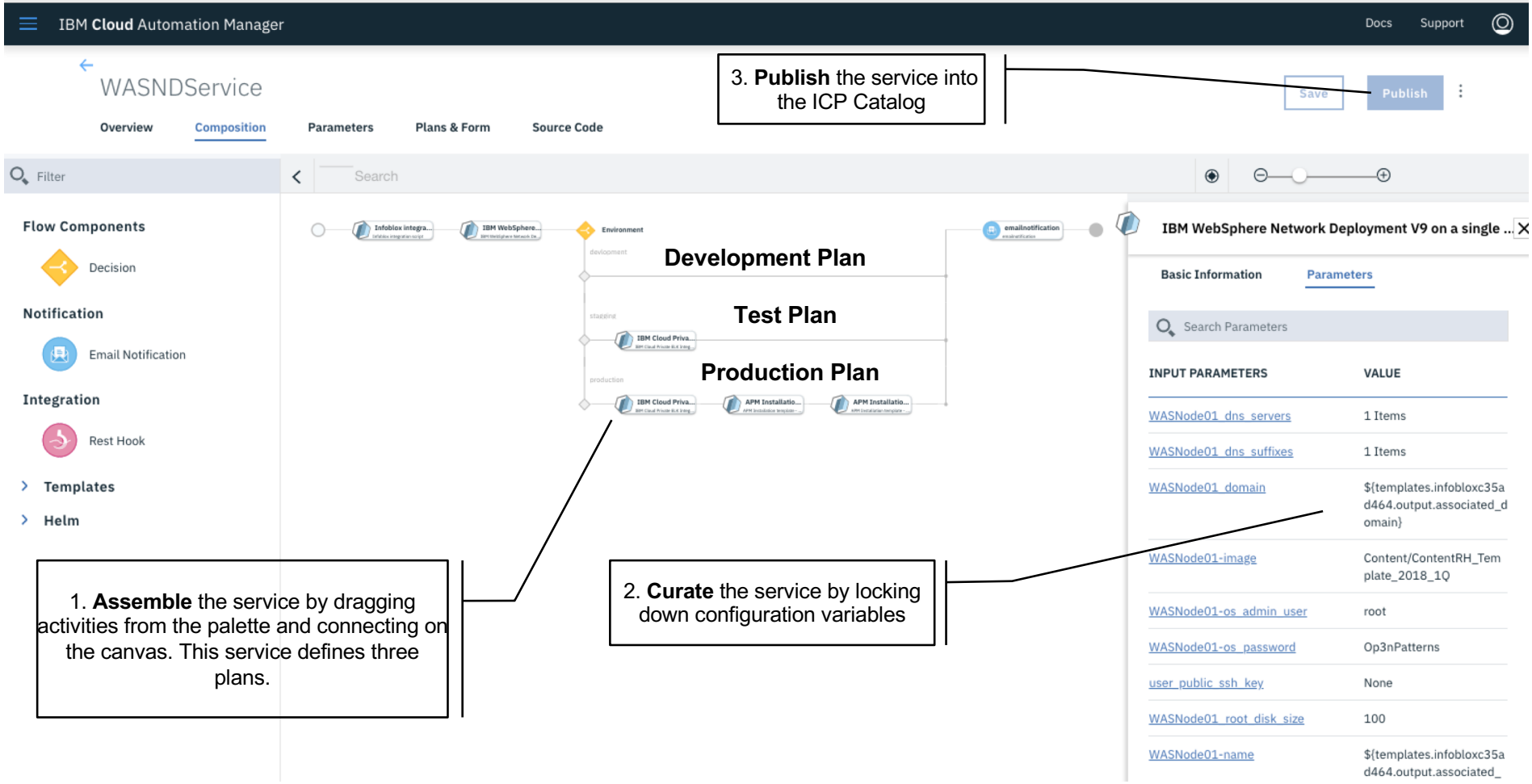
Fully containerized for maximum portability

Integrated software repository

Service Composition and Flow Engine



Assemble, Curate, Publish - with Service Composer



Resources

CAM Knowledge Center

https://www.ibm.com/support/knowledgecenter/SS2L37

CAM Developer Portal (Blogs, Tutorials, Videos)

http://developer.ibm.com/cloudautomation