Terraform Template Builder

Market & objective details

IBM®

Market opportunity

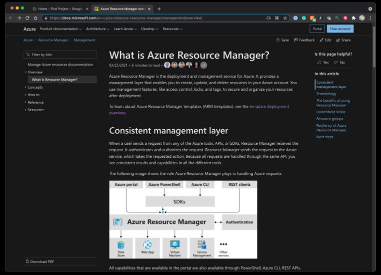
- At scale, development teams do not use the UI or CLI to build complex environments.
- A declarative IaC approach is more effective using Git repos as the source of truth and a common language to provision resources - Terraform in our case.
- IaC advantages: auditability of setup, drift detection, managed scaling, automation via GitOps
- Enterprise architects need to be able to effectively scale and provision resources across their organization.

Problem statement

IBM Cloud users have to manually set up
all infrastructure for new solutions
and need support for common
infrastructure-as-code standards.

Competitive landscape

All cloud providers have IaC solutions, often with their own languages that pre-date or compete with TerraForm by HashiCorp.



What is Azure Resource Manager?

03/25/2021 • 6 minutes to read • +1

Azure Resource Manager is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. You use management features, like access control, locks, and tags, to secure and organize your resources after deployment.

To learn about Azure Resource Manager templates (ARM templates), see the template deployment overview.

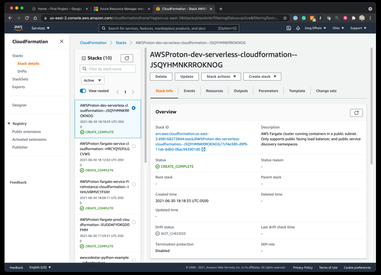
Consistent management layer

When a user sends a request from any of the Azure tools, APIs, or SDKs, Resource Manager receives the request. It authenticates and authorizes the request. Resource Manager sends the request to the Azure service, which takes the requested action. Because all requests are handled through the same API, you see consistent results and capabilities in all the different tools.

The following image shows the role Azure Resource Manager plays in handling Azure requests.

All capabilities that are available in the portal are also available through PowerShell, Azure CLI, REST APIs,

Resource Manager


Home – First Project – Google
Azure Resource Manager over
CloudFormation - Stack AWSP
us-east-2.console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/stackinfo?filteringStatus=active&filteringText=...
aws
Services
Search for services, features, marketplace products, and docs
[Option+S]
Greg Effrein
Ohio
Support
CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
Designer
Registry
Public extensions
Activated extensions
Publisher
Feedback
CloudFormation > Stacks > AWSProton-dev-serverless-cloudformation--JSQYHMNKRROKNOG
Stacks (10)
Filter by stack name
Active
View nested
1
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

AWSProton-dev-serverless-cloudformation--JSQYHMNKRROKNOG
2021-06-30 18:18:33 UTC-0500
CREATE_COMPLETE

AWSProton-fargate-service-cloudformation--HRCYQYGPJLGCVWS
2021-06-30 18:12:02 UTC-0500
CREATE_COMPLETE

AWSProton-fargate-service-FirstInstance-cloudformation--IHHUVBMVCYFIIAY
2021-06-30 18:09:11 UTC-0500
CREATE_COMPLETE

AWSProton-fargate-prod-cloudformation--EUDDAFYOKGDDFMM
2021-06-30 17:29:31 UTC-0500
CREATE_COMPLETE

awscodestar-python-example-infrastructure

AWSProton-dev-serverless-cloudformation--JSQYHMNKRROKNOG

Overview

Stack ID: arn:aws:cloudformation:us-east-2:498168273844:stack/AWSProton-dev-serverless-cloudformation--JSQYHMNKRROKNOG/7cf4e380-d9f9-11eb-8db0-06ac943901d0

Description: AWS Fargate cluster running containers in a public subnet. Only supports public facing load balancer, and public service discovery namespaces.

Status: CREATE_COMPLETE

Status reason: -

Root stack: -

Parent stack: -

Created time: 2021-06-30 18:18:33 UTC-0500

Deleted time: -

Updated time: -

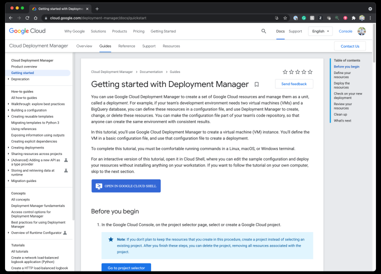
Drift status: NOT_CHECKED

Last drift check time: -

Termination protection: Disabled

IAM role: -

Cloud Formation


Getting started with Deployme
cloud.google.com/deployment-manager/docs/quickstart
Google Cloud
Why Google
Solutions
Products
Pricing
Getting Started
Docs
Support
English
Console
Cloud Deployment Manager
Overview
Guides
Reference
Support
Resources
Contact Us
Cloud Deployment Manager
Product overview
Getting started
Deprecation
How-to guides
All how-to guides
Walkthrough: explore best practices
Building a configuration
Creating reusable templates
Migrating templates to Python 3
Using references
Exposing information using outputs
Creating explicit dependencies
Creating deployments
Sharing resources across projects
(Advanced) Adding a new API as a type provider
Storing and retrieving data at runtime
Migration guides
Concepts
All concepts
Deployment Manager fundamentals
Access control options for Deployment Manager
Best practices for using Deployment Manager
Overview of Runtime Configurator
Tutorials
All tutorials
Create a network load-balanced logbook application (Python)
Create a HTTP load-balanced logbook
Cloud Deployment Manager > Documentation > Guides
Send feedback
Table of contents
Before you begin
Define your resources
Deploy the resources
Check on your new deployment
Review your resources
Clean up
What's next
OPEN IN GOOGLE CLOUD SHELL
Go to project selector

Getting started with Deployment Manager

You can use Google Cloud Deployment Manager to create a set of Google Cloud resources and manage them as a unit, called a *deployment*. For example, if your team's development environment needs two virtual machines (VMs) and a BigQuery database, you can define these resources in a *configuration* file, and use Deployment Manager to create, change, or delete these resources. You can make the configuration file part of your team's code repository, so that anyone can create the same environment with consistent results.

In this tutorial, you'll use Google Cloud Deployment Manager to create a virtual machine (VM) instance. You'll define the VM in a basic configuration file, and use that configuration file to create a deployment.

To complete this tutorial, you must be comfortable running commands in a Linux, macOS, or Windows terminal.

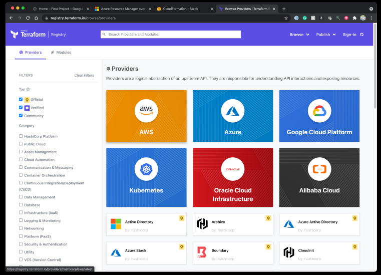
For an interactive version of this tutorial, open it in Cloud Shell, where you can edit the sample configuration and deploy your resources without installing anything on your workstation. If you want to follow the tutorial on your own computer, skip to the next section.

Before you begin

1. In the Google Cloud Console, on the project selector page, select or create a Google Cloud project.

Note: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

Deployment Manager



Providers

Providers are a logical abstraction of an upstream API. They are responsible for understanding API interactions and exposing resources.

☑ Official
☑ Verified
☑ Community

Category

☐ HashiCorp Platform
☐ Public Cloud
☐ Asset Management
☐ Cloud Automation
☐ Communication & Messaging
☐ Container Orchestration
☐ Continuous Integration/Deployment (CI/CD)
☐ Data Management
☐ Database
☐ Infrastructure (IaaS)
☐ Logging & Monitoring
☐ Networking
☐ Platform (PaaS)
☐ Security & Authentication
☐ Utility
☐ VCS (Version Control)

Terraform Registry

Personas

Allan
Enterprise Architect

- Works at large company with many developers
- Already familiar with infrastructure as code
- Seeking to streamline team efforts
- Values code consistency and scaling

Primary objective

Allan, an enterprise architect, can customize and provision a basic, compliant solution on the cloud in 15 minutes or less.

🟪 Who
🟥 What
🟩 Wow

Prior research: 2018-present

Generative & Evaluative studies 2019-2020

This benchmarking study follows multiple rounds of generative and evaluative research over the past several years:

- Generative research in 2018-2019 highlighted user needs for automating access management, duplicating projects and environments, and opportunity to make environments easier to self-provision.

- Project Kong, a 2019 cloud-wide evaluative study, showed a lack of integrated task flows and opportunity to better automate environment set-up

SYNTHESIS

IBM Confidential

IBM Design | Hybrid Cloud

11

IBM MD&I

IBM CONFIDENTIAL

IBM Developer Advisory Council

Q2 - 2019

May 10, 2019

Alex Torres - Senior Market Research Advisor, MD&I - Cloud Platforms and Infrastructure
Megan McNamee - Principal Advisor, Market Development Cloud Platforms and Infrastructure – Developers
Melissa Hennessey - Manager, Market Development & Insights, Cloud Platforms & Infrastructure
Tom Karpowitz - Principal, Market Development & Insights, Cloud Platforms & Infrastructure

IBM

Jul-Aug 2021: Benchmarking methods and overview

Goals

- Establish baseline usability measurement of Terraform Template Builder for future comparison

- Identify immediate usability improvements and opportunities for enhancements

- Examine discoverability of IBM’s Terraform solutions prior to DUX

Participant overview

- Roles included Cloud Engineer, Software Architect, Solutions Architect, similar
- All participants experienced in Terraform, from beginner to advanced
- Tool use varied, including other relevant cloud infrastructure tools

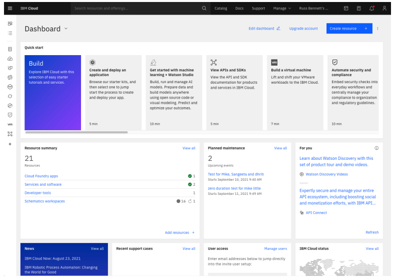
Sample size

n = 9 participants

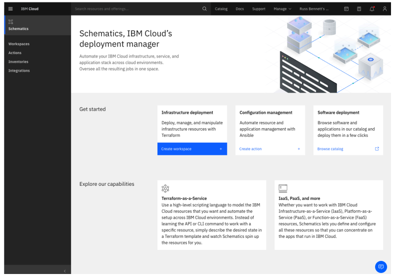
Session design

75 min, moderated

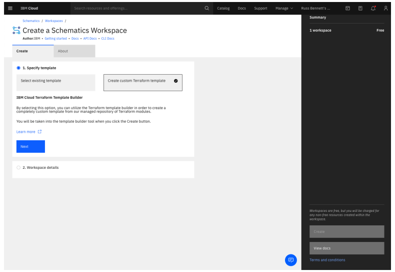
Code demo



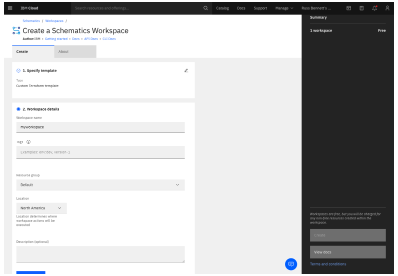
Global dash



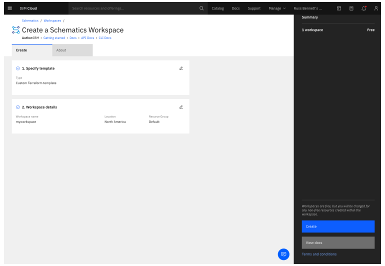
Schematics home



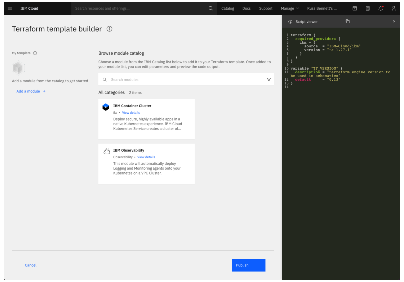
Schematics Workspace create



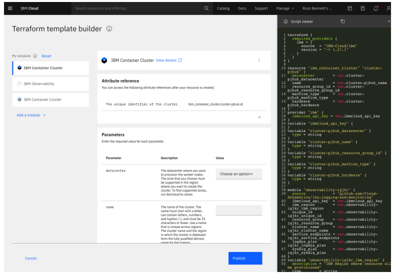
Schematics Workspace create



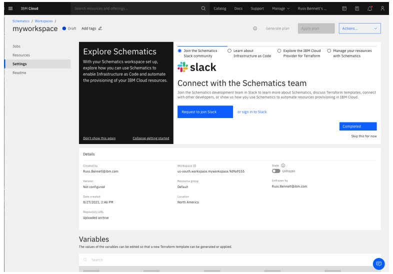
Schematics Workspace create



Template builder empty state



Template builder edit state



Schematics workspace manage