HOW TO OPTIMIZE YOUR CLOUD

IN THE ERA OF INFRASTRUCTURE AS CODE

Densify

Leverage Optimization as Code to Achieve Continuous Optimization That Fits into Your DevOps Continuous Integration/Continuous Delivery (CI/CD) Pipeline

Speed, agility, and efficiency—they're what every organization wants from the cloud. They're all about going faster, changing direction faster, and doing it with less risk and cost.

For CIOs and IT leaders, this is especially important—they are very driven to innovate and come up with more tailored services, as well as unique and better customer experiences for their end users—all driven by new applications. As they deploy more apps into the public cloud seeking scalability, elasticity, and agility, organizations are embracing DevOps culture to move faster, and at the same time, they are leveraging infrastructure as code (IaC) technologies like HashiCorp Terraform and AWS CloudFormation to automate cloud resource selection and achieve operational efficiency during the process.

However, this level of granular automation in cloud provisioning poses challenges for even the most sophisticated cloud operation teams around managing and optimizing the infrastructure on which those key applications are running.

The Challenge: Hard-Coded Infrastructure as Code

Cloud Micro-Purchasing and Manual Specifications Drive Potential Risks and Unnecessary Spend

As DevOps teams move faster and deploy more apps into the cloud using processes such as Continuous Integration and Continuous Delivery (CI/CD), it usually drives a lot of cloud micropurchasing—and as a side effect, creates cloud sprawl. In fact, one organization may have thousands of cloud micropurchasing transactions happening during any given month, creating challenges for operations teams that must systematically manage and optimize theses countless, granular cloud resource provisioning processes.

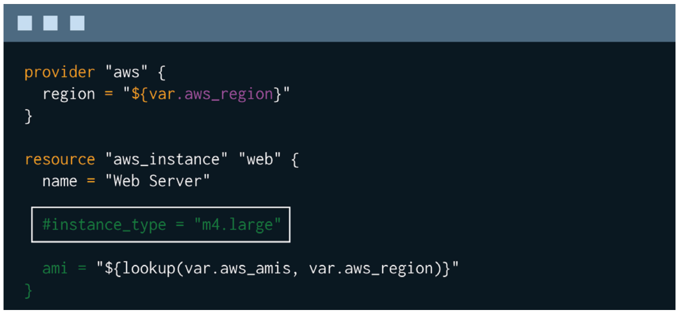
Coupled with this challenge is the manual specifications inherent infrastructure as code templates. Infrastructure as code tools simplify and automate the process of managing and provisioning infrastructure, but two challenges are often inadvertently encountered.

1. Developers often leverage tribal knowledge to manually size cloud instances, provisioning them larger (increasing costs) or smaller (introducing business outage risk) than required—or even selecting totally wrong instance types

2. These incorrect requirements become hardcoded through declarative IaC—even if you correct the instance type and size on the fly within your cloud, the instance will be kicked back to the original misconfiguration after every restart.

Hardcoded infrastructure as code not only limits the power of automation, but more importantly, creates operational risks and unnecessary cloud spend.

Below is an example of a Terraform template that automates the provisioning of an Amazon EC2 *m4.large* instance. It is hardcoded, even though that instance type and size may not consistently be optimal for that given workload.



```
provider "aws" {
  region = "${var.aws_region}"
}

resource "aws_instance" "web" {
  name = "Web Server"

  #instance_type = "m4.large"

  ami = "${lookup(var.aws_amis, var.aws_region)}"
}
```

The Solution: Optimization as Code

The Next Frontier of Infrastructure as Code Enables *Continuous Optimization*, Seamlessly Fitting into Your DevOps CI/CD Pipeline

Now that we understand that hardcoding cloud resource specifications is not best practice, we need to automate and parameterize the line of code in question to get the accurate cloud resource specifications that will make sure it is perfectly and continuously aligned with what your applications really need. This is *optimization as code*, and here is how to achieve it to enable *continuous optimization* for your cloud infrastructure.

Step One: Self-Aware Infrastructure – Imagine If Your Cloud Apps Just Knew

What if machine learning could make your cloud apps self-aware of their exact resource requirements, in any moment and over time? This is possible with sophisticated analytics that continuously model your apps' utilization patterns and compare these models against complete and holistic knowledge of all offerings across the major cloud providers.

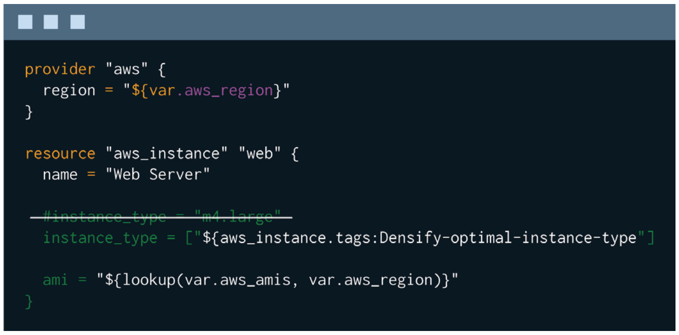
Each of your workloads has CPU, memory, and I/O parameters that fluctuate all day and may evolve over time. Multidimensional permutation analytics, packaged in an analytics and automation engine, can model these workload patterns and generate accurate recommendations for optimal instance types and sizes for each app and workload. This engine can then:

- Inject these findings into your cloud instances as tags to make the apps *self-aware*
- Build impact analysis reports for providing transparency, evidence, and benefits to application owners within the business, and attach these reports directly to change tickets for approval
- Document recommendations in machine-readable map files that can be automatically synced to repos like GitHub for optimization automation (for step two)

With self-aware cloud apps, developers no longer need to rely on tribal knowledge when selecting and sizing cloud instances. An analytics engine automatically provides a perfect, data-validated recommendation *every* time..

Step Two: Continuous Optimization – The Next Frontier of Infrastructure as Code

The next step for organizations that are successfully leveraging self-aware infrastructure is to evaluate *continuous optimization* (CO) as an addition to their CI/CD practice—an approach that makes infrastructure automatically self-optimizing. By replacing fixed resource assignments in your infrastructure as code templates with calls to reference the analytics engine's recommendations (as shown top right), your cloud apps and workloads can be automatically reprovisioned to the best-fit instances. And, as app resource demands change over time, or newer instance types are introduced by cloud providers, your infrastructure remains optimized. When your organization embraces continuous optimization, hardcoded instance selections in your infrastructure as code templates are replaced by calls to the analytics engine to leverage recommendations based on your apps' exact needs. Your cloud infrastructure becomes self-optimizing, 24/7.



```
provider "aws" {
  region = "${var.aws_region}"
}

resource "aws_instance" "web" {
  name = "Web Server"

 ~~#instance_type = "m4.large"~~
  instance_type = ["${aws_instance.tags:Densify-optimal-instance-type"]

  ami = "${lookup(var.aws_amis, var.aws_region)}"
}
```

Benefits of Optimization as Code

- Improves your environment's stability and application performance
- Significantly increases your cloud cost efficiency
- Automated, Continuous Optimization as your applications' demands evolve and change
- Operational excellence in the way your organization leverages and selects instances and deploys applications in the cloud
- Optimization as code tightly fits into your CI/CD pipeline, facilitating a new DevOps paradigm of CI/CD/CO (continuous integration/continuous delivery/continuous optimization)

Everybody Wins: Bridge the Gap Between App Developers, CloudOps, & Finance

With "Optimization as Code", every key cloud management stakeholder in your organization wins.

1. ***Developers*** can embed code snippets into their Infrastructure as Code templates to make them self-aware and self-optimizing, freeing them from the burden of determining granular resource requirements.
2. ***Cloud operations teams*** can provide API-based optimization services to app teams, enabling them to run a safe and efficient environment.
3. ***Finance*** benefits from the dramatically-improved cloud cost efficiency.

About Densify

Densify provides next-generation cloud optimization technology that makes cloud instances and containers self-optimizing by continuously and precisely matching app demands to the right cloud resources. Delivered as code, Densify integrates with popular Infrastructure as Code (e.g. Terraform) to deliver continuous optimization that fits into the DevOps CI/CD framework. With Densify, you'll achieve operational excellence with better app performance and lower cloud spend, in an automated fashion.

See Optimization as Code in Action

Visit www.densify.com/code