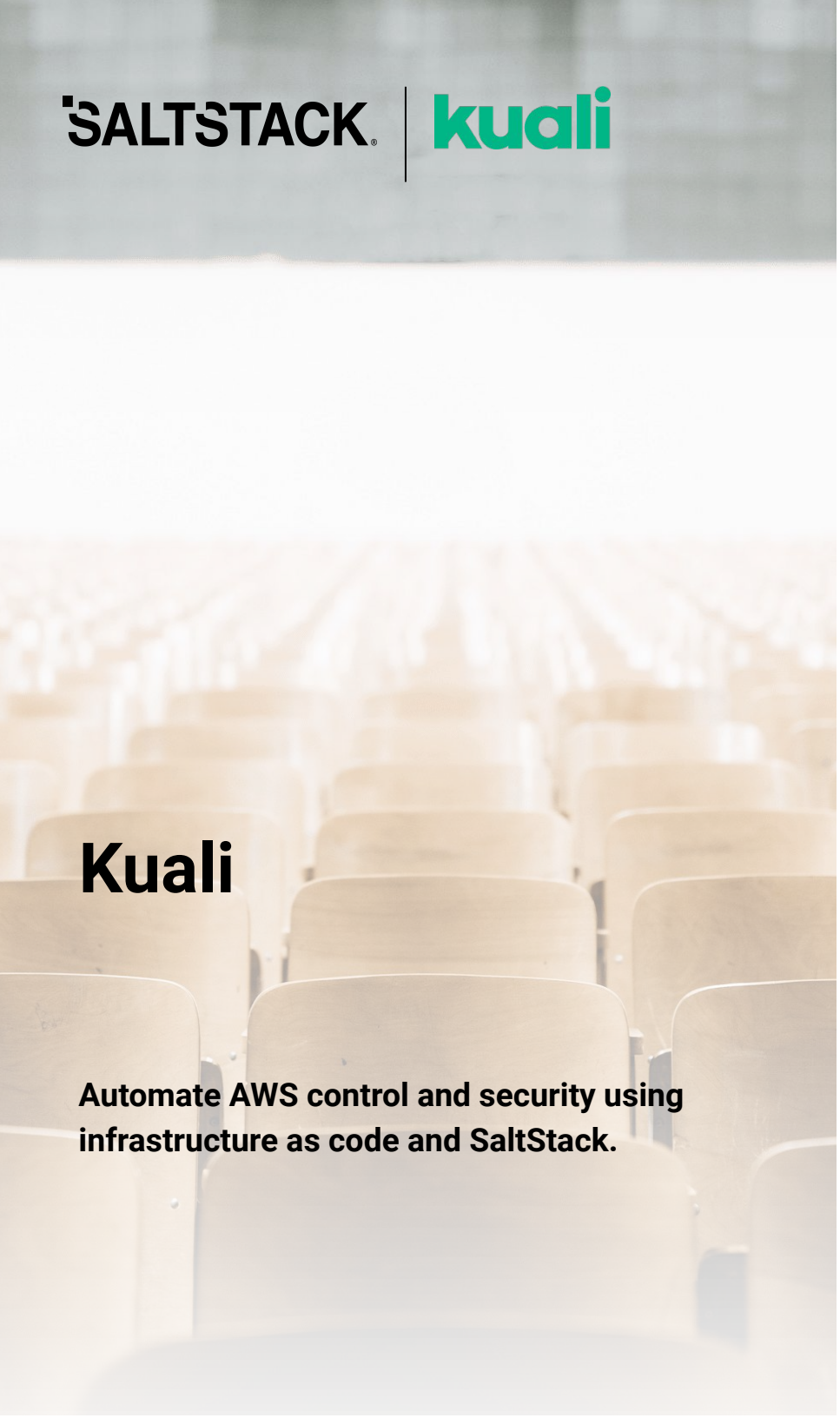


Kuali

Automate AWS control and security using infrastructure as code and SaltStack.

SUMMARY

Kuali sells multiple SaaS products, each with its own self-steering product team and unique AWS environment. The SRE team at Kuali uses SaltStack—combined with GitHub and Docker—to provide a best-practice IaC management system across AWS that delivers consistency and governance while simultaneously empowering each team to achieve diverse objectives.

RESULTS

Version control

Portability

Ease of use

SOLUTION

- SaltStack event-driven automation engine
- Github integration
- AWS integration
- Slack integration
- Docker support

"SaltStack helps us normalize control of AWS across teams. Through automated infrastructure security and management, our developers can focus on bringing revenue-generating products to market, instead of infrastructure maintenance."

Ben Gridley
Lead Site Reliability Engineer at Kuali

About Kuali

Kuali builds SaaS for higher education focused on five categories: Financials, Research, Curriculum and Catalog Management, Build (no-code applications), and Ready (business continuity). Each product is hosted in a separate AWS environment and managed by a self-steering product team.

The challenge: Wrangling an AWS wild west without stifling quality or speed

Ben Gridley, a lead site reliability engineer for Kuali, described the challenge with allowing each product team to manage their own AWS environment, "We found that each team would just go into the AWS console and click around until it did what they wanted. And, oftentimes they'd make changes, things would break, and nobody would discover the broken thing until the next day. Then, they'd forget that the changes were made or that it was even related to the broken thing at hand."

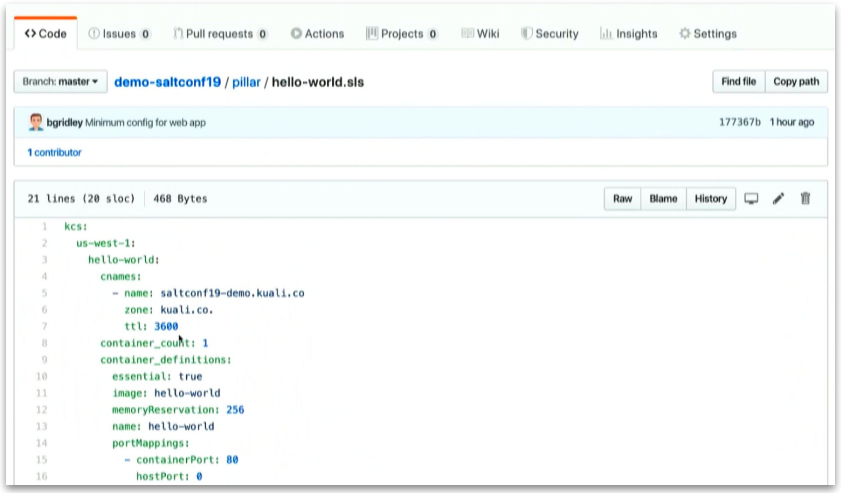
The SRE team at Kuali needed a solution that allowed them to use best-practice version control, enforce consistent security policies, and create tool and process cohesivity across environments—all while still enabling the individual product teams with the right degree of autonomy and control.

The solution: Governed self-service infrastructure automation

After evaluating several solutions—ranging from using only native AWS tools to creating custom proprietary tools—Kuali chose to manage its entire infrastructure with SaltStack, citing speed and performance, flexibility and pluggability as reasons for the decision.

SaltStack provides the team at Kuali with several unique advantages, including the ability to make changes to existing infrastructure without service disruption while preventing potential errors and vulnerabilities.

By leveraging SaltStack on AWS, Kuali was able to apply SaltStack infrastructure automation to address configuration irregularities, simplify infrastructure-as-code, reduce complexity, and apply version control to enable changes without any infrastructure configuration drift.



Integrating SaltStack with AWS allows teams to provision and manage all resources across multiple AWS environments using simple infrastructure as code templates.

Technical overview

In order to provide a full AWS automation solution that was both portable and version controlled, the SRE team at Kuali leveraged SaltStack, combined with Docker containers for portability and GitHub for version control.

The Kuali implementation of SaltStack is built on three simple facets: External Pillar, environment files, and Salt Cloud.

External Pillar

The external Pillar is a SaltStack feature that allows a user to define data and make it available to a system in SaltStack. Kuali engineers can create PIllar data, and subsequently create and update their IaC, by simply updating a human-readable YAML template in GitHub. In this way, engineers can manage their entire AWS environment—including vpcs, load balancers, S3 buckets, etc—in a familiar GitHub interface without being required to learn a new language or DSL, or even interact with SaltStack directly.

```
routes:
  us-west-1:
    public:
      vpc_name: demo-saltconf19
      routes:
        route1:
          destination_cidr_block: 0.0.0.0/0
          internet_gateway_name: igw-demo-saltconf19
      subnet_names:
        - public-b
        - public-c
```

Engineers can define AWS IaC using simple, human-readable YAML in GitHub. Unlike other IaC tools, SaltStack does not require users to learn a DSL or use a complex language like Ruby.

Environment files

Once the Pillar data is defined, the image is deployed in a Docker container and external variables, such as AWS credentials and GitHub credentials, are passed to provide proper access and direction. Additional optional variables can also be passed to enable a wide range of functionality. For example, the Kuali team passes a Slack token that allows SaltStack to trigger Slack alerts for changes or failures. Kuali also passes a test mode variable that performs a dry run and shows every change that would have been made if the code were enforced.

Salt Cloud modules

SaltStack provides out-of-the-box modules for many common AWS services along with the ability to customize existing modules or write new ones to meet specific needs. Engineers at Kuali can use SaltStack to manage and control any of the following AWS resources and the SRE team is able to add to and improve the list as often as needed:

- Users
- Codebuild
- ECS - tasks/services
- ECS - scaling groups
- EFS
- ELB
- ALB
- IAM policies
- IAM roles
- SSH keypairs
- VPC peers
- Routes
- S3
- Security groups
- Subnets
- Target groups (SaltStack capability)
- VPC

Deploying it in Docker

The resulting Docker image contains a Salt Minion (intelligent agent) and all of the SaltStack out-of-the-box modules and python libraries required for AWS. This containerized approach offers the Kuali team a way to provide a completely self-contained and portable tool that can be run from anywhere and has no dependencies.

The most common approach is to run the SaltStack docker container in AWS Codebuild. This allows the team to orchestrate a SaltStack test every time a PR is submitted. The results of the test are then passed to GitHub via webhooks to determine if there are any failures. If there are, the information is sent back to Codebuild and SaltStack, allowing SaltStack to create a Slack notification and Codebuild to create a record of the failure in AWS.

However, if a team prefers they can run the solution in a Codebuild alternative, such as CircleCI, Kubernetes, or even on an individual laptop.

The results

While the SRE team at Kuali will continue to iterate and improve their solution, they are satisfied that they've delivered automation for AWS that expands the company's ability to iterate faster and deliver better experiences to end users while ensuring that every managed environment remains optimized and secure. Here are a few of the immediate benefits Kuali experienced with SaltStack.

Version control and visibility

No more unplanned changes across AWS environments. Now every change is tracked and tested. Kuali SREs can now use SaltStack to track when changes are made, who made them, who approved them, and exactly what time they were done.

Portability and scale

Because each product team was independent, Kuali needed to make sure the solution was portable and could scale to an infinite number of AWS accounts. By combining a containerized deployment model with SaltStack speed and scale, both requirements are achieved.

Ease of use

One of the many benefits of SaltStack is its ability to act as an abstraction tool, abstracting away the complexities of any system it manages, including AWS. By utilizing SaltStack IaC through GitHub, that abstraction is taken a level further, allowing the teams to get all of the benefits and power of SaltStack without needing to interact with it directly. Instead, all changes can be made through a familiar and comfortable GitHub portal.

The SaltStack ability to test and report back infrastructure changes before enforcing the change is another substantial advantage.

Putting it all together

Companies worldwide choose SaltStack automation because it's flexible enough to meet the needs of any digital infrastructure — whether its built on cutting-edge tech or significant legacy investments — and powerful enough to manage the largest and most complex IT environments in the world.

For Kuali, the ability to define AWS infrastructure as code and automate it with a common toolset has allowed them to build superior applications and deliver best-in-class solutions to their higher-education customers.

© Copyright SaltStack, Inc. 2020

SaltStack, Inc.
2801 N Thanksgiving Way, #150
Lehi, UT 84043
USA

+1 801 207 7440
info@saltstack.com
www.saltstack.com

Produced in the United States of America

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. SaltStack products are warranted according to the terms and conditions of the agreements under which they are provided.

Statements regarding the future direction and intent of SaltStack are subject to change or withdrawal without notice, and represent goals and objectives only.

Please Recycle