WHITE PAPER

Using Digital Rebar to Create Composable Infrastructure Operations for Site Reliability Engineers

Authored by Rob Hirschfeld, Greg Althaus, and Victor Lowther

Executive Summary

Enterprises today face increasing diverse information technology (IT) choice and a resulting rise in complexity. The RackN Digital Rebar platform addresses this challenge by breaking large integrate automation challenges into manageable modular components. Our open software accelerates the rate at which site reliability engineering (SRE) or developer-operations (DevOps) teams can operate by helping them focus on the unique parts of their technology stack while reusing community best practices.

Introduction

In an economy where IT is core to the business, an application-centric development model is required to configure and deploy resources with the same amount of speed and agility as the applications it is hosting., Organizations are turning to platform as a service (PaaS), containers, software-defined networking, security and microservice technologies to facilitate this model. However, these technologies are heavily reliant on their underlying infrastructure. Thus, out of necessity, traditional developers must now specialize in storage, networking, security and compute infrastructure. The pace of innovation has put a tremendous amount of stress on data center operations (DC-OPS) teams that operate under the traditional information technology infrastructure library (ITIL) model.

To help increase this pace of change and adaptation, many organizations are adopting DevOps as a methodology for next-generation application delivery. DevOps involves continuous integration, automation and measurement and requires an agile infrastructure to keep up with the rapidly evolving environments of these applications. Provisioning and re-provisioning of resources must happen “on demand,” in a matter of seconds or minutes, not days or weeks. All of these business challenges and application demands have greatly increased IT complexity which makes IT more expensive, slower to execute and less secure. Therefore, IT organizations are now demanding infrastructure that is more flexible, efficient, and easier to manage in order to solve today’s problems.

The job function of the DevOps engineer was created to serve as a liaison between developers and DC-OPS teams. For many organizations without the proper operational discipline and focus, a DevOps engineer’s job function is too nebulous and tends to be more reactive than proactive. Benjamin Sloss Treynor, the Vice President of Engineering at Google, solved this problem at Google by establishing a

@rackngo
linkedin.com/company/rackn

rackn.com
rebar.digital

more operationally-focused job function called a Site Reliability Engineer (SRE)[1], the role of which will be explored below.

With the assumption that SREs cannot know everything about full-stack deployments, upgrades, operational differences and provisioning steps, the idea of Composable Infrastructure Operations was created in order to stem the tide of change. Forrester Research defines Composable Infrastructure as the logical combination of elements—such as CPU, memory, disk and networking—as a shared set of resources presented to the external environment as a functional system.[2] These systems must be programmatically composed and decomposed and be capable of sequencing resources appropriately in an automated and repeatable fashion.

Composable Operations enables arbitrary atomic operations to be composed into larger atomic operations. When discussing how to apply arbitrary infrastructure (cloud, hyperscale, servers and apps) to an operational workflow in a data center, it becomes critical to identify the "what" and the "when" in provisioning and management.

The RackN Digital Rebar platform fills a critical role for SRE teams by providing standard, cross-industry automation for routine tasks like hybrid provisioning and platform management. SRE teams cannot afford to waste time reinventing automation for common problems when they could focus their efforts on unique business value.

The Problem

From an operational standpoint, there are many significant challenges: a container management cluster on cloud or bare metal servers must be configured for a specific workload. This is a complex undertaking, especially when the SRE must account for the thousands of dependencies and steps related to what is required. Manual processes can take days—sometimes even weeks—of work to make a cluster fully operational. Even then, the infrastructure is never static. In the real world, clusters of solutions are constantly on an upgrade or improvement path. There is a continuous need to deploy new servers and cloud instances, add management capabilities and track upstream releases, all while keeping the cloud or cluster running and providing reliable services to end users. Thus, service continuity requirements dictate a need for automation and orchestration. There is no other way to reduce the cost of a cloud while simultaneously improving the cloud's uptime reliability.

SREs need to understand the implications of policy, template and, ultimately, application design at both the granular and stratospheric level. Further, they must also document and ensure that the end-to-end operational workflows and pipelines of the underlay and the platforms of the infrastructure are fully operational at all times.

How can IT organizations solve these problems with an appropriate budget of staff without expensive trial and error?

---

[1] "What Is 'Site Reliability Engineering'?" Google, n.d. Web. <https://landing.google.com/sre/interview/ben-treynor.html>.
[2] Fichera, Richard. "Composable Infrastructure: A Hardware Foundation For Extreme Service Agility." Forrester Research, Inc., 14 Apr. 2016. Web. 12 Jan. 2017. <http://reprints.forrester.com/#/assets/2/76/'RES132661'/reports>.

They should both hire people with an SRE operational mindset and make sure they have access to tools, like the RackN Digital Rebar platform, that eliminate wasted effort on reinventing automation to handle common tasks.

What is an SRE?

SREs are similar to DevOps engineers; however, the SRE focuses more the operational aspect of delivery and possesses deep systems and software knowledge with a strong focus on operational delivery of the end-to-end solution. An SRE does everything from setting up a deployment pipeline and maintaining a CI/CD server to ensuring logging, errors and remediation steps are included in a Production Readiness Review (PRR).

An SRE contributes to the business by helping to reduce overall cost through tracking resources and mean time to repair (MTTR), as well as reducing deployment time with automation and development time by finding pre-existing solutions that might not otherwise be seen.

Seven key areas of SRE focus:

1. **Reliability:** The SRE must build reliable systems.
2. **Monitoring:** The SRE must implement performance metrics and establish benchmarks in order to monitor the systems.
3. **Alerting:** The SRE must catch any issues and ensure that there is a closed-loop support process in place to solve them.
4. **Infrastructure:** The SRE must be highly informed on cloud and physical infrastructure scalability and limitations.
5. **Product Engineering:** The SRE must understand all product requirements, code-level, testing and readiness needs.
6. **Debugging:** The SRE must understand the systems, log files, code, use-case and troubleshooting, so they can debug as needed.
7. **Security:** The SRE must understand common security issues, as well as tracking and addressing vulnerabilities, to ensure the systems are properly secured.

What is Composable Infrastructure Operations?

Many IT organizations today are constrained by complex, siloed infrastructures that were initially designed to help reduce the costs of traditional architectures and applications. These infrastructures are not equipped to provide the degree of flexibility and efficiency required for the next-generation applications that are now central to business success. Further, due to operational ideology understanding and differences, departmental collaboration (security, networking, server and applications) is also hamstrung.

Software-based composable infrastructure operations[3] offers the potential to provide the best application and infrastructure performance possible. This is achieved by automating tasks that reduce the under-utilization and over-provisioning of infrastructure leading to more agile, cost-effective use of infrastructure. The ongoing, dynamic configuration and reconfiguration of a large number of infrastructure assets has the potential to increase IT complexity. Therefore, an intelligent, automated and

[3] "The Journey to Composable Infrastructure." Moor Insights & Strategy, Nov. 2015. Web. <http://www.moorinsightsstrategy.com/wp-content/uploads/2015/11/The-Journey-to-Composable-Infrastructure-by-Moor-Insights-and-Strategy.pdf>.

robust composable systems management framework is required to help address this potential complexity.

Using composability to manage complexity was a top design priority for the RackN team while designing Digital Rebar.

Proper composition and orchestration allow higher-level platform frameworks (e.g. OpenStack, Mesos and Kubernetes) to manage pools of resources dynamically without interrupting ongoing service operations. Whereas composition occurs at the subsystem level through the use of profiles, orchestration occurs across infrastructure domains. Together, composition and orchestration enable different devices and applications to work together seamlessly in a coordinated fashion. Composition and orchestration capabilities must enable consistent application of all policies across infrastructure resources.

Through this approach, IT is able to define workflow, resource requirements and policies in advance in order to drive "infrastructure-as-code" and keep applications dynamically optimized. Condition-based actions and action synchronization eliminate guesswork and manual processes, preventing potential operational slowdowns and increases in long-term operational costs.

What is Digital Rebar™?

Digital Rebar is an open source container-based, infrastructure-as-code provisioning and orchestration framework that provides automated infrastructure and software deployment capabilities sponsored by RackN. Digital Rebar enables upgrade and continuous deployment automation, which is important for large, production scale deployments of complex and evolving technologies that sit above the infrastructure, such as Docker, Kubernetes, Mesos, Cloud Foundry, OpenStack, and Ceph. Digital Rebar reduces the human resource cost of orchestrating continuous hardware and cloud redeployment, hypervisor, operating systems and application layer software installation and management. All of this adds up to significant gains in operational reliability, consistency and a corresponding reduction in defect handling costs.

The Digital Rebar Architecture

Digital Rebar delivers repeatable best practice deployments. Digital Rebar is not just about installation: we define success as a sustainable operations model in which we continuously improve how people use their infrastructure. The complexity and pace of technology change is accelerating, so we must have an approach that embraces continuous delivery in the data center. Digital Rebar's objective is to help both technology and operators become more efficient, stable, and resilient over time.

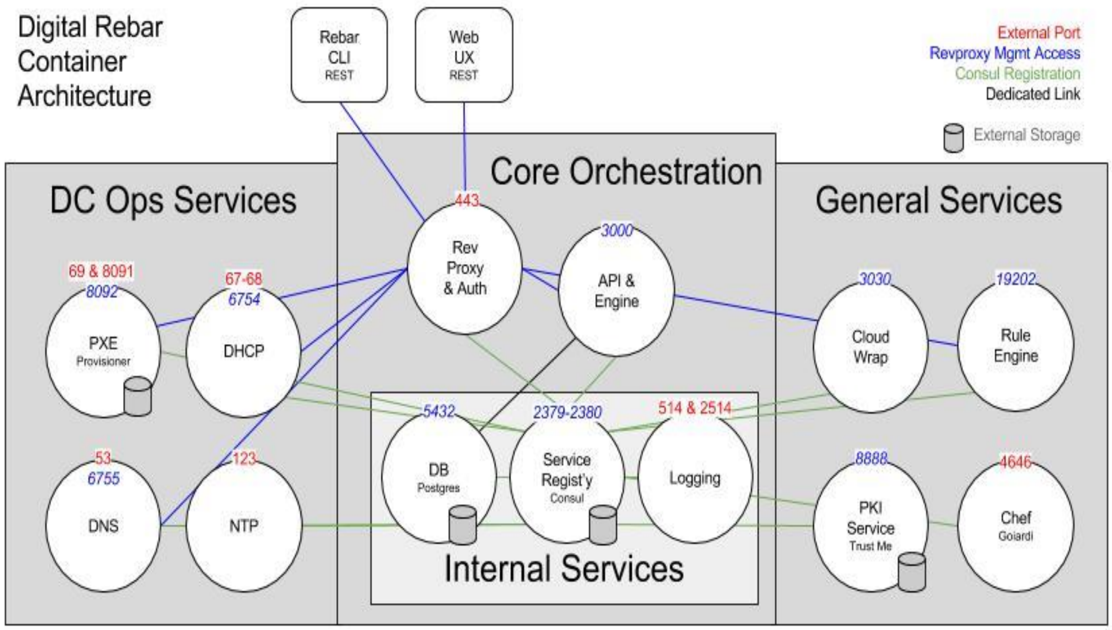

Figure 1. Digital Rebar Core Architecture

As a microservices application, Digital Rebar is composed of multiple, independent services that work together to provide a composed, orchestrated experience. Digital Rebar uses Compose as a service registry so that the services can find each other. However, this is only a small part of making the architecture work. Most of the Digital Rebar services have both internal and externally exposed components.

Our services—such as DNS, DHCP, PKI and provisioning—must expose those services in standard ways in order for Digital Rebar to do its job. Digital Rebar has its own application programing interface (API) and user interface (UI), as well as internal services such as the PostgreSQL database and Consul. Digital Rebar also has control APIs to manage services based on system orchestration. The control APIs must be protected by a consistent security policy (both authentication and authorization) that spans all the APIs.

Digital Rebar uses a reverse proxy (which is also a Digital Rebar container) to secure the environment. A reverse proxy acts as a single gateway for all the service APIs. When a request comes into the gateway, it is inspected and authenticated, allowing the system to provide delegated single sign on (SSO) for users and authorization for access based on user policy. This allows us to centralize user security in such a way that it does not become a burden on service authors.

Key Digital Rebar Concepts

Attribute Injection

Attribute Injection is an essential aspect of the Functional Operations story that helps clean the boundaries needed to implement consistent scripting behavior between divergent sites. Functional Operations is similar in concept to Functional Programing,[4] where coding blocks can be atomic units. It is the architectural pattern that allows Digital Rebar to abstract and isolate provisioning layers.

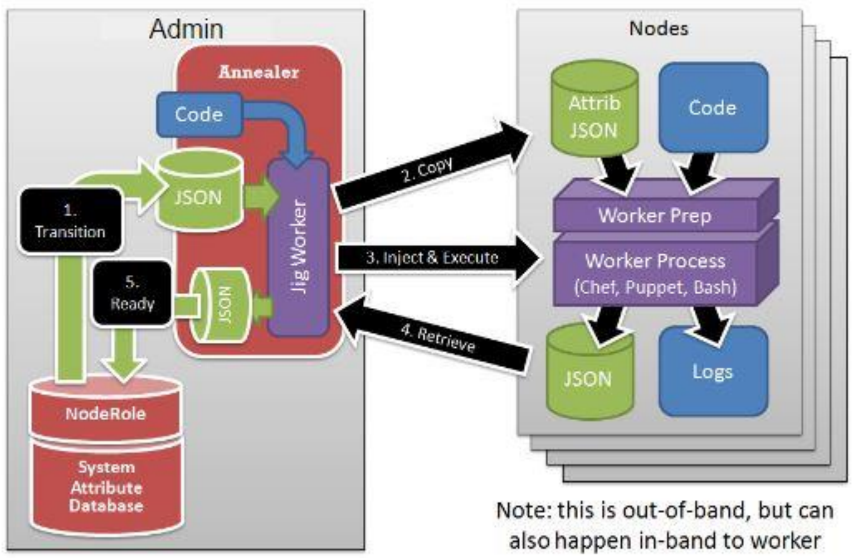

Figure 2. Attribute Injection

This operational approach means that deployments are composed of layered services instead of locked “golden” images. The layers can be maintained independently and allow users to compose specific configurations à la carte. This approach works because the layers have clean functional boundaries that can be scoped and managed atomically.

The Annealer

The Annealer combines the best of directed graphs with late binding and parallel execution. The Annealer design is loosely based on the computer science practice of Simulated Annealing,[5] in that the Digital Rebar Annealer is iterative and seeks to find a stable state. We believe annealing is the key ingredient for repeatable and shared code upgrades and on-going operations as opposed to installers that focus on a single ending state.

To understand the Digital Rebar Annealer, we have to break it down into three distinct components: deployment timeline, annealing, and node-role state.

[4] See https://en.wikipedia.org/wiki/Functional_programming

[5] See https://en.wikipedia.org/wiki/Simulated_annealing

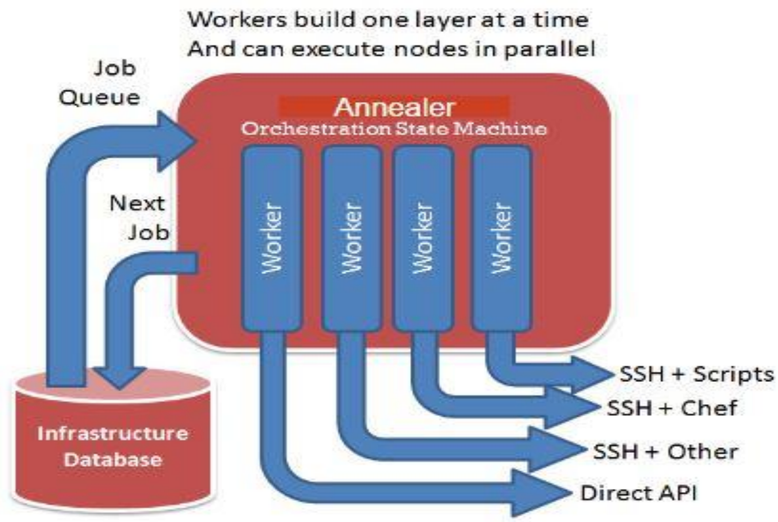

Figure 3. The Annealer

The deployment timeline represents externally (user, hardware, etc) initiated changes that propose a new target state. Once that new target is committed, Digital Rebar anneals by iterating through all the node-roles in a reasonable order. As the Annealer runs the node-roles, they update their own state. The aggregate state of all the node-roles determines the state of the deployment. In Digital Rebar, a deployment is a combination of user and system defined states. Digital Rebar's job is to stabilize the deployments and maintain the desired configuration over time.

The Next Step

RackN provides support for Digital Rebar components or as an integrated solution per node or virtual instance. In addition, RackN provides consulting and integration services to help enterprises realize the vision of SRE readiness by creating fully-automated operational workflow pipelines on any infrastructure or cloud.

To get started, contact us at sre@rackn.com or download Digital Rebar at http://rebar.digital